

Lihui Wang · Amos H. C. Ng  
Kalyanmoy Deb *Editors*

# Multi-objective Evolutionary Optimisation for Product Design and Manufacturing

 Springer

# Multi-objective Evolutionary Optimisation for Product Design and Manufacturing

Lihui Wang · Amos H. C. Ng  
Kalyanmoy Deb  
Editors

# Multi-objective Evolutionary Optimisation for Product Design and Manufacturing

Prof. Lihui Wang  
Virtual Systems Research Centre  
University of Skövde  
PO Box 408  
541 28 Skövde  
Sweden  
e-mail: lihui.wang@his.se

Prof. Kalyanmoy Deb  
Department of Mechanical Engineering  
Indian Institute of Technology  
Kanpur, Uttar Pradesh 208016  
India  
e-mail: deb@iitk.ac.in

Assoc. Prof. Amos H. C. Ng  
Virtual Systems Research Centre  
University of Skövde  
PO Box 408  
541 28 Skövde  
Sweden  
e-mail: amos.ng@his.se

ISBN 978-0-85729-617-7  
DOI 10.1007/978-0-85729-652-8  
Springer London Dordrecht Heidelberg New York

e-ISBN 978-0-85729-652-8

British Library Cataloguing in Publication Data  
A catalogue record for this book is available from the British Library

© Springer-Verlag London Limited 2011

Autodesk, AutoCAD and Inventor are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries.

SuperStir is a trademark of ESAB Holdings Ltd, 322 High Holborn, London WC1V 7PB, United Kingdom.

The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

*Cover design:* eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))



# Preface

Product design and manufacturing are tightly connected to innovation. They have thus been the key areas that support and influence a nation's economy since the eighteenth century. As the primary driving force behind economic growth, design and manufacturing serve as the foundation of and contribute to other industries, with products ranging from heavy-duty machinery to hi-tech home electronics. In the past centuries, they have contributed significantly to modern civilisation and created the momentum that drives today's economy. Despite various achievements, we are still facing challenges due to the growing complexity in design and manufacturing.

The complexity in product design and manufacturing becomes obvious when solving problems simultaneously against multiple objectives that conflict to each other. In solving such problems, with or without the presence of constraints, it gives rise to a set of trade-off optimal solutions, popularly known as Pareto-optimal solutions. Due to the multiplicity in solutions, these problems were proposed to be solved suitably using evolutionary algorithms that use a population approach in its search procedure. Nevertheless, multi-objective evolutionary optimisation problems remain highly challenging in product design and manufacturing with increasing complexity. Designers and engineers across organisations often find themselves in situations that demand advanced optimisation capability when dealing with their daily activities related to product design and manufacturing.

Targeting the challenge in solving complex problems, over the past decades, researchers have focused their efforts on multi-objective evolutionary approaches to improving the optimality of solutions. While these efforts have resulted in a large volume of publications and impacted both present and future practices in design and manufacturing, there still exists a gap in the literature for a focused collection of works dedicated to multi-objective evolutionary optimisation. To bridge this gap and present the state-of-the-art to a much broad readership, from academic researchers to practicing engineers, is the primary motivation behind this book.

The first three chapters form Part-1 of the book on literature survey and trends. [Chapter 1](#) begins with a clear definition of multi-objective optimisation. Based on a comparative analysis of the existing literature, this chapter provides an introduction to the operating principles of evolutionary optimisation and outlines the current research and application studies in both single- and multi-objective decision making. The chapter also highlights some research trends, particularly the issues of handling uncertainties, dynamic problems, many objectives, decision making, and knowledge discovery through the recently proposed *innovization* approach. The discussion on multi-objective optimisation is extended in [Chap. 2](#) to supply chains. Supply chains are in general complex networks composed of autonomous entities whereby multiple performance measures in different levels have to be taken into account. Particularly, it reviews the research and practices of the existing multi-objective optimisation applications, both analysis- and simulation-based, in supply chain management. This chapter also identifies the needs of an integration of multi-objective optimisation and system dynamics models and presents a case study on its application to the investigation of bullwhip effects in a supply chain. [Chapter 3](#) then introduces a unique perspective of state-of-the-art in multi-objective optimisation based on thermo-mechanical simulations. This perspective is reinforced through two case studies of friction stir welding and metal casting. Future challenges are also identified at the end of the chapter.

Part-2 of the book focuses on product design and optimisation, and is constituted from four chapters. Recognising the importance of optimisation in product family design, [Chap. 4](#) presents a novel approach based on multi-objective evolutionary optimisation and visual analytics to resolve trade-offs between commonality and performance objectives when designing a family of products. A design example of a family of aircraft with a 10-objective trade-off is provided to validate this approach. Based on the functional behaviour in product family design, [Chap. 5](#) introduces a product family hierarchy, where designs can be classified into phenomenological design family, functional part family and embodiment part family. Product portfolio selection is then possible after identifying and clustering non-dominated solutions. [Chapter 6](#) applies the product family design concept to a family of industrial robots. The design problem is treated as a multi-objective optimisation problem where a Pareto optimal front is used to visualise the trade-off between commonality and performance of individual family members. In the area of rapid prototyping using the fused deposition method (FDM), [Chap. 7](#) depicts a unique approach to simultaneously minimising two conflicting goals—average surface roughness and build time. Within the context, a comparative study between genetic algorithm and particle swarm optimisation is also conducted.

Optimisation issues in process planning and scheduling are covered in [Chaps. 8](#) through [12](#), and organised into Part-3 of the book. [Chapter 8](#) utilises ant colony optimisation for automatic machining setup planning of cast parts. It simultaneously considers the selection of available machines, tolerance analysis and cost modelling for achieving an optimal setup planning result. A tolerance cost factor is introduced when machining error stack-up occurs. The ant colony optimisation is extended in [Chap. 9](#) to include a preference vector when searching for a set of

Pareto-optimal scheduling solutions using meta-heuristics. The scheduling problem is to minimise make-span and energy consumption, whereas the preference vector allows the search to focus on specific areas of interest to decision makers instead of searching for the entire Pareto frontier. However, in order to greatly improve the performance of a manufacturing system, the scheduling problem is better integrated with process planning. This issue is dealt with in [Chap. 10](#) using a multi-agent approach that optimises the two functions simultaneously based on particle swarm optimisation. The feasibility and performance of this approach is verified through a comparative analysis against simulated annealing and genetic algorithm, with positive outcomes. The agent-based approach is also adopted in [Chap. 11](#) for real-time scheduling, whereas reinforcement learning is implemented to job agents and resource agents in order to improve their coordination processes. Two case studies are performed to verify the effectiveness of the proposed method in dynamic shop environment. However, operation disruptions often occur on dynamic shop floors, which increase manufacturing complexity and trigger frequent rescheduling. Targeting the problem, [Chap. 12](#) introduces a multiple ant colony optimisation approach to minimise changes during rescheduling while searching for trade-offs between time and cost.

In Part-4 of the book, the aspect of systems design and analysis is shared by [Chaps. 13–17](#). Dynamic operations not only demand for rescheduling but also affect shop floor layout. The latter is the focal point of [Chap. 13](#), looking into a hybrid approach for dynamic assembly shop layout. In this case, genetic algorithm is used to search for an optimal new layout if the change can justify a significant relocation cost. Otherwise, a function block-based approach is utilised to find the best routing of assembly jobs under a new condition but the existing layout. The multi-objective facility layout issue is further examined in [Chap. 14](#) using a simulation-based optimisation approach where a genetic algorithm helps generate new design parameters for optimisation. [Chapter 15](#) addresses a production system design problem by integrating the concept of innovization with discrete-event simulation and data mining techniques. The uniqueness of the integrated approach lies on applying data mining to the data sets generated from simulation-based multi-objective optimisation, in order to automatically or semi-automatically discover and interpret the hidden relationships and patterns for optimal production system analysis. An industrial case study of an automotive assembly line improvement is also presented to validate the new method. In reality, a good system design requires a good system optimisation, particularly from a cost perspective. [Chapter 16](#) thus proposes to expand simulation-based optimisation and post-optimality analysis to cover the cost aspects of a production system, such as investments and running cost. Industrial empirical results indicate that this approach has opened up the opportunity to identify a set of design solutions with great financial improvement, which are otherwise not feasible to be explored by using current industrial procedures. Another application domain of multi-objective optimisation is manufacturing supply chain. [Chapter 17](#) addresses the design of supply chain networks including both network configuration and related operational decisions such as order splitting, transportation allocation and inventory

control. The goal is to achieve the best compromise between cost and customer service level. To illustrate its effectiveness, the proposed methodology is applied to two real-life case studies from automotive industry and textile industry.

All together, the seventeen chapters provide an overview of some recent R&D achievements of multi-objective evolutionary optimisation applied to product design and manufacturing. We believe that this research field will continue to be active for years to come.

Finally, the co-editors would like to take this opportunity express their deep appreciation to all the authors for their significant contributions to this book. Their commitment, enthusiasm, and technical expertise are what made this book possible. We are also grateful to Springer for supporting this project, and would especially like to thank Anthony Doyle, Senior Editor for Engineering, and Claire Protherough, Senior Editorial Assistant, for their constructive assistance and earnest cooperation, both with the publishing venture in general and the editorial details. We hope that readers find this book informative and useful.

Skövde, Sweden, May 2011  
Kanpur, India, May 2011

Lihui Wang and Amos H. C. Ng  
Kalyanmoy Deb

# Contents

## Part I Literature Survey and Trends

<b>1 Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction</b> . . . . .	3
Kalyanmoy Deb	
<b>2 Multi-objective Optimisation in Manufacturing Supply Chain Systems Design: A Comprehensive Survey and New Directions</b> . . . . .	35
Tehseen Aslam, Philip Hedenstierna, Amos H. C. Ng and Lihui Wang	
<b>3 State-of-the-Art Multi-objective Optimisation of Manufacturing Processes Based on Thermo-Mechanical Simulations</b> . . . . .	71
Cem Celal Tutum and Jesper Hattel	

## Part II Product Design and Optimisation

<b>4 Many-Objective Evolutionary Optimisation and Visual Analytics for Product Family Design</b> . . . . .	137
Ruchit A. Shah, Patrick M. Reed and Timothy W. Simpson	
<b>5 Product Portfolio Selection of Designs Through an Analysis of Lower-Dimensional Manifolds and Identification of Common Properties</b> . . . . .	161
Madan Mohan Dabbeeru, Kalyanmoy Deb and Amitabha Mukerjee	

<b>6</b>	<b>Multi-objective Optimisation of a Family of Industrial Robots . . .</b>	<b>189</b>
	Johan Ölvander, Mehdi Tarkian and Xiaolong Feng	
<b>7</b>	<b>Multi-objective Optimisation and Multi-criteria Decision Making for FDM Using Evolutionary Approaches . . . . .</b>	<b>219</b>
	Nikhil Padhye and Kalyanmoy Deb	
 <b>Part III Process Planning and Scheduling</b>		
<b>8</b>	<b>A Setup Planning Approach Considering Tolerance Cost Factors . . . . .</b>	<b>251</b>
	Binfang Wang and A. Y. C. Nee	
<b>9</b>	<b>Preference Vector Ant Colony System for Minimizing Make-span and Energy Consumption in a Hybrid Flow Shop . . .</b>	<b>279</b>
	Bing Du, Huaping Chen, George Q. Huang and H. D. Yang	
<b>10</b>	<b>Intelligent Optimisation for Integrated Process Planning and Scheduling . . . . .</b>	<b>305</b>
	Weidong Li, Lihui Wang, Xinyu Li and Liang Gao	
<b>11</b>	<b>Distributed Real-Time Scheduling by Using Multi-agent Reinforcement Learning . . . . .</b>	<b>325</b>
	Koji Iwamura and Nobuhiro Sugimura	
<b>12</b>	<b>A Multiple Ant Colony Optimisation Approach for a Multi-objective Manufacturing Rescheduling Problem . . . . .</b>	<b>343</b>
	Vikas Kumar, Nishikant Mishra, Felix T. S. Chan, Niraj Kumar and Anoop Verma	
 <b>Part IV Systems Design and Analysis</b>		
<b>13</b>	<b>Reconfigurable Facility Layout Design for Job-Shop Assembly Operations . . . . .</b>	<b>365</b>
	Lihui Wang, Shadi Keshavarzmanesh and Hsi-Yung Feng	
<b>14</b>	<b>A Simulation Optimisation Framework for Container Terminal Layout Design . . . . .</b>	<b>385</b>
	Loo Hay Lee, Ek Peng Chew, Kee Hui Chua, Zhuo Sun and Lu Zhen	

**15 Simulation-Based Innovization Using Data Mining for Production Systems Analysis. . . . . 401**  
Amos H. C. Ng, Catarina Dudas, Johannes Nießen and Kalyanmoy Deb

**16 Multi-objective Production Systems Optimisation with Investment and Running Cost . . . . . 431**  
Leif Pehrsson, Amos H. C. Ng and Jacob Bernedixen

**17 Supply Chain Design Using Simulation-Based NSGA-II Approach. . . . . 455**  
Lyes Benyoucef and Xiaolan Xie

**Index . . . . . 493**

# Contributors

**Tehseen Aslam** Virtual Systems Research Centre, University of Skövde, PO Box 408, 541 28 Skövde, Sweden

**Lyes Benyoucef** INRIA, COSTEAM Project, ISGMP Bat. A, Ile du Saulcy, Metz 57000, France

**Jacob Bernedixen** Virtual Systems Research Centre, University of Skövde, PO Box 408, 541 28 Skövde, Sweden

**F. T. S. Chan** Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, China

**Huaping Chen** School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

**Ek Peng Chew** Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore

**Kee Hui Chua** Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore

**Madan Mohan Dabbeeru** Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, Uttar Pradesh, 208016, India

**Kalyanmoy Deb** Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, Uttar Pradesh, 208016, India

**Bing Du** School of Management, University of Science and Technology of China, Hefei, China

**Catarina Dudas** Virtual Systems Research Centre, University of Skövde, PO Box 408, 541 28 Skövde, Sweden

**Hsi-Yung Feng** Department of Mechanical Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada



**Xiaolong Feng** ABB Corporate Research, 721 78 Västerås, Sweden

**Liang Gao** State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074 China

**Jesper Hattel** Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark

**Philip Hedenstierna** Virtual Systems Research Centre, University of Skövde, PO Box 408, 541 28 Skövde, Sweden

**George Q. Huang** Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong, China

**Koji Iwamura** Graduate School of Engineering, Osaka Prefecture University, Sakai, Osaka, 599-8531, Japan

**Shadi Keshavarzmanesh** Department of Mechanical and Materials Engineering, The University of Western Ontario, London, ON N6A 5B9, Canada

**V. Kumar** Department of Management, Dublin City University Business School, Dublin, Ireland

**Loo Hay Lee** Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore

**Weidong Li** Faculty of Engineering and Computing, Coventry University, Coventry CV1 5FB, UK

**Xinyu Li** State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

**N. Mishra** School of Management and Business, Aberystwyth University, Ceredigion, SY23 2AX, UK

**Amitabha Mukerjee** Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur, Uttar Pradesh 208016, India

**Andrew Y. C. Nee** Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore

**Amos H. C. Ng** Virtual Systems Research Centre, University of Skövde, PO Box 408, 541 28 Skövde, Sweden

**Johannes Nießen** Virtual Systems Research Centre, University of Skövde, PO Box 408, 541 28 Skövde, Sweden

**Johan Ölvander** Department of Management and Engineering, Linköping University, 581 83 Linköping, Sweden

**Nikhil Padhye** Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

**Leif Pehrsson** Volvo Car Corporation, 405 31 Göteborg, Sweden

**Patrick M. Reed** Civil and Environmental Engineering, Pennsylvania State University, University Park, USA

**Ruchit A. Shah** Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, USA

**Timothy W. Simpson** Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, USA

**Nobuhiro Sugimura** Graduate School of Engineering, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

**Zhuo Sun** Centre for Maritime Studies, National University of Singapore, 12 Prince George's Park, Singapore

**Mehdi Tarkian** Department of Management and Engineering, Linköping University, 581 83 Linköping, Sweden

**Cem Celal Tutum** Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark

**A. Verma** Computer Aided Manufacturing Laboratory, Department of Mechanical Engineering, University of Cincinnati, Cincinnati, OH 45221, USA

**Binfang Wang** Institute of High Performance Computing, Agency for Science, Technology and Research, 1 Fusionopolis Way, #16-16, Singapore

**Lihui Wang** Virtual Systems Research Centre, University of Skövde, PO Box 408, 541 28 Skövde, Sweden

**Xiaolan Xie** ENSM.SE, 158 Cours Fauriel, 42023 Saint-Etienne cedex 2, France

**H. D. Yang** School of Automation, South China University of Technology, Guangzhou, China

**Lu Zhen** Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore

**Part I**  
**Literature Survey and Trends**

# Chapter 1

## Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction

Kalyanmoy Deb

**Abstract** As the name suggests, multi-objective optimisation involves optimising a number of objectives simultaneously. The problem becomes challenging when the objectives are of conflicting characteristics to each other, that is, the optimal solution of an objective function is different from that of the other. In the course of solving such problems, with or without the presence of constraints, these problems give rise to a set of trade-off optimal solutions, popularly known as Pareto-optimal solutions. Because of the multiplicity in solutions, these problems were proposed to be solved suitably using evolutionary algorithms using a population approach in its search procedure. Starting with parameterized procedures in early 90s, the so-called evolutionary multi-objective optimisation (EMO) algorithms is now an established field of research and application with many dedicated texts and edited books, commercial softwares and numerous freely downloadable codes, a biannual conference series running successfully since 2001, special sessions and workshops held at all major evolutionary computing conferences, and full-time researchers from universities and industries from all around the globe. In this chapter, we provide a brief introduction to its operating principles and outline the current research and application studies of evolutionary multi-objective optimisation (EMO).

---

K. Deb (✉)

Department of Mechanical Engineering, Indian Institute of Technology,  
Kanpur, Uttar Pradesh 208016, India

e-mail: deb@iitk.ac.in

URL: <http://www.iitk.ac.in/kangal/deb.htm>

## 1.1 Introduction

In the past 15 years, EMO has become a popular and useful field of research and application. Evolutionary optimisation (EO) algorithms use a population-based approach in which more than one solution participates in an iteration and evolves a new population of solutions in each iteration. The reasons for their popularity are many: (i) EOs do not require any derivative information, (ii) EOs are relatively simple to implement, and (iii) EOs are flexible and have a wide-spread applicability. For solving single-objective optimisation problems, particularly in finding a single optimal solution, the use of a population of solutions may sound redundant, in solving multi-objective optimisation problems an EO procedure is a perfect choice [1]. The multi-objective optimisation problems, because their attributes, give rise to a set of Pareto-optimal solutions, which need further processing to arrive at a single preferred solution. To achieve the first task, it becomes quite a natural proposition to use an EO, because the use of population in an iteration helps an EO to simultaneously find multiple non-dominated solutions, which portrays a trade-off among objectives, in a single simulation run.

In this chapter, we present a brief description of an evolutionary optimisation procedure for single-objective optimisation. Thereafter, we describe the principles of EMO. Then, we discuss some salient developments in EMO research. It is clear from these discussions that EMO is not only being found to be useful in solving multi-objective optimisation problems, it is also helping to solve other kinds of optimisation problems more efficiently than they are traditionally solved. As a by-product, EMO-based solutions are helping to elicit very valuable insight about a problem—a which is difficult to achieve otherwise. EMO procedures with a decision making concept are discussed as well. Some of these ideas require further detailed studies and this chapter mentions some such topics for current and future research in this direction.

## 1.2 Evolutionary Optimisation for Single-Objective Optimisation

EO principles are different from classical optimisation methodologies in the following main ways [2]:

- An EO procedure does not usually use gradient information in its search process. Thus, EO methodologies are direct search procedures, allowing them to be applied to a wide variety of optimisation problems.
- An EO procedure uses more than one solution (a *population* approach) in an iteration, unlike in most classical optimisation algorithms which updates one solution in each iteration (a *point* approach). The use of a population has a number of advantages: (i) it provides an EO with a parallel processing power

achieving a computationally quick overall search, (ii) it allows an EO to find multiple optimal solutions, thereby facilitating the solution of multi-modal and multi-objective optimisation problems, and (iii) it provides an EO with the ability to normalise decision variables (as well as objective and constraint functions) within an evolving population using the population-best minimum and maximum values.

- An EO procedure uses stochastic operators, unlike deterministic operators used in most classical optimisation methods. The operators tend to achieve a desired effect by using higher probabilities towards desirable outcomes, as opposed to using predetermined and fixed transition rules. This allows an EO algorithm to negotiate multiple optima and other complexities better and provide them with a global perspective in their search. An EO begins its search with a population of solutions usually created at random within a specified lower and upper bound on each variable. Thereafter, the EO procedure enters into an iterative operation of updating the current population to create a new population by the use of four main operators: selection, crossover, mutation and elite-preservation. The operation stops when one or more pre-specified termination criteria are met.

The initialisation procedure usually involve a random creation of solutions. If in a problem the knowledge of some good solutions is available, it is better to use such information in creating the initial population. Elsewhere [3], it is highlighted that for solving complex real-world optimisation problems, such a customised initialisation is useful and also helpful in achieving a faster search. After the population members are evaluated, the selection operator chooses above-average (in other words, better) solutions with a larger probability to fill an intermediate mating pool. For this purpose, several stochastic selection operators have been developed as discussed in the EO literature. In its simplest form (called the *tournament* selection [4]), two solutions can be picked at random from the evaluated population and the better of the two (in terms of its evaluated order) can be picked.

The ‘variation’ operator is a collection of a number of operators (such as crossover, mutation, etc.) which are used to generate a modified population. The purpose of the crossover operator is to pick two or more solutions (parents) randomly from the mating pool and create one or more solutions by exchanging information among the parent solutions. The crossover operator is applied with a crossover probability ( $p_c \in [0, 1]$ ), indicating the proportion of population members participating in the crossover operation. The remaining ( $1 - p_c$ ) proportion of the population is simply copied to the modified (child) population. In the context of real-parameter optimisation having  $n$  real-valued variables and involving a crossover with two parent solutions, such that each variable may be crossed at a time. A probability distribution which depends on the difference between the two parent variable values is often used to create two new numbers as child values around the two parent values [5]. Besides the variable-wise recombination operators, vector-wise recombination operators also suggested to propagate the correlation among variables of parent solutions to the created child solutions [6, 7].

Each child solution, created by the crossover operator, is then perturbed in its vicinity by a mutation operator [2]. Every variable is mutated with a mutation probability  $p_m$ , usually set as  $1/n$  ( $n$  is the number of variables), so that on an average one variable gets mutated per solution. In the context of real-parameter optimisation, a simple Gaussian probability distribution with a predefined variance can be used with its mean at the child variable value [1]. This operator allows an EO to search locally around a solution and is independent on the location of other solutions in the population.

The elitism operator combines the old population with the newly created population and chooses to keep better solutions from the combined population. Such an operation makes sure that an algorithm has a monotonically non-degrading performance. Rudolph [8] proved an asymptotic convergence of a specific EO but having elitism and mutation as two essential operators.

Finally, the user of an EO needs to choose termination criteria. Often, a pre-determined number of generations is used as a termination criterion. For goal attainment problems, an EO can be terminated as soon as a solution with a pre-defined goal or a target solution is found. In many studies [2, 9–11], a termination criterion based on the statistics of the current population vis-a-vis that of the previous population to determine the rate of convergence is used. In other more recent studies, theoretical optimality conditions (such as the extent of satisfaction of Karush–Kuhn–Tucker (KKT) conditions) are used to determine the termination of a real-parameter EO algorithm [12]. Although EOs are heuristic based, the use of such theoretical optimality concepts in an EO can also be used to test their converging abilities towards local optimal solutions.

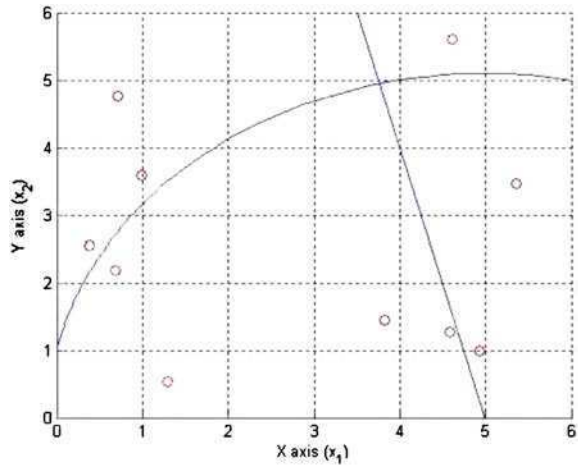
To demonstrate the working of the above-mentioned GA, we show four snapshots of a typical simulation run on the following constrained optimisation problem:

$$\begin{aligned}
 &\text{Minimise } f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \\
 &\text{subject to } g_1(x) \equiv 26 - (x_1 - 5)^2 - x_2^2 \geq 0, \\
 &\quad g_2(x) \equiv 20 - 4x_1 - x_2 \geq 0, \\
 &\quad 0 \leq (x_1, x_2) \leq 6.
 \end{aligned} \tag{1.1}$$

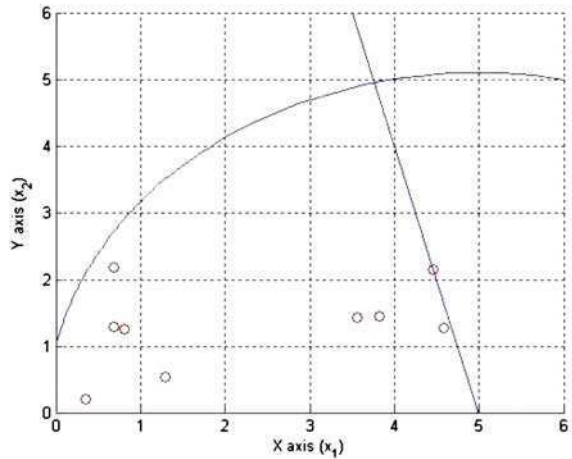
Ten points are used and the GA is run for 100 generations. The SBX recombination operator is used with probability of  $p_c = 0.9$  and index  $\eta_c = 10$ . The polynomial mutation operator is used with a probability of  $p_m = 0.5$  with an index of  $\eta_m = 50$ . Figures 1.1, 1.2, 1.3 and 1.4 show the populations at generation zero, 5, 40 and 100, respectively. It can be observed that in only five generations, all 10 population members become feasible. Thereafter, the points come close to each other and creep towards the constrained minimum point.

The EA procedure is a population-based stochastic search procedure which iteratively emphasises its better population members, uses them to recombine and perturb locally in the hope of creating new and better populations until a pre-defined termination criterion is met. The use of a population helps to achieve an

**Fig. 1.1** Initial population



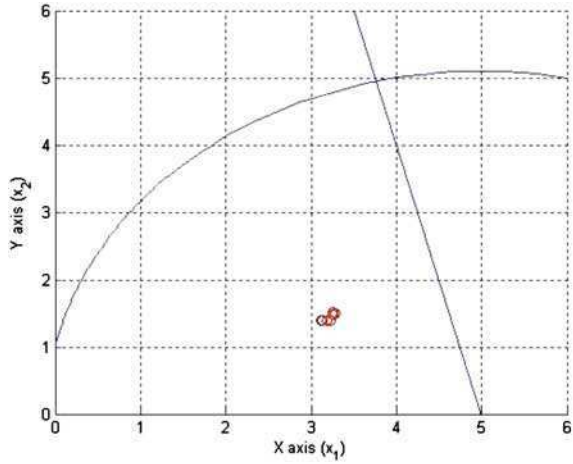
**Fig. 1.2** Population at generation 5



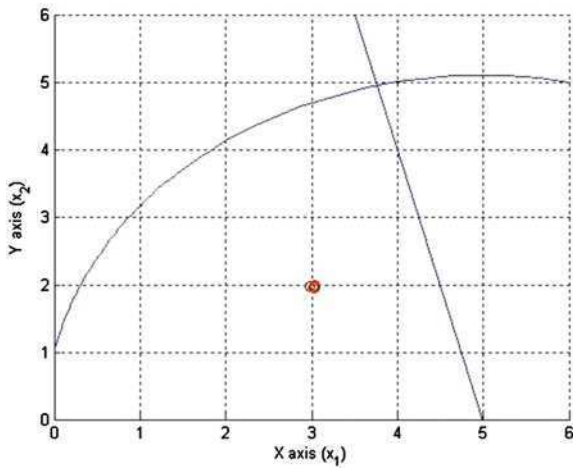
*implicit parallelism* [2, 13, 14] in an EO's search mechanism (causing an inherent parallel search in different regions of the search space), a process which makes an EO computationally attractive for solving difficult problems. In the context of certain Boolean functions, a computational time saving to find the optimum varying polynomial to the population size is proven [15]. On the one hand, the EO procedure is flexible, thereby allowing a user to choose suitable operators and problem-specific information to suit a specific problem. On the other hand, the flexibility comes with the onus on the part of a user to choose appropriate and tangible operators so as to create an efficient and consistent search [16]. However, the benefits of having a flexible optimisation procedure, over their more rigid and specific optimisation algorithms, provide feasibility in solving difficult real-world optimisation problems involving non-differentiable objectives and constraints,



**Fig. 1.3** Population at generation 40



**Fig. 1.4** Population at generation 100



non-linearities, discreteness, multiple optima, large problem sizes, uncertainties in computation of objectives and constraints, uncertainties in decision variables, mixed type of variables, and others.

A wiser approach to solving optimisation problems of the real world would be to first understand the niche of both EO and classical methodologies and then adopt hybrid procedures employing the better of the two as the search progresses over varying degrees of search-space complexity from start to finish. As demonstrated in the above typical GA simulation, there are two phases in the search of a GA. First, the GA exhibits a more *global* search by maintaining a diverse population, thereby discovering potentially good regions of interest. Second, a more *local* search takes place by bringing the population members closer together. Although the above GA degenerates to both these search phases automatically without any external intervention, a more efficient search can be achieved if the

later local search phase can be identified and executed with a more specialized local search algorithm.

### 1.3 Evolutionary Multi-objective Optimisation

A multi-objective optimisation problem involves a number of objective functions which are to be either minimised or maximised. As in a single-objective optimisation problem, the multi-objective optimisation problem may contain a number of constraints which any feasible solution (including all optimal solutions) must satisfy. Since objectives can be either minimised or maximised, we state the multi-objective optimisation problem in its general form:

$$\left. \begin{array}{ll} \text{Minimise/Maximise} & f_m(\mathbf{x}), \quad m = 1, 2, \dots, M; \\ \text{subject to} & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, J; \\ & h_k(\mathbf{x}) = 0, \quad k = 1, 2, \dots, K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n. \end{array} \right\} \quad (1.2)$$

A solution  $\mathbf{x} \in \mathbf{R}^n$  is a vector of  $n$  decision variables:  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ . The solutions satisfying the constraints and variable bounds constitute a *feasible decision variable space*  $S \subset \mathbf{R}^n$ . One of the striking differences between single-objective and multi-objective optimisation is that in multi-objective optimisation the objective functions constitute a multi-dimensional space, in addition to the usual decision variable space. This additional  $M$ -dimensional space is called the *objective space*,  $Z \subset \mathbf{R}^M$ . For each solution  $\mathbf{x}$  in the decision variable space, there exists a point  $\mathbf{z} \in \mathbf{R}^M$  in the objective space, denoted by  $\mathbf{f}(\mathbf{x}) = \mathbf{z} = (z_1, z_2, \dots, z_M)^T$ . To make the descriptions clear, we refer a ‘solution’ as a variable vector and a ‘point’ as the corresponding objective vector.

The optimal solutions in multi-objective optimisation can be defined from a mathematical concept of *partial ordering*. In the parlance of multi-objective optimisation, the term *domination* is used for this purpose. In this section, we restrict ourselves to discuss unconstrained (without any equality, inequality or bound constraints) optimisation problems. The domination between two solutions is defined as follows [1, 17]:

**Definition 1** A solution  $\mathbf{x}^{(1)}$  is said to dominate the other solution  $\mathbf{x}^{(2)}$ , if both the following conditions are true:

1. The solution  $\mathbf{x}^{(1)}$  is no worse than  $\mathbf{x}^{(2)}$  in all objectives. Thus, the solutions are compared based on their objective function values (or location of the corresponding points  $\mathbf{z}^{(1)}$  and  $\mathbf{z}^{(2)}$  on the objective space).
2. The solution  $\mathbf{x}^{(1)}$  is strictly better than  $\mathbf{x}^{(2)}$  in at least one objective.

For a given set of solutions (or corresponding points on the objective space, for example, those shown in Fig. 1.5a), a pair-wise comparison can be made using the

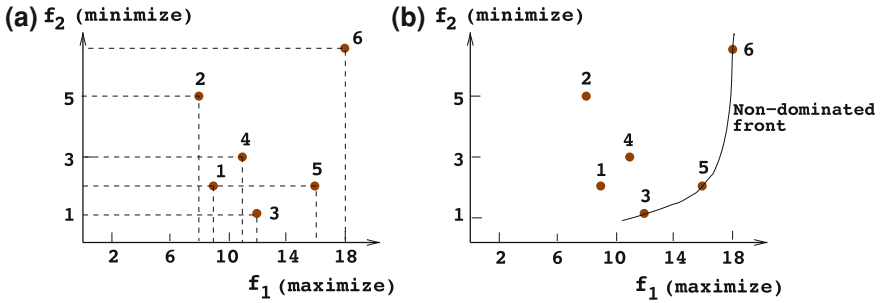


Fig. 1.5 A set of points and the first non-domination front are shown

above definition and whether one point dominates the other can be established. All points which are not dominated by any other member of the set are called the non-dominated points of class one, or simply the non-dominated points. For the set of six solutions shown in the figure, they are points 3, 5, and 6. One property of any two such points is that a gain in an objective from one point to the other happens only because of a sacrifice in at least one other objective. This *trade-off* property between the non-dominated points makes the practitioners interested in finding a wide variety of them before making a final choice. These points make up a front when they are viewed together on the objective space; hence the non-dominated points are often visualized to represent a *non-domination front*. The computational effort needed to select the points of the non-domination front from a set of  $N$  points is  $O(N \log N)$  for two and three objectives, and  $O(N \log^{M-2} N)$  for  $M > 3$  objectives [18].

With the above concept, now it is easier to define the *Pareto-optimal solutions* in a multi-objective optimisation problem. If the given set of points for the above task contain all points in the search space (assuming a countable number), the points lying on the non-domination front, by definition, do not get dominated by any other point in the objective space, hence are Pareto-optimal points (together they constitute the Pareto-optimal front) and the corresponding pre-images (decision variable vectors) are called Pareto-optimal solutions. However, more mathematically elegant definitions of Pareto-optimality (including the ones for continuous search space problems) exist in the multi-objective literature [17, 19].

### 1.3.1 Principle of EMO's Search

In the context of multi-objective optimisation, the extremist principle of finding the optimum solution cannot be applied to any one particular objective alone, when the rest of the objectives are also important. Different solutions may produce trade-offs (conflicting outcomes among objectives) among different objectives. A solution that is extreme (in a better sense) with respect to one objective requires a

compromise in other objectives. This prohibits one to choose a solution which is optimal with respect to only one objective. This clearly suggests two ideal goals of multi-objective optimisation:

1. Find a set of solutions which lie on the Pareto-optimal front, and
2. Find a set of solutions which are diverse enough to represent the entire range of the Pareto-optimal front. EMO algorithms attempt to follow both the above principles similar to the other a posteriori multiple criteria decision making (MCDM) methods (refer to this chapter).

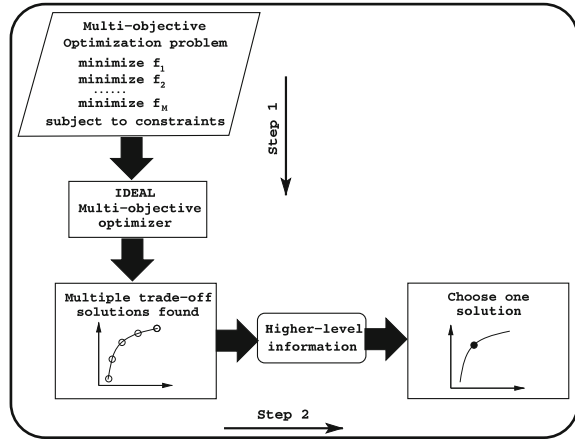
Although one fundamental difference between single and multiple objective optimisation lies in the cardinality in the optimal set, from a practical standpoint a user needs only one solution, no matter whether the associated optimisation problem is single or multi-objective. The user is now in a dilemma. As a number of solutions are optimal, the obvious question arises: Which of these optimal solutions must one choose? This is not an easy question to answer. It involves higher-level information which is often non-technical, qualitative and experience-driven. However, if a set of many trade-off solutions are already worked out or available, one can evaluate the pros and cons of each of these solutions based on all such non-technical and qualitative, yet important, considerations and compare them to make a choice. Thus, in a multi-objective optimisation, ideally the effort must be made in finding the set of trade-off optimal solutions by considering all objectives to be important. After a set of such trade-off solutions are found, a user can then use higher-level qualitative considerations to make a choice. As an EMO procedure deals with a population of solutions in every iteration, it makes them intuitive to be applied in multi-objective optimisation to find a set of non-dominated solutions. Like other a posteriori MCDM methodologies, an EMO based procedure works with the following principle in handling multi-objective optimisation problems:

*Step 1.* Find multiple non-dominated points as close to the Pareto-optimal front as possible, with a wide trade-off among objectives.

*Step 2.* Choose one of the obtained points using higher-level information.

Figure 1.6 shows schematically the principles, followed in an EMO procedure. As EMO procedures are heuristic based, they may not guarantee in finding Pareto-optimal points, as a theoretically provable optimisation method would do for tractable (for example, linear or convex) problems. But EMO procedures have essential operators to constantly improve the evolving non-dominated points (from the point of view of convergence and diversity discussed above) similar to the way most natural and artificial evolving systems continuously improve their solutions. To this effect, a recent simulation study [12] has demonstrated that a particular EMO procedure, starting from random non-optimal solutions, can progress towards theoretical KKT points with iterations in real-valued multi-objective optimisation problems. The main difference and advantage of using an EMO compared with a posteriori MCDM procedures is that multiple trade-off solutions can be found in a single simulation run, as most a posteriori MCDM methodologies would require multiple applications.

**Fig. 1.6** Schematic of a two-step multi-objective optimisation procedure

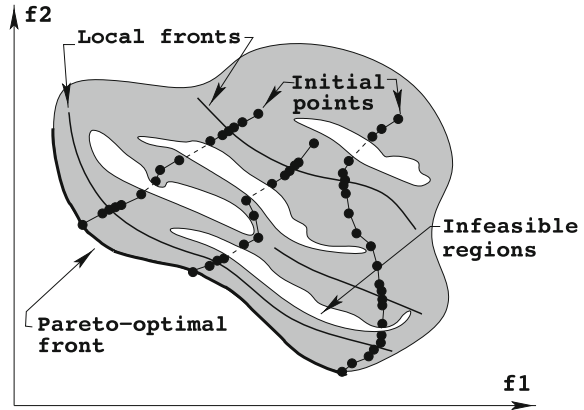


In Step 1 of the EMO-based multi-objective optimisation (the task shown vertically downwards in Fig. 1.6), multiple trade-off, non-dominated points are found. Thereafter, in Step 2 (the task shown horizontally, towards the right), higher-level information is used to choose one of the obtained trade-off points. This dual task allows an interesting feature, if applied for solving single-objective optimisation problems. It is easy to realize that a single-objective optimisation is a degenerate case of multi-objective optimisation, as shown in details in another study [20]. In the case of single-objective optimisation having only one globally optimal solution, Step 1 will ideally find only one solution, thereby not requiring us to proceed to Step 2. However, in the case of single-objective optimisation having multiple global optima, both steps are necessary to first find all or multiple global optima, and then to choose one solution from them by using a higher-level information about the problem. Thus, although seems ideal for multi-objective optimisation, the framework suggested in Fig. 1.6 can be ideally thought as a generic principle for both single and multiple objective optimisation.

### 1.3.2 Generating Classical Methods and EMO

In the generating MCDM approach, the task of finding multiple Pareto-optimal solutions is achieved by executing many independent single-objective optimisations, each time finding a single Pareto-optimal solution. A parametric scalarizing approach (such as the weighted-sum approach,  $\epsilon$ -constraint approach, and others) can be used to convert multiple objectives into a parametric single-objective function. By simply varying the parameters (weight vector or  $\epsilon$ -vector) and optimising the scalarised function, different Pareto-optimal solutions can be found. In contrast, in an EMO, multiple Pareto-optimal solutions are attempted to be found in a single simulation by emphasizing multiple non-dominated and isolated

**Fig. 1.7** Generative MCDM methodology employs multiple, independent single-objective optimisations

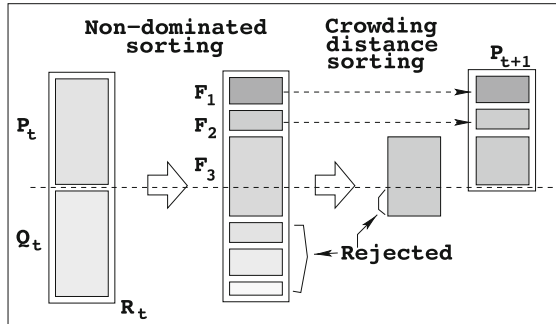


solutions. We discuss a little later some EMO algorithms describing how such dual emphasis is provided, but now discuss qualitatively the difference between a posteriori MCDM and EMO approaches.

Consider Fig. 1.7, in which we sketch how multiple independent parametric single-objective optimisations may find different Pareto-optimal solutions. The Pareto-optimal front corresponds to global optimal solutions of several scalarised objectives. However, during the course of an optimisation task, algorithms must overcome a number of difficulties, such as infeasible regions, local optimal solutions, flat regions of objective functions, isolation of optimum, etc., to converge to the global optimal solution. Moreover, because of practical limitations, an optimisation task must also be completed in a reasonable computational time. This requires an algorithm to strike a good balance between the extent of these tasks its search operators must do to overcome the above-mentioned difficulties reliably and quickly. When multiple simulations are performed to find a set of Pareto-optimal solutions, the above balancing act must have to be performed in every single simulation. Since simulations are performed independently, no information about the success or failure of previous simulations is used to speed up the process. In difficult multi-objective optimisation problems, such memory-less a posteriori methods may demand a large overall computational overhead to get a set of Pareto-optimal solutions. Moreover, even though the convergence can be achieved in some problems, independent simulations can never guarantee finding a good distribution among obtained points.

EMO, as mentioned earlier, constitutes an inherent parallel search. When a population member overcomes certain difficulties and make a progress towards the Pareto-optimal front, its variable values and their combination reflect this fact. When a recombination takes place between this solution and other population members, such valuable information of variable value combinations gets shared through variable exchanges and blending, thereby making the overall task of finding multiple trade-off solutions a parallelly processed task.

**Fig. 1.8** Schematic of the NSGA-II procedure



### 1.3.3 Elitist Non-dominated Sorting GA or NSGA-II

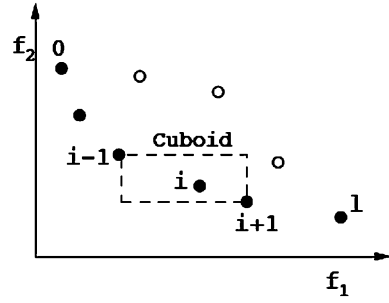
The NSGA-II procedure [21] is one of the popularly used EMO procedures which attempt to find multiple Pareto-optimal solutions in a multi-objective optimisation problem and has the following three features:

1. it uses an elitist principle,
2. it uses an explicit diversity preserving mechanism, and
3. it emphasises non-dominated solutions.

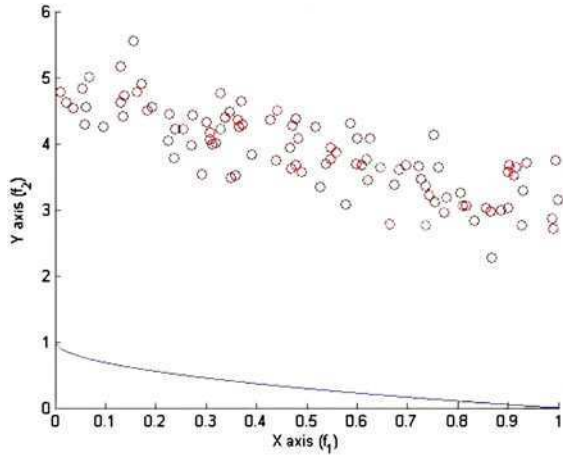
At any generation  $t$ , the offspring population (say,  $Q_t$ ) is first created by using the parent population (say,  $P_t$ ) and the usual genetic operators. Thereafter, the two populations are combined together to form a new population (say,  $R_t$ ) of size  $2N$ . Then, the population  $R_t$  is classified into different non-domination classes. Thereafter, the new population is filled by points of different non-domination fronts, one at a time. The filling starts with the first non-domination front (of class one) and continues with points of the second non-domination front, and so on. Since the overall population size of  $R_t$  is  $2N$ , not all fronts can be accommodated in  $N$  slots available for the new population. All fronts which could not be accommodated are deleted. When the last allowed front is being considered, there may exist more points in the front than the remaining slots in the new population. This scenario is illustrated in Fig. 1.8. Instead of arbitrarily discarding some members from the last front, the points which will make the diversity of the selected points the highest are chosen.

The crowded-sorting of the points of the last front which could not be accommodated fully is achieved in the descending order of their *crowding distance values* and points from the top of the ordered list are chosen. The crowding distance  $d_i$  of point  $i$  is a measure of the objective space around  $i$  which is not occupied by any other solution in the population. Here, we simply calculate this quantity  $d_i$  by estimating the perimeter of the cuboid (Fig. 1.9) formed by using the nearest neighbors in the objective space as the vertices (we call this the *crowding distance*).

**Fig. 1.9** The crowding distance calculation



**Fig. 1.10** Initial population



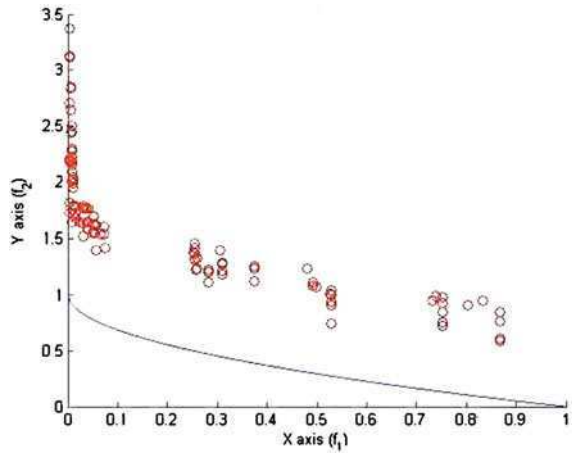
Next, we show snapshots of a typical NSGA-II simulation on a two-objective test problem:

$$\text{ZDT2} : \begin{cases} \text{Minimize} & f_1(\mathbf{x}) = x_1, \\ \text{Minimize} & f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})}], \\ \text{where} & g(\mathbf{x}) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i \\ & 0 \leq x_1 \leq 1, \\ & -1 \leq x_i \leq 1, \quad i = 2, 3, \dots, 30. \end{cases} \quad (1.3)$$

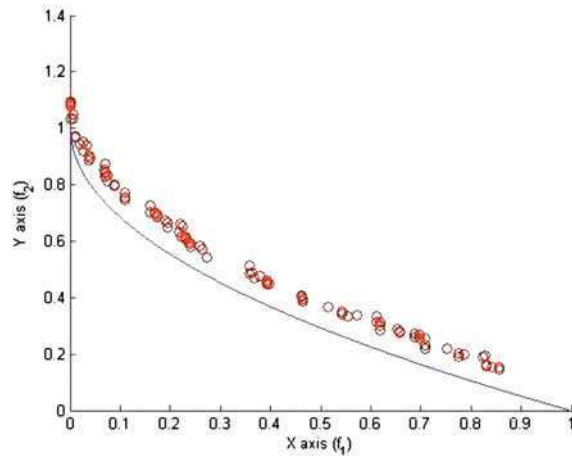
NSGA-II is run with a population size of 100 and for 100 generations. The variables are used as real numbers and an SBX recombination operator with  $p_c = 0.9$  and distribution index of  $\eta_c = 10$  and a polynomial mutation operator [1] with  $p_m = 1/n$  ( $n$  is the number of variables) and distribution index of  $\eta_m = 20$  are used. Figure 1.10 is the initial population shown on the objective space. Figures 1.11, 1.12 and 1.13 show populations at generations 10, 30 and 100, respectively. The figures illustrates how the operators of NSGA-II cause the population to move towards the Pareto-optimal front with generations. At generation 100, the population comes very close to the true Pareto-optimal front.



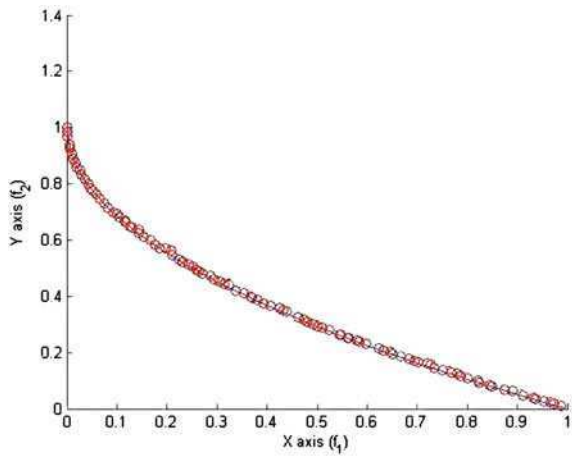
**Fig. 1.11** Population at generation 10



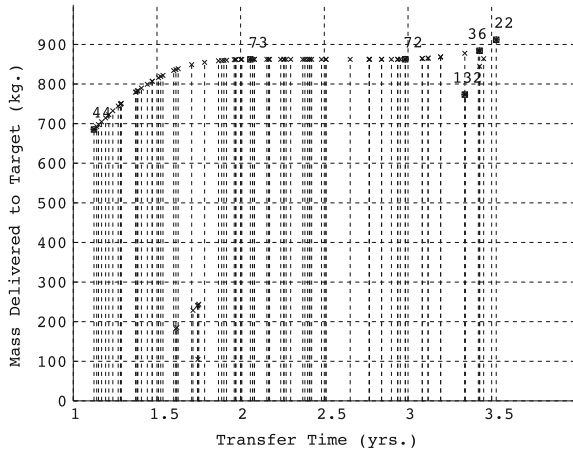
**Fig. 1.12** Population at generation 30



**Fig. 1.13** Population at generation 100



**Fig. 1.14** Obtained non-dominated solutions using NSGA



## 1.4 Applications of EMO

Since the early development of EMO algorithms in 1993, they have been applied to many real-world and interesting optimisation problems. Descriptions of some of these studies can be found in books [1, 22, 23], dedicated conference proceedings [24–27], and domain-specific books, journals and proceedings. In this section, we describe one case study which clearly demonstrates the EMO philosophy which we described in Sect. 1.3.1.

### 1.4.1 Spacecraft Trajectory Design

Coverstone-Carroll et al. [28] proposed a multi-objective optimisation technique using the original non-dominated sorting algorithm (NSGA) [29] to find multiple trade-off solutions in a spacecraft trajectory optimisation problem. To evaluate a solution (trajectory), the SEPTOP (Solar Electric Propulsion Trajectory optimisation) software [30] is called for, and the delivered payload mass and the total time of flight are calculated. The multi-objective optimisation problem has eight decision variables controlling the trajectory, three objective functions: (i) maximize the delivered payload at destination, (ii) maximize the negative of the time of flight, and (iii) maximize the total number of heliocentric revolutions in the trajectory, and three constraints limiting the SEPTOP convergence error and minimum and maximum bounds on heliocentric revolutions.

On the Earth–Mars rendezvous mission, the study found interesting trade-off solutions [28]. Using a population of size 150, the NSGA was run for 30 generations. The obtained non-dominated solutions are shown in Fig. 1.14 for two of the three objectives and some selected solutions are shown in Fig. 1.15. It is clear that there exist short-time flights with smaller delivered payloads (solution marked 44)

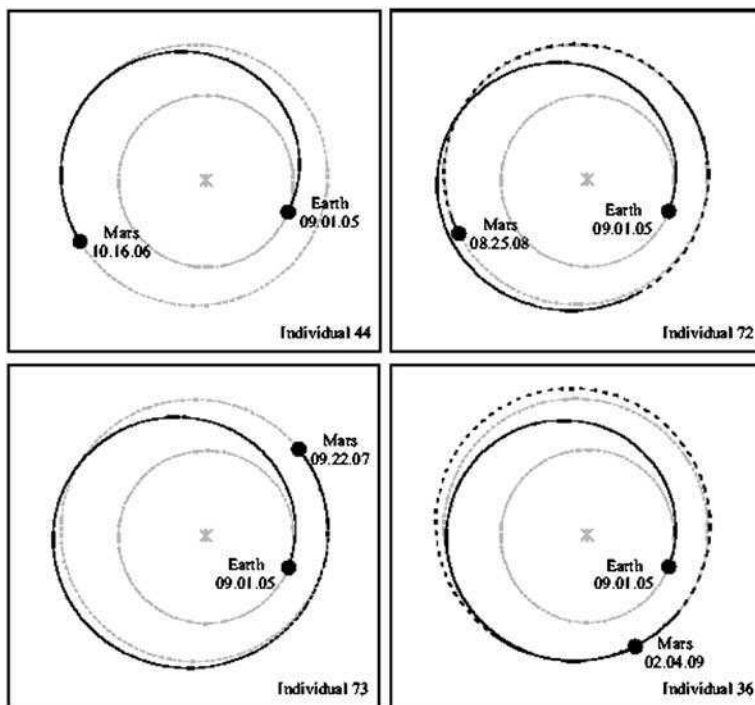
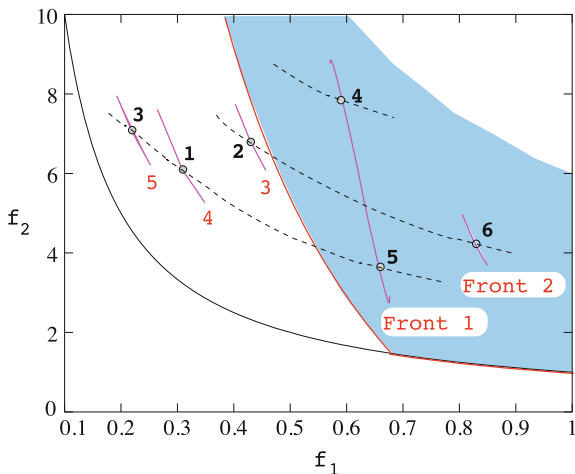


Fig. 1.15 Four trade-off trajectories

and long-time flights with larger delivered payloads (solution marked 36). Solution 44 can deliver a mass of 685.28 kg and requires about 1.12 years. On other hand, an intermediate solution 72 can deliver almost 862 kg with a travel time of about 3 years. In these figures, each continuous part of a trajectory represents a *thrusting* arc and each dashed part of a trajectory represents a *coasting* arc. It is interesting to note that only a small improvement in delivered mass occurs when comparing the solutions 73 and 72 with a sacrifice in flight time of about an year.

The multiplicity in trade-off solutions, as depicted in Fig. 1.15, is what we envisaged in discovering in a multi-objective optimisation problem by using a posteriori procedure, such as an EMO algorithm. This aspect was also discussed in Fig. 1.6. Once such a set of solutions with a good trade-off among objectives is obtained, one can analyze them for choosing a particular solution. For example, in this problem context, it makes sense to not choose a solution between points 73 and 72 attributable to poor trade-off between the objectives in this range. On the other hand, choosing a solution within points 44 and 73 is worthwhile, but which particular solution to choose depends on other mission related issues. But by first finding a wide range of possible solutions and revealing the shape of front, EMO can help narrow down the choices and allow a decision maker to make a better decision. Without the knowledge of such a wide variety of trade-off solutions, a

**Fig. 1.16** Non-constrained-domination fronts



proper decision-making may be a difficult task. Although one can choose a scalarised objective (such as the  $\epsilon$ -constraint method with a particular  $\epsilon$  vector) and find the resulting optimal solution, the decision-maker will always wonder what solution would have been derived if a different  $\epsilon$  vector was chosen. For example, if  $\epsilon_1 = 2.5$  years is chosen and mass delivered to the target is maximised, a solution in between points 73 and 72 will be found. As discussed earlier, this part of the Pareto-optimal front does not provide the best trade-offs between objectives that this problem can offer. A lack of knowledge of good trade-off regions before a decision is made may allow the decision maker to settle for a solution which, although optimal, may not be a good compromised solution. The EMO procedure allows a flexible and a pragmatic procedure for finding a well-diversified set of solutions simultaneously so as to enable picking a particular region for further analysis or a particular solution for implementation.

### 1.5 Constraint Handling in EMO

The constraint handling method modifies the binary tournament selection, where two solutions are picked from the population and the better solution is chosen. In the presence of constraints, each solution can be either feasible or infeasible. Thus, there may be at most three situations: (i) both solutions are feasible, (ii) one is feasible and other is not, and (iii) both are infeasible. We consider each case by simply redefining the domination principle as follows (we call it the *constrained-domination* condition for any two solutions  $\mathbf{x}^{(i)}$  and  $\mathbf{x}^{(j)}$ ):

**Definition 2** A solution  $\mathbf{x}^{(i)}$  is said to ‘constrained-dominate’ a solution  $\mathbf{x}^{(j)}$  (or  $\mathbf{x}^{(i)} \preceq_c \mathbf{x}^{(j)}$ ), if any of the following conditions are true:

1. Solution  $\mathbf{x}^{(i)}$  is feasible and solution  $\mathbf{x}^{(j)}$  is not.
2. Solutions  $\mathbf{x}^{(i)}$  and  $\mathbf{x}^{(j)}$  are both infeasible, but solution  $\mathbf{x}^{(i)}$  has a smaller constraint violation, which can be computed by adding the normalised violation of all constraints:

$$CV(\mathbf{x}) = \sum_{j=1}^J \langle \bar{g}_j(\mathbf{x}) \rangle + \sum_{k=1}^K \text{abs}(\bar{h}_k(\mathbf{x})),$$

where  $\langle \alpha \rangle$  is  $-\alpha$ , if  $\alpha < 0$  and is zero, otherwise. The normalization is achieved with the population minimum ( $\langle g_j \rangle_{\min}$ ) and maximum ( $\langle g_j \rangle_{\max}$ ) constraint violations:  $\bar{g}_j(\mathbf{x}) = (\langle g_j(\mathbf{x}) \rangle - \langle g_j \rangle_{\min}) / (\langle g_j \rangle_{\max} - \langle g_j \rangle_{\min})$ .

3. Solutions  $\mathbf{x}^{(i)}$  and  $\mathbf{x}^{(j)}$  are feasible and solution  $\mathbf{x}^{(i)}$  dominates solution  $\mathbf{x}^{(j)}$  in the usual sense (Definition 1).

The above change in the definition requires a minimal change in the NSGA-II procedure described earlier. Figure 1.16 shows the non-domination fronts on a six-membered population because of the introduction of two constraints (the minimization problem is described as CONSTR elsewhere [1]). In the absence of the constraints, the non-domination fronts (shown by dashed lines) would have been ((1,3,5), (2,6), (4)), but in their presence, the new fronts are ((4,5), (6), (2), (1), (3)). The first non-domination front consists of the ‘best’ (that is, non-dominated and feasible) points from the population and any feasible point lies on a better non-domination front than an infeasible point.

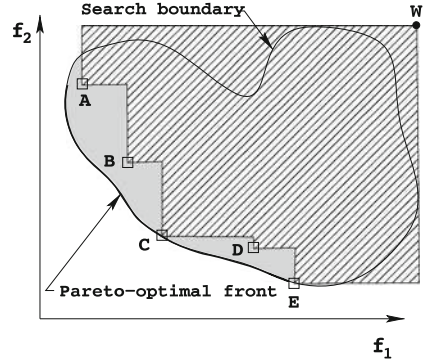
## 1.6 Performance Measures Used in EMO

There are two goals of an EMO procedure: (i) a good convergence to the Pareto-optimal front and (ii) a good diversity in obtained solutions. As both are conflicting in nature, comparing two sets of trade-off solutions also require different performance measures. In the early years of EMO research, three different sets of performance measures were used:

1. metrics evaluating convergence to the known Pareto-optimal front (such as error ratio, distance from reference set, etc.),
2. metrics evaluating spread of solutions on the known Pareto-optimal front (such as spread, spacing, etc.), and
3. metrics evaluating certain combinations of convergence and spread of solutions (such as hypervolume, coverage, R-metrics, etc.).

A detailed study [31] comparing most existing performance metrics based on out-performance relations has concluded that R-metrics suggested by [32] are the best. However, a study has argued that a single unary performance measure (any of the first two metrics described above in the enumerated list) cannot adequately determine a true winner, as both aspects of convergence and diversity cannot be

**Fig. 1.17** The hypervolume enclosed by the non-dominated solutions



measured by a single performance metric [33]. That study also concluded that binary performance metrics (indicating usually two different values when a set of solutions  $A$  is compared with  $B$  and  $B$  is compared with  $A$ ), such as epsilon-indicator, binary hypervolume indicator, utility indicators R1 to R3, etc., are better measures for multi-objective optimisation. The flip side is that the binary metrics computes  $M(M - 1)$  performance values for two algorithms in an  $M$ -objective optimisation problem, by analysing all pair-wise performance comparisons, thereby making them difficult to use in practice. In addition, unary and binary attainment indicators of [34, 35] are of great importance. Figures 1.17 and 1.18 illustrate the hypervolume and attainment indicators. Attainment surface is useful to determine a representative front obtained from multiple runs of an EMO algorithm. In general, 50% surface can be used to indicate the front that is dominated by 50% of all obtained non-dominated points.

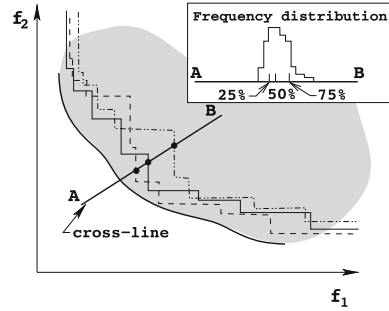
## 1.7 EMO and Decision-Making

Finding a set of representative Pareto-optimal solutions using an EMO procedure is only half the task; choosing a single preferred solution from the obtained set is also an equally important task. There are three main directions of developments in this direction.

In the a priori approach, preference information of a decision-maker (DM) is used to focus the search effort into a part of the Pareto-optimal front, instead of the entire frontier. For this purpose, a reference point approach [36], a reference direction approach [37], ‘light beam’ approach [38], etc. have been incorporated in a NSGA-II procedure to find a preferred part of the Pareto-optimal frontier.

In the a posteriori approach, preference information is used after a set of representative Pareto-optimal solutions are found by an EMO procedure. The MCDM approaches including reference point method, Tschebyscheff metric method, etc. [17] can be used. This approach is now believed to be applicable only to two, three or at most four-objective problems. As the number of objectives increase, EMO

**Fig. 1.18** The attainment surface is created for a number of non-dominated solutions



methodologies exhibit difficulties in converging close to the Pareto-optimal front and the a posteriori approaches become a difficult proposition.

In the interactive approach, decision maker (DM) preference information is integrated to an EMO algorithm during the optimisation run. In the progressively interactive EMO approach [39], the DM is called after every  $\tau$  generations and is presented with a few well-diversified solutions chosen from the current non-dominated front. The DM is then asked to rank the solutions according to preference. The information is then processed through an optimisation task to capture DM's preference using an utility function. This utility function is then used to drive NSGA-II's search till the procedure is repeated in the next DM call.

The decision-making procedure integrated with an EMO procedure makes the multi-objective optimisation procedure complete. More such studies must now be executed to make EMO more usable in practice.

## 1.8 Multi-objectivisation

Interestingly, the act of finding multiple trade-off solutions using an EMO procedure has found its application outside the realm of solving multi-objective optimisation problems per se. The concept of finding multiple trade-off solutions using an EMO procedure is applied to solve other kinds of optimisation problems that are otherwise not multi-objective in nature. For example, the EMO concept is used to solve constrained single-objective optimisation problems by converting the task into a two-objective optimisation task of additionally minimizing an aggregate constraint violation [40]. This eliminates the need to specify a penalty parameter while using a penalty based constraint handling procedure. A recent study [41] utilises a bi-objective NSGA-II to find a Pareto-optimal frontier corresponding to minimizations of the objective function and constraint violation. The frontier is then used to estimate an appropriate penalty parameter, which is then used to formulate a penalty based local search problem and is solved using a classical optimisation method. The approach is shown to require an order or two

magnitude less function evaluations than the existing constraint handling methods on a number of standard test problems.

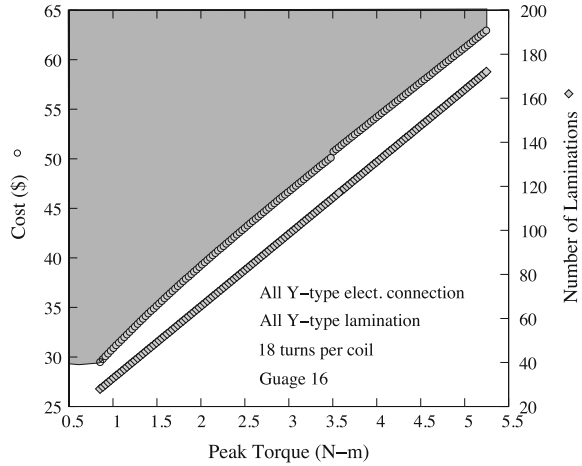
A well-known difficulty in genetic programming studies, called the ‘bloating’, arises because of the continual increase in size of genetic programs with iteration. The reduction of bloating by minimizing the size of programs as an additional objective helped find high-performing solutions with a smaller size of the code [42]. Minimizing the intra-cluster distance and maximizing inter-cluster distance simultaneously in a bi-objective formulation of a clustering problem is found to yield better solutions than the usual single-objective minimization of the ratio of the intra-cluster distance to the inter-cluster distance [43]. A recently published book [44] describes many such interesting applications in which EMO methodologies have helped solve problems which are otherwise (or traditionally) not treated as multi-objective optimisation problems.

### ***1.8.1 Knowledge Discovery Through EMO***

One striking difference between a single-objective optimisation and multi-objective optimisation is the cardinality of the solution set. In the latter, multiple solutions are the outcome and each solution is theoretically an optimal solution corresponding to a particular trade-off among the objectives. Thus, if an EMO procedure can find solutions close to the true Pareto-optimal set, what we have in our hand are a number of high-performing solutions trading-off the conflicting objectives considered in the study. As they are all near optimal, these solutions can be analyzed for finding properties which are common to them. Such a procedure can then become a systematic approach in deciphering important and hidden properties which optimal and high-performing solutions must have for that problem. In a number of practical problem-solving tasks, the so-called *innovation* procedure is shown to find important insight into high-performing solutions [45]. Figure 1.19 shows that of the five decision variables involved in an electric motor design problem involving minimum cost and maximum peak-torque, four variables have identical values for all Pareto-optimal solutions [46]. Of the two allowable electric connections, the ‘Y’-type connection; of three laminations, ‘Y’-type lamination; of 10–80 different turns, 18 turns, and of 16 different wire sizes, 16-gauge wire remain common to all Pareto-optimal solutions. The only way the solutions differ, relates to having different number of laminations. In fact, for a motor having more peak-torque, a linearly increasing number of laminations becomes a recipe for optimal more design. Such useful properties are expected to exist in practical problems, as they follow certain scientific and engineering principles at the core, but finding them through a systematic scientific procedure had not been paid much attention in the past. The principle of first searching for multiple trade-off and high-performing solutions using a multi-objective optimisation procedure and then analysing them to discover useful knowledge certainly remains a viable way forward. The current efforts [47] to automate the knowledge



**Fig. 1.19** Innovization study of an electric motor design problem



extraction procedure through a sophisticated data-mining task is promising and should make the overall approach more appealing to the practitioners.

## 1.9 Hybrid EMO Procedures

The search operators used in EMO are generic. There is no guarantee that an EMO will find any Pareto-optimal solution in a finite number of solution evaluations for a randomly chosen problem. However, as discussed above, EMO methodologies provide adequate emphasis to currently non-dominated and isolated solutions so that population members progress towards the Pareto-optimal front iteratively. To make the overall procedure faster and to perform the task with a more guaranteed manner, EMO methodologies must be combined with mathematical optimisation techniques having local convergence properties. A simple-minded approach would be to start the optimisation task with an EMO and the solutions obtained from EMO can be improved by optimising a composite objective derived from multiple objectives to ensure a good spread by using a local search technique. Another approach would be to use a local search technique as a mutation-like operator in an EMO so that all population members are at least guaranteed local optimal solutions. A study [48] has demonstrated that the latter approach is an overall better approach from a computational point of view.

However, the use of a local search technique within an EMO has another advantage. As, a local search can find a weak or a near Pareto-optimal point, the presence of such super-individual in a population can cause other near Pareto-optimal solutions to be found as a an outcome of recombination of the super-individual with other population members. A recent study has demonstrated this aspect [49].

## 1.10 Practical EMOs

Here, we describe some recent advances of EMO in which different practicalities are considered.

### 1.10.1 EMO for Many Objectives

With the success of EMO in two and three objective problems, it has become an obvious quest to investigate if an EMO procedure can also be used to solve four or more objective problems. An earlier study [50] with eight objectives revealed somewhat negative results. EMO methodologies work by emphasizing non-dominated solutions in a population. Unfortunately, as the number of objectives increase, most population members in a randomly created population tend to become non-dominated to each other. For example, in a three-objective scenario, about 10% members in a population of size 200 are non-dominated, whereas in a 10-objective problem scenario, as high as 90% members in a population of size 200 are non-dominated. Thus, in a large-objective problem, an EMO algorithm runs out of space to introduce new population members into a generation, thereby causing a stagnation in the performance of an EMO algorithm. Moreover, an exponentially large population size is needed to represent a large-dimensional Pareto-optimal front. This makes an EMO procedure slow and computationally less attractive. However, practically speaking, even if an algorithm can find tens of thousands of Pareto-optimal solutions for a multi-objective optimisation problem, besides simply getting an idea of the nature and shape of the front, they are simply too many to be useful for any decision making purposes. Keeping these views in mind, EMO researchers have taken two different approaches in dealing with large-objective problems.

#### 1.10.1.1 Finding a Partial Set

Instead of finding the complete Pareto-optimal front in a problem having a large number of objectives, EMO procedures can be used to find only a part of the Pareto-optimal front. This can be achieved by indicating preference information by various means. Ideas, such as reference point based EMO [36, 51], ‘light beam search’ [38], biased sharing approaches [52], cone dominance [53], etc. are suggested for this purpose. Each of these studies have shown that up to 10 and 20-objective problems, although finding the complete frontier is a difficulty, finding a partial frontier corresponding to certain preference information is not that difficult a proposition. Despite the dimension of the partial frontier being identical to that of the complete Pareto-optimal frontier, the closeness of target points in representing the desired partial frontier helps make only a small fraction of an EMO population

to be non-dominated, thereby making rooms for new and hopefully better solutions to be found and stored.

The computational efficiency and accuracy observed in some EMO implementations have led a distributed EMO study [53] in which each processor in a distributed computing environment receives a unique cone for defining domination. The cones are designed carefully so that at the end of such a distributed computing EMO procedure, solutions are found to exist in various parts of the complete Pareto-optimal front. A collection of these solutions together is then able to provide a good representation of the entire original Pareto-optimal front.

### 1.10.1.2 Identifying and Eliminating Redundant Objectives

Many practical optimisation problems can easily list a large number of objectives (often more than 10), as many different criteria or goals are often of interest to practitioners. In most instances, it is not entirely definite whether the chosen objectives are all in conflict with each other or not. For example, minimization of weight and minimization of cost of a component or a system are often mistaken to have an identical optimal solution, but may lead to a range of trade-off optimal solutions. Practitioners do not take any chance and tend to include all (or as many as possible) objectives into the optimisation problem formulation. There is another fact which is more worrisome. Two apparently conflicting objectives may show a good trade-off when evaluated with respect to some randomly created solutions. But if these two objectives are evaluated for solutions close to their optima, they tend to show a good correlation. That is, although objectives can exhibit conflicting behavior for random solutions, near their Pareto-optimal front, the conflict vanishes and optimum of one can approach close to the optimum of the other.

Thinking of the existence of such problems in practice, recent studies [54, 55] have performed linear and non-linear principal component analysis (PCA) to a set of EMO-produced solutions. Objectives causing positively correlated relationship between each other on the obtained NSGA-II solutions are identified and are declared as redundant. The EMO procedure is then restarted with non-redundant objectives. This combined EMO-PCA procedure is continued until no further reduction in the number of objectives is possible. The procedure has handled practical problems involving five and more objectives and has shown to reduce the choice of real conflicting objectives to a few. On test problems, the proposed approach has shown to reduce an initial 50-objective problem to the correct three-objective Pareto-optimal front by eliminating 47 redundant objectives. Another study [56] used an exact and a heuristic-based conflict identification approach on a given set of Pareto-optimal solutions. For a given error measure, an effort is made to identify a minimal subset of objectives which do not alter the original dominance structure on a set of Pareto-optimal solutions. This idea has recently been introduced within an EMO [57], but a continual reduction of objectives through a successive application of the above procedure would be interesting.

This is a promising area of EMO research and definitely more and more of computationally faster objective-reduction techniques are needed for the purpose. In this direction, the use of alternative definitions of domination is important. One such idea redefined the definition of domination: a solution is said to dominate another solution, if the former solution is better than latter in more objectives. This certainly excludes finding the entire Pareto-optimal front and helps an EMO to converge near the intermediate and central part of the Pareto-optimal front. Another EMO study used a fuzzy dominance [58] relation (instead of Pareto-dominance), in which superiority of one solution over another in any objective is defined in a fuzzy manner. Many other such definitions are possible and can be implemented based on the problem context.

### 1.10.2 Dynamic EMO

Dynamic optimisation involves objectives, constraints, or problem parameters which change over time. This means that as an algorithm is approaching the optimum of the current problem, the problem definition has changed and now the algorithm must solve a new problem. Often, in such dynamic optimisation problems, an algorithm is usually not expected to find the optimum, instead it is best expected to track the changing optimum with iteration. The performance of a dynamic optimiser then depends on how close it is able to track the true optimum (which is changing with iteration or time). Thus, practically speaking, optimisation algorithms may hope to handle problems which do not change significantly with time. From the algorithm's point of view, as in these problems the problem is not expected to change too much from one time instance to another and some good solutions to the current problem are already at hand in a population, researchers ventured to solving such dynamic optimisation problems using evolutionary algorithms [59].

A recent study [60] proposed the following procedure for dynamic optimisation involving single or multiple objectives. Let  $\mathcal{P}(t)$  be a problem which changes with time  $t$  (from  $t = 0$  to  $t = T$ ). Despite the continual change in the problem, we assume that the problem is fixed for a time period  $\tau$ , which is not known a priori and the aim of the (offline) dynamic optimisation study is to identify a suitable value of  $\tau$  for an accurate as well computationally faster approach. For this purpose, an optimisation algorithm with  $\tau$  as a fixed time period is run from  $t = 0$  to  $t = T$  with the problem assumed fixed for every  $\tau$  time period. A measure  $\Gamma(\tau)$  determines the performance of the algorithm and is compared with a pre-specified and expected value  $\Gamma_L$ . If  $\Gamma(\tau) \geq \Gamma_L$ , for the entire time domain of the execution of the procedure, we declare  $\tau$  to be a permissible length of stasis. Then, we try with a reduced value of  $\tau$  and check if a smaller length of stasis is also acceptable. If not, we increase  $\tau$  to allow the optimisation problem to remain stasis for a longer time so that the chosen algorithm can now have more iterations (time) to perform better. Such a procedure will eventually come up with a time period  $\tau^*$  which would be the smallest time of stasis allowed for the optimisation algorithm to work based on

chosen performance requirement. Based on this study, a number of test problems and a hydro-thermal power dispatch problem have been recently tackled [60].

In the case of dynamic multi-objective problem solving tasks, there is an additional difficulty which is worth mentioning here. Not only does an EMO algorithm needs to find or track the changing Pareto-optimal fronts, in a real-world implementation, it must also accommodate an immediate decision about which solution to implement from the current front before the problem changes to a new one. Decision-making analysis is considered to be time-consuming involving execution of analysis tools, higher-level considerations, and sometimes group discussions. If dynamic EMO is to be applied in practice, *automated* procedures for making decisions must be developed. Although it is not clear how to generalize such an automated decision-making procedure in different problems, problem-specific tools are certainly possible and certainly a worthwhile and fertile area for research.

### 1.10.3 Uncertainty Handling Using EMO

A major surge in EMO research has taken place in handling uncertainties among decision variables and problem parameters in multi-objective optimisation. Practice is full of uncertainties and almost no parameter, dimension, or property can be guaranteed to be fixed at a value it is aimed at. In such scenarios, evaluation of a solution is not precise, and the resulting objective and constraint function values becomes probabilistic quantities. optimisation algorithms are usually designed to handle such stochasticities by using crude methods, such as the Monte Carlo simulation of stochasticities in uncertain variables and parameters and by sophisticated stochastic programming methods involving nested optimisation techniques [61]. When these effects are taken care of during the optimisation process, the resulting solution is usually different from the optimum solution of the problem and is known as a ‘robust’ solution. Such an optimisation procedure will then find a solution which may not be the true global optimum solution, but one which is less sensitive to uncertainties in decision variables and problem parameters. In the context of multi-objective optimisation, a consideration of uncertainties for multiple objective functions will result in a robust frontier which may be different from the globally Pareto-optimal front. Each and every point on the robust frontier is then guaranteed to be less sensitive to uncertainties in decision variables and problem parameters. Some such studies in EMO are [62, 63].

When the evaluation of constraints under uncertainties in decision variables and problem parameters are considered, deterministic constraints become stochastic (they are also known as ‘chance constraints’) and involves a *reliability index* ( $R$ ) to handle the constraints. A constraint  $g(\mathbf{x}) \geq 0$  then becomes  $\text{Prob}(g(\mathbf{x}) \geq 0) \geq R$ . In order to find left side of the above chance constraint, a separate optimisation methodology [64], is needed, thereby making the overall algorithm a bi-level optimisation procedure. Approximate single-loop algorithms exist [65] and recently one such methodology has been integrated with an EMO [61] and shown

to find a ‘reliable’ frontier corresponding a specified reliability index, instead of the Pareto-optimal frontier, in problems having uncertainty in decision variables and problem parameters. More such methodologies are needed, as uncertainties is an integral part of practical problem-solving and multi-objective optimisation researchers must look for better and faster algorithms to handle them.

#### ***1.10.4 Meta-Model Assisted EMO***

The practice of optimisation algorithms is often limited by the computational overheads associated with evaluating solutions. Certain problems involving expensive computations, such as numerical solution of partial differential equations describing the physics of the problem, finite difference computations involving an analysis of a solution, computational fluid dynamics simulation to study the performance of a solution over a changing environment, etc. In some such problems, evaluation of each solution to compute constraints and objective functions may take a few hours to a day or two. In such scenarios, even if an optimisation algorithm needs 100 solutions to get anywhere close to a good and feasible solution, the application needs an easy three to six months of continuous computational time. In most practical purposes, this is considered a ‘luxury’ in an industrial set-up. optimisation researchers are constantly on their toes in coming up with approximate yet faster algorithms.

Meta-models for objective functions and constraints have been developed for this purpose. Two different approaches are mostly followed. In one approach, a sample of solutions are used to generate a meta-model (approximate model of the original objectives and constraints) and then efforts have been made to find the optimum of the meta-model, assuming that the optimal solutions of both the meta-model and the original problem are similar to each other [66, 67]. In the other approach, a successive meta-modelling approach is used in which the algorithm starts to solve the first meta-model obtained from a sample of the entire search space [68–70]. As the solutions start to focus near the optimum region of the meta-model, a new and more accurate meta-model is generated in the region dictated by the solutions of the previous optimisation. A coarse-to-fine-grained meta-modelling technique based on artificial neural networks is shown to reduce the computational effort by about 30–80% on different problems [68]. Other successful meta-modelling implementations for multi-objective optimisation based on Kriging and response surface methodologies exist [70, 71].

### **1.11 Conclusions**

This chapter has introduced the fast-growing field of multi-objective optimisation based on evolutionary algorithms. First, the principles of single-objective EO techniques have been discussed so that readers can visualize the differences

between EO and classical optimisation methods. The EMO principle of handling multi-objective optimisation problems is to find a representative set of Pareto-optimal solutions. Since an EO uses a population of solutions in each iteration, EO procedures are potentially viable techniques to capture a number of trade-off near-optimal solutions in a single simulation run. This chapter has described a number of popular EMO methodologies, presented some simulation studies on test problems, and discussed how EMO principles can be useful in solving real-world multi-objective optimisation problems through a case study of spacecraft trajectory optimisation.

Finally, this chapter has discussed the potential of EMO and its current research activities. The principle of EMO has been utilised to solve other optimisation problems that are otherwise not multi-objective in nature. The diverse set of EMO solutions have been analyzed to find hidden common properties that can act as valuable knowledge to a user. EMO procedures have been extended to enable them to handle various practicalities. Finally, the EMO task is now being suitably combined with decision-making activities in order to make the overall approach more useful in practice.

EMO addresses an important and inevitable fact of problem-solving tasks. EMO has enjoyed a steady rise of popularity in a short time. EMO methodologies are being extended to address practicalities. In the area of evolutionary computing and optimisation, EMO research and application currently stands as one of the fastest growing fields. EMO methodologies are still to be applied to many areas of science and engineering. With such applications, the true value and importance of EMO will become evident.

**Acknowledgments** The author acknowledges the support and his association with University of Skövde, Sweden and Aalto University School of Economics, Helsinki. This chapter contains some excerpts from previous publications by the same author entitled ‘Introduction to Evolutionary Multi-Objective optimisation’, in J. Branke, K. Deb, K. Miettinen and R. Slowinski (Eds.) *Multiobjective Optimization: Interactive and Evolutionary Approaches* (LNCS 5252) (pp. 59–96), 2008, Berlin: Springer and ‘Recent Developments in Evolutionary Multi-Objective Optimization’ in M. Ehrgott et al. (Eds.) *Trends in Multiple Criteria Decision Analysis* (pp. 339–368), 2010, Berlin: Springer.

## References

1. Deb, K. (2001). *Multi-objective optimisation using evolutionary algorithms*. Chichester, UK: Wiley.
2. Goldberg, D. E. (1989). *Genetic algorithms for search, optimisation, and machine learning*. Reading, MA: Addison-Wesley.
3. Deb, K., Reddy, A. R., & Singh, G. (2003). Optimal scheduling of casting sequence using genetic algorithms. *Journal of Materials and Manufacturing Processes* 18(3):409–432.
4. Deb, K. (1999). An introduction to genetic algorithms. *Sādhanā*. 24(4):293–315
5. Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems* 9(2):115–148

6. Deb, K., Anand, A., Joshi, D. (2002). A computationally efficient evolutionary algorithm for real-parameter optimisation. *Evolutionary Computation Journal* 10(4):371–395
7. Storn, R., Price, K. (1997). Differential evolution—A fast and efficient heuristic for global optimisation over continuous spaces. *Journal of Global Optimization* 11:341–359
8. Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Network* 5(1):96–101
9. Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer.
10. Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design*. New York: Wiley.
11. Bäck, T., Fogel, D., & Michalewicz, Z. (Eds.). (1997). *Handbook of evolutionary computation*. Bristol/New York: Institute of Physics Publishing/Oxford University Press.
12. Deb, K., Tiwari, R., Dixit, M., & Dutta, J. (2007). Finding trade-off solutions close to KKT points using evolutionary multi-objective optimisation. In *Proceedings of the congress on evolutionary computation (CEC-2007)* (pp. 2109–2116)
13. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: MIT Press.
14. Vose, M. D., Wright, A. H., & Rowe, J. E. (2003). Implicit parallelism. In *Proceedings of GECCO 2003 (lecture notes in computer science)* (Vol. 2723–2724). Heidelberg: Springer.
15. Jansen, T., & Wegener, I. (2001). On the utility of populations. In *Proceedings of the genetic and evolutionary computation conference (GECCO 2001)* (pp. 375–382). San Mateo, CA: Morgan Kaufmann.
16. Radcliffe, N. J. (1991). Forma analysis and random respectful recombination. In *Proceedings of the fourth international conference on genetic algorithms* (pp. 222–229).
17. Miettinen, K. (1999). *Nonlinear multiobjective optimisation*. Boston: Kluwer.
18. Kung, H. T., Luccio, F., & Preparata, F. P. (1975). On finding the maxima of a set of vectors. *Journal of the Association for Computing Machinery* 22(4):469–476.
19. Ehrgott, M. (2000). *Multicriteria optimisation*. Berlin: Springer.
20. Deb, K., & Tiwari, S. (2008). Omni-optimiser: A generic evolutionary algorithm for global optimisation. *European Journal of Operations Research* 185(3):1062–1087
21. Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197
22. Coello, C. A. C., Van Veldhuizen, D. A., & Lamont, G. (2002). *Evolutionary algorithms for solving multi-objective problems*. Boston, MA: Kluwer.
23. Osyczka, A. (2002). *Evolutionary algorithms for single and multicriteria design optimisation*. Heidelberg: Physica-Verlag.
24. Zitzler, E., Deb, K., Thiele, L., Coello, C. A. C., & Corne, D. W. (2001). *Proceedings of the first evolutionary multi-criterion optimisation (EMO-01) conference (lecture notes in computer science 1993)*. Heidelberg: Springer.
25. Fonseca, C., Fleming, P., Zitzler, E., Deb, K., & Thiele, L. (2003). *Proceedings of the Second Evolutionary Multi-Criterion Optimization (EMO-03) conference (lecture notes in computer science)* (Vol. 2632). Heidelberg: Springer.
26. Coello, C. A. C., Aguirre, A. H., & Zitzler, E. (Eds.). (2005). *Evolutionary multi-criterion optimisation: Third international conference LNCS* (Vol. 3410). Berlin, Germany: Springer.
27. Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., & Murata, T. (Eds.). (2007). *Evolutionary multi-criterion optimisation, 4th international conference, EMO 2007, Matsushima, Japan, March 5–8, 2007, Proceedings. Lecture notes in computer science* (Vol. 4403). Heidelberg: Springer.
28. Coverstone-Carroll, V., Hartmann, J. W., & Mason, W. J. (2000). Optimal multi-objective low-thrust spacecraft trajectories. *Computer Methods in Applied Mechanics and Engineering* 186(2–4):387–402
29. Srinivas, N., & Deb, K. (1994). Multi-objective function optimisation using non-dominated sorting genetic algorithms. *Evolutionary Computation Journal* 2(3):221–248.
30. Sauer, C. G. (1973). Optimization of multiple target electric propulsion trajectories. In *AIAA 11th aerospace science meeting* (pp. 73–205).



31. Knowles, J. D., & Corne, D. W. (2002). On metrics for comparing nondominated sets. In *Congress on evolutionary computation (CEC-2002)* (pp. 711–716). Piscataway, NJ: IEEE Press.
32. Hansen, M. P., & Jaskiewicz, A. (1998). *Evaluating the quality of approximations to the non-dominated set IMM-REP-1998-7*. Lyngby: Institute of Mathematical Modelling Technical University of Denmark.
33. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Fonseca, V. G. (2003). Performance assessment of multiobjective optimisers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7(2):117–132
34. Fonseca, C. M., & Fleming, P. J. (1996). On the performance assessment and comparison of stochastic multiobjective optimisers. In H. M. Voigt, W. Ebeling, I. Rechenberg, & H. P. Schwefel (Eds.), *Parallel problem solving from nature (PPSN IV)* (pp. 584–593). Berlin: Springer. Also available as Lecture notes in computer science (Vol. 1141).
35. Fonseca, C. M., da Fonseca, V. G., & Paquete, L. (2005). Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In *Third international conference on evolutionary multi-criterion optimisation, EMO-2005* (pp. 250–264). Berlin: Springer.
36. Deb, K., Sundar, J., Uday, N., & Chaudhuri, S. (2006). Reference point based multi-objective optimisation using evolutionary algorithms. *International Journal of Computational Intelligence Research* 2(6):273–286
37. Deb, K., & Kumar, A. (2007). Interactive evolutionary multi-objective optimisation and decision-making using reference direction method. In *Proceedings of the genetic and evolutionary computation conference (GECCO-2007)* (pp. 781–788). New York: The Association of Computing Machinery (ACM).
38. Deb, K., & Kumar, A. (2007). Light beam search based multi-objective optimisation using evolutionary algorithms. In *Proceedings of the congress on evolutionary computation (CEC-07)* (pp. 2125–2132).
39. Deb, K., Sinha, A., & Kukkonen, S. (2006). Multi-objective test problems, linkages and evolutionary methodologies. In *Proceedings of the genetic and evolutionary computation conference (GECCO-2006)* (pp. 1141–1148). New York: The Association of Computing Machinery (ACM).
40. Coello, C. A. C. (2000). Treating objectives as constraints for single objective optimisation. *Engineering Optimization* 32(3):275–308
41. Deb, K., & Datta, R. (2010). A fast and accurate solution of constrained optimisation problems using a hybrid bi-objective and penalty function approach. In *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI-2010)*.
42. Bleuler, S., Brack, M., & Zitzler, E. (2001). Multiobjective genetic programming: Reducing bloat using SPEA2. In *Proceedings of the 2001 congress on evolutionary computation* (pp. 536–543).
43. Handl, J., & Knowles, J. D. (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation* 11(1):56–76
44. Knowles, J. D., Corne, D. W., & Deb, K. (2008). *Multiobjective problem solving from nature. Springer natural computing series*. Berlin: Springer.
45. Deb, K., & Srinivasan, A. (2006). Innovization: Innovating design principles through optimisation. In *Proceedings of the genetic and evolutionary computation conference (GECCO-2006)* (pp. 1629–1636). New York: ACM.
46. Deb, K., & Sindhya, K. (2008). Deciphering innovative principles for optimal electric brushless D.C. permanent magnet motor design. In *Proceedings of the world congress on computational intelligence (WCCI-2008)* (pp. 2283–2290). Piscataway, NY: IEEE Press.
47. Bandaru, S., & Deb, K. (in press). Towards automating the discovery of certain innovative design principles through a clustering based optimisation technique. *Engineering Optimization*. doi:[10.1080/0305215X.2010.528410](https://doi.org/10.1080/0305215X.2010.528410)

48. Deb, K., & Goel, T. (2001). A hybrid multi-objective evolutionary approach to engineering shape design. In *Proceedings of the first international conference on evolutionary multi-criterion optimisation (EMO-01)* (pp. 385–399).
49. Sindhya, K., Deb, K., & Miettinen, K. (2008). A local search based evolutionary multi-objective optimisation technique for fast and accurate convergence. In *Proceedings of the parallel problem solving from nature (PPSN-2008)*. Berlin, Germany: Springer.
50. Khare, V., Yao, X., & Deb, K. (2003). Performance scaling of multi-objective evolutionary algorithms. In *Proceedings of the second evolutionary multi-criterion optimisation (EMO-03) conference (LNCS)* (Vol. 2632, pp. 376–390).
51. Luque, M., Miettinen, K., Eskelinen, P., & Ruiz, F. (2009). Incorporating preference information in interactive reference point based methods for multiobjective optimisation. *Omega* 37(2):450–462
52. Branke, J., & Deb, K. (2004). Integrating user preferences into evolutionary multi-objective optimisation. In Y. Jin (Ed.), *Knowledge incorporation in evolutionary computation* (pp. 461–477). Heidelberg, Germany: Springer.
53. Deb, K., Zope, P., & Jain, A. (2003). Distributed computing of Pareto-optimal solutions using multi-objective evolutionary algorithms. In *Proceedings of the second evolutionary multi-criterion optimisation (EMO-03) conference (LNCS)* (Vol. 2632, pp. 535–549).
54. Deb, K., & Saxena, D. (2006). Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimisation problems. In *Proceedings of the world congress on computational intelligence (WCCI-2006)* (pp. 3352–3360).
55. Saxena, D. K., & Deb, K. (2007) Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimisation problems: Employing correntropy and a novel maximum variance unfolding. In *Proceedings of the fourth international conference on evolutionary multi-criterion optimisation (EMO-2007)* (pp. 772–787).
56. Brockhoff, D., & Zitzler, E. (2007) Dimensionality reduction in multiobjective optimisation: The minimum objective subset problem. In K. H. Waldmann, & U. M. Stocker (Eds.), *Operations research proceedings 2006* (pp. 423–429). Heidelberg: Springer.
57. Brockhoff, D., & Zitzler, E. (2007). *Offline and online objective reduction in evolutionary multiobjective optimisation based on objective conflicts* (p. 269). ETH Zürich: Institut für Technische Informatik und Kommunikationsnetze.
58. Farina, M., & Amato, P. (2004). A fuzzy definition of optimality for many criteria optimisation problems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 34(3):315–326.
59. Branke, J. (2001). *Evolutionary optimisation in dynamic environments*. Heidelberg, Germany: Springer.
60. Deb, K., Rao, U. B., & Karthik, S. (2007). Dynamic multi-objective optimisation and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling bi-objective optimisation problems. In *Proceedings of the fourth international conference on evolutionary multi-criterion optimisation (EMO-2007)*.
61. Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A., & Padmanabhan, D. (2009). Reliability-based optimisation using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 13(5):1054–1074
62. Deb, K., & Gupta, H. (2006). Introducing robustness in multi-objective optimisation. *Evolutionary Computation Journal* 14(4):463–494
63. Basseur, M., & Zitzler, E. (2006). Handling uncertainty in indicator-based multiobjective optimisation. *International Journal of Computational Intelligence Research* 2(3):255–272
64. Cruse, T. R. (1997). *Reliability-based mechanical design*. New York: Marcel Dekker.
65. Du, X., & Chen, W. (2004). Sequential optimisation and reliability assessment method for efficient probabilistic design. *ASME Transactions on Journal of Mechanical Design* 126(2):225–233.
66. El-Beltagy, M. A., Nair, P. B., & Keane, A. J. (1999). Metamodelling techniques for evolutionary optimisation of computationally expensive problems: Promises and limitations.

- In *Proceedings of the genetic and evolutionary computation conference (GECCO-1999)* (pp. 196–203). San Mateo, CA: Morgan Kaufmann.
67. Giannakoglou, K. C. (2002). Design of optimal aerodynamic shapes using stochastic optimisation methods and computational intelligence. *Progress in Aerospace Science* 38(1):43–76.
  68. Nain, P. K. S., & Deb, K. (2003). Computationally effective search and optimisation procedure using coarse to fine approximations. In *Proceedings of the congress on evolutionary computation (CEC-2003)* (pp. 2081–2088).
  69. Deb, K., & Nain, P. K. S. (2007). In *An Evolutionary multi-objective adaptive meta-modeling procedure using artificial neural networks* (pp. 297–322). Berlin, Germany: Springer.
  70. Emmerich, M. T. M., Giannakoglou, K. C., & Naujoks, B. (2006). Single and multiobjective evolutionary optimisation assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation* 10(4):421–439
  71. Emmerich, M., & Naujoks, B. (2004). Metamodel-assisted multiobjective optimisation strategies and their application in airfoil design. In *Adaptive computing in design and manufacture VI* (pp. 249–260). London, UK: Springer.

## Chapter 2

# Multi-objective Optimisation in Manufacturing Supply Chain Systems Design: A Comprehensive Survey and New Directions

Tehseen Aslam, Philip Hedenstierna, Amos H. C. Ng and Lihui Wang

**Abstract** Research regarding supply chain optimisation has been performed for a long time. However, it is only in the last decade that the research community has started to investigate multi-objective optimisation for supply chains. Supply chains are in general complex networks composed of autonomous entities whereby multiple performance measures in different levels, which in most cases are in conflict with each other, have to be taken into account. In this chapter, we present a comprehensive literature review of existing multi-objective optimisation applications, both analytical-based and simulation-based, in supply chain management publications. Later on in the chapter, we identify the needs of an integration of multi-objective optimisation and system dynamics models, and present a case study on how such kind of integration can be applied for the investigation of bullwhip effects in a supply chain.

---

T. Aslam (✉) · P. Hedenstierna · A. H. C. Ng · L. Wang  
Virtual Systems Research Centre, University of Skövde,  
PO Box 408, 541 28 Skövde, Sweden  
e-mail: tehseen.aslam@his.se

P. Hedenstierna  
e-mail: philip.hedenstierna@his.se

A. H. C. Ng  
e-mail: amos.ng@his.se

L. Wang  
e-mail: lihui.wang@his.se

## 2.1 Introduction

Supply Chain Optimisation (SCO) is an area that has been studied for more than two decades. Traditionally, the main focus of the research studies has been regarding minimising the overall cost or maximising the total revenue as a single-objective optimisation problem. The majority of these single-objective studies have been conducted with the help of various *mathematical programming* approaches. For instance, in the late 1980s, Cohen and Lee [1] presented a four-tier (suppliers, assembly plant, distribution centres and customers) global supply chain model based on *mixed-integer non-linear programming* (MINLP). The intent of the study was to help companies to establish a global manufacturing strategy through the evaluations of various *economic order quantity* (EOQ) techniques by maximising the total profit after tax for the manufacturing facilities and distribution centres. In the 1990s, Arntzen et al. [2] helped an electronic manufacturer to solve their supply chain design problem, by looking at location selection of facilities as well as the production, inventory and shipping quantities. In this study, they developed a supply chain model based on *mixed-integer programming* (MIP), which minimised the total cost, including production costs, distribution costs and inventory expenses, etc. In addition to just looking at the cost, their model also considered time in the objective function. The time variable was measured as the amount of days that were required for production and transportation between each connection in the supply chain. Hence, both cost and time could be weighted as a combination in the single-objective optimisation function. Voudouris [3] presented a *mixed-integer linear programming* (MILP) model which was used to streamline the supply chain operation and increase the efficiency by improving the scheduling process. In contrast to the aforementioned studies, here the objective function was formulated to maximise the supply chain flexibility, i.e. the capability to meet the fluctuating demands. In recent years, researchers such as Jayraman and Pirkul [4] as well as Amiri [5] have continued to model supply chains as single-objective problems. Jayraman and Pirkul [4] studied an integrated logistic model with which they explored the facility location problem regarding production and warehouse facilities. They provided an MIP model of the logistic network in which their objective was to minimise the overall cost of the supply chain. Amiri [5] examined a similar problem by addressing the supply chain distribution network design problem, i.e. locating production plants and distribution warehouses and determining the optimal strategy for distributing the product between production plant, distribution warehouse and customers, using MIP. The goal of the study was to select the optimum number of plants and warehouses that can fulfil customer demand at a minimum total cost for the supply chain. In other literature review studies, such as [6] and [7], one can find many more studies that optimise supply chains with single-objective functions.

Despite the successful implementations of aforementioned studies, in our view, supply chain decisions are much more complex than treating them as single-objective optimisation problems. For instance, while cost, revenue and flexibility

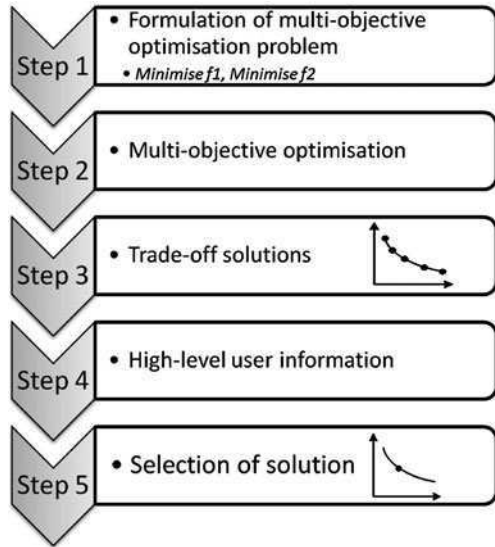
as presented in [3] can be the indicators to determine the performance of a supply chain, there are other important metrics used in supply chain analyses, like lead time, inventory levels, service levels, Work-In-Process (WIP), etc. that should be considered when optimising a supply chain network. A short average lead time means that the total time a product stored in the system is short, which also means that customer orders can be fulfilled within a shorter time and thus leverages the overall performance of the supply chain. A low WIP means that the cost spent on transportation and inventory is lowered and thus is also highly desired. Therefore, to a decision maker, an ideal configuration is the one that maximises delivery service level while simultaneously minimising lead time and WIP. Because of the conflicting nature of the above mentioned metrics, modelling a system using traditional optimisation techniques in which one optimises a single objective or a single weight-based objective to combine multiple objectives would very likely lead to misleading results in a dynamic system such as a supply chain.

In a general *Multi-Objective Optimisation* (MOO) problem, there exists no single best solution with respect to all objectives; a solution might be optimal in one objective but worse in the other objective. In an MOO problem, a decision maker is presented with *Pareto-optimal solutions*, which are a set of trade-offs between the different objectives. These solutions are *non-dominated* solutions, i.e., there exists no other solution which would increase a performance measure without causing a simultaneous decrease in at least one of the other objectives [8]. In this chapter, we present a comprehensive literature review of MOO applications in supply chain management (SCM). Such a review has led us to identify the need for a multi-objective and multi-level optimisation (MLO) framework for SCM which considers not only optimisation of the overall supply chain, but also each entity within the supply chain. The content of this chapter is as follows. In Sect. 2.2, we present a comprehensive literature review of MOO for SCM, which is summarised in Sect. 2.3. In Sect. 2.4, a case study in which MOO is applied for to investigate the behaviour of the bullwhip effect in a supply chain is presented.

## 2.2 Literature Review: MOO for SCM

Multi-objective optimisation (MOO) is a discipline that has been studied since 1970s, and its application areas range widely from resource allocation, transportation, investment decision to mechanical engineering, chemical engineering, automation applications, to name a few. The main concept of MOO is to evaluate two or more conflicting objectives against each other. A simple method to handle an MOO problem is to form a composite objective function as the weighted sum of the conflicting objectives. Because the weight for an objective is proportional to the preference factor assigned to that specific objective, this method is also called preference-based strategy [8]. Apparently, preference-based MOO is simple to apply, because by scalarising an objective vector into a single

**Fig. 2.1** General Pareto-based MOO procedure



composite objective function (e.g., combining all performance measures into a weighted average objective function to represent the overall system cost), an MOO problem can be converted into a single-objective optimisation problem and thus a single trade-off optimal solution can be sought effectively. However, the major drawback is that the trade-off solution obtained by using this procedure is very sensitive to the relative preference vector. Therefore, the choice of the preference weights and thus the obtained trade-off solution is highly subjective to the particular decision maker.

At the same time, it is also argued that using preference-based MOO to obtain a single “global” optimal solution for multi-tier systems, like supply chains, is not desirable if the “global” optimum suggests a set of decision variable values that may sacrifice the performance of the sub-system level. For example, the optimal solution found by the simulation optimisation may be optimal when considering the overall supply chain but totally not acceptable to the company that plays the role as the manufacturer. Therefore, for a decision maker, it would be useful if the posterior Pareto front can be generated quickly by using an MOO algorithm, as shown in Fig. 2.1, so that he/she can choose the most suitable configuration among the trade-off solutions generated.

Examining a supply chain, one clearly sees that a supply chain is a complex system consisting of multiple entities (e.g., suppliers, manufacturers, distributors and retailers, as mentioned earlier), which individually have their own performance measures and objectives to optimise for their internal process e.g., maximising the throughput whilst minimising the WIP. However, optimising these individual entities is not adequate when optimising a supply chain as it is a dynamic network consisting of multiple transaction points with complex

transportations, information transactions and financial transactions between entities. Hence, optimising the supply chain as whole is as crucial as optimisation of the individual entities, and the aim of SCM is to align and combine all these objectives, individually as well as in supply chain, so that they work towards a common goal—increasing the efficiency and profitability of the overall supply chain. SCM is thus multi-objective in nature and involves several conflicting objectives, both on the individual entity level and on the supply chain level.

The literature review that has been conducted in this chapter focusses on research in which different authors have utilised MOO techniques for managing supply chains, in contrast to other reviews like [6, 7, 9]. The survey in [6] mainly focusses on the issues regarding supply chain design problem and the majority of the review papers are based on single-objective optimisation. In [7], the authors concentrate their survey on MIP models for supply chains and as in the previous review the majority of papers are based on single-objective optimisation. Additionally, the scope of the current chapter attempts to cover all supply chain areas, including supply chain design, operation, facility location, supplier selection etc., as long as the SCM problem has been solved with the help of MOO. In this sense, the review reported in [9] can be seen as the closest related work to our targeted scope. However, their main focus lies in presenting a research trend in a specific area of SCM, namely, supply chain revenue management. With the above mentioned scope in mind, we have conducted a search ranging over various major international journals in management science and operations research including: *European Journal of Operational Research*, *International Journal of Production Economics*, *International Journal of Production Research*, *International Journal of Management Science*, *International Journal of Information Science*, *Journal of Computers & Industrial Engineering*, *Journal of Transportation Research*, *International Journal of Revenue Management*, *Journal of Computers & Operations Research*, *European Journal of Purchasing & Supply Management*, etc.

From these sources, we selected publications that span over the last two decades in the area of applying MOO for SCM. After examining the articles we have divided the publications into three main areas, namely, *mathematical programming techniques*, which include MIP, MILP, MINLP etc., *simulation techniques*, which include discrete-event simulation (DES), system dynamic (SD), Petri nets, Multi Agent Systems (MAS) incorporating Agent Based Simulation (ABS), etc. and *modelling technique* not depicted in which we have gathered all the papers in which the authors have not specified explicitly what approach they utilise to model the supply chain.

### ***2.2.1 Mathematical Programming Techniques***

In this section, we present publications in which authors have used mathematical programming techniques to model the supply chain. For instance, Yimer and Demirli [10] present a two-phase MILP model over a multi-product, multi-plant



build-to-order (BTO) supply chain. The purpose of the paper is to address the dynamic scheduling of materials through the supply chain, ranging from replenishment, component manufacturing, customised assembly to distribution of the products. In the proposed approach, the authors break down the supply chain into two subsystems which are then evaluated and analysed sequentially. In the first phase, they looked at the assembly and distribution schedule of the customisable product, whereas in the second phase they looked at manufacturing and procurement planning of component and raw materials. The two subsystems were formulated as MILP models with the objective to minimise the associated aggregate costs whilst maximising the customer satisfactions. The authors used a genetic algorithm (GA) based solution procedure to solve the sub problems.

Another MILP model for MOO using GA in supply chains is presented in [11] whereby a company that produces plastic products needed to design a supply chain, i.e. to choose suppliers and to define the subsets of manufacturing plants and distribution centres and create the distribution network strategy that would satisfy the capacity and demand requirements for the product. The objective of the study was to minimise the total cost of the supply chain, maximise the customer service level in terms of acceptable delivery times and maximise the capacity utilisation balance for the distribution centres. Some observation that can be made from the investigation was that the cost of the supply chain decreased when the service quality and equity on utilisation ratios for the supply chain was reduced. The authors also noticed that when all Pareto-optimal solutions were examined one specific plant was operational in each solution and that four distribution centres were operational in 90% of the solutions. The authors also compared the performance of the GA with an approach where they implemented *simulated annealing*; the comparison showed that the GA outperformed the simulated annealing approach in respect to finding the most Pareto-optimal solutions and with better quality of these Pareto-optimal solutions.

In [12], the authors investigate a future hydrogen cell supply chain. They argue that in order for hydrogen to succeed as sustainable fuel source for cars in the future an entirely new infrastructure needs to be created, from production, through storage, distribution and disposal. To assist the strategic decision making process of designing a new supply chain network, they present a generic MILP optimisation model in order to identify optimal investment strategies and integrated supply chain configurations. When optimising they look at both investment and environmental criteria, in which they try to establish the optimal trade-off between net present value (NPV) and the greenhouse gas emissions throughout the lifecycle of the hydrogen cell. The authors also conducted a case study in which they showed that the MILP model could identify an optimised supply chain design as well as capacity expansion policies and investment strategies.

Authors in [13] examine the simultaneous MOO of a multi-echelon supply chain with uncertain customer demand and product prices by implementing fuzzy MOO method. They develop a supply chain model as a MILP problem that investigate how to maximise each participant's expected profit, the average safety inventory levels for each entity, the average customer service level for the retailer,

the robustness of selected objectives to demand uncertainty and maximise the acceptability levels of buyers and sellers regarding product price. The demand uncertainty is handled as discrete scenarios with specified probabilities whereas the product price uncertainty is handled as fuzzy variables. In their results, Chen and Lee [13] point out that considering robustness measures as part of multiple objectives significantly reduces the variability of other objective values to product demand uncertainties. They also show that the proposed fuzzy decision method provided a compensatory solution for the multiple conflicting objectives.

Guillén et al. [14] also present a MILP model of a supply chain to solve a stochastic multi-objective problem by using the standard  $\varepsilon$ -constraint method and branch-and-bound techniques. The problem statement that they investigate is regarding configuration of a supply chain that maximises the NPV and the demand satisfaction while minimising the financial risk that they define as a probability of not meeting a certain profit target level. The result from this study provided the decision makers with a set of Pareto-optimal solution from which they could choose their supply chain configuration based on their preferences.

In [15], Sabri and Beamon present a supply chain model for simultaneous strategic and operations planning of the supply chain. The model consists of a four echelon (suppliers, manufacturing plants, distribution centres and customer zones) supply chain and is divided in two sub-models, namely, the strategic sub-model and the operational sub-model. The strategic sub-model's objective is to optimise the supply chain configuration and material flow. More specifically, the authors use the  $\varepsilon$ -constraint method to: (1) seek the optimal number and locations for the plants and distribution centres; (2) determine the best distribution centres for the customer zones; (3) optimise the material flow throughout the supply chain. The objective function for this sub-model is to minimise cost whilst ensuring sufficient volume flexibility. The operational sub-model is integrated with the strategy in order to incorporate the uncertainty of production, transportation and distribution. Hence, when the output variables of the strategic sub-model have been determined, customer demand, required service and flexibility levels, cost, lead times etc. are estimated under uncertainty. The multi-objective function in the operational sub-model incorporates all trade-offs between costs, service levels and flexibility levels. Thus, the model that is presented in this paper is based on an iterative structure, first one optimises the strategic sub-model for an existing or a proposed supply chain configuration, after that the output variables from the strategic sub-model are sent to the operational sub-model as input data and the operational sub-model is optimised based on the determined supply chain configuration. Output variables from the operational optimisation runs are sent back to the strategic sub-model where a new optimisation is performed with the new variables which also incorporate uncertainty.

In all of the above presented publications supply chains have been modelled utilising MILP; however there are several publications that model supply chains with the help of MINLP. In contrast to [13], the authors in [16] develop a fuzzy multi-objective MINLP model for a single product, multi-stage, multi-objective supply chain design problem where they propose an approach based on *spanning*

*tree-based GA* (st-GA) for a Chinese liquor company. The company intends to start producing fruit beverages, and they wish to design a supply chain network for the new product to determine the amount and location of plants and distribution centres (DC), as well as to determine the optimal distribution strategy that will satisfy the demand in a cost-effective manner under a fuzzy market demand. To do so, the MINLP model considers two objectives namely minimisation of total cost, which includes fixed plant and DC cost as well as inbound and outbound distribution costs, and maximisation of customer service level which basically is the acceptable delivery time. At the end the authors compare st-GA to a *matrix-based GA*, in order to see the efficiency and effectiveness of the st-GA in a random fuzzy environment.

Guillén-Gosalbez and Grossman [17] investigate a supply chain design and planning problem for a sustainable chemical supply chain. The aim of the paper is to identify a supply chain configuration e.g., number, location and capacities of plants and DC, and transportation links between the entities, together with its optimal planning decisions, e.g., production rate at plants, material flow between plants, DC and market etc. The authors formulate the overall problem as a bi-criterion stochastic non-convex MINLP that attempts to optimise two objectives namely, maximise the NPV and minimise the environmental impact. The authors solve the bi-criterion stochastic non-convex MINLP problem by applying  $\varepsilon$ -constraint method and spatial branch-and-bound technique. They also present two examples of a case study in which they show that there clearly exist a conflict between economical and environmental factors in SCM, however they point out that the approach presented in their paper allows them to identify process alternatives that can reduce the environmental impact by adjusting the strategic supply chain decisions.

In [18] the authors study a supply chain network design problem for a forward/reverse logistic network. They present a bi-objective MINLP model representing a multi-stage logistic network which includes production, distribution, customer zones, collection/inspection centres, as well as disposal centres. In their paper the authors argue that in the vast majority of cases logistic networks are designed for forward logistic activities, i.e. a traditional supply chain, without taking into account the reverse flow of the products i.e. from customer to disposal centres. The main focus of their study is to determine the location, amount and capacity of production, distribution, collection and disposal centres together with determining the product flow between the mentioned facilities. For the MOO and finding the non-dominated solutions they develop a multi-objective memetic algorithm (MOMA) which, similar to GA, is a population based heuristic algorithm. The authors point out that pure GAs often lack the capability of sufficient search intensification whereas memetic algorithms provide additional local searches and combines the advantages of efficient heuristics incorporating domain knowledge and population based search approaches. The objective functions that they consider is minimising the total cost which includes fixed costs for opening the different centres, transportation cost and cost savings from integrating different centres at one location. The second objective is maximising the responsiveness of

both the forward and reverse networks. At the end they compare their MOMA with the multi-objective genetic algorithm (MOGA) presented in [12] and with LINGO 8.0 which is an optimisation software. According to their results their MOMA performed better than the MOGA in terms of average ratio of obtained Pareto-optimal solutions, and the comparison between LINGO and MOMA showed that the quality of the solutions that they got from their MOMA were reasonable.

In both [19] and [20] the authors apply fuzzy methods for the MOO and both studies formulate the supply chain as MINLP model; however they investigate two different topics. In [19] they try to establish a fair profit distribution for a traditional supply chain. The authors develop a multi-stage, multi-product, multi-period production and distribution planning model which they later on formulate to an MINLP problem. The optimisation objectives in the study are to maximise the customer service level, maximise the safe inventory level and maximise the profit for each supply chain entity. Fuzzy sets were used in order to get the trade-off solutions among the participating entities in the supply chain. In contrast to this, in [20] the authors attempt to study the effects of uncertain parameters for a mid-term supply chain planning problem where there are no available probability distributions for these parameters. For their study the authors build five different models; two LP, one MILP and two MINLP models. The authors found that the MINLP mid-term planning model developed by [21] performs the best, and they applied a fuzzy programming approach for the MOO with the aim to minimise the costs whilst maximising the demand satisfaction.

In [22] as in [17] the authors also use the  $\varepsilon$ -constraint method, but here they develop a multi-objective stochastic model which uses Six Sigma measures to evaluate the financial risk. They also propose a two-stage approach where first the strategic variables are investigated i.e. which manufacturing plants and distribution centres should be opened, and then investigate the operational variables i.e. material flow between the entities, capacity, production etc. The authors seek to maximise the total profit for the supply chain and to increase the Sigma quality level by minimising the total number of defects obtained from the suppliers.

A multi-objective stochastic MINLP approach is used in [23] to study a supply chain design problem under uncertainty. Demands, supplies, shortage, processing, transportation and costs for capacity expansions are all considered uncertain parameters. Here as in [22] the authors also propose a two-stage approach where the first stage deals with the network configuration and the second with decision variables related to number of product to manufacture and store, material flow etc. For the MOO and general Pareto-optimal solutions they use a variation of the goal programming approach called the *goal attainment* technique. The three objective functions in this study are minimising the total investment cost for the first-stage and the second-stage processing, transportation, shortage and capacity expansion costs, minimising the variance of the total cost and minimising the financial risk.

The authors in [24] present a mathematical modelling approach for three sub-chains for a supply chain in which the authors have decomposed the supply chain participants into four models. The three sub-chains, which are the supplier, manufacturer and the retailer, incorporate decision making regarding *stochastic*

*customer demands, procurement scheduling, production scheduling and resource allocation.* Each model of the above mentioned supply chain entity tries to maximise its utility which is achieved by maximising the net profit. The fourth model, which the authors present, is the *broker*; this model manages all the issues regarding resource allocation amongst the supply chain participants. The broker also deals with maximising the utility for the entire supply chain network. Hence, the authors present an approach where each entity model represents its optimal solutions with a set of tuples and then given these solutions the broker maximises the supply network and presents possible configurations of the supply chain as a set of tuples. To perform this optimisation the authors propose a distributed multi-objective GA (DMOGA), which according to the authors is known to be an efficient algorithm for distributed GA. The authors point out that the main difference between traditional MOGA and DMOGA is that being able to use subpopulations in DMOGA allows one to exploit information in a better way. Utilising the DMOGA the authors broke down the population of the entire supply chain decisions into four subpopulations, where each subpopulation implemented traditional GA to generate sets of optimal solutions. After generating a set of optimal solutions the subpopulations swap the strings with the help of the migration operator, this procedure is only done for the optimal solutions which are attained when running the local GA on the individual models. In their paper the authors also present a case study where they investigated how to obtain the best combination of products, customers and parts that maximises the revenue of the supply chain participants as well as the utility for the entire supply chain. From their study the authors could see that the DMOGA could find optimal or near-optimal solutions for the problem in hand and also that the DMOGA improved the computational performance.

Researchers in [25] also have presented a paper investigating a supplier selection problem; here the authors aimed to integrate the *analytic network process* (ANP) with multi-objective MINLP in order to examine tangible and intangible factors for picking the best of the suppliers and defining the optimal order quantities among the selected suppliers. The authors present a two-stage approach, namely the selection stage and the shipment stage. In the selection stage, the authors evaluate the suppliers based on 14 criteria that concern benefits, opportunities, cost and risk, for a company. To determine the suitable suppliers the authors use the ANP which is an extension of the *analytic hierarchy process* (AHP) with the difference that in the ANP there exist a feedback loop between elements in different levels of the hierarchy as well as between elements in the same level. The shipment stage utilises the developed multi-objective MINLP model to attain non-dominated solutions for the objective functions. To solve MOO problem the authors implemented two approaches;  $\varepsilon$ -constraint method and a *reservation level-driven Tschebyscheff procedure* (RLTP), which later on were evaluated against each other. For the optimisation the authors considered three objectives: maximising the total value of purchasing and minimising the budget and the defect rate. Results from a numerical example showed that the RLTP approach was better than the  $\varepsilon$ -constraint method.

Another supplier selection issue is presented in [26]; here the authors propose a mathematical model using Microsoft Excel, for a single buyer, multi-objective, multi-supplier, multi-product procurement problem incorporating product lifecycle (PLC) aspects in a buying firm's sourcing strategy. For the MOO a standard MINMAX technique is utilised in order to obtain the Pareto-optimal solutions with the objective to minimise the overall cost of procurement and maximise the total quality level of a product and the delivery performance.

Du and Evans [27] address a closed loop reverse logistic network problem which deals with product returns requiring service. Addressing the problem the authors developed a multi-objective MIP optimisation model with the objective to minimise total reverse supply chain cost and the total tardiness of cycle time. To run the optimisation the authors used a combination of three different algorithms, namely; *scatter search* (SS), *dual simplex method* and *constraint method*. The SS algorithm is used to deal with discrete/binary variables in the model which e.g., could represent capacity planning among potential facilities in the reverse network. Whereas the dual simplex algorithm is implemented to represent transportation arrangement and to deal with continuous parameters in the MIP model, and the constraint method is utilised to attain the non-dominated solutions for the reverse supply network. The numerical results from their study showed that they were able to attain trade-off relationships between the analysed objectives and configure a reverse logistic supply chain.

Both Jayaraman [28] and Farahani and Asgari [29] present a facility location problem. [28] investigate a service facility location problem in order to find the location of a given number of service facilities in a supply chain network. In the study the authors develop a MIP model implementing the *non-inferior set estimation* (NISE) method for the MOO incorporating three objective functions. The first two objectives are cost-related, where one cost objective relates to the costs occurred when opening a facility e.g., fixed cost required to open a facility and the other cost related objective is the operating cost. The third objective is to fulfil the customer demand as quickly as possible. Hence, the authors seek to minimise the fixed costs incurred when opening a facility, minimise the operating cost incurred when satisfying customer demand and minimise the average response time for serving the customer demand. In contrast to finding the optimal service facility locations, the authors in [29] investigate a DC location problem in a real-world military logistic system with the objective to establishing the least number of DC and locating them at the best possible location, hence minimise the cost for locating the DC and maximise the quality of the location. To solve this problem the authors develop a LP model that implements the *utility function* method to perform the location optimisation. Both these studies showed that one could establish, operate and locate the respective studied facilities in a cost effective manner whilst satisfying customer demands.

In the papers [30–33] the researchers have looked into the supplier/vendor selection problem. [30] proposes a mathematical supplier selection model that utilises *visual interactive goal programming* (VIG) in order to obtain non-dominated solutions. A case study was presented for a hydraulic gear pump

manufacturer examining a multiple replenishment purchasing problem and helping them select suppliers and allocating orders among them. The multiple objectives of this case study were to minimise the purchasing costs, maximise the quality of the products purchased from different suppliers and maximise the delivery reliability of each product. The result showed that the purchasing team were able to find solutions with the help of MOO that increased the quality whilst decreasing the costs. Wadhwa and Ravindran [31] formulate a vendor selection problem exploring quality, lead-time and total cost of purchasing under quantity discount as the three objective functions for the optimisation. The authors develop a mathematical model considering multiple buyers and vendors. The multi-objective problem is solved utilising three approaches; *weighted objective*, *goal programming* and *compromise programming*. A comparison of these techniques together with a single-objective formulation, where the objective is to minimise the price, is performed using the *value path approach*. The result showed that the goal programming technique was the most suitable approach for this vendor selection problem. Erol and Ferrell Jr [32] present an integrated methodology that simultaneously addresses the supplier selection problem and the customer assignment problem. One aspect of the proposed methodology is to select appropriate suppliers from the point of view of each warehouse and the other aspect of the methodology is to assign the warehouses to the customers. To realise their methodology the authors present an example study where they consider a distributor supply chain consisting of ten warehouses, ten suppliers and ten customers. To resolve this problem a multi-objective mathematical programming model is developed with the objective to maximise the supply chain satisfaction, which includes element from both the supplier selection and customer selection, and minimise the total cost. Solving the multi-objective problem the authors use an approach similar to *pre-emptive goal programming* but with a slight modification. The authors assume that for an objective in one of the sub problems experts can specify an appropriate level of the objective decreasing the search space for that objective. The study showed that trade-offs between the examined objectives could be gained implementing the proposed methodology and the results pointed out some suppliers and warehouses that were recurring in different solutions. In [33] Li and Zabinsky develop a two-stage *stochastic programming* (SP) model, a *chance-constrained programming* (CCP) model as well as a MIP model with the aim to identify minimal set of supplier and optimal order quantities considering volume discounts. The first two modelling approaches incorporate uncertainty in the shape of uncertain customer demand and supplier capacity whereas the MIP model is deterministic. The research question that the authors intend to investigate is: how many suppliers are appropriate, which suppliers should we choose and what are the optimal ordering or replenishing policies? To represent the uncertainty the SP model utilises a scenario-based approach called *penalty coefficients* whereas the CCP model undertakes a probability distribution and constraints the probability of not meeting the demand. The authors argue that the SP model is more suitable when decision maker do not have a clear definition about the distribution of the stochastic variables but may instead have access to historical data to define



scenarios and investigate possible future scenarios. Whereas the CCP model is developed as an alternative to the SP model in order to incorporate the uncertainties and in contrast the CCP model requires that the demand and capacity constraints are fulfilled with some predetermined probability. The optimisation objective of these models is to minimise the number of selected suppliers and to minimise the total cost which includes purchasing costs, transportation, coordination and inventory costs. To find the Pareto-optimal solutions the model utilises the  $\varepsilon$ -constraint method. As opposed to the CCP approach the SP model included two-stages; in the first stage it deals with decision regarding which supplier to select, whereas the order amounts and shipments plans are considered in the second stage. Thus, the minimisation of number of selected suppliers is done in the first stage and the minimisation of the expected total cost is done in the second stage. The third approach which was a MIP model is based on the CCP model and has the same objectives and utilises the same optimisation method for finding the Pareto-optimal solutions as the other two models. The sample problem presented in this paper consists of ten potential suppliers and four plants with the option to order 50 different types of components. The result from this study showed that out of 38 Pareto-optimal solutions the CCP model was able to find 26, the SP model 24 and the MIP model only found 6 solutions. Furthermore, the result showed that the two stochastic models provided more robust solutions as compared with the deterministic MIP model, and the SP model was preferable. The uncertainty was represented by scenarios whereas the CCP model could provide the Pareto-front in more straightforward way and with less computational time when the uncertainties were represented by distributions. At the end the authors found the CCP approach to be the most suitable for the problem in hand because of computational advantage and straightforwardness of exploring the solutions.

Che and Chiang [34] aim to investigate a supply chain planning problem for a build-to-order (BTO) supply chain. They present a mathematical model for the BTO supply chain integrating the supplier selection, product assembly, and the logistic distribution planning. The main purpose of the paper as the authors described is first of all to develop a multi-objective mathematical model for the investigated supply chain as well as implement and evaluate their own *modified Pareto GA* (mPaGA) optimisation technique. mPaGA which is based on the *Pareto GA* (PaGA) has the intention to improve the crossover and mutation operators of the PaGA in order to attain a higher solving efficiency. Moreover, an equilibrium and feasibility-adjustment mechanism are proposed in order to maintain the feasibility of each individual with the aim of reducing the computational time for searching after feasible individuals. A two-stage supply chain example study implementing the proposed model and algorithm as well as a comparison between the mPaGA and PaGA approach is presented at the end. The study was divided into three scenarios, in the first scenario the optimisation objectives were to minimise the cost and the delivery time, second scenario objectives were minimisation of cost and maximisation of quality and the third scenario contained all of the above i.e. minimisation of cost and delivery time along with maximisation of quality. The result generated from the study showed that mPaGA was significantly



superior to PaGA in all three scenarios and that it gives less variation and greater solution stability solving the Pareto-optimal solutions sets.

In [35], the authors present a study covering two research questions: (1) how to find the preferred solutions showing the trade-offs between environmental and business issues and (2) how to improve the understanding of the decision maker for the trade-offs between these performance measures. For answering these questions, the authors present a two-phased heuristic approach in which they implement a multi-objective linear model with the objective to minimise the costs, the cumulative energy demand (CED) and the waste associated with the reverse logistic network. In the first phase the model generates a number of non-dominated frontiers for the multiple objectives by implementing the  $\epsilon$ -constraint method. The second phase deals with question of how to increase the decision maker's understanding. Here the authors focus on visual representation of generated frontier; selecting his/her preferred solutions from the frontiers provided from phase one, the algorithm projects the selected solutions into the efficient frontier of the problem with the three objectives. Another reverse logistic problem is presented in [36], wherein the author conducts a study based on the concept of green-SCM, and addresses the optimisation of the nuclear power generation and the corresponding issue of reverse logistics for the nuclear waste. The author formulates a linear MOO model implementing the *composite* method for the optimisation. The objective of the study is to increase the total net profit of the nuclear supply chain by maximising the power supply chain-based net profit and minimising the reverse logistic chain-based costs. Results gained from the executed numerical study showed that implementing the proposed approach the total performance of the nuclear supply chain could be improved by 7–18%, depending on the weights associated with the investigated objective functions.

A mathematical model for a food processing supply chain is presented in [37], the author propose a hybrid meta-heuristic approach combining a multi-objective *Bee Colony algorithm* with *constructive rough set* heuristics for a supply chain process scheduling problem. Hence, the authors introduce the concept of *Pareto Bee Colony Optimisation* (PBCO) and present a case study concerning scheduling of several processes in a milk production centre. The PBCO approach is also compared and evaluated in terms of performance with two other meta-heuristic methods, namely; *Ant Colony Optimisation* (ACO) and *Tabu Search* (TS). Their results showed that the TS method provided the most likelihood results within the shortest execution time. The proposed PBCO performed slightly better than the standard Bee Colony approach.

### 2.2.2 Simulation Techniques

Attributable to its rich expressiveness to handle complexity and its powerful programming flexibility, simulation is capable of predicting system performance with extremely high accuracy. Unfortunately, using simulation alone is not

sufficient to yield optimal solutions. Simulation by itself is not a real optimisation tool and “an extra step is needed—a step that joins simulation and optimisation” [38]. This technique is called simulation-based optimisation (SBO), whereby simulation models are integrated with meta-heuristic search algorithms (e.g., Genetic Algorithms or Tabu Search). When compared to classical optimisation methods, SBO is, because of its inherent attributes, very suitable for solving real-world industrial-based complex problems. In this section, we present a number of papers that use SBO for the MOO of SCM.

In [39] the authors present a toolbox called ONE (Optimisation methodologies for Networked Enterprises) that supports the decision makers in their effort to assess, design and improve supply chain networks. The ONE architecture consists of four modules: (1) a network module that supports development of supply chain models, (2) an optimisation module that offers different methods e.g., mathematical programming and GAs, (3) a statistical data miner that offers various data mining methods, and (4) a simulation module for evaluating the supply chain models. Besides presenting the toolbox the authors also presented two case studies that both required MOO, one for the automotive industry and the second for the textile industry. In the automotive case study the authors consider an existing multi-facility supply chain where the objective is to increase the profit and the responsiveness of the supply chain by redesigning the distribution network. The company involved wants to investigate which facilities should be closed down and which should continue to operate, how the production order assignments should be distributed among the manufacturing facilities and what inventory policies should be applied. In order to assist the company in their decisions the authors develop a simulation model over the supply chain based on their object-oriented simulation framework, and use the NSGA-II algorithm for the MOO in which the optimisation objectives are to minimise the average total cost for each product unit and minimise the average demand response time. In the case study conducted for the textile industry the authors look into a supplier selection problem with the objective to evaluate new supply chain configuration because of new supplier selection and transportation links, evaluate the sensitivity of the solutions in respect to demand variations, evaluate the effect of uncertainty of the data for the reliability of the supply chain configuration and evaluate the effects of different inventory policies on the supply chain. As in the first case study they use GAs for the MOO with the aim of minimising the total cost and maximising the service level. In [40] the authors present a more comprehensive investigation of the same case study and in [41] the automotive case study is presented in a greater detail.

Amodeo et al. [42] present approach based on Petri nets where they first model the supply chains as batch deterministic and stochastic Petri nets and then they develop a multi-objective search engine for simulation based optimisation in order to evaluate inventory policies for the modelled supply chain. The presented case study is a three-echelon supply chain where the company wants to determine the optimal inventory policies between the entities. As in the previously mentioned article the authors use the NSGA-II algorithm in order to find the Pareto-optimal solutions where the two conflicting objectives are total inventory cost and service

level. In another publication [43] the authors have presented the same approach and case study, however, with a more comprehensive literature review and detailed description of their Petri net model. In both publications, they showed that with help of their approach the company involved in the case study could obtain much better inventory policies with reduced inventory costs and improved service levels than the policies that the company is using currently.

Researchers in [44] demonstrate the applicability of combining SD and MOO. The author presents a simplified version of the well-known beer game developed by Sterman [45] in the late 1980s. In this study only a two-echelon supply chain is modelled consisting of a wholesaler and a retailer. In contrast to the original beer game where the aim is to minimise the total costs for the entire supply chain in this study the aim of the optimisation is to investigate the trade-offs between the two conflicting minimisation objectives: wholesaler cost and retailer cost.

Mahnam et al. [46] investigate an assembly supply chain network and develop an inventory model where each production entity can have several suppliers feeding the entity but it only has one subsequent predecessor. Other uncertainty parameters such as customer demand and supplier reliability are represented by utilising fuzzy sets. The aim of the study is to determine the order-up-to level for each storage-keeping unit (SKU) in the supply chain. To meet the aims the authors purpose a hybrid approach where simulation optimisation strategy and particle swarm algorithms are combined for a two-objective optimisation problem. They use a multi-objective particle swarm algorithm (MOPSO) with an elitist strategy where the efficient solutions are kept during the generations. The strategy also evaluates the particles and compares the swarm particles with the non-dominated solutions and updates the Pareto front. The objective function regarded in the study is to minimise the cost satisfying an appropriate fill rate of the SKUs.

Authors in [47] propose a DES and optimisation model for investigating supply chain design problem that considers supply chain operations and end-of-life process under an uncertain environment. Their study aims to analyse original equipment manufacturers (OEMs) capability to reconfigure their supply chains and end-of-life operations. In the study they consider three performance measures: (1) total cost, (2) environmental impact and (3) rate of market fulfilment. These performance measures are also transformed for the MOO where the aim is to minimise the total cost and environmental impact whilst maximising the delivery performance. GAs are used for finding the Pareto-optimal solutions and the optimisation is run for four end-of-life scenarios. The first scenario considers disposing the products immediately after they have been collected. In scenario two, the product are disassembled and the components are reused for product remanufacturing. The third scenario investigates the outcome of when the broken products are disposed whereas the healthy products are stored in OEMs inventories for reuse and redistributed into the market depending on the demand. The fourth scenario combines the second and third scenario i.e. healthy products are stored for reuse, however they are disassembled after a while when they no longer fulfil the redistribution conditions and the components are then reused in the product remanufacturing. The results show that scenario three i.e. reuse of products gives

the best results in terms of average performance whereas scenario two is the poorest.

The concepts of MAS is used in [48] in which the supply chain is modelled, simulated and managed by computational agents with the objectives to minimise the lead time and maximise the revenue using NSGA-II algorithm. The author developed a model in which the multi-objective, multi-criteria and multi-role nature of supply chains is represented by utilising and combining evolutionary MOO and multi-criteria decision making within the autonomous agents. The agents in the model have the possibility to take multiple roles as clients selecting supplier and suppliers managing their production. Furthermore, besides optimising their manufacturing strategy they also have the ability to modify their decision parameters for supplier selection using analytical hierarchy process (AHP). Implementing this approach the author presents an agent based simulator (ABS) model, which includes functionalities such as discovery of supplier service, manufacturing, service bidding, shipping, multi-criteria supplier selection, internal parameter optimisation etc. The research questions of the study were divided into three simulation experiments to: (1) compare the local parameter optimisation of suppliers with classical reactive strategy aiming to investigate its local parameter optimisation that generates better global results, (2) compare the performance of the multi-objective multi-role supply chain with a single-objective supply chain approach, and (3) compare the multi-objective multi-role supply chain with a single-role supply chain. The result showed that the multi-objective multi-role supply chain approach outperformed the supply chains that had single objective or single role. Among other things it was also observed that the internal optimisation feature allowed agents to earn more revenue and obtain shorter lead times when compared with runs without the optimisation.

The same authors in [39–41] present a supplier selection problem in [49]. Addressing the problem they develop a simulation optimisation methodology composed of GA which optimises the supplier selection, discrete event simulator (DES) in order to evaluate operational performance and a framework for modelling supply chains. The case study presented in this paper is based on a part of a supply chain for boot distribution, where the overall objective is to redesign the supply chain by selecting new suppliers and evaluating different solutions in terms of overall cost and robustness to changes in demand etc. The main difference between the current paper and the previous ones is that in this paper the authors explain the simulation optimisation methodology in a greater extent.

### ***2.2.3 Other Modelling Techniques***

In this section we have gathered two categories of papers; first category includes all the papers that do not utilise mathematical or simulation approaches, e.g., Multi Agent Systems or other program-based approaches. The second category includes the papers which the authors have not explicitly specified what approach they utilise.

Mansouri [50] investigates a multi-objective batch sequencing problem between two successive stages in a supply chain of a kitchen manufacturer in order to coordinate setups between the two stages in a flow-shop manner. The author point out that different processes might have different preferences on how to group products into batches e.g., before the cutting process you might group them by shape or material whilst before the paint shop parts might be grouped according to colour. Accordingly, in an assembly system the parts may be arranged to fit the final product. This procedure indicates that each stage will try to minimise its own total setup costs, however the objectives of the stages might be conflicting. Outlining the problem the author aims to minimise the total number of setups in two stages and minimise the maximum number of setups in each stage. Doing so, the author proposed a *multi-objective simulating annealing* (MOSA) solution approach to discover the Pareto-optimal solutions. The MOSA approach is also compared with an existing MOGA approach. The case study presented in the paper is based on the production chain of a kitchen manufacturer, where they consider 32 cutting groups and 14 colour groups and the plant applies an assemble-to-order production strategy. Finishing the study the author observed that the proposed MOA was capable of finding Pareto-optimal solutions and outperformed the MOGA in respect to quality of the discovered solutions.

In [51] the authors present a model over a multi-objective supply chain inventory optimisation problem with the aim of calculating base-stock-levels in a serial supply chain. The problem was first solved as a single-objective inventory cost problem and subsequently as a MOO problem considering two cost objectives, namely, holding cost and shortage cost. In the single-objective study the authors aim to obtain optimal installation base-stock policies in order to minimise total supply chain cost. For the optimisation they implement and compare standard GA with three variations of GA, namely, *gene-wise GA* (GGA), *random-key GA* (RKGA) and *random-key gene-wise GA* (RKGGA). After presenting the result for the single-objective study, in which they found the RKGGA to be the best approach, the authors present the multi-objective study with a key question in mind: how would the total supply chain cost and the holding cost be affected if the supply chain were to increase its service levels in comparison to the results gained from the single-objective solutions? Addressing this question the authors develop a MOGA for supply chain inventory problem (MOGA-SCIP) for generating the non-dominated solutions, with the objective to minimise the total holding cost and the total shortage cost. Completing the study, 183 non-dominated solutions were discovered and the authors argued that the MOGA-SCIP could be implemented to other multi-objective inventory problems within supply chains with simple modification. Pokharel [52] presents another multi-objective supply chain design problem proposing a deterministic simulation model involving minimisation of cost and maximisation of the reliability of supply from one entity to another. For the MOO the author proposes to use the *STEPmethod* to locate the non-dominated solutions.

Authors in [53] address a supply chain scheduling problem that considers the availability of both internal and outsourced machines with the objective to

minimise the utilisation of the outsourced machines and the total number of late orders. In their proposed model the authors assume that there are similar machines available within the internal organisation as well as the outsourced to process a number orders. But the manufacturing costs at the outsourced locations are higher than those of internal production, thus the focus on minimising the total external machine utilisation. For the MOO the authors use four heuristic approaches presented by Ho and Chang [54]; these approaches are later compared using three evaluation methods, namely, *best deviation method* (DEV), *integrated preference functional method* (IPF) and the *free disposal hull method* (FDH). Attaining the results the authors could see that the heuristic methods used here were not sufficient, and thus recommended the development of additional methods for the problem in order to generate non-dominated solutions.

In [55] the authors investigate optimisation of vehicle routing where multiple depots, multiple customers and multiple products have been considered. The model objective is to minimise total travel distance for all vehicles and the total time required to serve customers for all vehicles. To solve the optimisation problem the authors utilise a multi-objective evolutionary algorithm (MOEA) called *fuzzy logic guided non-dominated sorting GA II* (FL-NSGA II), where the fuzzy logic is implemented to dynamically adjust crossover rate and mutation rate in ten consecutive generations in order to improve the search performance of the MOEA approaches. To demonstrate the efficiency of their proposed algorithm the authors also compare it with five other MOEA algorithms, namely, the standard *NSGA II*, *strength Pareto evolutionary algorithm II* (SPEA II), *fuzzy logic guided strength Pareto evolutionary algorithm II* (FL-SPEA II), *micro-GA* (MICROGA) and *fuzzy logic guided micro-GA* (FL-MICROGA). Three scenarios were implemented to evaluate the algorithms; first scenario consisted of 5 depots and 50 customers, second scenario included 15 depots and 150 customers and the third scenario contained 25 depots and 250 customers. In order to compare the quality of the solutions from the algorithms the authors implemented two performance metrics; *the convergence metric* proposed by Deb and Jain [56] and the *spread metric* developed by Deb [57]. Final result from the study showed that the proposed algorithm (FL-NSGA II) was able to find non-dominated solutions with better convergence and diversity than the other algorithms examined in this study. When implementing the fuzzy logic guidance to the algorithms, they were able to attain better results than without it. The authors explain that this could depend on the ability of the fuzzy logic to adjust the crossover rate and the mutation rate to suitable values for various evolution states of the population.

## 2.3 Review Summary

This section concludes the presented literature review. Presenting this review summary we hope to give the reader opportunity, in a quick way, to overview the available literature treating MOO for supply chain management.

In summary, it has been observed from this literature review that the majority of the research conducted on MOO for SCM is based on mathematical approaches e.g., LP, MIP MILP etc. In comparison with the large amount of publications on applying simulation approaches to SCM problems, it seems that the exploration of using SBO, in the context of MOO, is far from adequate. As it can be seen in Table 2.1, we have tried to display the content of each paper considering MOO for SCM. The table is divided into five sections, *Article* that displays the article reference, *Modelling techniques* which show what approach has been used in order to model the supply chain, *Research scope* presents the main research scope of the paper, *Optimisation technique* displays the technique that has been utilised in order to attain Pareto-optimal solutions and *Optimisation objective* presents the various objectives that have been studied in the surveyed papers.

In total, 42 journals have been reviewed which concern MOO for SCM, ranging from various major international journals in management science and operational research. It should be noted that in some of the papers the authors have implemented several approaches e.g., [11, 33, 37], etc., the best approach/technique as defined by the authors in the paper has been highlighted by writing the favourable approach in *italic*.

### 2.3.1 Modelling Techniques

In this section we present a summary over which modelling approaches have been used the most in order to model supply chains. We would like to point out that in some papers (e.g., [33]) the author/authors have investigated several modelling approaches, however in our data collocation we have only considered the most favourable approach defined by the author/authors.

In terms of modelling approaches, Fig. 2.2 depicts that majority of papers, or more exactly 53% of them, have used a mathematical approach. By further investigating this approach we see in Fig. 2.3 that the most popular mathematical approach to model supply chains is by MINLP which counts for 33% of the papers that applies a mathematical approach. MINLP is followed by MILP as the second most implemented mathematical approach at 21%, the rest of approaches are fairly equally distributed, as shown in Fig. 2.3. The group called *Other* in Fig. 2.3 consists of approaches that have only been used once or twice such as, non-linear stochastic programming, MS Excel etc.

Coming back to Fig. 2.2, we observe that simulation approaches only count for 24% of implemented modelling techniques. This clearly indicates that more research is needed in this field i.e., performing MOO for supply chain management using a simulation model. The group *Na* includes the papers where a modelling technique could not be identified.



Table 2.1 List of papers and their contents

Article	Modelling technique	Research scope	Optimisation technique	Optimisation objective
Yimer and Demiri [10]	MILP	SC scheduling	GA	Min: Aggregated costs Max: Customer satisfaction
Al-Mutawah et al. [24]	Mathematical	SC design	Distributed GA	Max: Revenue for SC participant Utility for the entire SC
Altıparmak et al. [11]	MINLP	SC design	GA & Simulated annealing	Min: Total SC cost Max: Customer service level & Capacity utilisation for DCs
Amodeo et al. [42, 43]	Petri net	SC inventory planning	GA	Min: Total inventory costs Max: Customer service level
Azaron et al. [23]	MINLP	SC design & planning	Goal attainment	Min: Total investment costs, processing costs, transportation costs, shortage costs, capacity expansion costs, variance of the total cost, financial risk Min: Preferred path Max: Average profitability
Banarjee et al. [37]	Mathematical	SC scheduling	<i>Pareto bee colony with constructive rough set &amp; Ant colony optimisation &amp; Tabu search</i>	Min: Lead-time Max: Total revenue
Brintrup [48]	ABS	Supplier selection and SC planning	NSGA II	Min: Total cost, delivery time Max: Quality
Che and Chiang [34]	Na*	SC planning	mPaGA & PaGA	Max: Profit/entity, Avg. safety inventory levels/entity, Avg. customer service level, robustness of selected objectives, acceptability level of product price for buyer & seller.
Chen and Lee [13]	MINLP	SC scheduling	Fuzzy optimisation	

(continued)



Table 2.1 (continued)

Article	Modelling technique	Research scope	Optimisation technique	Optimisation objective
Chen et al. [19]	MINLP	SC production & distribution planning	Fuzzy optimisation	Max: Customer service levels, Safe inventory levels, Profit per entity Min: Total holding cost, Total shortage cost
Daniel and Rajendran [51]	Na*	SC inventory planning	MOGA-SCIP	Min: Budget Defect rate Max: Total purchasing value
Demirtas and Ozden [25]	MILP	Supplier selection	RLTP $\epsilon$ -constraint weighting method	Case study 1: Min: Average total cost/product unit, average demand response time
Ding et al. [39-41]	DES	SC design & planning [39, 41], supplier selection & SC planning [40]	NSGA II	Case study 2: Min: Total supply chain cost Max: Service level Mini: Overall supply chain cost, Lead time
Ding et al. [49]	DES	Supplier selection & SC design	GA	Min: Total reverse supply chain cost, Total tardiness of cycle time
Du and Evans [27]	MIP	Reverse logistics	Constraint method in combination with scatter search & dual simplex	Min: Wholesaler costs Retailer costs Min: Total cost Max: SC satisfaction
Duggan [44]	SD	SC inventory planning	GA	
Erol et al. [32]	Mathematical programming	Supplier selection & customer assignment	Pre-emptive goal programming	

(continued)

Table 2.1 (continued)

Article	Modelling technique	Research scope	Optimisation technique	Optimisation objective
Farahani and Asgari [29]	LP	Facility location	Utility function	Min: Location cost Max: Location quality
Franca et al. [22]	Nonlinear stochastic programming	SC quality & cost assurance	$\varepsilon$ -constraint	Mini: Total number of defects in raw material from supplier. Max: Total profit for the company
Guillén et al. [14]	MILP	SC design	$\varepsilon$ -constraint and branch & bound	Min: Financial risk Max: Net present value Demand satisfaction
Guillen-Gosalbez et al. [17]	MINLP	Supply chain design and planning	$\varepsilon$ -constraint and branch & bound	Min: Environmental impact Max: Net present value
Hugo et al. [12]	MILP	Supply chain design	Na*	Min: Greenhouse gas emissions Max: Net present value
Jayaraman [28]	MIP	Facility location	NISE	Min: Fixed cost, operating cost
Karpak et al. [30]	Na*	Supplier selection	VIG	Min: Purchasing costs Max: Product quality, Delivery reliability
Komoto et al. [47]	DES	SC design	GA	Min: Totals SC cost Environmental impact Max: Delivery performance

(continued)

Table 2.1 (continued)

Article	Modelling technique	Research scope	Optimisation technique	Optimisation objective
Lau et al. [55]	Na*	Vehicle routing	<i>FL-NGA II</i> , <i>NSGA II</i> , <i>SPEA II</i> , <i>FL-SPEA II</i> , <i>MICROGA</i> , <i>FL-MICROGA</i>	Min: Total travel distance Total time serving customer
Li and Zabinsky [33]	CCP, MIP & SP	Supplier selection	$\epsilon$ -constraint	Mini: Total cost Number of selected suppliers
Mansouri [50]	Na*	Batch sequencing	Simulated annealing & GA	Min: Total number of set-up multiple stages, total number of set-up at each stage
Mahnham et al. [46]	Fuzzy sets	SC inventory planning	Particle swarm	Min: Cost Max: SKUs fill rate
Mitra et al. [20]	MINLP, MILP, LP	SC planning	Fuzzy mathematical programming	Min: Cost Max: Demand satisfaction
Neto et al. [35]	LP	Reverse Logistics	$\epsilon$ -constraint	Min: Cost cumulative energy demand Waste
Narasimhan [26]	MS Excel	Supply selection	MINMAX	Mini: Total cost of procurement Max: Total product quality
Pokharel [52]	Na*	SC design	STEP	Total delivery performance Min: Cost Max: Supply reliability
Pishvae et al. [18]	MINLP	SC design	<i>Memetic algorithm</i> & GA	Min: Total cost Max: Responsiveness of the forward & reverse network
Ruiz-Torres et al. [53]	Na*	SC scheduling	Bi-criteria heuristics	Min: Utilisation of outsourced machines & late orders

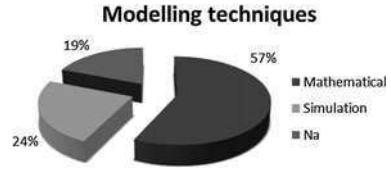
(continued)

Table 2.1 (continued)

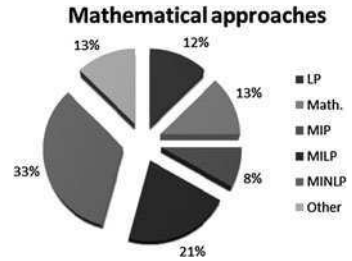
Article	Modelling technique	Research scope	Optimisation technique	Optimisation objective
Sabri and Beamon [15]	MILP	SC planning	$\epsilon$ -constraint	Strategic sub-model: Min: Total cost Max: Volume flexibility Operational sub-model: Min: Cost Max: Service levels Flexibility levels
Sheu [36]	LP	Reverse logistics	Composite	Min: Reverse logistic chain based cost Max: Power SC-based net profit
Wadhwa & Ravindran [31]	Na*	Supplier selection	Goal programming & weighted objective & compromise programming	Min: Purchasing cost Lead-time Max: Quality
Xu et al. [16]	Fuzzy-MINLP	SC design	Spanning tree-based GA & matrix based GA	Min: Total supply chain costs Max: Customer service level

Na\* information not available

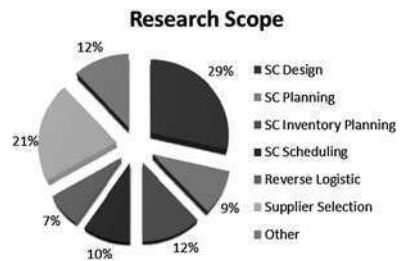
**Fig. 2.2** Modelling techniques



**Fig. 2.3** Mathematical approaches



**Fig. 2.4** Research scope

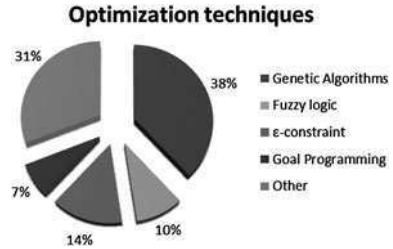


### 2.3.2 Research Scope

Here, we summarise the various research scopes that have been investigated in the reviewed journals. As with modelling techniques, some papers had more than one research scope e.g., [17] that investigates supply chain design and planning, [19] that examines supply chain production and distribution planning, etc. In this case we have only considered the main scope of the paper as presented by the authors, i.e. [17] has been categorised as supply chain design and [19] is regarded as supply chain planning paper.

Figure 2.4 shows the majority of research that has been conducted in the field of supply chain design, representing 29% of reviewed articles. This category is closely followed by another research, namely, supplier selection that accounts for 21% of the reviewed articles. Here, it could be argued that selecting new supplier changes the supply chain configuration and thus the supply chain design; however as we reviewed the papers we found that some papers only dealt with the issue of selecting an appropriate supplier but there were also papers that extended this

**Fig. 2.5** Optimisation techniques



notion and considered alternate supply chain configurations. In these cases we choose to classify these papers as a supplier selection problem as pointed out by the authors to be the main topic of the papers.

The rest research areas other than the two mentioned above show a quiet equal distribution among the surveyed papers.

### 2.3.3 Optimisation Techniques

In this section we summarise the various optimisation techniques that has been used in the reviewed papers. Similar to the previous two sections, we only consider one optimisation approach from each paper and categorise it into the following groups of optimisation techniques: GA, fuzzy logic,  $\epsilon$ -constraint and goal programming. The optimisation technique that is selected is the one that has proven to be the most suitable for the problem in hand and favoured by the authors.

The pie-chart presented in Fig. 2.5 shows that the most utilised optimisation technique has been GA, implemented in 38% of the reviewed papers. It should be pointed out here that in this category (i.e. GA) we include all approaches that are based on GA, e.g., NSGA-II, distributed-GA, memetic algorithm etc. The second largest category *other*, contains all the optimisation approaches that only has been implemented once such as Pareto Bee Colony optimisation, NISE, Particle Swarm, etc. whereas the other approaches (i.e. GA, fuzzy logic,  $\epsilon$ -constraint, etc.) have been implemented in several papers.

### 2.3.4 Optimisation Objectives

During the review, we found 101 optimisation objectives that had been investigated in the 42 reviewed papers. In this section we have categorised the vast amount of optimisation objectives in seven categories, namely, cost, time, profit/revenue, quality, environment, delivery/demand/service level (DDS) and other.

The *cost* category includes all the optimisation objectives related to cost such as, total/overall supply chain cost, transportation cost, purchasing cost, inventory cost, investment cost, etc. In the *time* category we have included all the

**Fig. 2.6** Optimisation objectives



optimisation objectives related to time, such as lead-time, preferred path, time serving customers, etc. As one can forebode the *profit/revenue* category includes the objectives where authors have tried to maximise the profit/revenue, and following the same pattern, *quality* and *environment* objectives relate to respective category. Two interesting categories are *DDS* and *other*; in *DDS* we include all the optimisation objectives related to demand satisfaction, delivery reliability and customer service level and in the *other* category we include optimisation objectives that are only used once or twice, such as number of selected suppliers, SKUs fill rate, utilisation of outsourced machines, late orders, etc.

As one can see from Fig. 2.6, 35% of the 101 optimisation objectives are somehow cost related, and the second most evaluated optimisation objectives are DDS-related.

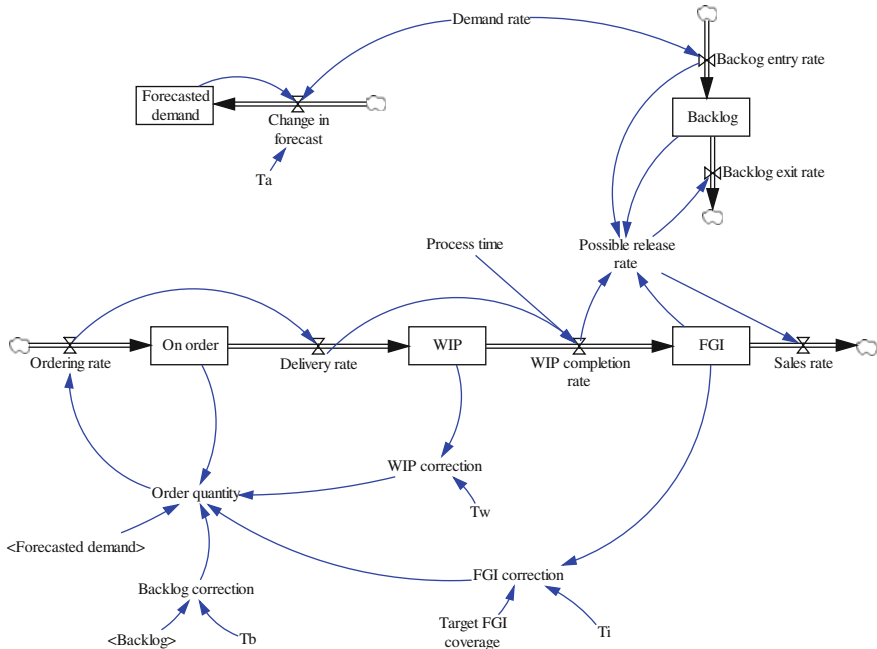
## 2.4 Simulation-Based MOO: A Case Study

Given the lack of simulation-based MOO applications, there should be plenty of problems, especially those that mathematical models cannot describe properly. In this section, we describe the optimisation of a system dynamics model that describes a classic production and inventory control problem.

The principal problem within supply chain management is concerned with finding a way to render the highest customer service at the lowest expense. Service is usually considered as either delivery time or as fill rate, while expense can be expressed in various dimensions, such as unit cost and tied-up capital requirements. Given that there are two (or more) objectives, the problem is multi-objective by design. Maintaining inventory is a preferred way of keeping service levels high while keeping production rates stable, which serves to lower the cost of production. However, keeping and managing inventories has been proven to increase the volatility of production, rather than to decrease it. This happens due to the desire to maintain a desired level of safety stock at any time, meaning that discrepancies from the desired stock level will be compensated for when placing production orders. The consequence is that the stock-keeping policy not only transfers variability upstream, but amplifies it. This phenomenon is termed the “bullwhip effect,” and is known for inducing large swings in production orders, especially when several tiers of production are concerned.

**Table 2.2** Objective functions

Goal	Parameter	Objective
Bullwhip	MAX(order quantity)	Minimise
Capital tied-up	WIP + FGI	Minimise
Service level	Backlog	Minimise



**Fig. 2.7** The system dynamics model

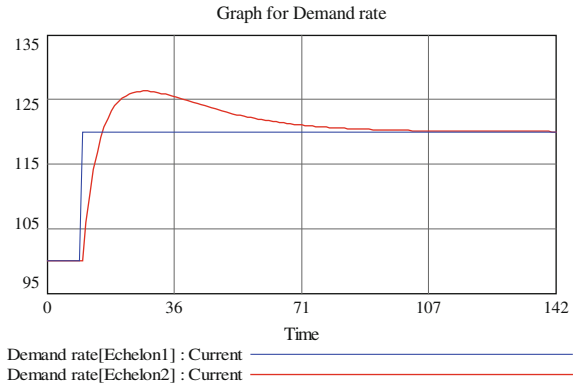
The initial discovery of the effect was done in [58] that developed a simulation model to describe the inventory swings observed at a company. Later developments include [59–62]. A comprehensive review is presented in [63], which covers the area from a control-theoretic perspective. The bullwhip problem is of interest, because it considers the basic trade-off between three different dimensions, being production rate, inventory levels and service level. Using a modified version of the well-established APIOBPCS model (see [60]), we demonstrate what insights multi-objective optimisation can give, even for simple problems that have been thoroughly explored by single-objective analyses. The objectives are presented in Table 2.2.

The model in Fig. 2.7 reflects the APIOBPCS model with the exception of an added backlog, which records the demand, which may not be fulfilled immediately; orders remain in the backlog until they are fulfilled. The “On order” stock, which mirrors eventual stock-outs on the supply side is also an addition over



**Table 2.3** Optimisation parameters

Stocks	Controlling parameters
FGI	$T_i$ , desired coverage
WIP	$T_w$
Backlog	$T_b$
Forecasted demand	$T_a$

**Fig. 2.8** Input signal (Echelon 1) and output signal (Echelon 2)

APIOBPCS; however, it is not used in the experiment as supply-side shortages are not considered.

The only decision rule investigated in the model is the reorder quantity; orders are placed continuously, so there is no need to consider order timing or whether order quantities are economically optimal. Equation 2.1 shows the ordering policy, with the two discrepancies shown in Eqs. 2.2 and 2.3.

$$\begin{aligned} \text{Order quantity} &= \text{Forecasted demand} - \text{On order} + \text{WIP discrepancy}/T_w \\ &\quad + \text{FGI discrepancy}/T_i + \text{Backlog}/T_b \end{aligned} \quad (2.1)$$

$$\begin{aligned} \text{WIP discrepancy} &= \text{WIP} - \text{Desired WIP} \\ &= \text{WIP} - \text{Lead time} \times \text{Forecasted demand} \end{aligned} \quad (2.2)$$

$$\begin{aligned} \text{FGI discrepancy} &= \text{FGI} - \text{Desired FGI} \\ &= \text{FGI} - \text{Desired FGI coverage} \times \text{Forecasted demand} \end{aligned} \quad (2.3)$$

The parameters used in the model are shown in Table 2.3; as only one echelon is optimised,  $T_a$  is not considered due to its limited effect on the dynamics of the model.

The demand rate used in the experiment is a step increase from a base level of 100 units of demand to 120. Figure 2.8 shows the demand rate and an example of a possible resulting order rate; note that the order rate peaks well above the step increase, representing how the system adjusts its stocks to reach the new equilibrium.

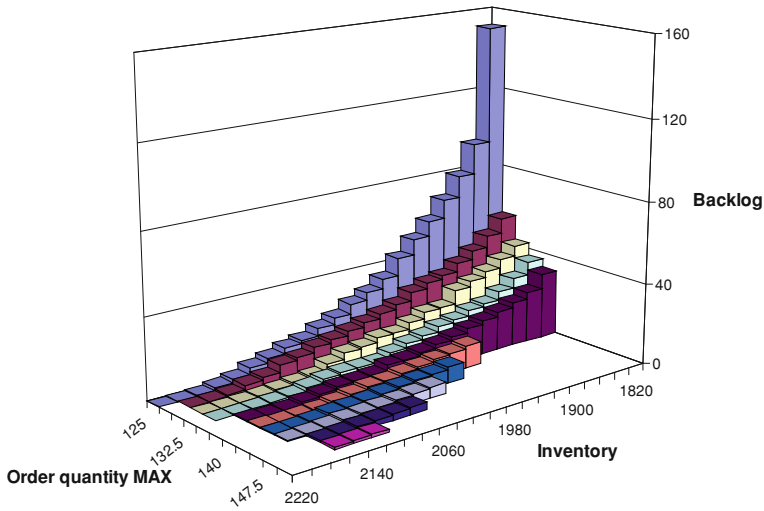


Fig. 2.9 Optimisation results

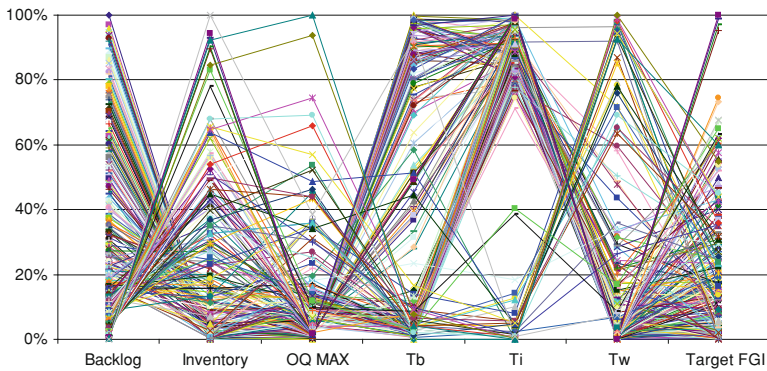


Fig. 2.10 Parameter configurations for a set of random Pareto-optimal solutions

Using the NSGA-II [64] algorithm to optimise backlog, max order quantity and inventory gives the results in Fig. 2.9 (a small number of points have been interpolated to complete the Pareto front). To attain the lowest backlog, a combination of inventory and production rate variance (maximum order quantity) is required. A key insight that the optimisation shows is that if only inventory and backlog were to be considered, production variance (max order quantity) would be high; however the same service can be attained with less variance in production rate if slightly more inventory is held.

With this simple experiment, it has been possible to show the trade-off between production order quantity variance (which often is a determinant of production cost), inventory (which determines how much capital must be tied up), and backlog (which reflects the average delivery delay). The insight gained is that

production stability can be achieved by keeping inventory in excess of the bare minimum, and at the same time adjusting the control parameters of the production system, especially  $T_i$  and  $T_w$  (see Fig. 2.10).

While the insights gained from MOO of theoretical experiments are of some value, the greatest benefit may well come from applying it to industrial scenarios, where multiple conflicting goals must be taken into account.

## 2.5 Conclusions

In this chapter, we presented a literature review of MOO for supply chain management. In total, 42 journal papers were reviewed which concern MOO for SCM, ranging from various major international journals in management science and operations research. The review concerned four main topics, namely, which modelling techniques have been used, which research scopes have been popular, which type of optimisation algorithms have been implemented and what optimisation objectives have been investigated.

The review showed that majority of papers have used a mathematical approach such as LP, MINLP, MILP etc. to model supply chains. Some researchers have used simulation techniques such as DES, SD, ABS etc. but in comparison to the mathematical approaches, implementation of simulation approaches were lacking. Hence, this clearly indicates that more research is need in the field of simulation-based multi-objective optimisation for supply chain management. For the research scope, supply chain design and supplier selection were the two most investigated topics. From the review, we found 101 optimisation objectives that had been investigated in the 42 reviewed papers in which cost and objectives related to time, such as lead-time, preferred path, time serving customers etc., were the mostly investigated. The most implemented optimisation algorithm was GA and approaches based on GA such as NSGA-II, Distributed-GA, Memetic algorithm etc.

In this chapter we also presented a case study that shows an example of how simulation models (in this particular case, an SD model) can be better understood by using MOO. The model in question was a modified version of APIOBPCS, popularly used when approaching the bullwhip problem. By adjusting the desired inventory coverage, as well as three feedback controllers, it was possible to choose whether demand volatility should be buffered by quickly changing the production rate, or by the finished goods inventory. A MOO of the model provided Pareto-optimal solutions across the dimensions of production rate, inventory and backlog. Apart from indicating the trade-offs, the optimisation results also showed the required parameter configurations, something which can be used when designing real supply chain systems. To address specific industry problems, companies wishing to utilise simulation-based MOO should create models specific to their supply chain, to get a better correspondence between model and reality. After a model is created, the approach demonstrated in this chapter can be used to evaluate how the supply chain performs, and how it can be optimally configured.

## References

1. Cohen, M. A., & Lee, H. L. (1989). Resource deployment analysis of global manufacturing and distribution networks. *Journal of Manufacturing and Operations Management*, 2, 81–104.
2. Arntzen, B. C., Brown, G. G., Harrison, T. P., & Trafton, L. L. (1995). Global supply chain management at digital equipment corporation. *Interfaces*, 25, 69–93.
3. Voudouris, V. T. (1996). Mathematical programming techniques to debottleneck the supply chain of fine chemical industries. *Computers and Chemical Engineering*, 20, S1269–S1274.
4. Jayaraman, V., & Pirkul, H. (2001). Planning and coordination of production and distribution facilities for multiple commodities. *European Journal of Operational Research*, 133, 394–408.
5. Amiri, A. (2006). Designing a distribution network in a supply chain system: formulation and efficient solution procedure. *European Journal of Operational Research*, 171, 567–576.
6. Meixell, M. J., & Gargeya, V. B. (2005). Global supply chain design: a literature review and critique. *Transportation Research Part E*, 41(6), 531–550.
7. Vidal, C. J., & Goetschalckx, M. (1997). Strategic production–distribution models: a critical review with emphasis on global supply chain models. *European Journal of Operational Research*, 98, 1–18.
8. Deb, K. (2001). *Multi-objective optimisation using evolutionary algorithms*. Chichester: Wiley.
9. Kadadevaramath, R. S., & Mohanasundaram, K. M. (2008). Evolutionary multiobjective decision making in supply chain revenue management: A literature review. *International Journal of Revenue Management*, 2(2), 137–179.
10. Yimer, A. D., & Demirli, K. (2010). A genetic approach to two-phase optimisation of dynamic supply chain scheduling. *Computers & Industrial Engineering*, 58(3), 411–442.
11. Altıparmak, F., Gen, M., Lin, L., & Paksoy, T. (2006). A genetic algorithm approach for multi-objective optimisation of supply chain networks. *Computers & Industrial Engineering*, 51(1), 196–215.
12. Hugo, A., Rutter, P., Pistikopoulos, S., Amorelli, A., & Zoia, G. (2005). Hydrogen infrastructure strategic planning using multi-objective optimisation. *International Journal of Hydrogen Energy*, 30(15), 1523–1534.
13. Chen, C.-L., & Lee, W.-C. (2004). Multi objective optimisation of multi echelon supply chain networks with uncertain product demands and prices. *Computers & Chemical Engineering*, 28(6–7), 1131–1144.
14. Guillén, G., Mele, F. D., Bagajewicz, M. J., Espuña, A., & Puigjaner, L. (2005). Multi objective supply chain design under uncertainty. *Chemical Engineering Science*, 60(6), 1535–1553.
15. Sabri, E. H., & Beamon, B. M. (2000). A multi-objective approach to simultaneous strategic and operational planning in supply chain design. *Omega*, 28(5), 581–598.
16. Xu, J., Liu, Q., & Wang, R. (2008). A class of multi-objective supply chain networks optimal model under random fuzzy environment and its application to the industry of Chinese liquor. *Information Sciences*, 178(8), 2022–2043.
17. Guillen-Gosalbez, G., & Grossmann, I. (2010). A global optimisation strategy for the environmentally conscious design of chemical supply chains under uncertainty in the damage assessment model. *Computers & Chemical Engineering*, 34(1), 42–58.
18. Pishvae, M. S., Farahani, R. Z., & Dullaert, W. (2010). A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Computers & Operations Research*, 37(6), 1100–1112.
19. Chen, C.-L., Wang, B.-W., & Lee, W.-C. (2003). Multiobjective optimisation for a multienterprise supply chain network. *Industrial & Engineering Chemistry Research*, 42(9), 1879–1889.
20. Mitra, K., Gudi, R. D., Patwardhan, S. C., & Sardar, G. (2009). Towards resilient supply chains: uncertainty analysis using fuzzy mathematical programming. *Chemical Engineering Research and Design*, 87(7), 967–981.

21. McDonald, C. M., & Karimi, I. A. (1997). Planning and scheduling of parallel semicontinuous processes: 1. production planning. *Industrial and Engineering Chemistry Research*, 36, 2691–2700.
22. Franca, R. B., Jones, E. C., Richards, C. N., & Carlson, J. P. (2009). Multi-objective stochastic supply chain modeling to evaluate tradeoffs between profit and quality. *International Journal of Production Economics*, <http://www.sciencedirect.com/science/article/B6VF8-4XVRYM3-1/2/ff388e5186e3235bf2e280869f1b5148>.
23. Azaron, A., Brown, K. N., Tarim, S. A., & Modarres, M. (2008). A multi-objective stochastic programming approach for supply chain design considering risk. *International Journal of Production Economics*, 116(1), 129–138.
24. Al-Mutawah, K., Lee, V., & Cheung, Y. (2006). Modeling supply chain complexity using a distributed multi-objective genetic algorithm (vol. 3980, pp. 586–595). Lecture Notes in Computer Science, Berlin: Springer.
25. Demirtas, E. A., & Ozden, U. (2008). An integrated multi-objective decision-making process for multi-period lot-sizing with supplier selection. *Omega*, 36(4), 509–521 (Special Issue on Logistics: New Perspectives and Challenges).
26. Narasimhan, R., Talluri, S., & Mahapatra, S. (2006). Multiproduct, multicriteria model for supplier selection with product life-cycle considerations. *Decision Sciences*, 37(4), 557.
27. Du, F., & Evans, G. W. (2008). A bi-objective reverse logistics network analysis for post-sale service. *Computers & Operations Research*, 35, 2617–2634.
28. Jayaraman, V. (1999). A multi-objective logistics model for a capacitated service facility problem. *International Journal of Physical Distribution & Logistics Management*, 29(1), 65–81.
29. Farahani, R. Z., & Asgari, N. (2007). Combination of MCDM and covering techniques in a hierarchical model for facility location: a case study. *European Journal of Operational Research*, 176, 1839–1858.
30. Karpak, B., Kumcu, E., & Kasuganti, R. R. (2001). Purchasing materials in the supply chain: managing a multi-objective task. *European Journal of Purchasing & Supply Management*, 7, 209–216.
31. Wadhwa, V., & Ravindran, R. (2007). Vendor selection in outsourcing. *Computers & Operations Research*, 34(12), 3725–3737.
32. Erol, I., & Ferrell, W. G., Jr. (2004). A methodology to support decision making across the supply chain of an industrial distributor. *International Journal of Production Economics*, 89, 119–129.
33. Li, L., & Zabinsky, Z. B. (2010). Incorporating uncertainty into a supplier selection problem. *International Journal of Production Economics*, <http://www.sciencedirect.com/science/article/B6VF8-4XSJVMB-2/2/2b225fe98a5f3bb6361257d75b200ff8>.
34. Che, Z. H., & Chiang, C. J. (2010). A modified Pareto genetic algorithm for multi-objective build-to-order supply chain planning with product assembly. *Advances in Engineering Software*, 41(7–8), 1011–1022.
35. Quariguasi Frota Neto, J., Walther, G., Bloemhof, J., van Nunen, J. A. E. E., & Spengler, T. (2009). A methodology for assessing eco-efficiency in logistics networks. *European Journal of Operational Research*, 193(3), 670–682.
36. Sheu, J.-B. (2008). Green supply chain management, reverse logistics and nuclear power generation. *Transportation Research Part E: Logistics and Transportation Review*, 44(1), 19–46.
37. Banarjee, S., Dangayac, G. S., Mukherjee, S. K., & Mohanti, P. K. (2008). Modelling process and supply chain scheduling using hybrid meta-heuristics. In *Metaheuristics for Scheduling in Industrial and Manufacturing Applications* (vol. 128 of Studies in Computational Intelligence, pp. 277–300). Heidelberg: Springer.
38. Fu, M. C., Glover, F., & April, J. (2005). Simulation optimisation: a review, new development, and applications. In *Proceedings of the 2005 Winter Simulation Conference* (pp. 83–95). Orlando, FL.
39. Ding, H., Benyoucef, L., & Xie, X. (2006). A simulation-based multi-objective genetic algorithm approach for networked enterprises optimisation. *Engineering Applications of Artificial Intelligence*, 19(6), 609–623.

40. Ding, H., Benyoucef, L., & Xie, X. (2008). Simulation-based evolutionary multi-objective optimisation approach for integrated decision-making in supplier selection. *International Journal of Computer Applications in Technology*, 31(3/4), 144–157.
41. Ding, H., Benyoucef, L., & Xie, X. (2009). Stochastic multi-objective production-distribution network design using simulation-based optimisation. *International Journal of Production Research*, 47(2), 479–505.
42. Amodeo, L., Chen, H., & El Hadji, A. (2007). Multi-objective supply chain optimisation: an industrial case study. In *Applications of Evolutionary Computing* (vol. 4448, pp. 732–741). Berlin: Springer.
43. Amodeo, L., Chen, H., & El Hadji, A. (2007). Supply chain inventory optimisation with multi-objectives: an industrial case study. In *Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management* (vol. 144, pp. 211–230). Berlin: Springer.
44. Duggan, J. (2007). Using system dynamics and multiple objective optimisation to support policy analysis for complex systems. In *Complex Decision Making* (pp. 59–81). Berlin: Springer.
45. Serman, J. D. (1989). Modeling managerial behaviour: misperceptions of feedback in a dynamic decision making environment. *Management Science*, 35(3), 321–339.
46. Mahnam, M., Yadollahpour, M. R., Famil-Dardashti, V., & Hejazi, S. R. (2009). Supply chain modeling in uncertain environment with bi-objective approach. *Computers and Industrial Engineering*, 56(4), 1535–1544.
47. Komoto, H., Tomiyama, T., Silvester, S., & Brezet, H. (2009). Analyzing supply chain robustness for OEMs from a life cycle perspective using life cycle simulation. *International Journal of Production Economics*, Available online <http://www.sciencedirect.com/science/article/B6VF8-4XT3HM4-1/2/9c370f65041e794302ad6536aac249b5>.
48. Brintrup, A. (2010). Behaviour adaptation in the multi-agent, multi-objective and multi-role supply chain. *Computers in Industry*, 61(7), 636–645.
49. Ding, H., Benyoucef, L., & Xie, X. (2005). A simulation optimisation methodology for supplier selection problem. *International Journal of Computer Integrated Manufacturing*, 18(2), 210–224.
50. Mansouri, S. A. (2006). A simulated annealing approach to a bi-criteria sequencing problem in a two-stage supply chain. *Computers & Industrial Engineering*, 50, 105–119.
51. Daniel, S. R., & Rajendran, C. (2006). Heuristic approaches to determine base-stock levels in a serial supply chain with a single objective and with multiple objectives. *European Journal of Operational Research*, 175, 566–592.
52. Pokharel, S. (2008). A two objective model for decision making in a supply chain. *International Journal of Production Economics*, 111(2), 378–388.
53. Ruiz-Torres, A. J., Ho, J. C., & Lopez, F. J. (2006). Generating Pareto schedules with outsource and internal parallel resources. *International Journal of Production Economics*, 103(2), 810–825.
54. Ho, J. C., & Chang, Y.-L. (1995). Minimizing the number of tardy jobs for m parallel machines. *European Journal of Operational Research*, 84(2), 343–355.
55. Lau, H., Chan, T., Tsui, W., Chan, F., Ho, G., & Choy, K. (2009). A fuzzy guided multi-objective evolutionary algorithm model for solving transportation problem. *Expert Systems with Applications*, 36(4), 8255–8268.
56. Deb, K., & Jain, S. (2004). Evaluating evolutionary multi-objective optimization algorithms using running performance metrics. In K. C. Tan, M. H. Lim, X. Yao, & L. Wang (Eds.), *Recent Advances in Simulated Evolution and Learning* (pp. 307–326). Singapore: World Scientific Publishers.
57. Deb, K. (2001). *Multi-objective optimisation using evolutionary algorithms*. Chchester, UK: Wiley.
58. Forrester, J. (1958). Industrial dynamics: a major breakthrough for decision makers. *Harvard Business Review*, 36(4), 37–66.

59. Towill, D. R. (1982). Dynamic analysis of an inventory and order based production control system. *International Journal of Production Research*, 20(6), 671–687.
60. John, S., Naim, M. M., & Towill, D. R. (1994). Dynamic analysis of a WIP compensated decision support system. *International Journal of Manufacturing System Design*, 1(4), 83–297.
61. Lee, H. L., Padmanabhan, V., & Whang, S. (1997). Information distortion in a supply chain: the bullwhip effect. *Management Science*, 43(4), 546–558.
62. Disney, S. M., Naim, M. M., & Potter, A. (2004). Assessing the impact of e-business on supply chain dynamics. *International Journal of Production Economics*, 89, 109–118.
63. Sarimveis, H., Panagiotis, P., Tarantilis, C. D., & Kiranoudis, C. T. (2008). Dynamical modeling and control of supply chain systems: a review. *Computers & Operations Research*, 35, 3530–3561.
64. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, 6(2), 181–197.

# Chapter 3

## State-of-the-Art Multi-objective Optimisation of Manufacturing Processes Based on Thermo-Mechanical Simulations

Cem Celal Tutum and Jesper Hattel

**Abstract** During the last couple of decades the possibility of modelling multi-physics phenomena has increased dramatically, thus making simulation of very complex manufacturing processes possible and in some fields even an everyday event. A consequence of this has been improved products with respect to properties, weight/stiffness ratio and cost. However this development has mostly been based on “manual iterations” carried out by the user of the relevant simulation software rather than being based on a systematic search for optimal solutions. This is, however, about to change because of the very tough competition between manufacturers of products in combination with the possibility of doing these highly complex simulations. Thus, there is a crucial need for combining advanced simulation tools for manufacturing processes with systematic optimisation algorithms which are capable of searching for single or multiple optimal solutions. Nevertheless, despite this crucial need, it is interesting to notice the very limited number of contributions in this field and consequently this makes us wonder about the underlying reasons for it. The understanding of the physical phenomena behind the processes, the current numerical simulation tools and the optimisation capabilities which all mainly are driven by the industrial or academic demands as well as computational power and availability of both the simulation and the multi-objective optimisation oriented software on the market are the main concerns to look for. These limitations eventually determine what is in fact possible today and hence define what the “state-of-the-art” is. So, seen from that perspective the very definition of the state-of-the-art itself in the field of

---

C. C. Tutum (✉) · J. Hattel  
Section of Manufacturing, Department of Mechanical Engineering,  
Technical University of Denmark, Kgs. Lyngby, Denmark  
e-mail: cctu@mek.dtu.dk

J. Hattel  
e-mail: jhat@mek.dtu.dk



optimisation of manufacturing processes constitutes an important discussion. Moreover, in the major research fields of manufacturing process simulation and multi-objective optimisation there are still many issues to be reserved.

### 3.1 Introduction

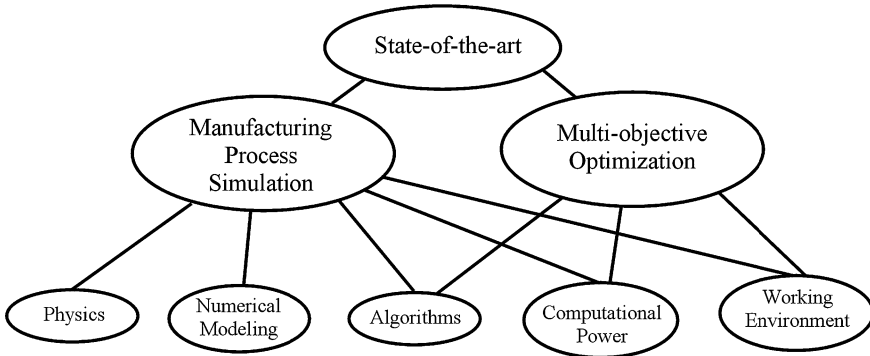
Manufacturing processes have been and still are exposed to a major transformation because of unforeseen dynamic challenges arising in different production scales varying from huge cast parts to the current trend of miniaturised devices such as cell phone lenses. The emergence of new materials and the evolving interaction between natural sciences and engineering applications promotes this transformation in a substantial way. Both design and manufacturing practices are inspired from nature and living objects day by day. This challenge drives engineers and scientists to explore “bottom-up” approaches rather than traditional “top-down” approaches to manufacture today’s highly complex products. It is quite worthwhile to give the following quotation which belongs to a forming expert in one of the leading car manufacturing companies, Schacher emphasising the pressure on the manufacturing industry leading to drastic reduction of development periods via “virtual production” technology: “In the past we introduced three new models every 10 years, now we introduce 10 new models every 3 years” [1]. This approach obviously requires a good understanding of the interaction of multi-disciplinary research fields, e.g., thermal, mechanical, flow, magnetic, electro-static, etc.,. Consequently, manufacturers have to define, most likely, several success and/or failure criteria concerning different aspects of these disciplines, next evaluate the designs and finally construct a robust work frame for different process conditions. Efforts in this direction of research, which aim at simultaneous improvements in process efficiency and product quality, are typically conflicting with each other, i.e., better product quality requires higher production technology, therefore each solution, the so-called “trade-off”, having different combinations of importance of these goals gives an idea of how much gain could be obtained and sacrifice could be accepted. Different numerical modelling approaches are utilised to evaluate the performance of these trade-off designs related to the manufacturing processes, where it is most likely not possible to have closed form solutions in terms of design variables, e.g., process parameters, geometry, etc., Computational fluid dynamics (CFD), computational solid mechanics (CSM) and computational materials science (CMS) are such branches of numerical modelling used for simulation of these processes and these in general require high computational power varying from hours to days or even weeks. Thus, engineers have to use some approximation methods in order to reduce the number of high fidelity simulations, in other words, the computational cost. Moreover, high performance computing has almost become a standard tool to exploit the inherent parallelism built into some stochastic or statistical algorithms used both in the evaluation and exploitation phases of

simulation and optimisation. Moreover, their population-based search strategy could seem to be overkill for a single objective optimisation; however it is a perfect fit for multi-objective optimisation problems. Some examples of exploiting numerical simulation together with multi-objective optimisation have resulted in, for instance, (i) reducing the residual stresses in a welded mechanical component together with a simultaneous improvement of the production efficiency (e.g., higher welding speed) in case of friction stir welding [2–4], (ii) increasing the casting yield via riser optimisation meanwhile reducing the porosity in a gravity sand-cast steel part [5] or (iii) optimising the chemistry of bulk metallic glasses for improved thermal stability [6], (iv) optimising alloy composition of high temperature austenitic stainless steels for desired mechanical and corrosion resistant properties [7], as well as (v) improving the formability of the fender drawing process by minimising wrinkle tendency, thinning ratio and spring back caused by elastic deformation by controlling blank-holding force and draw-bead restraining force parameters [8], and a few other challenging manufacturing applications [9–18]. These are some of the very limited number of already analysed examples of among many potential real world multi-objective optimisation problems in manufacturing.

This chapter is outlined as follows. First, a brief discussion of the modelling of thermal and mechanical phenomena as well as the combination of the two are given followed by two important application fields, i.e., friction stir welding and metal casting processes. Apart from the obvious influence of the past history of these processes, i.e., the metal casting process could be said to be much more mature as compared with the friction stir welding process which has been developed just two decades ago, also the process simulation level and the performed optimisation studies in each of the fields have their own share of this historical development. The presentation of the different studies should be seen in this light. Moreover, the focus on these two specific topics is biased towards the relative expertise of the authors in these two research fields based on thermo-mechanical simulations however the thoughts and ideas put forward in the present chapter are to a large extent also applicable to optimisation of other branches of manufacturing processes.

## 3.2 What Determines State-of-the-Art?

In order to be able to give an overview of the research in multi-objective optimisation of manufacturing process based on numerical simulations with focus on thermo-mechanical aspects, the major actors behind the stage should be introduced. This attempt will, or should, eventually initiate the following question: what is meant by the state-of-the-art and what makes it state-of-the-art? Here, this question will constitute the basis of fundamental discussions in this challenging, integrated research area which is still very much in its infancy. This initial discussion of current capabilities, resources and limitations will naturally emphasise



**Fig. 3.1** State-of-the-art in manufacturing process simulation and multi-objective optimisation composed of a network of interrelated issues

as well as hopefully clarify some of the important determining factors and moreover give some directions awaiting future quests in this interesting field.

The development in the field of applying optimisation for manufacturing processes and ultimately the products that they produce is affected by many factors, see Fig. 3.1: (i) First of all, the most important physical phenomena should be understood before anything else and for some processes there is still a substantial demand concerning this issue. (ii) Regarding the mathematical models describing these physical phenomena, some have been developed recently and others have evolved into more advanced levels. This gives huge variations in quality and applicability of these models. (iii) The computational power is obviously an important factor in itself. For matured processes with well-established mathematical models, it will be a determining bottle-neck, however, for cases where both processes and simulation tools are still less-developed the bottle-neck for development will be somewhere else. (iv) The degree of available solution algorithms can of course be very determining for the development. For the process simulation models the important issue here will be the quality of the solvers for the resulting algebraic equation systems as well the efficiency of the numerical schemes used for iterations due to non-linearities whereas for the optimisation part the available MOO-algorithms at hand will be determining, i.e., classical or evolutionary algorithms, DOE studies, machine/statistical learning and so forth.

### ***3.2.1 Lack of Knowledge About the Physical Phenomena***

Many manufacturing processes that seem straightforward at first glance are in fact extremely complex being governed by multi-physical behaviour. Although metal casting has been around for several thousands of years, it is a process

which involves phenomena like fluid flow with free surfaces during mould filling, phase change and microstructural development during solidification and build-up of residual stresses and deformations during solid state cooling. All these phenomena involve complex models and for some alloy systems like ductile cast iron the knowledge of what really happens in the microstructure during e.g., solidification is very limited. In solid state processes, like forging the material flow is relatively well described however for friction stir welding which can be characterised as a hybrid between a thermal and a mechanical process a lot of questions are still unanswered. This for instance goes for the material flow under the rotating tool as well as the origin of tunnel defects and worm holes. Despite all these uncertainties the processes are used on a daily basis, producing products even though all the mechanisms are not fully understood and described.

### ***3.2.2 Development of Numerical Models***

The level and applicability of the different mathematical models for process simulation is of course linked closely to how well the physics, that they are supposed to describe, is understood. So, even though there are many general purpose software systems around for solid mechanics, fluid mechanics and to some extent also materials sciences it does not necessarily mean that sufficient mathematical models are readily available. It always depends very much on the purpose of the model and also the experience of the user of the simulation system. For optimisation purposes it is also reasonable to ask the question whether we actually need all the physical details if only trends and tendencies are sufficient to come closer to a more optimised solution. In these cases simpler solutions based on “coarse-meshed numerical simulations” or even closed form analytical solutions might very well be applicable.

### ***3.2.3 Computational Power***

The rapid evolution of technology in terms of processors (multi-core CPUs or many-core GPUs [19]), networks (Infiniband [20]), architectures (GRIDs, clusters), memory (shared, e.g., [21] or distributed, e.g., [22]), storage capacity, etc., (i.e., the components which eventually serve for the computational power) has made the parallelisation very popular for both numerical modelling and optimisation applications [23]. In particular, parallelisation is even more tempting for stochastic and mostly population-based evolutionary single- or multi-objective optimisation algorithms which are inherently built for parallel execution [23–25]. Alongside with this, the capability to approximate the physics and search for an optimum set of design variables is improving with an unpredictable pace;

however the level and the amount of questions arising from the numerical tools are also increasing with an even higher rate. For instance, the lack of availability of multi-objective optimisation case studies of manufacturing process simulations based on residual stresses points out the challenge clearly; the level of complexity of the numerical manufacturing process models steer the resources to be mostly used at the fitness evaluation level rather than the distribution of them.

### ***3.2.4 Algorithms***

Numerical simulation of manufacturing processes in general requires efficient ways of solving the field problems, e.g., thermal, mechanical or flow fields as well as the interaction of these, expressed in terms of Partial Differential Equations (PDEs) [26]. As opposed to applying continuous functions in closed-form solutions (valid for an infinite number of points), the domain is discretised into a finite number of elements, volumes or cells in numerical methods. The particular arrangement of these elements constitutes the mesh. Expressing the PDEs on discretised form in this mesh leads to a system of algebraic equations (be it linear or non-linear) in the unknowns to be solved for at the nodes, i.e., the connection points of the elements. These algebraic equations can be written in terms of matrices and vectors, and they can be solved either in a direct manner or iteratively depending on the characteristics of the system, such as band structure, sparseness as well as the nonlinearity arising from the terms of the unknown field variable [27–30]. This nonlinearity, be it a material nonlinearity in the case of high temperature manufacturing process simulations such as in casting, or a geometrical nonlinearity in the case of high deformations experienced in the FSW process while stirring the workpiece material, should be solved in a reasonable manner. Whatever formulation is used, e.g., finite element method (FEM), finite difference method (FDM) or finite volume method (FVM) as well as smooth particle hydrodynamics (SPH) or arbitrary Lagrangian–Eulerian (ALE), the numerical solution algorithms at hand need to be developed aiming at a better performance, i.e., either better accuracy or lower solution time; hence developers also have to bear in mind to exploit the currently available hardware architecture in parallel. This is also the case in optimisation as well. There are quite a number of successful evolutionary multi-objective optimisation (EMO) algorithms which already have been applied in engineering optimisation applications [24, 31–37], but a few coupled with manufacturing process simulations. The computational expense is obviously the most critical limitation. However, on the other hand, the population-based search strategy of the EMO algorithms gives the user an important possibility to obtain multiple optima in a single run. Covering all the relevant challenges in this field is beyond the scope of the present work, however further details can be found in the following references [24, 25, 37–39].

### 3.2.5 Working Environment

The working environment, in other words the software resources at hand, is one of the most decisive factors in accomplishing state-of-the-art work. The software available in the market, which enables users (from different, but related engineering or manufacturing backgrounds) to apply the multi-objective optimisation based on the manufacturing process simulations, could be sorted into commercial versus freeware and dedicated versus general purpose ones. Although there are dozens from each, it would still be worthwhile to mention some of them to motivate newcomers to this field who are willing to initiate their own applications. Development of general purpose commercial multi-physics simulation software (e.g., ANSYS, ABAQUS, Msc. NASTRAN, MARC, COMSOL, LS-DYNA, DEFORM, etc.) dates back as compared with commercial multi-objective optimisation software (e.g., modeFRONTIER, iSIGHT, OPTIMUS, etc., [40]) in parallel to the development of the theory and demands as well as technological trends. Besides these, dedicated commercial software (e.g., WELDSIM, MORFEO, MAGMASOFT, etc.) which is built for more specific purposes provide users the possibility of investigating some of the physical phenomena in more detail and in shorter time since the time to develop the model is more straightforward. In this respect, as expected, the general purpose codes call for more fundamental knowledge about the application, e.g., the physical phenomena needs to be decomposed into thermal, metallurgical, flow and mechanical models in case of simulation of a welding process, since the user needs to express most of the physical phenomena, e.g., boundary conditions, load steps, etc., in terms of mathematical expressions available in the program. This might even be more challenging in case of freely downloadable (open source) multi-physics codes (Code Aster, Elmer, OpenFOAM, PARAFEM, etc.) since the capabilities could be either limited or technical support is not available (this is even more crucial in debugging), apart from the online forums. However, availability of the source code sometimes gives some users, who have strong background in that specific application and experience in programming, the possibility to implement more detailed features that are not available in commercial software. This is mostly preferred in research and development departments of some industrial companies and universities for education or advanced research as well as consultancy purposes. This need is also somehow fulfilled in commercial software by opening some backdoors for users to integrate their own subroutines (e.g., distributed heat flux or visco-elasto-plastic material models, etc.) written in Fortran, C, Python, etc., or other general purpose programming languages. The integration of this varying set of software with multi-objective optimisation software is another challenge as it calls for even further knowledge although the severity of difficulties is reduced in case of using commercial tools. However, as aforementioned, the pros and cons in using open source codes also apply in the application of multi-objective optimisation. It is interesting to mention that some of these open source codes (e.g., NSGA-II [31], SPEA [32], NIMBUS [41], etc.) became very successful, therefore popular, and consequently implemented in several commercial competitors.

**Table 3.1** Governing (heat conduction) equations in thermal models with respect to different reference frames and time domains [45, 46]

Reference frame	Transient	Steady-state
Lagrangian	$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + q_{\text{vol}}$ (3.1a)	$0 = \nabla \cdot (k \nabla T) + q_{\text{vol}}$ (3.1b)
Eulerian	$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + q_{\text{vol}} - \rho c_p u \nabla T$ (3.1c)	$0 = \nabla \cdot (k \nabla T) + q_{\text{vol}} - \rho c_p u \nabla T$ (3.1d)

### 3.3 Essence of Thermo-Mechanical Modelling

The theory of thermo-mechanics is not just a trivial extension to ordinary solid mechanics, although it may seem that way when looking at the governing equations. But there is more to it than that. For example, in the purely mechanical case positive strain is normally accompanied by positive stress, i.e., tension. In other words when we pull (tension) in each end of a bar, it expands (positive strain). For thermally induced stresses and strains, it is very often the other way around. That is, even though a body which is locally heated will expand, i.e., positive strain, the stress state itself will typically be in compression, i.e., negative stress. Therefore, in this case we have a positive strain and a negative stress. This might seem difficult to understand for a newcomer to the field of thermo-mechanics, but it is easily explained with the use of the terms *elastic* strain, *thermal* strain and *total* strain, which are given later in Eq. 3.3. Indeed, this example does not cover all differences between solid mechanics at room temperature and thermo-mechanics at elevated temperatures. On the other hand, however, it serves very well to show just one of the principal differences [42].

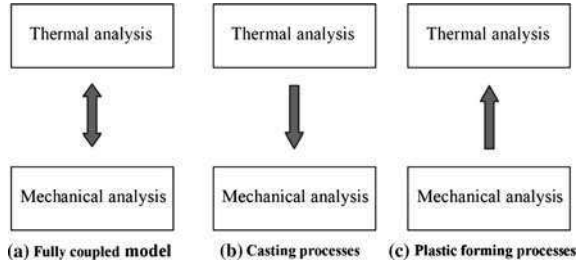
Apart from discussions of specific details on numerical nonlinearities, couplings, discretisation of time, heat generation, boundary conditions and material property, the essence of pure thermal modelling can be understood as the solution of the heat conduction equation given in Eq. 3.1 [43, 44], for the general case,

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + q_{\text{vol}} \quad (3.1)$$

where  $\rho$  denotes the material density,  $c_p$  the specific heat capacity,  $T$  the temperature field to be solved,  $k$  the thermal conductivity, and  $q_{\text{vol}}$  the volumetric heat source term. This time-dependent problem can be solved equivalently in both Lagrangian and Eulerian reference frames with an appropriate set of initial and boundary conditions in the case of a moving heat source, see Table 3.1 for an overview. The combination of the choice of reference frame and degree of enmeshed modelling level offers a huge range of possibilities, and the “correct” choice depends on the objective of each model.

For calculation of the transient as well as the residual stress field during many manufacturing processes, a standard mechanical model based on the solution of the three static force equilibrium equations can be used, i.e.

**Fig. 3.2** Typical *main* couplings in thermo-mechanical analyses [42]



$$\sigma_{ij,i} + p_j = 0 \quad (3.2)$$

where  $p_j$  is the body force at any point within the calculation domain and  $\sigma_{ij}$  is the stress tensor. The well-known Hooke's law and the linear decomposition of the strain tensor as well as small strain theory are applied together with the expression for the thermal strain, i.e.

$$\begin{aligned} \varepsilon_{ij}^{tot} &= \varepsilon_{ij}^{el} + \varepsilon_{ij}^{pl} + \delta_{ij}\varepsilon^{th} \\ \sigma_{ij} &= C_{ijkl}^{el}\varepsilon_{kl}^{el} = C_{ijkl}^{el}\left(\varepsilon_{ij}^{tot} - \varepsilon_{kl}^{pl} - \delta_{kl}\varepsilon^{th}\right) \\ C_{ijkl}^{el} &= \frac{E}{1+\nu} \left[ \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \frac{\nu}{1-2\nu}\delta_{ij}\delta_{kl} \right] \\ \varepsilon_{ij}^{tot} &= \frac{1}{2}(u_{i,j} + u_{j,i}) \\ \varepsilon^{th}(T_1 \rightarrow T_2) &= \int_{T_1}^{T_2} \alpha dT \end{aligned} \quad (3.3)$$

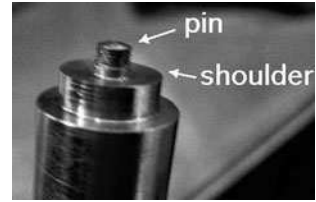
where  $\varepsilon_{ij}^{tot}$  denotes the total strain,  $\varepsilon_{ij}^{el}$  the elastic strain,  $\varepsilon_{ij}^{pl}$  the plastic strain,  $\varepsilon_{ij}^{th}$  the thermal strain,  $\sigma_{ij}$  the stress,  $C_{ijkl}^{el}$  the elastic constitutive (stiffness) tensor,  $\nu$  the Poisson's ratio and  $u_i$  the displacement field to be solved for. The plastic strain can often be assumed to follow the standard  $J_2$ -flow theory with a temperature-dependent Von Mises yield surface. More information on the numerical treatment of this in an FE-framework can be found in e.g., the textbook by Simo and Hughes [28].

After this brief overview on thermal and mechanical fields and the governing equations for each, a few last words should be spent on coupling frames to sum up the major thermo-mechanical modelling issues. Depending on the purpose of the thermo-mechanical simulation, numerical coupling frames can be classified in three major sets according to the main driving force for the simulated physics: (a) fully coupled model, (b) thermally driven semi-coupled model, and (c) mechanically driven semi-coupled model [42, 43], as shown in Fig. 3.2.

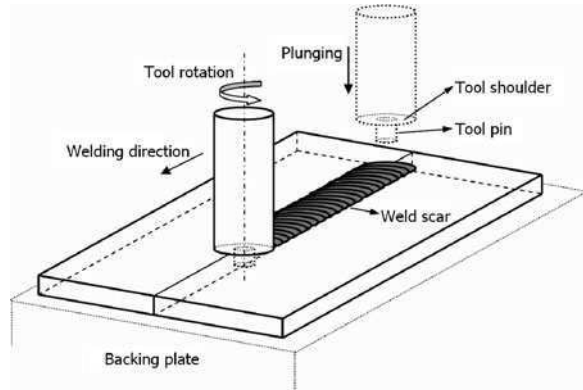
In the first case (a), the unknowns in both the thermal and the mechanical fields, i.e.,  $T$  and  $u$ , are solved simultaneously (e.g., friction stir welding process), while in the second case the thermal analysis is performed first and the mechanical



**Fig. 3.3** A standard tool having a cylindrical shoulder and probe (pin) design [49]



**Fig. 3.4** Schematic view of the FSW process



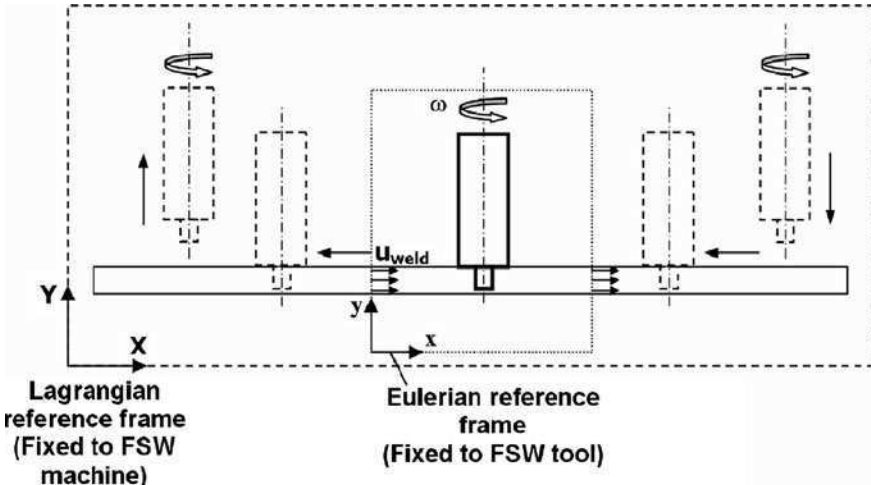
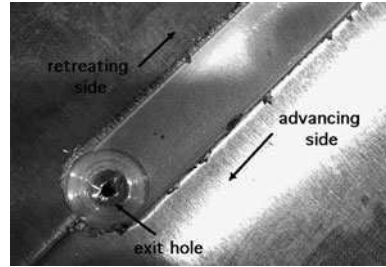
analysis afterwards (e.g., casting process) or vice versa in the third case (c) (e.g., sheet metal forming process).

### 3.4 Case Study-1: Friction Stir Welding

The FSW process is an efficient solid-state joining technique (i.e., the metal is not melted during the process) that was invented by Wayne Thomas and a team of his colleagues at The Welding Institute (TWI), UK, in December 1991 [47]. It is used especially for heat treated, high strength aluminium alloys which in general are difficult to weld with traditional welding techniques [47, 48].

Figure 3.3 shows a standard welding tool having a cylindrical shoulder and probe which in general are designed in different sizes and shapes with/without thread features or manufactured with different materials based on workpiece and process-specific needs or limitations. The process, which is schematically shown in Fig. 3.4, consists of several subsequent procedures denoted as plunging, dwelling, actual welding and pulling the tool out of the workpiece. First, the tool is submerged vertically into the joint line with high rotational speed in the plunge period and then dwelling takes place, where the tool is held steady relative to the workpiece while keeping rotation and heating the surrounding workpiece material locally. Following dwelling, the tool is moved forward while stirring two workpiece materials to be joint (welding period) and is pulled out of the workpiece leaving a key hole behind as seen in Fig. 3.5.

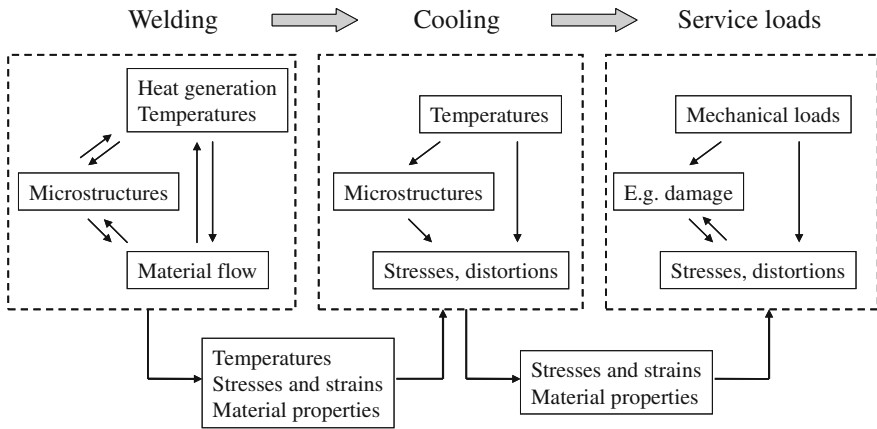
**Fig. 3.5** A welded structure having a key hole at the end [49]



**Fig. 3.6** The FSW process in different reference frames

These sequences have also been represented schematically in Fig. 3.6 emphasising different computational modelling approaches with respect to different reference frames, i.e., the Lagrangian (also known as “global approach” [2–4, 50], where transient effects are captured) and the Eulerian (“local approach” [51, 52], in general used for steady-state conditions).

In FSW, heat is generated by friction (mainly at the interface between the tool shoulder and the upper surface of the workpiece) and plastic deformation (by the tool probe or pin in the plunging stage and during the welding period via stirring the two workpiece materials along the joining line). The heat flows into the workpiece as well as the tool. The amount of heat conducted into the workpiece influences the quality of the weld, distortion and residual stress in the workpiece [48]. Insufficient heat generation from the tool shoulder and the probe could lead to failure of the tool pin as the workpiece material is not soft enough. Therefore, understanding the heat aspect of the FSW process, which is the main driving force for all subsequent coupled simulations, e.g., microstructure and solid mechanics models, is extremely important, not only for understanding the physical phenomena, but also for improving the process efficiency, e.g., welding faster and



**Fig. 3.7** Major modelling couplings in FSW during welding, cooling and loading [45]

safer [51, 52]. This complex coupling sequence is also represented schematically in Fig. 3.7.

Some examples of using numerical optimisation methods in combination with process modelling of FSW have been given in literature. Most of them are based on thermal models and they are typically targeted at obtaining optimal process parameters with respect to predefined single-objectives or used as a means of inverse modelling to obtain unknown thermo-physical material properties and a few will be mentioned in the following. Liao and Daftardar [53] implemented a thermal model in FLUENT in combination with two simpler surrogate models to investigate the performance of different optimisation algorithms for obtaining the three process parameters, heat input, weld speed and shoulder diameter. Tutum et al. [51] combined a gradient-based optimisation technique (i.e., SQP) with a simple analytical thermal model in order to obtain heat input and welding speed for a desirable average temperature distribution under the tool shoulder in the FSW process. The same process criterion is studied using space and manifold mapping by Larsen et al. [54]. An application of the differential evolution algorithm for reducing the uncertainty associated with specific process parameters, i.e., the friction coefficient, the extent of slip between the tool and the workpiece, the heat transfer coefficient at the bottom of the workpiece, the mechanical efficiency, and the extent of viscous dissipation converted to heat, is studied by Nandan et al. [55]. It should be mentioned that this application is based on a coupled viscoplastic thermal-flow model for FSW. A recent contribution is given by Tutum et al. [52], and this will be explained in detail in Sect. 3.4.3, it encompasses a 2-D steady state Eulerian TPM (thermal pseudo mechanical) heat source model including an analytically prescribed flow field with a hybrid evolutionary multi-objective optimisation algorithm (i.e., NSGA-II and SQP) to find multiple trade-off designs. The only example in literature so far, regarding optimisation of FSW based on residual stress calculations has been given by Tutum and Hattel [2], see Sect. 3.4.7

for further details. This work combines a thermo-mechanical model implemented in ANSYS (neglecting the material flow) with the NSGA-II algorithm investigating the optimal process parameters, i.e., the welding speed and rotational speed, to simultaneously minimise the peak residual longitudinal stress in the weld and increase the production rate which is related to welding speed.

### 3.4.1 Thermal Model

The core part of any thermal model of FSW is how the heat generation from the rotating tool is described and applied as either a boundary condition, or a source term in the heat conduction equation, or a combination of the two. Most pure thermal models apply a surface flux as representing the entire heat generation, thereby avoiding the source term in the heat conduction equation, which is governed by the plastic dissipation and therefore in essence calls for knowledge about the material flow and hence the formation of the shear layer. As there is no access to the mechanical field in a pure thermal model, the heat generation because of plastic work needs to be represented somehow! Several suggestions in literature are given for this surface flux formulation, but common for them all is the need for “calibration” parameters. If one has access to an experiment and it is possible to back out the total heat generation from the measurements, one can use the well-known expression given in Eq. 3.4 to obtain the surface flux.

$$\frac{q_{\text{total}}(r)}{A} = \frac{3Q_{\text{total}}r}{2\pi R_{\text{shoulder}}^3} \quad (3.4)$$

Moreover, if you have information about the friction coefficient and the total downward force from the tool, and you assume full sliding you can express the total heat generation as

$$Q_{\text{total}} = \frac{2}{3}\pi\omega R_{\text{shoulder}}^3\mu p \quad (3.5)$$

Either way, you need experimental information, so in the present work, the slightly different thermal model proposed by Schmidt and Hattel [56] is applied. In this model the heat generation is again expressed as a surface heat flux from the tool shoulder (without the tool probe) into the workpiece, however it is a function of the tool radius and the temperature dependent yield stress as follows

$$\frac{q_{\text{surface}}(r, T)}{A} = \omega r \tau(T) = \left(\frac{2\pi n}{60}\right) r \frac{\sigma_{\text{yield}}(T)}{\sqrt{3}}, \quad \text{for } 0 \leq r \leq R_{\text{shoulder}} \quad (3.6)$$

where  $n$  is the tool revolutions per minute,  $r$  is the radial position originating in the tool centre,  $R_{\text{shoulder}}$  is the tool shoulder radius and the temperature-dependent yield stress  $\sigma_{\text{yield}}$  is defined as

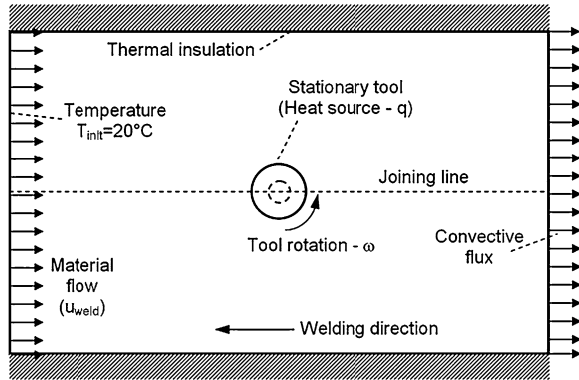
$$\sigma_{\text{yield}}(T) = \sigma_{\text{yield,ref}} \left( 1 - \frac{T - T_{\text{ref}}}{T_{\text{melt}} - T_{\text{ref}}} \right) \quad (3.7)$$

where  $\sigma_{\text{yield,ref}}$  is the yield stress at room temperature,  $T_{\text{ref}}$  is 20°C and  $T_{\text{melt}}$  is the solidus temperature (500°C). Once the temperature reaches the solidus temperature, i.e.,  $T$  becomes equal to  $T_{\text{melt}}$  in Eq. 3.7, the “self-stabilising effect” causes the heat source to “turn itself off”, i.e., the material loses all its resistance, and the heat generation decreases automatically because of thermal softening. The model is often denoted “thermal-pseudo-mechanical” as the heat generation is expressed via the temperature dependent yield stress, thus taking some mechanical effects into account, however, it should be underlined that the model is a purely thermal model involving a temperature-dependant heat generation and in that sense it also uses a “calibration” parameter like the more conventional procedure in Eqs. 3.4 and 3.5. Obviously, this adds a non-linearity to the thermal model, meaning that the calculation time is increased by roughly a factor of two as compared with other thermal models where the heat source is prescribed itself, like in Eq. 3.4. The heat source model expressed by Eqs. 3.6 and 3.7 has been validated against experimentally obtained thermal profiles, see e.g., [56] for a more detailed description. Finally, it should be mentioned that both expressions in Eqs. 3.4 and 3.5 can be directly derived from the more general formulation of analytically modelling the heat source in FSW given by the authors in [57, 58].

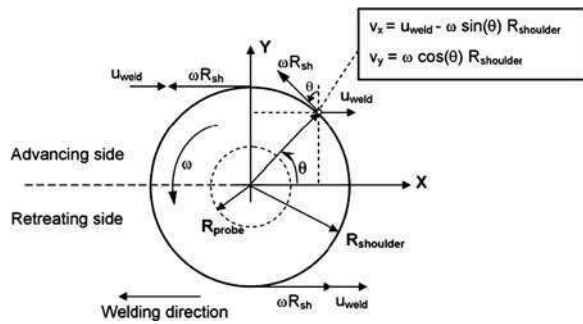
### 3.4.2 Implementation of Steady-State Eulerian Thermal Model in COMSOL

Arising out of the relatively high heat generation contribution from the surface of the tool shoulder, an assumption based only on modelling the tool shoulder is taken into account. The radius or in other words location of the tool probe is hypothetically included as a design variable to compute the first objective function (i.e., temperature difference) and temperature variation under the tool, between the tool shoulder and the probe, to be used in the decision making step. Modelling the whole welding process, i.e., plunge, dwell and pull out periods, holds some notable complexities. In order to reduce the computational cost regarding the moving heat source, meanwhile preserving the applicability, only the welding period is taken into account and a moving coordinate system (i.e., Eulerian reference frame) which is located on the heat source is applied. The shear layer formed below the tool shoulder caused by the high tool rotational speed is also included; hence an asymmetric temperature field along the joint line is obtained in the present numerical model. Equation 3.1d in Table 3.1, describes the steady-state heat transfer in the plate (the transient term on the left side disappears). The temperature field,  $T$ , is solved including the temperature-dependent material density ( $\rho$ ), the specific heat capacity ( $c_p$ ) and the heat conductivity ( $k$ ) of the AA2024-T3 (the workpiece material) respectively, besides  $u$  expressing the material flow

**Fig. 3.8** 2-D Eulerian steady state thermal model



**Fig. 3.9** Mathematical modelling of the flow field under the tool shoulder in detail



vector and  $q_{vol}$  which is the volumetric stationary heat source representing the tool as a circle in Fig. 3.8.

In the present model [52], the heat generation is a function of the tool radius and the temperature-dependent yield stress of the workpiece material ( $\sigma_y(T)$  of AA2024-T3) and assumed to be uniform through the thickness (3 mm) of the plates to be welded. Apart from the brief explanation in the former section, the details of this temperature and position dependent heat source model entitled as TPM model are given in detail in [56]. The traverse motion of the tool and the relatively complex flow field under it are modelled by prescribing a material flow through the rectangular plate region, as shown in Fig. 3.8. Because of this flow prescription, Eq. 3.1d includes a convective term ( $u$ ) in addition to the conductive term. The derivation of the mathematical prescription of the material flow is also schematically represented in Fig. 3.9 and components of the flow vector in the welding and the transverse directions are formulated for an arbitrary point on the periphery of tool shoulder as a function of  $\theta$  (in Cartesian reference frame). Equation 3.8 generalises the flow field description ( $u(\theta) = u(x, y) = (v_x, v_y)$ ) for the whole domain as follows,

$$u = \begin{cases} [u_{weld} - \sin(\theta)\omega R_{shoulder}, \cos(\theta)\omega R_{shoulder}] & \text{if } r(x, y) \leq R_{shoulder} \\ [u_{weld}, 0] & \text{if } r(x, y) > R_{shoulder} \end{cases} \quad (3.8)$$

where  $r(x, y)$  is the radius or the position vector. As a boundary condition, the room temperature (20°C) is defined at the left edge of the rectangular region where the tool is assumed to be moving towards. The heat flux on the right edge of the plate region, where the material leaves the computational domain, is dominated by convection. On the upper and lower edges of the plate boundaries, thermal insulation is enforced.

### 3.4.3 Optimisation Study

In this section, optimum process parameters and tool geometries in FSW are investigated to minimise the temperature difference between the leading edge of the tool probe and the workpiece material in front of the tool shoulder, i.e., to soften the material enough to move the tool probe forward without failure, and simultaneously to maximise traverse welding speed, hence production rate, subjected to hot and cold weld conditions [52]. More specifically, the choices of the tool rotational speed and the traverse welding speed together with the radii of the tool shoulder and the probe have been investigated in order to achieve the goals mentioned above which are in essence conflicting. The steady-state Eulerian thermal finite element model described in the previous section, with temperature-dependent thermo-physical (i.e., heat-treated aluminium alloy, AA2024-T3) material data has been implemented using the commercial multi-physics simulation software COMSOL for the function evaluations. An evolutionary multi-objective optimisation (EMO) algorithm, i.e., non-dominated sorting genetic algorithm (NSGA-II) is initially performed to find the Pareto-optimal front. The non-dominated solutions found so far have been clustered based on their Euclidean distances (in the objective space) in a prefix grid structure to reduce the number of the solutions, which in turn will be serving as initial starting points for the gradient-based local search technique, i.e., sequential quadratic programming (SQP). The  $\varepsilon$ -constraint method [59] is applied by fixing the second objective (i.e., welding speed) as a constraint for each clustered non-dominated solutions independently to obtain the modified optimised front. Further improvement in accuracy and confidence in the convergence of the Pareto-optimal front is achieved, and following this, a brief post-optimality study is performed to unveil some common design principles among members of the clustered Pareto-optimal set. Finally, two reasonable design solutions among those multiple trade-off solutions have been selected based on different characteristics of the temperature distribution under the tool shoulder induced by the material flow, tool selection and production rate preferences. More details are given in the following.

As briefly described above, the multi-objective optimisation problem (MOP), which is related to the thermal aspects of the FSW process, is formulated. Optimum process parameters, i.e., the tool rotational and traverse welding speeds ( $n_{rev}$  and  $u_{weld}$ ), and geometrical tool parameters, i.e., tool shoulder and probe radii ( $R_{shoulder}$  and  $R_{probe}$ ), are investigated to minimise the temperature difference

( $\Delta T$ ) between the leading edge of the tool probe and the workpiece material in front of the tool shoulder, and simultaneously to maximise traverse welding speed. The second objective, based on the duality principle, is reformulated as the minimisation of  $-u_{\text{weld}}$  due to the way of implementation of the EMO algorithm, i.e., MATLAB implementation of the original NSGA-II [31] algorithm by the first author [46]. This MOP problem is constrained with hot and cold weld conditions, geometrical constraints (the tool shoulder radius is desired to be 5 mm larger than the tool probe radius), besides lower and upper limits of the design variables. In order to evaluate hot and cold weld conditions, the average temperature ( $T_{\text{avg}}$ ) is computed under the tool shoulder, in other words, the temperature values on each element inside the circular region (i.e., Area =  $\pi R_{\text{sh}}^2$ ) are integrated and divided by the number of elements. The constrained multi-objective optimisation problem is given below

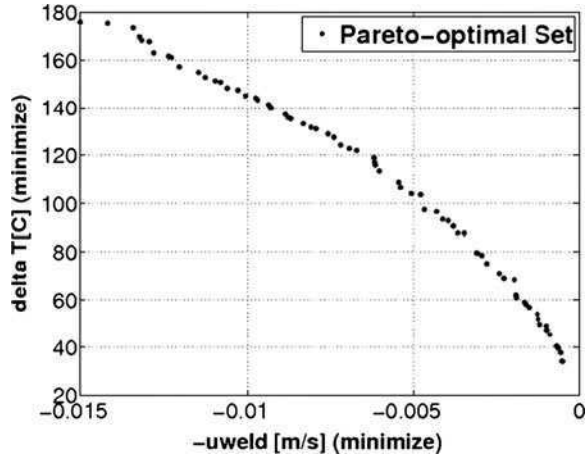
$$\begin{aligned}
 \text{Minimise: } f_1(x) &= \Delta T = T_{\text{probe}} - T_{\text{ahead}} \\
 \text{Minimise: } f_2(x) &= -u_{\text{weld}} \\
 \text{subject to: } g_1(x) &= 450^\circ\text{C} \leq T_{\text{avg}} \leq 500^\circ\text{C}, \\
 g_2(x) &= R_{\text{probe}} + 5 \text{ mm} \leq R_{\text{shoulder}}, \\
 g_3(x) &= 8 \text{ mm} \leq R_{\text{shoulder}} \leq 17 \text{ mm}, \\
 g_4(x) &= 3 \text{ mm} \leq R_{\text{probe}} \leq 12 \text{ mm}, \\
 g_5(x) &= 100 \text{ rpm} \leq n_{\text{rev}} \leq 1250 \text{ rpm}, \\
 g_6(x) &= 0.5 \text{ mm/s} \leq u_{\text{weld}} \leq 15 \text{ mm/s}.
 \end{aligned} \tag{3.9}$$

As mentioned above, NSGA-II, which is an EMO algorithm enabling finding well-spread multiple Pareto-optimal solutions for an MOP by incorporating three substantial features, i.e., elitism, non-dominated sorting, and diversity preserving mechanism (crowding distance), is used for the proposed constrained problem. The population size is 100 and the number of generations is fixed to 10 because of relatively high computational cost of the function evaluations, i.e., the simulation time for each set of designs is approximately 10 min on a PC having Core 2 CPU, 2.33 GHz, and 2 GB of RAM. Real variable-coding is used for the design variables. Therefore the simulated binary crossover (SBX) and the polynomial mutation [24], with distribution indices of 5 and 10, are used as crossover and mutation operators, respectively. Figure 3.10 shows all NSGA-II (non-dominated) solutions composing a non-convex Pareto-optimal front, having  $-u_{\text{weld}}$  on the horizontal axis and  $\Delta T$  on the vertical axis. As expected, the higher welding speeds result in higher temperature difference indicating steeper gradients in front of the tool, which is not desirable in case of limitations due to improper tool or machine designs. More detailed analysis of these trade-off designs is performed after the local search procedure which aims for further improvement in the convergence of the obtained trade-off frontier.

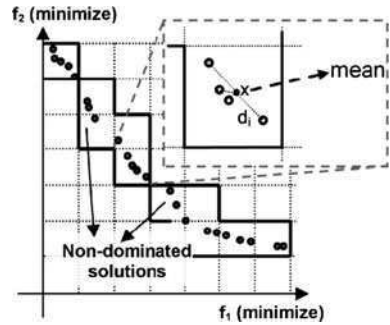
Prior to the local search step, the non-dominated solutions found so far are clustered simply based on their Euclidean distances (i.e., minimum  $d_i$ ) with respect



**Fig. 3.10** The non-convex Pareto-optimal front obtained with NSGA-II



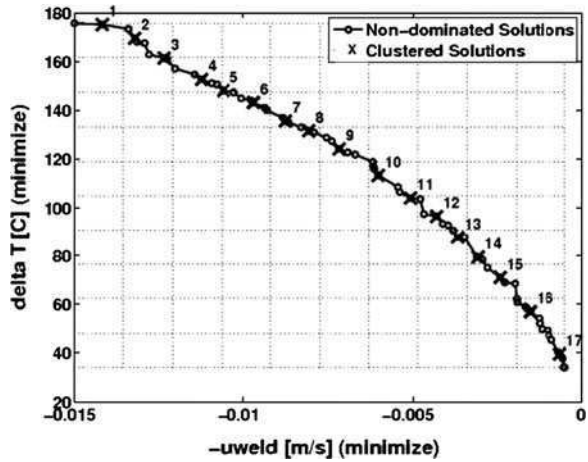
**Fig. 3.11** Clustering scheme



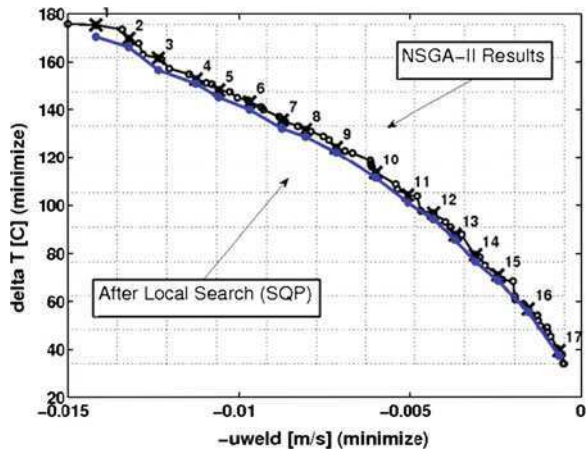
to their mean, which is computed in each cell and in each axis, in a prefix grid structure to reduce the number of the solutions (for the sake of computational cost), as represented in Fig. 3.11 on a hypothetically distributed points in the objective space. Figure 3.12 shows 17 clustered solutions, indicated by cross markers, out of 72 non-dominated solutions for the FSW problem in a 10-by-10 grid.

After completing the multi-objective optimisation task, a set of optimal solutions specifying the design variables and their trade-offs is obtained. If these optimal solutions are sorted according to the worse order of the first objective (min.  $\Delta T$ ), they would also get lined up in the second objective (min.  $-u_{\text{weld}}$ ) in an ascending order. Having such a wide variety of solutions provides a much better basis for the decision-making process as compared with having only one optimal solution. This enables engineers or designers to judge or plan the performance of a product or a process in a larger perspective in terms of sacrifices and gains with respect to multiple criteria [24]. Moreover, a basic post-optimality study can unveil interesting design knowledge that is common to all of these trade-off solutions or a partial set of them [60]. This design methodology, which was

**Fig. 3.12** Clustered non-dominated solutions indicated by crosses and the corresponding numbers on top of them



**Fig. 3.13** Pareto-optimal front modified after the local search on each clustered non-dominated solutions



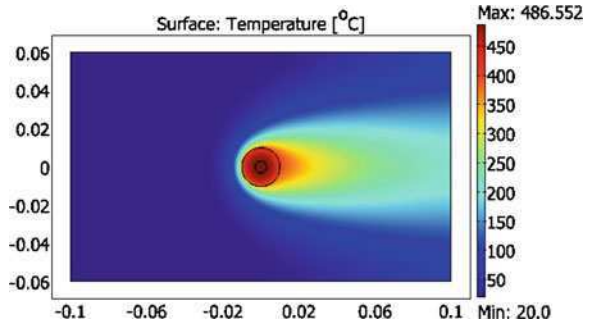
originally formulated as “innovization” (the creation of innovative knowledge through multi-objective optimisation) [60–63], has also been applied manually to the current FSW problem [52]. A partial set of design variables on the Pareto-curve (Fig. 3.13) have been listed in Table 3.2.

Three intervals can be distinguished looking at Table 3.2, i.e.,  $0.5 \text{ mm/s} \leq u_{\text{weld}} < 4 \text{ mm/s}$ ,  $4 \text{ mm/s} \leq u_{\text{weld}} < 10 \text{ mm/s}$  and  $10 \text{ mm/s} \leq u_{\text{weld}} < 14 \text{ mm/s}$ . In the first and the third intervals, the tool shoulder and probe radii are approximately the same (10 mm and 5 mm, respectively). In the middle interval, there is a significant increase in the tool dimensions, but there is also a common tendency to have a 5 mm difference in the radius dimensions (the second constraint is active) along the Pareto-front. Moreover, in the middle interval, as the tool shoulder is getting larger, the heat generation is increasing because of the increase in the frictional surface area (consequently, the hot weld condition becomes active),

**Table 3.2** Set of designs corresponding to some of the members on the modified Pareto-optimal front in Fig. 3.13

$R_{\text{shoulder}}$ (mm)	$R_{\text{probe}}$ (mm)	$u_{\text{weld}}$ (mm/s)	$n_{\text{rev}}$ (rpm)
10.932	5.155	13.224	1037.8
10.819	5.377	12.359	1014.9
10.896	5.831	11.261	1119.6
14.959	9.887	9.694	974.33
13.793	8.263	7.171	856.07
13.579	8.005	4.304	718.17
10.998	5.74	3.672	1057.7
11.113	5.527	1.539	880.33
10.694	4.581	0.643	968.65

**Fig. 3.14** Thermal field for a parameter set:  
 $u_{\text{weld}} = 11.3$  mm/s,  
 $R_{\text{shoulder}} = 10.5$  mm,  
 $R_{\text{probe}} = 5.5$  mm,  
 $n_{\text{rev}} = 1,100$  rpm



and the tool rotational speed ( $n_{\text{rev}}$ ) shows a decrease as compared with other two intervals on the Pareto-curve. In most of the designs, the distribution of the temperature field under the tool is almost uniform (i.e., the cold weld condition is not active), thus the standard deviation is close to the mean value, which is a desired process condition. The main criterion for the manufacturer to select one or two designs out of these possibilities would be welding speed which is related to investment and operating cost of different kinds of tool-machine combinations. In case of limited financial resources, the manufacturer or engineer would like to weld slower (need to sacrifice in production rate) in order to improve the lifetime of the tools. In this case, such a design set:  $u_{\text{weld}} = 2-3$  mm/s,  $n_{\text{rev}} = 700-800$  rpm,  $R_{\text{shoulder}} = 9-10$  mm and  $R_{\text{probe}} = 4-5$  mm would be preferable. In an opposite case, where financial limitations are negligible, the production rate would be a dominant criterion (e.g.,  $u_{\text{weld}} > 11$  mm/s), but similar tool geometries with higher rotational speeds would be sufficient (e.g., Figure 3.14).

### 3.4.4 Thermo-Mechanical Model

The maximum tensile residual stresses are typically found on, or at either side of, the weld line. The mechanisms behind the evolution of residual stresses in different welding processes, in general, are the same, only the magnitudes and distribution of

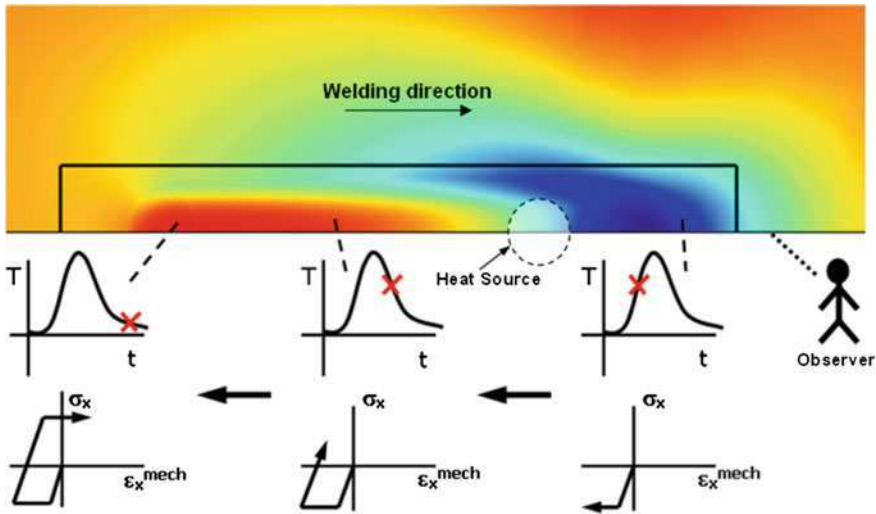


Fig. 3.15 The FSW process in different reference frames

these show some difference depending on the modelling of the heat sources. A schematic view of these thermo-mechanical mechanisms are shown in Fig. 3.15 on a half-plate (under symmetry assumption) clamped on the sides and described with respect to a fixed coordinate system (i.e., Lagrangian reference frame) represented by an observer standing on the lower-right side of the workpiece. The thermal history profiles are shown below the workpiece together with the corresponding longitudinal stress-(mechanical) strain curves in the welding direction for convenience since they are dominating. The heat source, i.e., the welding tool, is assumed to be moving from left to right with a constant speed. When the heat source is approaching the observer, where the workpiece material is still at room temperature which is relatively colder than the heat source, the material in front of the tool is heated up and expands meanwhile softening, but as it is constrained by the surrounding colder material, this causes compressive stresses as well as compressive plastic strains after exceeding the yield limit. It should be mentioned that the stress-strain curves shown at the lower row of the graphs in Fig. 3.15 for simplicity are schematically drawn under the assumption of ideal plasticity, i.e., no hardening after yielding as well as no temperature dependency of the yield stress, which is not the case in real applications, but still representative.

After the heat source passes by, the material in the joint line starts to cool down as seen from the schematic graphs in the middle column in Fig. 3.15. Following the cooling, tensile stresses (or shrinkage forces in other words) start to evolve caused by negative thermal strain increments, i.e., positive mechanical strain increments because of the constraint, in the longitudinal direction. At the last, left-most graphs, the workpiece cools to the reference room temperature and the tensile stresses, which have been following the stress-strain curve in the elastic regime, eventually reach to the critical level where the material yields in tension.

These tensile stresses, so-called residual stresses, lower the loading capacity of the component and the compressive plastic misfit situated at the end of the welding process causes distortion, i.e., shrinkage, in the plate unless some removal techniques, i.e., thermal and mechanical tensioning [64–66], shot peening [67, 68] and local-dynamic cooling [69] are applied.

Many contributions regarding modelling of residual stresses in FSW have been given in the literature [2, 50, 66, 70–86] and common for them all are that they somehow predict the thermally induced stresses arising from the welding process. Some models only consider the rotating tool as a moving heat source [2, 50, 66, 70–77, 86] whereas others take the coupling between the temperatures and the material flow into account [78–85]. All models take the thermal softening into account somehow, however while some just employ temperature dependent yield strength, others apply metallurgical models of varying complexity for predicting the evolution of e.g., hardness and thereby yield strength [87].

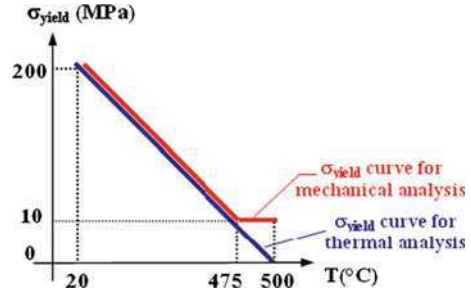
### ***3.4.5 Implementation of Thermo-Mechanical Model in ANSYS***

As described above, the semi-coupled thermo-mechanical model of the FSW process used in this work [2] consists of a transient thermal model and a quasi-static elasto-plastic mechanical model, which is accomplished by utilising the commercial finite element software ANSYS. In order to facilitate the automation of the optimisation procedure both models are implemented by means of the Parametric Design Language (APDL) of ANSYS.

The model represents the welding of two flat plates by considering the bead on plate. Arising out of symmetry assumptions, e.g., neglecting the asymmetric shear layer and the asymmetric heat source, only one of the plates is modelled. The dimension of the workpiece is  $300 \times 100 \times 3$  mm. This means that the thermally induced out-of-plane stresses will be negligible and a plane-stress analysis is reasonable. Regarding boundary conditions, the effect of the thermal contact with the backing plate is modelled by an equivalent heat transfer coefficient of  $700 \text{ W m}^{-2} \text{ K}^{-1}$  at the bottom of the workpiece and with an ambient temperature of  $20^\circ\text{C}$ . For mechanical boundary conditions, as the first assumption, the plates are assumed to be free (however fixed in one point in a corner to avoid rigid-body motion). This is obviously not the case for real FSW applications; however, this very important assumption is made in order to avoid the releasing step as it might give some problems with out of plane deformations that would complicate the optimisation analysis considerably. This way, we will still obtain residual stresses being in self-equilibrium but of course not entirely reflecting the right clamping history. This was for instance taken into account by the authors in [50] where the effect of different clamping conditions together with releasing and mechanical loading during in-service was investigated. However, for the present feasibility analysis with focus on the multi-objective optimisation methodology combined with residual stresses, the simpler approach as described above is used.

**Table 3.3** Temperature-independent material properties of benchmark material

Heat conductivity, $k$	( $\text{W m}^{-1} \text{K}^{-1}$ )	160
Heat capacity, $c_p$	( $\text{J kg}^{-1} \text{K}^{-1}$ )	900
Young's modulus, $E$	(GPa)	70
Tangent modulus, $E_t$	(GPa)	7
Thermal expansion coefficient, $\alpha$	( $\text{K}^{-1}$ )	$2.3 \times 10^{-5}$
Density, $\rho$	( $\text{kg m}^{-3}$ )	2,700

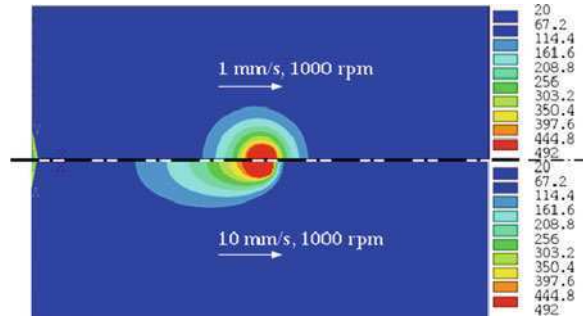
**Fig. 3.16** The FSW process in different reference frames

The temperature-independent material properties, which are given in Table 3.3, do not correspond to any specific commercial aluminium alloy but should be understood as a representative aluminium benchmark material. This assumption is in agreement with the study presented by Zhu and Chao [88], which concludes that the temperature-dependent yield stress has a significant effect on the residual stress and distortion, and except for this yield stress, using material properties at room temperature gives reasonable predictions of the transient temperature fields, the residual stresses and distortions. The temperature dependence of the yield stress for both thermal and mechanical analyses is shown in Fig. 3.16 as a linear function of temperature with a negative slope, decreasing from 200 MPa to 0 MPa at 20°C and 500°C respectively. There is an exception for the mechanical analysis that 475°C is chosen to be the cut-off temperature and the yield stress is kept constant at 10 MPa above this temperature. This engineering simplification of using a linear relationship together with a lower bound for the yield stress provides a substantial convenience for controlling computational cost as it reduces the nonlinearities that do not have a significant effect on the global behaviour of the thermo-mechanical model [88].

Because of the very low contribution to the heat generation coming from the tool pin, only the tool shoulder is considered in the heat source. The diameter of the tool shoulder is 20 mm. The mechanical effects of the tool are not included, and thus residual stresses are assumed to be primarily a function of the thermal load history [66, 71, 86]. The moving heat source starts and stops at 50 mm away from the left and the right edges of the plate, respectively. The SHELL 131, 4-Node layered thermal shell element is used for the transient thermal analysis while the PLANE 182, 2-D 4-node structural solid element is used for the quasi-static mechanical analysis and the same structured finite element mesh is used in both cases.

The accuracy of the thermal and mechanical simulation using shell and plane stress models has been compared with a 3-dimensional solid linear 8-node element

**Fig. 3.17** Contour plots of the temperature fields for rotational speed of 1,000 rpm for each traverse welding speed of 1 mm/s (*top*) and 10 mm/s (*bottom*), respectively



model. Here, SOLID 70 elements are used for the transient thermal analysis and SOLID 45 for the quasi-static mechanical analysis, respectively. As expected, when comparing the thermal profiles obtained at the position of the moving heat source when it is in the middle of the plate along the transverse direction as well as the longitudinal stress profiles there is very little difference between the plane stress model and the full 3-dimensional model. Thus, it can be concluded that for the present case the sequentially coupled shell and plane-stress models can be used for the purpose of doing preliminary optimisation studies while being both accurate and computationally efficient.

### 3.4.6 Parameter Study

Before doing the actual optimisation it is beneficial to do a parameter study with the simulation tool. This is also done in the present work. Figure 3.17 shows the contour plots of the resulting temperature field of the symmetric models (with increments of 47.2°C) for a chosen welding speed of 1 mm/s and 10 mm/s, respectively, for a rotational speed of 1,000 rpm. Figure 3.18 shows a parameter study for the thermal profiles along the transverse direction at the longitudinal location of the heat source for two welding speeds, i.e., 1 and 10 mm/s, and two rotational speeds, i.e., 100 and 1,000 rpm.

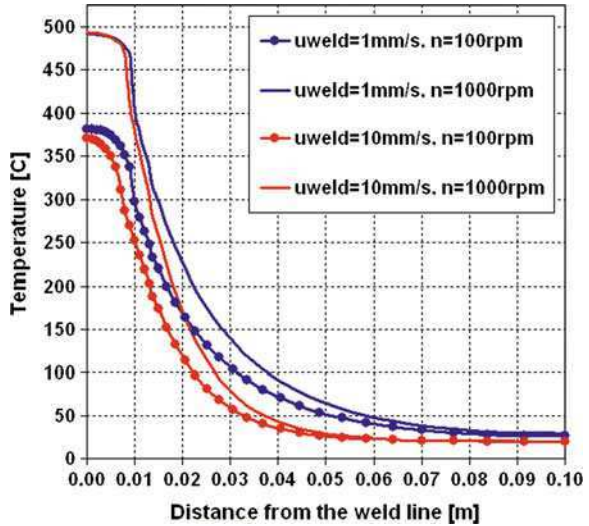
These results show the main characteristics of the applied thermal model:

- (i) A higher welding speed for a fixed rotational speed yields lower temperatures in general, but the decrease in peak temperature right under the tool is very small as it is mainly governed by the temperature dependent yield stress.
- (ii) A higher rotational speed for a fixed welding speed results in substantially higher temperatures in general.
- (iii) The gradients in thermal profiles along the transverse direction of the plate become higher with increasing welding speed, while more uniform and wider thermal profiles are obtained for the lower welding speeds.

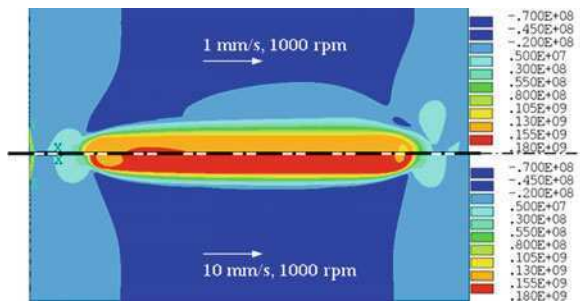
Figure 3.19 shows the contour plots of the resulting longitudinal stress field of the symmetric models (isotherms in 25 MPa) for a chosen welding speed of 1 mm/s



**Fig. 3.18** Temperature profiles for different process variables along transverse direction



**Fig. 3.19** Contour plot of the longitudinal stress field with increments of 22 MPa



and 10 mm/s, respectively, for a rotational speed of 1,000 rpm. A parameter study regarding longitudinal stress profiles along the transverse direction and at the middle of the plate with two welding speeds, i.e., 1 and 10 mm/s, and two rotational speeds, i.e., 100 and 1,000 rpm, is presented in Fig. 3.20.

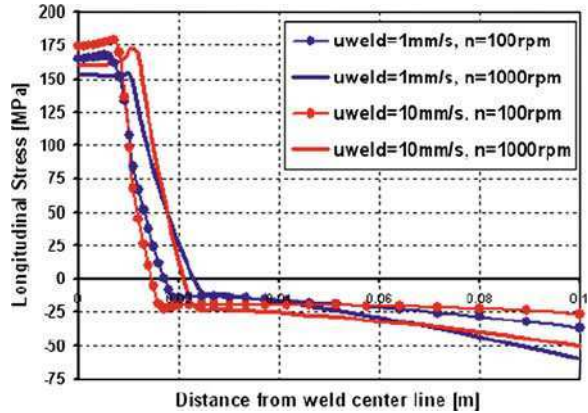
These results show the main characteristics of the applied model for the residual stresses:

- (i) A higher welding speed for a fixed rotational speed results, in general, in slightly higher stress levels in the tension zone.
- (ii) A higher rotational speed for a fixed welding speed yields somewhat lower peak residual stress, however a much wider tension zone leading to a substantially higher residual tensile force.
- (iii) The gradients in residual stress profiles along the transverse direction of the plate become steeper with increasing welding speed.

Some comments should be made regarding these observations. First of all, a microstructure model predicting final hardness/yield strength was not included in



**Fig. 3.20** Residual normal stress in longitudinal direction as a function of distance from weld line



the analysis. If this was done it might influence the trend of the analyses. In particular, the final strength recovery in the middle of the weld is more pronounced for hot weld conditions than for cold weld conditions [89] and this will of course influence the ability to build-up residual stresses.

Another important issue is the choice of hardening model. In the present work, bilinear kinematic hardening was used and this gives a very pronounced displacement of the yield surface towards compression at high temperatures resulting in lower residual stresses when returning to tension upon cooling as compared with e.g., isotropic hardening in which case the yield surface expands due the hardening, thus resulting in considerably higher residual stresses in tension after welding.

Finally some comments on computational time should be given. Depending on the chosen welding speed, for a thermal analysis it is minimum 1 h (for a cold weld, i.e., fast welding speed) while the mechanical analysis, for the same welding speed, is approximately 4 h on an Intel Xeon 3.00 GHz processor using two cores (the limitation is because of the shared memory parallelisation allowed by the ANSYS v11 Academic License).

### 3.4.7 Optimisation Study

#### 3.4.7.1 Case-1

In the following sections a study of the chosen multi-objective optimisation problem in the FSW process is presented [2, 3]. It considers the minimisation of the peak residual stresses in the workpiece together with the maximisation of the production efficiency expressed in terms of traversing welding speed, respectively. These two objectives are conflicting and techniques to deal with this issue are also considered.

The optimisation procedure, which includes process integration of the ANSYS software and the EMO algorithm NSGA-II, is handled by applying modeFRONTIER. The optimisation cycle is initiated by creating an initial population of 16

pre-chosen Design-of-Computational-Experiments (DOCE), for the considered process variables, i.e., the tool rotational speed (revolutions per minute),  $n$ , and the traverse welding speed,  $u_{\text{weld}}$ . The FSW thermal and mechanical simulations, which are built by using APDL in ANSYS, are coupled in a sequential way by execution of parametric input files in a batch mode. The design variables are updated by the optimisation algorithm and are read by the thermal analysis. Then the peak temperature obtained at the end of the welding session is saved in order to be used as a thermal constraint together with the transient temperature field results in order to yield the thermal strains for the mechanical analysis. The mechanical analysis gives as a result the maximum longitudinal stress value at the middle of the plate in the transverse direction, and this is used as an objective to be minimised. This optimisation cycle runs until the stopping criterion, i.e., the total number of generations, is reached.

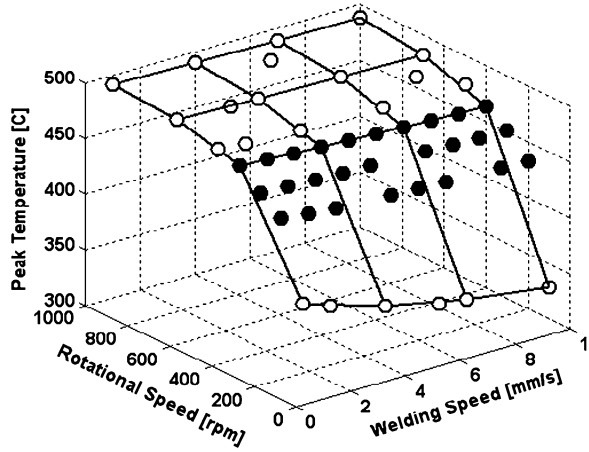
The specific optimisation problem here is stated as the goal of finding the FSW process parameters, i.e., tool rotation speed and traverse welding speed, which provide a set of trade-off solutions for the minimisation of two conflicting objectives. As mentioned earlier, these are the peak residual stresses, which are measured at the middle of the plate along the transverse direction, and the welding time that can also be stated equally as the maximisation of the traverse welding speed. The optimisation problem is constrained by the process-specific thermal constraints, which are given as the upper and the lower bounds on the peak temperatures in the workpiece. The lower bound of 420°C on the peak temperature represents the need for easy traversing of the tool, i.e., to minimise the tool loads along the weld line by contributing to thermal softening of the workpiece material. The upper bound of 480°C is defined in order to consider the tool life and the workpiece properties which are affected by hot welding conditions. This constrained multi-objective problem can then be expressed in mathematical terms as

$$\begin{aligned}
 &\text{Minimise: } f_1(x) = \sigma_{x,\text{max}} \\
 &\text{Maximise: } f_2(x) = u_{\text{weld}} \\
 &\text{subject to: } g_1(x) = 420^\circ\text{C} \leq T_{\text{peak}} \\
 &\quad g_2(x) = T_{\text{peak}} \leq 480^\circ\text{C} \\
 &\quad x = \begin{cases} u_{\text{weld}} = 1, 2, \dots, 10 \text{ mm/s} \\ n = 100, 200, \dots, 1000 \text{ rpm} \end{cases}
 \end{aligned} \tag{3.10}$$

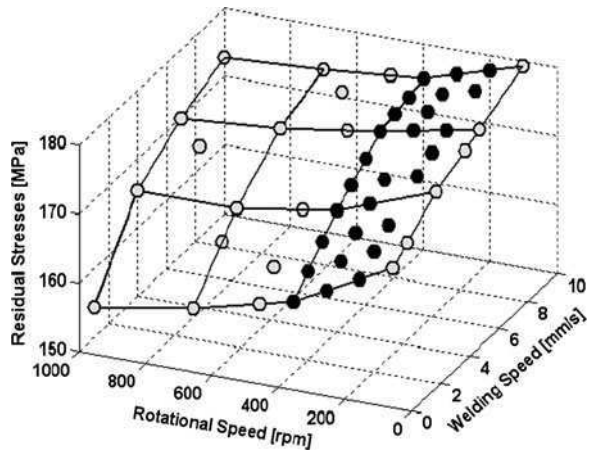
where  $\mathbf{x}$  represents the design variable vector, i.e.,  $u_{\text{weld}}$ , the traverse welding speed that is changing from 1 mm/s to 10 mm/s in 1 mm/s increments and  $n$ , the tool rotational speed which varies from 100 rpm to 1,000 rpm in 100 rpm increments (this results in 10 discrete values in each design variable),  $\sigma_{x,\text{max}}$  defines the peak longitudinal stress, and  $T_{\text{peak}}$  is the peak temperature in the workpiece.

The initial population for the NSGA-II algorithm that is used for both cases is chosen as a modified Full Factorial Design with 4-levels ( $n$ : 100, 400, 700, 1,000 rpm and  $u_{\text{weld}}$ : 1, 4, 7, 10 mm/s) resulting totally in 16 designs.

**Fig. 3.21** Peak temperature versus design variables



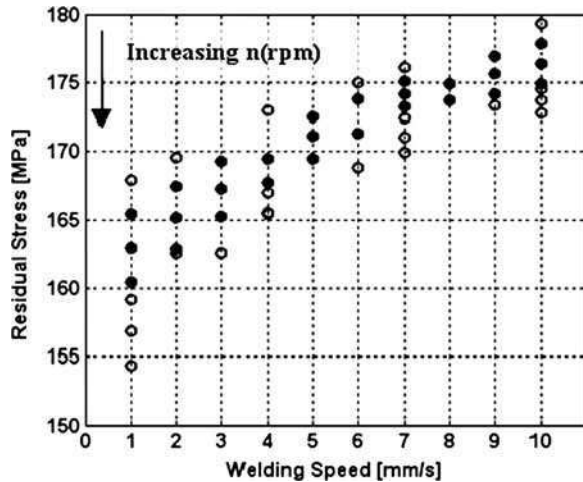
**Fig. 3.22** Residual stress versus design variables



The crossover and the mutation probabilities of 0.9 and 1.0, respectively, are chosen for running a total of 20 generations giving in a total number of 320 solutions.

The solution of the optimisation case which is formulated as in Eq. 3.10, is presented in both the design and the criterion space in the following figures. Some of the designs out of a total of 320 designs are overlapping because of the relatively coarse discretisation of the chosen design variables and also the selection operator. This lies in the nature of the genetic algorithm that implies the survival of some designs without evolution. Figures 3.21 and 3.22 represent feasible and unfeasible designs with dark and light colours respectively; constituting 49 different design points out of 100 (we use a 10 × 10 discretisation). The surface in each figure, i.e., the peak temperature and the peak residual stress, is constructed by 16 DOE points which are evaluated as an initial population for the NSGA-II. It can be clearly seen from these figures that the feasible region, which can also be called the robust process parameter region in this case, is defined by  $n$ -values in the

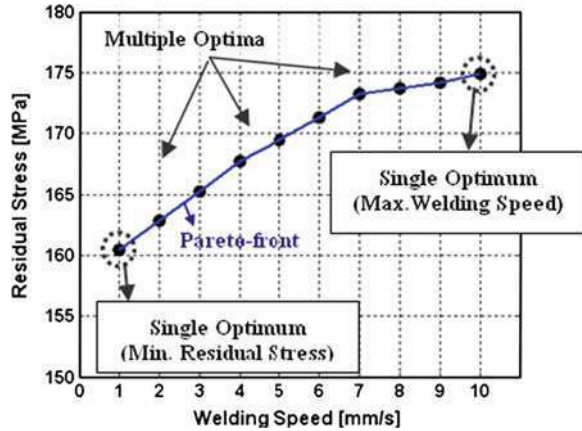
**Fig. 3.23** Objective space of the solution



interval between 200 and 400 rpm. Because of the random evolution strategy of EA, some of the feasible solutions (4, 6, 8 mm/s with 200, 300 and 400 rpm), are missing but the quality of these can be estimated from the surrounding solutions which are positioned on a linear varying region. Although the coarse discretisation results in a few missing solutions, it is advantageous to have an overall idea regarding the optimal feasible process frame with a moderate computational cost. In addition to this, the purpose of this study is to focus on the optimisation methodology to find and discuss alternative trade-off solutions for minimisation of welding residual stresses, not to conclude precise values. Having such solutions provides the engineer or manufacturer practical insight about the relationship among process variables corresponding to the Pareto-optimal solutions.

The objective space that is constructed by minimisation of the peak residual stresses and maximisation of the welding speed is shown in Fig. 3.23. Most of the designs lie close to the Pareto-front, which is shown in Fig. 3.24. This is caused by the relatively low sensitivity of the  $n$  parameter towards the peak residual stresses for a given welding speed. If the minimum-residual stress solution is emphasised for the MOO problem, i.e., choosing weightings of 1.0 for the objective of minimum-residual stress and 0.0 for the objective of maximum-welding speed, the combination of 1 mm/s and 400 rpm would be chosen. If the other extreme solution, i.e., with opposite weightings, is considered, the combination of 10 mm/s and 400 rpm would be preferred. In the case where one is looking for a 70–30% trade-off solution for the same objectives, respectively, it is not clear how to estimate the optimal combination of the process variables. Because of relatively this desired preference in objectives, the solution would be expected to be more similar to the minimum residual stress solution than the maximum welding speed solution. It is important to note that there are some different solutions satisfying such trade-off, but there is only one which is the optimum, i.e., 4 mm/s and 400 rpm in this case. In other words, that solution makes the optimum trade-off,

**Fig. 3.24** Pareto set of the solution



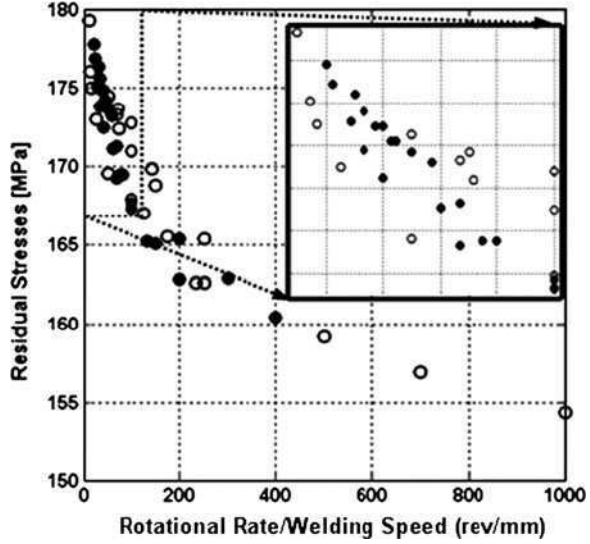
meaning that for a particular amount of sacrifice in one objective, the optimum solution will correspond to the maximum gain in the other objective [24, 60].

All Pareto-optimal solutions correspond to the maximum feasible rotational speed of 400 rpm while having different welding speeds varying from 1 mm/s to 10 mm/s. The Pareto-front shown in Fig. 3.24 gives an idea of ranking the alternative trade-off solutions depending on the available working conditions. If a manufacturer is able to use a standard milling machine instead of an advanced FSW machine and can afford using simple tool designs with low welding speed, he would probably not dare to go from 1 to 7 mm/s in welding speed because the residual stresses yielded per unit increment in welding speed would cost higher comparing to those at higher welding speeds. The amount of sacrifice of the manufacturer relatively depends on the welding speed while one can keep the rotation speed between 200 and 400 rpm.

### 3.4.7.2 Case-2

The specific optimisation problem here is stated as the goal of finding the FSW process parameters, i.e., tool rotation speed and traverse welding speed, which provide a set of trade-off solutions for the minimisation of two conflicting objectives [4]. As mentioned earlier, these are the peak residual stresses, which are measured at the middle of the plate along the transverse direction, and the wear path of an arbitrary point on the tool shoulder which can equally be written as the ratio of the tool rotational speed divided by the traverse welding speed (the mathematical formulation is given below in Eq. 3.11). The optimisation problem is constrained by the process-specific thermal constraints, which are given as the upper and the lower bounds on the peak temperatures in the workpiece. The lower bound of 420°C on the peak temperature represents the need for easy traversing of the tool, i.e., to minimise the tool loads along the weld line by contributing to thermal softening of the workpiece material. The upper bound of

Fig. 3.25 Objective space of the solution



480°C is defined in order to consider the tool life and the workpiece properties which are affected by hot welding conditions.

The wear path at the radius  $r$  at any point on the tool/workpiece interface can be approximated by the following expression,

$$L_{\text{path}} = v_{\text{circumf}} t_{\text{circumf}} = \omega r t_{\text{weld}} = \omega r \frac{L_{\text{weld}}}{u_{\text{weld}}} = \frac{(rL_{\text{weld}})\omega}{u_{\text{weld}}} = C \frac{\omega}{u_{\text{weld}}} \quad (3.11)$$

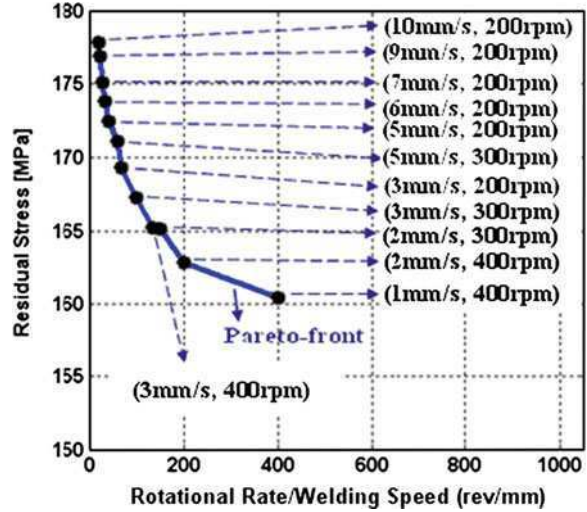
where  $C$  is a parameter that will remain constant for the point under consideration. Minimisation of the ratio of the rotational speed/welding speed corresponds to maximising the tool advance per revolution. The constrained multi-objective optimisation problem can then be expressed in mathematical terms as,

$$\begin{aligned} \text{Minimize: } & f_1(x) = \sigma_{x,\text{max}} \\ \text{Minimize: } & f_2(x) = \omega/u_{\text{weld}} \\ \text{subject to: } & g_1(x) = 420^\circ\text{C} \leq T_{\text{peak}} \\ & g_2(x) = T_{\text{peak}} \leq 480^\circ\text{C} \end{aligned} \quad (3.12)$$

The objective space that is constructed by minimisation of both the peak residual stresses and the ratio of the rotational speed divided by the traverse welding speed is shown in Fig. 3.25. Most of the designs lie close to the non-dominated-front, which is shown in Fig. 3.26. This is because of the relatively low sensitivity of the  $n$  parameter towards the peak residual stresses for a given welding speed.

The Pareto-front shown in Fig. 3.26 gives an idea of ranking the alternative trade-off solutions depending on the available working conditions. Two extreme sets of solutions are obtained as (1 mm/s, 400 rpm) and (10 mm/s, 200 rpm) for

**Fig. 3.26** Non-dominated set of the solution



the minimum residual stress and the minimum wear path or in other words the ratio of the rotational rate divided by welding speed, respectively. The other non-dominated solutions between these extrema are varying between 200 rpm and 400 rpm for different welding speeds, while most of these solutions have a rotational speed of 200 rpm. If a manufacturer is able to use a standard milling machine instead of an advanced FSW machine, such as the ESAB SuperStir<sup>TM</sup> machine [90] or RoboStir Robotic FSW Machine [91], and can afford using simple tool designs (flat tools instead of threaded ones) with low welding speed, he would probably prefer to use a welding speed of 1 or 2 mm/s with relatively high rotational speed, i.e., 400 rpm, because the residual stresses yielded per unit increment in the ratio of rotational speed/welding speed would be lower comparing to those at higher welding speeds with lower rotational rate (lower values of the wear path criterion). For instance having a machine which enables us to weld with 2 mm/s speed and 400 rpm would be sufficient. On the other hand, if a manufacturer who is able to afford higher residual stresses and lower wear path criterion values, in other words lower heat exposure, it will be relatively easier to sacrifice in quality, thereby a welding speed of 7–9 mm/s with the minimum rotational speed of 200 rpm will be preferable.

### 3.5 Metal Casting

Casting is a manufacturing process in which molten metal flows by gravity or other forces into a mould, containing a hollow cavity of the desired shape, where it solidifies in the shape of the mould cavity, see Fig. 3.27. The solidified part is also known as a casting, which is ejected or broken out of the mould to complete the



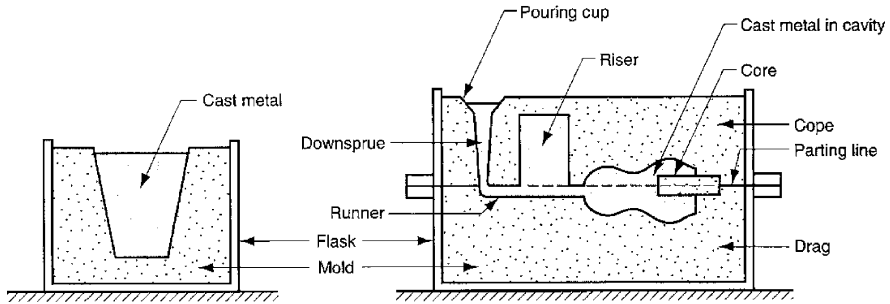


Fig. 3.27 Schematic view of a casting process [93]

process [92, 93]. The casting process is mainly used to produce ingots, i.e., large cast parts which are simple in shape and intended for subsequent reshaping by processes such as rolling or forging, and cast parts having more complex geometries that are much closer to the final desired shape of the product (i.e., shape casting). Different variants of shape casting techniques are available in industry, thus making it one of the most flexible and attractive of all manufacturing processes. Besides, it allows manufacturers to produce products having both external and internal geometrical complexities, and moreover some casting processes are capable of producing parts having no need for further manufacturing operations to satisfy required tolerances and are suitable for mass production as well as producing very large parts weighing more than a hundred tons. There are also some varying disadvantages associated with different casting processes which include limitations on mechanical properties, porosity, poor dimensional accuracy and surface finish for some casting processes, safety hazards to humans when processing hot molten metal, and environmental problems [93].

The physical description of the metal casting process in a virtual environment, i.e., via numerical models, demands the quantification of process parameters and process steps as they directly impact the casting quality. The idea of utilising numerical models to predict the filling and solidification of castings, instead of intuition and trial-and-error based incremental procedures came from physicists, mathematicians, and mechanical engineers. Today, it is well established by foundry engineers and management as well as customers that casting simulation tools provide more than just a look into a black box. The heat transfer simulations, i.e., solidification process in particular, paved the initial steps to the fundamental research for understanding the process. Mould filling is a complementary and complex step of the casting simulation. This is not only important for the gating layout which is also designed to support the feeding effectively, but for the detection of filling related defects as well, e.g., a potential premature solidification. Indeed, the inhomogeneous temperature distribution in the melt during the filling process, which mostly represents highly turbulent flow (as understood from the rheological properties of metal melts), has in many cases an impact on the solidification process. This is still the case even if the melt surface appears to be



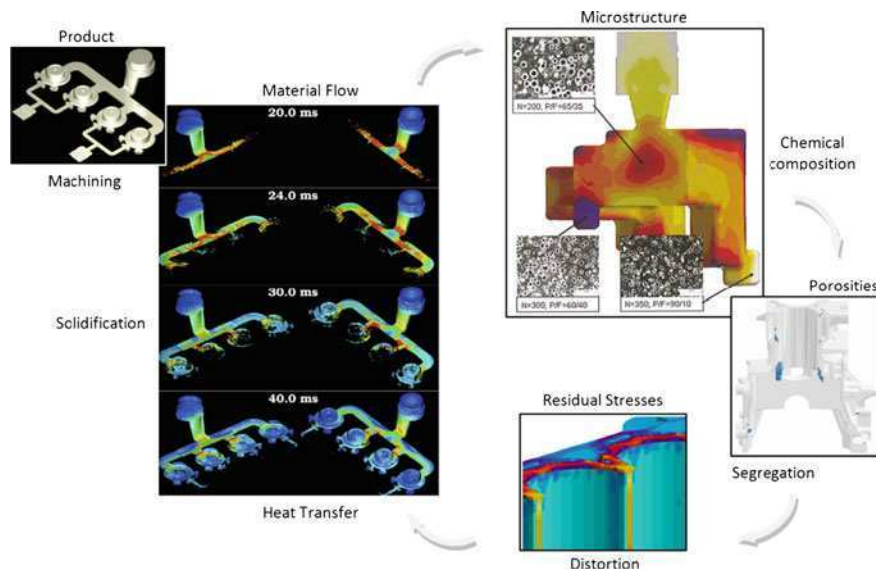


Fig. 3.28 Multi-physics modelling of the casting process [94, 99]

rising quietly. Many casting defects originate from these under-surface movements, as well as reactions between melt and mould material. These defects include mould defects, air entrainments, oxidation defects, slag entrainments or metallurgical challenges which have strong effects on the feeding behaviour [94]. The local shrinking and expansion behaviour of a casting can only be predicted under the consideration of the locally developing phases (graphite, austenite, cementite). The evolution of each phase should therefore be considered together with the alloying elements and the inoculation, i.e., microstructure modelling, throughout the entire solidification analysis prior to the residual stress and distortion analysis or prediction of hot tears. A schematic representation of some of the simulation steps for the casting process is presented in Fig. 3.28. Readers who are interested in further details about the historical development of casting simulation and future challenges in this area are encouraged to refer to the following references [94–98].

Casting process modelling in essence involves the simulation of mould filling and solidification of the cast metal as well as the solid state cooling [42]. At the macroscopic scale, these processes are governed by basic equations which describe the conservation of mass, momentum, and energy. Heat transfer is perhaps the single most important discipline in casting simulation. The solidification process depends on heat transfer from the part to the mould and from the mould to the environment. Solidification modelling involves the application of the heat transfer concepts along with techniques to account for the release of latent heat during solidification [95]. The mould and any other solid materials (chill, insulation, feeder, etc.) are modelled using the standard heat conduction

equation given in Sect. 3.3, Eq. 3.1. The extent of solidification at any location within the casting is represented by the fraction of solid  $f_s$ . At temperatures greater than or equal to the *liquidus* temperature, the cast metal is in a completely liquid state with a solid fraction of zero. As the latent heat is removed, the fraction of solid increases and reaches a value of unity when the metal is in a completely solid state (i.e., at *solidus* temperature). The region where the solid fraction is between zero and unity is referred to as the *mushy zone*. There are two approaches to determine the solid fraction value: (1) *Solid Fraction-Temperature Equilibrium*, where the solid fraction is assumed to be a known function of temperature (i.e., temperature dependent property of the metal). (2) *Solidification Kinetics*, which permits the accurate prediction of phenomenon such as undercooling. It considers the evolution of the solid fraction in time depending on several parameters and it requires detailed metallurgical data which is not easy to access.

The release of latent heat during solidification can be accounted for by a volumetric heat generation term in the heat conduction equation (i.e.,  $q_{\text{vol}}$  in Eq. 3.1) as shown below,

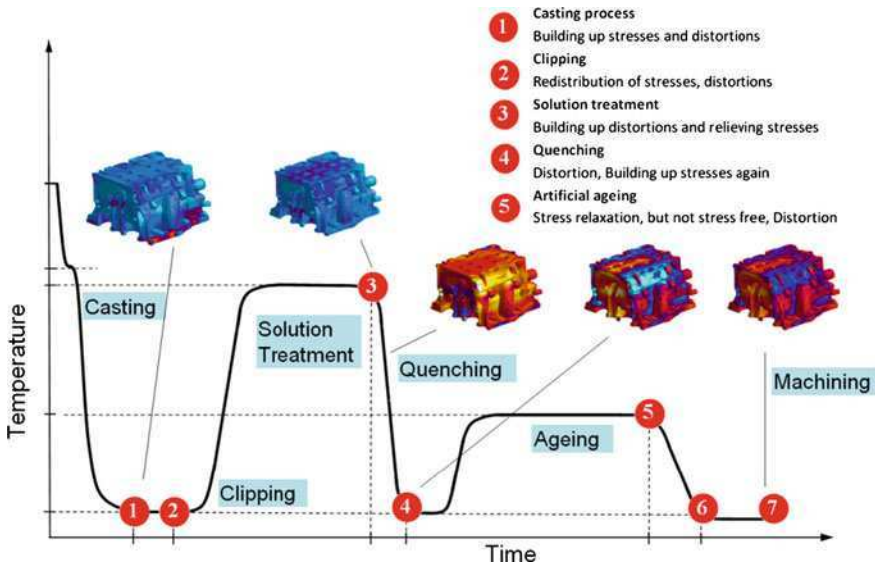
$$q_{\text{vol}} = \rho \Delta H_f \frac{\partial f_s}{\partial t} \quad (3.13)$$

where  $\Delta H_f$  represents the latent heat of solidification. Equation 3.10 assumes that the latent heat varies in proportion to the solid fraction, which is a reasonable approximation for casting process modelling [95].

The most accurate method of determining the initial conditions is to perform a mould filling simulation. The temperature distribution at the end of the mould filling simulation will then serve as the initial temperature distribution for the solidification simulation. However, because of the computational demand, the mould filling simulation is frequently omitted and some reasonable initial temperature values are used. For the cast metal, the initial temperature usually is set somewhere between the liquidus temperature and pouring temperature, whereas the initial mould temperature depends on the type of casting (e.g., for sand castings, the initial mould temperature will most likely correspond to the ambient temperature).

Another modelling issue is the contact condition at the interface between the cast and the mould (not in perfect contact) which affects the heat transfer having a discontinuity in temperature. If the unknown characteristics of the gap such as the thickness variation with time (caused by shrinkage or distortion in the casting) and the properties of the gas within the gap were known, then the heat transfer across the gap could be computed directly, i.e., via defining heat convection boundary condition assuming a heat transfer coefficient that is empirically determined [42, 95, 96].

Mathematical modelling of stress/strain phenomena in casting processes is a complex subject, which in the general case among other phenomena involves a coupled 3-dimensional thermo-mechanical analysis including solidification and other phase transformations, shrinkage-dependent interfacial heat transfer caused by relative motion between casting and mould as mentioned above, mould

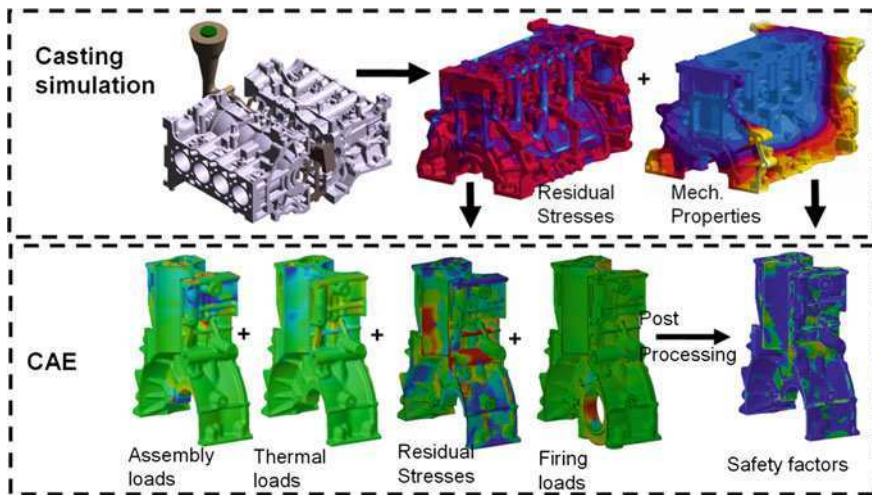


**Fig. 3.29** Stress history in a cylindrical head over the entire manufacturing process (courtesy of MAGMA GmbH) [94, 100]

distortion, temperature and time-dependent plasticity, hot tears, hydrostatic pressure from the liquid and crack formation. All of these phenomena are obviously not equally important and as for other branches of numerical process modelling, one of the keys to a successful stress/strain analysis of a casting process is to take into account only what matters for the solution of the problem at hand [42].

Making castings today requires more than just pouring liquid metal into a mould; it is actually only one part of an integrated chain of processes. Most castings receive their final properties through processes after the casting process, i.e., heat treatment or machining, as illustrated in Fig. 3.29. Therefore it is crucial to be able to predict the performance of the cast part, e.g., final mechanical properties, in a reliable process window under a set of uncertainties [94]. In a similar way, the integrated modelling approach combining the thermo-mechanical simulation of the casting and in-service load performance supplies a substantial efficiency for engineering companies to make modifications in the process or the product design to stay competitive. Figure 3.30 represents an integrated engine development cycle for improving the robustness under considerable loads [94].

As known from previous studies which have also been mentioned in the beginning of this section [14, 15, 94], the casting design, in particular the gating and riser system design has a direct influence on the quality of cast components. Changes in one process parameter impact many casting quality-defining features during the process, i.e., a change of the pouring temperature does not only change the solidification behaviour, it also changes the fluidity of the melt, which can lead to a misrun. The metallurgy of the melt might be impacted, which could lead to



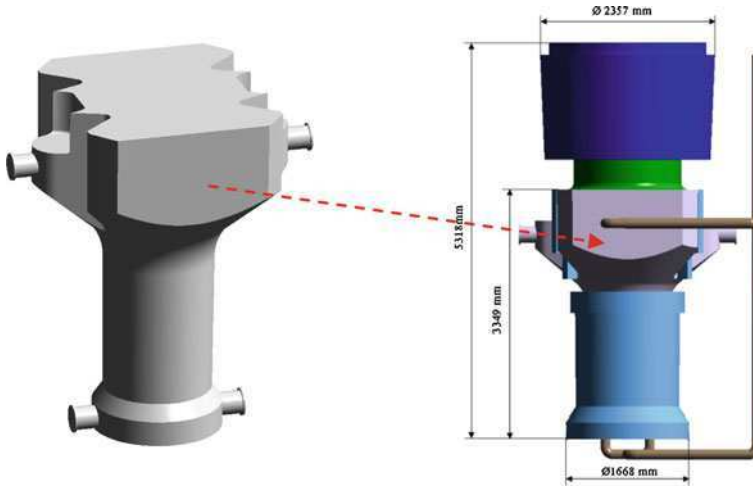
**Fig. 3.30** An integrated engine development cycle for improving the robustness under considerable loads (courtesy of MAGMA GmbH) [94]

changes in the temperature balance of the mould or die, which again can lead to problems with overheating or erosion [94, 98]. Thus, it is almost impossible for a human-being to analyse the different effects of all these nonlinear interactions between process parameters through several process steps. However, most of the design activities were, or still up to some extent are, based on past experience and empirical rules [94]. Despite high computational expenses, the autonomous integration of casting process simulations with numerical optimisation methodologies has recently been initiated and it seems to be getting more attractive for researchers and manufacturers [5, 14–18, 94]. In the next section, a recent case study on a real world multi-objective optimisation application for casting of a forging ram will be given in more detail which considers several manual (intuitive) and autonomous design iterations to improve the casting yield.

### 3.5.1 A Casting Yield Optimisation Case Study

This section presents a multi-objective optimisation case study in the gravity sand casting process of a forging ram in which top riser volume and shrinkage porosity are minimised, subjected to a constraint on centreline porosity, via searching for the optimum set of design variables, i.e., dimensions of the riser and the chills. Readers are encouraged to find further details in the original work [5].

The iterative product design cycle includes five different layouts of a steel forging ram (as see Fig. 3.31) manufactured by a gravity sand casting process. The initial layout is obtained from a foundry, Vitkovice Heavy Machinery a.s.,



**Figs. 3.31 and 3.32** (3.31) 3-D view of the cast part used in the project, (courtesy of Vitkovice Heavy Machinery, a.s.) [5]. (3.32) Initial casting layout. The riser is indicated by *green*, chills indicated by *light blue*; insulation is denoted *dark blue*, (courtesy of Vitkovice Heavy Machinery, a.s.) [5]

which manufactures the forging ram. The second layout with manually rearranged gating system and chills is also provided by the foundry, and last three layouts are generated by using numerical optimisation. The first two designs are analysed both in terms of filling and solidification using MAGMASOFT (the commercial software dedicated to the numerical simulation of the casting process), and then the results are compared with experimental casting trials, no numerical optimisation are involved at this stage. The last three designs are optimised numerically using MOGA (Multi-Objective Genetic Algorithm [16]) implemented in MAGMAfrontier [16–18] (the numerical optimisation module integrated into MAGMASOFT) and are assessed only in terms of solidification as the filling pattern remains unchanged; however the temperature fields at the beginning of the solidification are inherited from the filling stage.

A riser is designed and placed on top of the heaviest section based on the thermal analysis of the part itself. In order to enhance the feeding ability of the riser, insulation is applied (see the dark blue section in Fig. 3.32). The main cylindrical padding is insulated and the melt surface is covered by an exothermic powder as well as the additional insulating powder is applied. Next, it is determined that the part will be bottom-filled using a gating system comprised of the refractory tiles of which the cross-sectional areas are constant over the entire gating system. Last, the chills (the light blue parts in Fig. 3.32) are added around the cylindrical section of the casting to establish directional solidification and to push the macro segregation-related flaws from the surface further inside the casting. The original casting layout is simulated using the casting conditions and

**Table 3.4** Material settings in MAGMAsoft [5]

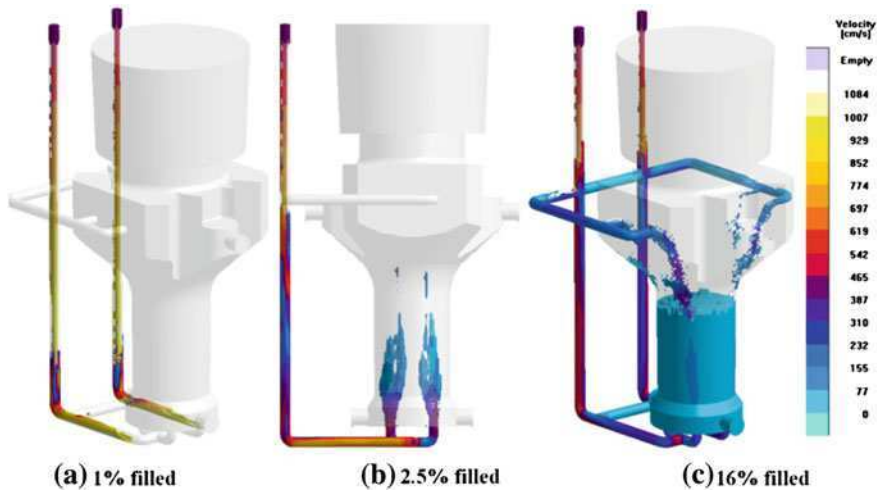
Material of the casting	GS20Mn5 (DIN 1.1120)
Material of the mould	Furan sand
Material of chills	Common steel
Material of the insulating padding	IN5
Material of the exothermic powder	Ferrox
Material of the insulating powder	Vermikulit
Initial (pouring) temperature of the casting	1,540°C (2,804 F)
Initial temperature of the mould	20°C (68 F)
Filling time	120 s
Feeding efficacy of the applied steel alloy	40%
Sand permeability	Activated-value taken from a standard database
Weight of the casting- incl. risers and gating system	59,596 kg (131,111.2 lbs)
Weight of the casting itself (fettled)	29,898 kg (65,775.6 lbs)

parameters listed in Table 3.4. Both filling and solidification analyses are conducted. A stress analysis is not considered in any of the presented cases.

The heat transfer coefficients (HTC) at the casting/mould interface are assumed to be constant ( $800 \text{ W/m}^2\text{K}$ ), i.e., temperature independent. This assumption holds only in the case of gravity sand casting. The reasoning is that in sand casting the contact between the melt and the mould is poor from the very beginning because of the rough surface of the mould. As a result, there is a high resistance to heat removal, giving low interface HTCs. When the casting shrinks during solidification and solid state cooling, an air gap is formed in the casting/mould interface, inducing even more resistance to the heat removal. Nevertheless, since the heat transfer has been poor from the very beginning the decrease in HTC caused by volumetric changes is not that determining for the results and the HTC can be assumed more or less constant (low) over the entire casting process. Moreover, it is not primarily the interface that induces the largest resistance to heat transfer; it is the large sand mould and its poor thermal properties that really govern the heat removal.

Figure 3.33 shows three intermediate stages of the filling process, i.e., 1%, 2.5% and 16% filled, respectively, which is supposed to indicate whether the current gating system will provide a uniform filling without any melt aspiration in the downsprue or surface turbulence that would likely lead to excessive oxidation of the propagating melt thus causing various filling-related defects, i.e., re-oxidation inclusions, entrapped air pockets, etc., [101]. The primary source of oxygen in re-oxidation inclusion formation is air, which contacts the metal stream during pouring as well as the metal free surface in the mould cavity during filling. It can be seen in Fig. 3.33a that because of a constant cross-sectional area over the entire downsprue, the melt starts to spire from the mould walls and gets oxidised. This phenomenon can actually be explained simply via the continuity equation. The melt experiences a free fall from the nozzle of the pouring ladle down to the



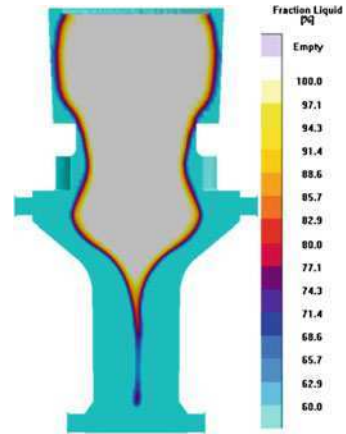


**Fig. 3.33** Three different stages of filling process of the bottom-filled forging ram [5]

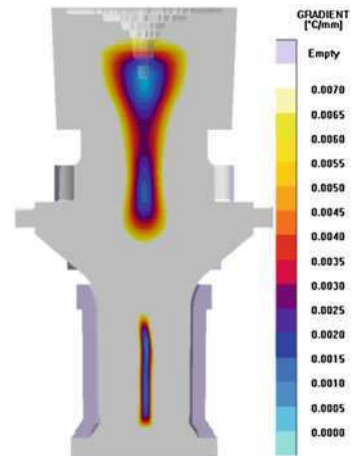
bottom of the gating system. During the free fall it accelerates because of the effect of the gravity force and changes its area (the area decreases with the increasing velocity). In order to compensate for the area reduction resulting in the aspiration from the mould walls, one has to decrease the area of the downsprue accordingly. The nearly ideal solution would be an application of a streamlined gating system [102]. However, for such a large cast part, this solution is infeasible. The streamlined gating system is used mainly in the gravity die casting. Another option would be the use of “choke” conical elements at several locations in the downsprue. Figure 3.33b shows that due to no velocity control during the early stage of the filling process the melt reaches the mould cavity with a quite high velocity (approx. 5 m/s). A very rapid entrance naturally leads to a formation of fountains (1.47 m high) inside the mould cavity. When the melt starts to fall down again, it splashes, becoming highly turbulent and disintegrated. In most of the bottom-filled casting assemblies it is a difficult task to fully avoid this formation. Nevertheless, it should always be the primary objective of a designer to design such a gating system with all necessary attributes to keep this phenomenon at a minimum. Finally, Fig. 3.33c indicates the melt falling down from the two top runners on top of the melt front progressing from the bottom of the mould cavity. This is again a feature that should be kept at a minimum during filling for the same reason as the fountains, i.e., oxidation. Moreover, there is an oxide layer already present on top of the melt front coming from the bottom which very likely gets torn apart by the melt streams and a surface turbulence and disintegration of the melt front is thus established.

The thermal analysis during solidification evaluates the efficacy of the top riser and the cooling effect of the chills placed around the cylindrical section of the cast part. In Fig. 3.34, one can see that during the solidification process, particularly at

**Fig. 3.34** Fraction liquid criterion function indicating an isolated liquid pool in the bottom section of the cast part at 42% solidified [5]



**Fig. 3.35** Gradient criterion function depicting a very shallow gradient in various areas of the casting [5]

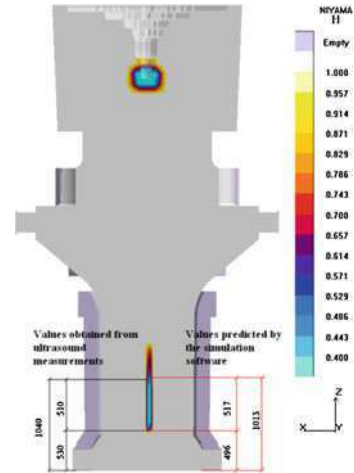


42% solidified, there are indications of isolated liquid pools in the lower section of the casting. This fact raises a probability of porosity formation attributable to a lack of liquid feeding. The chills cool the cylindrical section too rapidly, when compared with the very bottom area in which the cooling rate is lower because of the enlarged cross-section, thus creating a hot spot. A potential remedy might be either to increase the thickness of the chills towards the bottom area or to redesign the gating system so that there is a chance to add a chill plate underneath the casting bottom. Cooling of the bottom of the casting would be significantly promoted and directional solidification towards the riser would be established.

A direct consequence of having both isolated liquid pools (see Fig. 3.34) and very flat temperature gradients as seen in Fig. 3.35 in the casting domain is the presence of a shrink (porous area). Areas solidifying early always “suck out” the liquid melt from areas solidifying last to compensate for the volumetric changes evoked by the solidification process. As long as there is an open and



**Fig. 3.36** Prediction of the centreline macro/micro shrinkage and its experimental validation obtained from the foundry [5]



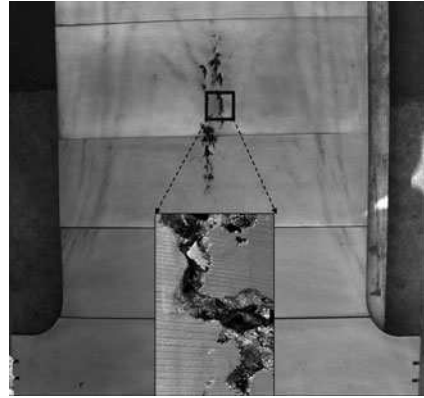
active feeding path to these areas no problem occurs. However, when the liquid melt supply is cut off and drained, areas solidifying last will be short of melt and so when the time comes for them to solidify no compensation for the volumetric shrinkage will be available, giving rise to porosity. This issue can be addressed by means of the Niyama criterion function [103–105], that is a local thermal parameter defined as the relationship between the gradient ( $G$ ) in K/mm and the cooling rate ( $R$ ) in K/s, both of which are assessed at a specified temperature near the end of solidification, when the solidification shrinkage forms, see Eq. 3.14. In the present study, the Niyama criterion is evaluated at a temperature 10% of the solidification range above the solidus temperature. This is important to state, as the choice of Niyama evaluation temperature can remarkably affect the resulting Niyama values [106].

$$\text{Niyama} = \frac{G}{\sqrt{R}} \quad (3.14)$$

With the help of the Niyama criterion, it is feasible to predict the presence of centreline shrinkage porosity, i.e., micro- and macro-shrinkage in steels, caused by shallow temperature gradients [107, 108]. It indicates that in regions that solidify quickly, there must be hot metal nearby to establish a high gradient to feed the shrinkage during solidification. It has been proven by numerous trials that for sufficiently large Niyama values, no shrinkage porosity forms. When the Niyama value decreases below a critical value, small amounts of micro-shrinkage begin to form. As the Niyama value decreases further, the amount of micro-shrinkage increases until it becomes detectable on a standard radiograph. This transition occurs at a second critical value. Both of these threshold values are heavily dependent on the composition of the alloy and in some cases on the casting process conditions.

Figure 3.36 shows the numerically predicted presence of centreline porosity in the lower areas together with results obtained from the casting trials

**Fig. 3.37** Results from the casting trial—presence of porous areas and bands of macro-segregation, (courtesy of Vitkovice Heavy Machinery a.s.) [5]

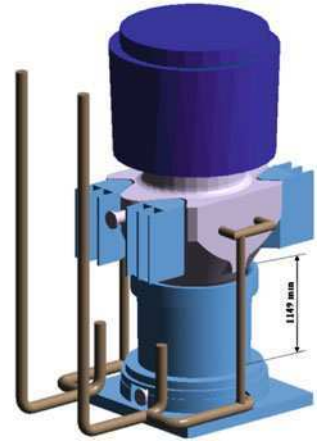


(by the radiographic technique). It is seen that indeed, the porous areas occurred in the casting where the isolated pools of liquid were once present. Looking at the dimensions of the defect area, one can see a very good agreement between the two types of results (numerical-red (right) versus experimental-black (left)). The close correlation also justifies the use of the Niyama threshold value of 0.45 for this particular steel alloy. However, it should be emphasised that the geometrical extension of the shrinkage obtained numerically, to some extent is approximate because of its dependency on the mesh quality.

Figure 3.37 shows the results obtained from the casting trials. The cast part was cut into several sections and the porous area (also predicted by the simulation) was detected and measured. Besides the macro-shrinkage both V- and A-type macro-segregation bands are spotted in the casting. A general cause of the macro-segregation is relative movement or flow of segregated liquid and solid during solidification. The most common form of solid movement is the settling or floating up of small solid pieces formed early in the solidification process. These solid pieces may be dendrite fragments that separated from an existing solid structure or equiaxed grains that nucleated in the bulk liquid. They settle or float, depending on their density relative to the liquid. The solid pieces generally have a composition different from the nominal alloy composition, and their movement to different parts of the casting thus induces macro-segregation [109–111].

Next, the new casting arrangement with manually redesigned gating system and rearranged chills is developed. It may be seen in Fig. 3.38 that the shape of the forging ram is somewhat different when compared with the initial one in Fig. 3.32. It is attributable to the fact that the second design is manufactured for a different customer who requested these geometrical adjustments, i.e., larger diameter of the cylindrical section and geometrical adjustments in the ram's head. The aim is to investigate the effects of a cooling plate located underneath the casting. In order to place the plate, the gating system has to be changed from bottom filling to side filling. Also, new vertical runners are added to support filling in higher sections of the casting. Furthermore, the chills are rearranged a bit. In order to ensure sufficient cooling of the very bottom, thicker chills are added around the conical

**Fig. 3.38** Manually optimised casting design [5]

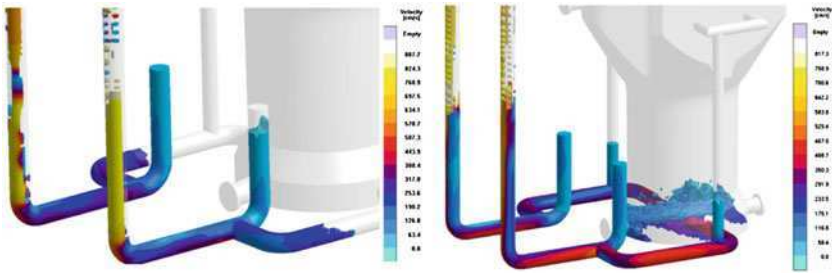


bottom section of the casting. Furthermore, new chills are incorporated into the top head section of the casting. Besides the aforementioned geometrical changes, all casting and simulation parameters remain the same as in the first casting arrangement.

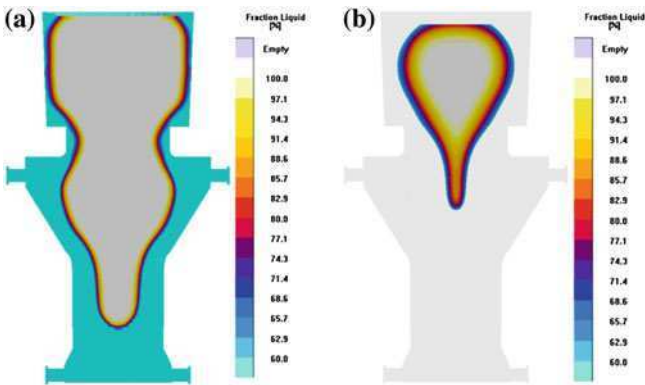
After the initial assessment of various stages of the manufacturing process, potential drawbacks and defects were recognised. This was then verified by casting trials creating a solid ground for subsequent improvements and optimisation. As centreline porosity depends primarily on thermal gradients and the cooling rate, the next step was to induce steeper thermal gradients in the section surrounded by the chills and to establish a directional solidification towards the heaviest top section where the riser is placed to keep the feeding path open as long as it is necessary. This was pursued by the rearrangement of the chills and by adding a chill plate underneath the casting bottom. Because of the chill plate, the bottom filling was no longer feasible. Therefore, the gating system was somewhat redesigned, which after the first filling analysis was strongly recommended anyhow. The reason for adding the vertical extensions of the two horizontal runners is to reduce the kinetic energy of the melt. This is an easy way to slow down the melt front. However, in many foundries world-wide it has been overlooked and not applied.

Figure 3.39 shows early stages of the filling process. One can argue that the new gating system really improved the filling pattern. The two vertical extension channels significantly slowed down the melt (approximately 4 m/s in the runners as compared with almost 10 m/s in the original layout) and thus it propagates uniformly towards the thin gates. When the melt reaches the cavity it does not form fountains but creates a relatively small splash during impingement of the streams. A solution to this problem might be an application of tangentially oriented thin gates which then help to avoid any impingement of propagating melt fronts.

However, as the shape of the downsprue remained untouched, melt aspiration is still seen at that area. This issue should really be addressed by the manufacturing foundry to eliminate oxidation of the melt in the downsprue area. It was expected



**Fig. 3.39** New filling pattern arising out of the redesigned gating system [5]

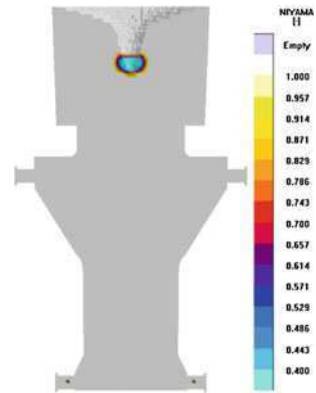


**Fig. 3.40** **a** Improved solidification pattern arising out of the reworked system of the chills- 42% solid [5]. **b** Fraction liquid at 75% solidified [5]

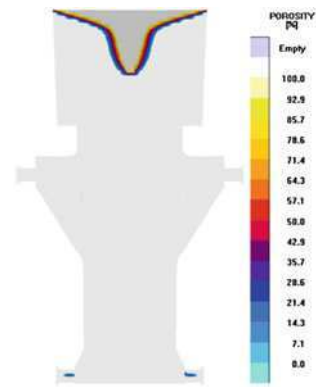
that the new system of chills will change the solidification pattern of the casting and will have positive effects on the formation and distribution of the defects. The results are captured in Figs. 3.40, 3.41 and 3.42. In Fig. 3.40a and b, by means of the fraction liquid function a new solidification pattern was predicted at 42% and 75% solidified.

The size of the riser together with the new arrangement of the chills obviously positively affected solidification, completely avoiding the isolation of the liquid pools seen in Fig. 3.34. Moreover, the steel plate placed below the casting evoked rapid cooling and facilitated directional solidification towards the riser. The comparison of centreline porosity indicates that the increased temperature gradients via enhanced chilling readily eliminated likelihood of its formation see Fig. 3.41. The only two areas that might be of a concern are the two bottom pins. In the original layout no porosity was present in these areas however this is not true for the current layout. This is depicted in Figs. 3.41 and 3.42. Considering that they are used for transportation purposes and the entire casting weighs around 58 tons, this porosity cannot be neglected as it corrupts the mechanical properties and weakens those areas. This issue can be eliminated by adding a sufficient draft to

**Fig. 3.41** Prediction of centreline porosity [5]



**Fig. 3.42** Distribution of macroscopic shrinkage [5]



the pins and changing their solidification pattern. The critical value for porosity in Fig. 3.42 has been set to 1%. Areas containing less than 1% porosity are considered “healthy” and thus they are filtered out by the X-Ray function.

After inserting all datasets and parameters into the standard simulation environment, the manually refined casting layout in Fig. 3.38 is assessed to create a reference solution to compare the optimisation results with. It has been decided to try to reduce the size of the top riser as much as possible to increase the casting yield, providing that there will be no defects occurring in the casting because of the riser’s reduced size. When the riser and the chills are transformed into parametric objects, the optimisation process is initiated. The design variables (those which are subjected to optimisation) are: dimensions of the chills (height and thickness of the bottom cylindrical chills), dimensions of the top riser (height, bottom and top diameters), and the top diameter of the riser neck, see Table 3.5.

The main optimisation objectives are (i) to design the top riser so that the casting is sound (i.e., with minimum shrinkage and centreline porosity), and (ii) at the same time having a top riser’s volume as small as possible to increase the casting yield. In this context, ‘casting yield’ is defined as the gross weight

**Table 3.5** Design variables for optimisation [5]

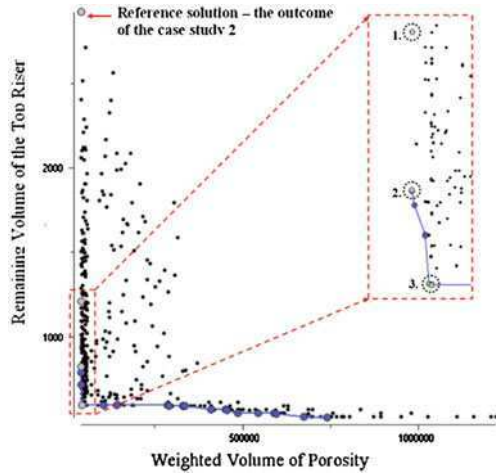
Design variable	Lower limit (mm)	Upper limit (mm)	Step (mm)
Cylindrical chill- height	549	1,149	50
Cylindrical chill- thickness	100	200	10
Riserneck- top diameter	1,200	1,300	10
Riser- bottom diameter	1,260	1,660	20
Riser- top diameter	(1,260 × 1.06)	(1,660 × 1.06)	20
Riser- height	500	1,350	50

including the riser and the gating system divided by the weight of the fettled casting. Based on the number of design variables and their ranges of variation, the optimiser generates the initial population which is in size of 100 and based on the Sobol algorithm providing a quasi-random distribution [112]. An elitist EMO algorithm MOGA is used with the directional crossover probability of 0.6, the selection probability of 0.3 and the mutation probability of 0.1. Moreover, constraints are handled with the penalisation methodology [24].

After the two preceding analyses one could think that a reasonably good solution has been found. The gating system has been improved; the solidification pattern changed and fewer defects are present. This was also confirmed by the manufacturing foundry which is using this improved design at the present state. Thus it is a favourable state for a subsequent geometry optimisation. Both of the previous designs weighted approximately 60 tons together with the riser and the gating system. From Figs. 3.41 and 3.42, it is seen that the major shrinkage pipe in the riser is still too far from the actual casting body to be critical therefore, there is room for a volume reduction to obtain an increased casting yield. From now on, only the solidification results will be discussed since the gating system remained unchanged, thus the filling pattern does not change.

The objective space for the optimisation problem in Fig. 3.43 is constructed by the two following objectives: minimisation of shrinkage porosity and minimisation of the remaining volume of the top riser. The first objective is represented by the Weighted Volume Porosity which stands for the total volume of areas having issues with porosity. The remaining volume of the riser is then calculated as the geometrical volume of the riser minus the volume of the shrinkage pipe in the riser. There are several features in that figure that should be addressed. The blue line is the Pareto set which comprise the non-dominated solutions, although, it is up to the user to determine which solution out of the Pareto set will be the most desirable. In other words, the decision-maker has to figure out whether she wants to minimise the riser as much as possible, at the cost of increased porosity or to have a porosity-free casting with a slightly larger riser. In this case, three distinct designs were selected. The first one, marked as 1 in Fig. 3.43 does not lie on the Pareto set and represents the most modest solution i.e., the largest riser volume. It should be emphasised that although it may not be clear from the figure, solution 1 is dominated by solution 2. The second one marked as 2 resembles a single optimum case—the lowest amount of porosity, and the third one marked as 3 stands for a trade-off solution.

**Fig. 3.43** Design space with the highlighted Pareto set [5]



Then, the three designs have been analysed in the standard simulation environment. In order to obtain realistic temperature fields during solidification, filling has also been considered in the simulation, but its results are not shown here. The reason for choosing such designs was the following: the primary aim has been to keep the level of porosity very low, possibly not above the value of the original design but still increasing the casting yield. That is why the focus was put on the solutions very close to the Y-axis. Moreover, the manufacturing foundry wanted to see different layouts—from “modest” to those “on the edge” to make a better comparison and decision as to which solution to select for the subsequent production. From the optimisation perspective it is given that the best solutions constitute the Pareto line, so why should we pick a design not on the Pareto line that is solution 1? Many foundries will rather prefer a very safe solution to compensate for potential flaws during production, e.g., human factors, deviations from alloy compositions, etc., Solution 1 was selected for the subsequent analysis because it has a large enough riser to keep porosity far from the casting and still has its total volume remarkably smaller than the original layout. Figure 3.44 depicts the three selected designs. Information regarding dimensions of the optimised designs is listed in Table 3.6.

Regarding the solidification patterns of the three optimised designs, no isolated liquid areas are forming in the bottom area as in the original layout, depicted in Fig. 3.34. The bottom chill plate and the stair-type chill around the conical section induced directional solidification towards the thermal axis and the riser. Solidification patterns were checked over the entire solidification interval. It was found that none of these designs exhibit apparent isolated liquid areas although solution 3 is really on the edge later in the solidification in an area right below the riser-neck, see Fig. 3.45. The reason is that the riser is too small and the heaviest section of the casting slowly begins to be the last to solidify. If one should fully rely on the numerical results, solution 3 would be good enough for production. However, as

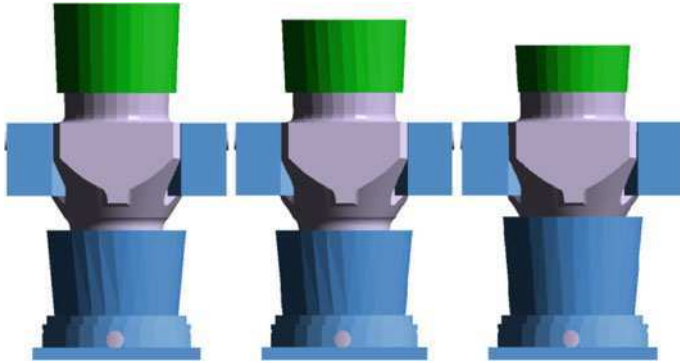


Fig. 3.44 Three distinct designs proposed by the optimisation tool [5]

Table 3.6 Comparison of the three optimised designs [5]

	Solution 1	Solution 2	Solution 3
Total height	4,037 mm	3,837 mm	3,537 mm
Total weight	48,406.8 kg	45,850 kg	40,968 kg
Height of the bottom cylindrical chills	999 mm	999 mm	1,149 mm
Thickness of the bottom cylindrical chills	160 mm	160 mm	160 mm

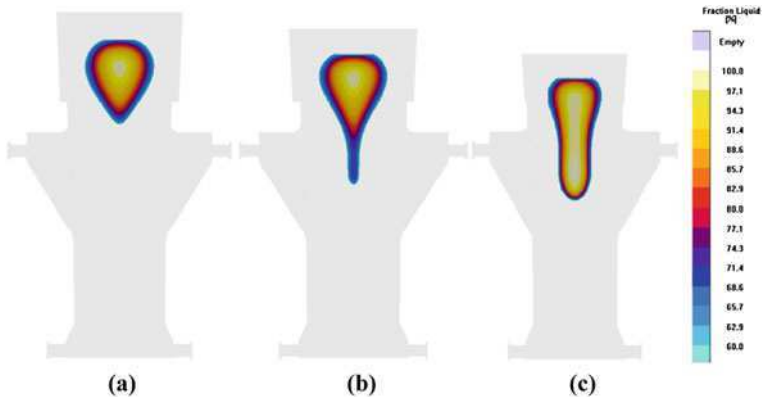
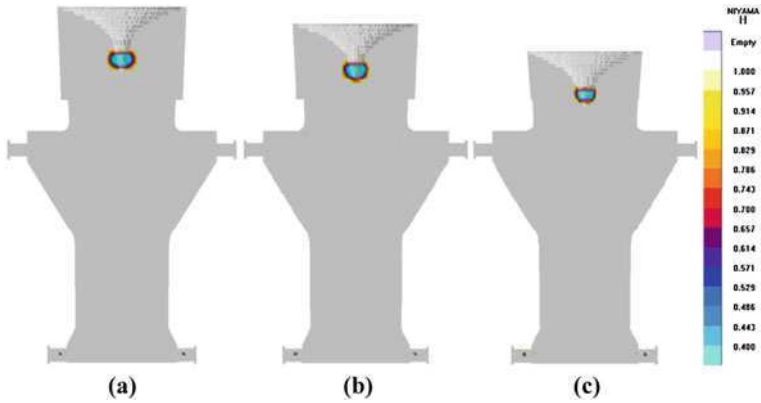


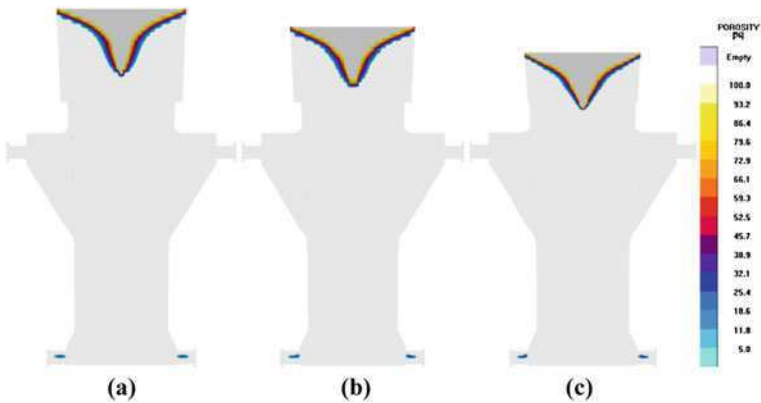
Fig. 3.45 Solidification pattern of the three optimised designs at 90% solidified [5]

argued before simulation does not take into account all crucial factors that occur in practice. For instance the quality of the melt can be compromised by a dirty ladle with residuals from the previous batch. Next could be the human factor, which often compromises the quality of a casting process. Having all this in mind it was decided together with the foundry not to go for solution 3 to avoid failure in production. But this solution is still shown and discussed here.





**Fig. 3.46** Occurrence of centreline porosity in the optimised designs [5]



**Fig. 3.47** Occurrence of macroscopic shrinkage in the optimised designs [5]

In Fig. 3.46, the centreline porosity is expressed by the Niyama criterion. The light blue areas stand for values of 0.4 and lower which will contain macroscopic shrinkage. Everything above 0.45 up to 1 will most likely be micro-porosity not detectable by the radiography techniques. It is seen that despite solution 3 being on the edge, it still shows no occurrence of porosity in the casting body. Only the bottom pins contain small porous areas. The reasonable remedy for this was addressed earlier.

A similar situation applies for shrinkage porosity shown in Fig. 3.47. The casting body appears to be porosity free in all three cases, except for the pins again.

The last assessment concerns the casting yield. The aim of the entire project has been primarily to eliminate the presence of various casting defects. Once this was achieved, the next step was to optimise the riser volume for the casting yield improvement. The results of this assessment are given in Table 3.7. Compared with

**Table 3.7** Casting yield assessment [5]

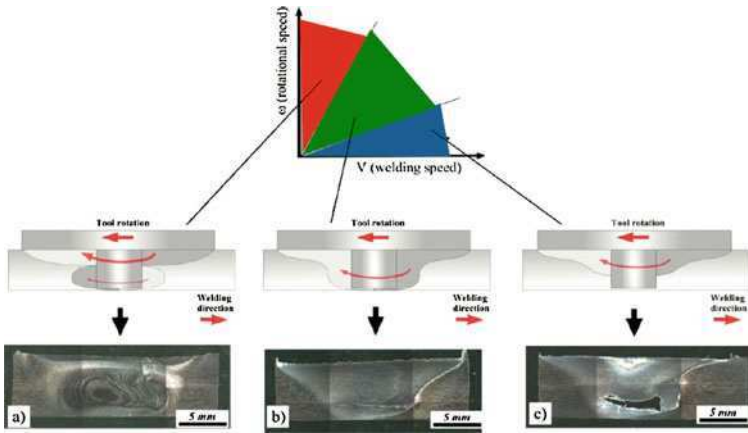
	Original solution	Optimised solution 1	Optimised solution 2	Optimised solution 3
Total height	4,337 mm	4,037 mm	3,837 mm	3,537 mm
Total weight	59,640 kg	48,406.8 kg	45,850 kg	40,968 kg
Casting yield	55.36%	61.76%	72.01%	80.59%

the original design the casting yield could be increased by approximately 25% if the optimised solution 3 was applied. Because of a high risk a shrinkage occurrence below the riser neck, solution 3 was however not approved for production.

### 3.6 Future Challenges

Unsurprisingly, some of the issues which are going to be presented briefly here in the present section have some overlaps with the joint work written by Miettinen et al. [39] on the new trends and future challenges which the multi-objective optimisation and decision making field is facing. However, the effort put in here is more directed towards the manufacturing process simulations and moreover the thermo-mechanical aspects of it.

Depending on the demands for the manufacturing process simulations, more specifically the level of interaction and complexity between different simulation domains, more than three objectives will eventually have to be optimised simultaneously. On top of that, considering improvement of the service-load performance of the products already during the initial design and manufacturing stages will contribute to this complexity even further [87, 113]. Therefore, many-objective problems will need to be solved and this field has actually received increasing attention recently [114–119]. Although current EMO procedures are quite successful in solving two or three objective problems, they have some computational deficiencies in finding multiple and well-spread solutions in case of problems comprising more than three objectives. Besides improving the inefficiency of selection operators available in current EMO algorithms (i.e., insufficient selective pressure, driven by the dominance, towards the true Pareto optimal front) without using very large population sizes since this is not practical in computationally expensive simulations, the number of objective functions or design variables could efficiently be reduced via some pattern recognition methodologies or data mining and clustering techniques. Moreover, the preference-based methods which utilise decision-maker preferences a priori, a posteriori or *progressively* [114], are arguably the best current techniques for handling large numbers of conflicting objectives. Such methodologies will be an essential part of the manufacturing process and product design using multi-objective optimisation tools. Improving the level of understanding the physical phenomena and implementation of the outcome of it into the numerical models to better capture the essential behaviour will also increase the interaction between experimentalists and the

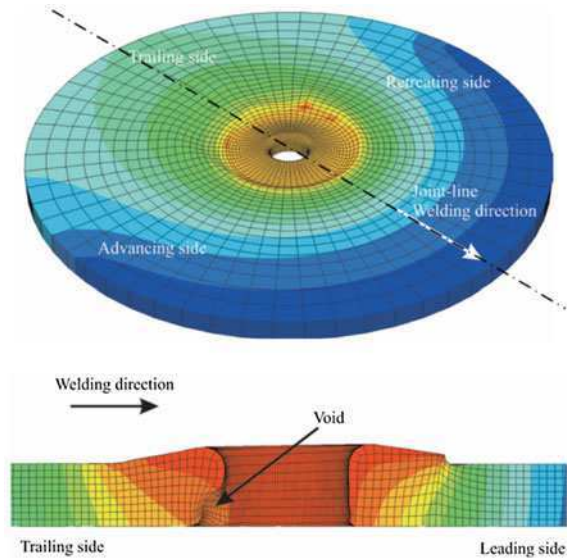


**Fig. 3.48** **a** Red denotes hot condition (over-stirring), low  $k$ ; **b** green denotes stable (robust) condition, intermediate  $k$ ; **c** blue denotes cold condition, high  $k$  ( $k$ : advancement per revolution (APR),  $k = u_{\text{weld}}/n_{\text{rev}}$ ) [89]

theoreticians for a more interactive decision-making procedure [120, 121] in a MOP. This will also be a potential path to deal with many-objective optimisation problems to reduce the search space, or in other words, focus on a partial set of it. However, these different expertises should be combined in an efficient and user-friendly way, for instance leading to a working environment combining dedicated process simulator and multi-objective optimisation capabilities powered by advanced algorithms (including meta-modelling techniques [122], hybrid algorithms, etc.), post-processing tools (scatter charts, parallel coordinates) and the aforementioned multi-criterion decision making tools. In this way, practitioners (e.g., foundry men), apart from academicians can also be involved in this iterative process of manufacturing product design without really considering the theoretical basis of the applied procedures.

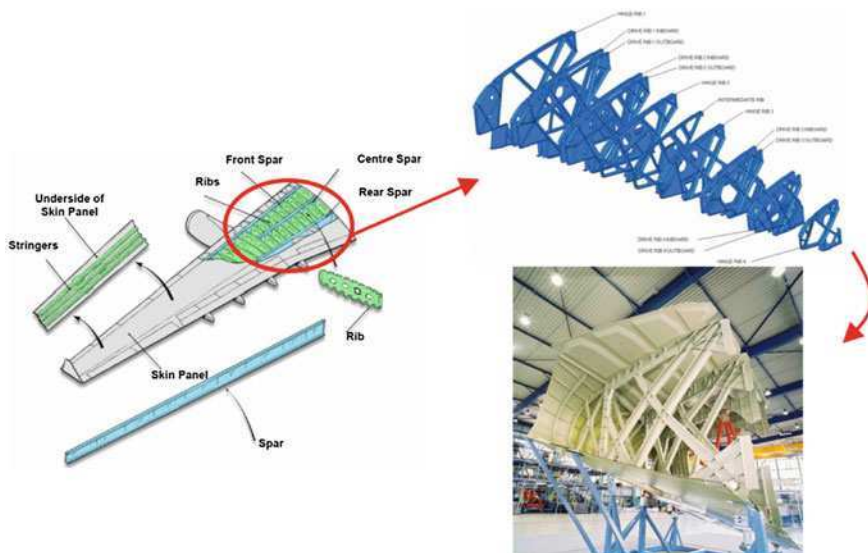
The field of knowledge discovery in MOO, which recently has been addressed in a more structured way under the name of “innovization” [61–63], as briefly mentioned in Sect. 3.4.3, seems to hold a big potential for the manufacturers. This autonomous way of discovering the common principles among the trade-off designs, which point out either the optimal process conditions or the optimal product designs, will help them to save time and resources, therefore money. For instance, referring to the problem briefly investigated in Sect. 3.4.3, investigation of defect-free welds while having higher production rate and keeping other manufacturing benefits in mind has always been a crucial problem for engineers. The main purpose, in case of the FSW process, is to find a robust work-frame (see Fig. 3.48b) which avoids hot and/or cold weld conditions (see Fig. 3.48a, c, respectively). Identification of these unknown “utopic” regions for different scales of mass production will allow manufacturers to keep their tools in certain geometrical sizes and shape for different welding speeds and different workpiece materials. This again requires efficient integration of realistic process simulations

**Fig. 3.49** *Top* three-dimensional local FE model in the FSW process; *bottom* an example of void formation predicted by the model [80]



with multi-objective optimisation tools. That being said, one should bear in mind that defect predictions via simulation in FSW have not been addressed satisfactorily yet due to several challenges. Extreme deformation of the FE mesh due to the stirring motion is one of the severe difficulties to be handled, besides complex contact boundary conditions at the tool/workpiece interface. Figure 3.49 shows one of the most advanced FSW thermo-mechanical models incorporating an ALE formulation and having the contact boundary conditions as part of the solution set therefore enabling the separation at the interface to be captured (see void formation behind the tool in Fig. 3.49, bottom row) [80].

Improvement in computational resources, including clusters, grid computing, etc., will always be a positive side effect for both of the research areas. Parallelisation of non-overlapping regions of the Pareto optimal front (which has already been applied by [123]) is another way of using distributed resources. Besides these more common issues assuming that the simulation times for each design set are more or less equal, the more unique case, such as having a range of different welding speeds leading to different computation time, should also be taken into account. Thus, a more efficient distribution of these non-homogeneous large-scale simulations among resources will be more crucial. This will call for an efficient hybridisation strategy of shared and distributed memory applications. Graphical processing units (GPUs) are also good candidates for these types of applications even though the current attempts are mostly at an individual or non-standard level. This potential gap will play an important role for commercial (simulation and multi-objective optimisation) software companies to stay competitive. For instance, implementation of a GPU-based Conjugate Gradient solver in a commercial software that is used to simulate a casting process will not only allow to investigate further details in understanding of particular physical phenomena, but also open the

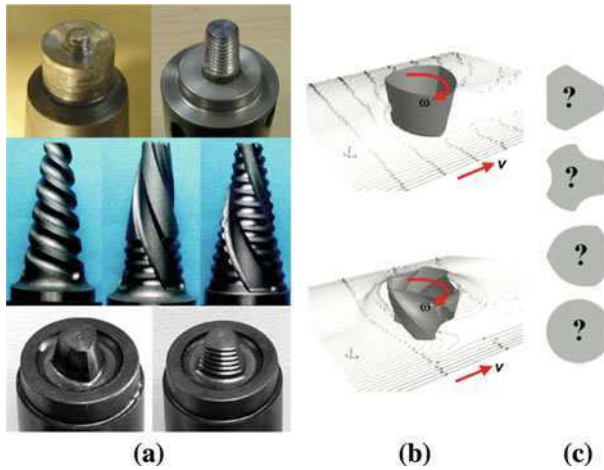


**Fig. 3.50** *Left* topology optimisation (weight minimisation) of the Airbus A380 (main wing box) inner leading edge ribs; *Right* actual prototype at the final stage after shape and size optimisation [127, 128]

doors to perform a multi-objective optimisation using a fully coupled, e.g., thermo-mechanical, simulation both to reduce the porosities, residual stresses and to maximise the product performance under service-loads.

Material layout (or “pseudo-density” in SIMP approach [124]) optimisation, i.e., topology optimisation [125] in common terminology, is in essence a semi-definite optimisation application [126] where the objective is minimised with respect to a constraint represented as a positive semi-definite matrix. In other words, the optimum distribution of a fixed amount of material in a restricted domain is sought. A recent aerospace design application, i.e., weight minimisation of the Airbus A380 (main wing box) inner leading edge ribs [127, 128], is shown in Fig. 3.50 where compliance of the structure is treated as a constraint resembling an  $\varepsilon$ -constraint problem [59] as also mentioned in Sect. 3.4.3. Initially the ribs are having few holes enabling wiring, next further material removal in optimum locations are performed with specialised algorithms which use direct or adjoint sensitivity calculations [125], then shape and size optimisation are applied, and following this, CAD models of the ribs are prepared. Figure 3.50 (right) shows the actual prototype at the final stage. Similar variations of this material distribution problem under structural loads with/without flow around the structure, or thermo-elastic behaviour, etc., have been investigated [125].

This design procedure naturally excites us and brings several questions to our minds, for instance: can we design a FSW tool pin to increase the plastic work and the friction heating, to promote the material stirring, closure of voids and dispersion of surface oxides by formulating this design assignment as a semi-definite

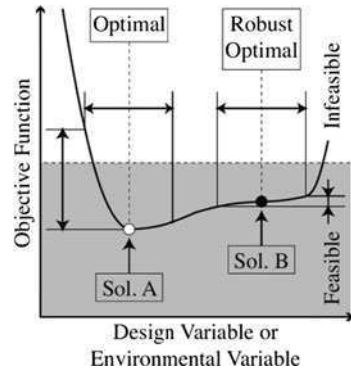


**Fig. 3.51** **a** There are many FSW tool probe designs currently available in the market [47, 89]; **b** A few examples of CFD simulations of different FSW tool probes [129]; **c** The unknown cross-sectional designs [89]

optimisation problem? There are many tool probe designs currently available in the market, as seen in Fig. 3.51a [47, 89], however almost all of them have been designed by trial-and-error approach, a systematic approach has not been presented yet. For such a systematic approach, first of all one should have a robust CFD solver (the shear thinning effect in the workpiece material brings extra nonlinearities) and on top of that, a CSM solver should be applied in order to investigate the strength and fatigue endurance of the pin while traversing ahead without failure for a reasonable range of welding speeds. Figure 3.51b shows CFD simulations of some well-known FSW tool probes [129] and Fig. 3.51c indicates the unknown cross-sectional designs [89].

Engineers know that every parameter of an analysis is subjected to scatter and randomness, e.g., material property values differ inherently from one specimen to the next, geometric properties of components can only be reproduced within certain manufacturing tolerances, and almost all thermal input parameters such as heat transfer coefficients used in finite element analyses are inexact and the degree of uncertainty grows sharply at elevated temperatures. It is neither physically possible nor financially feasible to eliminate the scatter of input parameters completely. The reason for this is that the reduction of scatter typically is associated with higher costs either through better and more precise manufacturing methods and processes or increased efforts in quality control; hence, accepting the existence of scatter and dealing with it rather than trying to eliminate it, makes products more affordable and production of those products more cost-effective [130–132]. Therefore, real world optimisation applications, e.g., manufacturing of a wind turbine blade aiming at reduction of its weight meanwhile improving its aeroelastic behaviour and endurance in varying wind loads by controlling the curing process to keep the residual stress related defects in an acceptable range, inevitably involve

**Fig. 3.52** Comparison between conventional optimisation and robust optimisation (for a minimisation problem); conventional optimal solution A versus robust optimal solution B [133]



uncertainties in manufacturing processes and operating conditions. A brief and self-explaining comparison between traditional optimisation and robust optimisation is represented in Fig. 3.52 [133], where solution-A is the global optimum in a traditional sense, although having a risk of getting an infeasible response when being exposed to small variations in design variables or environmental parameters, whereas solution-B is moderately good in terms of optimality and moreover it lies on a more flat region of the objective function, thus the dispersion of the objective function is narrow against perturbations in the design variable.

Having said that, optimal solution(s) in a real world optimisation problem should provide higher performance having satisfactory robustness which might be conflicting with the former in some cases, however the latter has been included in a few engineering fields using different optimisation methodologies [134–136]. Robustness can be studied either by replacing the original objective function by an expression measuring both the performance and the expectation of each criterion in the vicinity of a specific solution, or by inserting an additional optimisation criterion assessing robustness in addition to the original criteria [13] and this has recently been addressed in multi-objective optimisation problems [137, 138]. It is the firm expectation of the authors, that these theoretical studies will soon be combined with manufacturing process simulations such as the ones investigated briefly in this section and numerous other processes having similar physical aspects, as well.

## References

1. Tekkaya, E. (2000). State-of-the-art of simulation of sheet metal forming. *Journal of Materials Processing Technology*, 103, 14–22.
2. Tutum, C. C., & Hattel, J. H. (2010). Optimisation of process parameters in friction stir welding based on residual stress analysis: A feasibility study. *Journal of Science and Technology of Welding and Joining*, 15(5), 369–377.
3. Tutum, C. C., & Hattel, J. H. (2010). Multi-objective optimization of process parameters in friction stir welding. In *Genetic and evolutionary computation conference (GECCO 2010)*, Portland, Oregon (pp. 1323–1324).



4. Tutum, C. C., & Hattel, J. H. (2010). A multi-objective optimization application in friction stir welding: considering thermo-mechanical aspects. In *IEEE congress on evolutionary computation (IEEE CEC 2010), Barcelona, Spain* (pp. 427–434).
5. Kotas, P., Tutum, C. C., Snajdrova, O., Thorborg, J., & Hattel, J. H. (2010). A casting yield optimization case study: Forging ram. *International Journal of Metalcasting*, 4(4), 61–76.
6. Dulikravich, G. S., Egorov, I. N., & Colaco, M. J. (2008). Optimizing chemistry of bulk metallic glasses for improved thermal stability. *Modelling and Simulation in Materials Science and Engineering*, 16(7), 1–19.
7. Dulikravich, G. S., Sikka, V. K., & Muralidharan, G. (2006). *Development of semi-stochastic algorithm for optimizing alloy composition of high-temperature austenitic stainless steels (H-series) for desired mechanical and corrosion properties*. Technical Report, U.S. Department of Energy, (<http://www.osti.gov/bridge>).
8. Wei, L., & Yuying, Y. (2008). Multi-objective optimization of sheet metal forming process using Pareto-based algorithm. *Journal of Materials Processing Technology*, 208, 499–506.
9. Schenk, O., & Hillmann, M. (2004). Optimal design of metal forming die surfaces with evolution strategies. *Journal of Computers and Structures*, 82(20–21), 1695–1705.
10. Jansson, T., Andersson, A., & Nilsson, L. (2005). Optimization of draw-in for an automotive sheet metal part—an evaluation using surrogate models and response surfaces. *Journal of Materials Processing Technology*, 159, 426–434.
11. Oduguwa, V., & Roy, R. (2002). Multi-objective optimisation of rolling rod product design using meta-modelling approach. In *Genetic and evolutionary computation conference (GECCO 2002), San Francisco, CA, USA* (pp. 1164–1171).
12. Katayama, T., Nakamachi, E., Nakamura, Y., Ohata, T., Morishita, Y., & Murase, H. (2004). Development of process design system for press forming—multi-objective optimization of intermediate die shape in transfer forming. *Journal of Materials Processing Technology*, 155–156, 1564–1570.
13. Han, Z. X., Xu, L., Wei, R., Wang, B. P., & Reinikainen, T. (2004). Reliability-based design optimization for land grid array solder joints under thermo-mechanical load. In *5th international conference on thermal and mechanical simulation and experiments in micro-electronics and micro-systems (EuroSimE2004)* (pp. 219–224).
14. Kor, J., Chen, X., & Hu, H. (2009). Multi-objective optimal gating and riser design for metal-casting. In *IEEE international symposium on intelligent control, Saint Petersburg, Russia* (pp. 428–433).
15. Esparza, C. E., Guerrero-mata, M. P., & Ríos-mercado, R. Z. (2006). Optimal design of gating systems by gradient search methods. *Computational Materials Science*, 36, 57–467.
16. Poloni, C., Poles, S., Odorizzi, S., Gramegna, N., & Bonollo, F. (2002). MAGMAfrontier: State of the art of an optimisation tool for the MAGMASOFT environment. MAGMASOFT international user meeting.
17. Hahn, I., & Hartmann, G. (2008). Automatic computerized optimization in die casting processes. *Casting Plant & Technology*, 4, 2–14.
18. Georgiev, G., & Ivanov, G., (2010). New interactive and automatic optimization procedures offered of the recent foundry simulation software. [www.nts-bg.ttm.bg/journal/papers/25.pdf](http://www.nts-bg.ttm.bg/journal/papers/25.pdf).
19. Kirk, D. B., & Hwu, W.-M. W. (2010). *Programming massively parallel processors—a hands-on approach*. San Francisco, USA: Morgan Kaufmann Publishers.
20. <http://en.wikipedia.org/wiki/InfiniBand>.
21. Chapman, B., Jost, G., & Pas, R. V. D. (2008). *Using openMP: Portable shared memory parallel programming*. Cambridge: The MIT Press.
22. Pacheco, P. S. (1997). *Parallel programming with MPI*. San Francisco, USA: Morgan Kaufmann Publishers, Inc.
23. Talbi, E.-G., Mostaghim, S., Okabe, T., Ishibuchi, H., Rudolph, G., & Coello, C. A. C. (2008). Parallel approaches for multiobjective optimization. In J. Branke et al. (Eds.), *Multiobjective optimization, LNCS 5252* (pp. 349–372). Berlin: Springer.
24. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester: Wiley.



25. Deb, K. (2008). Introduction to evolutionary multiobjective optimization. In J. Branke et al. (Eds.), *Multiobjective optimization, LNCS 5252* (pp. 59–96). Berlin: Springer.
26. Betounes, D. (1998). *Partial differential equations for computational science: With Maple and vector analysis*. New York: Springer.
27. Cook, R., Malkus, D., Plesha, M., & Witt, R. (2001). *Concepts and applications of finite element analysis*. New York: Wiley.
28. Simo, J. C., & Hughes, T. J. R. (1998). *Computational inelasticity*. New York, USA: Springer.
29. Crisfield, M. (1991). *Non-linear finite element analysis of solids and structures—volume 1: Essentials*. Chichester: Wiley.
30. Crisfield, M. (1991). *Non-linear finite element analysis of solids and structures—volume 2: Advanced topics*. Chichester: Wiley.
31. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182–197.
32. Zitzler, E., & Thiele, L. (1998). An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Swiss Federal Institute of Technology (ETH) Zurich.
33. Knowles, J., & Corne, D. (1998). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimization. Parallel problem solving from nature—PPSN V. In 5th *International Conference* (pp. 250–259).
34. Horn, J., Nafpliotis, N., & Goldberg, D. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *First IEEE conference on evolutionary computation. IEEE world congress on computational intelligence* (pp. 82–87).
35. Corne, D., Knowles, J., & Oates, M. (2000). The pareto envelope-based selection algorithm for multiobjective optimization. In 6th *international conference on parallel problem solving from nature* (pp. 839–848).
36. Fonseca, C., & Fleming, P. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In 5th *international conference on genetic algorithms, USA*.
37. Knowles, J., Corne, D., & Deb, K. (2008). *Multi-objective problem solving from nature. Natural computing series*. Heidelberg: Springer.
38. Coello, C. A. C., Pulido, G., & Montes, E. (2005). Current and future research trends in evolutionary multiobjective optimization. *Advanced Information and Knowledge Processing*, pp. 213–231.
39. Miettinen, K., Deb, K., Jahn, J., Ogryczak, W., Shimoyama, K., & Vetschera, R. (2008). Future challenges. In J. Branke et al. (Eds.), *Multiobjective optimization, LNCS 5252* (pp. 435–461). Berlin: Springer.
40. Poles, S., Vassileva, M., & Sasaki, D. (2008). Multiobjective optimization software. In J. Branke et al. (Eds.), *Multiobjective optimization, LNCS 5252* (pp. 329–348). Berlin: Springer.
41. Miettinen, K., & Mäkelä, M. M. (2006). Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3), 909–922.
42. Hattel, J. H. (2005). *Fundamentals of numerical modelling of casting processes*. Kgs. Lyngby: Polyteknisk Forlag.
43. Lindgren, L. (2007). *Computational welding mechanics: Thermomechanical and microstructural simulations*. Cambridge: Woodhead Publishing, Ltd.
44. Boley, B., & Weiner, J. (1960). *Theory of thermal stresses*. New York: Dover.
45. Hattel, J. H., Schmidt, H. B., & Tutum, C. C. (2008). Thermomechanical modelling of friction stir welding. In 8th *international conference on trends in welding research conference, ASM, Atlanta, USA*.
46. Tutum, C. C. (2009). *Optimization of thermo-mechanical conditions in friction stir welding*. Ph.D. thesis, ISBN: 978-87-89502-89-2.
47. <http://www.twi.co.uk/>.

48. Mishra, R. S., & Ma, Z. Y. (2005). Friction stir welding and processing. *Materials and Science Engineering A*, 50, 1–78.
49. Nandan, R., DebRoy, T., & Bhadeshia, H. (2008). Recent advances in friction stir welding—process, weldment structure and properties. *Progress in Materials Science*, 53, 980–1023.
50. Tutum, C. C., Schmidt, H. N. B., & Hattel, J. H. (2008). Assessment of benchmark cases for modelling of residual stresses and distortions in friction stir welding. In 7th *international symposium friction stir welding, TWI, Awaji Island, Japan*.
51. Tutum, C. C., Schmidt, H., Hattel, J., & Bendsøe, M. (2007). Estimation of the welding speed and heat input in friction stir welding using thermal models and optimization. In 7th *world congress on structural and multidisciplinary optimization, Seoul* (pp. 2639–2646).
52. Tutum, C. C., Deb, K., & Hattel, J. H. (2010). Hybrid search for faster production and safer process conditions in friction stir welding. In *The eighth international conference on simulated evolution and learning (SEAL-2010), IITK Kanpur, India (accepted)*.
53. Liao, T. W., & Daftardar, S. (2009). Model based optimization of friction stir welding processes. *Science and Technology of Welding and Joining*, 14(5), 426–435.
54. Larsen, A. A., Bendsøe, M. P., Hattel, J. H., & Schmidt, H. N. B. (2009). Optimization of friction stir welding using space mapping and manifold mapping—an initial study of thermal aspects. *Structural and Multidisciplinary Optimization*, 38(3), 289–299.
55. Nandan, R., Lienert, T. J., & DebRoy, T. (2008). Toward reliable calculations of heat and plastic flow during friction stir welding of Ti-6Al-4V alloy. *International Journal of Materials Research*, 99(4), 434–444.
56. Schmidt, H. N. B., & Hattel, J. H. (2008). Thermal modelling of friction stir welding. *Scripta Materialia*, 58, 332–337.
57. Schmidt, H. N. B., & Hattel, J. H. (2005). Modelling heat flow around tool probe in friction stir welding. *Science and Technology of Welding and Joining*, 10(2), 176–186.
58. Schmidt, H. N. B., Hattel, J. H., & Wert, J. (2004). An analytical model for the heat generation in friction stir welding. *Modeling and Simulation in Materials Science and Engineering*, 12(1), 143–157.
59. Haimes, Y. Y., Lasdon, L. S., & Wismer, D. A. (1971). On a bi-criterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 1(3), 296–297.
60. Deb, K. (2003). Unveiling innovative design principles by means of multiple conflicting objectives. *Engineering Optimization*, 35(5), 445–470.
61. Deb, K., & Srinivasan, A. (2006). Innovization: Innovating design principles through optimization. In *Genetic and evolutionary computation conference (GECCO 2006), New York, NY, USA* (pp. 1629–1636).
62. Deb, K., & Chaudhuri, S. (2005). Automated discovery of innovative designs of mechanical components using evolutionary multi-objective algorithms. In *Evolutionary machine design: methodology and applications* (pp. 143–168).
63. Bandaru, S., & Deb, K. (2010). Automated discovery of vital knowledge from Pareto-optimal solutions: First results from engineering design. In *IEEE congress on evolutionary computation (CEC 2010)* (pp. 1224–1231).
64. Michaleris, P., & Sun, X. (1997). Finite element analysis of thermal tensioning techniques mitigating weld buckling distortion. *Welding Journal*, 76, 451–457.
65. Michaleris, P., Dantzig, J., & Tortorelli, D. (1999). Minimization of welding residual stress and distortion in large structures. *Welding Journal*, 78, 361–366.
66. Richards, D. G., Prangnell, P. B., Williams, S. W., & Withers, P. J. (2008). Global mechanical tensioning for the management of residual stresses in welds. *Materials Science and Engineering*, 489, 351–362.
67. Hatamleh, O., Lyons, J., & Forman, R. (2007). Laser and shot peening effects on fatigue crack growth in friction stir welded 7075–T7351 aluminum alloy joints. *International Journal of Fatigue*, 29(3), 421–434.

68. Hatamleh, O. (2008). The effects of laser peening and shot peening on mechanical properties in friction stir welded 7075-T7351 aluminum. *Journal of Materials Engineering and Performance*, 17, 688–694.
69. Richards, D. G., Prangnell, P. B., Withers, P. J., Williams, S. W., Nagy, T., & Morgan, S. (2008). Simulation of the effectiveness of dynamic cooling for controlling residual stresses in friction stir welds. In 7th international symposium friction stir welding, TWI, Japan.
70. Feng, Z., Wang, X., David, S. A., & Sklad, P. (2007). Modelling of residual stresses and property distributions in friction stir welds of aluminium alloy 6061-T6. *Science and Technology of Welding & Joining*, 12(4), 348–356.
71. Chao, Y. J., & Qi, X. H. (1999). Thermal and thermo-mechanical modelling of friction stir welding of aluminium alloy 6061-T6. *Journal of Materials Processing Manufacturing Science*, 7, 215–233.
72. Zhu, X. K., & Chao, Y. J. (2004). Numerical simulation of transient temperature and residual stresses in friction stir welding of 304L stainless steel. *Journal of Materials Processing Technology*, 146, 263–272.
73. Shi, Q., Dickerson, T., & Shercliff, H. (2003). Thermal-mechanical FE modeling of friction stir welding of Al-2024 including tool loads. In *Proceedings of the 4th international symposium on FSW, TWI, Utah, USA*.
74. Chen, C. M., & Kovacevic, R. (2006). Parametric finite element analysis of stress evolution during friction stir welding. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 220(8), 1359–1371.
75. Bastier, A., Maitournam, M. H., Van, K. D., & Roger, F. (2006). Steady state thermomechanical modelling of friction stir welding. *Science and Technology of Welding and Joining*, 11, 278–288.
76. Li, T., Shi, Q. Y., & Li, H. K. (2007). Residual stresses simulation for friction stir welded joint. *Science and Technology of Welding and Joining*, 12(8), 634–640.
77. Dubourg, L., Doran, P., Larose, S., Gharghoury, M. A., & Jahazi, M. (2010). Prediction and measurements of thermal residual stresses in AA2024-T3 friction stir welds as a function of welding parameters. *Materials Science Forum*, 638–642, 1215–1220.
78. Qin, X., & Michaleris, P. (2009). Thermo-elasto-viscoplastic modelling of friction stir welding. *Science and Technology of Welding and Joining*, 14(7), 640–649.
79. Xu, S., & Deng, X. (2003). Two and three-dimensional finite element models for the friction stir welding process. In 4th international symposium on friction stir welding, UT, USA.
80. Schmidt, H. N. B., & Hattel, J. H. (2005). A local model for the thermomechanical conditions in friction stir welding. *Modelling and Simulation in Materials Science and Engineering*, 13, 77–93.
81. Zhang, H., & Zhang, Z. (2007). Numerical modelling of friction stir welding process by using rate-dependent constitutive model. *Journal of Materials Science and Technology*, 23(1), 73–80.
82. Hamilton, R., MacKenzie, D., & Li, H. (2011). Multi-physics simulation of friction stir welding process. *Engineering Computations*, 27(8), 967–985.
83. Bastier, A., Maitournam, M. H., Roger, F., & Dang Van, K. (2008). Modelling of the residual state of friction stir welded plates. *Journal of Materials Processing Technology*, 200, 25–37.
84. Grujicic, M., Arakere, G., Yalavarthy, H. V., He, T., Yen, C.-F., & Cheeseman, B. A. (2010). Modeling of AA5083 material-microstructure evolution during butt friction stir welding. *Journal of Materials Engineering and Performance*, 19(5), 672–684.
85. Vuyst, T. D., Madhavan, V., Ducoeur, B., Simar, A., Meester, B. D., & D'Alvise, L. (2008). A thermo-fluid/thermo-mechanical modeling approach for computing temperature cycles and residual stresses in FSW. In 7th international symposium on friction stir welding, Japan.
86. Altenkirch, J., Steuwer, A., Peel, M., Richards, D. G., & Withers, P. J. (2008). The effect of tensioning and sectioning on residual stresses in aluminum AA7749 friction stir welds. *Materials Science and Engineering A*, 488, 16–24.

87. Hattel, J. H., & Tutum, C. C. (2011). Modelling residual stresses in friction stir welding of Al-alloys - A review of possibilities and future trends. *International Journal of Advanced Manufacturing Technology* (under review).
88. Zhu, X. K., & Chao, Y. J. (2002). Effects of temperature-dependent material properties on welding simulation. *Computers & Structures*, 80, 967–976.
89. Jahazi, M., Dubourg, L., & Cao, X. (2008). Friction stir welding of aerospace alloys. <http://www.smeacanada.ca/montreal/presentations/Jahazi-Mohammad-FSW.pdf>.
90. ESAB Holdings Ltd., UK, <http://www.esab.com>.
91. [http://www.twi.co.uk/content/twi\\_yorks\\_fsw.html](http://www.twi.co.uk/content/twi_yorks_fsw.html).
92. Groover, M. P. (2002). *Fundamentals of modern manufacturing—materials, processes, and systems*. Hoboken: Wiley.
93. <http://en.wikipedia.org/wiki/Casting>.
94. Flender, E., & Sturm, J. (2010). Thirty years of casting process simulation. *International Journal of Metalcasting, Spring*, 10, 7–23.
95. Winterscheidt, D. L., & Huang, G. X. (2002). Fundamentals of casting process modeling. In Yu Kuang-o (Ed.), *Modeling for casting and solidification processing* (pp. 17–54). New York, USA: Marcel Dekker, Inc.
96. Chandra, U., & Ahmed, A. (2002). Stress analysis. In Yu Kuang-o (Ed.), *Modeling for casting and solidification processing* (pp. 55–93). New York, USA: Marcel Dekker, Inc.
97. Suri, V., & Yu, K.-O (2002). Defects formation. In Yu Kuang-o (Ed.), *Modeling for casting and solidification processing* (pp. 95–122). New York, USA: Marcel Dekker, Inc.
98. Stefanescu, D. M. (2002). Microstructure evolution. In Yu Kuang-o (Ed.), *Modeling for casting and solidification processing* (pp. 123–187). New York, USA: Marcel Dekker, Inc.
99. Cleary, P. W., Ha, J., Prakash, M., & Nguyen, T. (2006). 3D SPH flow predictions and validation for high pressure die casting of automotive components. *Applied Mathematical Modelling*, 30, 1406–1427.
100. Thorborg, J., Hattel, J. H., & Bellini, A. (2006). Thermo-mechanical conditions in heat treated aluminium cast parts. In *Proceedings of 11th modelling of casting, welding and advanced solidification processes* (pp. 193–200).
101. Campbell, J. (2003). *Castings*, (2<sup>nd</sup> ed.). Oxford: Butterworth Heinemann.
102. Hansen, S. S. (2007). *Reduced energy consumption for melting in foundries*. Ph.D. thesis, Technical University of Denmark, ISBN 978-87-91035-63-5.
103. Niyama, E., Uchida, T., Morikawa, M., & Saito, S. (1982). Method of shrinkage prediction and its application to steel casting practice. *AFS International Cast Metals Journal*, 7(3), 52–63.
104. Carlson, K. D., & Beckermann, Ch. (2009). Prediction of shrinkage pore volume fraction using a dimensionless Niyama criterion. *Metallurgical and Materials Transactions A*, 40A, 163–175.
105. Carlson, K. D., Ou, S., & Beckermann, C. (2005). Feeding of high-nickel alloy castings. *Metallurgical Materials Transactions B*, 36B, 843–856.
106. Jain, N., Carlson, K. D., & Beckermann, C. (2007). Round robin study to assess variations in casting simulation Niyama criterion predictions. In *Proceedings of the 61st technical and operating conference, Steel Founders' Society of America, Chicago*.
107. Carlson, K. D., Ou, S., Hardin, R. A., & Beckermann, C. (2001). Development of a methodology to predict and prevent leaks caused by microporosity in steel castings. In *Proceedings of the 55th technical and operating conference, SFSA, Chicago*.
108. Hardin, R. A., Ou, S., Carlson, K. D., & Beckermann, C. (2002). Development of new feeding distance rules using casting simulation; Part I: Methodology. *Metallurgical Materials Transactions B*, 33B, 731–740.
109. Beckermann, C. (2002). Modelling of macrosegregation: Applications and future needs. *International Materials Reviews*, 47, 243–261.
110. Beeley, P. (2001). *Foundry technology* (2<sup>nd</sup> ed.). Oxford: Butterworth Heinemann.
111. Porter, D. A., & Easterling, K. E. (2004). *Phase transformations in metals and alloys*, (2<sup>nd</sup> ed.). London: Taylor & Francis Group.

112. Sobol, I. (1979). On the systematic search in a hypercube. *SIAM Journal on Numerical Analysis*, 16, 790–793.
113. Hattel, J. H., Nielsen, K. L., & Tutum, C. C. (2010). The effect of post-welding conditions in friction stir welds: From weld simulation to ductile failure. *European Journal of Mechanics - A/Solids* (under review).
114. Fleming, P. J., Purshouse, R. C., & Lygoe, R. J. (2005). Many-objective optimization: An engineering design perspective. In C. A. Coello Coello et al. (Eds.), *EMO 2005, LNCS 3410* (pp. 14–32). Berlin: Springer.
115. Deb, K., & Saxena, D. K. (2005). On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. KanGAL Report Number 2005011.
116. Saxena, D. K., Ray, T., Deb, K., & Tiwari, A. (2009). Constrained many-objective optimization: A way forward. In *IEEE congress on evolutionary computation (CEC 2009)* (pp. 545–552).
117. Brockhoff, D., & Zitzler, E. (2006). Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In T. P. Runarsson et al. (Eds.), *PPSN IX, LNCS 4193* (pp. 533–542). Berlin: Springer.
118. Corne, D., & Knowles, J. (2007). Techniques for highly multiobjective optimisation: Some nondominated points are better than others. In *Genetic and evolutionary computation conference (GECCO 2007), London, England, United Kingdom* (pp. 773–780).
119. Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K., & Zhang, Q. (2010). Objective reduction in many-objective optimization: Linear and nonlinear algorithms. KanGAL Report Number 2010008.
120. Deb, K., Sundar, J., Rao, U. B., & Chaudhuri, S. (2006). Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(3), 273–286.
121. Deb, K., & Kumar, A. (2007). Light beam search based multi-objective optimization using evolutionary algorithms. KanGAL Report 2007005.
122. Jin, Y. (2005). A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1), 3–12.
123. Branke, J. (2008) Consideration of partial user preferences in evolutionary multiobjective optimization. In J. Branke et al. (Eds.), *Multiobjective optimization, LNCS 5252* (pp. 157–178). Berlin: Springer.
124. Sigmund, O. (2001). A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21, 120–127.
125. Bendsoe, M. P., & Sigmund, O. (2003). *Topology optimization—theory, methods and applications*. Berlin: Springer.
126. Jahn, J. (2004). *Introduction to the theory on nonlinear optimization*. Heidelberg: Springer.
127. Krog, L., Tucker, A., & Rollema, G. (2002). *Application of topology, sizing and shape optimization methods to optimal design of aircraft components*. Seattle: Altair Engineering Ltd.
128. Krog, L., Tucker, A., Kemp, M., & Boyd, R. (2004). Topology optimization of aircraft wing box ribs. In *The Altair technology conference* (pp. 6.1–6.16).
129. Colegrove, P. A., & Shercliff, H. R. (2004). Development of trivex friction stir welding tool. Part 1—two-dimensional flow modeling and experimental validation. *Science and Technology of Welding and Joining*, 9, 345–351.
130. Vlahinos, A., & Kelkar, S. G. (2002). Designing for six-sigma quality with robust optimization using CAE. In *Proceedings of the 2002 SAE international body*. Deb, K. (2007). Current trends in evolutionary multi-objective optimization. *International Journal for Simulation and Multidisciplinary Design Optimization*, 1, 1–8.
131. Vlahinos, A., Suryatama, D., Ullahkhan, M., TenBrink, J. T., & Baker, E. (2002). Robust design of a catalytic converter with material and manufacturing variations. In *Powertrain & fluid systems conference & exhibition, San Diego, California USA*.
132. ANSYS Inc: Probabilistic design techniques.

133. Ferreira, J., Fonseca, C. M., Covas, J. A., & Gaspar-Cunha, A. (2008). *Evolutionary multi-objective robust optimization. Advances in evolutionary algorithms* (pp. 261–278). Vienna, Austria: I-Tech Education and Publishing.
134. Wiesmann, D., Hammel, U., & Back, T. (1998). Robust design of multilayer optical coatings by means of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2(4), 162–167.
135. Du, X., & Chen, W. (1998). Towards a better understanding of modelling feasibility robustness in engineering design. In *Proceedings of DETC 99, Las Vegas, USA*.
136. Arrol, D. V., & Beyer, H.-G. (2003). A comparison of evolution strategies with other direct search methods in the presence of noise. *Computational Optimization and Applications*, 24(1), 135–159.
137. Deb, K., & Gupta, H. (2005). Searching for robust pareto-optimal solutions in multi-objective optimization. In C. A. C. Coello et al. (Eds.), *EMO 2005, LNCS 3410* (pp. 150–164). Heidelberg: Springer.
138. Daum, D. A., Deb, K., & Branke, J. (2007). Reliability-based optimization for multiple constraints with evolutionary algorithms. In *IEEE congress on evolutionary computation (CEC 2007), Singapore* (pp. 911–918).

**Part II**  
**Product Design and Optimisation**

# Chapter 4

## Many-Objective Evolutionary Optimisation and Visual Analytics for Product Family Design

Ruchit A. Shah, Patrick M. Reed and Timothy W. Simpson

**Abstract** Product family design involves the development of multiple products that share common components, modules and subsystems, yet target different market segments and groups of customers. The key to a successful product family is the product platform—the common components, modules and subsystems—around which the family is derived. The fundamental challenge when designing a family of products is resolving the inherent trade-off between commonality and performance. If there is too much commonality, then individual products may not meet their performance targets; however, too little sharing restricts the economies of scale that can be achieved during manufacturing and production. Multi-objective evolutionary optimisation algorithms have been used extensively to address this trade-off and determine which variables should be common (i.e., part of the platform) and which should be unique in a product family. In this chapter, we present a novel approach based on many-objective evolutionary optimisation and visual analytics to resolve trade-offs between commonality and many performance objectives. We provide a detailed example involving a family of aircraft that demonstrates the challenges of solving a 10-objective trade-off between commonality and the nine performance objectives in the family. Future research

---

R. A. Shah · T. W. Simpson (✉)  
Industrial & Manufacturing Engineering, Pennsylvania State University,  
University Park, USA  
e-mail: tws8@psu.edu

R. A. Shah  
e-mail: ruchit@psu.edu

P. M. Reed  
Civil & Environmental Engineering, Pennsylvania State University,  
University Park, USA  
e-mail: preed@enr.psu.edu



directions involving the use of multi-objective optimisation and visual analytics for product family design are also discussed.

## **4.1 Balancing Commonality and Performance During Product Family Design**

For most companies, product variety is a key to maintaining their market share. Today there are wide arrays of choices available for nearly all consumer products and services; thus, for a company to create a niche for itself its product offerings must be diverse enough to appeal to multiple market segments. However, offering a wide variety of products has its downsides as proliferation of product variety may incur substantial costs to the company [1–6] and reduce its profitability. Many companies struggle to provide variety in their product offerings while maintaining reasonably low costs. This often results from a company's failure to embrace commonality, compatibility, standardisation and modularisation across the product lines [7], which degrades a company's ability to achieve economies of scale across their production/manufacturing process.

Unique product offerings are advantageous to customers but expensive for companies to achieve. High product variety offers customers options customised to their specific needs and preferences but reduces the margins for the company as the increased price might not be proportional to the perceived value estimated by the customer. Commonality on the other hand is cost-effective for the company, but it can compromise customer needs and requirements. Increased commonality allows the company to share resources across products, decrease inventory and take advantage of economies of scale to reduce procurement costs [8]. However, if the products are too common, then they can lose their distinctiveness [9]. In a customer-driven and highly competitive marketplace a company must effectively balance customer preferences against its profitability and economic stability. Thus the challenge is to meet the individual customer's wants and needs while keeping overall costs low.

Developing product platforms and designing families of products based on these platforms is one way to address the challenge associated with sharing assets across the products [7]. Product family design involves concurrent design of multiple products that share common features, components and subsystems based on a common product platform [10]. Optimising the design of product families is the key to resolving the trade-off between the conflicting objectives of commonality and individual product performance. A successful design of a product family maximises the commonality as much as possible without sacrificing the distinctiveness of the individual products in the family.

Many researchers have focussed on multi-objective optimisation approaches for balancing the conflicting objectives of commonality and performance. Simpson [11] reviews and categorises over 40 such approaches. For instance, Nelson et al. [12]

use multi-objective optimisation to analyse the Pareto sets of two derivative products to find a suitable product platform for a family of nail guns. Fellini et al. [13, 14] use a similar approach to study a family of three automobiles with varying levels of commonality in the powertrain. Fujita et al. [15] perform a similar analysis for a family of two aircrafts. Fellini et al. [13, 14] introduce a shared penalty vector and performance loss constraints to study the Pareto sets of automotive bodies. Gonzalez-Zugasti et al. [16] use real options concepts to help select the most appropriate product family design from a set of alternatives; they also investigated the use of multi-objective optimisation to design modular product platforms [17, 18]. Allada and Rai [19] introduce an agent-based multi-objective optimisation framework to capture the Pareto frontier for module-based product families of power screwdrivers and electric knives. Simpson et al. [20] examine the trade-off between different levels of platform commonality within a family of three aircraft. Tseng and Jiao [21] use optimisation techniques to facilitate design for mass customisation, and Chidambaram and Agogino [22] present a catalogue-based optimisation strategy for customising goods. Finally, Nayak et al. [23] and Messac et al. [24] have proposed methods for using commonality indices as part of multi-objective optimisation for product family design.

Resolving the commonality–performance trade-off inherent in product family design yields a set of efficient or Pareto solutions where each solution is better than the other solutions in at least one other objective. Based on the size of the product family and number of decision variables, single-stage and multi-stage optimisation approaches exist to help determine the best design variable settings for the product family and individual variants within the family [11]. Single-stage approaches optimise the product platform and the family simultaneously whereas multi-stage approaches initially optimise the product platform followed by optimisation of the individual products in the family [25]. Single-stage approaches have been shown to yield the best overall performance for product family design problems [26]; however, the high dimensionality of single-stage optimisation problems poses computational challenges to many traditional methods. The curse of dimensionality and limitations of traditional methods have motivated researchers to approach these problems with multi-objective evolutionary algorithms (MOEAs). MOEAs evolve solutions through a process analogous to Darwinian selection [27] with search operators that mimic selection, mating and mutation. Over the past few decades, evolutionary algorithms have been extensively used to address a broad range of single- and multi-objective problems. MOEAs have been shown capable of approximating solution sets that compose the trade-offs for highly nonlinear, discrete and non-convex objective space landscapes [28–30].

This chapter presents a MOEA-based many-objective analysis of product family design to help resolve the trade-off between commonality and individual product performance in a product family. In this chapter the phrase “many-objective” refers generally to problems with four or more objectives and is an area of increasing interest in a range of applications [31, 32]. We also demonstrate the benefits of visual-analytic techniques [31–33] to analyse the high-dimensional trade-offs evolved by MOEAs and guide designers in identifying the best possible

compromise solution based on their needs. In short, this chapter introduces a decision-making method for product family design based on many-objective MOEA search and visual analytics. Section 4.2 presents an overview of the process and introduces an example involving the design of a family of aircraft. We describe the problem parameters and state the problem formulation and constraints used for optimisation. Section 4.3 provides a brief description of the MOEA used in the study. Section 4.4 provides a detailed description of the computational experiment required to evolve high quality approximate solutions from the algorithm. Section 4.5 discusses the results and describes the use of visual analytics in the decision-making process. Finally, Sect. 4.6 provides key findings and recommendations for future work.

## 4.2 Method for Product Family Optimisation

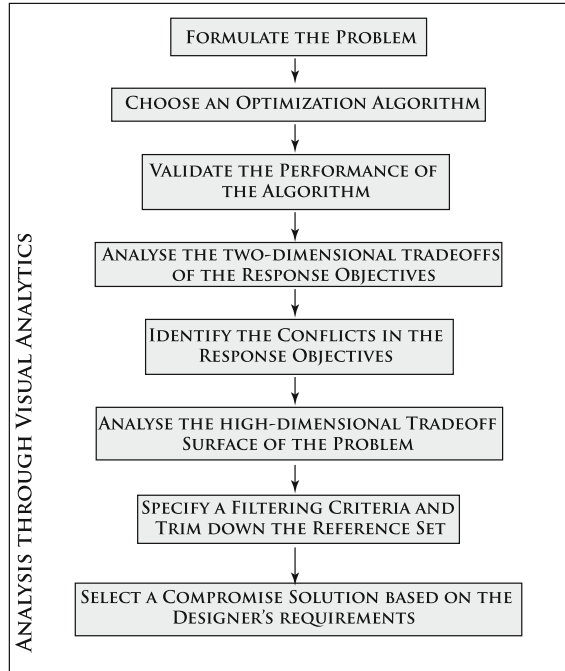
### 4.2.1 Overview

As discussed earlier, performance and commonality are inherently conflicting objectives during the process of developing product family design. Many-objective optimisation provides a mechanism for discovering high-dimensional multivariate dependencies between performance objectives and exploiting these dependencies in negotiated trade-offs. This method requires both the identification of many-objective Pareto approximate solutions and interactive visual exploration. This chapter uses a many-objective product family design problem to demonstrate how designers can navigate the high-dimensional trade-off surfaces and make well-informed decisions.

Figure 4.1 provides an overview of our method which seeks to facilitate design insights and negotiated solution selection. Overall our many-objective visual design analytics work bridges the historical work in joint cognitive systems [34] and visual analytics [35]. Our framework in Fig. 4.1 approaches many-objective design as a “top-down sense-making” exercise [36, 37] by progressively increasing the complexity of trade-off representations presented to decision-makers while utilising solution filtering to focus their attention on key design discoveries and potential compromises. Initially, the problem formulation should be carefully constructed to reflect the key decisions and performance measures that will strongly shape designer preferences and potential conflicts. The formulation should be flexible enough such that if the designer’s requirements change then a new design can be created without reformulating and resolving the problem. Problem formulation dictates subsequent analysis, and thus considerable time should be spent to ensure its correctness and appropriateness for the product family at hand.

The next step after the problem formulation is the selection of an optimisation algorithm. The designer has to identify an optimisation algorithm that would provide a sufficient approximation for a given problem’s Pareto optimal set. It should be ensured that the results from the heuristic are repeatable and the results

**Fig. 4.1** Overview of suggested steps for exploiting many-objective optimisation and visualisation for improving product family designs



should be validated to measure the performance of the algorithm. The next stage is the most critical part of the method: analyses of the solution set. Decisions made after the analyses of the solution set are transformed into products. Success or failure of the product family can have a significant impact on the reputation and profitability of the company. Thus extreme care and diligence should be taken to fully exploit the information captured in a many-objective solution set to guide design decisions.

Extracting meaningful and relevant information from a high-dimensional solution set is a challenging task. The visual analytics method discussed in this chapter presents an organised and structured approach to filter out the relevant and useful information from the solution set. It gradually guides the designer from simple two-dimensional tradeoffs to high-resolution high-dimensional trade-off surfaces. It informs the designer on how each objective contributes to the problem and thus guides the designer to select his/her solution filtering criteria. The designer eventually thins out the solution set and selects the best compromise solution that fits his/her requirements. An important aspect of the method is that none of the decisions require perfect *a priori* knowledge of design preferences, variable interactions and constraints. These problem properties emerge with exploration of the solution set, which itself has the potential to reshape problem conceptions, expert decision rules/heuristics and designer preferences.

The rest of the chapter walks the reader through the method with the help of a real-world product family design problem. It highlights the challenges encountered

during the problem formulation and the corresponding analyses. The example also shows how the proposed method can aid designers in discovering product family designs that offer the best compromises between their objectives.

### ***4.2.2 Test Case Development***

The General Aviation Aircraft (GAA) problem was introduced by Simpson et al. [38] as an example problem focussing on the design of a family of aircraft based on three different seating configurations. The term “General Aviation” encompasses all flights except military operations and commercial carriers. Potential buyers form a diverse group that includes weekend and recreational pilots, training pilots and instructors, travelling business executives and even small commercial operators. Satisfying a group with such diverse needs and economic potential poses a constant challenge for the General Aviation industry because a single aircraft cannot meet all of the market needs. Hence, the example seeks to design a family of three aircraft to accommodate two, four, or six people that can easily be adapted to satisfy distinct groups of customer demands. For this example, the configuration for the GAA is a fixed-wing, single engine, single-pilot, propeller-driven aircraft. The challenge is to determine the best values of top-level design specifications for the fuselage, wing and engine to satisfy a variety of performance and economic requirements. The problem parameters are described in the next section, which provides a detailed discussion of the problem formulation used in the study.

### ***4.2.3 Problem Parameters***

For this example the baseline configuration has been derived from a Beechcraft Bonanza B36TC, a four-to-six seat, single-engine business-and-utility aircraft, which is one of the most popular GAA sold. The general aircraft configuration has been fixed at three propeller blades, high wing position and retractable landing gear based on prior studies [38]. The design variables used in this study and the corresponding ranges of interest are mentioned in Table 4.1.

The General Aviation Synthesis Program (GASP) [39] is used to determine the aircraft sizing and performance estimates. Input variables for GASP are general descriptors of aircraft type, size and missing requirements. The numerical output from GASP includes various performance characteristics of aircraft such as empty weight (WEMP), fuel weight (WFUEL), direct operating cost (DOC) and maximum flight range (RANGE). To reduce the computational expense of performing these calculations, statistical approximations (i.e., response surface models) are employed to provide simplified, yet accurate, approximations of each performance parameter as a function of the input design variables [38].

**Table 4.1** Design parameters and their respective ranges

S. No.	Design variable	Name	Units	Min	Max
1	Cruise speed	CSPD	Mach	0.24	0.48
2	Aspect ratio	AR	–	7	11
3	Sweep angle	SWEEP	degrees	0	6
4	Propeller diameter	DPROP	ft	5.5	5.968
5	Wing loading	WINGLD	lb/ft <sup>2</sup>	19	25
6	Engine activity factor	AF	–	85	110
7	Seat width	SEATW	inch	14	20
8	Tail length/diameter ratio	ELODT	–	3	3.75
9	Taper ratio	TAPER	–	0.46	1

**Table 4.2** Constraints and preferences for the performance parameters

S. No.	Performance parameters	Name	Units	Preference	Performance limits		
					2-seater	4-seater	6-seater
1	Takeoff noise	NOISE	dB	Min	75	75	75
2	Empty weight	WEMP	lb	Min	2200	2200	2200
3	Direct operating cost	DOC	\$/h	Min	80	80	80
4	Ride roughness	ROUGH	–	Min	2	2	2
5	Fuel weight	WFUEL	lb	Min	450	475	500
6	Purchase price	PURCH	1970\$	Min	–	–	–
7	Flight range	RANGE	nm	Max	2000	2000	2000
8	Max life/drag ratio	LDMAX	–	Max	–	–	–
9	Max cruise speed	VCMAX	kts	Max	–	–	–

There are a total of nine responses that are of interest for each aircraft: takeoff noise (NOISE), DOC, ride roughness (ROUGH), WEMP, WFUEL, purchase price (PURCH), maximum cruise speed (VCMAX), RANGE and lift/drag ratio (LDMAX). The constraint values and the min/max preferences for the performance variables are summarised in Table 4.2. Overall a product family design that satisfies the constraints; minimises the NOISE, WEMP, DOC, ROUGH, WFUEL, PURCH and maximises the RANGE, LDMAX and VCMAX is preferred.

#### 4.2.4 Objective Formulation and Constraints

The problem was first solved using robust design methods embodied in the Robust Concept Exploration Method [38, 40]. Product variety trade-off studies were later performed using the compromise Decision Support Problem (DSP) for the family of aircraft [20]. In the prior work of the GAA problem, the values of the response variables were consolidated into one function to reflect the product-performance. A deviation function was adapted from goal programming to measure product performance, with lower deviations being preferred [41]. This approach requires

the designer to specify target values for the response variables *a priori* to optimisation. With very little information available about the problem's objective space, specifying reasonable target values might be difficult, and the specified target values might not reflect the true requirements of the designer.

This chapter presents a novel approach to the problem formulation. Ideally, a designer would like to have the knowledge of the interaction between the various performance parameters and their respective contributions towards commonality and overall product performance. This information would enable the designer to design and introduce products that cater to specific performance parameters without sacrificing the commonality across the families. For instance, DOC (a response variable) might have a greater contribution to the market success of an aircraft as compared with the contributions of other response variables. Thus, a designer might seek to balance DOC and commonality to introduce products that have both economic and market viability. In other words, a designer would like to balance each of the performance parameter independently with the commonality objective to design products based on their requirements. However, such an approach requires solving a high-dimensional and far more complicated problem which may be too computationally expensive or intractable. Toward that end, this chapter introduces a novel approach to the problem formulation which seeks to resolve the trade-off between commonality and individual performance parameter using visual analytics.

During the optimisation process, we seek to find values of the design variables that optimise the performance parameters while maintaining high commonality. Ideally, one would like to optimise the performance parameters for each of the three (2-seater, 4-seater and 6-seater) aircraft in the family. With nine performance parameters per aircraft and three aircraft per family, it would lead to optimising 27 ( $= 9 \times 3$ ) objectives in addition to the objective of maximising commonality within the family. It can be immensely challenging and overwhelming to analyse trade-offs for such high-dimensional problem; thus, we adopt a min-max/max-min optimisation approach and try to optimise the worst-case performance measure across the three product families. If a performance parameter has to be minimised on all the three product families, then a min-max criterion aims at constructing solutions that minimise the maximum performance value across the three aircraft. This formulation minimises the maximum deviation and ensures the best possible performance in the worst case. Similar explanation holds true for the max-min optimisation approach on the response metric that has to be maximised. A significant advantage of the min-max/max-min optimisation approach is that it ensures that variation of design variables does not degrade the performance of a specific aircraft significantly. Another advantage in terms of problem formulation is that the problem reduces from 27 independent performance objectives to nine robust performance objectives. A 10-objective problem (nine robust performance objectives plus the commonality objective) is relatively more tractable and easier to solve as compared with the original 28-objective problem.

To measure the commonality across the product families we use the Product Family Penalty Function (PFPF) developed by Messac et al. [24]. PFPF penalises the uniqueness within the product family by measuring the percentage variation of

**Table 4.3** Objectives used in the General Aviation aircraft formulation

S. No.	Objectives	Value	Preference
1	Maximum NOISE	Max (NOISE <sub>2</sub> , NOISE <sub>4</sub> , NOISE <sub>6</sub> )	Minimise
2	Maximum WEMP	Max (WEMP <sub>2</sub> , WEMP <sub>4</sub> , WEMP <sub>6</sub> )	Minimise
3	Maximum DOC	Max (DOC <sub>2</sub> , DOC <sub>4</sub> , DOC <sub>6</sub> )	Minimise
4	Maximum ROUGH	Max (ROUGH <sub>2</sub> , ROUGH <sub>4</sub> , ROUGH <sub>6</sub> )	Minimise
5	Maximum WFUEL	Max (WFUEL <sub>2</sub> , WFUEL <sub>4</sub> , WFUEL <sub>6</sub> )	Minimise
6	Maximum PURCH	Max (PURCH <sub>2</sub> , PURCH <sub>4</sub> , PURCH <sub>6</sub> )	Minimise
7	Minimum RANGE	Min (RANGE <sub>2</sub> , RANGE <sub>4</sub> , RANGE <sub>6</sub> )	Maximise
8	Minimum max LDMAX	Min (LDMAX <sub>2</sub> , LDMAX <sub>4</sub> , LDMAX <sub>6</sub> )	Maximise
9	Minimum max VCMAX	Min (VCMAX <sub>2</sub> , VCMAX <sub>4</sub> , VCMAX <sub>6</sub> )	Maximise
10	PFPF	–	Minimise

the design variables within the product family. The percentage variation of design variables is measured as follows:

$$\text{pvar}_j = \frac{\text{var}_j}{\bar{x}_j} \quad (4.1)$$

where,

$$\text{var}_j = \sqrt{\frac{\sum_{(i=1)}^p (x_{ij} - \bar{x}_j)^2}{(p-1)}} \quad \text{and} \quad \bar{x}_j = \frac{\sum_{(i=1)}^p x_{ij}}{p} \quad (4.2)$$

$x_{ij}$  is the value of the  $j$ th design variable for the  $i$ th product,  $i = 1, 2, \dots, p$  and  $j = 1, 2, \dots, n$ . PFPF is computed by summing the percentage variations of all  $n$  design variables across all  $p$  products:

$$\text{PFPF} = \sum_{j=1}^n \text{pvar}_j \quad (4.3)$$

Unlike many commonality indices available in the literature [42–44], the PFPF compares commonality not only on how many variables are common but also on how similar the values of the unique variables are to one other (i.e., parametric variation). Product families with high variation in design parameters (i.e., distinct inputs) have higher values of PFPF while product families with low variation in design parameters (i.e., common inputs) have lower values of PFPF. As high commonality is desired, a lower PFPF value is preferred. A summary of the problem objectives used in this study is given in Table 4.3.

As mentioned in Table 4.2, each aircraft is associated with certain performance limits. These are rigid constraints that establish the feasibility of each product. The violation of a constraint can be measured as follows:

$$c_{\text{in}} = \begin{cases} \frac{(\text{value} - \text{limit})}{\text{limit}}, & \text{if value} > \text{limit} \\ 0, & \text{if value} \leq \text{limit} \end{cases}, \quad n \in (1, 2, 3), \quad i \in (1, 2, \dots, 9) \quad (4.4)$$



The total constraint violation for the product family is computed by summing the violation of each constraint for each aircraft.

$$CV = \sum_{i=1}^3 (c_{1i} + c_{2i} + c_{3i} + c_{4i} + c_{5i} + c_{6i}) \quad (4.5)$$

In summary, the size of the product family optimisation problem in this study is: 27 design variables, 10-objectives and 1 constraint ( $CV < 0$ ).

### 4.3 Optimisation Algorithm Selection

The Epsilon-dominance Nondominated Sorted Genetic Algorithm-II ( $\epsilon$ -NSGAI) is based on the NSGAI [45] an elitist MOEA. NSGAI uses a non-domination sorting approach to classify solutions according to the level of non-domination and a crowding distance operator to maintain solution diversity across approximation solution sets. The  $\epsilon$ -NSGAI developed by Kollat and Reed [46, 47] reduces the extensive parameter calibration by using the concepts of  $\epsilon$ -dominance archiving [48, 49], adaptive population sizing [50] and self-termination. The  $\epsilon$ -NSGAI has been validated extensively across a suite of test problems and applications [46] and has been shown to perform as well or better than state-of-the-art MOEAs [47, 51].

The  $\epsilon$ -NSGAI algorithm generates an initial small random population and uses non-domination and crowding distance to assign fitness to each individual. A non-domination sort is performed across all the solutions, and individuals are classified into fronts based on their ranks, with rank 1 assigned to the solutions that are non-dominated. Additionally, crowding distance is calculated for all individuals based on the average Euclidean distance between an individual and the individuals within the population which are assigned the same rank. Selection is done using binary tournaments and is based on the rankings and crowding distances of the individuals with a preference given to larger crowding distance. Individuals with larger crowding distance add to the diversity of the population and help to ensure that the  $\epsilon$ -NSGAI explores the entire trade-off landscape. Selected individuals now become parents of the next generation, and the evolution process is repeated. These individuals are also eligible to enter an offline archive that stores the best solutions throughout the run. To achieve entry into the archive, individuals should be  $\epsilon$ -non-dominated with respect to solutions in the archive.

The  $\epsilon$ -NSGAI thereafter uses a series of “connected runs” to inject the archive solutions into the population of the next run using a 25% injection scheme. The injection scheme requires that the present archive forms 25% of the next population and the remaining 75% is filled with randomly generated individuals. This assists the performance of the  $\epsilon$ -NSGAI by directing the search towards previously known good solutions; however the 75% random solutions help to ensure that the algorithm does not pre-converge while encouraging the exploration of new

regions in the objective space. The algorithm can increase or decrease its population size as the search progresses and adapts its population based on the solutions obtained.

The  $\varepsilon$ -dominance archive allows the user to control the computational costs of evolution by specifying their precision requirements for each of the objectives. Based on the user's preferences, the algorithm applies a grid to the search space of the problem that can significantly reduce its computational costs when solving multi-objective problems by avoiding unnecessary precision in calculations [52]. Larger  $\varepsilon$  values result in a coarser grid (and ultimately fewer trade-off solutions) while smaller  $\varepsilon$  values produce a finer grid. The fitness of each solution is then mapped to a box fitness based on the specified  $\varepsilon$  values. Non-domination sorting is then conducted using each solution's box fitness, and solutions with identical box fitness (i.e., solutions that occur in the same grid block) are compared, and those that are dominated within the grid block are eliminated. Only a single non-dominated solution is permitted in any one grid block, preventing clustering of solutions and promoting a more diverse search of the objective space. We refer the reader to Laumanns et al. [48] and Deb et al. [49] for additional details. Meanwhile, dynamic population sizing allows  $\varepsilon$ -NSGAI to start with a small initial population to pre-condition the search at a low computational cost in terms of the number of function evaluations. When the size of the  $\varepsilon$ -dominated archive stabilises, the connected runs are equivalent to a diversity-based EA search enhancement recommended by Goldberg [53] termed *time continuation*, where diverse search is sustained as long as it is required or feasible. Prior work using the  $\varepsilon$ -NSGAI by Kollat and Reed [46, 47] can be referenced for more details on the algorithm and its dynamic search features.

## 4.4 Computational Experiment

The  $\varepsilon$ -NSGAI was used to approximate the trade-offs in the family of aircraft and its evolutionary operators were parameterised as follows: probability of cross-over— $pc = 1.0$ , probability of mutation— $pm = 0.04$ , cross-over distribution index— $gc = 15$  and the mutation distribution index— $gm = 20$ . The  $\varepsilon$ -NSGAI's adaptive population sizing was initialised using 152 individuals, and maximum number of function evaluations per trial was set at 500,000. Epsilon resolution settings ( $\varepsilon$ ) for the 10 objectives are given in Table 4.4. These values represent the precision with which each objective is quantified and were chosen to represent the full precision Pareto-optimal set. Since MOEA search is initialised with randomly generated populations and as evolutionary operators are probabilistic, the process can yield high variability in search efficiency and reliability. It is standard practice to overcome this variability by running MOEA for a distribution of “seeds” for the random number generator which is used to initialise and guide their probabilistic search. In this study, our analysis across the 10-objective GAA problem was characterised using 50 random seed trial runs.

**Table 4.4** Epsilon settings and ranges of the objectives

S. No.	Objectives	Name	$\varepsilon$	Range	
				Min	Max
1	Maximum NOISE	MAX_NOISE	0.05	73.25	74.46
2	Maximum WEMP	MAX_WEMP	10	1879.20	2032.91
3	Maximum DOC	MAX_DOC	2	58.67	80.00
4	Maximum ROUGH	MAX_ROUGH	0.01	1.81	2.00
5	Maximum WFUEL	MAX_WFUEL	10	367.87	500.00
6	Maximum PURCH	MAX_PURCH	1000	41901.85	44925.33
7	Minimum RANGE	MAX_RANGE	50	2000.00	2496.87
8	Minimum max LDMAX	MAX_LDMAX	0.1	14.20	16.00
9	Minimum max VCMAx	MAX_VCMAx	1	185.33	200.17
10	PFPF	PFPF	0.1	0.07	2.50

## 4.5 Results and Discussions

This chapter aims to evolve the non-dominated trade-off for a 10-objective problem and demonstrate the value of visual analytics in understanding key trade-offs. Solving such high-dimensional problems is a challenging proposition for most of the domination-based MOEAs. It becomes extremely difficult to effectively parameterise the algorithm and effectively guide the evolution process. Many authors have highlighted that on high-dimensional problems (objectives more than five or six) that some MOEAs may struggle in evolving high quality approximation sets and in some cases devolve into a “random walk” [31, 54–56]. As a test of the value and quality of the  $\varepsilon$ -NSGAIi attained results, we have utilised a Monte-Carlo analysis to establish a pure random search baseline for the GAA problem where any selected trial solution is fully independent of any previous choice and its outcome [30]. If the results of the Monte-Carlo simulations are comparable with those obtained from the optimisation algorithm, then it negates the value of the optimisation algorithm and may also highlight the ease of solving a product family design problem such as this.

To validate the performance of the  $\varepsilon$ -NSGAIi and test the quality of solutions generated by it, the results obtained from the optimisation algorithm were compared against the results obtained from Monte-Carlo simulation. The optimisation algorithm had 50 random trials with 500,000 function evaluations per trial. Thus, the algorithm used a total of 25 million ( $=50 \times 500,000$ ) function evaluations to generate a non-dominated set for the 10-objective GAA problem. The comparison was biased towards the Monte-Carlo simulation as it was allowed to generate 50 million samples (twice the number of function evaluations used by  $\varepsilon$ -NSGAIi) to identify the non-dominated set for the problem. Table 4.5 presents a summary of run results from both approaches.

Comparative analysis of the Monte-Carlo simulation and  $\varepsilon$ -NSGAIi revealed some interesting insights about the objective space of the problem. Of the 50 million random samples generated by the Monte-Carlo simulation study, only four solutions

**Table 4.5** Number of solutions by Monte-Carlo simulation and the optimisation algorithm

Method	Total number of non-dominated solutions generated	Contribution to the reference set
Monte-Carlo simulation	4	0
Optimisation algorithm	16900	16900

were found feasible. The identification of only four feasible solutions from a set of 50 million solutions shows that the GAA problem is heavily constrained with respect to its performance parameters and a challenging overall search space. In the GAA's 27-dimensional decision (input) space it is almost impossible to randomly pick a point that would be feasible in the objective space. On the other hand  $\epsilon$ -NSGAI generated a non-dominated set of 16,900 solutions from the 25 million function evaluations. On each run of the algorithm,  $\epsilon$ -NSGAI found its first feasible solutions after only 1000–2000 function evaluations. The algorithm struggled for brief duration during the onset of a run; however, once it identified a feasible solution it quickly adapted and redirected its search to the favourable region of the objective space to generate more feasible solutions. Non-domination sorting of results from the Monte-Carlo simulation study and the  $\epsilon$ -NSGAI indicated that the four solutions generated by the Monte-Carlo simulation were dominated by the solutions generated by the  $\epsilon$ -NSGAI.

Superior performance of  $\epsilon$ -NSGAI can be attributed to the use of  $\epsilon$ -dominance archiving and adaptive population sizing. The combination of adaptive population sizing and epsilon archiving represents a diversity enhancement that also ensures stable and bounded archiving of high-dimensional approximation sets. As the dimensionality of a problem's objective space increases, generally the size of their Pareto-optimal solution sets grows rapidly yielding an impediment to search that Purshouse and Fleming [54] termed "dominance resistance". Dominance resistance represents the increasing difficulty of converging a high-dimensional set towards Pareto-optimality. In  $\epsilon$ -NSGAI, the population size grows commensurate with the  $\epsilon$ -dominance archive. In this strategy it controls the dominance resistance by setting epsilons [48] and uses archive size as a proxy for problem difficulty that triggers increases in the population size. Increased population sizes serve to both add diversity and selective pressure due to the truncation selection used in the  $\epsilon$ -NSGAI algorithm framework. Moreover,  $\epsilon$ -dominance archiving provides a theoretical bound to the approximation set size and population size [57].

The comparative analyses justified the need of an optimisation algorithm like  $\epsilon$ -NSGAI to solve the problem. Thus a reference set was generated by pooling the non-dominated solutions across the 50 runs of the optimisation algorithm. The reference set consisted of 16,900 solutions. Analysing the high-resolution trade-off solutions on a 10-dimensional objective space can be difficult and overwhelming for a designer. Thus before analysing the high-dimensional trade-off we analyse relatively simple and easier to understand two-dimensional trade-offs of the performance parameters. Two-dimensional trade-offs provide valuable insights about the performance parameters and their mutual interactions and are much simpler

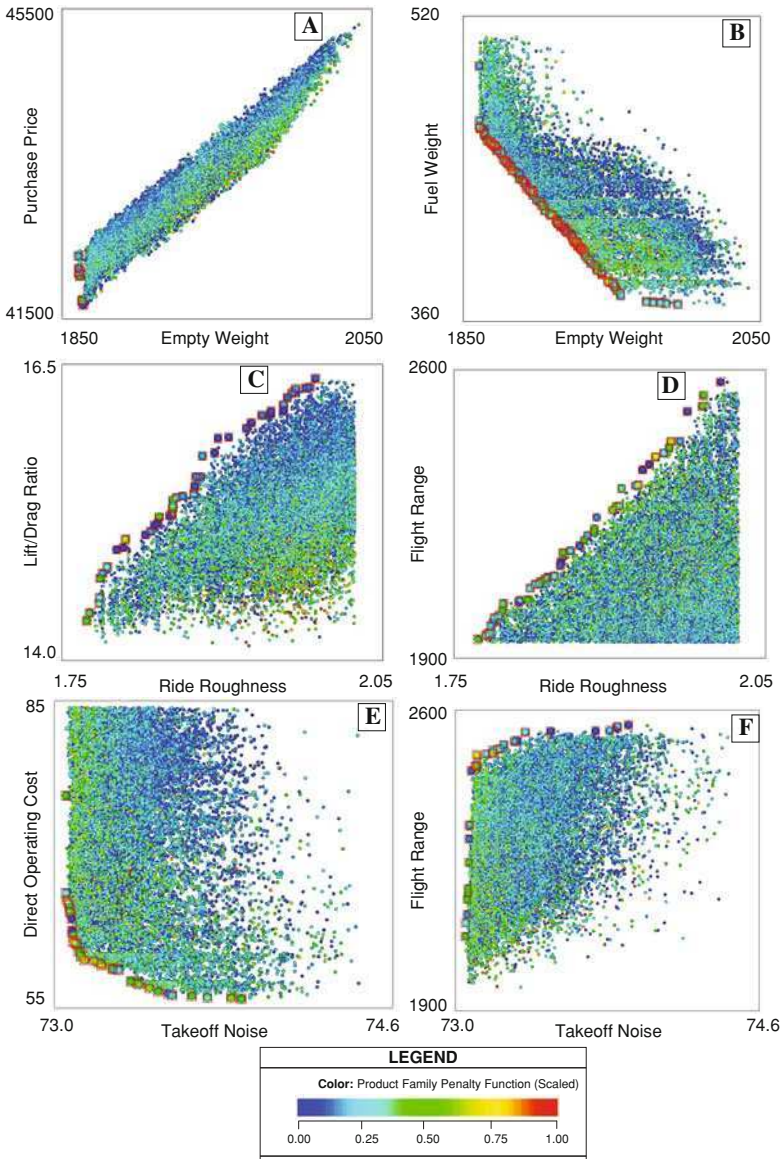
than analysing 10-objectives at a time. Having some prior knowledge about the mutual interactions of performance parameters assists the designer in analysing the high-dimensional trade-off surface by eliminating the redundant information content.

A 10-objective problem yields 45 two-dimensional trade-offs. Figure 4.2 highlights some of the interesting two-dimensional trade-offs identified by the algorithm. The colour in the subplots indicates the performance on the commonality objective. Blue solutions indicate high commonality, and green solutions indicate low commonality. In each subplot the solutions highlighted with a red-coloured outline represent the trade-off for the corresponding set of objectives.

Figure 4.2a shows the interactions between PURCH and WEMP, both of which are to be minimised. The plot clearly indicates that there is a strong positive correlation between the PURCH and WEMP: an increase in WEMP results in an increase in PURCH. As the two objectives are positively correlated the trade-off solution set for this sub-problem essentially reduces to one solution. Figure 4.2b represents the interactions between the WFUEL and WEMP, where both the objectives are to be minimised. The plot shows there is a strong negative correlation between the two objectives, indicating a strong conflict between the two objectives. Thus a design with low WEMP results in higher WFUEL and vice versa. Figure 4.2c and d represent the interactions between LDMAX and ROUGH, and RANGE and ROUGH, respectively. RANGE and LDMAX are to be maximised and ROUGH has to be minimised. The plots indicate that as the RANGE and LDMAX increases, there is proportional increase in ROUGH. Figure 4.2e and f represent the interactions between the DOC and NOISE, and RANGE and NOISE, respectively. DOC and NOISE are to be minimised and VCMAX is to be maximised. An interesting aspect of the trade-off seen here is that there is a steep drop in DOC (and a steep rise in VCMAX) for a relatively small increase in NOISE. However, beyond a threshold (65 for DOC and 198 for VCMAX) any further decrease in DOC (or increase in VCMAX) requires a significant increase in NOISE.

While Fig. 4.2 presents only a small subset of the 45 two-dimensional plots, it presents valuable information to the designer. It highlights the facts that while the designer is optimising WEMP, the PURCH is also being optimised; meanwhile, the designer cannot optimise WFUEL and WEMP at the same time—he/she will need to prioritise one over the other. Furthermore, the designer cannot target extreme performances for DOC and VCMAX as they might result in unacceptable values for NOISE which is a constraint. The rest of the two-dimensional plots can be analysed to extract further information about the performance parameters and their behaviour. In summary, a few other strong relationships observed in other plots were: (1) decreases in WEMP results in increases in VCMAX and (2) low PURCH results in high WFUEL. Having some prior information about the interaction between the performance objectives the designer is better placed to analyse the complete reference set.

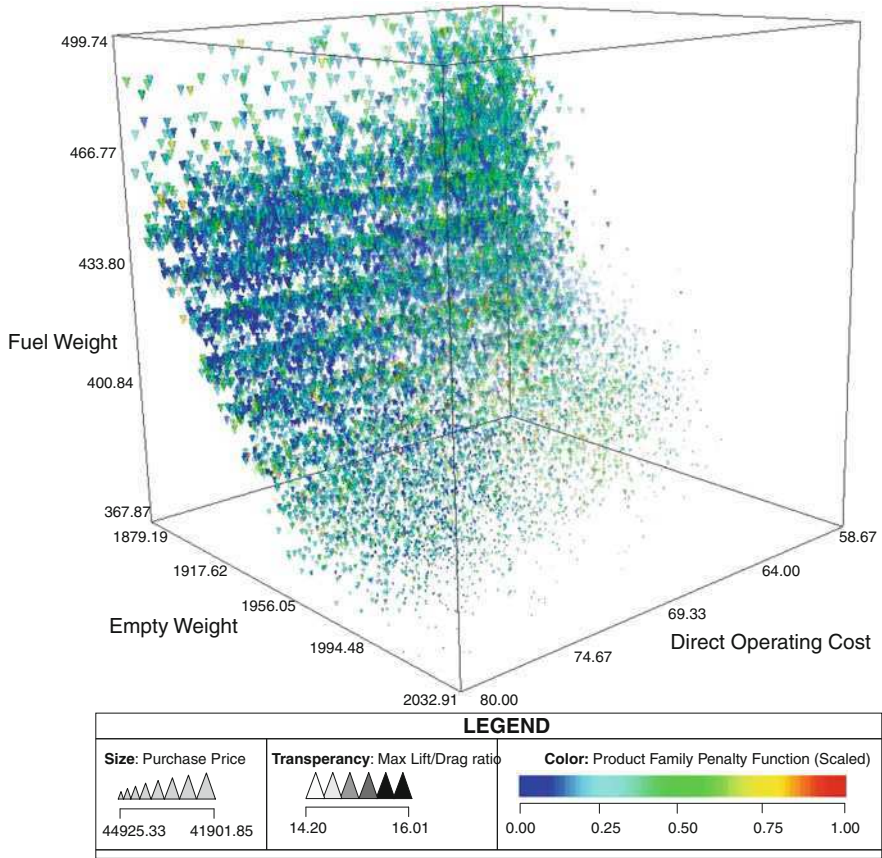
The two-dimensional analysis in Fig. 4.2 informs the designer about the strong trade-offs for the problem, and this information can be used to when visualising the



**Fig. 4.2** Scatter plot analysis of the two objective interactions as subspaces of the overall 10-objective formulation

higher-dimensional reference set. Figure 4.3 represents the reference set for the 10-objective GAA problem. It uses the information captured in Fig. 4.2 to organise the response objectives into corresponding axes such that it highlights the conflicts at the higher-dimension. The WEMP, DOC and WFUEL objectives have been





**Fig. 4.3** Reference set

plotted on X-, Y- and Z-axes respectively. PURCH is represented by the size of the cones, which is scaled so that smaller cones represent higher PURCH while larger cones represent lower PURCH. Transparency is used to represent the LDMAX with lighter cones representing lower values of LDMAX and darker cones representing higher values of LDMAX. The colours of the cones represent the scaled values of the PFPF. Blue cones represent low PFPF values (high commonality) and red cones represent high PFPF values (low commonality). The values of PFPF range from 0.065 to 2.5.

Objective values for this non-dominated set range from 367.87 to 500lbs on WFUEL, 1879–2032lbs on WEMP, 58–80\$/h on DOC, 41901.85–44925.33 (1970\$) on PURCH, 14.20–16 on LDMAX and 0.065–2.5 on PFPF. For the ease of understanding and analysis the PFPF values have been scaled to vary between 0 and 1, with 0 representing high commonality and 1 representing no commonality. PFPF values have been scaled using the following equation:

$$PFPF_{Scaled} = \frac{PFPF - 0.065}{2.5 - 0.065} \quad (4.6)$$

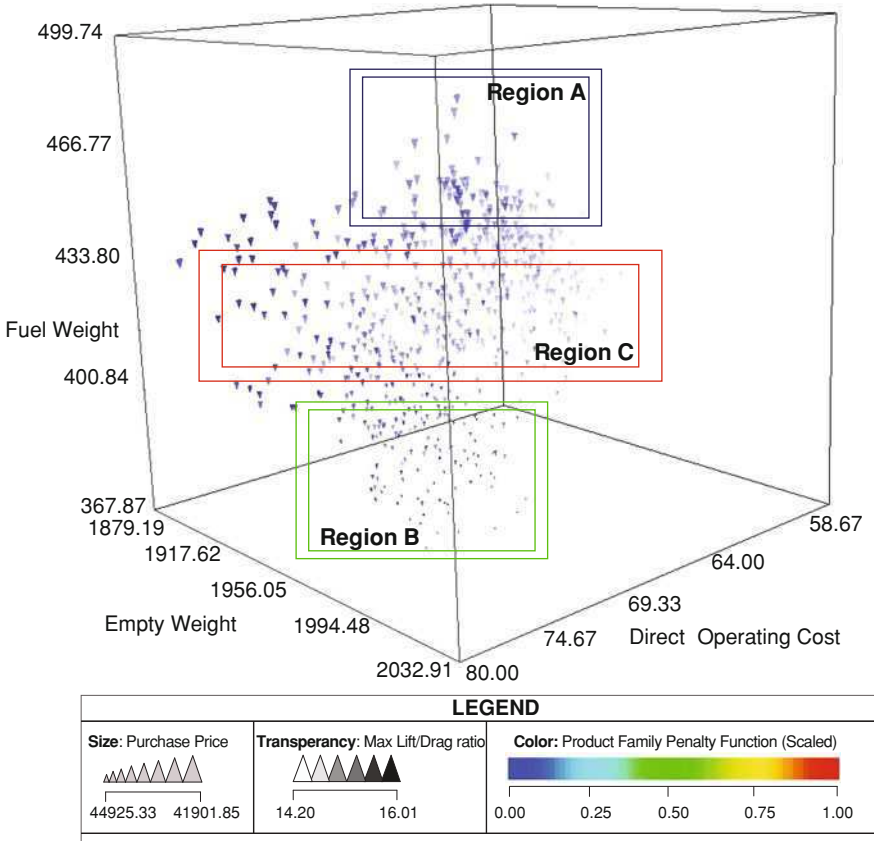
WFUEL, WEMP, DOC, PURCH and PFPF (scaled) are to be minimised whereas LDMAX has to be maximised. As discussed earlier, the figure shows a clear conflict between the WFUEL and WEMP. LDMAX is in conflict with DOC as the solutions on the farther end of DOC axis yield more favourable performance on LDMAX. To optimise commonality in the family, a designer should focus on dark, large, blue solutions.

Ideally, a designer would like to optimise the product performance while maintaining high commonality (low PFPF). In other words, the designer's focus would be to maximise the commonality (low PFPF) and then subsequently get the best performance measure available from the performance parameters. Thus, we would like to concentrate only on the solutions with high commonality and eliminate the low commonality solutions.

Figure 4.4 filters out the low commonality from the set and displays only the top 5% ( $PFPF_{Scaled}$  values from 0 to 0.05) of the high commonality solutions. The figure clearly shows there are two conflicting regions of the objective space: (1) Regions A and B, and (2) Region C. Region A (marked in blue box) offers high WFUEL, low PURCH, low WEMP, low DOC and low LDMAX. The PFPF values for the solutions in Region A are in the range of 0.04–0.05. Region B (marked in green box) offers low WFUEL, high WEMP, high PURCH, high DOC and high LDMAX. The PFPF values for the solutions in Region B are in the range of 0.04–0.05. Thus, Regions A and B offer designers a few conflicting design options while maintaining relatively high commonality. Based on his/her priorities she/he can focus on a specific region of the objective space. However, if a designer is interested in extremely high commonality ( $PFPF_{Scaled}$  values from 0–0.01) there is a small trade-off region available in terms of Region C. Region C offers solutions that compromise on the values of the performance metrics to yield extremely low PFPF values. Thus if a designer is willing to sacrifice performance and is willing to accept a compromise value then s/he can design products with extremely high commonality.

Figure 4.5 shows solutions highlighted in Fig. 4.4 on a parallel coordinate plot. Parallel coordinate plots help visualise the performance across many-objectives simultaneously [58]. The vertical axes represent the individual objectives. The lower and higher end values of the vertical axes represent the ranges of the objective values. Each coloured line represents a solution with its intersection point on the vertical axes representing the performance on the corresponding objective. Colour coding of the boxes in Fig. 4.4 correspond to the colour coding of solutions in Fig. 4.5. Blue solutions represent solutions from Region A in Fig. 4.4, green solutions represent the solutions from Region B and red solutions represent the solutions from Region C. As discussed earlier, the blue set performs favourably on WEMP, DOC, PURCH, RANGE, VCMA and performs fairly well on PFPF. The green set on the other hand performs favourably on WFUEL, LDMAX and shows similar performance as the blue set on the PFPF objective. The red set offers the compromise performance values on most of the objectives and the best performance value for ROUGH and PFPF. The





**Fig. 4.4** Brushed reference set

dotted solutions represent the best possible compromise in their respective region. The blue dotted line represents the best possible compromise solution among all the solutions of Region A. It represents the best performance for DOC, ROUGH, RANGE, LDMAX and VCMAX while maintaining a fairly decent level of commonality. The green dotted line represents the compromise solution in the Region B with favourable performances on WFUEL, PURCH and LDMAX. Finally, the red dotted line represents a high commonality (3 inputs common across all three aircraft) solution with acceptable compromise values on the rest of the performance objectives.

Depending on the market demands and economic viability the designer can select a suitable product family design from either of the regions. This visual analytics-based approach gives the designer the flexibility in terms of focussing on specific performance metrics while maintaining high commonality. Thus a designer can create dedicated and customised products for various market segments without compromising on the commonality of the products.

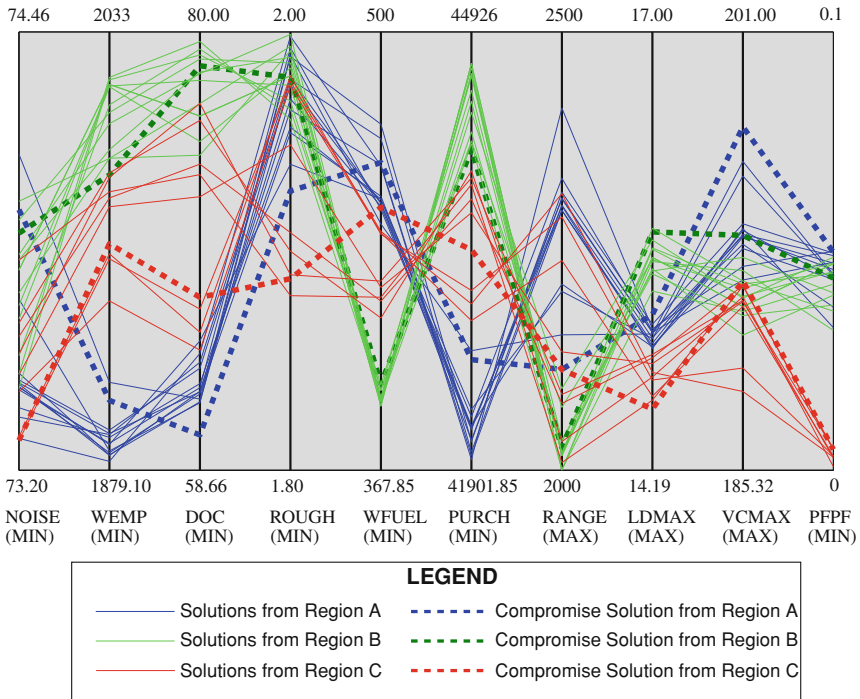


Fig. 4.5 Parallel coordinate plot for solutions displayed in Fig. 4.4

### 4.6 Conclusions

Multi-objective optimisation provides a useful tool for resolving the trade-offs between commonality and individual product performance (i.e., distinctiveness) within a family of products. This chapter presents a novel method to the product family optimisation based on MOEA and visual analytics. It uses a General Aviation Aircraft (GAA) example to demonstrate the relative merits of the proposed method to optimise a family of products for specific market needs without sacrificing commonality across the family. It introduces a 10-objective robust problem formulation where each objective represents a different performance parameter in the family. This formulation expands the dimensionality of the problem and seeks to resolve the trade-off between commonality and individual performance parameters. MOEAs are known to struggle on such high-dimensional problems, and selecting an algorithm that effectively solves the problem was a challenge. The  $\epsilon$ -NSGAII has been shown to perform reasonably well on high-dimensional real-world problems. It uses the concepts of  $\epsilon$ -dominance archiving and adaptive population sizing to balance the dominance resistance associated with high-dimensional problems. Thus,  $\epsilon$ -NSGAII was used to resolve the trade-off for the 10-objective problem and the results were benchmarked against a Monte-Carlo

simulation. The results indicate the GAA is a highly constrained problem, thus making it virtually impossible to generate a feasible solution through random search. The  $\epsilon$ -NSGAI on the other hand performs well, navigating the search space to identify feasible solutions.

The proposed method integrates the use of visual analytical-techniques to gain insight into the high-dimensional trade-off surfaces generated by the algorithm. We illustrate some of the tools that are available for designers to identify strong conflicts between the performance parameters. These tools inform designers about the trade-offs for the problem, and this information is further used to effectively visualise the higher-dimensional reference set. The method allows designers to reduce the full-resolution set to a tractable set focussing on the most relevant and useful information. Aware of the interactions between the performance parameters, the designer can select the best compromise solution from the reduced set to satisfy his/her requirements. The key aspect of this method is that it does not require *a priori* knowledge of the problem, and it provides designers with a plethora of solutions from which to choose. Designers can enjoy the flexibility of creating a wide variety of products customised to different market segments without re-solving the problem.

The proposed method provides an efficient way to analyse the high-dimensional trade-offs for many-objective problems. The method explains through an example, on how best compromise solutions can be identified from a high-resolution high-dimensional trade-off surface. Future work can be based on improving the method by including the designer's requirements as an integral part of the method while allowing for interactivity between designers and the optimisation algorithms as solutions evolve. This also includes developing effective strategies for dealing with high-dimensional input spaces and handling product family problems with many product variants. Finally, extending the visual analytics techniques to help identify platform variables within a product family would be beneficial to designers.

**Acknowledgments** The first and second authors were partially supported by the National Science Foundation (NSF) under CAREER Grant No. CBET-0640443, and the third author acknowledges support from NSF Grant No. CMMI-0620948. The computational experiments in this work were supported in part through instrumentation funded by NSF Grant No. OCI-0821527. Any opinions, findings and conclusions or recommendations in this chapter are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

1. Anderson, D.M. (1997). *Agile product development for mass customization: How to develop and deliver products for mass customization, Niche Markets, JIT, Build-to-Order and Flexible Manufacturing*. Chicago, IL: Irwin.
2. Galsworth, G. D. (1994). *Smart, simple design: Using variety effectiveness to reduce total cost and maximize customer selection*. Essex Junction, VT: Omneo.
3. Ho, T. H., & Tang, C. S. (1998). *Product variety management: Research advances*. Boston, MA: Kluwer Academic Publishers.

4. Child, P., Diederichs, R., Sanders, F.-H., & Wisniowski, S. (1991). The management of complexity. *Sloan Management Review*, 33(1), 73–80.
5. Ishii, K., Juengel, C., & Eubanks, C.F. (1995). Design for product variety: Key to product line structuring. *Proceedings of the ASME Design Engineering Technical Conferences—Design Theory and Methodology* (pp. 499–506). Boston, MA 83(2)
6. Lancaster, K. (1990). The economics of product variety. *Marketing Science*, 9(3), 189–206.
7. Meyer, M. H., & Lehnerd, A. P. (1997). *The power of product platforms: Building value and cost leadership*, Free Press. NY: New York.
8. Thevenot, H. J., & Simpson, T. W. (2007). A comprehensive metric for evaluating commonality in a product family. *Journal of Engineering Design*, 18(6), 577–598.
9. Robertson, D., & Ulrich, K. (1998). Planning product platforms. *Sloan Management Review*, 39(4), 19–31.
10. Simpson, T. W., Siddique, Z., & Jiao, J. (Eds.). (2005). *Product platform and product family design: methods and applications*. New York: Springer.
11. Simpson, T. W. (2005). Methods for optimizing product platforms and product families: Overview and classification. In T. W. Simpson, Z. Siddique, & J. Jiao (Eds.), *Product platform and product family design: Methods and applications* (pp. 133–156). New York: Springer.
12. Nelson, S. A., I. I., Parkinson, M. B., & Papalambros, P. Y. (2001). Multicriteria optimization in product platform design. *ASME Journal of Mechanical Design*, 123(2), 199–204.
13. Fellini, R., Kokkolaras, M., Michelena, N., Papalambros, P., Saitou, K., Perez-Duarte, A., & Fenyes, P.A. (2002). A sensitivity-based commonality strategy for family products of mild variation, with application to automotive body structures. *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, AIAA, AIAA-2002-5610.
14. Fellini, R., Kokkolaras, M., Papalambros, P., & Perez-Duarte, A. (2002). Platform selection under performance loss constraints in optimal design of product families. *Proceedings of the ASME Design Engineering Technical Conferences—Design*
15. Fujita, K., Akagi, S., Yoneda, T., & Ishikawa, M. (1998). Simultaneous optimization of product family sharing system structure and configuration. *Proceedings of the ASME Design Engineering Technical Conferences*, Atlanta, GA, ASME, Paper No. DETC98/DFM-5722.
16. Gonzalez-Zugasti, J. P., Otto, K. N., & Baker, J. D. (1999). 12–15 September. *Assessing value for product family design and selection*. Advances in Design Automation, Las Vegas, NV, ASME, Paper No. DETC99/DAC-8613.
17. Gonzalez-Zugasti, J.P., & Otto, K.N. (2000). Modular platform-based product family design. *Proceedings of the ASME Design Engineering Technical Conferences—Design Automation Conference*, Baltimore, MD, ASME, Paper No. DETC-2000/DAC-14238.
18. Gonzalez-Zugasti, J. P., Otto, K. N., & Baker, J. D. (2000). A method for architecting product platforms. *Research in Engineering Design*, 12(2), 61–72.
19. Allada, V., & Rai, R. (2002). *Module-based multiple product design*, IIE Annual Conference 2002, Orlando, FL, IIE.
20. Simpson, T. W., Seepersad, C. C., & Mistree, F. (2001). Balancing commonality and performance within the concurrent design of multiple products in a product family. *Concurrent Engineering: Research and Applications*, 9(3), 177–190.
21. Tseng, M.M., & Jiao, J. (1998). Design for mass customization by developing product family architecture. *ASME Design Engineering Technical Conferences—Design Theory and Methodology*, Atlanta, GA, ASME, Paper No. DETC98/DTM-5717.
22. Chidambaram, B., & Agogino, A. M. (1999). 12–15 September. Catalog-based customization. *Advances in Design Automation*, Las Vegas, NV, ASME, Paper No. DETC99/DAC-8675.
23. Nayak, R. U., Chen, W., & Simpson, T. W. (2002). 10–13 September. A variation-based method for product family design. *Engineering Optimization*, 34(1), 65–81.
24. Messac, A., Martinez, M. P., & Simpson, T. W. (2000). Introduction of a product family penalty function using physical programming. 8th AIAA/NASA/USAF/ISSMO Symposium

- on *Multidisciplinary Analysis and Optimization*, Long Beach, CA, AIAA, AIAA-2000-4838, to appear in ASME Journal of Mechanical Design.
25. Khajavirad, A., Michalak, J. J., & Simpson, T. W. (2009). An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality. *Structural and Multidisciplinary Optimization*, 39(2), 187–201.
  26. Messac, A., Martinez, M. P., & Simpson, T. W. (2002). Effective product family design using physical programming. *Engineering Optimization*, 34(3), 245–261.
  27. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. New York, NY: Addison-Wesley Publishing.
  28. Back, T., Fogel, D., & Michalewicz, Z. (2000). *Handbook of evolutionary computation*. Bristol, UK: Oxford University Press.
  29. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. New York: John Wiley & Sons LTD.
  30. Coello, C. C., Van Veldhuizen, D. A., & Lamont, G. B. (2002). *Evolutionary algorithms for solving multi-objective problems*. New York, NY: Kluwer Academic Publishers.
  31. Fleming, P. J., Purshouse, R. C., & R. J. Lygoe (2005), Many-objective optimization: An engineering design perspective, in *Evolutionary Multi—Criterion Optimization*, ser. Lecture Notes in Computer Science. Springer: Berlin, Heidelberg pp. 14–32.
  32. Kasprzyk, J. R., Reed, P. M., Kirsch, B., & Characklis, G. *Managing population and 761 drought risks using many-objective water portfolio planning under uncertainty*, Water Resources 762 Research, doi:10.1029/2009WR008121.
  33. Kollat, J. B., & Reed, P. M. (2007). A framework for visually interactive decisionmaking and design using evolutionary multiobjective optimization (VIDEO). *Environmental Modelling and Software*, 22(12), 1691–1704.
  34. Woods, D. (1986). “Paradigms for intelligent decision support,” *Intelligent Decision Support in Process Environments*, Springer, New York, NY
  35. Keim, D. A., Mansmann, F., Schneidewind, J., & Ziegler, H. (2006). Challenges in visual data analysis. *Proceedings of Information Visualization, IEEE Computer Society* 9–16, London, UK.
  36. Russell, D. M., Stefik, M. J., Pirolli, P., & Card, S. K. (1993). The cost structure of sensemaking. *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, Amsterdam, The Netherlands, April 24–29.
  37. Qu, Y., & Furnas, G. W. (2005). Sources of structure in sensemaking. *Proceedings of the SIGCHI '05 Conference on Human Factors in Computing Systems*, ASM Press: Portland, OR, April 2–7.
  38. Simpson, T. W., Chen, W., Allen, J. K., & Mistree, F. (1996). 4–6 September. Conceptual design of a family of products through the use of the robust concept exploration method. 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, AIAA, Vol. 2, pp. 1535–1545. AIAA-96-4161-CP.
  39. NASA. (1978). GASP—General aviation synthesis program, NASA CR-152303, Contract NAS 2-9352, NASA Ames Research Center, Moffett Field, CA.
  40. Simpson, T. W., Chen, W., Allen, J. K., & Mistree, F. (1999). Use of the robust concept exploration method to facilitate the design of a family of products. In U. Roy, J. M. Usher, et al. (Eds.), *Simultaneous engineering: Methodologies and applications* (pp. 247–278). Amsterdam, The Netherlands: Gordon and Breach Science Publishers.
  41. Mistree, F., Hughes, O.F., & Bras, B.A. (1993). The compromise decision support problem and the adaptive linear programming algorithm, In: M. P. Kamat (ed.), *Structural optimization: Status and promise*, Washington
  42. Jiao, J., & Tseng, M. M. (2000). Understanding product family for mass customization by developing commonality indices. *Journal of Engineering Design*, 11(3), 225–243.
  43. Kota, S., Sethuraman, K., & Miller, R. (2000). A metric for evaluating design commonality in product families. *ASME Journal of Mechanical Design*, 122(4), 403–410.

44. Siddique, Z., Rosen, D.W., & Wang, N. (1998). On the applicability of product variety design concepts to automotive platform commonality, *Design Theory and Methodology—DTM'98*, Atlanta, GA, ASME, Paper No. DETC98/DTM-5661.
45. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
46. Kollat, J. B., & Reed, P. M. (2005). The value of online adaptive search: A performance comparison of NSGA-II,  $\epsilon$ -NSGAI, and  $\epsilon$ MOEA. In C. C. Coello, A. H. Aguirre, & E. Zitzler (Eds.), *The Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science 3410* (pp. 386–398). Guanajuato, Mexico: Springer.
47. Kollat, J. B., & Reed, P. M. (2006). Comparing state-of-the-art evolutionary multiobjective algorithms for long-term groundwater monitoring design. *Advances in Water Resources*, 29(6), 792–807.
48. Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3), 263–282.
49. Deb, K., Mohan, M., Mishra, S. (2003). A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. Tech. Rep. KanGAL 2003002, Indian Institute of Technology Kanpur.
50. Harik, G. R., Lobo, F. G. (1999). A parameter-less genetic algorithm. Tech. Rep. IlliGAL 99009, University of Illinois at Urbana-Champaign.
51. Tang, Y., Reed, P., Wagener, T. (2006). How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? *Hydrology and earth system sciences* 10 (2).
52. Kollat, J., & Reed, P. (2007). A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. *Advances in Water Resources*, 30(3), 408–419.
53. Goldberg, D. E. (2002). *The design of innovation: lessons from and for competent genetic algorithms*. Norwell, MA: Kluwer Academic Publishers.
54. Purshouse, R. C., & Fleming, P. J. (2007). On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11(6), 770–784.
55. Farina, M., & Amato, P. (2004). A fuzzy definition of “optimality” for manycriteria optimization problems. *Systems, man and cybernetics, part A: Systems and humans*, IEEE Transactions on, vol. 34, no. 3, pp. 315 – 326, May 2004.
56. Teytaud, O. (2006). How entropy-theorems can show that on-line approximating high-dim pareto-fronts is too hard. in PPSN BTP Workshop, 2006.
57. Laumanns, M., L. Thiele, K. Deb, & Zitzler, E. (2001), On the convergence and diversity-preservation properties of multi-objective evolutionary algorithms.
58. Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, 1(1), 69–91.

# Chapter 5

## Product Portfolio Selection of Designs Through an Analysis of Lower-Dimensional Manifolds and Identification of Common Properties

Madan Mohan Dabbeeru, Kalyanmoy Deb and Amitabha Mukerjee

**Abstract** Functional commonalities across product families have been considered by a large body of product family design community but this concept is not widely used in design. For a designer, a functional family refers to a set of designs evaluated based on the same set of qualities; the embodiments and the design spaces may differ, but the semantics of what is being measured (e.g., strength of a spring) remain the same. Based on this functional behaviour we introduce a product family hierarchy, where the designs can be classified into phenomenological design family, functional part family and embodiment part family. And then, we consider the set of possible performances of interest to the user at the embodiment level, and use multi-objective optimisation to identify the non-dominated solutions or the Pareto-front. The designs lying along this front are mapped to the design space, which is usually far higher in dimensionality, and then clustered in an unsupervised manner to obtain candidate product groupings which the designer may inspect to arrive at portfolio decisions. We highlight and discuss two recently suggested techniques for this purpose. First, with help of dimensionality reduction techniques, we show how these clusters in low-dimensional

---

M. M. Dabbeeru (✉) · K. Deb  
Kanpur Genetic Algorithms Laboratory (KanGAL),  
Indian Institute of Technology Kanpur, Kanpur 208016,  
Uttar Pradesh, India  
e-mail: mmd@umd.edu

K. Deb  
e-mail: deb@iitk.ac.in

A. Mukerjee  
Computer Science and Engineering Department,  
Indian Institute of Technology Kanpur, Kanpur 208016,  
Uttar Pradesh, India  
e-mail: amit@cse.iitk.ac.in

manifolds embedded in the high-dimensional design space. We demonstrate this process on three different designs (water faucets, compression springs and electric motors), involving both continuous and discrete design variables. Second, with the help of a data analysis of Pareto-optimal solutions, we decipher common design principles that constitute the product portfolio solutions. We demonstrate this so-called ‘innovization’ principles on a spring design problem. The use of multi-objective optimisation (evolutionary and otherwise) is the key feature of both approaches. The approaches are promising and further research should pave their ways to better design and manufacturing activities.

## 5.1 Function in Portfolio Planning

Product portfolio selection [1, 2] is a key question facing the firm as it goes from design to manufacture—which set of designs in a *part family* best meet the multiplicity of user expectations across market segments without overly increasing manufacturing and servicing complexity? Arriving at a good portfolio leads to reduced inventory and efficient service, and crucially impinges on profitability.

There are two aspects of product portfolio selection. The first is to maximise the commonality between the parts, metrics for which have been the focus of a large body of work. A second, and relatively less modelled aspect is to consider the functional diversity among the objects. While work on part families have considered performance requirements to various degrees [3–5], and other aspects such as manufacturing process design [6], it has proved challenging to apply these ideas to portfolio standardisation.

Various methodologies have been proposed to aid various manufacturing industries to reduce product family manufacturing costs and assist marketing managers in product portfolio decision making [7]. A two level optimisation is proposed that switches back and forth between the upper (family) level and the lower (variant) level to determine the best combination of the product platforms and product variants that yields maximum overall profit [8]. The product family variables are market segment, product family architecture, product platform architecture and the number of platforms.

In product portfolio decision making, clustering can be used to find the group of products based on similar performative behaviours or similar forms [7, 9]. Agard et al. [9] used neural networks to learn user preferences to build a target user and also clustered users based on similar behaviour in the design of standardised products. In many of these situations, as in most design contexts, the task involves trade-offs between various functional requirements. One of the difficulties has been to map the idea of function itself. Recently Stone et al. [10] proposed a customer-need-motivated conceptual design method, to plan a product portfolio before any embodiment design occurs by using Functional Basis developed by [11] as the language for representation and modelling of product function in the early design phase.



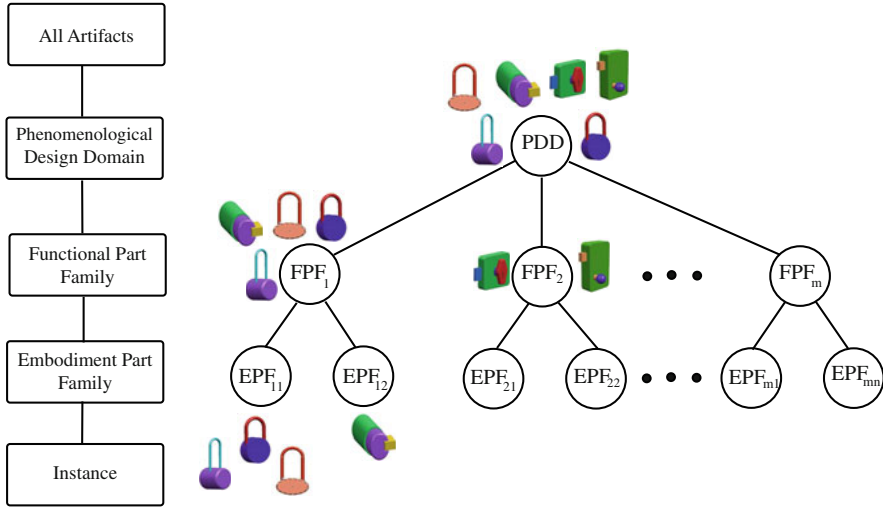
## 5.2 Product Family Design

A product family refers to a set of similar products that are derived from a common platform and yet possess specific features/functionality to meet particular customer requirements [12]. Within a product family, the set of common elements, interfaces and processes is generally called the product platform. Here, we consider a product family hierarchy based on a set of products that serve a related set of market applications—they are similar in form and function, share a phenomenological premise (e.g., faucets control water by constricting a valve) and may adopt similar embodiments [12] (our usage is focussed more on a design perspective, but see [13] for a general review).

In its broadest generality, design deals with all possible artefacts (Fig. 5.1 top level), at which point, ‘function’ is at its most vague. Next, we may consider designs that use similar principles to serve similar functional needs, which constitute the *phenomenological design domain* (PDD)—e.g., arranging for light and air in an architectural space, or restricting access to some interior space. Still there are many ways in which these design goals may be specified and met. Within a PDD, the designs that are evaluated in terms of the same performative behaviours are what we call the *functional part family* (FPF). Thus, single panel windows, multi-panel windows, possibly even rolling shutters, if evaluated using similar performative behaviours, may belong to the same FPF. However, some other structures, like fixed windows (which do not have ‘letting in air’ as a performative behaviour) would constitute a different FPF. Within a specific functional class  $FPF_1$  the designs that meet a set of functions using the same physical structures, so that the design variables map to the performance metrics in the same way, constitute the *embodiment part family* (EPF, or the embodiment class).

For example in case of locking devices, the phenomenological level is based on some physical mechanism for restricting access. Of these, some may share the same set of performative behaviours (FPF). In Fig. 5.1,  $FPF_1$  and  $FPF_2$  both involve keys moving a latch in and out, except that in the latter, the object which will be constrained by the latch is external to the design object. For  $FPF_1$  (padlocks, rotating barrel locks, etc.), shared performance metrics may involve the maximum force it can resist (strength), weight, ease of use, etc. On the other hand the class  $FPF_2$  which are intended to be fixed to something like a door frame, the performative behaviour may consider volume instead of (or in addition to) weight, and thus the set of performative behaviours are different. The class of padlocks, which share the same design structures, constitute an EPF within  $FPF_1$ .

Each EPF is associated with a design space  $\Omega$ , characterised by a  $n$ -tuple design vector  $\underline{v} = (x_1, x_2, \dots, x_n) \in \Omega$  where  $x_i$  are the independent design variables or driving variables for the design. Any other variables needed for specifying the final structure (dependent variables) are defined in terms of these driving variables. At lower levels in the hierarchy, there are fewer degrees of freedom (i.e., number of design variables go down), but the function becomes more crisply specified. At the bottom of the hierarchy are specific design instances, each of which is



**Fig. 5.1** *Hierarchy in design.* Starting with all possible artefacts, the designs that share some principles of operation constitute the phenomenological design domain (PDD). Within these, those that have the same set of shared user needs constitute a functional part family (FPF). Among these, designs with the same embodiment constitute the embodiment part family (EPF)

completely specified (degrees of freedom is zero). The EPF is the niche class that we shall consider throughout this chapter where we present a computational model for designing scalable product family. Later, we will also demonstrate the procedure for visualising higher dimensional scalable product platforms in low-dimensional design spaces by using non-linear dimensionality reduction techniques.

### 5.2.1 *Multi-objective Optimisation in Product Family Design*

In product family design we consider the products at the embodiment level. At this level of design, generally the user preferences involve multiple aspects of performance, which are often conflicting. Once these performative measures are available, we formulate the product family optimisation problem in terms of design variables as multi-objective optimisation problem. Multi-objective optimisation approaches have been used for designing families of products [14–16]. Nelson et al. [14] formulate the product platform design problem using multi-criteria optimisation to resolve the trade-off between commonality and individual product performance within the product family. Simpson and D’Souza [15] use genetic algorithms based approach (NSGA II) for product family design, which is capable of designing the product platform and its corresponding family of products while considering varying levels of platform commonality within the product family.

Simpson et al. summarised two approaches in multi-objective optimisation in product family design, (a) single-stage approaches, wherein the product platform and the individual products are optimised simultaneously [15–18], (b) two-stage approaches, wherein the product platform is designed during the first stage of optimisation, followed by instantiation of the individual products from the product platform during the second stage of optimisation [19, 20]. Simpson and D’Souza [15] have used single-stage approach to simultaneously optimise the product platform and the associated product family, where the designer need not specify the platform commonality a priori to optimisation; at the end of optimisation, the designer will know whether design variables should be made common/unique within the product family and the non-dominated front for different product families based on varying levels of platform commonality within each family. The main limitation in this work is the commonality decision for each variable to whether it is shared by all products in the product family or has different values in each product. To overcome this limitation Khajavirad and Michalek [21] introduce a two-dimensional chromosome to control the commonality use an upper-level GA that controls commonality decisions and a set of lower-level GA that controls the design variables of each product. Similar to [15], Akundi et al. [17] have proposed a multi-objective optimisation method with three objectives, maximising the efficiency, minimising the mass and minimising the variance coefficient among the eight design variables for ten universal motors.

In all these approaches, commonality indices have been used for resolving the trade-off between the commonality and achievement of distinct performance targets. Also, in these approaches, the number of products within a product family is decided a priori, for example the number of products in a family of general aviation aircraft (GAA) is to be scaled around 2-, 4- and 6-seated aircraft to meet different goal targets [15]; in a family of universal motor example the number of products is 10 to meet different torque requirements [17]. Most of these approaches to the product portfolio problem emphasise minimising product and component variety, especially in terms of finding better commonality measures [1, 21–23]. Before we propose our methodology, we briefly mention another similar methodology which uses a multi-objective optimisation and a post-optimality analysis to find properties that are common to the set of Pareto-optimal solutions.

### ***5.2.2 Innovization: Innovation Through Optimisation***

In 2006, Deb [24] suggested a post-optimality procedure which works in two steps. First, a set of near Pareto-optimal solutions are found using a multi-objective optimisation procedure. Either a generative classical method [25] or an evolutionary multi-objective optimisation method [26] can be used for this purpose. Second, a manual [24] or an automatic data-mining method [27] is used to unveil hidden properties involving the decision variables and objectives that are common to the obtained trade-off solutions. As these properties are common to

near Pareto-optimal solutions, they directly indicate innovative principles which if present in a solution would make it a high-performing solution. These properties are useful to practitioners, as they indicate valuable knowledge pertaining the problem.

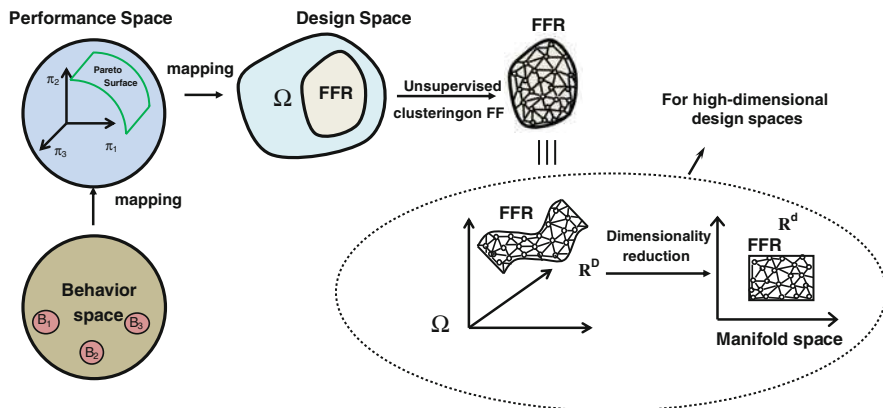
### 5.3 Proposed Approach

Product variety is the diversity among the products that a production system provides to the marketplace [28]. And this is meaningful to customers if products meet widest functions. To predict the user-preferred groupings here we propose a methodology based on a simple insight regarding the nature of non-dominated fronts in multi-criteria decision making—the reduced dimensionality these embody in function space is reflected in terms of small clusters based on similarity in the variable space—and these regions may constitute the nucleus based on which product portfolio choice can proceed. While a number of approaches have considered issues of multi-function optimisation in design, and even explore the ramifications on the design space [14, 29, 30], these approaches have not carried out this idea to extend it to portfolio selection.

Given a product family design, the key steps in the proposed approach are (see Fig. 5.2):

- Identify the desired *performative dimensions*, and the associated *performance metrics* for the user preferences.
- Estimate the non-domination front, using a suitable technique. This corresponds to a lower-dimensional hyper-surface or manifold in the design space along which these non-dominated designs lie.
- Map these non-dominated designs from the performative space to design space and cluster these into groups based on some notion of product similarity (e.g., a simple Euclidean metric in normalised design variable space), using an unsupervised clustering algorithm—for our demonstrations, we use a neural gas algorithm, but one may also use dbScan or hierarchical clustering.
- Identify the lower-dimensional manifold ( $R^d$ ) embedded in the high-dimensional design space ( $R^D$ ) using nonlinear dimensionality reduction algorithms to visualise the clusters in ( $R^d$ ). For this we use locally linear embedding (LLE) as discussed in Sect. 5.6.1.
- Each resulting clusters may be considered as a product grouping. The design team may inspect these groups in low dimension space and choose a single exemplar from each to constitute the product portfolio.

In the product family literature, we observe the notion of a ‘scalable part family’ [1] is a family where designs that can be ‘scaled up’. While this usage shares some aspects of an integral design [28], in that certain variables are continuous and can take different values, it is not quite the same idea. To take an example, a Boeing 747 may be scaled up, but clearly, its body height would not



**Fig. 5.2** In this chapter, our main focus is to determine clusters in the design space of a product family using unsupervised learning algorithms. In case of higher dimensional design spaces we use dimensionality reduction techniques for visualising these clusters in low-dimensional manifold

vary continuously, as in integral designs, but would go from a range appropriate to a single deck, to another appropriate to a double deck. Thus the considerations in scalable product families, such as that of a ‘scale factor’ may be quite different.

In our approach we consider product family optimisation, in which all products share the same set of variables and we assume that the part family is associated with an embodiment (e.g., the spring as a coiled wire with certain material properties). Given this embodiment, we may now define the *performance metrics*, i.e., the quantitative relation by which each performative measure can be evaluated instead of determined from the design choices. In order to validate these performance metrics, clearly a considerable amount of prototyping, user validation and other measures may need to have been done on some sample designs from the part family. Also, over the lifetime of the product, the degree to which these functions reflect actual performance keeps improving. Also, new functions may be added, resulting in different product groups forking off based on these differences [31]. However, for the purposes of this work, we simply assume that some reasonable estimates are available.

Next we can use any multi-criteria optimisation algorithm to identify the non-dominated set of designs (in the demonstration below, we use an evolutionary algorithm, NSGA-II [26]). It appears that using NSGA for multi-objective optimisation may results superior to those obtained by gradient-based approach [26]. Given a set of  $k$  performance criteria  $f_1, f_2, \dots, f_k$ , usually there is no single solution which is optimum with respect to all performance criteria. The resulting problem usually has a set of optimal solutions, known as Pareto-optimal solutions, non-inferior solutions. It stands to reason that user preferences would lie among the designs which constitute the non-dominated set, which amounts to a vast pruning and a significant dimensionality reduction in the design space. Finally,

we show how given this non-domination set, one may cluster these designs in a completely unsupervised manner to obtain regions of the design space corresponding to different functional trade-offs. These product groupings reflect similar behaviours both in terms of design space as well as function, and the design team may then consider these groupings as candidate product classes, each of which may be represented by a single exemplar or product variant, the set of which would constitute the product portfolio.

In the following sections, we explain our proposed approach with the help of three examples.

1. Water faucets.
2. Compression springs.
3. Universal motors.

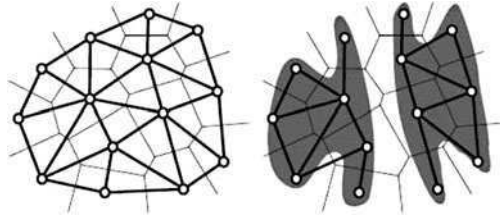
To demonstrate our proposed approach we first start with the basin faucets involving only continuous variables and then move to the second example, that of spring design, is well known in optimisation [32]. In Sect. 5.5 we show the spring product family, where we consider both discrete and continuous variables. Finally we consider the universal motor product family design problem involving eight design variables and three objective functions in Sect. 5.6. In Sect. 5.6.1, we use the well known dimensionality reduction technique to visualise the high-dimensional product groupings in low-dimensions.

### 5.3.1 Unsupervised Clustering: Growing Neural Gas

In order to find out the clusters based on product similarities we have used an unsupervised learning algorithm in which the designs, defined in terms of the corresponding design vectors, are clustered in an unsupervised manner based on the notion of distance between products. For this purpose, we use a neural gas algorithm [33] which learns important topological relations in a given set of input vectors (signals) in an unsupervised manner by means of a simple Hebb-like learning rule. It takes a distribution of high-dimensional data,  $P(\xi)$  and returns a densely connected network resembling the topology of the given data.

A fixed number of random neurons are taken in  $\mathcal{R}^n$ ,  $n$  being the dimension of the design (or signal) space. For every signal  $v_i$  an edge is introduced between the two closest neurons. The resulting network would be a sub-graph of Delaunay triangulation of the set of neurons (Fig. 5.3) with edges present in the regions of high similarity. The neurons that do not participate in this process are called *dead units*. To make use of all the neurons, a Vector Quantisation procedure called Neural Gas is used [33]. For every signal the neurons are adapted towards the signal; the adaptation falls off exponentially as the distance of neuron from the signal increases. This step makes the *dead units* move towards the signal area and participate in the *edge growing* process. An *edge aging* mechanism is introduced, to remove the edges made obsolete by the neuron movement, by setting an upper

**Fig. 5.3** (Left) Delaunay triangulation of vectors in  $\mathfrak{R}^n$ . (Right) Induced triangulation (dark edges) in high similarity (dark) region



bound (*edge aging*  $a_{\max}$  parameter) for the edge ages. These steps are repeated over the signal set till the adaptation or movement of neurons goes to zero and the closely connected neurons lie in the signal activity region. The resulting connected subsets constitute the clusters or product groupings found in the design space. As the process grows out of local neighbourhoods, it will preserve any manifold connectivities inherent in the data. The important input parameters for this algorithm are  $\lambda$ , the fixed node insertion rate,  $e_b$ ,  $e_n$  are the fractions of distances for movement of nodes,  $a_{\max}$  is an upper bound for edge aging parameter and  $T_{\max}$  is the total number of iterations of this algorithm (see [33]).

In the following section we first explain the product grouping based on the unsupervised learning in the design space with three examples (i) water faucets, (ii) spring design (having three design variables) and (iii) electric motors (having eight design variables) and then proceed to dimensionality reduction analysis for the high-dimensional design space of universal motor problem.

## 5.4 Example A: Water Faucet

Here we demonstrate the process of obtaining clusters in the design space based on the user preferences in the form of performance behaviours. With suitable metrics on these performance behaviours and the search in the design space based on these metrics we can come up with a set of Pareto-optimal solutions in the space high dimensional performance measures. In this chapter, we consider performance metrics (user preferences) to be optimised may be defined over product families involving (a) continuous design variables and (b) discrete design variables.

As an example task, we first take up the detailed design of a basin faucet modelled using simple geometric elements for the inlet, outlet and knob. Each of these design elements has a set of driving design variables, in terms of which all other shape parameters as well as joining constraints can be defined. Also the design space for overall product family may have up to 20 parameters, in the analysis below, we restrict the driving parameters to three for the ease of demonstration in this chapter. In this example, we consider all three driving variables are continuous variables.

Figure 5.4 shows the water faucets and their design space spanned by  $\vec{v} = \{w_o, L_o, \theta_2\}$ . The design parameters  $w$ ,  $L$ , and  $\theta_2$  are continuous design variables.

After deciding these three independent design variables, the design problem is formulated as multi-objective optimisation problem shown in Eq. 5.1 for the performative behaviours (a) maximum discharge and (b) minimum weight.

### 5.4.1 Estimating the Non-dominance Front for Faucets

During any design, designers tries to find optimum solutions through searching the design space. The member of the part family is characterised by a set of design variables. In which we focus on a 3-tuple design vector  $w$ ,  $L$ , and  $\theta_2$  which we call as driving variables as the other design dimensions internal to the faucet (Fig. 5.4) are defined in terms of these driving variables. For example, the radius ( $R$ ) of the knob is  $\frac{w}{2}$ . Given a set of values for a design vector, one can determine its shape. The optimal solutions can be obtained by modelling the above problem as a multi-objective optimisation problem for searching the design space. The task of multi-objective problems is different from that of single objective optimisation. Usually in multi-objective solution, there is no single solution which is optimum with respect to all objectives. The resulting problem usually has a set of optimal solutions, known as Pareto-optimal solutions, non-inferior solutions, or effective solutions. As there exists more than one optimal solution and since without further information no one solution can be said to be better than any other Pareto-optimal solution, one of the goals of multi-objective optimisation is to find as many non-dominated (Pareto-optimal) solutions as possible [26].

*Multi-objective optimisation for faucets.*

$$\begin{aligned}
 \text{Minimise} \quad & \pi_{\text{weight}}(\underline{y}) = C_d AV, \quad A = wh, \quad V = \sqrt{2gH_{\text{net}}} \\
 \text{Maximise} \quad & \pi_{\text{discharge}}(\underline{y}) = \frac{\rho V_{\text{vol}}}{L^2}, \\
 \text{Subject to} \quad & g(\underline{y}) \equiv 55.0 < \theta_1 < 60.0, \quad 5.0 < w, h < 8.0 \\
 & 20.0 < L < 40.0, \quad 70.0 < \theta < 150.0 \\
 & \theta_2 = \frac{55.0w - 20.0}{3.0}, \quad \theta_1 = 0.5L + 40.0 \\
 & h = w + 0.5
 \end{aligned} \tag{5.1}$$

where  $H_{\text{net}} = ((H - L \cos(\theta_1)) - (0.85h) \cos((\theta_1 - \theta_2)) - H_f) H = 1,000$  and  $H_f = 0.8H$ .  $V_{\text{vol}} = V_{\text{tap}}, V_{\text{tap}} = V_{\text{body}} + V_{\text{spout}} + V_{\text{mouth}}, V_{\text{body}} = \frac{\pi w^2}{2} h + wh^2, V_{\text{mouth}} = 0.8h^2 w \sin(\frac{\theta_2}{2}), V_{\text{spout}} = 0.5w(L^2 \sin(\theta_1) \sin(\theta_0) + L^2 \cos(\theta_0) \sin(\theta_0) + wL \cos(\theta_0)), C_d = 0.95$ .

In our NSGA-II run, the probabilities of recombination and mutation operators used are  $p_c = 0.8$  and  $p_m = 0.3$  respectively. Considering the two objective functions flow of water, and weight of faucet we obtain a set of non-dominated solutions (Pareto-front). Pareto-optimal front aids the decision maker to choose the non-dominated solution. Any point on the front gives the respective design vector



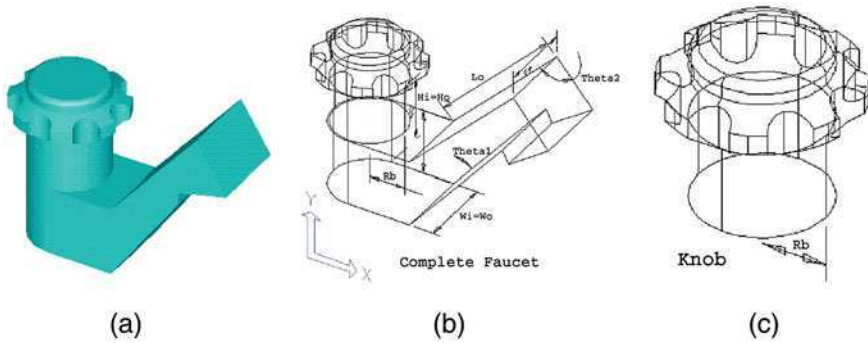


Fig. 5.4 Complete faucet with knob. The driving parameter set  $\{w_o, L_o, \theta_2\}$

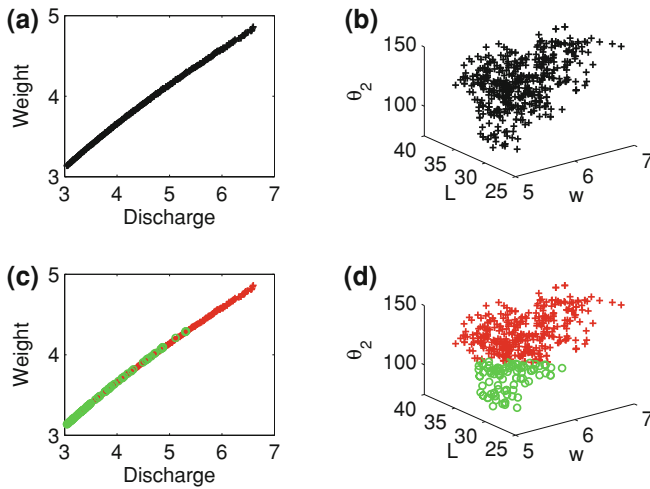


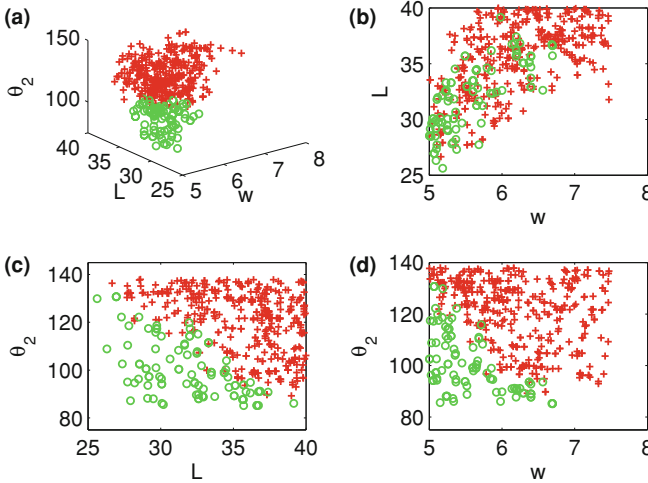
Fig. 5.5 Product groupings for faucet based on non-domination front. The non-dominated solutions (a) are mapped into the design space (b) where clusters are obtained using unsupervised clustering. Here two clusters are shown in design space (d) and in the corresponding non-dominated space (c) for data input of 500. Note that some clusters obtained from the design space overlap along the Pareto front

which defines a design with a set of desired functions. In each generation, new Pareto-fronts are computed based on further explorations in the design space.

Having obtained the non-dominated solutions for maximising the discharge and minimising the weight (Fig. 5.5a), the non-dominated solutions are mapped into the design space, where the good designs are distributed in three-dimensional design space as shown in Fig. 5.5b. On this design data distribution we use the GNG clustering, based on the product similarity. The product similarity is measured by using the Euclidean distance between the design vectors  $\vec{v} = \{w, L, \theta_2\}$ .

**Table 5.1** The mean values of two faucet clusters obtained (Fig. 5.6)

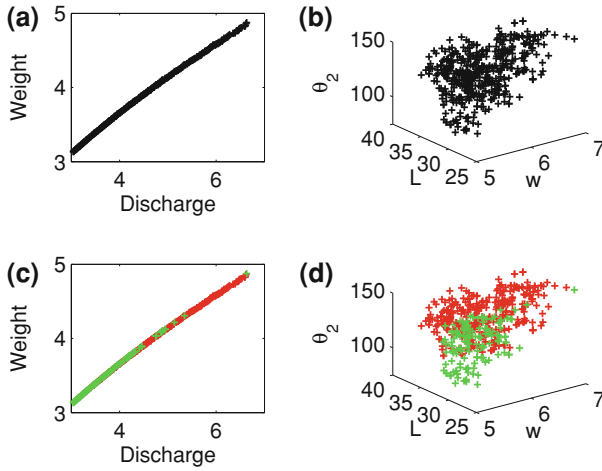
Faucet	Faucet no.	Design variables			Performance measures	
		$w$	$L$	$\theta_2$	$\pi_{\text{Discharge}}$	$\pi_{\text{Weight}}$
+	1.	6.16	35.44	120.91	4.54	4.02
deg	2.	5.41	31.19	95.92	3.53	3.41

**Fig. 5.6** Product groupings for faucet based on non-dominance front: 2D sectional distributions. The projections of clusters are shown in design subspaces **a** ( $w, L$ ), **b** ( $L, \theta_2$ ), and **c** and **d** ( $w, \theta_2$ )

The main purpose of GNG is to generate a graph structure which reflects the topology of the input ‘good designs’ data and the internal structure of the data. After running GNG on this input data we identified clusters in the design space. Initially we experimented with different input GNG parameters. The results shown in Fig. 5.5c, d are obtained for  $\lambda = 200$ ,  $e_b = 0.06$ ,  $e_n = 0.009$ ,  $a_{\max} = 4$ ,  $T_{\max} = 40,000$ ,  $\alpha = 0.01$ . With these GNG parameters we obtain two product groupings (clusters) and when these two clusters are mapped back on to the non-dominated space, the mapping is almost linear from the design space to function space. There is also few members of one cluster are overlapped with the other cluster.

Now this will constitute a product family having two product groupings. Each group shares similar performance behaviours (discharge, weight) but the particular performance values’ ranges are different to satisfy two market segments. Faucets having high discharge and heavy weight are grouped into one group and the other group has low discharge and low weight faucets. The design vector and the performance measures for the mean of the two product groupings are shown in Table 5.1.

Sometimes it is useful to visualise these clusters in the design spaces for identifying the common design variable values for inventory, manufacturing and



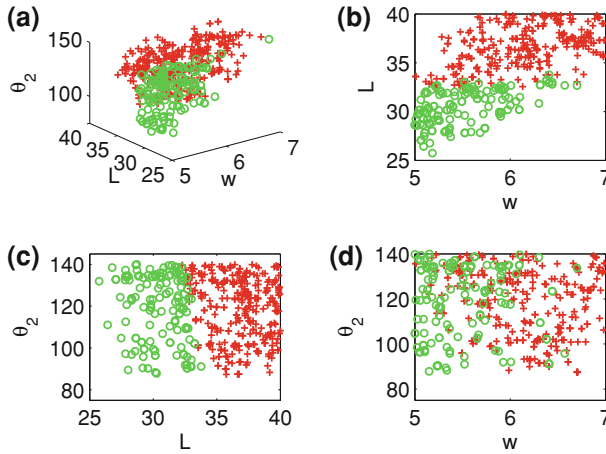
**Fig. 5.7** Product groupings for varying GNG parameters

for other decision making purposes. Figure 5.6 shows the 2D sectional distribution of the projected clusters in the design subspaces  $(w, L)$ ,  $(L, \theta_2)$ , and  $(w, \theta_2)$ . From Fig. 5.6c, d, faucets having both high discharge and heavy weight are having long spout lengths  $L$  and high spout angle  $(\theta_2)$  values whereas the low discharge and light weight faucets are having low width  $w$ , low spout lengths  $(L)$  and less spout angles  $(\theta_2)$ . From these design spaces, user can take any product from each cluster as one standard product. The mean values of each cluster are shown in Table 5.1. From the mean values of these clusters it is clearly understood that faucets having less spout angle are grouped into second cluster (deg) and high spout angles are grouped into first cluster (+).

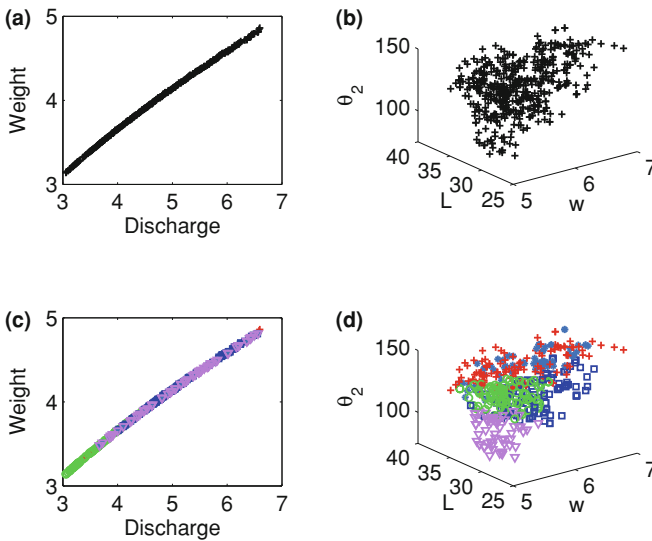
### 5.4.2 Varying GNG Parameters

#### Faucet Family with 2-Clusters

With a different set of GNG parameters  $\lambda = 200$ ,  $e_b = 0.00006$ ,  $e_n = 0.00004$ ,  $a_{\max} = 80$ ,  $T_{\max} = 40,000$ ,  $\alpha = 0.01$ , we obtain two clusters as shown in Fig. 5.7. Figure 5.8 shows the 2D sectional distribution of the projected clusters in the design subspaces  $(w, L)$ ,  $(L, \theta_2)$ , and  $(w, \theta_2)$ . From Fig. 5.8b, c, the spout length of the faucet  $L$  is partitioned into two groups, one  $\approx 25\text{--}32$  cm and the other  $\approx 32\text{--}40$  cm. Thus, for high discharge values, higher lengths are preferred though the weight is also heavy. Hence, this product family obtained here is divided based on the (high-discharge, low-weight) and (low-discharge, high-weight). From Fig. 5.8d it is partly helpful to decide the  $w$  values between 5–6 cm as one group and 6–7 cm as another group.



**Fig. 5.8** Product groupings for faucet based on non-domination front: 2D sectional distributions. The projections of clusters are shown in design subspaces **a** ( $w, L$ ), **b** ( $L, \theta_2$ ), and **c** and **d** ( $w, \theta_2$ )



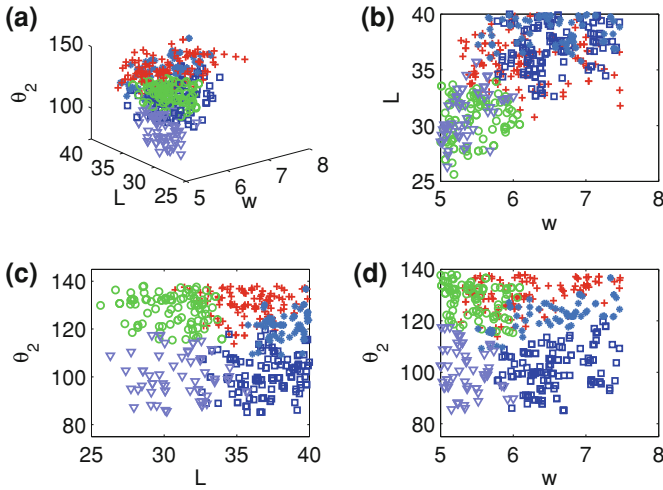
**Fig. 5.9** Product groupings for varying GNG parameters. The non-dominated solutions (a) are mapped into the design space (b) where clusters are obtained using unsupervised clustering. Here five clusters are shown for data input of 500. Note that some clusters obtained from the design space overlap along the Pareto front

### Faucet Family with 5-Clusters

Next, we experiment with another GNG parameters  $\lambda = 200$ ,  $e_b = 0.0001$ ,  $e_n = 0.0004$ ,  $a_{\max} = 10$ ,  $T_{\max} = 40,000$ ,  $\alpha = 0.1$  and Fig. 5.9 shows the same Pareto-front shown in Fig. 5.7a but with five clusters. Now, we have a product

**Table 5.2** The mean values of five faucet clusters obtained (Fig. 5.10)

	Faucet no.	Design variables			Performance measures	
		$w$	$L$	$\theta_2$	$\pi_{\text{Discharge}}$	$\pi_{\text{Weight}}$
+	1.	6.05	35.35	123.84	4.39	3.85
deg	2.	5.74	31.43	118.02	3.96	3.64
	3.	6.30	36.73	108.93	4.73	4.01
*	4.	6.17	36.68	116.54	4.55	3.93
$\nabla$	5.	5.50	31.01	108.29	3.66	3.48

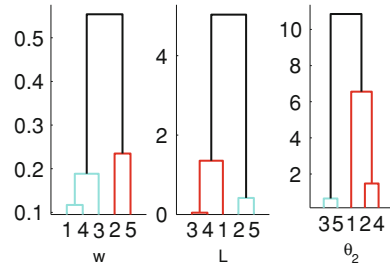


**Fig. 5.10** Product groupings for faucet based on non-domination front: 2D sectional distributions. The projections of clusters are shown in design subspaces **a** ( $w, L$ ), **b** ( $L, \theta_2$ ), and **c** and **d** ( $w, \theta_2$ )

family consisting five groupings satisfying five groups of users. Now, selecting a standardised product from each cluster is a difficult task. The actual product instances within each cluster are not determined by us. We assume here that situational aspects—e.g., set of existing products, competitor’s products, etc. may impact this, and the cluster are simply a guide for the end user. One possible approach may be to consider the means of each cluster. Table 5.2 shows the set of such means to illustrate the variation between the clusters (Fig. 5.10).

From Table 5.2, we can observe that some products in the product family are very close on some common variables. For example, the product 2 has nearly the same  $w$  as 5. Similarly, product 3 shares a similar  $L$  with the 4. To determine these common values, we have used Dendrograms to determine the common variables in the product family (see [20, 34] for more details on product portfolio optimisation). Figure 5.11a–c are showing the Dendrograms for  $w$ ,  $L$  and  $\theta_2$  respectively. The horizontal axis is the product number and the vertical axis is the index of dissimilarity.

**Fig. 5.11** *Commonisation among the design variables:* Dendrograms resulting from cluster analysis of five values of each design variable  $w, L$  and  $\theta_2$



**Table 5.3** Faucet product family having five faucet groupings

Design Variable	Faucet 1	Faucet 2	Faucet 3	Faucet 4	Faucet 5
$w$	$w_1$	$w_2$	$w_3$	$w_4$	$w_2$
$L$	$L_1$	$L_2$	$L_3$	$L_3$	$L_2$
$\theta_2$	$\theta_{21}$	$\theta_{22}$	$\theta_{23}$	$\theta_{22}$	$\theta_{23}$

It is observed that some of these mean values in each cluster share nearly same values across different groupings. For example, the lengths of Faucets 3 and 4 can have a single value  $L_3$  and Faucets 2 and 5 can have  $w_2$

From this Dendrograms one may take a decision to choose the common variables based on the similarity observed in Table 5.3. Faucet 1 is having individual design variable values, faucet 2 is sharing  $w_2$  and  $L_2$  with faucet 5 and faucet 3 and 4 are sharing the same  $L$  values.

For choosing the value of these common variables showed in Table 5.3 one can use sensitivity of design variables with respect to overall performances to select the commonisation values for each product platform variable [20].

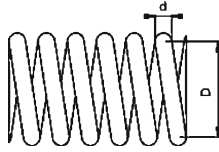
### 5.5 Example B: Spring Design

Here, we consider the design of a helical compression spring, which is a well-studied problem in the context of optimisation in design [32]. We are interested in obtaining clusters in the design space based on the performative dimensions of (a) weight and (b) strength. The design space is represented by the three design variables: the number of spring coils  $N$ , which is an integer value in the range [1, 32], the wire diameter  $d$ , which is a discrete variable in the range 0.009 and 0.5 in and in unequal steps as presented in [29, 32], and the mean coil diameter  $D$ , which is a continuous variable in the range [1, 30].

These constitute the design vector  $\underline{v} = (x_1, x_2, x_3) = (N, d, D)$ , which is the signal space both for the optimisation, and subsequently for the GNG clustering (Fig. 5.12). We define the optimisation problem as follows:

*Multi-objective optimisation for compression spring.*

**Fig. 5.12** Compression spring and its design variables  $\underline{v} = \{N, d, D\}$



Design variables	
Number of turns	$N$
Diameter of the wire	$d$
Diameter of the spring	$D$

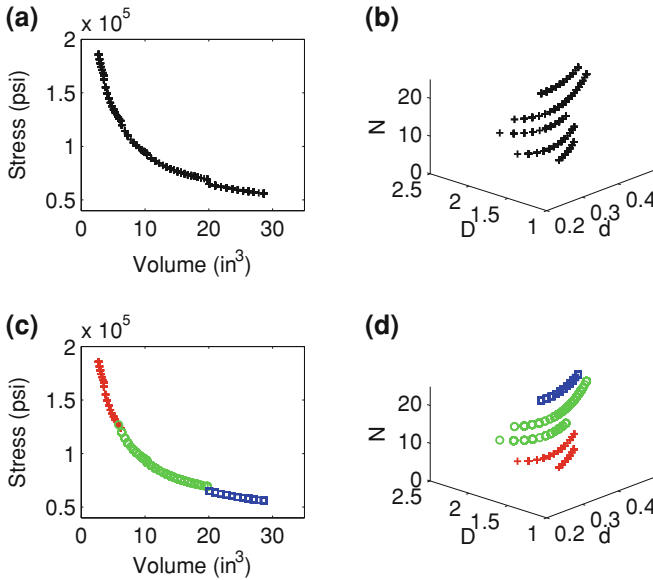
$$\begin{aligned}
 &\text{Minimise } \pi_{\text{volume}}(\underline{v}) = 0.25\pi^2 x_2^2 x_3 (x_1 + 2) \\
 &\text{Minimise } \pi_{\text{stress}}(\underline{v}) = \frac{8KP_{\text{max}}x_3}{\pi x_2^3}, \\
 &\text{Subject to } g_1(\underline{v}) \equiv l_{\text{max}} - \frac{P_{\text{max}}}{k} - 1.05(x_1 + 2)x_2 > 0 \\
 &g_2(\underline{v}) \equiv x_2 - d_{\text{min}} \geq 0, \quad g_3(\underline{v}) \equiv C - 3 \geq 0, \\
 &g_4(\underline{v}) \equiv \delta_{pm} - \delta_p \geq 0, \\
 &g_5(\underline{v}) \equiv D_{\text{max}} - (x_2 + x_3) \geq 0 \\
 &g_6(\underline{v}) \equiv \frac{P_{\text{max}} - P}{k} - \delta_w \geq 0, \\
 &g_7(\underline{v}) \equiv S - \frac{8KP_{\text{max}}x_3}{\pi x_2^3} \geq 0 \\
 &g_8(\underline{v}) \equiv V_{\text{max}} - 0.25\pi^2 x_2^2 x_3 (x_1 + 2) \geq 0 \tag{5.2}
 \end{aligned}$$

$$\begin{aligned}
 P_{\text{max}} &= 1,000 \text{ lb} & P &= 300 \text{ lb} \\
 D_{\text{max}} &= 3 \text{ in} & k &= \frac{Gx_2^4}{8x_1x_3^3} \\
 \delta_w &= 1.25 \text{ in} & \delta_p &= \frac{P}{k} \\
 \delta_{pm} &= 6 \text{ in} & S &= 189 \text{ ksi} \\
 C &= \frac{x_3}{x_2} & V_{\text{max}} &= 30 \text{ in}^3 \\
 K &= \frac{4C-1}{4C-4} + \frac{0.615x_2}{x_3} & l_{\text{max}} &= 14 \text{ in} \\
 d_{\text{min}} &= 0.2 \text{ in} & G &= 11,500,000 \frac{\text{lb}}{\text{in}^2}
 \end{aligned}$$

As in the previous section for the faucets, here also we model the multi-objective optimisation problem and obtain a set of Pareto-optimal solutions (Fig. 5.13) based on these two performative behaviours represented by the performance metrics  $\pi_{\text{volume}}$  and  $\pi_{\text{stress}}$ , as used in the literature. The obtained Pareto-front (Fig. 5.13) closely follows [29], where it is shown to distribute well over the Pareto-front as obtained by the normal constraint method [30].

Having obtained the Pareto-front, we now map all these ‘good springs’ to the design space. In the design space shown in Fig. 5.13b, the designs are distributed in  $d$ - $D$ - $N$  space, where we use unsupervised clustering to obtain the clusters based on product similarities. The GNG parameters considered here are:  $\lambda = 600$ ,  $e_b = 0.08$ ,  $e_n = 0.0009$ ,  $a_{\text{max}} = 90$ ,  $T_{\text{max}} = 90,000$ ,  $\alpha = 0.5$ . The resulting three clusters are shown in Fig. 5.13d and the corresponding non-dominated space is shown in Fig. 5.13c.

With this, the designer can divide all products into three different groups. Now, this product platform has three spring groups and if a spring has to be designed with a material having yield strength ranging from 185,000–123,900 psi then from the cluster  $A$ - $B$  the optimal diameter can be either 0.283 or 0.331 and in the similar way, if the designer is looking for material having yield strength 119,000–69,240 psi then he can choose the diameter of the wire either 0.394 or 0.437 from the cluster  $B$ - $C$  and for the yield strength 64,870–55,950 psi with a fixed diameter



**Fig. 5.13** Non-dominated function space for the spring and the corresponding design space. **a** Non-dominated solutions are obtained with three variables  $N$  (integer),  $d$  (discrete),  $D$  (continuous). The objective functions are minimising both volume and stress for the helical compression spring. **b** The non-dominated solutions are mapped into the design space. **d** Three clusters are determined using the unsupervised learning and the corresponding non-dominated front is shown in (c)

**Table 5.4** Design variables and the corresponding performance measures for three clusters shown in Fig. 5.13

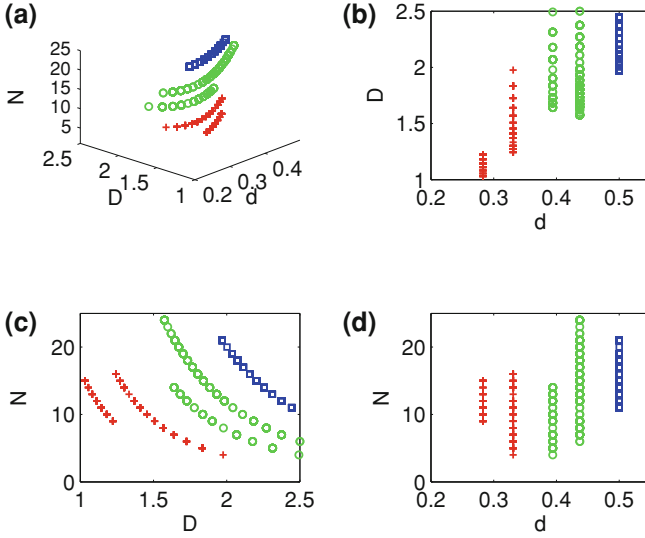
	Spring no.	Design variables			Performance measures	
		$d$	$D$	$N$	$\pi_{\text{volume}}$	$\pi_{\text{stress}}$
+	1.	0.3104	1.35	10	4.15	155,082
☆	2.	0.4226	1.91	13	13.04	86,043
	3.	0.5000	2.15	16	25.17	59,466

of the wire  $d = 0.5$  in in the cluster  $C-D$ . The main advantage with this clustering is to group the products based on their product similarities (Table 5.4).

### 5.6 Example C: Universal Electric Motors

In this section we consider the universal motor, a product family involving a high-dimensional design spaces. Universal motor problem is a well studied product family problem in portfolio optimisation literature. Simpson [35] considers the





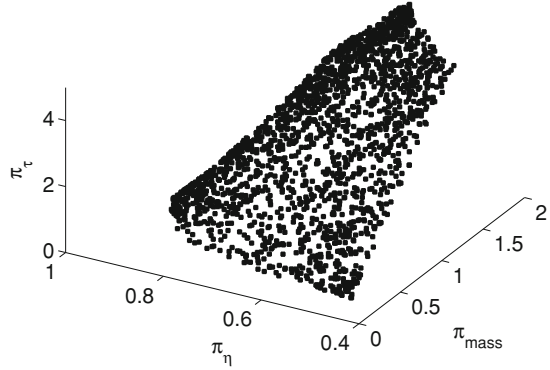
**Fig. 5.14** Product groupings for spring based on non-dominance front: 2D sectional distributions. The projections of clusters are shown in design subspaces **a** ( $d, D$ ), **b** ( $D, N$ ), and **c** and **d** ( $d, N$ )

case of universal motor, with ten instantiations being considered, in order to reduce the cost, size and mass and finally have developed the universal motor family. The design space for embodiment design consists of ten design variables  $\vec{v} = \{N_c, N_s, A_{wa}, A_{wf}, r_o, t, l_{gap}, I, V_t, L\}$  (see [35]), and the performance behaviours are taken as strength, mass, energy and efficiency and the corresponding performance metrics in terms of these design variables can be  $\pi_{torque}(\vec{v}) = \frac{N_c \phi I}{\Pi}$ ,  $\pi_{mass}(\vec{v}) = mass_{windings} + mass_{armature} + mass_{windings}$ ,  $\pi_{power}(\vec{v}) = V_t I - I^2(R_a + R_s) - 2I$ , and  $\pi_{efficiency}(\vec{v}) = \frac{\pi_{power}}{V_t I}$ .

Akundi et al. [17] have considered the same universal motor example to have the highest efficiency and least possible mass to develop a family of ten universal electric motors to satisfy a range of torque requirements. Along with the two objective functions maximising efficiency and minimising the mass, they have considered the third objective function to minimise the variance coefficient of the design variables of ten motors to maximise the commonality. However, this requires a considerably larger population in the GA ( $10n$  instead of original population  $n$ ) (Fig. 5.14).

Here in our method, we do not have any information either on the commonality of product platform or the number of products in the product family. Our method is helpful for decision makers who would like to visualise the possible product families for a given set of user preferences. We initially model the user preferences as a multi-objective optimisation problem and based on the unsupervised learning we determine the product groupings of ‘good designs’ in the design space based on

**Fig. 5.15** *Non-dominated space for universal motor.*  
The non-dominated solutions are lying on the Pareto surface in the 3-objective space of mass, efficiency and torque



the product similarity. The mathematical formulation of multi-objective optimisation problem as follows:

*Multi-objective optimisation.*

$$\begin{aligned}
 &\text{Minimise} && \pi_{\text{mass}}(\mathcal{Y}) \\
 &\text{Maximise} && \pi_{\text{efficiency}}(\mathcal{Y}) \\
 &\text{Maximise} && \pi_{\text{torque}}(\mathcal{Y}) \\
 &\text{Subject to} && g_1(\mathcal{Y}) \equiv r_o - t > 0 \\
 & && g_2(\mathcal{Y}) \equiv 5,000 - H > 0, \\
 & && g_3(\mathcal{Y}) \equiv 2.0 - \pi_{\text{Mass}} \geq 0, \\
 & && g_4(\mathcal{Y}) \equiv 0.5 \leq \pi_{\text{torque}} \leq 5.0, \\
 & && g_5(\mathcal{Y}) \equiv 300 \leq \pi_{\text{Power}} \leq 600 \\
 & && g_6(\mathcal{Y}) \equiv \pi_{\text{efficiency}} - 0.15 \geq 0
 \end{aligned} \tag{5.3}$$

Figure 5.15 is the showing the non-dominated Pareto-optimal surface for three performative behaviours mass, efficiency and torque. After mapping these non-dominated solutions into the eight-dimensional design space for unsupervised grouping, we can obtain different number of clusters by varying GNG parameters. So far we have considered examples (water faucets, springs), those are not having more than three design variables. But in real life design problems, we may have design problem consisting of more than hundred design variables and hence constitute high-dimensional design spaces. In these cases it is difficult to visualise these design spaces.

In the following section, we demonstrate the process of mappings from high-dimensional design spaces to low-dimensional spaces for visualising the clusters obtained with different sets of GNG parameters.

### 5.6.1 Clusters in Low-Dimensional Manifolds

We now present the algorithm used for obtaining a low-dimensional representation for the ‘good design’ subspace of the original high-dimensional design space. Although, the design is defined in terms of a hundred parameters, for the class of

‘good designs’, there are often many interrelations between these; each such interrelation constitutes a chunk or a dimension in the resulting low-dimensional surface or manifold.

---

**Algorithm 1** Local Linear Embedding
 

---

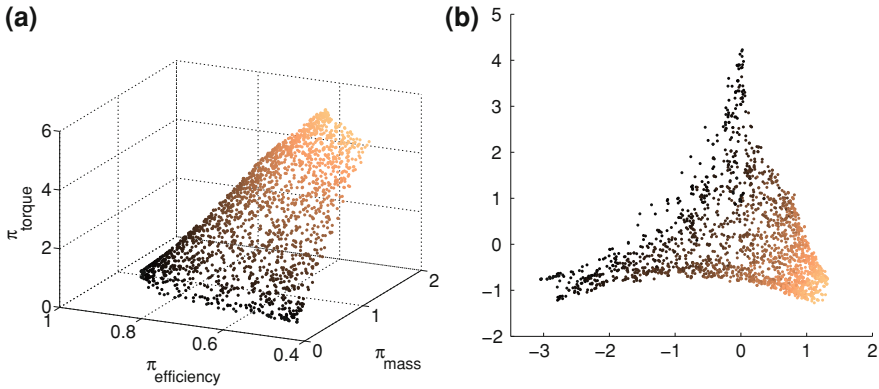
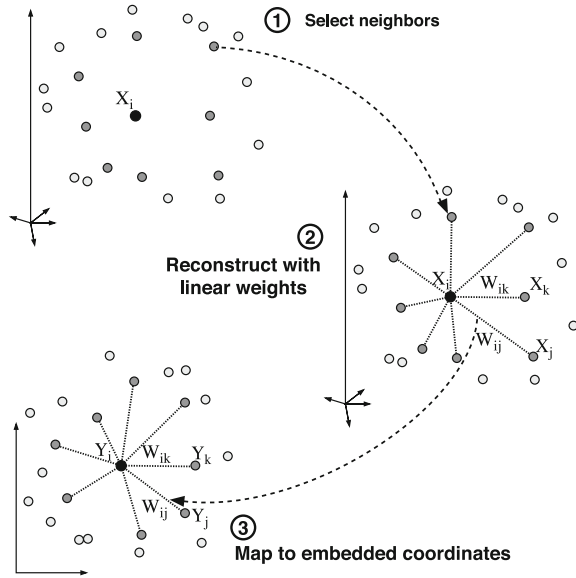
1. Compute the neighbors  $X_j$  of each data point,  $X_i$ .
  2. Compute the weights  $W_{ij}$  that best reconstruct each data point  $X_i$  from its neighbors, minimizing the reconstruction error ( $\epsilon(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2$ ) by constrained linear fits.
  3. Compute the vectors  $\Gamma_i$  best reconstructed by the weights  $W_{ij}$ , minimizing the quadratic form ( $\Phi(\Gamma) = \sum_i |\Gamma_i - \sum_j W_{ij} \Gamma_j|^2$ ) by its bottom nonzero eigenvectors.
- 

One of the strategies to handle high dimension data is *dimensionality reduction*, involves finding low-dimensional structures in high-dimensional space. A large number of different methods have been developed for this purpose. There have been linear dimensionality reduction methods [36]—e.g., independent component analysis, linear discriminate analysis, principal component analysis but these linear methods fail when the data lies on a nonlinear manifold; in such situations the linear algorithms give the smallest convex subspace encapsulating the manifold, which is often of a much higher dimension. In practice, non-linear relations between design variables are extremely common, and in such situations, nonlinear dimensionality reduction yields superior results [37]. Approaches for obtaining the non-linear representation of the data include Global methods (Isomaps [37]) and Local methods (LLE [38] and Laplacian Eigenmaps [39]). Local approaches try to preserve the local geometry of the data. By approximating each point on the manifold with a linear combination of its neighbors, and then using the same weights to compute a low-dimensional embedding, LLE tries to map the nearby points on the manifold to nearby points in the low-dimensional representation.

High dimensionality and computational complexity are curses typically associated with many product family design problems [40]. In this chapter, we have applied an eigenvector method—called LLE for the problem of non-linear dimensionality reduction. The basic idea in LLE is that of global minimisation of the reconstruction error of the set of all local neighbourhoods in the given data (Fig. 5.16) [38]. LLE is an unsupervised learning algorithm and it was first proposed by Roweis and Saul [41]. The main interesting property of this algorithm [38] is that it preserves the relationships between neighbours in a data set and represents high dimensional data  $X = \{x_1, x_2, \dots, x_n\}, x_i \in R^D$  in a lower dimensional space  $Y = \{y_1, y_2, \dots, y_n\}, y_i \in R^d$ .

Having obtained non-dominated sets of designs, and mapping these to the design space reveals that the good designs are often restricted to a few patches on a low-dimensional manifold, thus resulting in significant dimensionality reductions for the design space. Figure 5.17a shows the non-dominated surface and the

**Fig. 5.16** Local linear embedding (LLE) algorithm



**Fig. 5.17** *Low-dimensional design space.* The non-dominated solutions shown in (a) are mapped into 8-dimensional design space and the corresponding 2-dimensional design space is obtained through non-linear dimensionality reduction. **a** Non-dominated space, **b** LLE in design space

corresponding low-dimensional design space is shown in Fig. 5.17b. In the following sections, we show the product families in these low-dimensional design spaces.

**Universal Motors: 3 Clusters**

Next, we experiment with another set of GNG parameters  $\lambda = 100$ ,  $e_b = 0.008$ ,  $e_n = 0.004$ ,  $a_{max} = 5$ ,  $T_{max} = 10,000$ ,  $\alpha = 0.02$ . Figure 5.18a shows the non-dominated space with three clusters.

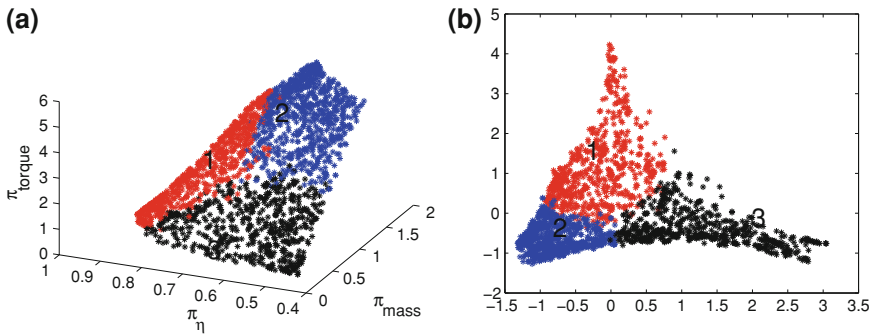
Table 5.5 is showing the mean values of each cluster and their corresponding mappings in low-dimensional manifold. From the mean values of the performance

**Table 5.5** Design variables and the corresponding performative behaviours for universal motor designs in three clusters shown in Fig. 5.18

High-dimensional design variables $D = 8$								$\pi$ for cluster means			
$N_c$	$N_s$	$A_{wa}$	$A_{wf}$	$r_o$	$t$	$I$	$L$	$\pi_{mass}$	$\pi_\eta$	$\pi_\tau$	
1	1,461	495	0.966	0.930	11.3	4.25	5.98	18.9	1.41	83.2	1.703
2	1,472	492	0.7741	0.587	15.1	4.5	6.0	26.7	1.62	68.4	3.602
3	1,071	500	0.259	0.265	10.4	4.0	6.0	10.2	0.24	67.1	0.614

Low-dimensional design variables $d = 2$		$\pi$ for cluster means			
$q_1$	$q_2$	$\pi_{mass}$	$\pi_\eta$	$\pi_\tau$	
1	-0.3600	1.5752	1.40	0.83	1.70
2	-0.8196	-0.3512	1.6	0.684	3.6
3	1.9885	-0.11283	0.24	0.67	0.61



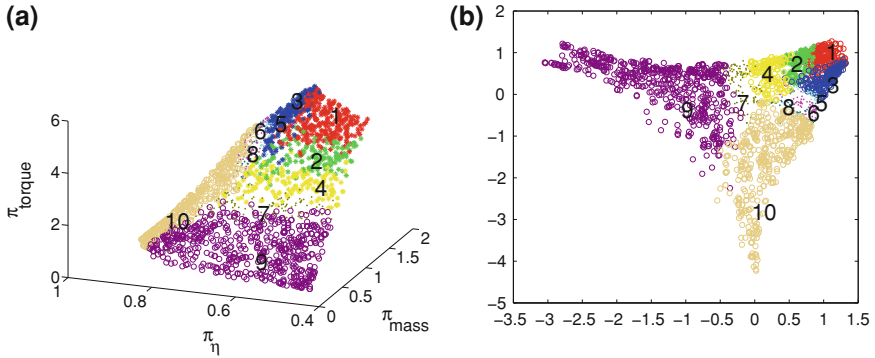
**Fig. 5.18** Clusters in the non-dominated space for universal motor. **a** The non-dominated solutions with three clusters (Pareto-front) in the three-objective space of mass, efficiency and torque. **b** The manifold space corresponding to three clusters, the map from the high-dimensional design space  $D = 8$  to low-dimensional design space  $d = 2$  obtained with the help of LLE

measures, we can observe that motors are grouped based on three different torque values ranging from low torque (0.614 Nm), medium torque (1.703 Nm) and high torque (3.6 Nm) while having different mass and efficiency values. Some of the design variables like current  $I$ , number of winding on the field  $N_s$  and thickness  $t$  are sharing among these three clusters. Figure 5.18b shows these three clusters in the low-dimensional manifold embedded in high-dimensional design space obtained by using the LLE. We can observe that some of the members of each cluster are overlapping with other clusters in this low-dimensional space.

**Universal Motors: 10 Clusters**

In this case, with  $\lambda = 110$ ,  $e_b = 0.0001$ ,  $e_n = 0.00025$ ,  $a_{max} = 5$ ,  $T_{max} = 20,000$ ,  $\alpha = 0.2$ , we obtain ten clusters shown in Fig. 5.19a in non-dominated space. Table 5.6 is showing the mean values of each clusters obtained here.

Figure 5.19b is showing the corresponding low-dimensional manifold space embedded in eight-dimensional design space. Here it is clearly shown that there is overlapping among these clusters.



**Fig. 5.19** Clusters in the non-dominated space for universal motor. **a** The non-dominated solutions (Pareto-front) in the three-objective space of mass, efficiency and torque with ten clusters. **b** The manifold space showing ten clusters corresponding to the mapping from the high-dimensional design space  $D = 8$  to low-dimensional design space  $d = 2$  obtained with the help of LLE. Note that there is overlapping between the ten clusters in this low-dimensional space

**Table 5.6** Design variables and the corresponding performative behaviours for universal motor designs in ten clusters shown in Fig. 5.19

	Design variables								$\pi$ for cluster means		
	$N_c$	$N_s$	$A_{wa}$	$A_{wf}$	$r_o$	$t$	$I$	$L$	$\pi_{\text{mass}}$	$\pi_{\eta}$	$\pi_{\tau}$
1.	1,418	492	0.454	0.417	12.75	4.09	5.97	18.46	0.728	64.6	1.93
2.	1,417	492	0.475	0.431	13.17	4.09	5.98	19.63	0.806	64.0	2.16
3.	1,336	493	0.781	0.821	14.51	4.46	5.93	23.57	1.541	76.5	2.67
4.	1,346	494	0.768	0.812	14.77	4.46	5.94	24.46	1.582	75.3	2.87
5.	1,356	495	0.756	0.806	14.98	4.45	5.94	25.16	1.617	74.3	3.04
6.	1,446	493	0.539	0.469	14.54	4.05	5.98	23.66	1.091	61.3	3.06
7.	1,366	496	0.728	0.786	15.46	4.48	5.95	26.92	1.680	71.8	3.43
8.	1,463	495	0.570	0.488	15.28	4.07	5.99	26.24	1.270	59.5	3.66
9.	1,383	496	0.712	0.764	15.74	4.45	5.96	28.09	1.722	69.9	3.73
10.	1,472	496	0.612	0.531	16.07	4.07	5.99	29.09	1.512	58.9	4.35

The non-linear dimensionality reduction technique used here is not so powerful when confronted with noisy data, which is often the case for real-world problems [42]. When the given data is having high-noise, there will be a chance of overlapping clusters and becomes difficult to visualise [43]. There have been research on various non-linear dimensionality reduction techniques for classification and visualisation [42, 43], which is currently under our investigation.

## 5.7 Summary

In this chapter, we determined product portfolios based on user preferences modeled as a multi-objective optimisation problem. In the initial stages of introducing any product into the market, the method proposed here is helpful to

decision makers to have an idea on groupings of ‘good designs’ lying on the non-dominated Pareto-front.

In real world design situations, for example, a digital camera (in terms of all its components, and assembly processes) may have several hundred design variables (let us say design space with  $D = 100$ ) but only about ten performative measures. Now, restricting our attention to designs in the non-dominated set implies that  $\pi(\vec{v}) \in P$  ( $P$ —performance space). Clearly, this constitutes an additional restriction on  $\vec{v}$ , and thus bounds it more tightly than  $\vec{v} \in \Omega$ . Here, we claim that the obtained non-dominated sets of designs based on these ten performative measures reveals that these good cameras are restricted to a few patches on a low-dimensional manifold ( $d < 100$ ), thus resulting in significant dimensionality reductions for the design space and the design seems to be constrained to a much lower dimensional manifold.

After obtaining product groupings in the design space with the help of GNG algorithms, we have used LLE, a dimensionality reduction technique, for visualising the clusters embedded in the low-dimensional manifold. The advantage lies here is that it is possible to map any new design vector  $\vec{v}$  from  $R^D$  to  $R^d$  and vice versa. These mappings are provided by [38].

It is observed that in Fig. 5.19b the clusters in the low-dimensional spaces are mixed up. We are currently investigating to alleviate this problem.

## References

1. Jiao, J., Simpson, T. W., & Siddique, Z. (2007). Product family design and platform-based product development: A state-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1), 5–29.
2. Jiao, J., & Zhang, Y. (2005). Product portfolio identification based on association rule mining. *Computer-Aided Design*, 37(2), 149–172.
3. Simpson, T. W., Maier, J. R., & Mistree, F. (2001). Product platform design: Method and application. *Research in Engineering Design*, 13(1), 2–22.
4. Nayak, R. U., Chen, W., & Simpson, T. W. (2002). A variation-based method for product family design. *Engineering Optimization*, 34, 65–81.
5. Dabbeeru, M. M., & Mukerjee, A. (2008). *Functional part families and design change for mechanical assemblies*. In *Proceedings of DETC'08. 2008 ASME design engineering technical conferences DETC2008-49739*, New York, USA.
6. Huang, Z., & Yip-Hoi, D. (2003). Parametric modeling of part family machining process plans from independently generated product data sets. *Journal of Computing and Information Science in Engineering*, 3, 231.
7. Jiao, J., Zhang, Y., & Wang, Y. (2007). A generic genetic algorithm for product family design. *Journal of Intelligent Manufacturing*, 18(2), 233–247.
8. De Weck, O. L., Suh, E. S., & Chang, D. (2003). Product family and platform portfolio optimization. In *Proceedings of ASME design engineering technology conferences*.
9. Agard, B., & Kusiak, A. (2004). Standardization of components, products and processes with data mining. In *International conference on production research Americas, Santiago, Chile*.
10. Stone, R., Kurtadikar, R., Villanueva, N., & Arnold, C. B. (2008). A customer needs motivated conceptual design methodology for product portfolio planning. *Journal of Engineering Design*, 19(6), 489–514.
11. Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., & Wood, K. L. (2002). A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13(2), 65–82.

12. Meyer, M. H., & Utterback, J. M. (1993). The product family and the dynamics of core capability. *Sloan Management Review*, 34(3), 29–47.
13. Simpson, T. W. (2004). Product platform design and customization: Status and promise. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18, 3–20.
14. Nelson, S. A., Parkinson, M. B., & Papalambros, P. Y. (2001). Multicriteria optimization in product platform design. *ASME Journal of Mechanical Design*, 123(2), 199–204.
15. Simpson, T. W., & D'Souza, B. S. (2004). Assessing variable levels of platform commonality within a product family using a multiobjective genetic algorithm. *Concurrent Engineering*, 12(2), 119.
16. Messac, A., Martinez, M. P., & Simpson, T. W. (2002). Effective product family design using physical programming. *Engineering Optimization*, 34(3), 245–261.
17. Akundi, S., Simpson, T. W., & Reed, P. M. (2005). Multi-objective design optimization for product platform and product family design using genetic algorithms. In *Proceedings of DETC'05, ASME design engineering technical conferences and computers and information in engineering conference, Long Beach, CA*.
18. Fujita, K., & Yoshida, H. (2001). Product variety optimization: Simultaneous optimization of module combination and module attributes. In *Proceedings of the 2001 ASME design engineering technical conferences* (pp. 9–12).
19. Zugasti, J. P. G., Otto, K. N., & Baker, J. D. (2001). Assessing value in platformed product family design. *Research in Engineering Design*, 13(1), 30–41.
20. Dai, Z., & Scott, M. J. (2007). Product platform design through sensitivity analysis and cluster analysis. *Journal of Intelligent Manufacturing*, 18(1), 97–113.
21. Khajavirad, A., & Michalek, J. (2007). An extension of the commonality index for product family optimization. In *DETC2007* (pp. 4–7).
22. Kota, S., Sethuraman, K., & Miller, R. (2000). A metric for evaluating design commonality in product families. *Journal of Mechanical Design, ASME*, 122, 143–150.
23. Wacker, J. G., & Treleven, M. (1986). Component part standardization: An analysis of commonality sources and indices. *Journal of Operations Management*, 6(2), 219–244.
24. Deb, K., & Srinivasan, A. (2006). Innovization: Innovating design principles through optimization. In *Proceedings of the genetic and evolutionary computation conference (GECCO-2006)* (pp. 1629–1636). New York: ACM
25. Miettinen, K. (1999) *Nonlinear multiobjective optimization*. Boston: Kluwer.
26. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley.
27. Bandaru, S., & Deb, K. (2011). Towards automating the discovery of certain innovative design principles through a clustering based optimization technique. *Engineering Optimization*, 1–31.
28. Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24, 419–440.
29. Deb, K., & Srinivasan, A. (2007). *Innovization: Innovative design principles through optimization*. Kangal, IIT Kanpur: Indian Institute of Technology Kanpur. Kangal:2005007.
30. Messac, A., & Mattson, C. A. (2004). Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA Journal*, 42(10), 2101–2111.
31. Dabbeeru, M. M., & Mukerjee, A. (2008). Discovering implicit constraints in design. In *Third international conference on design computing and cognition*. Atlanta, GA, USA: Springer.
32. Kannan, B. K., & Kramer, S. N. (1994). An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, 116(2), 405–411.
33. Martinec, T. M., Berkovich, S. G., & Schulten, K. J. (1993). Neural gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4, 558–569.
34. Hölttä, K., Tang, V., & Seering, W. (2003). Modularizing product architectures using dendrograms. In *Proceedings 14th international conference on engineering design*.



35. Simpson, T. W. (1998). *A concept exploration method for product family design*. Georgia Tech University, Department of Mechanical Engineering.
36. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.
37. Tenenbaum, J. B., Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319–2323.
38. Saul, L. K., & Roweis, S. T. (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4, 119–155.
39. Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 1, 585–592.
40. Khire, R., Wang, J., Bailey, T., Lin, Y., & Simpson, T. W. (2008). Product family commonality selection through interactive visualization. In *ASME 2008 international design engineering technical conferences and computers and information in engineering conference. Proceedings of the DETC*.
41. Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500), 2323–2326.
42. Geng, X., Zhan, D. C., Zhou, Z. H. (2005). Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(6), 1098–1107.
43. Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G., & Koudas, N. (2002). Non-linear dimensionality reduction techniques for classification and visualization. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 645–651). New York: ACM.

# Chapter 6

## Multi-objective Optimisation of a Family of Industrial Robots

Johan Ölvander, Mehdi Tarkian and Xiaolong Feng

**Abstract** Product family design is a well recognised method to address the demands of mass customisation. A potential drawback of product families is that the performance of individual members are reduced because of the constraints added by the common platform, i.e., parts and components need to be shared by other family members. This chapter presents a framework where the product family design problem is stated as a multi-objective optimisation problem and where multi-objective evolutionary algorithms are applied to solve the problem. The outcome is a Pareto-optimal front that visualises the trade-off between the degree of commonality (e.g., number of shared components) and performance of individual family members. The design application is a family of industrial robots. An industrial robot is a mechatronic system that comprises a mechanical structure (i.e., a series of mechanical links), drive-train components (including motors and gears), electrical power units, and control software for motion planning and control.

---

J. Ölvander (✉) · M. Tarkian  
Department of Management and Engineering, Linköping University,  
581 83 Linköping, Sweden  
e-mail: johan.olvander@liu.se

M. Tarkian  
e-mail: mehdi.tarkian@liu.se

X. Feng  
ABB Corporate Research, 721 78 Västerås, Sweden  
e-mail: xiaolong.feng@se.abb.com

## 6.1 Introduction

Product family design based on a modular architecture, has for a long time been a well recognised method to address the demands of mass customisation. Based on the concept of product platforms, it is possible to deliver products within a short time frame and have a broad product range to meet specific customer requirements while maintaining low development and manufacturing costs. A possible drawback of product families is that the performance of individual members are reduced attributable to the constraints added by the common platform, i.e., parts and components need to be shared by other family members (a trade performance or cost for commonality).

A product family is represented by a number of variant products sharing a common platform. The platform typically consists of a set of components, modules or manufacturing and assembly processes. Product family design can reduce cost because of the commonality between the variants, but there is always a trade-off between commonality and individually optimised performance [1, 2].

The design of a product family has been the subject of research for several years, and many approaches have been presented for various product domains, for a survey of different methods, see [3]. The major tasks can however be summarised as indicated in [4]:

- Design of the platform for a specified family
- Design of the family based on a specified platform
- Simultaneous design of both platform and family

The above design tasks typically result in a combinatorial optimisation problem including both continuous and discrete optimisation parameters.

Naturally, there is a trade-off between commonality and performance, e.g., a large product family with a high degree of commonality is expected to have lower cost and performance than if the same product family is based on a larger number of different components with low commonality between the family members. It is therefore natural to look at the problem as a multi-objective optimisation problem. The problem consists of a mix of discrete and continuous variables and the objectives and constraints are in the general case represented by non-linear functions where no analytical derivatives are available. Examples of optimisation methods that can handle this type of problems in general are genetic algorithms (GA) [5] and specifically multi-objective genetic algorithms (MOGA) [6]. There are also many examples in the literature where GAs and MOGAs are applied to platform design problems, see [4, 7, 8].

### 6.1.1 Multi-Objective Optimisation

Real engineering design problems are usually characterised by the presence of many conflicting objectives, and hence it is natural to look at them as multi-objective optimisation problems. As most optimisation problems are multi-objective to their

nature, there are many methods available to tackle these kinds of problems. Generally, a multi-objective optimisation problem can be handled in four different ways depending on when the decision-maker articulates his or her preference on the different objectives; never, before, during or after the actual optimisation procedure, [9].

In the first approach (no preference articulation) the objective function is not depending on the preference of the decision-maker. Examples are the min–max formulation and global criterion method where the objective is to minimise the distance to the utopian solution, see [10].

The most common way of handling problems with multiple objectives is by priori articulation of the decision-makers preferences. This means that before the actual optimisation is conducted the different objectives are aggregated to one single figure of merit. The aggregation could be done in many ways, one of the most common methods being the weighted sum approach.

The third approach (during) includes iterative methods where the decision-maker articulate his or her preferences as the optimisation process evolves.

In the final approach the search is not for one optimal solution but for the complete Pareto-optimal front, which visualises the trade-off between the objectives. In this approach one could consider multiple run methods where the optimisation algorithms are run several times to sample points on the Pareto front. Alternatively a population-based method could be used that is capable of identifying the Pareto front within one single optimisation run. One of the most efficient techniques to obtain a good spread of Pareto optimal solution is to employ multi-objective evolutionary algorithms [6]. Within this chapter, one problem is solved using a multiple run approach based on a standard GA, whereas the second problem is solved using NSGA-II [11].

### ***6.1.2 Product Development***

Product development is a special form of problem solving where a number of frequently unclear objectives have to be balanced without violating a set of constraints. Based on this statement it could be said that design is essentially an optimisation process, as stated by Herbert Simon [12] already in 1967. By employing modern modelling, simulation and optimisation techniques, vast improvements could be achieved, even in the conceptual part of the design process.

A great deal of research has been done in the field of product development leading to different design processes and methods. Various authors present different models of the design process, see refs [13–17]. They all describe a phase type process of different granularity with phases such as: planning, concept development, system-level design, detail design, testing and refinement, and production ramp up, using the nomenclature from [17].

In this chapter two different design cases from the field of industrial robotics are studied. The first study considers the conceptual phase, whereas the second is more focused on system-level development and detailed design.

### ***6.1.3 Chapter Outline***

The remaining of the chapter has the following outline. First a brief introduction to industrial robotics is given describing both technical aspects as well as the robot design process. Thereafter a generic mathematical framework for product family optimisation is presented. In the following sections two multi-objective product family design problems are studied. The first considers a conceptual kinematics design study whereas the second focusses on detailed dynamic design. Eventually the chapter closes with discussions and conclusions.

## **6.2 Industrial Robotics**

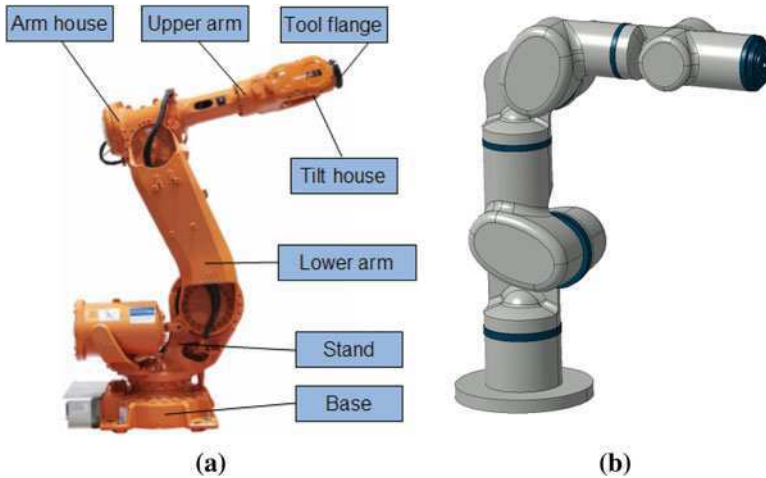
An industrial robot is a typical mechatronic system, consisting of a mechanical structure, or normally referred to as robot manipulator, and a controller. The mechanical structure of an industrial robot consists of a base followed by a series of structure links. The motion of each link is generated by a drive-train comprising permanent magnet electric motors and precision gears. Major components of the robot controller are power units, rectifier, transformer, axis computers and a high-level computer for motion planning and control. An example of a traditional serial manipulator is shown in Fig. 6.1a and a modular industrial robot is shown in Fig. 6.1b. These two types of robots will be used as examples in this chapter.

The traditional robot manipulator is of type IRB6640-185/2.8 from ABB and it consists of six rotational joints. Joint 1 between stand and base, joint 2 between lower arm and stand, joint 3 between arm house and lower arm, joint 4 between upper arm and arm house, joint 5 between tilt house and upper arm, and joint 6 between tool flange and tilt house.

The mechanical structure of the modular robot consists of a base followed by a series of modular structure parts. Each module consists of drive-train components (servo actuator, combining precision harmonic drive gearing with highly dynamic AC servo motors) integrated to one module. The modular robot also has six degrees of freedom.

Performance of the industrial robots are normally characterised by

- Number of degrees of freedom (number of rotational joints)
- Reach or shape of workspace
- Payload handling capacity
- Axis speeds or cycle time measured by some typical cycles
- Some pose or path accuracy measures



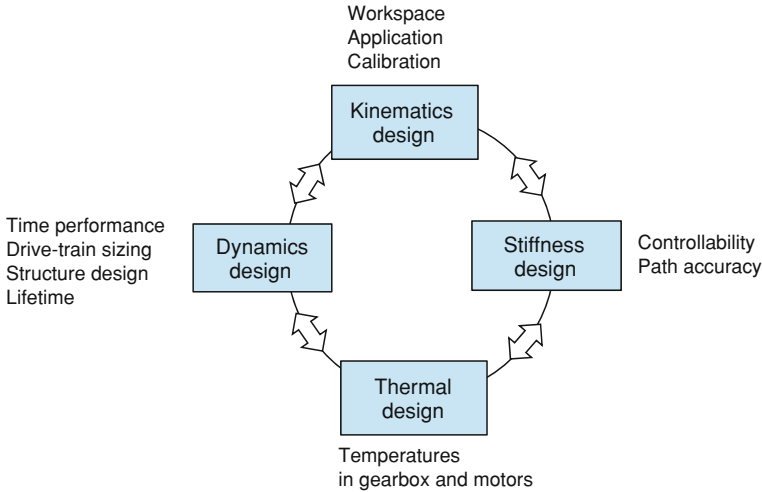
**Fig. 6.1** Example of industrial robots with serial kinematics, (a) IRB6640-185/2.8 from ABB (b) a modular industrial robot

Design of an industrial robot is a very complex process involving tremendous modelling and simulation efforts. Major steps in robot manipulator design are; kinematics design, dynamics design, thermal design, and stiffness design, see Figs. 6.2 and 6.1. In addition, the design of a robot manipulator is an iterative process because of the following complex issues: serial connection of robot links, configuration-dependent robot performance, multiple-domain nature of the robot system including mechanical, electrical, software, and control sub-systems.

### 6.2.1 Kinematics Design

Kinematics design is the first step in the design process of a robot manipulator. The ultimate goal of the kinematics design is to decide manipulator configuration, the number of robot joints, the link lengths, the link offsets, and the arm rotational limits in order to meet the performance requirement specification. In practice, manipulator configuration and the number of robot joints are normally chosen in the first place, and then link lengths, the link offsets, and the arm motion limits are determined in a more quantitative manner based on some robot kinematics performance measure. The different performance measures for kinematics design could be divided into the following groups:

- Based on maximum reach of a robot manipulator, a robot performance measure normally used by industrial robot manufacturers
- Shape or volume of workspace, or reach envelop of the wrist centre point (WCP) of the robot
- More physics-based robot kinematics design measures based on the manipulability of the robot.



**Fig. 6.2** Workflow for industrial robot design process

The more physics-based kinematics performance measures of a robot manipulator are often obtained by examining the Jacobian matrix of the manipulator [18].

### 6.2.2 Dynamics Design

As a first step in the dynamics design stage drive-train components, robot configuration and structure components are preliminarily designed. Tool centre point (TCP) acceleration or axis rotational speed and acceleration at a large number of predefined points in robot workspace, are normally used as design criteria. Based on this initial design preliminary mass properties of the robot are obtained. Detailed design of both structure components and drive-train components are then conducted based on motion simulations, where actual trajectories are run. This stage require detailed models of the geometry of the robot (CAD models) and dynamic simulation models incorporation rigid-body dynamics, dynamic models for the drive train, as well as control algorithms and software for motion planning and trajectory generation.

### 6.2.3 Stiffness Design

Stiffness of a robot manipulator is essential to ensure required accuracy-related performance, for example, path tracking accuracy and the accuracy and settling time when the tool centre point of a robot manipulator approaching a posture in its workspace. Two basic approaches are normally used for stiffness design, based on Eigen-frequency analysis and based on robot path tracking accuracy simulations. In both approaches, flexible multi-body modelling of a robot manipulator is required.

### 6.2.4 Thermal Design

Thermal design is essential because the thermal problems normally are noticed in the prototyping phase of a new robot development. Thermal design concerns structure cooling design and drive-train components thermal sizing. The design criteria are often that a number of critical temperatures in motors and gears may not exceed their maximum allowed temperatures. Both stiffness and thermal design are also essential steps in the iterative design process.

## 6.3 Product Family Formulation

This section presents a formal framework for product family optimisation, and a family of industrial robots will be used as an illustrative example. The robot is a rather modular product in its design including among others the following components/modules for the mechanics: stand, lower arm, upper arm and wrist. The robot typically can move in six degrees of freedom, which means that the robot structure has six axes. Each axis is driven by an electric motor and a gearbox which adds another set of modules. In the common product platform there are component libraries, e.g., motor and gearbox library to fill the different module slots.

The product family design problem is thus to find the optimal number of components in each library, to parameterise each component, and finally to select components for the modules for each member of the product family. The objectives could for example be to minimise the cost of the entire family and maximise the performance of all family members. Thus it is a multi-objective problem which yields a trade-off between degree of commonality and product performance. Figure 6.3 provides a visualisation of the problem.

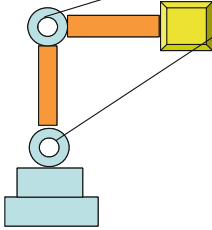
In order to minimise the cost a high degree of commonality is desired as it gives economics of scale in design, purchasing, logistics and maintenance. However, one possible drawback with a high degree of commonality is deterioration in performance for individual family members, as they cannot be optimised separately to meet the requirements for that particular family member.

### 6.3.1 Generic Problem Representation

Consider a product family with  $n_p$  different product variants denoted  $p^1, p^2, \dots, p^{n_p}$  or simply  $p^i$ , ( $i = 1, 2, 3, \dots, n_p$ ), where  $i$  is the index of a product member in the family. Each product  $p^i$  is composed of a series of modules  $M_1^i, M_2^i, \dots, M_{n_m}^i$  or simply  $M_j^i$ , ( $j = 1, 2, \dots, n_m$ ), where  $j$  is the index to a module in the product  $i$ . The product  $p^i$  is defined by the choice of components to fill each module slot  $M_j^i$ . This selection is represented by the integer variable  $m_j^i$ . For each module,  $j$ , there are  $c_j$



Physical product



Modular architecture

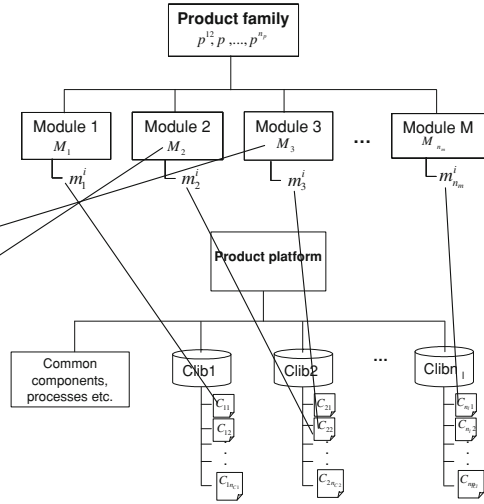


Fig. 6.3 Problem visualisation

different possible choices of components to fit in that slot. Thus  $m_j^i$  could be seen as a matrix where each row contains the choice of components for a particular family member. For example, the entries in column  $j$  are integers,  $l$ , less than or equal to  $c_j$  that refers to different components. Each component selected by  $m_j^i$  is described by a set of design variables  $\mathbf{x}^{c_{jl}} = [x_1^{c_{jl}}, x_2^{c_{jl}}, \dots, x_{n_{xc}}^{c_{jl}}]$ , where  $c_{jl}$  is the index to the component selected by  $m_j^i$ , and  $n_{xc}$  is the number of design variables required to describe component  $c_{jl}$ . In total, there is  $n_c$  number of different components in the common platform.

The cost for the entire product family could be obtained by summing up the cost for all modules for every product variant in the family, as expressed in Eq. (6.1)

$$\text{Cost} = \sum_i \sum_j n^i C_j^i(m_j^i, \mathbf{x}^{c_{jl}}, n_j^c) \tag{6.1}$$

$n^i$  is the expected sales volume of product  $p^i$ . The relationship  $C_j^i(m_j^i, \mathbf{x}^{c_{jl}}, n_j^c)$  represents that the cost for a particular module is a function of what component is selected ( $m_j^i$ ), how it is parameterised ( $\mathbf{x}^{c_{jl}}$ ) and how many units ( $n_j^c$ ) there are of that component in the entire product family.

A performance metric for a product family should consider the performance of all family members. One approach is thus to employ a weighted sum to aggregate the performance of each family member to an overall performance metric, where the weight expresses that some variants might be more important than others, see Eq. (6.2).

$$\text{Perf} = \sum_i w_i \text{perf}^i \left( m_j^i, \mathbf{x}^{c_{ji}} \right) \quad (6.2)$$

### 6.3.2 General Problem Formulation

Based on the problem representation, four general problem types are defined. In this section the structure of these problems will be outlined and later the problems will be explained in more detail for the two applications. The objectives for all problems are to minimise the cost and maximise the performance of the entire product family, as expressed in Eqs. (6.1–6.2).

**Problem One:** Given  $n_p$  products, select components  $m_j^i$  for each module from an existing set of predefined components, i.e., given the platform design the product variants.

$$\begin{aligned} \min_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Cost}^i(\mathbf{z}), \max_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Perf}^i(\mathbf{z}) \\ \mathbf{z} \in S_{p1} \\ \mathbf{z} = m_j^i, i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_m \end{aligned} \quad (\text{P1})$$

This is a combinatorial problem with  $n_p, n_m$  integer variables. The solution space  $S_{p1}$  expresses constraints such as which components could be selected for which module, and naturally constraints on performance and attributes of the different products, e.g., stress in the mechanical structure.

**Problem Two:** Given  $n_p$  products, select components  $m_j^i$  for each module, and determine how a predefined set of components should be parameterised, i.e., given the structure of the platform (the number  $c_j$  for each component library) define the components of the platform and design the product variants.

$$\begin{aligned} \min_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Cost}^i(\mathbf{z}), \max_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Perf}^i(\mathbf{z}) \\ \mathbf{z} \in S_{p2} \\ \mathbf{z} = \left[ m_j^i, \mathbf{x}^{c_{jl}} \right], i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_m, \\ l = 1, 2, \dots, c_j \end{aligned} \quad (\text{P2})$$

This problem consists of a set of integer variables  $m_j^i$  but also a set of variable vectors  $\mathbf{x}^{c_{jl}}$  predominantly containing continuous variables that describe the components. The solution space  $S_{p2}$  for this problem contains constraints inherited from  $S_{p1}$ , but also constraints regarding the parameterisation of the different components.

**Problem Three:** Given  $n_p$  products, select components  $m_j^i$  for each module, and determine the number of components available for each module and how they should be parameterised, i.e., design both the platform and the product variants

$$\begin{aligned} & \min_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Cost}^i(\mathbf{z}), \max_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Perf}^i(\mathbf{z}) \\ & \mathbf{z} \in S_{p3} \\ & \mathbf{z} = \left[ m_j^i, \mathbf{x}^{c_{jl}}, c_j \right], i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_m, \\ & l = 1, 2, \dots, c_j \end{aligned} \quad (\text{P3})$$

This problem is again a mixed integer programming problem with an extra  $n_m$  integer variables, which influence the number of variable vectors  $\mathbf{x}^c$ . Furthermore, as the structure of the platform is not given in this problem,  $S_{p3}$  will contain more constraints expressing rules for the structure of the product platform.

**Problem Four:** Given the demands of the customer, obtain a set of Pareto-optimal product families, i.e., (P3) is extended to also include the number of product variants that should be offered.

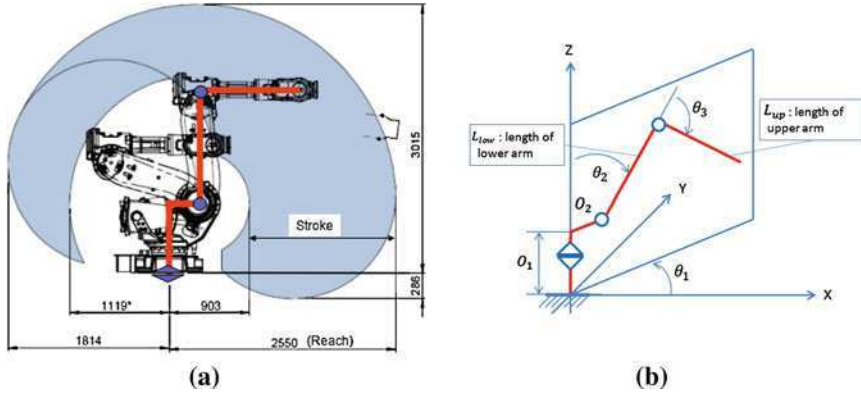
$$\begin{aligned} & \min_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Cost}^i(\mathbf{z}), \max_{\mathbf{z}} \sum_{i=1}^{n_p} \text{Perf}^i(\mathbf{z}) \\ & \mathbf{z} \in S_{p4} \\ & \mathbf{z} = \left[ m_j^i, \mathbf{x}^{c_{jl}}, c_j, n_p \right], i = 1, 2, \dots, n_p, j = 1, 2, \dots, n_m, \\ & l = 1, 2, \dots, c_j \end{aligned} \quad (\text{P4})$$

The problem has again the same structure and the solution space,  $S_{p4}$ , includes more constraints expressing the demands of the customer, e.g., required reaches and payloads for the different robots in the family.

In the following sections two different applications are studied and hence the product family design problem is discussed in more detail.

## 6.4 Optimal Kinematics Design

Most common commercial industrial robot manipulators have six degrees of freedom and have spherical wrist. The motion of such robot manipulators may be characterised by the translational motion of the wrist centre point (WCP) delivered by three main axes, and orientation of the centre point of robot tool interface flange delivered by three wrist axes normally intersecting at the WCP. In this work,



**Fig. 6.4** Kinematics definition for a serial manipulator, (a) Shape, reach, and stroke of the workspace of an ABB IRB6640-158/2.8 (b) Kinematics structure of main axes

a robot manipulator consisting of only the three main axes, shown in Fig. 6.4, is considered.

The three main axes are represented by three links, or the three bars in red colour, in Figs. 6.4a and b. The shape and stroke (difference between maximum and minimum reach) of the workspace is determined only by dimensions  $L_{low}$ , the length of the lower arm,  $L_{up}$ , the length of upper arm, the rotation angle bounds of joints 2 and 3 (for the shape), and the minimum allowed angle between link 2 and 3 (for the stroke). The rotation of joint 1 has no effect on the shape of the workspace. The offset  $O_1$  influences only the vertical position of the workspace and has no effect on the shape or reach of the workspace. The offset  $O_2$  influences the horizontal position of the workspace and thereby the reach. The dimensions  $L_{low}$  and  $L_{up}$  are referred to as the lengths of lower and upper arms, respectively and  $O_1$  and  $O_2$  are two offsets defining the location of joint 2.

As performance measure we will evaluate the stroke (the offset between maximum reach and minimum reach of the WCP of a robot, see Fig. 6.4a and an overall manipulability measure robot averaged over all pre-defined configurations in the entire workspace. Ideally, this type of overall performance measure should be independent of the size of the workspace, for example, reach. In this work, the overall manipulability index is obtained, based on the manipulability measure  $w$  [18], averaged over the entire workspace [19]

$$w_{overall} = \frac{(dx \times dz) \times \sum w}{(L_{low} + L_{up} + o_2)^5} \tag{6.3}$$

where  $w_{overall}$  is the overall manipulability index,  $dx$  and  $dz$  are dimensions of a grid,  $\sum w$  is the summation of manipulability measure over all grids that are inside of the workspace envelope.

Based on the Jacobian matrix of the manipulator,  $J(\theta)$ , Yoshikawa proposed the following quantitative manipulability measure ( $w$ ) of a robot manipulator [18]

$$w = \sqrt{\det(J(\theta)J^T(\theta))} \quad (6.4)$$

where  $w$  is a scalar value,  $\det$  denotes the determinate of a matrix and  $T$  the transpose of a matrix. Please observe that the Jacobian matrix varies with the pose of the manipulator, so it needs to be recalculated for every position of the manipulator.

### 6.4.1 Optimisation Problem Formulation

Consider the design of a product family with  $n_{\text{rob}}$  number of robots where each robot  $R_i$  should have the reach,  $r_i$ . Thus the vector  $\mathbf{R} = [r_1, r_2, \dots, r_{n_{\text{rob}}}]$  represents the reach for all robots in the family. The objective is to maximise the performance index for all robots in the product family. In this study, we will evaluate stroke and the overall performance index,  $w_{\text{overall}}$ .

The problem is parameterised so that  $x_l$  represents the length for the lower arm of the first robot,  $L_{\text{low}1}$ . The length of the upper arm,  $L_{\text{up}1}$ , of robot one could then be determined so that the reach is fulfilled, i.e.,

$$L_{\text{up}1} = r_1 - L_{\text{low}1} - o_2 \quad (6.5)$$

where the  $L_{\text{low}1}$  and  $L_{\text{up}1}$  are the lengths of lower arm and upper arm of robot 1 (the number in the subscript represents the robot index) respectively.

For the second robot, it could either share the upper or the lower arm of the first robot or have its own arms. For the first two cases the arm lengths could be calculated based on the arm that is shared and the required reach. For the last case the arm lengths need to be determined which introduces one more parameter. The selection is modelled using three binary decision variables,  $x_{21}$ ,  $x_{22}$  and  $x_{23}$ . If  $x_{21}$  equals 1, robot 2 shares the lower arm of robot 1. If  $x_{22}$  is 1, robot 2 shares the upper arm, and if  $x_{23}$  is 1, robot 2 shares no arms with robot 1. Only one of  $x_{21}$ ,  $x_{22}$  and  $x_{23}$  could be equal to 1.  $x_{24}$  represents the length of the lower arm of robot 2, should  $x_{23}$  equal 1. The arm lengths could be calculated according to the following.

$$\begin{aligned} L_{\text{low}2} &= x_{21}L_{\text{low}1} + x_{22}(r_2 - L_{\text{up}1}) + x_{23}x_{24} \\ L_{\text{up}2} &= r_2 - L_{\text{low}2} - O_2 \end{aligned} \quad (6.6)$$

The quotient,  $k_i$ , between the length of the lower and the sum of the lower and upper arms for each robot should be between the limits  $k_{\text{min}}$  and  $k_{\text{max}}$ . This constraint will assure that the length of the lower and upper arms are somewhat similar which will guarantee good manipulability of the robot. For the majority of industrial robots on the market,  $k$  is between 0.4 and 0.5. However for exceptional cases values of 0.35 and 0.6 exist.  $k_i$  is calculated as.

$$k_i = \frac{L_{\text{low}i}}{L_{\text{low}i} + L_{\text{up}i}} \quad (6.7)$$

where the  $L_{lowi}$  and  $L_{upi}$  are the lengths of lower arm and upper arm of robot  $i$ , respectively.

The maximum number of arms (or arm modules),  $n_{arms}$  in the robot family constitutes a measure of commonality and is a number between  $n_{rob} + 1$  and  $2n_{rob}$ . If  $n_{arms} = n_{rob} + 1$  all robots share one arm with at least one other robot, and if  $n_{arms} = 2n_{rob}$  no robot share arms with another robot.

Thus the number of arms in the product family determines how many robots that needs to share arms, i.e., how many  $x_{i3}$  that could be non-zero. This is expressed in the inequality below.

$$\sum_{i=2}^{n_{rob}} x_{i3} \leq n_{arms} - n_{rob} - 1 \quad (6.8)$$

Thus the product family optimisation problem could be described according to the equation below.

$$\begin{aligned} & \max \sum_{i=1}^{n_{rob}} \lambda_i \text{Perf}_i(x) \\ & \sum_{j=1}^3 x_{ij} = 1, \forall i \in \{2, 3, \dots, n_{rob}\} \\ & \sum_{i=2}^{n_{rob}} x_{i3} \leq n_{arms} - n_{rob} - 1 \\ & k_{\min} \leq k_i(x) \leq k_{\max}, i = 1, 2, \dots, n_{rob} \\ & \frac{x_1}{r_1 - O_2} \in [k_{\min}, k_{\max}] \\ & x_{ij} \in [0, 1], i = 2, 3, \dots, n_{rob}, j = 1, 2, 3 \\ & \frac{x_{i4}}{r_i - O_2} \in [k_{\min}, k_{\max}], i = 2, 3, \dots, n_{rob} \end{aligned} \quad (6.9)$$

The problem has  $n_{rob}$  continuous variables and  $3(n_{rob} - 1)$  binary variables. The objective function is non-linear and the problem has  $n_{rob} + 2$  linear constraints and  $2n_{rob}$  non-linear constraints. The input to the optimisation is a vector of required reaches  $\mathbf{R} = [r_1, r_2, \dots, r_{n_{rob}}]$ , a vector of weighting factors  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{n_{rob}}]$ , where in the simplest case all  $\lambda_i = 1/n_{rob} \cdot n_{rob}$  and  $k_{\min}$  and  $k_{\max}$ . Finally the maximum number of arms of the product family needs to be specified. If the problem is solved for different settings on  $n_{arms}$  the trade-off between performance and commonality (number of arms) will be obtained. Hence, here we use a single objective formulation (6.9) but solve the problem for increasing degree of commonality will yield a Pareto-optimal front showing the trade-off between performance and commonality.

Furthermore, the product family problem described in (6.9) is of problem type three (P3) as describe in Sect. 6.3.2. The problem does not have exactly the same parameterisation as (P3), but it has the same structure, i.e., to determine the

component for each individual robot (to decide on  $m_j^i$ ), to design the actual components ( $\mathbf{x}^{cjl}$  = length of the arms), and determine the number of components for each module ( $c_j$  = number of upper and lower arms).

### 6.4.2 Optimisation Method

The optimisation algorithm used in this application is an elitist GA with a population size of 40 individuals which is run for 100 generations. The design solutions are represented using a mixed chromosome including  $n_{\text{rob}}$  continuous variables ( $x_1$  and  $x_{i4}$ ) and  $3(n_{\text{rob}}-1)$  binary variables ( $x_{ij}$ ), as described in (6.9). The crossover operation is uniformed crossover where the real values are crossed with each other using blend crossover and the integer values are chosen randomly from either the mother or the father. After crossover a repair algorithm makes sure that only valid chromosomes are generated, i.e., the constraints in (6.9) need to be fulfilled before evaluating the chromosome. If an individual violate any of the  $n_{\text{rob}}$  constraints expressed in (6.9), it is modified in a random fashion until it does no longer violate any constraints. Selection is made using Roulette wheel selection where the fitness values are obtained using a linear ranking of the raw objective score.

### 6.4.3 Optimisation Results

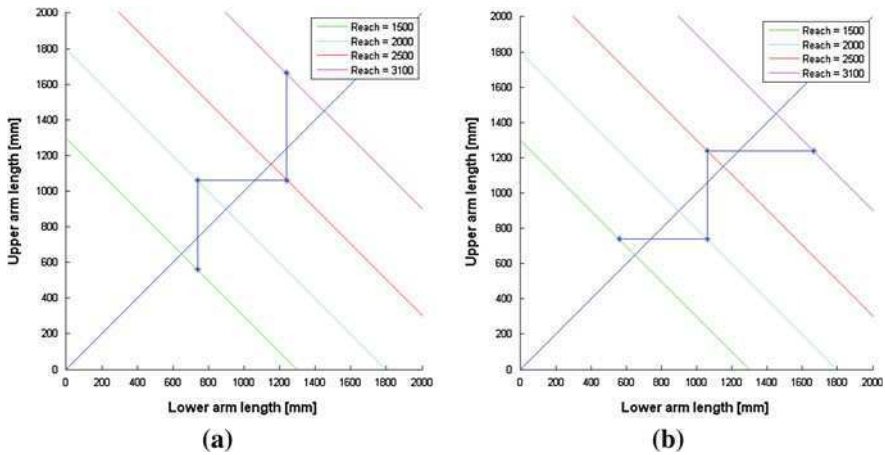
In this section the problem stated in (6.9) will be solved for two different objective functions, namely stroke and manipulability. For each objective the problem is solved for different degrees of commonality in order to obtain a Pareto front showing the trade-off between performance and number of parts in the product platform.

First, consider the case where we should design a robot family consisting of  $n_{\text{rob}} + 1$  arms, i.e., the maximum degree of commonality, with the relative stroke as the objective. The relative stroke is the stroke of each robot divided by the maximal stroke obtained if lower and upper arms have the same length. The stroke is calculated based on the length of the upper and lower arms of the robot, the minimal angle between the arms, and the offset ( $O_2$ ) between link 1 and 2, see Fig. 6.4.

It turns out that solving problem (6.9) gives two possible solution topologies, a family with 2 lower and 3 upper arms or 3 lower and 2 upper arms. Both have a mean stroke of 95.06% but one has smallest lower arm of 737.8 mm, while the other has smallest lower arm of 562.4 mm. However, these two designs are very similar, as shown in Table 6.1. In fact, they are mirror images obtained by mirroring one design over the line  $k = 0.5$ , see Fig. 6.5. In the figure, the parallel diagonal lines with different colours represent possible arm lengths for each reach when the offset is 200 mm. The stars represent optimal arm lengths of individual family members.

**Table 6.1** Optimal product families with stroke objective

<i>Family 1: 2 lower and 3 upper arms</i>				
Reach [mm]	1500	2000	2500	3100
Configuration [mm] [lower arm; upper arm]	[738; 562]	[738;1062]	[1238;1062]	[1238;1662]
$k$	0.568	0.410	0.538	0.427
Stroke [%]	64.70	64.90	70.94	69.29
Relative stroke [%]	95.28	92.02	98.41	94.52
<i>Family 2: 3 lower and 2 upper arms</i>				
Reach [mm]	1500	2000	2500	3100
Configuration [mm] [lower arm; upper arm]	[562; 738]	[1062; 738]	[1062; 1238]	[1662; 1238]
$k$	0.433	0.590	0.462	0.573
Stroke [%]	64.71	64.88	70.94	69.28
Relative stroke [%]	95.30	92.01	98.41	94.51

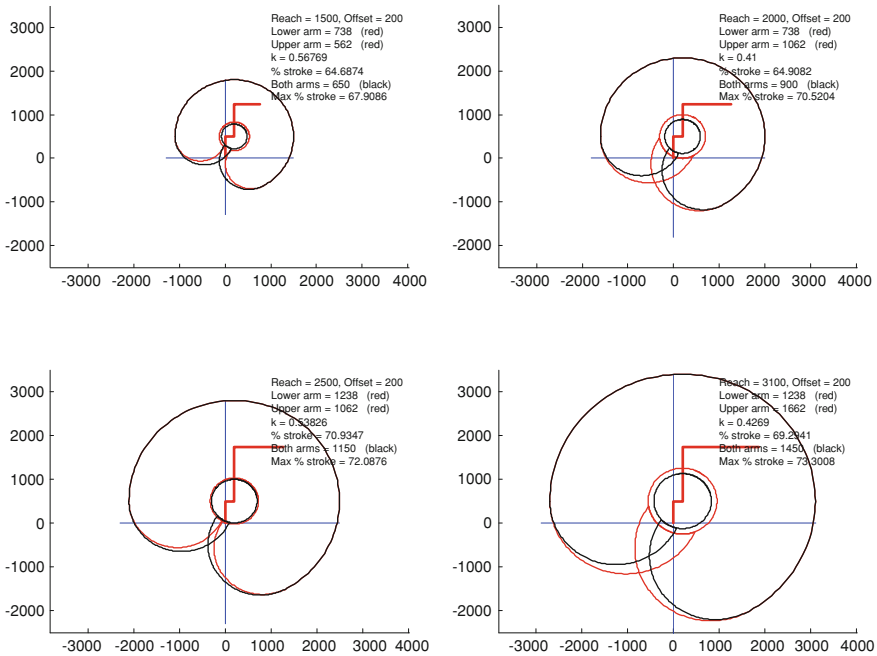


**Fig. 6.5** Two equivalent optimal product families, **a** with two lower and three upper arms and **b** with three lower and two upper arms

The kinematics performance of robot family (a) is shown in Fig. 6.6. For each robot shown, the shape of workspace in red line is compared with that of an ideal robot that has the same reach and the same length of lower and upper arms (in black). It is evident that for each robot in the family, an acceptable compromise in the shape of workspace has been made.

Now let us consider more arms in the common platform. First, if we allow one extra arm and solve the problem stated in (6.9), the average relative stroke could be increased to 97.6%. With a third arm the optimum stroke would be 98.27%, and finally, using eight arms there are no common components and the relative stroke will be 100% for all robots. These results are presented in Table 6.2.





**Fig. 6.6** Workspace and stroke visualisation of the four robots in the product family in Fig. 5a, minimum angle of 25° between lower and upper arms is used

If the objective is changed to manipulability and problem (6.9) is again solved for different degrees of commonality the results in Table 6.3 are obtained. A visualisation of the manipulability measure over the work space for a product family with the highest degree of commonality, i.e., family E is shown in Fig. 6.7. In the figure, dark red colour represents points with high manipulability whereas dark blue represents points with low manipulability.

In Figs. 6.8, 6.9 and 6.10 through Fig. 6.11 the different product families are visualised in the same graphs. Solid lines and stars represent families obtained with manipulability as objective, whereas circles and dashed lines represent families obtained with stroke as the objective. The solid diagonal lines represent the different reaches (actually reach minus offset  $O_2$ ), and each marker (star or circle) represent one individual robot, with the smallest robot to the left.

Studying Fig. 6.8 through Fig. 6.11, it is obvious that the manipulability measure longer upper arms, compared to the stroke measure. Figure 6.12 shows the trade-off between performance (relative stroke and manipulability) and commonality. The graph could be looked upon as Pareto front showing the trade-off between performance and cost; where on the vertical axis it is shown how the performance increases with the number of components (costs) of the common platform.

**Table 6.2** Optimisation results with stroke as the performance metric

Reach [mm]	1500	2000	2500	3100	Objective
<i>Family A, 5 module platform</i>					
Configuration {lower arm; upper arm} [mm]	{738; 562}	{738; 1062}	{1238; 1062}	{1238; 1662}	
k [-]	0.568	0.410	0.538	0.427	
Relative stroke [%]	95.28	92.02	98.41	94.52	95.06
Normalises w [-]	0.12	0.145	0.161	0.176	0.15
<i>Family B, 6 module platform</i>					
Configuration {lower arm; upper arm} [mm]	{650; 650}	{957; 843}	{957; 1343}	{1557; 1343}	
k [-]	0.5	0.532	0.416	0.537	
Relative stroke [%]	100	98.89	92.99	98.50	97.60
Normalises w [-]	0.126	0.146	0.161	0.174	0.152
<i>Family C, 7 module platform</i>					
Configuration {lower arm; upper arm} [mm]	{650; 650}	{900; 900}	{1262; 1038}	{1262; 1638}	
k [-]	0.50	0.50	0.549	0.435	
Relative stroke [%]	100	100	97.47	95.60	98.27
Normalises w [-]	0.126	0.149	0.159	0.176	0.153
<i>Family D, 8 module platform</i>					
Configuration {lower arm; upper arm} [mm]	{650; 650}	{900; 900}	{1150; 1150}	{1450; 1450}	
k [-]	0.50	0.50	0.50	0.50	
Relative stroke [%]	100	100	100	100	100
Normalises w [-]	0.126	0.149	0.164	0.177	0.154

**Table 6.3** Optimisation results with overall manipulability measure as the performance metric

Reach [mm]	1500	2000	2500	3100	Objective
<i>Family E, 5 module platform</i>					
Reach [mm]	1500	2000	2500	3100	
Configuration {lower arm; upper arm} [mm]	{725; 575}	{725; 1075}	{1254; 1075}	{1225; 1675}	
k [-]	0.541	0.391	0.523	0.415	
Relative stroke [%]	98.15	88.86	99.40	92.82	94.81
Normalises w [-]	0.121	0.144	0.161	0.175	0.151
<i>Family F, 6 module platform</i>					
Configuration {lower arm; upper arm} [mm]	{617; 682}	{783; 1017}	{1283; 1017}	{1283; 1617}	
k [-]	0.475	0.514	0.403	0.526	
Relative stroke [%]	99.29	95.63	96.44	96.51	96.97
Normalises w [-]	0.126	0.147	0.158	0.177	0.152
<i>Family G, 7 module platform</i>					
Configuration {lower arm; upper arm} [mm]	{617; 683}	{863; 937}	{1201; 1099}	{1201; 1699}	
k [-]	0.475	0.480	0.522	0.414	
Relative stroke [%]	99.28	99.55	99.45	92.68	97.74
Normalises w [-]	0.126	0.149	0.163	0.174	0.153
<i>Family H, 8 module platform</i>					
Configuration {lower arm; upper arm} [mm]	{617; 683}	{863; 937}	{1097; 1203}	{1384; 1516}	
k [-]	0.474	0.480	0.477	0.477	
Relative stroke [%]	99.28	99.55	99.41	99.43	99.41
Normalises w [-]	0.126	0.149	0.165	0.178	0.155

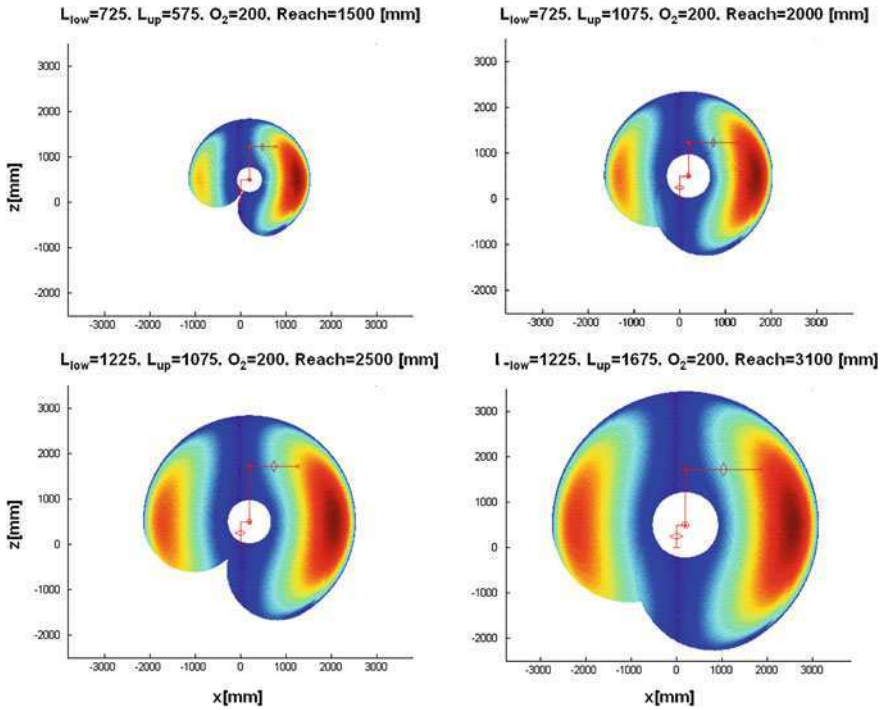
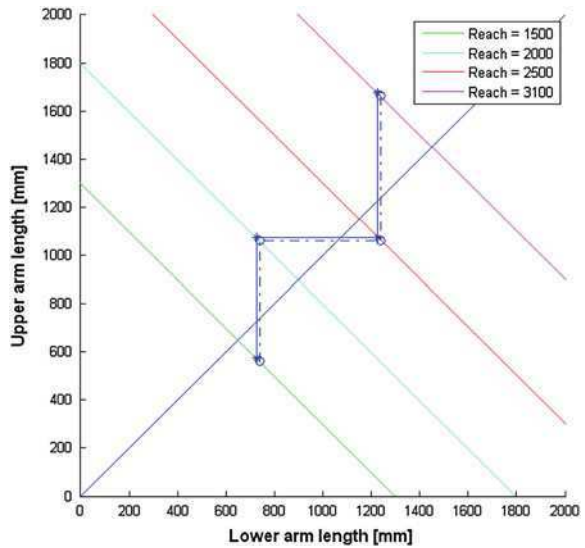
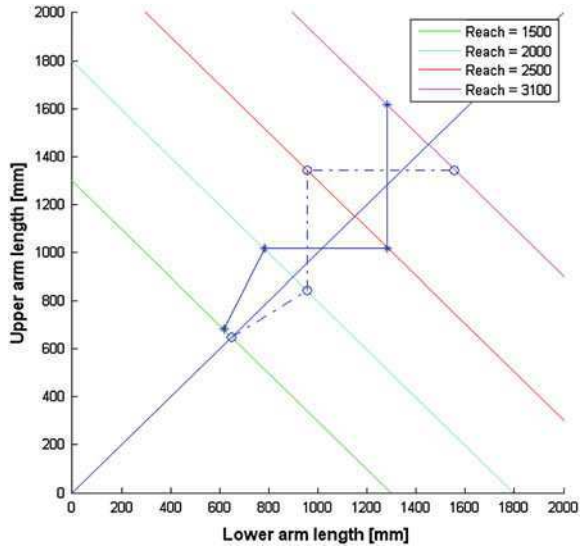


Fig. 6.7 Visualisation of the manipulability measure for family E

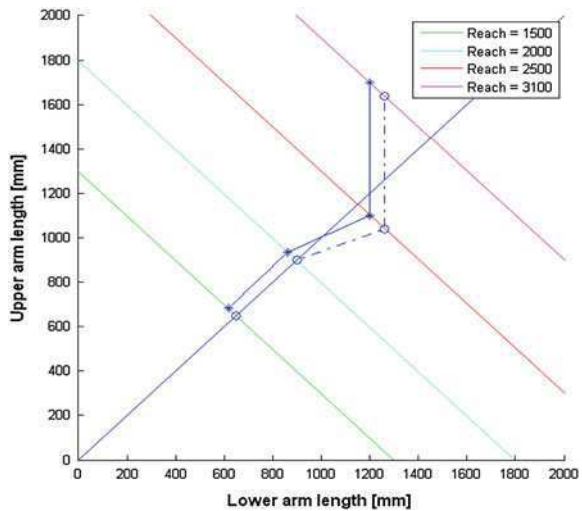
Fig. 6.8 Visualisation of family A (circles) and E (stars)



**Fig. 6.9** Visualisation of family B (*circles*) and F (*stars*)



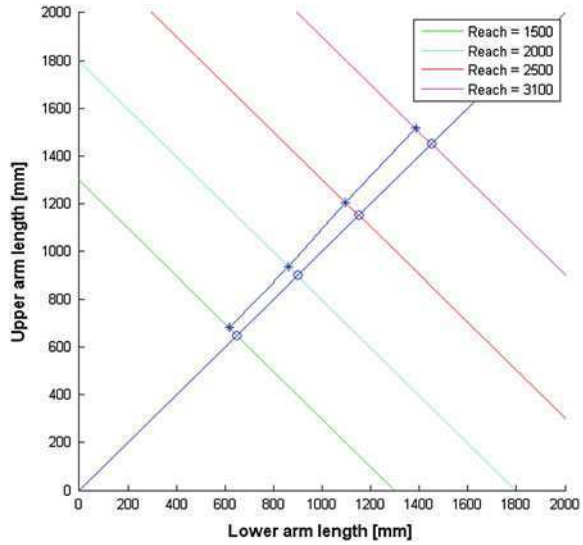
**Fig. 6.10** Visualisation of family C (*circles*) and G (*stars*)



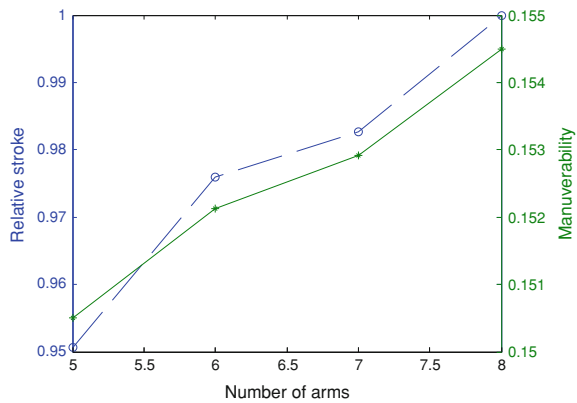
### 6.5 Optimal Dynamic Design

The natural step after performing the kinematics design is to continue with the dynamic design. However, in the dynamic design stage rather extensive simulation models are required, and it is necessary to combine models from several disciplines in order to obtain a holistic view of the system. Furthermore, in order to achieve an optimal design, the product must be treated as a complete system instead of developing the different subsystems independently. For all the various

**Fig. 6.11** Visualisation of family D (*circles*) and H (*stars*)



**Fig. 6.12** Pareto fronts showing the trade-off between performance (stroke and manipulability) and number of arms in the common platform

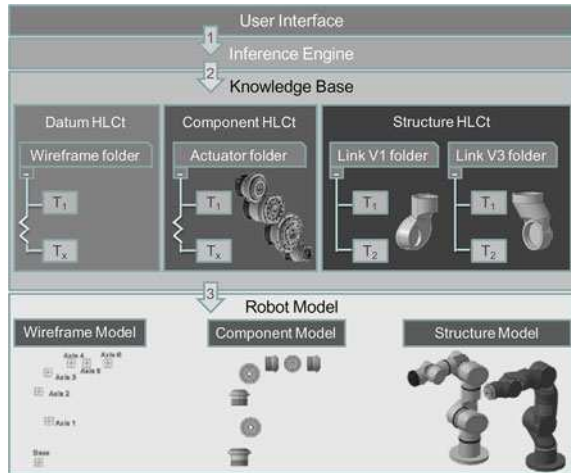


domains of robot design, the geometry plays a significant role as input provider. Therefore, it is essential to develop efficient methods for CAD-modelling in order to integrate the geometric models into the optimisation framework.

### 6.5.1 Geometric and Dynamic Modelling

One outcome of modularity within a product family is increased external variety and decreased internal variety, e.g., the number of components. The same principle is adopted here for the geometric modelling of the product family, where the geometries are saved as templates and instantiated with unique internal design

**Fig. 6.13** Relations between the robot models and the HLCT libraries



variables. Thereby the number of model variants is effectively increased by sharing few geometric templates between the model variants. By importing such High Level CAD template geometries (HLCT), the robot is defined in three steps. First, the number of axes is determined in a user interface, defining the skeleton model of the robot, which is stored in the Datum HLCT and placed according to the logic of the inference engine. The type of component HLCT for each axis is then decided and an appropriate structure, from Structure HLCT, is chosen in the final step. The model of the robot is thereby transformed from an empty initial model into a complete model in three steps, as shown in the Fig. 6.13.

To simulate the dynamic properties of a robot, a dynamic model has to be utilised. The dynamic model in the outlined framework is made using an in-house simulation tool developed at ABB. The motion of the rigid manipulator can be described by

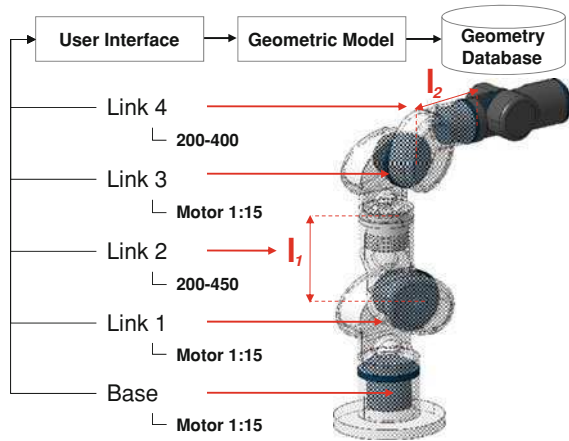
$$Q = M(q) \cdot \ddot{q} + V(q, \dot{q}) + G(q) + B(\dot{q}) \quad (6.10)$$

where  $M$  is the inertia matrix,  $V$  is the vector of Coriolis and centrifugal forces,  $G$  is a vector of gravity forces and  $B$  is a vector of viscous friction forces,  $q$  is a vector of generalised coordinates e.g., angular position of each joint in the manipulator. For more information about dynamic models and trajectory planning for robots, see [20, 21].

The geometric and dynamic models are seamlessly integrated through a user interface, where various engineering aspects of the robot are analysed concurrently. Furthermore the geometrical and dynamical aspects of the robot components are stored in a component library.

Although commercial CAD tools are well suited to generate high fidelity geometry for various analyses tools, they often require extensive time to update the model after a parameter change. Therefore, a geometry database has been created to reduce the simulation time required for generating the sought after geometry

**Fig. 6.14** The geometry database is created by altering the design variables of the geometric model through the user interface



**Table 6.4** Payload and reach requirements of the robot family

	Robot 1	Robot 2	Robot 3	Robot 4
Reach [mm]	760	860	960	1060
Payload [kg]	3	5	10	14

during the optimisation. To produce the database the morphology and topology of the robot structure are varied, and the geometric properties such as mass, centre of gravity and inertia are stored in a geometric database, see Fig. 6.14.

All links subjected to parametric modification are coloured white. For link 1 and 3 the morphology is altered by modifying the actuator parameters. These are modified by altering discrete values ranging from 1 to 15 representing different actuator choices. A change of actuator will initiate a topologically import of the actual detailed actuator geometries from the component HLCts. The logic stated in the inference engine will then update the internal design variables of the link housing the actuator so that it will fit. For link 2 and 4 the morphology is modified by varying the lengths between 200–450 mm and 200–400 mm respectively.

Creating the geometric database is essentially to span the largest possible size of the product platform as it contains all possible actuators and link configurations. Hence for this application the product family optimisation problem will be of type (P1) as describe in Sect. 6.3.2, i.e., the largest possible common platform has been created and the problem is to design the family members and by doing so also deciding the common platform.

### 6.5.2 Problem Formulation

The problem formulation consists of concurrently optimising the performance and commonality level of a product family consisting of four robots. The optimisation



variables are choice of servo actuators for axes 1, 2 and 3 as well as lengths of link 2 and 4, amounting to overall 16 optimisation variables for the entire product family. The four robots' reach and payload requirements are tabulated in Table 6.4. Moreover, various trajectories have been implemented for each robot.

The problem is multi objective with the performance and commonality being the objective functions. The performance objective,  $f_1$ , is the sum of cycle time (CT) and the robot weight (Weight) for all four robots,  $i$ . The performance objective is to be minimised, hence low weight and cycle time is preferred. The commonality objective is to maximise number of common components in the robot family, summing up both links and actuators. The unit for the commonality objective  $f_2$  is in percentage ranging from 0 to 100.

$$\begin{aligned}
 f_1 &= \sum (\lambda_1 \text{CT}_i + \lambda_2 \text{Weight}_i) \\
 f_2 &= 100 \cdot \left( k_1 \frac{\sum \text{Link}_{\text{shared}}}{\sum \text{Link}} + k_2 \frac{\sum \text{Actuator}_{\text{shared}}}{\sum \text{Actuator}} \right) \\
 i &= 1, 2, 3, 4
 \end{aligned} \tag{6.11}$$

$\lambda_i$  and  $k_i$  are weighting factors where  $\sum k_i = 1$ . The weighting factors  $k_i$  have been chosen to prioritise link commonality prior to actuator commonality. The weight and cycle time are normalised and  $\lambda_i$  are chosen for even weighting.

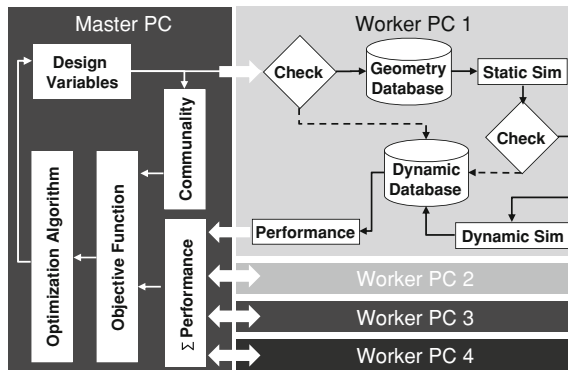
The presented problem consists of discrete variables, and the two objectives and the constraints are represented by non-linear functions where no analytical derivatives are available. Therefore a multi-objective GA [6] has been chosen as the optimisation algorithm. In references [4] and [7], GAs and MOGAs are applied to solve platform design problems. In this chapter, NSGA-II are used as the optimiser [11].

### 6.5.3 Computational Framework

In previous work [22] the robot design framework was utilised to design an optimal modular robot for a specific task and a set of requirements. A product family optimisation involves a higher computational burden, as all members in the family need to be evaluated. Generally also the number of evaluations increases because of the increased number of optimisation variables.

To shorten the optimisation procedure, several modifications have been made to the earlier framework. As stated previously the mass properties are stored in a geometry database prior to the optimisation. Moreover during the actual optimisation all dynamic and static simulation results are stored in a dynamic database. When a previously evaluated design is suggested by the optimisation algorithm, the results will be retrieved from the dynamic database, thereby skipping both the static and dynamic simulations. Naturally one should not evaluate a solution that has been visited before. However, for the product family problem there might be

**Fig. 6.15** To speed up the evaluation process, the robot family is concurrently computed on four worker workstations



**Table 6.5** Population size and generations settings

	Opt. 1	Opt. 2	Opt. 3	Opt. 4
Individuals	40	60	100	300
Generations	200	200	200	200

parts of a solution (some family members) that have been evaluated, and hence need not be simulated again.

Furthermore, the analyses of all robots in the family are executed in parallel, in a distributed framework where the master PC sends out design variables to four worker PCs. These will then return the performance objective for each family member, whereas the commonality objective is calculated within the master PC, see Fig. 6.15.

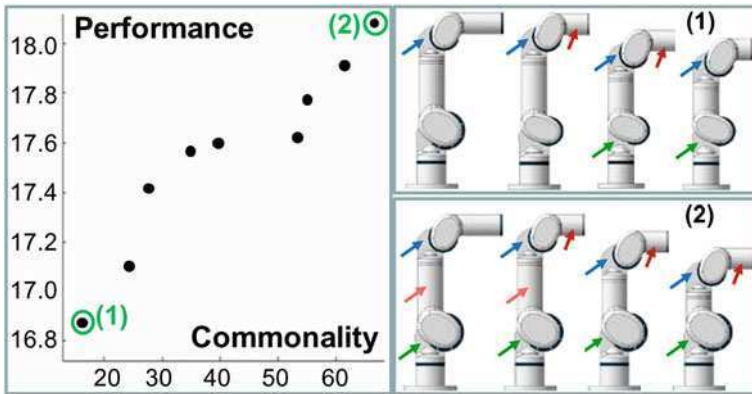
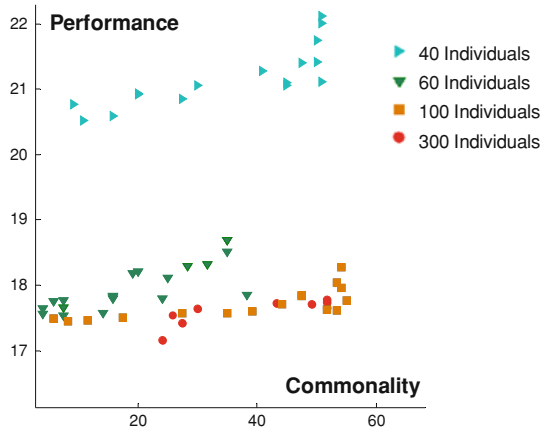
The static simulation calculates the torque required at different robot workspace positions in order to withstand the gravitational forces. If the configuration does not meet the gravitational forces i.e., the actuators are too weak, the performance objective is given a penalty value. The dynamic simulation will not be initiated, and hence the computational burden is reduced.

The geometrical data from the geometry database model is used to parameterise the matrix and vectors in Eq. (6.10). The equation of motion for the robot is implemented in a dynamic simulation program which also includes path and trajectory planner and calculates properties such as torques, accelerations, speed and cycle times.

### 6.5.4 Optimisation Results

The outlined optimisation framework has been utilised to search for the Pareto frontier of the presented problem. The performance objective is to be minimised with the aim of decreasing weight and cycle time, while the commonality

**Fig. 6.16** Pareto frontiers for 40, 60, 100, and 300 individuals



**Fig. 6.17** Pareto front showing the trade-off between performance and commonality (left), and the product family, with best performance (1) and highest commonality (2) (right). The shared modules are marked with an arrow

objective is to be maximised to increase component sharing. Table 6.5 shows the individual and generation settings that have been evaluated in order to estimate the probable Pareto frontier.

Final results of the Pareto frontiers, up to the 5th rank, are visualised in Fig. 6.16. Not surprisingly, as the number of individuals increase, the Pareto frontiers move to more optimal locations. However this movement is progressively minimised, suggesting that about 100 to 300 individuals is sufficient for finding the optimal-Pareto frontier.

Judging from the 1st order Pareto frontier in Fig. 6.17, the algorithm is well suited to find solutions for both high commonality and performance. In the robot family with best performance, the highest reach robot has more powerful actuators,

while the smaller robots are capable in performing the pre-set trajectories with smaller actuators. However the commonality level is low. For the robot family with highest commonality, the actuators and arm lengths are selected in order to maximise commonality, consequently punishing the overall performance.

## 6.6 Discussions and Conclusions

This chapter presents a mathematical framework for optimal design of product families. Furthermore it illustrates how multi-objective evolutionary algorithms could be employed to solve the product family problem for industrial robot applications. A set of generic mathematical formulation for the product family problem is presented, and later two of them are adapted for the specific applications.

In general, the product family problem has multiple conflicting objectives, e.g., the degree of commonality is in conflict with the performance or customisation of individual family members. Furthermore, the problem often contains of a mixture of both discrete and continuous variables. The discrete variables are needed to handle the combinatorial nature of the problem, i.e., to determine which module to use with which family member, whereas the continuous variables are needed to design the modules themselves. Finally, for real world problems a set of different computer aided engineering tools are needed to evaluate design proposals, and hence it is seldom possible to obtain any derivatives for the objective and constraint functions. These are all reasons for employing multi-objective evolutionary algorithms for this type of problems.

In this chapter two different design applications are considered. The first example is from the conceptual design stage and involves kinematics design of a traditional industrial robot, whereas the second example is from a more detailed design stage and considers dynamic design of a modular industrial robot. Hence the described methods are applied at different stages of the product development process.

For the kinematics design case two different performance measures are used, namely stroke and manipulability. The optimisation variables are the length of the lower and upper arms of the robots. The results obtained show the trade-off between performance (stroke and manipulability) and number of components in the common platform. For this application the Pareto front is obtained by sampling points on the Pareto front by running multiple GA optimisations.

Stroke is a rather simplistic performance measure which is easy and fast to calculate. From a stroke perspective only the difference between the length of the upper and lower arm is of importance, and in order to maximise the stroke the lower and upper arm should have equal length. As only the difference between the arms are of importance, multiple optima could be obtained by mirroring the design over the line  $k = 0.5$ . In order to calculate the manipulability measure the workspace need to be meshed and the Jacobian calculated for each discrete point. An overall kinematics performance measure is then obtained by aggregating the

manipulability for each point. Hence, manipulability is more complicated to compute as compared with stroke.

From a manipulability perspective it is advantageous with a longer upper arm compared to stroke, and hence the graphs in Fig. 6.8 through Fig. 6.11 are shifted upwards for the manipulability measure. The majority of the serial manipulators manufactured today have in fact a slightly longer upper arm, which favour the applicability of manipulability as a measure for kinematic performance.

For the dynamic design problem a framework for product family design of modular robots is presented. Complex products generally have an intricate dependency between geometry and dynamic performance. Knowledge based engineering is utilised to manage the framework complexity which automatically creates the geometric CAD model and seamlessly integrates the dynamic simulation model. Thereby, a framework for multidisciplinary design optimisation has been established.

Based on the framework a product family optimisation problem has been set up where the combination of discrete component selections invokes changes in the geometric model, as well as in the dynamic simulation model. The links and drive train of a modular robot family has been optimised, and a Pareto frontier generated using a multi-objective GA. The Pareto front shows the trade-off between commonality of the common platform and performance of the individual family members. It could be seen how individual performance is deteriorated when commonality is increased.

It seems natural that there is a trade-off between commonality and performance, especially as a high degree of commonality also implies low cost. However, this is not always the case. One means of obtaining a high degree of commonality for the modular robot is to use the largest actuators for all robots. Hence a high degree of commonality could be obtained by always sharing the best (or most expensive) component. This is obviously not cost effective. For the modular robot this problem was handled by including the weight in the performance measure and thus penalising too large actuators. An alternative is to introduce a commonality measure that better reflects the cost of the family.

For future work, continuous variables for the actuators, e.g., limits on torque and angular velocity can be taken into consideration during the optimisation. This will facilitate life time estimation of the drive train components. Furthermore, FE-analysis needs to be incorporated in order to evaluate the stress levels in the links, and hence facilitate detail design of the link cross sections. However by increasing the complexity of the problem formulation, the optimisation framework needs to undergo further modifications. One approach is to introduce several hierarchical layers for the optimisation, where one algorithm optimises the overall layout, whereas detailed component optimisation is performed by another algorithm.

Another future improvement is the development of cost measures, considering both development and manufacturing costs, cost saving as a result of commonality, component costs and life cycle costs of the robot family.

## References

1. Fellini, R., Kokoloras, M., Papalambros, P., & Perez-Duarte, A. (2005). Platform selection under performance bounds in optimal design of product families. *Journal of Mechanical Design*, *127*, 524–535.
2. Nelson, S., Parkinson, M., & Papalambros, P. (2001). Multicriteria optimization in product platform design. *Journal of Mechanical Design*, *123*, 199–204.
3. Jose, A., & Tollenaere, M. (2005). Modular and platform methods for product family design: literature analysis. *Journal of Intelligent Manufacturing*, *16*, 371–390.
4. Fujita, K., & Yoshida, H. (2004). Product variety optimization simultaneously designing module combination and module attributes. *Concurrent Engineering: Research and Applications*, *12*(2), 105–118.
5. Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.
6. Deb, K. (2001). *Multi-Objective Optimization using Evolutionary algorithms*. New York, NY: Wiley and Sons Ltd.
7. Jiao, J., Zhang, Y., & Wang, Y. (2007). A generic genetic algorithm for product family design. *Journal of Intelligent Manufacturing*, *18*(2), 233–247.
8. Simpson, T., D’Souza, B. (2004). Assessing variable levels of platform commonality within a product family using a multiobjective genetic algorithm. *Concurrent Engineering: Research and Applications*, *12*(2) pp 199–129.
9. Andersson, J. (2000). *A Survey of Multi-objective Optimization in Engineering Design, Technical Report LiTH-IKP-R-1097, Department of Mechanical Engineering*. Linköping, Sweden: Linköping University.
10. Steuer, R. (2001). *Multiple criteria optimization: Theory, computation and application*. New York: John Wiley & Sons, Inc.
11. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, *6*(2), 181–197.
12. Simon, H. (1969). *The Sciences of the Artificial*. Cambridge: MIT Press.
13. Cross, N. (2000). *Engineering design methods* (3rd edition). Chichester, UK: John Wiley & sons.
14. Pahl, G., Beitz, W. (1996). *Engineering Design—A Systematic Approach*. London: Springer-Verlag.
15. Suh, N., (2001). *Axiomatic Design—Advances and Applications*. New York: Oxford University Press.
16. Ullman, D. (1992). *The Mechanical Design Process*. New York: McGraw-Hill Inc.
17. Ullrich, K.T., Eppinger, S.D. (2000), *Product design and development* (2nd Edition). New York: McGraw-Hill Inc.
18. Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *International Journal of Robotics Research*, *4*(2):pp. 3–9.
19. Feng, X., Holmgren, B., Ölvander, J. (2009). Evaluation and optimization of industrial robot families using different kinematic measures. In *proceedings of ASME Design Automation Conference*. San Diego, August 30–September 2.
20. Sicilano, B. (2001) *Modeling and Control of Robot Manipulators*. London: Springer Verlag.
21. Spong, W. Mark & Vidyasagar M. (1989), *Robot Dynamics and Control*. New York: John Willey & Sons Inc.
22. Tarkian, M., Ölvander, J., Feng, X., Petterson, M. (2009). Design automation of modular industrial robots. In *proceedings of ASME Design Automation Conference*. San Diego, August 30–September 2.

# Chapter 7

## Multi-objective Optimisation and Multi-criteria Decision Making for FDM Using Evolutionary Approaches

Nikhil Padhye and Kalyanmoy Deb

**Abstract** In this chapter, we methodologically describe a multi-objective problem solving approach, concurrently minimising two conflicting goals—average surface roughness— $Ra$  and build time— $T$ , for object manufacturing in Fused Deposition Method (FDM) process by usage of evolutionary algorithms. Popularly used multi-objective genetic algorithm (NSGA-II) and recently proposed multi-objective particle swarm optimisation (MOPSO) algorithms are employed for the optimisation purposes. Statistically significant performance measures are employed to compare the two algorithms and approximate the Pareto-optimal fronts. To refine the solutions obtained by the evolutionary optimisers, an effective mutation-driven hill-climbing local search is proposed. Three new proposals and several suggestions pertaining to the issue of decision making in the presence of multiple optimal solutions are made. The overall procedure is integrated into an engine called MORPE—multi-objective rapid prototyping engine. Sample objects are considered and several case studies are performed to demonstrate the working of MORPE. Finally, a careful investigation of the optimal build orientations for several considered objects is done on selected basis and a trend is discovered,

---

N. Padhye  
Department of Mechanical Engineering, Massachusetts Institute of Technology,  
Cambridge, MA 02139 USA

K. Deb  
Department of Mechanical Engineering, Indian Institute of Technology Kanpur,  
Kanpur 208016, Uttar Pradesh, India  
e-mail: deb@iitk.ac.in

N. Padhye (✉)  
Laboratory of Manufacturing and Productivity, Massachusetts Institute of Technology,  
Building 35, Room 135, Cambridge, MA 02139, USA  
e-mail: npdhye@mit.edu

which can be considered highly useful for various practical rapid prototyping (RP) applications.

## 7.1 Introduction

Rapid prototyping (RP) or layered manufacturing refers to processes in which a component is fabricated by layer-by-layer deposition of material from 3D computer-assisted design models. It is an emerging technology which is becoming increasingly important in the market today. RP is playing a significant role in development of new products and for effecting cost reductions by enabling speedy development of prototypes.

Today there exist a multitude of RP techniques. Common examples of RP techniques are fused deposition method (FDM), stereolithography (SLA), selective laser sintering (SLS), laminated object manufacturing (LOM), 3D printing and direct metal deposition (DMD). The advent of these technologies has made it possible to fabricate prototypes directly from Computer -Aided Design (CAD) models. The prototypes can be checked for the feasibility of a design concept and prototype verification.

The first step in the RP cycle is creation of a geometric model using CAD tool. This is followed by determination of suitable deposition orientation, slicing, generation of material deposition paths, part deposition and post-processing operations. Many of these steps can be done automatically by the RP machine, but usually part deposition orientation is selected by the user. Part orientation has significant effect on build time and surface quality [1]. For some RP methods, e.g. FDM, build orientation also effects the support structure requirement.

While using any RP method, an obvious desire is to manufacture components with low surface roughnesses ( $Ra$ ) and build times ( $T$ ) and a systematic methodology to determine an orientation is required. This chapter proposes a novel approach to search for optimal build orientations, while simultaneously minimising  $Ra$  and  $T$  with respect to the FDM process. The minimisation of the considered objectives is conflicting in nature which leads to set of trade-off solutions with varying  $Ra$  and  $T$ . Further, in presence of such trade-off points the issue of decision making, i.e. selecting one orientation from a set of available optimal orientations becomes important which is addressed in this chapter. Interestingly, post-optimal analysis of obtained trade-off solutions, for various objects considered in this study, provides a deeper insight into FDM process and leads to development of knowledge via. optimisation.

The entire procedure is automated using a developed software—*multi-objective rapid prototyping engine* (MORPE). The software tool is developed for FDM system and is easily modifiable for other RP techniques. MORPE incorporates two evolutionary algorithms elitist non-dominated sorting genetic algorithm (NSGA-II) and multi-objective particle swarm optimisation (MOPSO) to perform



optimisation, variable slicing module to carry out slicing of a solid model and computing subsequently ( $Ra$ ,  $T$ ) at any given orientation, and inbuilt tools like attainment surface estimator and hypervolume calculator to arrive at results of statistical importance.

The rest of the chapter is organised as follows: [Sect. 7.2](#) reviews various studies carried out in past in context to build orientations. [Section 7.3](#) provides a multi-objective problem formulation for FDM process. [Section 7.4](#) develops a systematic approach to address the multi-objective optimisation task. This section discusses the variable slicing procedure and popular multi-objective evolutionary optimisers (NSGA-II and MOPSO). Then, an introduction to statistically comparable performance measures is made. Finally, a description on mutation-driven hill-climbing local is provided. In [Sect. 7.5](#) three decision making techniques are proposed for selecting a favourable build orientation. [Section 7.6](#) validates the proposed approach through several simulations. The results and discussions are carried out in [Sect. 7.7](#). This section also provides an insight into the decision making issue and innovative design principles are deciphered via. post-optimal analysis. Finally, conclusions are made in [Sect. 7.8](#).

## 7.2 Related Works

Choice of build orientation for part fabrication in layered manufacturing has been an active area of research for more than a decade. Broadly speaking, the goals are to minimise fabrication time (or cost) and maximise part accuracy. Usually these goals depend on the build orientation in accordance with the characteristics of the specific LM technology involved. The objective function formulation of such goals has been widely researched in the past. The measure employed for quantifying build time (or cost) is usually the number of layers [2–7] or, the part height when layer thickness is constant [1, 8, 9]. For LM technologies that require support structures during fabrication, the estimated support structure volume has also been applied as time-cost criterion [1, 10]. Post-processing time is another important cost factor that gets directly affected by the orientation choice and has been considered as a criterion for the orientation selection [8].

To account for the fabrication quality several indicators have been suggested: estimated surface roughness [3, 6, 11], weighted average surface roughness [12, 13], and total area of surfaces with estimated roughness above a certain limit [14]. Additionally, various criteria related with known sources of dimensional inaccuracies such as volumetric error [15, 16], the process planning or stair stepping error [2, 4, 11], and trapped volume error [17] for sintered layer parts have been proposed. Other quality related measures proposed are total overhang area [7, 10], the stability of the part structure during fabrication [4, 7], and perceived mechanical strength [18].

Once the measure to quantify time or cost (first objective) and surface quality (second objective) are decided, an appropriate search procedure is required to

discover favourable orientations—which optimise the considered objectives simultaneously. Since determination of these objectives at different orientations often involves substantial computational cost (i.e. rotation of CAD model and subsequent slicing) employment of efficient optimisation algorithms is desired. Depending on the part shape, model choice for surface quality or time the objective functions may exhibit discontinuity, rendering gradient-based methods ineffective. Evolutionary algorithms like genetic algorithms (GAs) have established themselves as potential candidates in addressing the challenges posed by real world problems where classical optimisation techniques fail [19]. In the past, evolutionary methods have also been applied for determination of optimal build orientations. A brief review of past works related to such optimisation tasks is in order as follows.

Previous studies in LM literature employing GAs for build orientation optimisation have mostly considered either single objective study or combination of multiple objectives into one using weighted approach. In [13] optimal build directions were explored using GA for different RP processes. Two goals, average weighted surface roughness (AWSR) and build times were combined to form a single objective and treated for minimisation. In [14], single objective GA was employed to determine optimal fabrication directions for LM processes so as to minimise the required post-machining region (RPMR) (as post-machining is often required to improve the surface quality). Here, authors developed an expression of the distribution of surface roughness and relation between the RPMR and fabrication direction. In [20] build orientations for parts fabricated with stereolithography were derived for optimising build time, surface roughness and post-processing times using single objective weighted approach. Other studies in literature that have also employed single objective weighted approach are [5, 7, 10, 11].

For the optimisation of a single criterion, like the part height, the average cusp height, or the total post-processing area, specific algorithms have been proposed in previous studies [9, 14, 21]. In [3, 9] authors selected orientations from a list of pre-selected set (determined by ranking of objectives and thus, allocating importance). Such a pre-selection mechanisms or minimisation of weighted single objective functions (discussed earlier) have well-known deficiencies and optimality of the solutions cannot be guaranteed [19]. However, more recently suitable multi-objective optimisation approaches using GAs, i.e. simultaneously minimising or maximising several goals, have been studied for different LM processes [6, 22–25]. Similar attempts to optimise multiple goals in this direction have been made [26–29].

Despite such studies, a systematic application of nature inspired heuristics addressing multi-objective optimisation, decision-making and knowledge discovery through optimisation is still missing. To address the existing shortcomings, we have chosen FDM process for which optimal build orientations are determined and post-optimal analysis is carried out.

### 7.3 Multi-objective Problem

Without loss of generality, we assume that the goal is to minimise  $m$  functions  $f_1, \dots, f_m$  of  $n$ -dimensional decision variables  $\phi$ . A decision vector  $\phi_1 \in S$  is called Pareto-optimal if there is no other decision vector  $\phi_2 \in S$  that dominates it. Any vector  $\phi_1$  is said to dominate  $\phi_2$ , if  $\phi_1$  is not worse than  $\phi_2$  in all of the objectives and it is strictly better than  $\phi_2$  in at least one objective. In case two solutions  $\phi_1$  and  $\phi_2$  do not dominate each other, we say that they are indifferent to each other or are non-dominated with respect to each other. To solve multi-objective problems, algorithms which can ensure a well distributed and well converged set of trade-off solutions are needed.

In current study the objectives of interest are average surface roughness  $Ra$  and total build time  $T$ . The formulation of bi-objective optimisation problem is done as follows:

$$\begin{aligned} &\text{Minimise } f_1 = Ra(\phi), \\ &\text{Minimise } f_2 = T(\phi), \\ &\text{where } \phi = \theta_x, \theta_y \end{aligned}$$

subject to:

$$\begin{aligned} 0 &\leq \theta_x \leq 180, \\ 0 &\leq \theta_y \leq 180. \end{aligned}$$

The decision variables of the problem are  $\theta_x$  and  $\theta_y$ , which represent the rotations about  $X$  and  $Y$  axes, respectively. Figure 7.1a, b describe the rotation scheme stated here by considering rotation of a facet or planar triangle (CAD model is represented in the form of facets and can be rotated by rotation of all the facets).

Computation of  $Ra$  and  $T$  has been borrowed from [6, 23, 24]. For FDM the surface roughness for each layer is a function of slice thickness  $t$  and build angle  $\theta$  as shown in Fig. 7.2. Note that  $\theta$  should not be confused with  $\theta_x$  and  $\theta_y$ . The  $Ra$  computation is done as follows: if build angle  $\theta$  is between  $0-70^\circ$ :

$$Ra (\mu\text{m}) = K \times \frac{t (\text{mm})}{\cos \theta}, \quad (7.1)$$

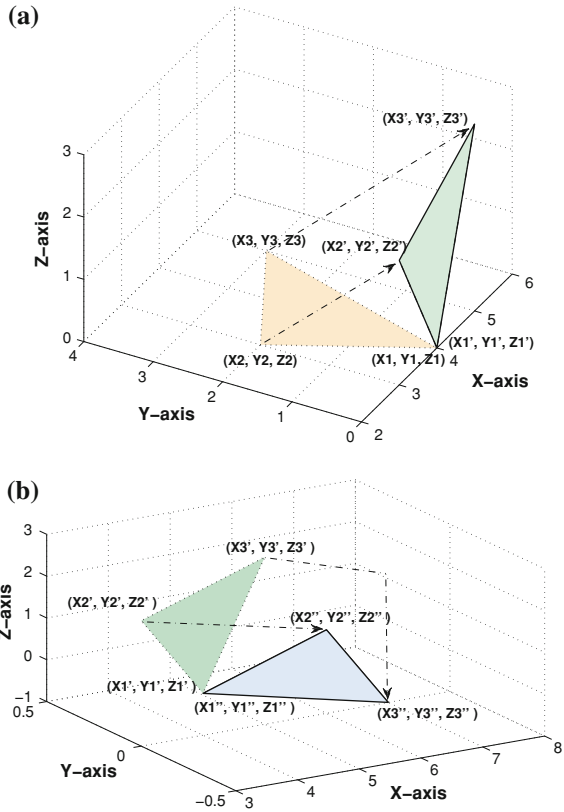
where  $K$  lies in (69.28–72.36). For build angle =  $90^\circ$  i.e. for a horizontal surface

$$Ra (\mu\text{m}) = 117.6 \times t (\text{mm}). \quad (7.2)$$

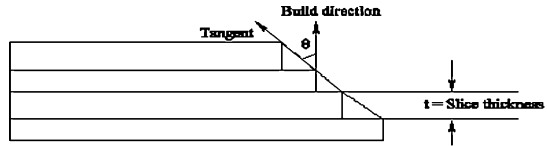
If build angle is greater than  $70^\circ$  and less than  $90^\circ$ :

$$Ra (\mu\text{m}) = \frac{1}{20} [90Ra_{70^\circ} - 70Ra_{90^\circ} + \theta(Ra_{90^\circ} - Ra_{70^\circ})], \quad (7.3)$$

**Fig. 7.1 a** Rotation of the facet about X axis by  $90^\circ$  with initial position  $(X_i, Y_i, Z_i)$  to final position  $(X'_i, Y'_i, Z'_i)$ .  
**b** Rotation of the facet about Y axis by  $90^\circ$  with initial position  $(X'_i, Y'_i, Z'_i)$  to final position  $(X''_i, Y''_i, Z''_i)$



**Fig. 7.2** Computation of  $Ra$  for a layer based on  $\theta$



where  $Ra_{70}$  and  $Ra_{90}$  are surface roughnesses at  $70^\circ$  and  $90^\circ$  build angles. Finally the average part surface roughness is calculated as:

$$Ra \text{ (}\mu\text{m)} = \frac{\sum Ra_i}{\text{total number of slices}}, \tag{7.4}$$

where  $Ra_i$  is the roughness of the  $i$ th trapezium, refer Fig. 7.2. The build time ( $T$ ) for a component is equal to the sum of build times of individual layers:

$$T_{\text{build}} = \sum_{i=1}^{N_{\text{layer}}} t_{\text{layer}_i} + N_{\text{layer}} \times t_{\text{zmove}} + \frac{N_{\text{layer}}}{k} \times t_{\text{wipe}}. \tag{7.5}$$

Here  $t_{\text{wipe}}$  is machine-specific time to wipe the nozzle and time to build  $i$ th layer is itself sum of times to lay the part  $t_{\text{part}_i}$ , support structure  $t_{\text{supp}_i}$  and table movement  $t_{\text{move}_i}$ .

$$t_{\text{layer}_i} = t_{\text{part}_i} + t_{\text{supp}_i} + t_{\text{move}_i}, \quad (7.6)$$

where  $t_{\text{part}_i}$  for  $i$ th layer is computed as follows:

$$t_{\text{part}_i} = \frac{A_{si}}{r_w} \times v. \quad (7.7)$$

Here,  $\frac{A_{si}}{r_w}$  is the material contained area in  $i$ th layer,  $r_w$  is the road width and  $v$  is the nozzle speed. It is assumed that because of support structure build time is negligibly affected as taken in [23]. The machine-specific parameters have been taken for Stratasys FDM 1650 system installed at IIT Kanpur equipped for prototyping with ABS plastic. It is worthwhile to mention that optimal build orientation directly depends on the model employed for computation of  $Ra$  and  $T$ , thus more realistic and accurate model is favourable. The focus in this study is to work with a reasonable model and demonstrate the multi-objective optimisation problem solving and decision making principles.

Since material laying deposition is assumed to be along  $z$ -axis, the rotation about  $z$ -axis is invariant for the computation of objectives. Thus, rotations only about  $x$ -axis and  $y$ -axis are considered.

## 7.4 Proposed Approach

The overall procedure is carried out by MORPE which comprises following modules: (a) adaptive slicing procedure, (b) multi-objective optimisers—NSGA-II and MOPSO, (c) performance comparison tools—hypervolume indicator and attainment surface approximator, (d) local search procedure, (e) decision making tools. Figure 7.3 portrays the working of MORPE.

### 7.4.1 Adaptive Slicing

The adaptive slicing procedure has been developed in MATLAB version R2007a. The optimisation routines and performance comparison measures are developed in C (gcc version 4.3.2) language. MATLAB code is compiled using MCR (MATLAB compiler runtime) version 7.6 and integrated with optimisation engine. The experiments reported in this study have been carried out on Intel single core 2.9 GHz, RAM-2.0 GB, Hard disk-80 GB, OS-Linux-Ubuntu-9.04, Computer architecture-32 bit. Figure 7.4 shows the flowchart for adaptive slicing module.

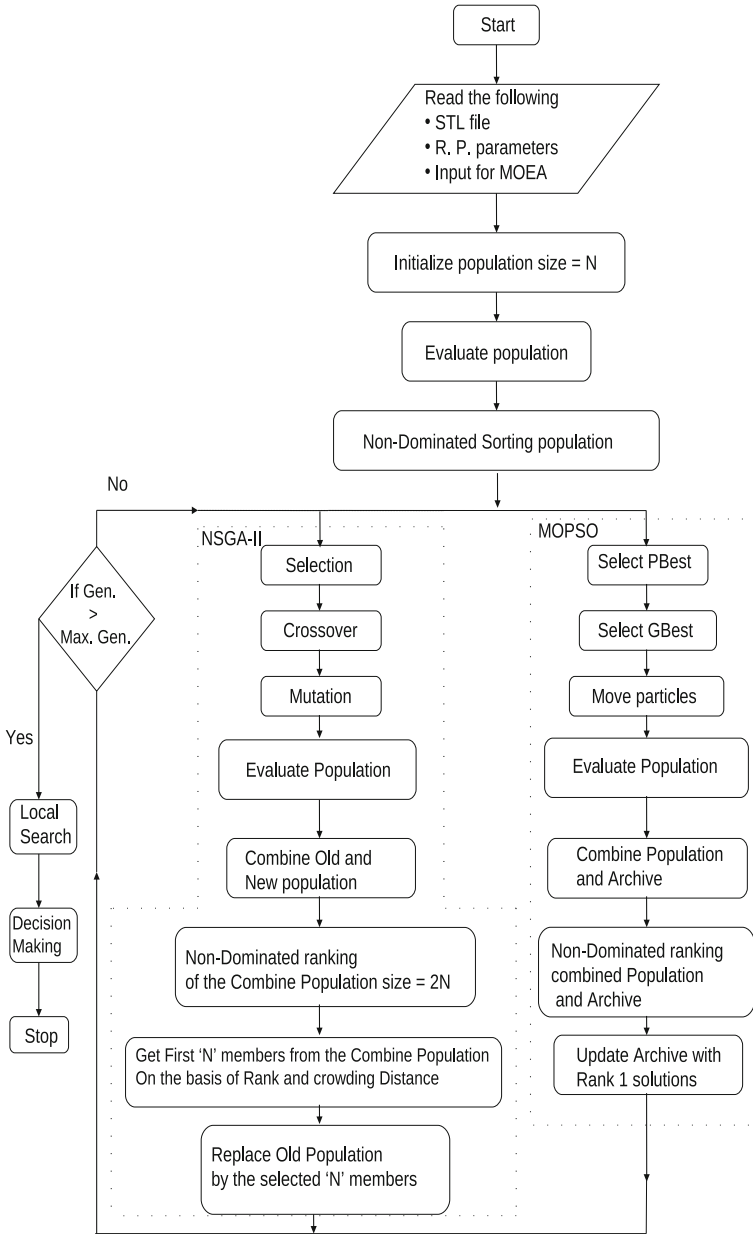
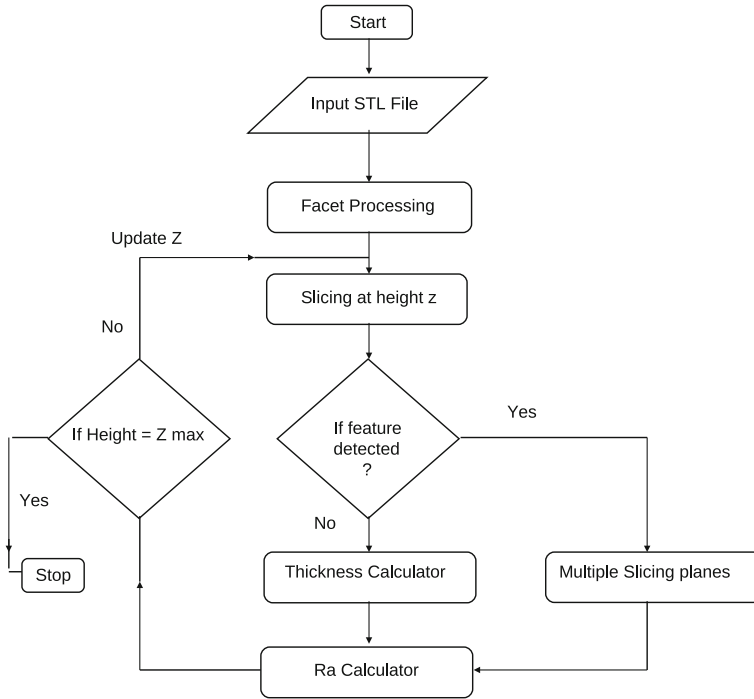


Fig. 7.3 Flowchart illustrating the working of developed MORPE procedure

In the past adaptive slicing procedure has been adopted to improve the surface quality and accuracy in LM processes. The adaptive procedure developed in this study is borrowed from [30]. Its salient features are discussed next.

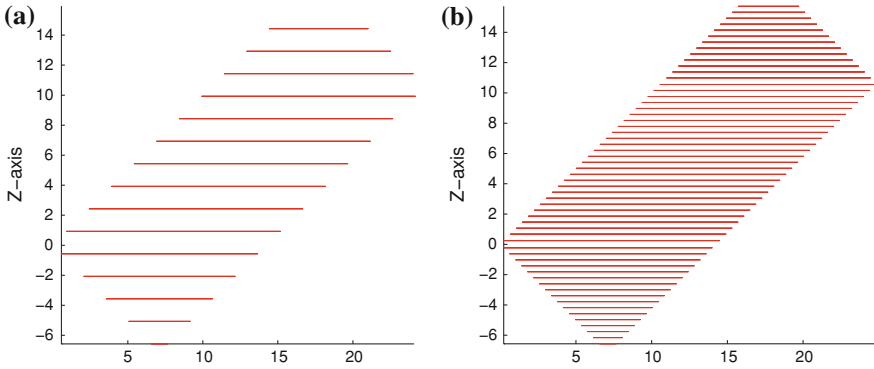


**Fig. 7.4** Flowchart for slicing procedure

The basic function of any slicing module is to generate two dimensional slices from a three-dimensional tessellated model. The input to the slicing engine is a STL file of the solid model under consideration. The STL file comprises coordinates of the triangular facets and their normals and entire solid model is represented by its constituent facets. A triangular facet comprises three points each of which is associated with  $(x, y, z)$  coordinate. For the purpose discussion we assume that  $z$ -axis denotes the vertical direction (direction of material deposition) and  $z_{\min}$  and  $z_{\max}$  denote the lowest and highest  $z$ -coordinates on the solid model.

For efficient slicing procedure an effective facet-processing technique is employed: first, facets are grouped into facet groups (based on same  $z_{\min}$ ) and then into sub-facet-groups (based on same  $z_{\min}$  and  $z_{\max}$ ). Next, slicing planes are considered at intervals from  $z_{\min}$  to  $z_{\max}$ . As facets have already been grouped and sub-grouped, as stated before, intersection of slicing planes with facets can be found efficiently, saving considerable amount of computational overhead. At each new slicing plane a check for a new feature using feature recognition rules is carried out. In case a new feature is detected a series of slicing planes is considered at small intervals so that feature informations are well captured.

If user has defined an upper limit on roughness value, then surface roughness is calculated at every slicing step for two adjacent slices and if the value exceeds the



**Fig. 7.5** **a** Constant height slicing procedure.  $Ra$  for object turns out to be 61.67 units. **b**  $Ra$  adaptive slicing procedure.  $Ra$  for object turns out to be 45.12 units

specified bound, position for the upper slicing planes are recomputed so that roughness value stays within the bounds.

In case of constant height slicing procedure, feature detection mechanism and bound-check on roughness are omitted. Here, the slicing planes are considered at specified interval height. Figure 7.5a, b compare constant and adaptive height slicing procedures for a cuboid oriented at an angle of  $45^\circ$  about  $X$ -axis. In constant slicing, interval height of 1.5 units is chosen. In adaptive slicing a maximum height interval of 1.5 units is allowed and bound on roughness value for each layer is set to 40.0 units. As observed, in adaptive case slice thicknesses are automatically calculated while keeping roughness values bounded. It should be noted that although roughness value for a layer can be controlled to stay below the chosen threshold but overall roughness  $Ra$  may be larger than the threshold.

## 7.4.2 Evolutionary Optimisers

Although there exist several multi-objective evolutionary algorithms (MOEAs) in literature, popularly used GA based NSGA-II and particle swarm based MOPSO optimisers have been utilised in this study. In the following paragraphs we briefly describe the working and salient features of these algorithms.

**MOPSO.** Particle swarm optimisation (PSO) is now a well established optimisation technique in variety of contexts. PSO is a population based technique, similar in some respects to other evolutionary algorithms, except that potential solutions (particles) move rather than evolve through the search space. PSO consists of several candidate solutions called particles each of which has a position and velocity, and experiences linear spring-like attractions towards two attractors:

1. the best position attained by that particle so far (particle attractor or personal best  $pbest$ );



2. the best of the particle attractors in a certain neighborhood (neighborhood attractor or global best *gbest*).

More recently, PSO has been successfully extended to multi-objective optimisation problems and such methods are called MOPSO. Its simple implementation, population based approach, success in handling continuous search spaces and notions of individual position and velocity are major reasons for its popularity. PSO works with a population of individuals each of which is subjected to movement in direction of ‘Pbest’—position corresponding to best fitness attained by an individual and a ‘Gbest’—position of best fitness individual in the entire population. In each generation or cycle ( $t$ ), every individual is associated with a position vector ( $\bar{\phi}_t$ ) and a velocity vector ( $\bar{v}_t$ ). The size of these vectors is equal to the number of variables in the problem. The position and velocity of each individual is updated according to following equations:

$$\bar{v}_{t+1} = w\bar{v}_t + c_1r_1 \cdot (\text{PBest}_t - \bar{\phi}_t) + c_2r_2 \cdot (\text{GBest}_t - \bar{\phi}_t), \quad (7.8)$$

$$\bar{\phi}_{t+1} = \bar{v}_t + \bar{\phi}_t, \quad (7.9)$$

$$\bar{\phi}_{t+1} = \bar{\phi}_{t+1} + \bar{\delta}. \quad (7.10)$$

Above are position and velocity update equations. The term  $w$  is known as inertia weight and  $c_1$  and  $c_2$  are known as learning factors. In our procedure  $w$  has been chosen as 0.5,  $c_1$  and  $c_2$  are both taken to be 1.0. Once the velocities and positions have been updated, a random perturbation, denoted by  $\bar{\delta}$ , is added to an individual’s position based on some probability. This is known as ‘turbulence factor’ and is analogous to ‘mutation’ employed in GAs. Goal of ‘turbulence’ is to preserve diversity in the population.

The MOPSO utilised in this study has been borrowed from [31, 32]. At the start of optimisation, for all  $N$  particles positions ( $\phi$ ) are initialised randomly and velocities ( $v$ ) are set to zero. At the onset ‘pbest’ for each particle is assigned as the particle itself. The current MOPSO maintains an external archive of non-dominated solutions of the population which is updated after every generation. This global archive is empty in the beginning and can store only a maximum number of non-dominated solutions which is specified at the start. In case the number of non-dominated solutions exceed the maximum size of the archive, in any generation, clustering is invoked to restore the archive size. For each particle in the population a personal archive, also called ‘pbest archive’ is maintained. The ‘pbest archive’ contains the most recent non-dominated positions that particle has encountered while searching the space. Such an additional archiving scheme for the particles is often found to be extremely effective.

In every generation, each particle is assigned two guides ‘pbest’ and ‘gbest’ from its ‘pbest archive’ and swarms global archive. The way in which these guides are allocated has a great impact on algorithms performance and there exist several methods for guide selection. In this study, ‘NWtd.’ and ‘Dom.’ methods for personal best selection and global best selection have been chosen. For more details

on guide selection in MOPSO reader is referred to [32, 33]. Maximum number of generations is set as the termination criterion.

*NSGA-II.* Elitist non-dominated sorting genetic algorithm (NSGA-II) is one of the most popularly used GA for multi-objective optimisation. Several salient features like elite preservation and explicit diversity preserving mechanisms ensure its good convergence and diversity. Brief description of NSGA-II procedure is described here, for further details reader is referred to [19, 34].

In NSGA-II, offspring population (size  $N$ ) is created using parent population (size  $N$ ) by usual genetic operators—selection, crossover and mutation. The created child population is combined with parent population to form a combined population of size  $2N$ , and then a non-dominated sorting is carried out to classify the entire population into several non-dominated fronts. The new population (size  $N$ ) is then filled by the members of combined population belonging to different non-dominated levels or starting from first level. As all members of combined population cannot be accommodated in the new population, several non-dominated fronts have to be discarded. As all members of last front entering the new population may not be accommodated, only few members (corresponding to number of available slots) are selected from the last front based on the crowding distance technique. Binary tournament selection, SBX, and polynomial mutation operators are used for NSGA-II.

### 7.4.3 Performance Comparisons

Arising out of the stochastic nature of evolutionary approaches, it is difficult to conclude anything about performance from just one simulation run. To eliminate the random effects and gather results of statistical significance, we perform multiple (11) runs of both the algorithms corresponding to different initial population. Two performance measures commonly used in EA literature have been employed in this study:

*Attainment surfaces.* Multiple runs of an evolutionary algorithm usually result in multiple non-dominated set. To deduce overall performance, an approximation of best non-dominated set, also referred to as first (0%) attainment surface, is computed from available non-dominated sets. As non-dominated set can be visualised easily in two and three dimensions, attainment surfaces provide a good insight the algorithms performance. The computation of attainment surfaces is done by using attainment surface package described in [35].

*Hypervolume indicator.* Hypervolume is a measurement which takes into account the diversity as well as the convergence of the solutions [36]. Hypervolume represents the sum of the areas enclosed within the hypercubes formed by the points on the non-dominated front and a chosen reference point. For minimisation type problems a higher value of hypervolume is desirable, as it is indicative of better spread and convergence of solutions. Hypervolume computation for a set of non-dominated points is done with respect to a reference point ‘ $R$ ’. It should be noted that contribution to hypervolume is only made by points which dominate the

reference point. Any other point which does not dominate the reference point has zero hypervolume contribution. In this study, we have computed average hypervolume curves over several generations for comparisons purposes. Although, hypervolume computation is dependent on the choice of reference point, yet it is regarded as a good measure and is applicable to two or more objectives.

#### 7.4.4 Local Search Method

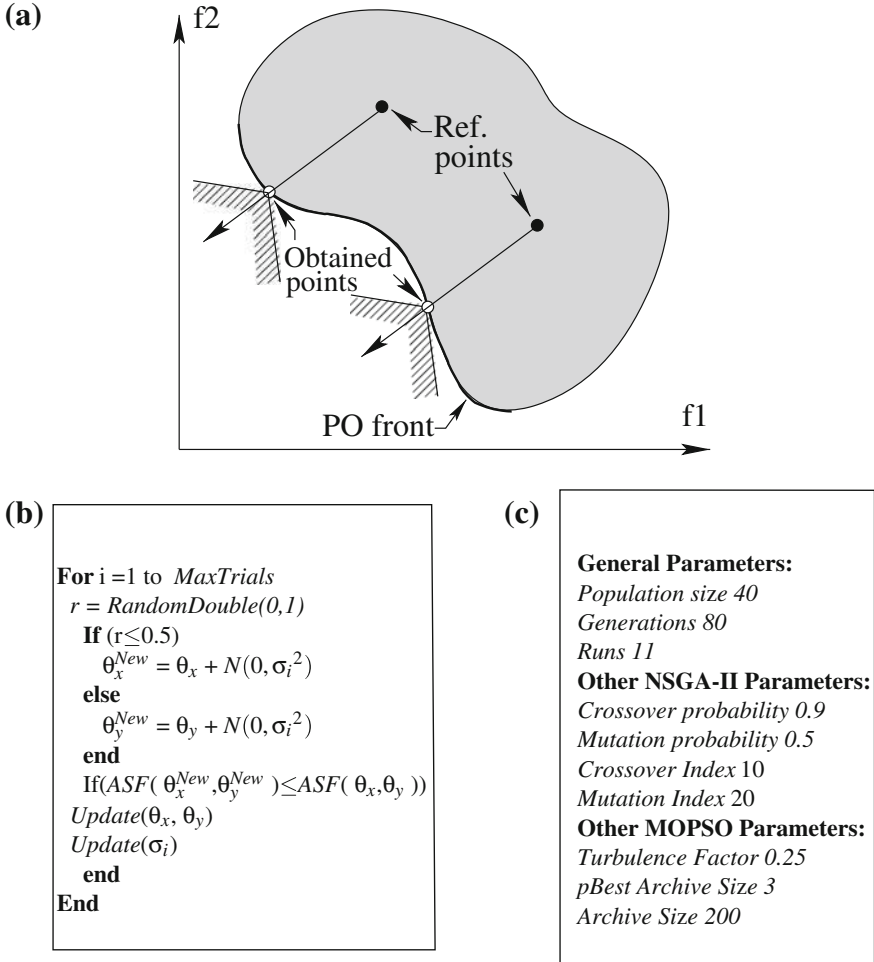
In general, for a real-world multi-objective optimisation problem location of Pareto-optimal front is unknown. Although, MOEAs provide a good means to reach approximate or close to Pareto-optimal solutions, often further improvement on obtained solutions is possible by conducting ‘local search’. Local search usually considers an already found non-dominated solution and tries to improve it by utilising a construction of some single objective function.

In this study we construct an achievement scalarising function (ASF) [37] and carry out its minimisation. A single-objective optimisation problem is formulated as follows. Consider a starting point  $y$  (having objective vector  $f(y)$ ) and setting  $z = f(y)$ , obtained from an multi-objective optimisation simulation and formulate the optimisation problem:

$$\min_{x \in S \subset \mathbb{R}^n} \max_{i=1}^M \frac{f_i(x) - z_i}{f_i^{\max} - f_i^{\min}} + \rho \sum_{j=1}^M \frac{f_j(x) - z_j}{f_j^{\max} - f_j^{\min}},$$

where  $z = f(y)$  is usually referred to as the reference point for local search, and  $f_i^{\max}$  and  $f_i^{\min}$  are maximum and minimum objective values of the ‘best non-dominated’ set. By this minimisation the solutions are projected on Pareto-optimal front and convergence can be guaranteed.

Although various single-objective optimisation techniques could be applied for minimising ASF, but because of discontinuous nature of objective functions (attributable to max operator) gradient-based methods are not applicable. We employed SQP (sequential quadratic programming)-based local search for this purpose and no improvement was found. To overcome this problem we propose a mutation-driven or hill-climbing strategy in this chapter. Figure 7.6b describes the hill-climbing approach. To conduct the local search a maximum number of trials (MaxTrials) are pre-set to limit the number of function evaluations. Then, with equal probability, problem variables  $\theta_x$  and  $\theta_y$  are perturbed according to Gaussian distribution (mean 0.0 and standard deviation  $\sigma_i$ ). Standard deviation ( $\sigma_i$ ) for Gaussian distribution is varied linearly from 10.0 to 1.0 over the iterations. Such a local search enables the investigator to explore wider regions in the starting and becomes more focussed towards the end. If ASF value at newly created orientations is lowered, then the perturbations in  $\theta_x$  and  $\theta_y$  are accepted. The whole procedure is continued till termination criteria is met. In this study MaxTrials is set at 1,500.



**Fig. 7.6** Description for ASF, Hill Climber and Parameter Settings. **a** Achievement scalarisation based *local search*, **b** mutation-driven hill-climbing local search, **c** parameter setting for evolutionary algorithms

## 7.5 Decision Making

When a set of trade-off solutions is obtained from a multi-objective optimisation exercise, a decision point needs to be chosen to proceed further. This is often a non-trivial task for an operator and certain guidelines are necessary. To address this task, we introduce three decision making techniques, namely—‘Reference Point Method’, ‘Marginal Utility Method’ and ‘ $L_2$  Metric Method’ [38]. The first method requires an ‘aspiration point’, as an input from the user. The remaining two

methods do not require any user input. The description of these methods follows next.

*Reference point method.* Here it is assumed that the designer has some pre-decided preference (or aspiration) for an operating point with which he/she is likely to settle. The goal is to find a solution which is better than the aspiration of the designer and thereby this method is known as the ‘Aspiration Point Method’. The aspiration point is allocated as the reference point for ASF scheme described in Sect. 7.4.4, and ASF is evaluated for all points on the Pareto-optimal front. The Pareto-optimal solution which corresponds to the minimum ASF value is selected.

For illustration purposes, following three aspiration points are considered:

$$\begin{aligned} \text{Asp}_1 &= \left( \frac{Ra_{\min} + Ra_{\max}}{2}, \frac{T_{\min} + T_{\max}}{2} \right), \\ \text{Asp}_2 &= \left( \frac{Ra_{\min} + Ra_{\max}}{2}, T_{\max} \right), \\ \text{Asp}_3 &= \left( Ra_{\max}, \frac{T_{\min} + T_{\max}}{2} \right). \end{aligned}$$

The corresponding decision choices obtained on the Pareto-front will be indicated as  $P_1$ ,  $P_2$  and  $P_3$ .  $\text{Asp}_1$ , for example, implies that user is willing to accept an available and better point in the proximity of the mean of best and worst obtained ( $Ra$ ,  $T$ ) values. In case of convex Pareto-optimal decision choice dominates the aspiration point, whereas in case of concave set decision choice gets dominated by the aspiration point.

*Marginal utility method.* This approach does not require any prior information from the user and searches for a Pareto-optimal solution which shows least affinity towards any of its neighbours in the objective space. For computing this affinity, consider three non-dominated points  $P_1$ ,  $P_0$  and  $P_2$ , such that ( $Ra_1 \leq Ra_0 \leq Ra_2$ ) and ( $T_1 \geq T_0 \geq T_2$ ) and we are interested in evaluating the affinity at the middle point  $P_0$ .  $P_1$  and  $P_2$  lie in the neighbourhood of  $P_0$  and are selected as follows. Consider  $k$  points,  $P_{0,m}$   $m = 1$  to  $k$ , nearest to  $P_0$ , with  $Ra_{0,m} \leq Ra_0$ . Then, the centroid of all  $P_{0,m}$ s is computed and out of the  $P_{0,m}$ s the one which is closest to the centroid is selected as  $P_1$ . For selecting  $P_2$  same exercise is repeated but this time considering points such that  $Ra_{0,m}$ s are greater than  $Ra_0$ .

Once  $P_1$  and  $P_2$  are computed for a  $P_0$ , ‘affinity function’ (AF), is calculated as:

$$\text{AF}_{P_0} = \max(W1, W2); \text{ where } W1 = \frac{Ra_{P_0} - Ra_{P_1}}{T_{P_1} - T_{P_0}} \text{ and } W2 = \frac{Ra_{P_2} - Ra_{P_0}}{T_{P_0} - T_{P_2}}.$$

For each point in the non-dominated set, except for  $k$  extreme points at both ends, AF is computed and the solution with minimum AF is assigned as the decision choice. This solution is argued to possess least affinity towards any of its neighbours. The value of  $k$  decides the resolution of the proximity in which we are interested to compute the affinity function. We have taken the value of  $k$  equal to 6. Decision point by this method is usually a ‘knee point’. ‘Knee points’ are often of

great practical importance as they denote a coordinate on Pareto-front where increase (decrease) in one objective is very large when compared with decrease (or increase) in the other objective. From a practical view-point there is not much gain in moving away from the ‘knee’ position.

*L<sub>2</sub>-metric.* This is a straight-forward method to select one solution out of many non-dominated solutions without any user information. Firstly, each objective is normalised in [0.0, 1.0]. Then, an ‘ideal point’ is constructed, which is origin in case of normalised space, and set as the reference point. Euclidean distance ( $L_2$ ) of each point in the non-dominated set is calculated from the reference point and the solution with smallest Euclidean distance is finally selected.

## 7.6 Experiments

In this section a series of simulations are performed on various solid models to demonstrate the working of MORPE. The major goals of this study are:

1. Compare the performances of MOPSO and NSGA-II by computing hypervolume over generations and draw conclusions on their convergence and diversity characteristics.
2. Approximate the Pareto-optimal set by computing first (0%) attainment surface from 11 runs of each MOPSO and NSGA-II.
3. Fine tune the best joint non-dominated of MOPSO and NSGA-II by carrying out ‘Local Search’, and find truly (or close to) Pareto-optimal solutions.
4. Analyse the trade-off solutions and validate the overall procedure. In particular, examine the extreme solutions and identify similarities.
5. Demonstrate the working of ‘Decision Making Methods’ and highlight their significance in selecting the build orientations.
6. Draw out practical guidelines for a designer through careful post-optimal analysis.

First, basic geometrical solid models like Cuboid, Cuboidal Pyramid, Prism and Pyramid are considered. The objective function evaluation is comparatively less intensive (computationally speaking) for these simple models as they are made up of flat and lesser number of faces. Results arrived here provide preliminary conclusions regarding MOPSO and NSGA-II performances and validate the working of MORPE. Next, more complicated solid models (with time consuming function evaluations)—Pentagon Bar, Cylinder, Pie, Diamond and Connector are considered for the bi-objective optimisation. For this set of objects only a single run of MOPSO and NSGA-II is performed and the *best joint non-dominated* is computed i.e. non-dominated set from MOPSO and NSGA-II are merged and non-dominated sorting is carried out to find non-dominated solutions in the combined set.

For all the solid models, non-dominated sets and orientations corresponding to minimum  $Ra$  and minimum  $T$  are plotted. It should be noted that minimum  $Ra$  and minimum  $T$  solutions are chosen from the *best joint non-dominated set* after doing

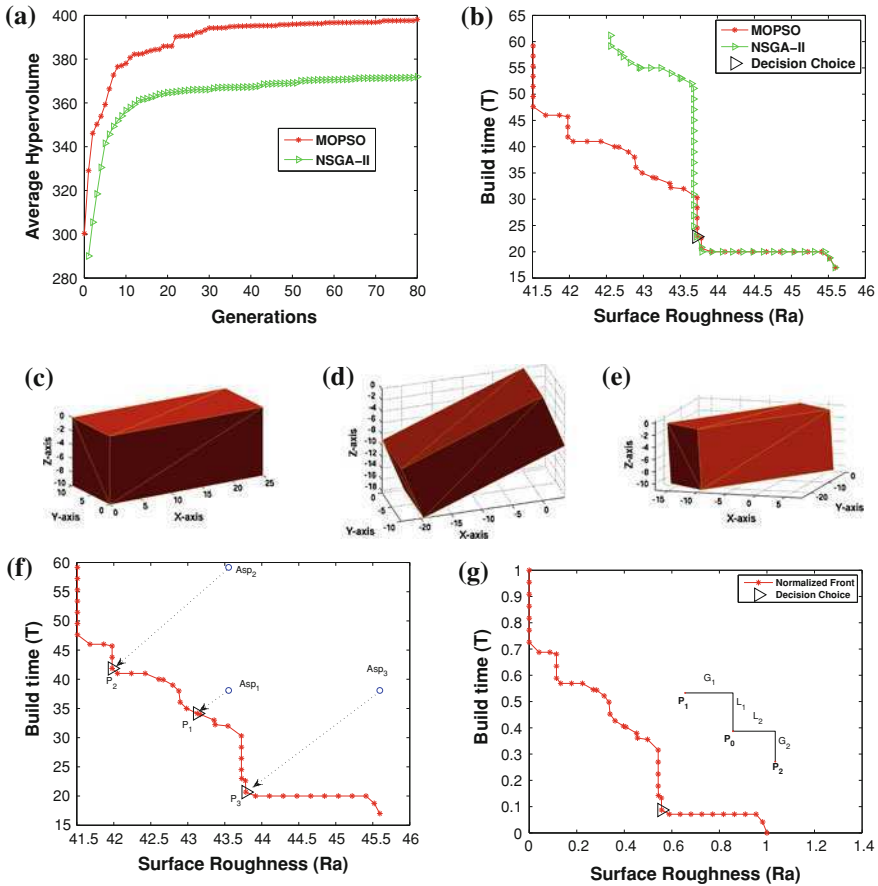
local search. Decision choice based on  $L_2$  metric for each solid object is also highlighted. Similarities amongst extreme solutions of different solid models is found and valuable insight is gained. Several practical aspects and design considerations are also addressed through careful analysis of trade-off fronts.

## 7.7 Results and Discussions

In general, it is difficult to predict an optimal build direction for any solid model. Major difficulty arises caused by complex and non-differentiable expressions for surface roughness. This is also the main reason for using an optimisation algorithm. For a minimum time orientation least number of layers are required. Since layer thicknesses vary according to the adaptive slicing procedure described earlier, an orientation with minimum length in the build direction may not lead to minimum number of slices. The total number of slices depends on slice thicknesses which in turn solely depends on the local geometry of the solid model.

An orientation in which facets are inclined with respect to vertical would require a support structure and is likely to result in higher surface roughness. However, one should note that because of adaptive slicing procedure local surface roughness is limited and appearance of a support structure is counteracted by thinner slices, thus increasing the number of slices. The thinner slices are also associated with smaller strip areas, and as local roughness is weighted with the strip areas the overall surface roughness tends to decrease. For an optimal solution these two opposing factors are balanced. This also explains why a trade-off exists between the build time and surface roughness. Based on this discussion, we see that minimum  $T$  and  $Ra$  orientations are non-intuitive. Now, we study few simple objects—Cuboid, Cuboidal Pyramid, Prism and Pyramid, and investigate their optimal orientations one by one.

For Cuboid, we find that MOPSO performs better by showing a faster rise in hypervolume as compared with NSGA-II and attains a steady value which is higher than NSGA-II hypervolume, Figure 7.7a. Better performance of MOPSO is also highlighted from the first (0 %) attainment surface, Figure 7.7b. MOPSO shows a better convergence and spread by dominating a major portion of NSGA-II attainment surface. The extreme solutions corresponding to minimum  $T$  and  $Ra$  found by MOPSO were more accurate and are shown in Fig. 7.7c and 7.7d, respectively. For the minimum  $T$  orientation shortest dimension occurs along the  $z$ -axis (the build direction). But, minimum  $Ra$  orientation is a tilted one and requires support structure. The appearance of support structure causes an increase in roughness locally which is compensated by lowering of layer thicknesses because of adaptive slicing. Smaller slice thicknesses result in larger number of layers and  $Ra$  gets minimised. The  $L_2$  metric based decision choice is a solution at the ‘knee’ of the Pareto-front and its orientation is shown in Fig. 7.7e. The decision choice has an orientation which is similar to minimum  $T$  orientation. ‘Aspiration point method’ is shown in Fig. 7.7f and solutions found by this method

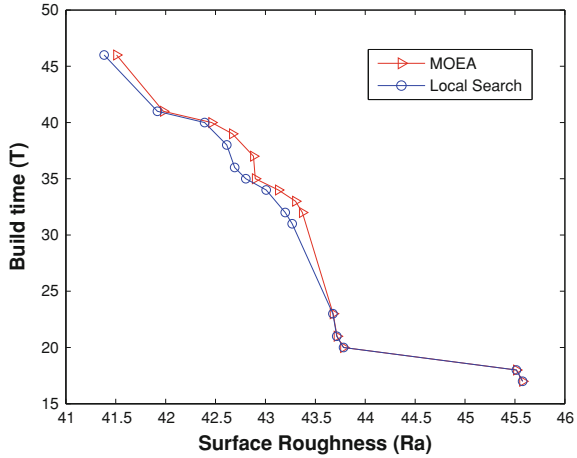


**Fig. 7.7** **a** Average hypervolume curves for Cuboid with reference point (50, 75.0). **b** First attainment surface for Cuboid. **c** Min.  $T$  orientation for Cuboid  $(\theta_x, \theta_y) = (0.0^\circ, 90.0^\circ)$ ,  $(Ra, T) = (45.58, 17.0)$ . **d** Min.  $Ra$  orientation for Cuboid  $(\theta_x, \theta_y) = (180.0^\circ, 68.62^\circ)$ ,  $(Ra, T) = (41.51, 46.0)$ . **e**  $L_2$ -metric decision choice orientation for Cuboid  $(\theta_x, \theta_y) = (133.4^\circ, 86.4^\circ)$ ,  $(Ra, T) = (43.725, 23.0)$ . **f** 'Aspiration point method' based on 'ASF'. **g** 'Marginal utility method' in Normalised Space

lie on the Pareto-front and dominate the corresponding 'aspiration point'. The solution found by the 'Marginal utility method' is shown in Fig. 7.7g and corresponds to a 'knee' point. It is important to understand the significance of 'knee' point and why for practical purposes a 'knee' solution is most favourable point. For e.g., Fig. 7.7b, at the 'knee' point one encounters a sharp increase in  $T$  on a very small reduction in  $Ra$  (if we move along decreasing  $Ra$ ) and a sharp increase in  $Ra$  without much lowering in  $T$  (if we move along increasing  $Ra$ ). Thus, at the 'knee' point moving in either direction is not advantageous as one needs to make a large sacrifice in one objective in order to gain marginal (or practically no)



**Fig. 7.8** ‘Hill Climbing Local Search’ using ASF on Cuboid. After local search many solutions dominate the original non-dominated set

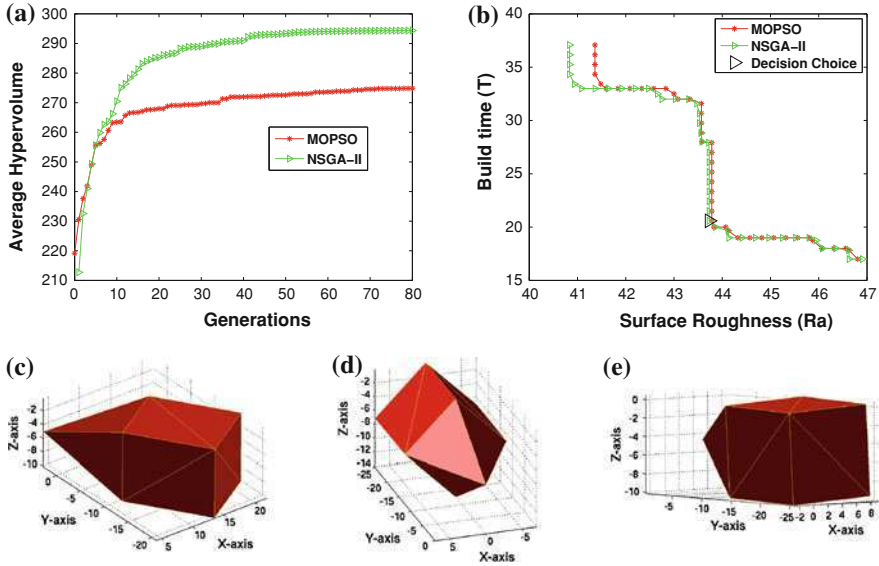


improvement in the other. In presence of multiple ‘knee’ points a higher level decision theory is needed to decide on the best ‘knee’ which we do not present here. Finally, for the purpose of illustration, we show the achievement of local search method in Fig. 7.8. Non-dominated sets from several runs of MOPSO and NSGA-II were combined and global non-dominated was found. On this non-dominated set (marked as MOEA) local search was applied. As shown, many solutions after the local search are modified and improved.

The next object considered is Cuboidal Pyramid, shown in Fig. 7.9. From the hypervolume curves and attainment surfaces shown in Fig. 7.9a, b, NSGA-II shows better performance compared to MOPSO. The minimum  $T$  solution, Fig. 7.9c, lies flat and is similar to minimum  $T$  orientation for Cuboid. The minimum  $Ra$  solution is rotated by  $135.029^\circ$  about  $Y$ -axis and requires a support structure. Due to the appearance of support material, local surface roughness increases and adaptive slicing sets in, decreasing the layer thicknesses and minimising overall  $Ra$ . The  $L_2$  metric decision choice is a ‘knee’ solution and is slightly lifted from the horizontal as shown in Fig. 7.9e.

For Pyramid, the performance of NSGA-II is better compared with MOPSO from hypervolume curves and attainment surfaces as shown in Fig. 7.10a, b. The minimum  $T$  orientation, Fig. 7.10c, is the one in which Pyramid rotates about both the axes and attains a position such that one of its faces is horizontal. Investigation of chosen dimensions revealed that this orientation led to minimum height along  $z$ -axis, and is justified. The minimum  $Ra$  orientation, Fig. 7.10d, is raised from the horizontal level and requires a support presence. The  $L_2$  metric decision choice is a non-rotated configuration and corresponds to a ‘knee’ point on the attainment curve.

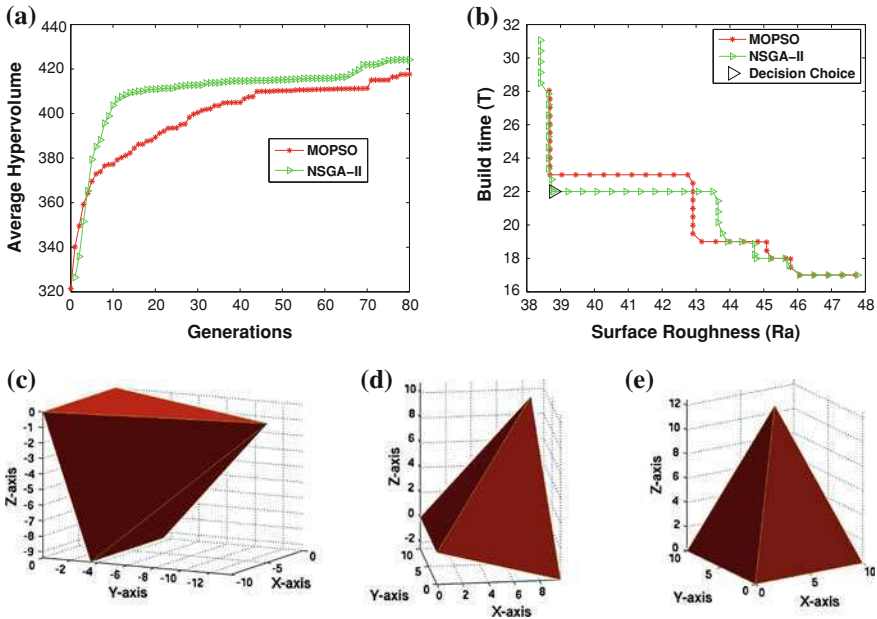
The last simple object considered is Bipyramid, Fig. 7.11. The performance of NSGA-II is again better based on hypervolume curves and attainment surfaces, Fig. 7.11a, b. As in the case of Pyramid, in the minimum  $T$  orientation one of the



**Fig. 7.9** **a** Average hypervolume curves for Cuboidal Pyramid with reference point (50.0, 55.0). **b** First attainment surface for Cuboidal Pyramid. **c** Min.  $T$  orientation for Cuboidal Pyramid  $(\theta_x, \theta_y) = (57.07^\circ, 90.36^\circ)$ ,  $(Ra, T) = (46.78, 17.0)$ . **d** Min.  $Ra$  orientation for Cuboidal Pyramid  $(\theta_x, \theta_y) = (90.0^\circ, 135.029^\circ)$ ,  $(Ra, T) = (40.84, 34.0)$ . **e**  $L_2$ -metric decision choice orientation Cuboidal Pyramid for  $(\theta_x, \theta_y) = (96.1^\circ, 86.43^\circ)$ ,  $(Ra, T) = (43.725, 20.0)$

faces on Bipyramid is horizontal, Fig. 7.11c. The minimum  $Ra$  orientation is shown in Fig. 7.11d and Bipyramid is rotated about both  $X$  and  $Y$  axes. The  $L_2$  metric decision choice is a ‘knee’ solution and has a more raised orientation, Fig. 7.11e, compared to minimum  $T$  orientation.

Based on the hypervolume curves and attainment surfaces for objects discussed till now, NSGA-II outperforms MOPSO in three out of four cases. It is interesting to note that in all the hypervolume curves MOPSO shows a faster hypervolume rise in initial generations, but in most cases the MOPSO hypervolume settles at values lower than that of NSGA-II. Hypervolume trends of MOPSO indicate *premature convergence*—a well known drawback in swarm optimizers. According to authors, pre-mature convergence of MOPSO highlights the absence of potential global guides due to discontinuities in the objective space. Presence of discontinuity slows the march towards Pareto-optimal solutions. It is fair to conclude that NSGA-II is a better performer in general, but it is equally important to note that extreme solutions (corresponding to minimum  $T$  and  $Ra$ ) found by MOPSO were often better. Thus, highlighting the importance of using two optimisers and using best solutions from the combined pool of solutions. Overall similar convergence and spread of trade-off fronts, except for Cuboid, builds our confidence in closeness of the obtained solutions to the true Pareto-set.

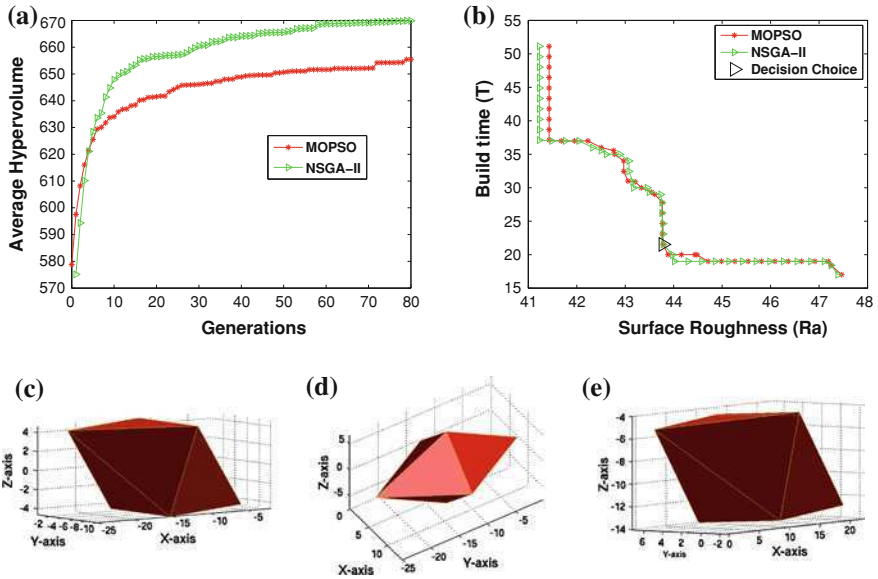


**Fig. 7.10** **a** Average hypervolume curves Pyramid with reference point (55.0,45.0). **b** First attainment surface for Pyramid. **c** Min.  $T$  orientation for Pyramid  $(\theta_x, \theta_y) = (110.0^\circ, 180.0^\circ)$ ,  $(Ra, T) = (47.68, 17.0)$ . **d** Min.  $Ra$  orientation for Pyramid  $(\theta_x, \theta_y) = (0.0^\circ, 14.64^\circ)$ ,  $(Ra, T) = (38.40, 28.0)$ . **e**  $L_2$ -metric decision choice orientation for Pyramid  $(\theta_x, \theta_y) = (0.0^\circ, 0.0^\circ)$ ,  $(Ra, T) = (38.8, 22.0)$

It is worthwhile to note that for these four objects the minimum  $T$  orientation occurs when minimum dimension aligns along  $z$ -axis, though this may not be true in general because of action of adaptive slicing method. These minimum  $T$  orientations can be explained on the basis of ‘planar’ or ‘flat’ surfaces on these solid objects (or absence of curved features).

Next, we consider more complicated solid models—Pentagon-Bar, Cylinder, Pie, Diamond and Connector in Figs. 7.12, 7.13, 7.14, 7.15, and 7.16. Calculation of  $Ra$  for these solid models is computationally intensive and takes large time. Hence, instead of 11 runs we perform only a single run of MOPSO and NSGA-II. Non-dominated sets obtained from NSGA-II and MOPSO runs are combined and global non-dominated sorting is carried out to form the best non-dominated set. This best non-dominated set represents the Pareto-front. It is important to mention that for various solid models, majority of NSGA-II solutions were found to dominate MOPSO solutions, which is consistent with the superiority of NSGA-II over MOPSO as noted earlier.

For Pentagon-Bar, the minimum  $T$  solution almost lies flat on one its larger faces, Fig. 7.12b. Although this orientation is not exactly horizontal, i.e.  $\theta_x$  found is equal to  $90.1^\circ$  and not  $90.0^\circ$ , but for practical purposes difference of  $0.1^\circ$  is of little importance. (Authors computed the estimate of build time for  $\theta_x = 90.0^\circ$

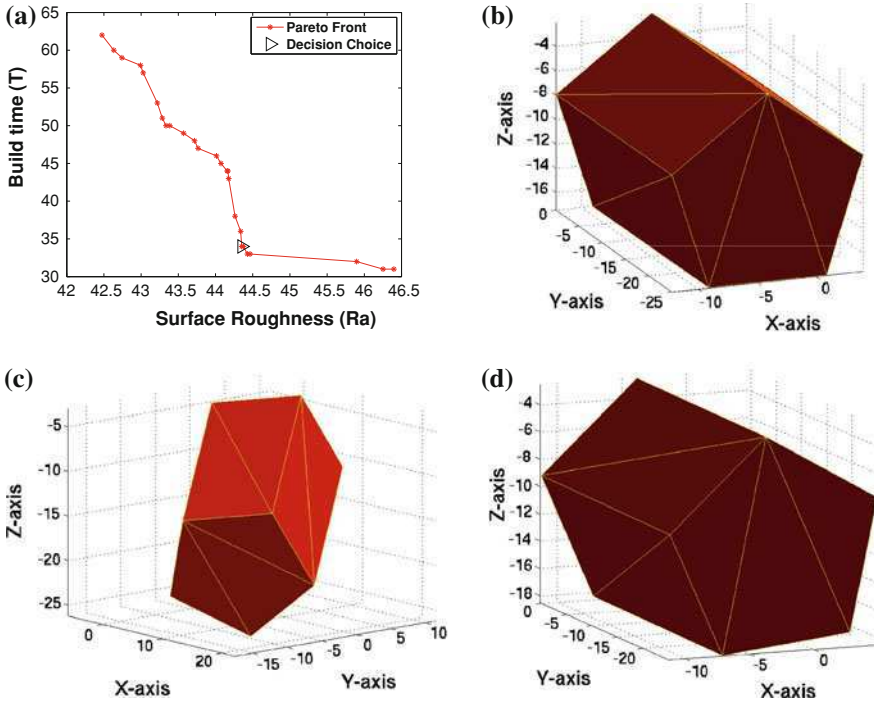


**Fig. 7.11** **a** Average Hypervolume Curves Bi-Pyramid with reference point (55.0, 70.0). **b** First attainment surface for Bi-Pyramid. **c** Min.  $T$  orientation for Bi-Pyramid  $(\theta_x, \theta_y) = (179.98^\circ, 111.79^\circ)$ ,  $(Ra, T) = (47.36, 17.0)$ . **d** Min.  $Ra$  orientation for Bi-Pyramid  $(\theta_x, \theta_y) = (91.25^\circ, 46.39^\circ)$ ,  $(Ra, T) = (41.23, 37.0)$ . **e**  $L_2$ -metric decision choice orientation for Bipyramid  $(\theta_x, \theta_y) = (11.93^\circ, 108.78^\circ)$ ,  $(Ra, T) = (43.8, 21.0)$

while keeping  $\theta_y$  fixed, and found  $T$  to be 33.0 as opposed to minimum  $T$  of 31.0 with  $\theta_x = 90.1^\circ$ . This behaviour can be explained on the basis of adaptive slicing procedure and/or possible numerical errors, resulting in more number of layers in perfectly horizontal configuration). The minimum  $Ra$  orientation is tilted in the space, Fig. 7.12c and requires a support structure. The  $L_2$  metric based decision choice is a ‘knee’ solution and has a configuration which is raised from the horizontal, Fig. 7.12d.

For Cylinder, it is found that minimum  $T$  orientation, Fig. 7.13b, is tilted in space and requires a support structure. Whereas, minimum  $Ra$  orientation is a flat-lying position. The nature of the minimum  $T$  and  $Ra$  orientations for Cylinder is opposite to the ones obtained earlier, where minimum  $T$  orientation was flat and minimum  $Ra$  orientation was tilted. The explanation for these orientations can be based on the curved surface of Cylinder and action of adaptive slicing. In the flat position, adaptive slicing causes smaller layer thickness due to Cylindrical curvature. Smaller thicknesses increase the count of slices which minimises  $Ra$  and maximises  $T$ . The  $L_2$  metric solution lies on the middle of 3-point-knee and has an orientation with rotations about both the axes, Fig. 7.13d.

For Pie, in the minimum  $T$  orientation the object lies flat, Fig. 7.14b. The Minimum  $Ra$  orientation stands-up, Fig. 7.14c, and requires a support structure in

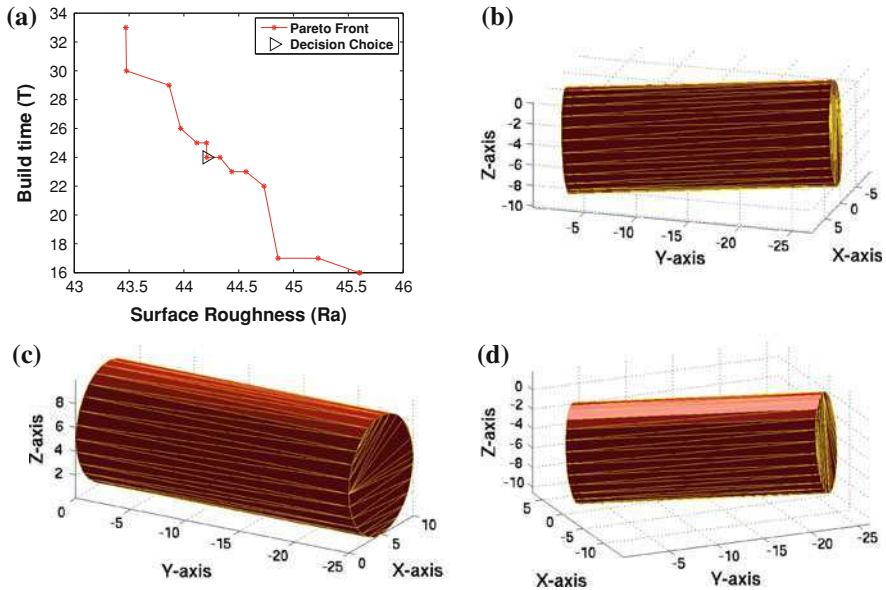


**Fig. 7.12** **a** Trade-off front for Pentagon-Bar. **b** Min.  $T$  orientation Pentagon-Bar  $(\theta_x, \theta_y) = (90.1^\circ, 153.5^\circ)$ ,  $(Ra, T) = (46.25, 31.0)$ . **c** Min.  $Ra$  orientation  $(\theta_x, \theta_y) = (44.7^\circ, 117.89^\circ)$ ,  $(Ra, T) = (42.47, 62.0)$ . **d**  $L_2$ -metric decision choice orientation for Pentagon Bar  $(\theta_x, \theta_y) = (89.34^\circ, 147.46^\circ)$ ,  $(Ra, T) = (44.35, 35.0)$

the lower half. It should be observed that in the minimum  $Ra$  orientation, the slice areas considered along  $z$ -axis while computing the weighted roughness  $Ra$  are smaller than the slice areas occurring in Fig. 7.14b. Small slice areas tend to reduce the  $Ra$  in accordance with Eq. 7.4. Moreover, because of the presence of support adaptive slicing causes smaller slice thicknesses leading to larger number of slices. Both these factors, smaller strip/slice areas and larger number of slices, minimise  $Ra$  in this orientation.

For an object like Diamond, curved surface invokes adaptive slicing in almost any orientation. The minimum  $T$  orientation occurs with Diamond resting on its conical surface, Fig. 7.15b. Although this orientation does have a minimum height along  $z$ -axis but results in minimum number of slices as compared with any other orientation. The minimum  $Ra$  occurs at an orientation slightly tilted from the vertical, Fig. 7.15c, with conical surface pointed upwards and requiring a support material. The  $L_2$  metric decision choice is a middle point on a 3 point knee and has an orientation, Fig. 7.15d, which is not too disparate from minimum  $Ra$  orientation.

For Connector, a flat orientation (similar to the case of Pentagon Bar) leads to minimum  $T$ , Fig. 7.16b. The minimum  $Ra$  is slanted away from vertical  $z$ -axis,

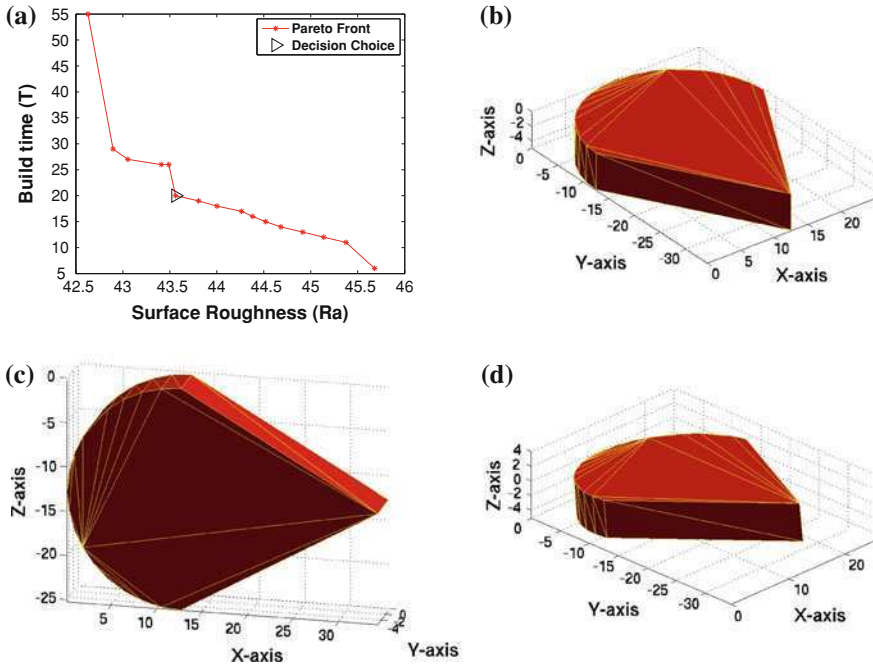


**Fig. 7.13** a Trade-off front for Cylinder. b Min.  $T$  orientation Cylinder  $(\theta_x, \theta_y) = (113.43^\circ, 88.27^\circ)$ ,  $(Ra, T) = (45.6, 16.0)$ . c Min.  $Ra$  orientation  $(\theta_x, \theta_y) = (90.0^\circ, 0.0^\circ)$ ,  $(Ra, T) = (43.31, 19.0)$ . d  $L_2$ -metric decision choice orientation for Cylinder  $(\theta_x, \theta_y) = (124.49^\circ, 100.5^\circ)$ ,  $(Ra, T) = (44.25, 24.0)$

Fig. 7.16c. It should be noted that the contribution to  $Ra$  because of holed features is very small as compared to the exterior surface and minimum  $Ra$  orientation is governed by the roughness of exterior surface.

Finally, two more objects are considered—Prism and Sharp. Both, MOPSO and NSGA-II were applied and a small spread of trade-off solutions was discovered. On conducting local search and manual inspection, the trade-off solutions converged to a single optimum solution which minimised both  $T$  and  $Ra$ . The orientations minimising both  $T$  and  $Ra$  for Prism and Sharp are shown in Fig. 7.17a, b, respectively.

Important observations from the case studies presented here can be summarised as follows. The minimum  $T$  orientation for objects with planar surfaces (Cuboid, Cuboidal Pyramid, Pyramid, Bipyramid, Pentagon Bar, Pie, Connector, Prism and Sharp) was found by aligning the shortest object dimension along  $z$ -axis. In the presence of curved features (Cylinder, Diamond) the minimum  $T$  orientation is unpredictable. The minimum  $Ra$ , in general, cannot be predicted. However, based on the results obtained in this study, minimum  $Ra$  orientations have some rotations about  $x$  and  $y$  axes and show a support presence. Although, the support material tends to increase the roughness locally but adaptive slicing compensates the effect of increased roughness by lowering the slice thickness. Another possible factor for titled orientations, which we have not discussed so far, could be the role of build



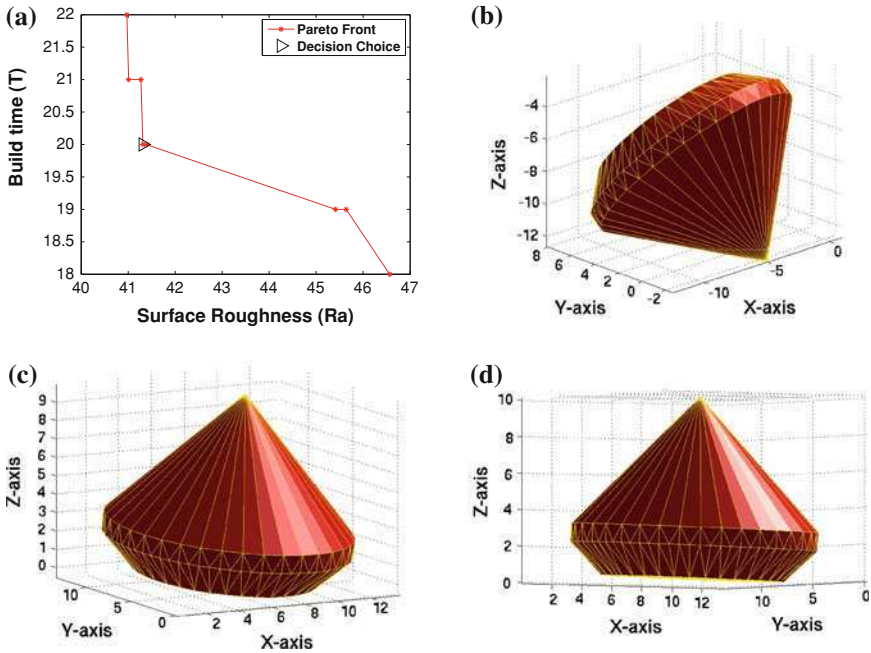
**Fig. 7.14** a Trade-off front for Pie. b Min.  $T$  orientation for Pie  $(\theta_x, \theta_y) = (180.0^\circ, 0.0^\circ)$ ,  $(Ra, T) = (45.8, 6.0)$ . c Min.  $Ra$  orientation for Pie  $(\theta_x, \theta_y) = (90.0^\circ, 90.0^\circ)$ ,  $(Ra, T) = (42.39, 54.0)$ . d  $L_2$ -metric decision choice orientation for Pie  $(\theta_x, \theta_y) = (172.8^\circ, 0.72^\circ)$ ,  $(Ra, T) = (43.51, 20.0)$

angle  $\theta$ . The roughness for any layer depends on  $\theta$  and the slice thickness  $t$ , as discussed in Sect. 7.3. For a large range of  $\theta$  around  $(0-70^\circ)$  roughness for a layer increases with an increase in  $\theta$ . Thus, to minimise the roughness of each layer (hence, minimise the  $Ra$  of entire object) a lower value of  $\theta$  will be preferred. Although, there may not exist any orientation in which  $\theta$  is minimum for all layers, but an orientation in which  $\theta$  is reduced for one or more surfaces may be preferred. An increase in  $\theta$  on other surfaces, caused by rotations aimed at decreasing  $\theta$  for one of more surfaces, is not a major consequence, since increased  $\theta$  for other surfaces will invoke adaptive slicing which will again limit the roughness.

## 7.8 Conclusions

In this chapter, a systematic approach has been presented to derive the optimal build orientations, simultaneously minimising surface roughness  $Ra$  and build time  $T$ , for the FDM process. To address the multi-objective optimisation task two

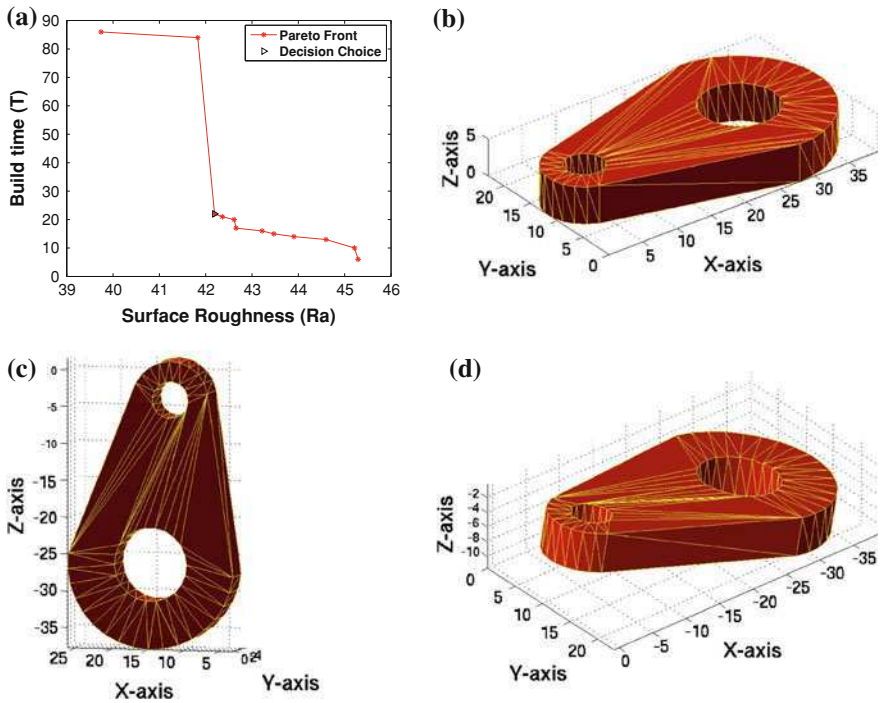




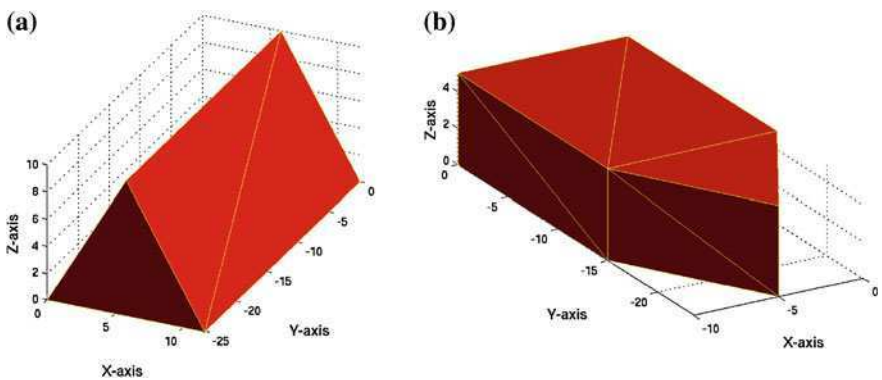
**Fig. 7.15** **a** Trade-off front for Diamond. **b** Min.  $T$  orientation for Diamond  $(\theta_x, \theta_y) = (46.41^\circ, 171.82^\circ)$ ,  $(Ra, T) = (46.56, 18.0)$ . **c** Min.  $Ra$  orientation for Diamond  $(\theta_x, \theta_y) = (5.45^\circ, 4.8^\circ)$ ,  $(Ra, T) = (40.97, 22.0)$ . **d**  $L_2$ -metric decision choice orientation for Diamond  $(\theta_x, \theta_y) = (2.40^\circ, 1.12^\circ)$ ,  $(Ra, T) = (41.3, 20.0)$

popularly used evolutionary approaches—elitist non-dominated sorting genetic algorithm (NSGA-II) and multi-objective particle swarm optimisation (MOPSO)—have been applied. A performance comparison of these two optimisers is carried out by evaluating the ‘hypervolume’ metric, and NSGA-II has been found to perform better. Attainment surfaces are computed to provide an approximation of Pareto-optimal-set. Employment of two optimisers is found useful in identifying the best non-dominated set, particularly the extreme solutions which are better found by either NSGA-II or MOPSO. To further refine the non-dominated solutions obtained from MOEAs a mutation driven hill climbing local search strategy based on ‘ASF’ has been proposed. The local search has been found effective in bringing solutions closer to the true Pareto-optimal solutions. Three decision making methods have been introduced to aid the designer in choosing a preferred solution once the Pareto-optimal set is found. A post-optimal analysis of several objects considered in this study indicates a trend, particularly amongst the extreme solutions found. Such an analysis can be useful in gathering valuable information about the optimal orientations from a practical view-point.





**Fig. 7.16** **a** Trade-off front for Connector. **b** Min.  $T$  orientation for Connector  $(\theta_x, \theta_y) = (.17^\circ, .106^\circ)$ ,  $(Ra, T) = (45.23899, 6.0)$ . **c** Min.  $Ra$  orientation for Connector  $(\theta_x, \theta_y) = (90.0^\circ, 83.0^\circ)$ ,  $(Ra, T) = (39.74, 86.0)$ . **d**  $L_2$ -metric Decision Choice orientation for Connector  $(\theta_x, \theta_y) = (0.44^\circ, 170.28^\circ)$ ,  $(Ra, T) = (42.20, 20.0)$



**Fig. 7.17** **a** Min.  $T$  and  $Ra$  orientation for Prism  $(\theta_{xm}, \theta_y) = (90.0^\circ, 0.0^\circ)$ ,  $(Ra, T) = (37.69, 20.0)$ . **b** Min.  $T$  and  $Ra$  orientation for Sharp  $(\theta_x, \theta_y) = (180.0^\circ, 180.0^\circ)$ ,  $(Ra, T) = (40.36, 7.000)$

## References

1. Alexander, P., Allen, S., & Dutta, D. (1998). Part orientation and build cost determination in layered manufacturing. *Computer Aided Design*, 30, 343–356.
2. Ablani, M., & Bagchi, A. (1995). Quantification of errors in rapid prototyping processes and determination of preferred orientation of parts. *Transactions of the North American Manufacturing Research Institution/SME*, 23, 319–323.
3. Cheng, W., Fuh, J. Y. H., Nee, A. Y. C., Wong, Y. S., Loh, H. T., & Miyazawa, T. (1995). Multi-objective optimization of part-building orientation in stereolithography. *Rapid Prototyping Journal*, 1, 22–33.
4. Hur, J., & Lee, K. (1998). The development of a CAD environment to determine the preferred build-up direction for layered manufacturing. *International Journal of Advanced Manufacturing Technology*, 14, 247–254.
5. Kim, H. C., & Lee, S. H. (2005). Reduction of post-processing for stereolithography systems by fabrication-direction optimization. *Computer Aided Design*, 37(7), 711–725.
6. Pandey, P. M., Thrimurthulu, K., & Reddy, N. V. (2004). Optimal part deposition orientation in FDM by using a multicriteria. *International Journal of Production Research*, 42(19), 4069–4089.
7. Xu, F., Wong, S. Y., Loh, T. H., Fuh, H., & Miyazawa, T. (1997). Optimal orientation with variable slicing in stereolithography. *Rapid Prototyping Journal*, 3(3), 76–88.
8. Hur, S. M., Choi, K. H., Lee, S. H., & Chang, P. K. (2001). Determination of fabricating orientation and packing in SLS process. *Material Process Technology*, 112(2–3), 236–243.
9. Lan, P. T., Chow, S. Y., Chen, L. L., & Gemmill, D. (1997). Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *Computer Aided Design*, 29, 53–62.
10. Pham, D. T., Dimov, D. T., & Gault, R. S. (1999). Part orientation in stereolithography. *International Journal of Advanced Manufacturing Technology*, 15, 674–682.
11. Thrimurthulu, K., Pandey, P. M., & Reddy, N. V. (2004). Optimum part deposition orientation in fused deposition modeling. *International Journal of Machine Tools and Manufacture*, 4, 585–594.
12. Byun, H. S., & Lee, K. H. (2006). Determination of optimal build direction in rapid prototyping with variable slicing. *International Journal of Advanced Manufacturing Technology*, 28, 307–313.
13. Byun, H. S., & Lee, K. H. (2006). Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision. *Robotics and Computer-Integrated Manufacturing*, 22(1), 69–80.
14. Ahn, D., Kim, H., & Lee, S. (2007). Fabrication direction optimization to minimize post-machining in layered manufacturing. *International Journal of Machine Tools and Manufacture*, 47(3–4), 593–606.
15. Masood, S. H., & Rattanawong, W. (2000). A generic part orientation system based on volumetric error in rapid prototyping. *International Journal of Advanced Manufacturing Technology*, 19(3), 209–216.
16. Masood, S. H., Rattanawong, W., & Iovenitti, P. (2000). Part build orientations based on volumetric error in fused deposition modeling. *International Journal of Advanced Manufacturing Technology*, 19, 162–168.
17. Yew, A. B., Kai, C. C., & Zhaohui, D. (2000). Development of an advisory system for trapped material in rapid prototyping parts. *International Journal of Advanced Manufacturing Technology*, 16, 733–738.
18. Thompson, D. C., & Crawford, R. H. (1995). Optimizing part quality with orientation. In *Proceedings of the 6th SFF symposium* (pp. 362–368).
19. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Dordrecht: Wiley.

20. Canellidis, V., Giannatsis, J., & Dedoussis, V. (2009). Genetic-algorithm-based multi-objective optimization of the build orientation in stereolithography. *The International Journal of Advanced Manufacturing Technology*, 4, 714–730.
21. Majhi, J., Janardan, R., Smid, M., & Gupta, P. (1999). On some geometric optimization problems in layered manufacturing. *Computational Geometry*, 12(3–4), 219–239.
22. Leitaó, J. A., Everson, R., Sewell, N., & Jenkins, M. (2008). Multi-objective optimal positioning and packing for layered manufacturing. In *Proceedings of the 3rd international conference on advanced research in virtual and rapid prototyping: Virtual and rapid manufacturing advanced research virtual and rapid prototyping* (pp. 655–660).
23. Padhye, N., & Kalia, S. (2009). Rapid prototyping using evolutionary algorithms: Part 1. In *GECCO '09: Proceedings of the 2009 GECCO conference companion on genetic and evolutionary computation* (pp. 2725–2728).
24. Padhye, N., & Kalia, S. (2009). Rapid prototyping using evolutionary algorithms: Part 2. In *GECCO '09: Proceedings of the 2009 GECCO conference companion on genetic and evolutionary computation* (pp. 2737–2740).
25. Pandey, P. M., Reddy, N. V., & Dhande, S. G. (2007). Part deposition orientation studies in layered manufacturing. *Journal of Materials Processing Technology*, 185, 125–131.
26. Hong, J., Wang, W., & Tang, Y. (2006). Part building orientation optimization method in stereolithography. *Chinese Journal of Mechanical Engineering* (English ed.), 19(1), 14–18.
27. Zhang, L. Q., Xiang, D. H., Chen, M., & Wang, B. X. (2005). Optimum design for RP deposition orientation by genetic algorithm. *Nanjing Hangkong Hangtian Daxue Xuebao/ Journal of Nanjing University of Aeronautics and Astronautics*, 37(Suppl.), 134–136.
28. Zhao, J. (2005). Determination of optimal build orientation based on satisfactory degree theory for RPT. In *Proceedings—ninth international conference on computer aided design and computer graphics* (pp. 225–230), CAD/CG 2005, art. no. 1604640.
29. Zhao, J., He, L., Liu, W., & Bian, H. (2006). Optimization of part-building orientation for rapid prototyping manufacturing. *Journal of Computer-Aided Design and Computer Graphics*, 18(3), 456–463.
30. Tata, K., Fadel, G., Bagchi, A., & Aziz, N. (1998). Efficient slicing for layered manufacturing. *Rapid Prototyping Journal*, 4(4), 151–167.
31. Padhye, N. (2008). Topology optimization of compliant mechanism using multi-objective particle swarm optimization. In *GECCO '08: Proceedings of the 2008 GECCO conference companion on genetic and evolutionary computation* (pp. 1831–1834).
32. Padhye, N., Juergen, J., & Mostaghim, S. (2009). Empirical comparison of MOPSO methods—guide selection and diversity preservation. In *Proceedings of congress on evolutionary computation (CEC)* (pp. 2516–2523). New York: IEEE.
33. Padhye, N. (2009). Comparison of archiving methods in multi-objective particle swarm optimization (MOPSO): Empirical study. In *GECCO '09: Proceedings of the 2009 GECCO conference companion on genetic and evolutionary computation* (pp. 1755–1756).
34. Deb, K., Agarwal, S., & Meyarvian, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
35. Knowles, J. (2005). A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In: *IEEE Intelligent Systems Design and Applications (ISDA V)* (pp. 552–557).
36. Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Aachen: Shaker.
37. Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In G. Fadel & T. Gal (Eds.), *Multiple criteria decision making theory and applications* (pp. 468–486). Berlin: Springer.
38. Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Boston: Kluwer.

**Part III**  
**Process Planning and Scheduling**

# Chapter 8

## A Setup Planning Approach Considering Tolerance Cost Factors

Binfang Wang and A. Y. C. Nee

**Abstract** In this study, an ant colony optimization (ACO)-based setup planning system focusing on an integrated procedure for automatic setup planning for machining cast parts is presented. It considers the selection of available machine tools, tolerance analysis and cost modelling simultaneously for achieving an optimal setup planning result. A tolerance cost factor is introduced when machining error stack-up occurs. The setup planning process can be divided into three stages: preliminary setup planning, tolerance planning and optimal setup planning. During the preliminary setup planning stage, design information is extracted from CAD models and each machining feature is assigned certain machine resource based on its tool access directions (TAD) and the tool orientation space of the available machine resource. During the tolerance planning stage, machining features are grouped into setups based on machine tools assigned and their TADs, and the machining datum for each setup is determined. The setups are next sequenced. Then the blueprint tolerances of the machining features are checked based on their ideal setup datum, and a tolerance cost factor is generated accordingly. During the optimal setup planning stage, the manufacturing cost of each setup plan is evaluated based on the cost model, in which, multiple objectives (setup change cost, machine tool cost, cutter change cost, etc.) that are possibly in conflict with each other are combined through the use of a weight vector and an aggregation function. The setup plan which incurs the least cost is taken as the

---

B. Wang (✉)

Institute of High Performance Computing, Agency for Science Technology and Research, 1 Fusionopolis Way, #16-16, Connexis, Singapore 138632, Singapore  
e-mail: wangbf@ihpc.a-star.edu.sg

A. Y. C. Nee

Department of Mechanical Engineering, National University of Singapore,  
10 Kent Ridge Crescent, Singapore 119260, Singapore  
e-mail: mpeneeyc@nus.edu.sg

final result. The feasibility of using the ACO algorithm is studied to address the NP-complete setup planning problem. A case study is carried out to illustrate the proposed approach. This approach can optimize product design and its manufacturing processes simultaneously to meet cost, time and performance objectives, achieving product quality and user satisfaction.

## 8.1 Introduction

Setup planning is a function of both process planning and fixture design [1]. Its task is to determine the number and sequence of setups, the features to be machined in each setup, and the part orientation and locating features of each setup. The purpose of a setup plan is to locate and fix a part in a specific manner on a machine tool so that machining can take place according to design specifications.

Two factors have to be considered in setup planning, design specifications and manufacturing resources. Design specifications include workpiece geometry, dimension, tolerance and features which can be both functional and aesthetic. Manufacturing resources include production requirements, available machines, cutting tools and fixtures. A setup plan which considers all these factors optimally can ensure to deliver the product not only with high quality but also with high-throughput rate and low cost.

From published work, these two factors of setup planning are treated separately. Most research attempted to satisfy the first factor, i.e., analysis of the design specifications, including tolerance analysis, precedence constraint satisfaction, geometric data analysis and tool access direction verification. The main objective of these studies is to reduce the locating error and minimize the number of setups. While the second factor was normally considered at the optimization stage in terms of cost, quality and lead time, and under an assumption of the availability of certain machine tools.

Different setup plans can be generated in a different manufacturing environment. Different setup plans may also lead to different locating methods and manufacturing cost, and different fixture configurations can result in different locating stack-up errors and stability. Machining accuracy and the capability of available machine tools would need to be considered simultaneously during setup planning in order to achieve a higher level of optimization. An optimized setup plan can eliminate unnecessary machining error stack-up, improve product quality and reduce production cost.

In this study, an optimized setup planning approach which considers machining error stack-up and the capability of available machine tools simultaneously is addressed. It considers seven conflicting cost objectives, i.e., the setup change cost, cutter change cost, machine tool cost, fixture cost, machining cost, scheduling cost and transport cost. It is assumed that a machining environment contains several machine resources which include 3-, 4- or 5-axis machining centres, and can be

distributed and located in different places. A tolerance cost factor, which will be applied in the case of a stack-up, has been introduced. The strategies are achieved by minimizing a cost model among the distributed machine resources.

## 8.2 Literature Review

In the literature, attempts were made to satisfy the design specifications, i.e., tolerance analysis, precedence constraint satisfaction, geometric data analysis and tool access direction verification. Various approaches have been applied. Fuzzy sets theory was used by Ong et al. [1–5] to present the geometrical relations, tolerance relations, fixturing relations, machining requirements, operation features, etc., in the setup planning systems for manufacturability and fixturability evaluation. Zhang et al. [6] proposed a hybrid approach in which various constraints other than tolerances in setup planning are identified and discussed. Precedence relationships among the features have been analyzed by Ong et al. [7] to generate a precedence relationship matrix. This matrix acts as the main constraint for setup planning optimization.

Most research adopted tolerance analysis as the main criterion in setup generation and sequencing. Boerma and Kals [8] reported on the development of a computer-aided planning system for the selection of setups and the design of fixtures in part manufacturing. The automated selection of setups is based on the comparison of the tolerance relations between the different shape elements of the part. A tolerance factor has been developed to compare the effect of different tolerances. The system selects the positioning faces automatically and supports the selection of tools for positioning, clamping and supporting the part. Zhang et al. [9] and Huang et al. [10] discussed the importance of setup planning in relation to tolerance control in process planning. A graphical approach was proposed to generate optimal setup plans based on design tolerance specifications. Wu and Chang [11] described an approach that uses the tolerance specification in a feature-based design system to generate setup plans with explicit datum elements. The focus of this research is an automated tolerance analysis approach for selecting setups and datum for prismatic workpieces in the design system. Zhang and Lin [12] introduced a systematic approach for automated setup planning in CAPP. The concept of “hybrid graph”, which can be transferred into directed graph by changing any two-way edge into one-way edge, is introduced. Tolerance relations are used as critical constraints for setup planning. Lin et al. [13] developed a variant CAPP system with tolerance charts to automate the generation of operation illustration for aircraft components. Zhang et al. [14] employed an extended graph to describe a feature and tolerance relationship graph (FTG) and a datum and machining feature relationship graph (DMG), which could be transferred to an analytical computer model, and a tolerance decomposition model to partition a tolerance into interoperable machining errors. These could be used for locating error analysis or for feedback to the design stage for design improvement.

Tseng and Huang [15] presented a multi-plant tolerance allocation model to determine the working tolerance of each of the components by considering all the feasible manufacturing operations of the available plants. The primary objective is to maximize the cumulative sum of the working tolerances. Hebbal and Mehta [16] focused on the development of a formalized procedure for automatic generation of feasible setups and then to select an optimal setup plan for machining the features of a given prismatic part. The proposed work considers simultaneously the basic concepts of setup planning from both machining and fixturing viewpoints in order to formulate feasible setup plans.




A few researchers have considered machine resources during setup planning. Zhang et al. [17] proposed object-oriented manufacturing resources modelling (OOMRM) and agent-based process planning (AAPP). OOMRM describes manufacturing resource capability and capacity in an object-oriented manner, which intends to encapsulate manufacturing system knowledge and the methods of using the knowledge. Based on OOMRM, an AAPP prototype is implemented as a man-machine integrated process planning platform. It supports an experienced manufacturing engineer in mapping out a more reasonable and flexible machining process. Ong et al. [7] presented a hybrid generative algorithm and simulated annealing (SA) approach for setup planning and re-setup planning in a dynamic workshop environment. Cai et al. [18] proposed an adaptive setup planning approach for various multi-axis machine tools, focusing on kinematic analysis of tool accessibility and optimal setup plan selection.

Since setup planning can produce alternate setup plans due to different considerations between design specifications and machine resources, the question of optimization arises. Different approaches have been applied to deal with this problem. Zhang et al. [6] used a numerically exhaustive approach to select the best solution from all the possible alternatives that satisfy the required constraints. Zhang et al. [9] and Huang et al. [10] proposed a graph theoretical approach to represent the design specification of a part. The problem of identifying the optimal setup plan is transformed into a graph search problem. Zhang et al. [19] applied SA to setup planning and Zhang [20] used GA for the optimization. Zhang et al. [14] presented seven setup planning principles to minimize machining error stack-up under a true positioning GD&T scheme assisted with the extended graph approach. An optimal tolerance assignment strategy has been developed and implemented by Song et al. [21]. The optimization criteria are to minimize the manufacturing cost and cycle time while maintaining product quality. The cost model considers effective factors at the machine level, part level and feature level. Optimization of tolerance assignment plan with genetic algorithm is formulated. The Monte Carlo simulation-based tolerance stack-up analysis is employed to determine the satisfaction of design tolerance requirements.

From the literature, it is observed that previous research on setup planning mainly focused on analysis of tolerance specifications of a workpiece, and there are few applications considering machine tools simultaneously with tolerance analysis. When dealing with tolerance analysis, the operation sequences are generated based on dimensions and shapes by checking whether the parts produced



**Table 8.1** Reasoning heuristics for machining features

	Small hole	Large hole	Plane
			
Reasoning heuristics	<ol style="list-style-type: none"> <li>1. It has three faces: one cylinder face and two plane faces;</li> <li>2. The cylinder face is machined</li> </ol>	<ol style="list-style-type: none"> <li>1. It has four faces, two cylinder faces and two plane faces;</li> <li>2. The outer cylinder face is machined</li> </ol>	<ol style="list-style-type: none"> <li>1. It has one machined plane face</li> </ol>

are within the designed tolerances. If the parts produced are out of the specified tolerances, it needs to use a more accurate machining centre or operation to meet the requirements. Nowadays products are fabricated in a distributed manufacturing environment, and the transportation cost should be considered as well. In addition, in most reported research, the tolerance charts are input manually and there is no clear extraction of machining and tolerance features from the CAD model.

This research reports on an approach which will extract design information automatically from CAD models through geometric reasoning and will integrate machine tools selection with tolerance analysis, and to achieve optimal setup planning by optimizing the real-time integration process with a cost model. The real-time integration is achieved by performing setup planning with the machining resources in real-time, which takes into account production schedules and some unexpected events, such as machine tool breakdown and an urgent job which needs to be rushed out. Seven cost objectives which address the manufacturing process are considered. A weight vector is applied to the multiple objectives and combined them into an aggregation cost function, i.e., the cost model. The optimization function is then to minimize the cost model. In addition, a tolerance cost factor is introduced in the cost model, which is applied when a more accurate machining centre or operation is required due to the effect of tolerance stack-up in a setup.

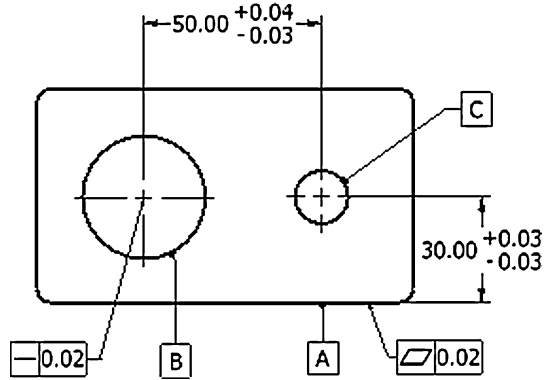
## 8.3 Setup Planning System

### 8.3.1 Consideration of Design Specifications

Setup planning should satisfy the design specifications, i.e., geometric, dimensional and tolerance requirements, precedence constraint satisfaction and tool approach direction (TAD) verification. Product design information in a CAD model would need to be recognized and extracted before setup planning.

In this research, hole and plane features, which commonly exist on a cast part, are considered. The heuristics used for reasoning the hole and plane features are shown in Table 8.1. There are two types of hole features. One is a smaller hole

**Fig. 8.1** Self and relative-tolerance

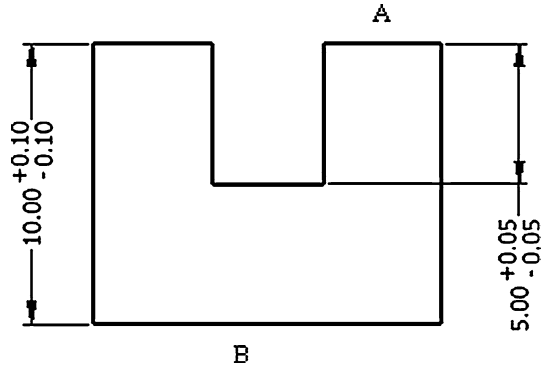


feature which is generated purely by machining, and it is the “Small Hole” shown in Table 8.1. The other is the larger hole generated by casting and requires finish machining. This type of hole is usually quite large, and it is the “Large Hole” shown in Table 8.1. The TAD of a feature is determined by searching whether there are any intersection entities in the candidate direction with a ray which has a radius similar to the cutter. If the result is negative, the candidate direction can be considered as a TAD. Otherwise, this candidate direction should be discarded. For a hole feature, the candidate directions are the two directions of the hole axis. For a plane feature, the candidate direction is the direction of the face normal.

Tolerances, which represent the characteristics and relationships of features on a part, serve as functional description of the design requirements which should be satisfied during manufacturing processes. Tolerances can usually be classified into self-tolerances and relative-tolerances. Self-tolerance is the tolerance reflecting the size deviation of a feature. It is related to the operations, but not directly related to other features. The examples are the straightness for feature B and flatness for feature A in Fig. 8.1. Both are typical casting features. While relative-tolerance reflects the position tolerance in relation to the other features, such as feature C, which is a machining feature having dimension tolerances with A and B, respectively, which are shown in Fig. 8.1. Relative-tolerance can be used to identify the locating datum of a feature. For example, in Fig. 8.1, to guarantee the dimensions of C, it is logical to take A and B as the locating datum. Otherwise, tolerance stack-ups would arise and tolerance compression might happen.

Tolerance compression means a feature has to be machined with higher tolerances compared with the blueprint values, and therefore a more accurate machining centre or operation may be needed. Tolerance compression can happen between setups and within a setup. The tolerance compression between setups usually happens due to tolerance stack-up. Figure 8.2 shows an example of how the compression of operational tolerances happens in this case. In Fig. 8.2, dimension  $10 \pm 0.10$  is to be obtained from the previous operation. To obtain dimension  $5 \pm 0.05$ , if it is machined taking B as the base, tolerance for dimension 10 has to be compressed to less than 0.05 by considering the tolerance stack-up. For a process plan with multiple setups, this could happen quite frequently.

**Fig. 8.2** Tolerance compression



For a CNC machine, no chain analysis is needed for the relative and positional relationships between the geometry surfaces in a setup because they can be programmed accurately. If the specified tolerances cannot be obtained, nothing can be done in the sequence unless a machining method or a machine tool with a higher process capacity is adopted or a higher rate of scrap can be accepted. For example, assuming the two dimensions in Fig. 8.2 have to be achieved in a setup. A more accurate machining method may be used for dimension  $10 \pm 0.10$  comparing with obtaining it in a separate setup. In a multi-axis machine tool environment where multiple operations can be carried out in a single setup and the design datum cannot always coincide with the setup datum, tolerance compression would occur quite frequently.

The compression of operational tolerances will lead to an increase of the manufacturing lead time and production costs and should be taken into consideration during setup planning.

### **8.3.2 Consideration of Real-time Machine Resources**

For setup planning, the application of setup datum may vary according to different machining environment, e.g., 3-, 4- or 5-axis machining centre, whether vertical or horizontal. The number of setups and the selection of the machining features in each setup depend on the machine tool configuration, that is, the number of axes and the orientation of the axes.

In setup planning, features are grouped into setups according to the type of machining centres. For a 3-axis machining centre, the machining features are grouped based on their TADs. Features with the same TADs are assigned to the same group. In this case, the number of setups is determined by the number of TADs of the machining features. For a 4/5-axis machining centre, the machining features are grouped based on the tool orientation space (TOS) of the machining centre. Features with TADs within the machining centre's TOS are assigned to the same group. In this case, the number of setups is determined by both the TOSs of the machining centres and the TADs of machining features. To determine the

locating features for a setup, the position tolerances for the machining features in a setup are verified.

In this research, it is assumed that a machining environment contains several machining centres which could be distributed and located in different places. Each machine has different capabilities (rigidity, power, accuracy, etc.), schedule, tooling, operation cost, with unique machine type, configuration, table size, main axis direction, machine ID code and location. Among them, the schedules of machining centres are very important when making setup planning. From a technical viewpoint, a setup plan may appear to be good, but by taking into account the schedules of candidate machining centres, it may not be the most economical.

Machine resources are provided in a manufacturing database. A user interface is developed, and it provides a way for the user to configure and update the machining environment in real-time, i.e., the currently available machining resources along with their capabilities, attributes and their operating schedules. It is integrated with the database and therefore each time the user updates the manufacturing environment using this interface, the database will be updated accordingly. The process planning module will read information from this database when performing setup planning. Therefore, the setup planning is performed with the machining resources with real-time response, which takes into account the production schedule and some unexpected events, such as the machine tool breakdown and an urgent job which needs to be handled immediately.

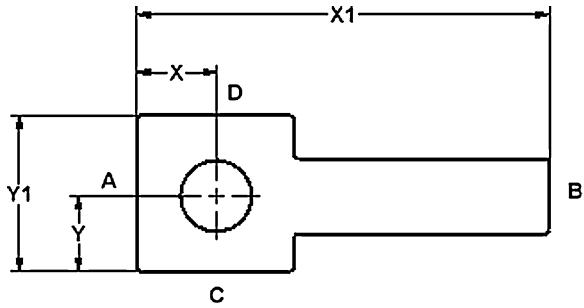
### ***8.3.3 Tolerance Analysis***

Depending on the accuracy of the machine tool, features machined in a single setup can be maintained in accurate relationship with respect to the machine tool coordinate system. This position will be lost if the part is dismounted from the machine tool and remounted again in a different fixture. The errors in the alignment of the part and fixture on the machine tool can be equal to or even larger than the accuracy requirements of small-tolerance relations. As a result, the position accuracy of a feature machined in a previous setup can be insufficient to realize the required accuracy in the relation to the features to be machined in the present setup. Even in a single setup, when the setup datum is different with the design datum, the required position tolerances of a feature may not be guaranteed. It is necessary to check the blueprint tolerances during setup planning to ensure that the setup to be used is a feasible one.

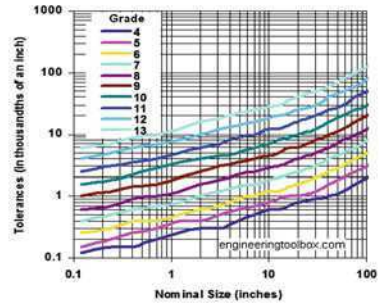
*Case 1:* Dimension datum coincides with setup datum If the setup datum coincides with a feature's dimension datum, then it is not necessary to check the tolerance for this feature. It is based on the assumption that the selected machining process and fixturing method can guarantee the dimensions and tolerances.

*Case 2:* Dimension datum does not coincide with setup datum In this case, it is necessary to take into consideration the stack-up error. For example, in the workpiece illustrated in Fig. 8.3, the position dimensions clearly state that the

Fig. 8.3 Tolerance chain



Tolerance Grades												
4	5	6	7	8	9	10	11	12	13			
Lapping and Honing												
	Cylinder Grinding											
	Surface Grinding											
	Diamond Turning											
	Diamond Boring											
	Broaching											
		Turning										
		Boring										
			Milling									
			Planning and Shaping									
			Drilling									



(a)

(b)

Fig. 8.4 Dimensional tolerance capabilities of operations [22]

centre of the hole (a machining feature) should be at the distance  $X$  from face  $A$  and  $Y$  and from face  $C$ . Consequently, it must use face  $A$  and face  $C$  as datum to locate the workpiece while drilling this hole. This would ensure that the hole is at the specified distance from face  $A$  and face  $C$ . If one uses face  $B$  as a stopper, the derivation in length  $X1$  between faces  $A$  and  $B$  would cause inaccuracies in the position of the hole. If length  $X1$  is oversized by 1 mm, the centre of the hole will be at  $(X + 1)$  millimetre away from face  $A$ . If the length  $X1$  is undersized, the hole would shift towards face  $A$  and would be nearer than distance  $X$  from face  $A$ . However, if location is on face  $A$ , the hole would always be at the same distance from face  $A$  irrespective of the variation in length  $X1$ . Similarly, the same situation will occur when locating with face  $D$  instead of face  $C$  for dimension  $Y$ .

To satisfy the dimension requirements, sometimes a more accurate process or even a more accurate operation has to be chosen, and it would be more expensive. To reflect the additional cost if a higher accuracy machining process/operation is required, a tolerance cost factor ( $f$ ) is introduced, which will be applied when calculating the machining time. Each operation can achieve a typical tolerance, and it is always within a certain range (Fig. 8.4). Machining processes operating under normal conditions would produce parts within the tolerances as indicated in Fig. 8.4a. Figure 8.4b indicates the ANSI B4.1 Standard Tolerances. According to

blueprint tolerances specified on the workpiece, suitable machining processes will be selected to generate the machining features. To calculate  $f$ , it is first assumed that the operation selection for a machining feature is according to the lowest tolerance which this operation can achieve. For example, if there is a hole with tolerance around 0.25 mm, a drilling operation with the lowest tolerance 0.254 as shown in Fig. 8.4 is chosen for this. The tolerance range is defined as grades, and a grade represents a cell in Fig. 8.4a. For example, for the drilling process, the tolerance range can be divided into four grades: grade 10 to grade 13.  $f$  is calculated based on the grade. The initial value  $f$  is set to 1. If the tolerance jumps to a new grade,  $f$  is increased by the number of grades jumped. If the jump is in between the grades, half a jump is used. If a selected machining operation cannot achieve this higher tolerance, a more accurate process will be selected.

### 8.3.4 Cost Model

One of the ultimate goals of an enterprise is to be profitable. Hence, every company has the mandate to reduce cost and increase profit margin, which can be achieved more effectively at the design planning stage rather than the manufacturing stage. In this research, setup planning is performed based on a cost model and an optimization methodology has been formulated to minimize the overall cost of machining all the features on a workpiece. It justifies the machining overhead with machining time, and considers the tolerance requirements simultaneously.

Minimizing the manufacturing cost is a multi-objective problem which can be achieved by minimizing several cost objectives, such as the setup change cost, cutter change cost, machine and fixture cost, etc. Those objectives, however, are possibly in conflict with each other. Depending on how the features are to be located on the faces of a workpiece, they can be grouped into different TADs. Usually, the smallest number of TAD groups would be the best as the cost of setups will be lower. However that is not always true because the grouping is dependent on the type of tooling used. The schedules and the locations of different machine resources will result in different machining times thus the manufacturing costs. A planned machine tool with a schedule requiring additional waiting time to start work will cost more considering the wasted waiting time, and a machine tool located elsewhere may cost more than one which is nearby, considering the transport cost. In addition, different machine tools have different fixturing methods, leading to different costs. For example, for a 5-axis machining centre, fewer setups are needed, so the total machining time would be less. However, there may be a trade-off between reduced machining time and a higher overhead on a 5-axis machine. In addition, since the fixturing method is likely to be more complex, it will cost more. Conducting tolerance analysis of a setup incurs additional steps and this will move up the manufacturing cost. Therefore, the cost model should consider all these objectives and is a composite of: (1) machine tool overhead; (2) cutter cost due to wear and tear; (3) fixture cost; (4) schedule-based cost per unit

time; (5) setup time cost; (6) tool change time cost and (7) transport cost. Seven cost factors with respect to the seven objectives are described in detail in the following.

*Machine tool cost per unit time (MCP)* The machine tool cost per unit time is the summation of the operating cost per unit time and the fixed investment cost amortized over time. It is constant for each machine.

*Cutter cost per unit time (CCP)* Similar to MC, the cost of a cutter per unit time is the summation of operating cost per unit time and fixed investment cost amortized over time. It can be considered a constant for each cutter when machining a specific part of a particular material.

*Fixture cost per unit time (FCP)* Fixture cost is also treated as a constant that occurs when a fixture is used in a setup. It is the summation of the operating cost per unit time and fixed investment cost amortized over time. Modular fixtures are considered and used in this research for all types of machine tools.

*Schedule-based cost per unit time (SCP)* The cost is also treated as a constant that occurs when a machine is needed to wait for some time to perform the operations planned. It is based on the schedule of the machine.

*Setup change cost per time (SCCP)* Setup change is required when a machine tool change is needed which will also require a new fixture. The setup change cost is treated as a constant per time.

*Cutter change cost per time (CCCP)* Cutter change is required when two adjacent operations are performed on the same machine tool using different cutters. In addition, machine tool change may also result in cutter change. The cutter change cost incurred between any two operations is also treated as a constant per time.

*Transport cost per unit distance (TCP)* Transport is required when operations on the same workpiece have to be performed on different machine tools which are located in different places. The transport cost incurred between any two machine tools is treated as a constant per unit distance.

Although the seven cost factors are treated as constants, for different machining environments, their values would need to be changed. Therefore, when performing for each setup planning, one has to set these values accordingly.

A setup plan usually contains several setups. Each setup contains resources which includes a machine tool, a fixture and different cutters to complete the machining processes in this setup. The cost for the tooling is calculated using the machining times of machining the features in this setup. The setup change cost and the cutter change cost is based on the change times. The transport cost depends on the distance of the current machine to the next machine.

As stated previously, minimizing the manufacturing cost is a multi-objective problem and there are two general approaches for solving this. One approach is the Pareto-optimal solution. It is suitable for problems where the objectives are not conflictive. If the objectives are possibly in conflict with each other, the second approach known as the classical weighted-sum approach where the objective function is formulated as a weighted sum of the multiple objectives can be adapted. A weight vector is applied to the objectives in the cost model to evaluate the overall cost. Therefore, the cost model can be formulated as Eq. 8.1. The setup

planning is to solve this NP-complete problem. It is mentioned in Eq. 8.1 that the objectives of machine cost and fixture cost are combined and the same weight is used. This is due to the fact that in this study, it is assumed that a machine tool uses a particular fixture, therefore the machine and fixture cost factors can be considered as a constant together. Using this cost model as the objective function for obtaining an optimized solution, a feasible setup plan with the minimum cost can be found.

$$\begin{aligned}
 & w_m \sum_i^I (\text{MCP}_i + \text{FCP}_i) \times T_i^{\text{Machining}} + w_c \sum_k^K \text{CCP}_k T_k^{\text{Machining}} \\
 & + w_s \sum_i^I \text{SCP} \times T_i^{\text{Waiting}} + w_{sc} \times \text{SCCP} \times N_s + w_{cc} \times \text{CCCP} \times N_c + w_t \sum_i^I D_{mn} \\
 & \times \text{TCP}, T_i^{\text{Machining}} = \sum_{ij}^{IJ} f_{ij} T_{ij}^{\text{Machining}}, T_k^{\text{Machining}} = \sum_{kj}^{KJ} f_{kj} T_{kj}^{\text{Machining}} \quad (8.1)
 \end{aligned}$$

where,

$T_i^{\text{Machining}}$  machining time for  $i$ th machine tool, a summation of operations' machining time on this machine;

$T_k^{\text{Machining}}$  machining time for  $k$ th cutter, a summation of operations' machining time on this cutter;

$T_{ij}^{\text{Machining}}$  machining time for  $j$ th operation performed on  $i$ th machine;

$T_i^{\text{Waiting}}$  waiting time for  $i$ th machine tool, determined by the schedule of this machine tool;

$T_{kj}^{\text{Machining}}$  machining time for  $j$ th operation performed on  $k$ th cutter;

$f_{ij}$  tolerance cost factor for  $j$ th operation performed on  $i$ th machine;

$f_{kj}$  tolerance cost factor for  $j$ th operation performed on  $k$ th cutter;

$N_s$  number of setups in the current setup plan;

$N_c$  number of tool changes in the current setup plan;

$D_{mn}$  distance between  $m$ th and  $n$ th machine,  $m, n \subset i, m \neq n$ ;

$\text{MCP}_i$  machine cost per unit time for  $i$ th machine;

$\text{CCP}_i$  tool cost per unit time for  $i$ th machine;

$\text{FCP}_i$  fixture cost per unit time for  $i$ th machine;

$I$  number of machines selected in the current setup plan;

$J$  number of operations selected in the current setup plan;

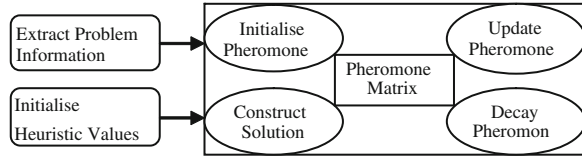
$K$  number of cutters selected in the current setup plan;

$W$  weight element for each of the cost objectives.

For a specific machine, the machining time, which considers the tolerance cost factor,  $f$  is computed by calculating and summing the individual operation times for all the operations performed on a particular machine in a setup plan. The individual operation time is estimated by computing the volume of material removed in that operation divided by the material removal rate. The tool approach time and other travelling time from feature to feature where no materials are



**Fig. 8.5** ACO meta-heuristic framework



removed are not considered. The volume of material removed in an operation can be obtained from the machining feature geometry, while the material removal rate can be computed from tool geometry and processing parameters.

In this study, it is assumed that a setup corresponds to a particular machine tool and fixture. It is also assumed that a machining feature can be generated in an operation with a specific cutter. In this way, the number of setups can be obtained after the completion of the setup plan, and the total number of cutter changes is the summation of the cutter changes in each setup.

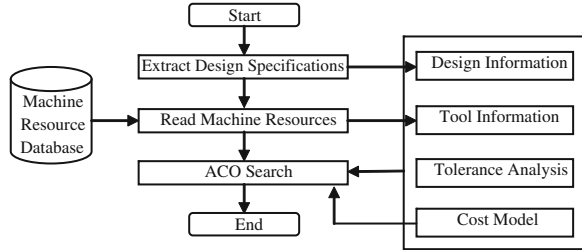
The distances between the locations of the machine tools and their schedules can be obtained from the machine resource database. Cost factors can be used either individually or collectively as a compound cost factor based on the actual requirement and the data availability of the machine resources in a machining environment.

## 8.4 System Implementation

As setup planning is an NP-complete problem, different optimization techniques are commonly employed to achieve an optimal or near-optimal setup plan. The ant colony optimization (ACO) algorithm is a probabilistic technique for solving computational problems. ACO mimics ant activities of finding food in the real world. It is a meta-heuristic-like generic algorithm and has been used for solving many different discrete optimization problems. Dorigo and Gambardella [23] proposed ACO and applied it to the travelling-salesman problem. They also compared the solutions of ACO and showed it to be better than other heuristic approaches like GA, evolutionary programming (EP), simulated annealing (SA) and a combination of GA and SA. Dorigo et al. [24], Jayaraman et al. [25] and McMullan [26] have proved that the ACO is a useful technique and it has been successful in solving NP-complete problems in engineering applications. The ACO meta-heuristic framework describes the scheduling of several processes and is presented in Fig. 8.5.

*Construct solution:* this process is responsible for the construction of new solutions. This is achieved using probabilistic stepwise solution construction. The probability of a particular solution component being added to a growing solution is based on a combination of problem specific (heuristic) information and learned (pheromone) information of how well this component is used in the past solutions. The exact combination of this information and the greediness of the selection mechanism are important implementation specific details.

**Fig. 8.6** Overall system flowchart



*Pheromone trail update and decay:* once solutions have been evaluated, they can influence the pheromone matrix through a pheromone update process. To allow the replacement of old information with new information, a pheromone decay process is also employed that removes the influence of past solutions over multiple successive algorithm cycles.

*Daemon actions:* any action which does not fit into the regular cyclic processes of solution generation, evaluation, update and decay are called Daemon actions. An example of such an action is the storage of an elite solution.

In this study, the feasibility of using the ACO algorithm is studied to address the multi-objective NP-complete setup planning problem.

### 8.4.1 System Structure

Setup planning starts with extracting workpiece information from the raw and final CAD parts. A file, recording the extracted information inclusive of machining features, tolerances, datum, etc., is generated for subsequent searching use. The extracted information can be displayed in an interface through which users can check and modify, and can also add other necessary information, such as form tolerances, to certain features. Figure 8.6 shows the overall flowchart of this developed system. An interface which links with the machine resource database is provided. Through this, users can check, modify and update the machining environment. In this way, the machine resources used in the search are able to reflect the current resource status and make the setup planning more reliable. During the ACO optimizing process, tolerance analysis is conducted, and the cost is evaluated for each solution based on the cost model. Finally, an optimal or near-optimal result can be obtained.

The extracted design information is saved in a structure as follows. The ideal datum is the datum obtained through tolerance definition.

{*Feature Name, TAD, Self\_tolerance, Relative\_tolerance, Operation, Ideal Datum, Length (L), depth, Machining\_time, tolerance\_cost\_factor*}

Features can be machining features and cast features. For cast features, only self-contained tolerance is saved, other attributes are set to null. The machining time is estimated as:

$$T^{\text{Machining}} = L/fc + (\text{Depth}/\text{cut of depth}) \quad (8.2)$$

**Table 8.2** Distance matrix between machine tools

	M1	M2	M3	...	Mn
M1	0	$d_{12}$	$d_{13}$	...	$d_{1n}$
M2	$d_{21}$	0	$d_{23}$	...	$d_{2n}$
M3	$d_{31}$	$d_{32}$	0	...	$d_{3n}$
...	...	...	...	0	...
Mn	$d_{n1}$	$d_{n2}$	$d_{n3}$		0

where,  $L$  is the length of a machining feature,  $f$  the feed and  $n$  the rotation speed. The technical parameters other than  $L$  are taken based on the average capability of the given machine resources. The machining time of each machining feature is estimated before setup planning starts.

The machine resources are saved in a structure as:

{*Machine Name, f, n, cut of depth, schedule, TOS, MCP, FCF*}  
 {*Cutter ID, CCP, Type, Radius*}

Schedule indicates the available time of the machine tool. If it is zero, it means the machine tool is available currently. If it is larger than zero, it means the machine tool is only available at a later time. If tasks are assigned to this machine tool now, a waiting time is required. Each operation is performed with a cutter, and each has a unique CCP. A planner can select suitable cutters for operations to be performed. In this study, the machine tools are assumed to be located in different places to reflect the distributed manufacturing environment in the real world. Therefore, if workpieces are machined on machine tools in different locations, they have to be transported from one place to another and transport cost will occur. The transport cost depends on the distances ( $d_{ij}$ ) between the locations of machine tools. An example of the distance matrix for  $n$  machine tools is shown in Table 8.2, where,  $d_{ij} = d_{ji}$ ;  $d_{ij} = d_{ji} = 0$ , if  $i = j$ .

SCCP, CCCP, SCP and TCP are saved outside the data structure of the machine resources as they are applied at the setup level.

### 8.4.2 ACO-Based Setup Planning

The setup planning process can be divided into three stages: preliminary setup planning, tolerance planning and optimal setup planning. During the preliminary setup planning stage, each machining feature is assigned certain machine resource based on their TADs and the TOSs from the available machine resources. During the tolerance planning stage, the machining features are grouped into setups based on the machine tool assigned and their TADs, and the machining datum for each setup is determined. The determination is performed according to the two rules: (1) if there are more than two machining features sharing the same ideal datum, this datum is taken as the setup datum; (2) if Rule 1 cannot be applied, choose the ideal datum of a machining feature with tighter blueprint tolerances. After that, the setups are sequenced. Then the blueprint tolerances of the machining features are checked based on their ideal datum and the setup datum, and a tolerance cost

factor is generated according to the rules described in Sect. 8.3.3. During the optimal setup planning stage, the manufacturing cost of each setup plan is evaluated based on the cost model described in Sect. 8.3.4. The setup plan which has the least cost is taken as the final result. This setup planning process is adapted in the ACO algorithm which is described as follows.

#### 8.4.2.1 Pheromone Structure

During setup planning, a machining feature on a given workpiece is assigned to a setup based on its TAD, operation type and the TOSs of machine tools, linked to a specific operation for the processing of this feature on a chosen machine tool (M), using a suitable cutter (T) and fixture (F), and in a particular setup orientation (TAD). It can be represented by the set of M, T, F and TAD. Given a particular job shop with available machine tools, cutters and fixtures, a set of alternative operation methods can be generated for a feature by traversing all the possible combinations of M, T, F and TAD that can be used to perform the operation. Thus, the method to process a machining feature can be represented as a set of feasible combinations of M/T/F/TAD.

A setup plan can be specified as a linking of the operation methods for machining all the features on a given part. Therefore, the pheromone dimension can be determined by the number of the machining features. In this study, it is assumed that when a machine tool is selected for a machining feature, its fixture is decided since a machine tool would correspond to a particular fixture in a setup. The cutter would need to be selected among the available cutters. Thus, the pheromone has two levels. One is the machine tool level which contains all the information exclusive of the cutters, and the other is the cutter level.

#### 8.4.2.2 Initialize Pheromone

The design information and machine resources are loaded. The pheromone dimension is assigned accordingly and some heuristic variables are initialized. Matrix structures  $M$  and  $T$  (Tables 8.3, 8.4) are used to represent the pheromone at the machine tool level and cutter level, respectively. They are initialized with a zero value.

#### 8.4.2.3 Construct Solution

Solution construction is the preliminary setup planning. Each machining feature is taken as a region that an ant has to visit. At each region, the ant has to select a machine tool with a cutter from the loaded machine resource. The fixture information can be obtained from the attribute of the machine tool. Figure 8.7 presents a graph that an ant travels. It contains eight regions, i.e., eight machining features in the design space. Tool selection is based on the TAD of the machining feature and the TOS of available machine tools. The TAD of the machining feature must be inside the TOS of the selected machine tool. The cutter is selected according to

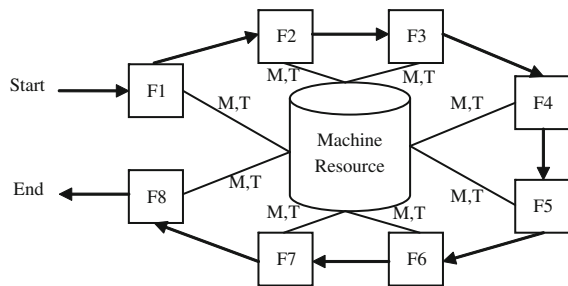
**Table 8.3** Pheromone matrix at machine tool level

	M1	M2	M3	...	Mn
M1	0	m <sub>12</sub>	m <sub>13</sub>	...	m <sub>1n</sub>
M2	m <sub>21</sub>	0	m <sub>23</sub>	...	m <sub>2n</sub>
M3	m <sub>31</sub>	p <sub>32</sub>	0	...	m <sub>3n</sub>
...	...	...	...	0	...
Mn	m <sub>n1</sub>	m <sub>n2</sub>	m <sub>n3</sub>	...	0

**Table 8.4** Pheromone matrix at cutter level

	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	...	T <sub>n</sub>
T <sub>1</sub>	0	t <sub>12</sub>	t <sub>13</sub>	...	t <sub>1n</sub>
T <sub>2</sub>	t <sub>21</sub>	0	t <sub>23</sub>	...	t <sub>2n</sub>
T <sub>3</sub>	t <sub>31</sub>	t <sub>32</sub>	0	...	t <sub>3n</sub>
...	...	...	...	0	...
T <sub>n</sub>	t <sub>n1</sub>	t <sub>n2</sub>	t <sub>n3</sub>	...	0

**Fig. 8.7** An example of the travelling graph



the dimension and operation of the machining feature and the radius and type of the cutter. The radius of a cutter should be smaller than the dimension of the machining feature. The type of the cutter should also match with the operation of the machining feature.

**8.4.2.4 Refinement of Solution**

The constructed solutions are analyzed at this stage. Setups together with setup datum are determined, and the setups are sequenced. A tolerance analysis is conducted, and a tolerance cost factor is generated for each machining feature. A setup plan is generated for each solution.

Upon satisfying the above rules, the machine tools and cutters are selected based on probability. The probability with which ant *k* on nodes *i* chooses the next node *j* at the current iteration *h* is according to the State Transition Rule [27] Eq. 8.3. It is directed by both the pheromone amount and the heuristic value.

$$p_{ij}^k(h) = (\tau_{ij}(h))^\alpha (\eta_{ij})^\beta / \sum (\tau_{ij}(h))^\alpha (\eta_{ij})^\beta, j \in N_i^K \tag{8.3}$$

where,  $h$  iteration index;  $\tau_{ij}$  pheromone value between nodes  $i$  and  $j$ ;  $\eta_{ij}$  heuristic value between nodes  $i$  and  $j$ ;  $p_{ij}$  probability to travel from node  $i$  to node  $j$ ;  $N_1^K$  nodes not yet traversed in the ant-tour so far.

Where,  $i, j \subset (1, n)$  and  $n$  are the number of nodes. Parameters  $\alpha$  and  $\beta$  are used to tune the relative importance of the pheromone and the heuristic distance in decision-making. The heuristic value at the machine level is determined from Eq. 8.4 and the heuristic value at the cutter level is determined from Eq. 8.5.

$$\eta_{ij}^m = 1/(\text{MCP}_i + \text{FCP}_i + \text{TCP} \times d_{ij}) \quad (8.4)$$

$$\eta_{ij}^c = 1/\text{CCP}_i \quad (8.5)$$

#### 8.4.2.5 Evaluate Solution

The feasible solutions are evaluated based on the objective function Eq. 8.1 described in Sect. 8.3.4. The parameters in Eq. 8.1 are obtained as follows:

- $N_s$ : it is obtained from the setup plan of each constructed solution.
- $N_c$ : the number of cutters selected in a solution is obtained first, and then the tool change number is obtained.
- MCP/CCP/FCP/Distance/Schedule: these parameters are obtained from the attributes of the machined tools.
- SCCP/TCCP/TCP: these parameters are obtained from the machine resource database.
- Tolerance cost factor: it is obtained from the attributes of the machining feature, which have been stored in the machining feature data structure during the solution refinement stage.

#### 8.4.2.6 Updating Pheromone

After each iteration, an updating process is triggered if there are better solutions in the population which is used to store the global best results. The pheromone values are updated at both the machine level and the cutter level. It is based on Eq. 8.6.

$$\tau_{ij}(h+1) = \rho\tau_{ij}(h) + \partial\tau_{ij}^{\text{best}}(h), \partial\tau_{ij}^{\text{best}}(h) = 1 \left/ \sum_s^S C_s(h) \right. \quad (8.6)$$

where,  $S$  is the number of solutions at the current iteration that are better than anyone in the population,  $C_s$  the cost of a solution and  $\rho$  the pheromone evaporation rate.

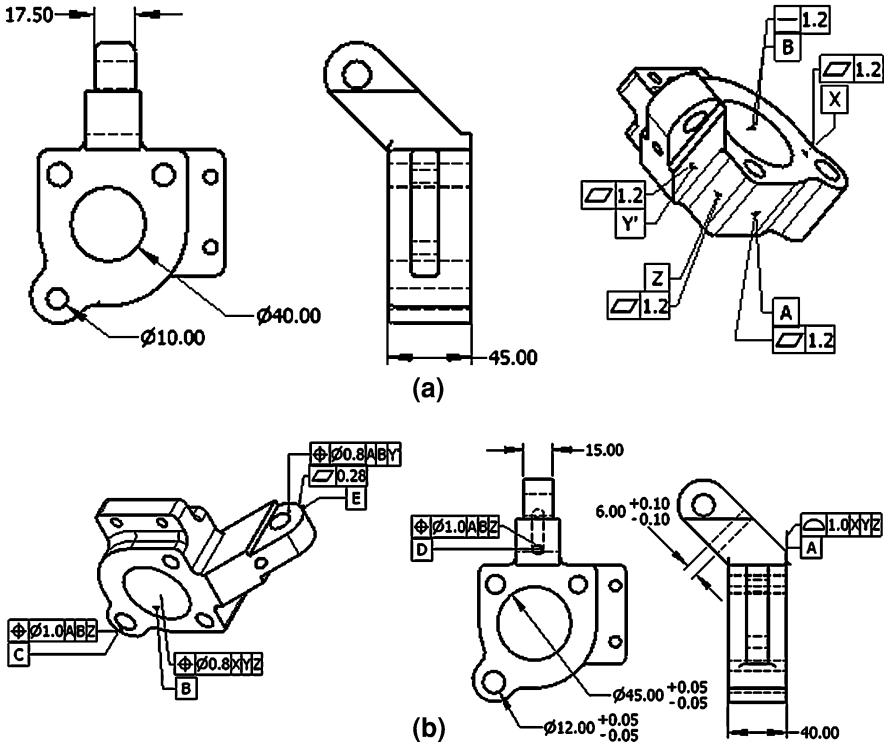


Fig. 8.8 Example part

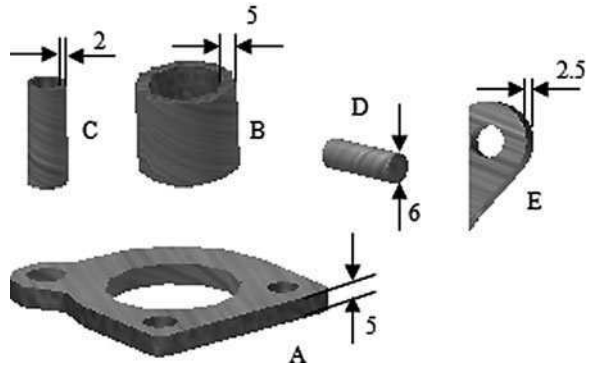
## 8.5 System Performance

### 8.5.1 Illustration of an Example Part

An example part is presented in this section to demonstrate the proposed approach and present the test results. It is a simplified front knuckle of an automotive chassis system, and it is cast followed by machining. Figure 8.8 gives the details of the cast and machined parts. The input CAD model, which contains the dimensions and tolerance information, is constructed using Inventor®.

The planning procedure starts with design information extraction and machine resource configuration, followed by setup planning. For setup planning, the system selects the features on the part to be machined in a setup and determines the sequence of the setup. The setup plan depends on both the tolerance requirement of the geometrical relations between the features and the required orientation of the part with regard to the machine tool orientation.

**Fig. 8.9** Extracted machining features



**Table 8.5** Extracted design information

	Features	TAD	Operation	Self-tolerance	Relative-tolerance	Design datum	L (mm)
Machining features	Plane A	$-z$	Milling	$\pm 0.3$	1.0	X/Y/Z	378
	Hole B	$\pm z$	Reaming	$\pm 0.1$	1.0	X/Y/Z	40
	Hole C	$\pm z$	Boring	$\pm 0.2$	0.8	A/B/Z	40
	Hole D	$45@x$	Drilling	$\pm 0.3$	0.8	A/B/Z	30
	Plane E	$+x$	Milling	$\pm 0.3$	0.8	A/B/Y'	38
Cast features	X, Y, Z, Y', A, B			1.2			

**8.5.1.1 Design Information Extraction**

The design information, i.e., the geometric and dimensional specifications in the machined and cast CAD models, is recognized and extracted. By performing Boolean operations on the two models and using rules stated in Table 8.1, five machining features, i.e., Plane A, Hole B, Hole C, Hole D and Plane E, together with the machining depths, are obtained and shown in Fig. 8.9. Other information listed in Table 8.5, which include TADs, operations, tolerances, design datum and machining length are obtained by geometric reasoning using the API provided by Inventor®.

**8.5.1.2 Machine Resource Configuration**

The specifications of machine resources available for the setup planning are shown in Tables 8.6, 8.7, 8.8 and 8.9, which are configured through an interactive interface.

Other parameters such as TCP are assumed to be \$10 per hour, CCCP is \$10 per change, SCCP is \$20 per change and SCP is \$100 per hour.



**Table 8.6** Machine information

Available machine centre	Feed $f$ (mm)	Rotation speed $n$ ( $s^{-1}$ )	FCP	MCP	Schedule (s)
3-axis machine M1	0.5	2	70	100	480
3-axis machine M2	0.5	2	100	150	120
4-axis machine M3	0.5	2	150	220	60
4-axis machine M4	0.5	2	150	200	180
5-axis machine M5	0.5	2	180	250	600

**Table 8.7** Machine tool orientation space

Available machine centre	TOS					
	$X$		$Y$		$Z$	
	$\Phi_A^-$	$\Phi_A^+$	$\Phi_B^-$	$\Phi_B^+$	$\Phi_C^-$	$\Phi_C^+$
3-axis machine M1	0	0	0	0	0	0
3-axis machine M2	0	0	0	0	0	0
4-axis machine M3	0	0	-90	+90	0	0
4-axis machine M4	-135	+90	0	0	0	0
5-axis machine M5	-135	+45	-90	+90	0	0

**Table 8.8** Cutter information

Cutter no.	CCP	Type	Radius
C1	3	Drill	2
C2	3	Drill	4
C3	3	Drill	10
C4	4	Drill	12
C5	3	Drill	18
C6	3	Drill	24
C7	4	Drill	30
C8	10	Mill	10
C9	10	Mill	15
C10	12	Mill	20
C11	12	Mill	30
C12	15	Mill	50

**Table 8.9** Distance matrix between machine tools

	M1	M2	M3	M4	M5
M1	0	2	4	5	6
M2	2	0	6	8	4
M3	4	6	0	3	4
M4	5	8	3	0	10
M5	6	4	4	10	0

**Table 8.10** Operation times for the machining features

	Plane A	Hole B	Hole C	Hole D	Plane E
Machining time (s)	1890	200	80	72	95

### 8.5.2 Results and Discussions

At the start of setup planning, a suitable weight vector needs to be assigned according to Eq. 8.1. Since there are possible conflicts between the seven objectives and there is no bias for any of them, the same weight of value 1 is chosen and applied to all of them, i.e.,  $\{w_m w_c w_s w_{sc} w_{cc} w_t\} = \{1 1 1 1 1 1\}$ . Arguably through this way, how some objectives are to be traded-off by others can be observed through some cases studies.

The machining time would also need to be calculated. It is performed for each machining feature according to Eq. 8.2. The depth of cut for drilling is 2.5 mm and for milling/boring/reaming is 1 mm. The results are shown in Table 8.10. It is assumed that a feature's operation time is the same no matter which machining centre is used.

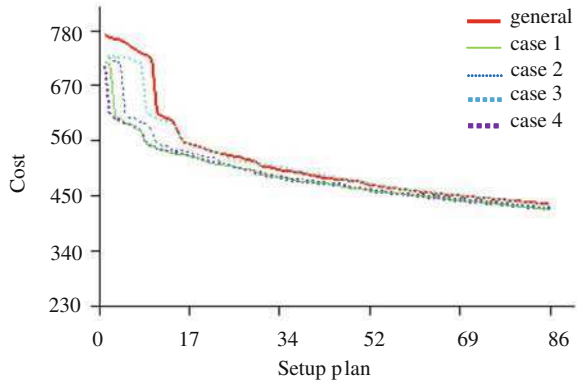
During setup planning, the blueprint tolerances of the machining features in the candidate plans are checked. From Table 8.5, the design datum of feature C is A, B and Z and the relative-tolerance with respect to A, B, Z is 0.8 mm. While the design datum for feature A, B is X, Y, Z with relative-tolerance 1.0 mm. In this case, if machining C uses X, Y, Z as the setup datum, the relative-tolerance between C and A, B would be more than 1.0 mm due to the tolerance stack-up. Therefore the relative-tolerance between A, B and X, Y should not be more than 0.8 mm, which should be ensured when machining A and B with datum X and Y. Since a more accurate method would be needed to achieve the tolerance of 0.8 mm instead of 1.0 mm, a tolerance cost factor is applied to A and B in this setup. The same analysis would be conducted automatically for feature D and E if they are planned to be machined in the same setup with A and B. The setup sequence is arranged according to the sequence of the design datum. The distances used to calculate the transport cost are distances between machines based on the setup sequence.

The results based on the above machining environment are shown in Fig. 8.10 and Table 8.11. The first (top) thick solid curve in Fig. 8.10 shows the cost of the optimization results, and Table 8.11 shows the optimal setup plans. There is one setup change and two cutter changes. The tolerance cost is applied to feature E since the setup datum does not coincide with its dimensional datum. However there is a trade-off by the other cost factors, making it the optimal one.

To demonstrate the effects of some objectives considered in the cost model, four different situations are considered, and the results are shown in Fig. 8.10 and Tables 8.12, 8.13, 8.14 and 8.15 for the comparison.

*Case 1* there is no tolerance consideration in each setup plan, i.e., the tolerance cost factor for all machining feature is one. There is no waiting time based on the machine schedules in the setup plans, and the distances between the machine centres are the same as 4 h away. The forth thin solid curve in Fig. 8.10 shows the optimized cost results.

**Fig. 8.10** Cost of setup plans



**Table 8.11** Optimal setup plan

Optimal setup plan					
Setups	Feature	Machine	Cutter	Datum	Cost
Setup 1	A	M4	C11	X/Y/Z	527
	B		C5		
	E		C9		
Setup 2	C	M3	C1	A/B/Z	
	D		C1		

**Table 8.12** Optimal setup plan for case 1

Optimal setup plan					
Setups	Feature	Machine	Cutter	Datum	Cost
Setup 1	A	M5	C11	X/Y/Z	445
	B		C5		
	C		C1		
	D		C1		
	E		C8		

**Table 8.13** Optimal setup plan for case 2

Optimal setup plan					
Setups	Feature	Machine	Cutter	Datum	Cost
Setup1	A	M1	C12	X/Y/Z	460
	B		C5		
	C		C2		
Setup2	D	M1	C1	A/B/Z	
Setup3	E	M1	C9	A/B/Y'	

Case 2 the tolerance analysis is conducted for each setup plan, and different tolerance cost factors other than one are assigned to each machining feature based on the analysis results. The other two conditions are kept the same as in Case 1. The third thin dot curve in Fig. 8.10 shows the optimized cost results.

**Table 8.14** Optimal setup plan for case 3

Optimal setup plan					
Setups	Feature	Machine	Cutter	Datum	Cost
Setup 1	A	M4	C10	X/Y/Z	519
	B		C4		
	C		C1		
	E		C1		
Setup 2	D	M4	C8	A/B/Z	

**Table 8.15** Optimal setup plan for case 4

Optimal setup plan					
Setups	Feature	Machine	Cutter	Datum	Cost
Setup 1	A	M5	C11	X/Y/Z	445
	B		C5		
	C		C1		
	D		C1		
	E		C8		

*Case 3* the schedules of the machining centres are considered, i.e., some machining centres may not be available, and thus the waiting cost will apply to them accordingly. The other two conditions are kept the same as in Case 1. The second thick dot curve in Fig. 8.10 shows the optimized cost results.

*Case 4* the machining centres are considered to be located in places with different travel times between them, and not the same distances as considered in Case 1. This will result in different optimal results. The other two conditions are kept the same as in Case 1. The fifth thick dot curve in Fig. 8.10 shows the cost results.

Tables 8.12, 8.13, 8.14 and 8.15 show the optimal setup plans for each case separately. They show the differences in costs due to different machining situations. For case 1, the optimal one is using machine M5 to manufacture the five features in one setup. Therefore, there is no setup change and therefore no setup change cost, but there are three cutter changes. Although the tolerance cost factors have been applied to feature C, D and E, there is a trade-off by no setup change and less expensive machine tool cost. For case 2, the optimal solution has two setup changes and two cutter changes, and the tolerance factor has been applied to feature C. It is the lowest machine tool cost without any transport cost which makes it the lowest one. For case 3, there is one setup change, two cutter changes and the tolerance cost has been applied to feature C and E. Its lowest cost is due to fewer setup changes, no transport cost and less waiting time. For case 4, the result is the same as case 1. This is because if all the features are machined in one setup using one machine tool, the transport cost will not be incurred. In this case, it is the same situation as in case 1.

The results show that different machining environment would result in different setup planning results, and this depends much on the dynamic situation of a

company at a particular instant. The seven objectives are conflictive and the trade-offs among them results in the final optimal result.

### **8.5.3 Performance Comparison**

This research considers dynamic machining environment as by Ong et al. [7], and the machine resources include not only the 3-axis machining centre as it was considered by Zhang [20] and Zhang et al. [19], but also 4- and 5-axis machining centres. Compared with the cost model in the studies of Ong et al. [7], Zhang [20] and Zhang et al. [19], the cost model in this research takes more objectives into account, and therefore the cost evaluation is more reliable. There are several improvements in the cost model. Firstly, it considers the machining cost based on the machining time of the machining feature. Dimension differences in the machining features will result in large differences in machining time thus the cost incurred. Secondly, it considers the distributed machining environment which is the current pervasive manufacturing scenario. This is reflected by considering the transport cost between machining centres located in different places. It also considers the cost arising from the current schedules of a machining centre. Another important factor which was not fully addressed in the cost is the tolerance cost factor. It is applied to situations where the setup datum does not coincide with ideal datum of a machining feature, and a more accurate machining method will be needed, and hence the cost will be increased. Other objectives taken into account include machine tool cost, cutter cost, fixture cost, setup change cost and cutter change cost. The consideration of all the cost objectives would make the cost evaluation of a setup plan more accurate and reliable. These multiple objectives are combined into an aggregation function through a weight vector, and the ACO algorithm is applied to solve this multi-objective problem.

This approach has been compared with the methodology presented by Ong et al. [7] since they also considered 4- and 5-axis machining centres. The comparison was done based on the case study shown in [7]. It is worth mentioning that the machining time, the tolerance cost factor, the schedules of machine tools and the transport cost are not considered in the cost model in the comparison.

## **8.6 Conclusions**

In this study, a setup planning system, which focuses on the development of an integrated procedure for automatic setup planning for machining features of a given cast part, is presented. It considers both machine tools selection and tolerance analysis, and is able to achieve an optimal setup planning result by incorporating a cost model. This cost model considers the optimal setup planning as a multi-objective problem. ACO is employed to solve this multi-objective

NP-complete problem. Parts can be produced within designed tolerances and with the lowest cost in a particular production environment. The contributions of this research are:

- The design specifications and the machining environment are considered in an integrated manner, and a cost model is used to optimize the setup planning process.
- Both dimension and position tolerance requirements are taken into account and a tolerance cost factor is introduced to consider the compression of operational tolerances.
- A distributed machining environment, i.e., machine tools located in different places, is considered. A variety of machine tools, e.g., 3- to 5-axis machine tools can be taken into account.
- Real-time machining environment, i.e., the uncertain events that may occur to machine resources, can be taken into account by updating the machine resources in real-time.
- Except for higher tolerance cost consideration, the schedules of the machine tools and the transport cost are taken into account, making the cost model more realistic.
- Optimal setup planning is treated as a multi-objective optimization problem, and the weighted-sum approach is used for its solution.
- ACO is adapted to solve the multi-objective NP-complete setup planning problem.

## References

1. Ong, S. K., & Nee, A. Y. C. (1994). Application of fuzzy set theory to set-up planning. *Annals of the CIRP—Manufacturing Technology*, 43(1), 137–144.
2. Ong, S. K., & Nee, A. Y. C. (1996). An intelligent fuzzy set-up planner for manufacturing and fixturability evaluations. *International Journal of Production Research*, 34(3), 665–686.
3. Ong, S. K., & Nee, A. Y. C. (1997). Automating set-up planning in machining operations. *Journal of Materials Processing Technology*, 63(1–3), 151–156.
4. Ong, S. K., & Nee, A. Y. C. (1998). A systematic approach for analyzing the fixturability of parts for machining. *ASME Transactions Journal of Manufacturing Science and Engineering*, 120(2), 401–408.
5. Ong, S. K., & Chew, L. C. (2000). Evaluating the manufacturability of machined parts and their set-up plans. *The International Journal of Production Research*, 38(11), 2397–2415.
6. Zhang, Y. F., Nee, A. Y. C., & Ong, S. K. (1995). A hybrid approach for set-up planning. *International Journal of Advanced Manufacturing Technology*, 10(3), 183–190.
7. Ong, S. K., Ding, J., & Nee, A. Y. C. (2002). Hybrid GA and SA dynamic set-up planning optimization. *International Journal of Production Research*, 40(18), 4697–4719.
8. Boerma, J. R., & Kals, H. J. J. (1988). FIXES, a system for automatic selection of set-ups and design of fixtures. *Annals of the CIRP—Manufacturing Technology*, 37(1), 443–446.
9. Zhang, H.-C., Huang, S. H., & Mei, J. (1996). Operational dimensioning and tolerancing in process planning: setup planning. *International Journal of Production Research*, 34(7), 1841–1858.

10. Huang, S. H., Zhang, H.-C., & Oldham, W. J. B. (1997). Tolerance analysis for setup planning: a graph theoretical approach. *International Journal of Production Research*, 35(4), 1107–1124.
11. Wu, H.-C., & Chang, T.-C. (1998). Automated setup selection in feature-based process planning. *International Journal of Production Research*, 36(3), 695–712.
12. Zhang, H.-C., & Lin, E. H. (1999). A hybrid-graph approach for automated setup planning in CAPP. *Robotics and Computer-Integrated Manufacturing*, 15(1), 89–100.
13. Lin, A. C., Lin, M.-Y., & Ho, H.-B. (1999). CAPP and its integration with tolerance charts for machining of aircraft components. *Computers in Industry*, 38(3), 263–283.
14. Zhang, Y., Hu, W., Rong, Y., & Yen, W. David. (2001). Graph-based set-up planning and tolerance decomposition for computer-aided fixture design. *The International Journal of Production Research*, 39(14), 3109–3126.
15. Tseng, Y.-J., & Huang, F.-Yi. (2009). A multi-plant tolerance allocation model for products manufactured in a multi-plant collaborative manufacturing environment. *International Journal of Production Research*, 47(3), 733–749.
16. Hebbal, S. S., & Mehta, N. K. (2008). Set-up planning for machining the features of prismatic parts. *International Journal of Production Research*, 46(12), 3241–3257.
17. Zhang, Y.Y., Feng, S. C., Wang, X. K., Tian, W. S., & Wu, R. R. (1999). Object oriented manufacturing resource modelling for adaptive process planning. *The International Journal of Production Research*, 37(18), 4179–4195.
18. Cai, N., Wang, L., & Feng, H.-Y. (2008). Adaptive set-up planning of prismatic parts for machine tools with varying configurations. *International Journal of Production Research*, 46(3), 571–594.
19. Zhang, Y. F., Ma, G. H., & Nee, A. Y. C. (1999). *Modelling process planning problems in an optimization perspective*. *Proceedings of the 1999 IEEE International Conference on Robotics & Automation* (pp. 1764–1769), Detroit, Michigan.
20. Zhang, F. (1997). Genetic algorithm in computer-aided process planning. *MEng thesis, National University of Singapore*.
21. Song, H., Yang, Y. D., Zhou, Y. & Rong, Y. K. (2007). Tolerance Assignment using Genetic Algorithm for Production Planning. *Models for Computer Aided Tolerancing in Design and Manufacturing*, pp. 213–224, doi:10.1007/1-4020-5438-6\_22.
22. [http://www.engineeringtoolbox.com/machine-processes-tolerance-grades-d\\_1367.html/](http://www.engineeringtoolbox.com/machine-processes-tolerance-grades-d_1367.html/).
23. Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66.
24. Dorigo, M., Colomi, A. & Maniezzo, V. (1992). An investigation of some properties of an Ant Algorithm. In Manner, R & Manderick, B (Eds.) *Proceedings of the Parallel Problem Solving from Nature Conference, Brussels*, pp. 509–520.
25. Jayaraman, V. K., Kulkarni, B. D., Karale, S., & Shalokar, P. (2000). Ant colony framework for optimal design and scheduling of batch plants. *Computers & Chemical Engineering*, 24(8), 1901–1912.
26. McMullan, P. R. (2001). An ant colony optimization approach to address a JIT sequencing problem with multiple objective. *Artificial Intelligence in Engineering*, 15(3), 309–317.
27. Dorigo, M., Maniesso, V., & Colomi, A. (1996). The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on System, Man and Cybernetics-Part B*, 26(1), 1–13.

# Chapter 9

## Preference Vector Ant Colony System for Minimizing Make-span and Energy Consumption in a Hybrid Flow Shop

Bing Du, Huaping Chen, George Q. Huang and H. D. Yang

**Abstract** Traditionally, scheduling problems usually deal with the objectives related to production efficiency (e.g., the make-span, the total completion time, the maximum lateness and the number of tardy jobs). However, sustainable manufacturing should minimize the energy consumption during production process. Energy consumption not only constitutes a major portion of total production cost but also results in significant environmental effects. In this chapter, we discuss a multi-objective scheduling problem in a hybrid flow shop. Two objectives considered in the proposed model are to minimize make-span and energy consumption. These two objectives are often in conflict with each other. A Preference Vector Ant Colony System (PVACS) is developed to search for a set of Pareto-optimal solutions using meta-heuristics for multi-objective optimization. PVACS allows the search in the solution space to focus on the specific areas which are of particular interest to decision-makers, instead of searching for the entire Pareto frontier. This is achieved by maintaining a separate pheromone matrix for each objective, respectively and assigning each ant a preference vector that represents

---

B. Du (✉)

School of Management, University of Science and Technology of China, Hefei, China  
e-mail: toto@mail.ustc.edu.cn

H. Chen

School of Computer Science and Technology, University of Science and Technology of China, Hefei, China  
e-mail: hpchen@ustc.edu.cn

G. Q. Huang

Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong, China  
e-mail: gqhuang@hku.hk

H. D. Yang

School of Automation, South China University of Technology, Guangzhou, China



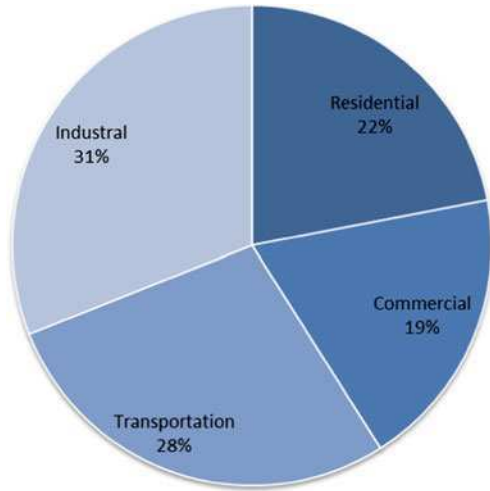
the preference between the two objectives of the decision-makers. The performance of PVACS was compared to two well-known multi-objective genetic algorithms: SPEA2 and NSGA-II. The experimental results show that PVACS outperforms the other two algorithms.

## 9.1 Introduction

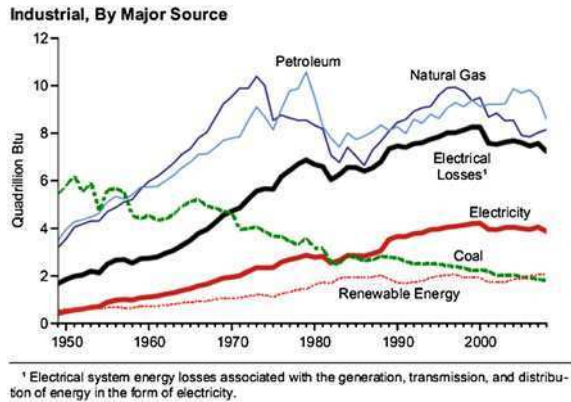
The consideration of environmental issues in manufacturing has taken on increasing importance nowadays. The first decade of the twenty-first century was 10 years of change for the environment, as new environmental issues emerged and existing issues evolved. Pollution, environmental degradation, resource depletion, climate change and global warming, such environmental crises constitute a serious threat to human health, reduce economic productivity and lead to the loss of amenities. It has been increasingly realized that economic development without environmental consideration can cause irreversible damage to the world. Therefore, the concept of “green economy” has been advanced as a solution to many problems afflicting the world at present. The green economy refers to businesses that care about environmental protection, energy efficiency, preservation of biodiversity and sustainable development. The importance of green economy has been recognized by United Nations Environment Programme that has launched the Green Economy Initiative, aiming to assist governments in “greening” their economies by reshaping and refocusing policies, investments and spending towards a range of sectors, such as clean technologies, renewable energies, water services, waste management, green transportation and green manufacturing [1].

The manufacturing sector plays a critical role in the economy. For example, according to the Manufacturing Institute, the manufacturing sector generated \$1.64 trillion worth of goods in the United States and accounted for nearly 57% of total exports in 2008 [2]. By contrast, manufacturing is still far from a dominant sector in the “green” economy. According to a recent report by the Economics and Statistics Administration at the US Department of Commerce, the manufacturing sector accounted for only 13% of green business activity [3]. In addition, the impact of manufacturing on the environment is enormous. Manufacturing industries are predominant in their environmental impact in areas such as toxic chemicals, waste, energy and carbon emissions. When all of these facts are considered, it can be concluded that green manufacturing should be given more attention and play a more important role in the green economy, for the sake of reducing negative environmental impacts and achieving sustainable development. Green manufacturing (also sometimes referred to as sustainable manufacturing or environmentally benign manufacturing) is defined by the US Department of Commerce as “the creation of manufactured products that use processes that are non-polluting, conserve energy and natural resources, and are economically sound and safe for employees, communities, and consumers”. Generally speaking, it involves the

**Fig. 9.1** US energy consumption by sector in 2008. (Source Annual Energy Review 2008 Report by US Energy Information Administration)



**Fig. 9.2** US energy consumption by major source in industrial sector. (Source Annual Energy Review 2008 Report by US Energy Information Administration)



technologies, operational practices, analytical methods and strategies for sustainable production within the industrial ecology framework. It covers a number of important issues such as minimizing waste and toxicity, reducing energy consumption, using renewable energy sources and process optimization.

Energy conservation has always been a major concern in green manufacturing. There are several reasons why energy conservation is critical to the manufacturing industry. First of all, energy consumed by industrial sector is considerable. According to Annual Energy Review released by the US Energy Information Administration, the industrial sector consumes about 31% of all energy in the US in 2008 (see Fig. 9.1) [4]. Secondly, the major sources of the energy consumed by industrial sector are non-renewable, e.g., petroleum, natural gas and coal (see Fig. 9.2) [5]. When energy is produced from non-renewable resources, CO<sub>2</sub> will be emitted into the atmosphere, resulting in greenhouse effect and undesired climate change. For example, for every kilowatt-hour of

electricity used, approximately two pounds of carbon dioxide are released into the atmosphere. If the world continues on its current path of increasing energy consumption, CO<sub>2</sub> emissions are predicted to rise up to 43 billion tons by 2030 [6]. Thirdly, from the point of view of a manufacturing company, the cost of energy consumption usually constitutes a major portion of the total production cost. Energy costs for US manufacturer were 100 billion US dollars annually [7]. Consequently, saving cost could also be an incentive for a company to think about reducing energy consumption.

Production scheduling is an essential activity in manufacturing. In this chapter, the authors would like to discuss a multi-objective scheduling model in a hybrid flow shop considering minimizing both the make-span and energy-consumption criteria. Although multi-objective scheduling problems have been well addressed before, most previous literature focuses on optimizing objectives related to production efficiency, such as the make-span, the total completion time, the maximum lateness or the number of tardy jobs [8]. The issues related to energy conservation have seldom been investigated. However, in a hybrid flow shop, jobs can be processed on one of several machines at each stage. Machines at each stage may have different speed and power. Consequently, different processing routes may lead to different make-spans, as well as different energy consumptions. As the two objectives are usually in conflict with each other, a Preference Vector Ant Colony System (PVACS) is developed to search for a set of Pareto-optimal solutions. PVACS is an Ant Colony Optimization (ACO)-based meta-heuristic for multi-objective optimization problems. Unlike other multi-objective evolutionary algorithms, it allows the search in the solution space to focus on the specific areas which are of particular interest to decision-makers, instead of searching for the entire Pareto frontier. With the set of Pareto-optimal solutions, production managers could strike a balance between production efficiency and energy consumption.

The remainder of the chapter is organized as follows. In the next section, previous studies related to energy conservation, multi-objective evolutionary optimization and hybrid flow shop are reviewed, respectively. [Section 9.3](#) describes the multi-objective scheduling problem in a hybrid flow shop where each stage consists of a set of uniform parallel machines. PVACS is described in detail in [Sect. 9.4](#). Its performance is evaluated through extensive computational experiments in [Sect. 9.5](#). A summary and discussion of future research directions concludes the chapter.

## 9.2 Literature Review

The problem considered in this chapter involves several important research areas. They are: (1) energy conservation in manufacturing industry, (2) multi-objective evolutionary optimization and (3) hybrid flow shop scheduling problems. The rest of the section will discuss the related work in these areas.

### 9.2.1 Energy Conservation in Manufacturing Industry

Over the past 20 years, advances have been made by academic and industrial researchers in energy conservation in manufacturing industry. In an earlier work in 1992, Ross [9] showed that the energy intensities of all production processes should continue to decline through new technologies and appropriate public policies. He also discussed some potential difficulties in energy conservation. Park et al. [10] proposed a decomposition method to divide a change in manufacturing energy consumption into three effects: output growth, energy intensity and structural change. Fromme [11] surveyed the energy conservation in the Russian manufacturing and showed that energy savings of 47% of current demand can be achieved. The most important obstacles for energy conservation in Russia were also discussed. As for energy consumption in the US, Golove and Schipper [12] performed an analysis to examine long-term trends in US manufacturing energy consumption, as well as carbon dioxide emissions. Adenikinju [13] examined the impact of efficiency in energy consumption on the growth in productivity in the Nigerian manufacturing sector using a panel data technique. Bentzen [14] noticed that there is a “rebound effect” in energy consumption, which is estimated for the US manufacturing sector using time series data applying the dynamic OLS method (DOLS).

The studies mentioned above were all conducted from a macro perspective. Meanwhile, there have been a number of studies addressing the technologies and applications for energy saving as well as measuring energy efficiency. Draganescu et al. [15] carried out experiments for statistic modelling of machine tool efficiency and of specific consumed energy in machining as a function of different working parameters. From this model, the amount of the mean economic specific-energy consumed can be determined for a given amount of material. However, this model cannot be easily applied to other machining processes and machine configurations. Dietmair et al. [16] introduced a generic model for the energy consumption behaviour of machines. Successful forecasts of energy consumption and optimizations of machines for minimal energy consumption under a given application scenario were demonstrated with this model. By focusing on the interdependencies and dynamics of all technical processes, Herrmann and Thiede [17] presented an integrated chain concept to foster energy efficiency in manufacturing companies for different layers (e.g., input, logic, user and evaluation layer). A holistic five-step approach for increasing energy efficiency was also developed. Wolters et al. [18] studied sequencing problems in designing energy efficient production systems. They showed that by taking decisions sequentially, the energy conservation potential may be reduced drastically. Mouzon et al. [6] observed there can be a significant amount of energy savings when non-bottleneck equipment are turned off when they will be idle for a certain amount of time, and thus developed operational methods to minimize energy consumption of manufacturing equipment. A Comprehensive review of energy conservation in manufacturing could be found in [19].

### 9.2.2 Multi-objective Evolutionary Optimisation

Multi-objective evolutionary algorithm (MOEA) stands for a class of stochastic optimization methods that simulate the process of natural evolution for multi-objective problems. They are able to achieve better performance than other blind search strategies in multi-objective optimization [20]. In addition, they deal with a set of possible solutions simultaneously which allows us to find several members of the Pareto optimal [21]. MOEA therefore becomes the most popular issue in multi-objective optimization. The earliest work in designing a MOEA appears to be that of Schaffer [22]. He proposed a Vector Evaluated Genetic Algorithm (VEGA) based on the traditional genetic algorithm, but using a modified selection mechanism. After VEGA, researchers have introduced the concept of Pareto optimality into evolutionary algorithms [23]. The basic idea of such algorithms is to identify the set of solutions in the population that are Pareto non-dominated by the rest of the population. These solutions are then given more opportunities to participate in further competition. The earlier works were Non-dominated Sorting Genetic Algorithm (NSGA) proposed by Srinivas and Deb [24], Niche-Pareto Genetic Algorithm (NPGA) by Horn et al. [25] and Multi-Objective Genetic Algorithm (MOGA) by Fonseca and Fleming [26].

The next generation of MOEA started when elitism became a standard mechanism. Zitzler and Thiele [27] were generally recognized as the first researchers who introduced the concept of elitism in a MOEA. After the publication of their work, most researchers in this area started to incorporate external populations in their MOEAs and the use of this mechanism became a common practice. In fact, the use of elitism is a theoretical requirement in order to guarantee convergence of a MOEA [28]. There were several representative MOEAs that incorporate elitism. The first one was Strength Pareto Evolutionary Algorithm (SPEA) [27], as well as an improved version SPEA2 [29], which incorporates a fine-grained fitness assignment strategy that considers for each individual the number of individuals that dominate it and the number of individuals by which it is dominated. The second one was Pareto Archived Evolution Strategy (PAES) proposed by Knowles and Corne [30]. PAES employs a (1 + 1) evolution strategy (i.e., a single parent that generates a single offspring), and uses a reference archive recording the non-dominated solutions previously found. Another notable algorithm was Non-dominated Sorting Genetic Algorithm II (NSGA-II) [31], introduced by Deb et al. as an improved version of NSGA. NSGA-II introduced the concept of “crowding distance”, and during selection, the NSGA-II takes into account both the non-domination rank of an individual in the population and its crowding distance. Besides, the elitist mechanism of NSGA-II is different and does not incorporate external memory. A recent study has been conducted by Chaudhuri and Deb [32], who developed an interactive MOEA procedure integrating both multi-objective optimization process and decision-making process into a unified framework. The procedure not only allows a user to find a set of well-distributed non-dominated

solutions, but also helps the user to impose preference information so as to obtain a particularly preferred solution.

All the MOEAs discussed above are GA-based, however, there have also been some works addressing other meta-heuristics in multi-objective optimization. Doerner et al. [33, 34] developed a Pareto ant colony optimization to solve the multi-objective project portfolio selection problem. Xia and Wu [35] investigated multi-objective flexible job-shop scheduling problem using a hybrid approach by combining particle swarm optimization (PSO) and simulated annealing (SA). A comparative study performed by Sedenka and Raida [36] has reported that a novel multi-objective PSO outperforms NSGA-II in many cases, and shows better ability to find the extreme solutions. Another interesting study was published by Berrichi et al. [37], who have developed a multi-objective ant colony optimization (MOACO) approach to optimize production and maintenance scheduling problem. The results of their experiments indicated the advantage of MOACO over SPEA2 and NSGA-II. For a comprehensive review on MOEA, the authors can refer to Jones [38], Tan [39], Zitzler [40] and Coello [21].

### 9.2.3 Hybrid Flow Shop Scheduling

A hybrid flow shop (HFS), also sometimes referred to as flexible flow shop, is a generalization of the flow shop and the parallel machine environments. Most hybrid flow shop scheduling problems are difficult to solve and have been proved to be NP-hard [41, 42]. Therefore, a large number of heuristics and approximation algorithms have been proposed for different HFS problems. As there have been extremely extensive studies on HFS [43–45], in this section the authors would focus only on the problems with uniform parallel machines addressed in this study.

In a HFS environment with uniform parallel machines, each machine  $i$  is associated with a speed  $v_i$ , the actual time that operation  $O_{jk}$  spends on machine  $i$  is equal to  $p_{jk}/v_i$ , where  $p_{jk}$  is the processing time of job  $j$  at stage  $k$ . Huang and Li [46] investigated the two-stage problem with uniform machines in the second stage. Two heuristics, along with eight effective sequencing rules were developed to assign the jobs to the machines. Besides, Dessouky et al. [47], Soewandi and Elmaghraby [48] and Kyparisis and Koulamas [49] considered the problems with uniform machines at both stage. Bertel and Billaut [50], as well as Dessouky et al. [47] have considered the three-stage problems. The generalized  $k$ -stage HFS problems with uniform machines at each stage have been studied by Sevastianov [51], Kyparisis and Koulamas [52, 53] and Verma and Dessouky [54]. In addition, Voss and Witt [55] considered a real-world application in steel manufacturing where the HFS consists of 16 production stages, and 30,000 production jobs forming several thousand projects. The problem includes sequence-dependent setup costs and the ability to form batches. A heuristic solution procedure based on

different dispatching rules that are capable to realize low tardiness and to form batches was developed.

### 9.3 Problem Description

The problem under study is formally stated as follows. The shop floor consists of  $s$  stages in series. There are  $m_k$  uniform machines in parallel at stage  $k$ ,  $k = 1, \dots, s$ . The  $i$ th machine at stage  $k$  is denoted by  $M_{i,k}$  and the power of machine  $M_{i,k}$  is  $W_{i,k}$ . The speed of machine  $M_{i,k}$  is  $V_{i,k}$ . The impact of speed  $V_{i,k}$  is that machine  $M_{i,k}$  can carry out  $V_{i,k}$  units of processing in one time unit. We assume that the power of machine is constant during processing. As industrial machines and equipment usually cannot be switched off completely during processing, there is power consumption even if the machine is idle. However, such standby power is generally trivial compared to the total power consumption. We therefore set the standby power of machines to zero for simplicity.

There are  $n$  jobs to be processed. Each job  $J_j$  ( $j = 1, \dots, n$ ) consists of a chain of operations  $(O_{j,1}, \dots, O_{j,s})$ . An operation  $O_{j,k}$  is to be processed at stage  $k$  on one of  $m_k$  uniform parallel machines. The operation  $O_{j,k}$  requires  $P_{j,k}$  units of processing ( $P_{j,k}$  is the task time of operation  $O_{j,k}$ ). If operation  $O_{j,k}$  is assigned to machine  $M_{i,k}$ , then it requires  $P_{j,k}/V_{i,k}$  time units to be completed. An operation  $O_{j,k+1}$  may start only after the previous operation  $O_{j,k}$  has been completed.

The following assumptions are considered for the problem as well:

1. All machines and jobs are simultaneously available at time zero.
2. Machine used at each stage cannot process operations corresponding to any other stages.
3. Each machine can process at most one job at a time.
4. Pre-emption of jobs is not allowed, i.e., any commenced operation must be completed without interruptions.

Let  $C_{j,s}$  denote the completion time (of operation  $O_{j,s}$ ) of job  $J_j$  at stage  $s$ . The first objective is to minimize the make-span  $C_{\max} = \max_{1 \leq j \leq n} \{C_{j,s}\}$ , which is the completion time of the last job leaving the system. The other objective is to minimize the total energy consumption of all machines. Let  $x_{j,k}^i$  be a binary variable that is equal to 1 if operation  $O_{j,k}$  is assigned to machine  $M_{i,k}$ , and 0 otherwise. Then the total energy consumption (EC) can be formulated as follows:

$$EC = \sum_{k=1}^s \sum_{i=1}^{m_k} \left( W_{i,k} \sum_{j=1}^n \frac{x_{j,k}^i P_{j,k}}{V_{i,k}} \right) \quad (9.1)$$

Using the well-known three-field notation for scheduling problem [56], the above problem can be denoted by  $HFK(QM_1, \dots, QM_s) || C_{\max}, EC$ .



## 9.4 Preference Vector Ant Colony System

### 9.4.1 Ant Colony Optimisation and its Implementation

Ant colony optimization (ACO), first introduced by Colormi and Dorigo [57–59], is a probabilistic technique for solving computational problems. It has drawn extensive attention since it was proposed and has been successfully applied to many applications in practice, including some scheduling problems [60–64]. ACO algorithms are stochastic search procedures. They are based on a parameterized model called the pheromone model, which is used to sample the search space probabilistically. In the model, artificial ants incrementally construct solutions by adding opportunely defined solution components to a partial solution until a complete solution is built. The construction of solutions is guided by pheromone trails and problem-specific heuristic information. In the context of combinatorial optimization problems, pheromones indicate the intensity of ant trails with respect to solution components, and such trails are determined on the basis of the contribution to the objective function. Before the next iteration starts, some of the solutions are used for performing a pheromone update. The algorithm iteratively searches the solution space following the above procedures until some stopping criteria are satisfied.

The motivation for using an ACO-based algorithm in this chapter is as follows. First, the ACO algorithm includes an important component, i.e., the heuristic information. Heuristic information allows the users to provide problem dependent knowledge to guide the search and is helpful in identifying high quality areas in the search space. Consequently, the ACO-based algorithm can effectively find satisfactory solutions to a combinatorial problem if the heuristic information is well structured [65]. Second, the ACO algorithm is a constructive method that generates solutions step by step by adding a solution component to the current partial solution. It is easier to incorporate users' preference into the process of solution construction because the transition rule, also called the transition probabilities, can easily be redefined with preference consideration.

Traditionally, ACO algorithms are aimed at solving single-objective optimization problems. However, some recent efforts have been directed to develop ACO algorithms for multi-objective optimization problems. For example, Mariano and Morales [66] proposed a Multiple-Objective Ant-Q algorithm (MOAQ) for the design of water distribution irrigation networks. Yagmahan and Yenisey [67] presented a multi-objective ant colony system algorithm (MOACSA), which combines ant colony optimization approach and a local search strategy in order to solve a flow shop scheduling problem. Pareto Ant Colony Optimization, proposed by Doerner et al. [33, 34] was dedicated to solve the multi-objective portfolio selection problem.

The PVACS approach presented in this chapter is basically a Pareto optimization approach. It tries to find a set of solutions that are Pareto non-dominated by other solutions. Additionally, users' preferences are also incorporated into



the algorithm. Such preferences are important because in real-world applications it is normally the case that the users do not need the entire Pareto-optimal set, but only a small portion of it. Consequently, it is desirable that the users can define a preference vector that can narrow the search and magnify certain portions of the Pareto frontier. As the ACO algorithm was originally designed for single-objective optimization problems, and does not contain any user preference, the procedure of the basic ACO algorithm should be modified for the proposed multi-objective problem. For each objective, a separate matrix to store pheromone trail is maintained. For instance, the element  $\tau_{MK}(j, h)$  represents the pheromone information with respect to the make-span objective of job  $h$  to follow job  $j$ . In the construction phase of the algorithm, each ant tries to construct a feasible solution using both heuristic information and combined pheromone information which is a weighted sum of the corresponding elements in the pheromone matrices. The weight is determined by the preference vector. If the solution constructed is not dominated by any other solution previously obtained, then it is stored and will be used to perform a global pheromone update. The details of the PVACS approach will be provided in the following sections.

### 9.4.2 Solution Encoding

The problem presented in Sect. 9.3 consists of two tasks. The first task is to determine the sequence of jobs. The other is to assign jobs to one of the machines. A possible encoding scheme is to consider all the sequences of jobs at each stage in view of the fact that different sequences may occur at different stages, as well as job assignment. However, this representation may lead to extremely large search space. In addition, this representation can cause difficulties in solution construction and pheromone updating. An alternative is to consider the permutation of jobs only at the first stage, and then a list scheduling algorithm is used to decode solutions to a full schedule. This representation may be able to search only a small portion of search space. However, if the list scheduling algorithm is effective, such representation will also provide high quality solution, with much less complexity. Hence, in the encoding scheme of PVACS, a string of  $n$  integers, which is a permutation of  $1, 2, \dots, n$ , corresponding to the job list at the first stage, is used to represent a solution. Each integer in the permutation that represents a job is a solution component in the PVACS algorithm.

To obtain a valid schedule, a solution is decoded by using a list scheduling algorithm. Generally, a list scheduling algorithm is to make an ordered list of jobs by assigning them some priorities. Then select from the list the job with the highest priority for scheduling, and a machine is also selected to accommodate the job. In this study, the list scheduling (LS) algorithm selects jobs according to their sequences in a solution at the first stage. Then a new list is created at each stage, and the sequence of jobs is arranged in the increasing order of their completion time at the previous stage. For machine selection, there are generally two different strategies. The first one is Makespan-First (MF), which assigns the jobs to the

**Table 9.1** Data for the job set

Job	1	2	3	4	5
$P_{j,1}$	12	20	6	15	4
$P_{j,2}$	15	10	8	12	6

machine with the highest speed in a set of available machines (AMS). The other is Energy Conservation-First (ECF), which assigns the jobs to the machine with the lowest energy consumption. The set of available machines is dynamically determined. If machine  $M_{i,k}$  is idle at time  $t$ , then  $M_{i,k} \in AMS(t)$ . If at time  $t$ , there is no machine in the set of available machines, namely  $AMS(t) = \Phi$ , then the job is assigned to the machine with the earliest available time.

Let  $\pi_k$  denote the list for stage  $k$  and  $\pi_k(h)$  denote the job at position  $h$  in list  $\pi_k$ . The pseudo-code of Algorithm LS is represented as follows:

**Algorithm LS**

*Input:* the permutation of jobs  $\pi_1$  obtained from a solution of the algorithm PVACS, the assignment strategy MF or ECF.

*Output:* a valid schedule.

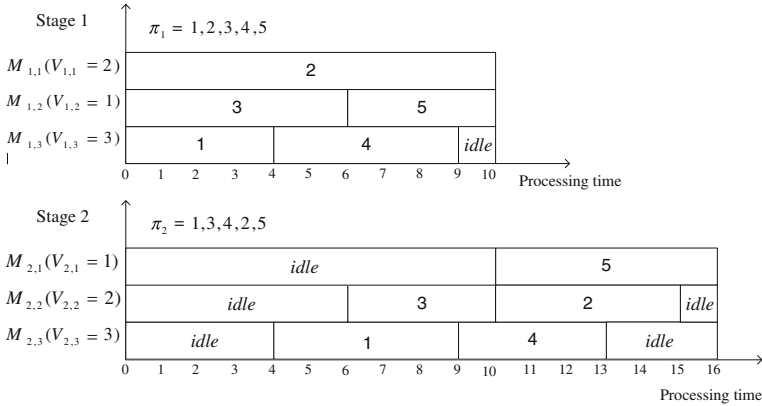
```

for  $h=1,2,\dots,n$  do
    Obtain the available time  $t_1(h)$  of job  $\pi_1(h)$ .
    Create a set of available machines  $AMS(t_1(h))$ .
    if  $AMS(t_1(h)) \neq \Phi$  then assign job  $\pi_1(h)$  to the machine according to
    the assignment strategy MF or ECF.
    else assign job  $\pi_1(h)$  to the machine with the earliest available time.
    end if
end for
for  $k=2,\dots,s$  do
    Create a new list  $\pi_k(h)$  by arranging the jobs in the increasing order
    of their completion time at the previous stage  $k-1$ .
    for  $h=1,2,\dots,n$  do
        Obtain the available time  $t_k(h)$  of job  $\pi_k(h)$ .
        Create a set of available machines  $AMS(t_k(h))$ .
        if  $AMS(t_k(h)) \neq \Phi$  then assign job  $\pi_k(h)$  to the machine
        according to the assignment strategy MF or ECF.
        else assign job  $\pi_k(h)$  to the machine with the earliest available
        time.
        end if
    end for
end for
    
```

In order to illustrate how Algorithm LS works, an example of five jobs to be scheduled in a two-stage hybrid flow shop, with three machines at each stage is considered. Tables 9.1 and 9.2 provide the data for the job set and machine set,

**Table 9.2** Data for the machine set

Machine	$M_{1,1}$	$M_{2,1}$	$M_{3,1}$	$M_{1,2}$	$M_{2,2}$	$M_{3,2}$
$V_{j,k}$	2	1	3	1	2	3
$W_{j,k}$	10	6	12	4	8	10



**Fig. 9.3** A numerical example of Algorithm LS

respectively. And Fig. 9.3 demonstrates a full schedule decoded from a job permutation  $\pi_1 = 1, 2, 3, 4, 5$ . It should be noted that Algorithm LS will build a non-permutation schedule, that is, the sequence of jobs at each stage may be different. In this example, we assume that Algorithm LS adopts a Makespan-First strategy.

### 9.4.3 Pheromone Trails

As stated in the previous section, each solution is coded as a permutation of jobs in the algorithm PVACS. Another important issue before applying the algorithm is to define the pheromone trails. Instead of using a single pheromone matrix for all the objectives, the algorithm maintains a separate pheromone matrix for each objective, respectively. Such arrangement can distinguish impacts from different objectives. Let pheromone trail  $\tau_{MK}(j, h)$  represent the sequence desire of job  $h$  to follow job  $j$  for the make-span criterion, and  $\tau_{EC}(j, h)$  for the energy-consumption criterion. The range of possible pheromone trails on each solution component is limited to an interval  $[\tau_{min}, \tau_{max}]$  to avoid premature convergence. In the initialization phase, the pheromone information are initialized to  $\tau_{max}$ , achieving in this way a higher exploration of solution space at the beginning. Furthermore, a preference vector  $\mathbf{u} = (u_{MK}, u_{EC})$  ( $0 \leq u_{MK}, u_{EC} \leq 1$ ) provided by users is introduced to determines the relative importance of different objectives so that we can narrow the search and magnify certain portions of the Pareto front that are of

particular interest to the users.  $u_{MK}$  is the vector component for the make-span criterion, and  $u_{EC}$  for the energy-consumption criterion.

#### 9.4.4 Heuristic Information

Heuristic information is an optional ingredient in ACO, but usually the use of heuristic information to direct the ants' probabilistic solution construction is important because it provides problem-specific knowledge. In the original Ant Colony System, the heuristic information  $\eta(j, h)$  is defined by the reciprocal of a cost measure from node  $j$  to node  $h$ . In this study, Heuristic information used is based on Palmer's heuristic [68], which is a quick method to obtain a near-optimum solution for flow-shop problems. This heuristic measures the slope index of job  $j$  ( $SI_j$ ) using the following equation and then constructs the sequence based on the non-increasing order of the magnitude of  $SI_j$ .

$$SI_j = \sum_{k=1}^s (2k - s - 1)P_{j,k} \quad j = 1, 2, \dots, n \quad (9.2)$$

The idea of Palmer's heuristic is to assign priority to the jobs that have the strongest tendency to progress from short- processing times to long-processing times in the sequence of processes. The heuristic information  $\eta(j, h)$  in PVACS is defined by  $\eta(i, j) = SI_i - \min(SI) + 1$ .

#### 9.4.5 State-Transition Rule

When constructing a job permutation for the first stage, an ant  $a$  at the current job  $j$  selects the next job  $h$  to be added to the job permutation from  $\Phi_a(j)$ , where  $\Phi_a(j)$  represents the set of candidate jobs that have not been added to the job permutation. The state-transition rule is given as follows:

$$y = \begin{cases} \arg \max_{h \in \Phi_a(j)} \left\{ [u_{MK}\tau_{MK}(j, h) + u_{EC}\tau_{EC}(j, h)]^\alpha [\eta(j, h)]^\beta \right\} & \text{if } q \leq q_0 \\ Y & \text{otherwise} \end{cases} \quad (9.3)$$

where  $q$  is a random number uniformly distributed in  $[0, 1)$ ,  $q_0$  is a parameter ( $0 \leq q_0 < 1$ ) set by the user that determines the relative importance of exploitation versus exploration. Additionally, the random variable  $Y$  is selected according to the probability distribution given:

$$Y = \begin{cases} \frac{[u_{MK}\tau_{MK}(j, h) + u_{EC}\tau_{EC}(j, h)]^\alpha [\eta(j, h)]^\beta}{\sum_{l \in \Phi_a(j)} [u_{MK}\tau_{MK}(j, l) + u_{EC}\tau_{EC}(j, l)]^\alpha [\eta(j, l)]^\beta} & \text{if } h \in \Phi_a(j) \\ 0 & \text{otherwise} \end{cases} \quad (9.4)$$

This probability distribution is biased by the parameters  $\alpha$  and  $\beta$ , which determines the relative importance of the pheromone trails and heuristic information, respectively.

### 9.4.6 Pheromone Trail Update

A local pheromone update is performed once an artificial ant has found a solution. The amount of pheromone on the solution components  $\tau_{\text{MK}}(j, h)$  and  $\tau_{\text{EC}}(j, h)$  is decreased for the make-span objective and the energy-consumption objective, respectively. The local pheromone update rule for these solution components can be represented as follows:

$$\tau_{\text{MK}}(j, h) = (1 - \rho_l) \cdot \tau_{\text{MK}}(j, h) + \rho_l \cdot \tau_0 \quad (9.5)$$

$$\tau_{\text{EC}}(j, h) = (1 - \rho_l) \cdot \tau_{\text{EC}}(j, h) + \rho_l \cdot \tau_0 \quad (9.6)$$

where  $\tau_0$  is the initial value of pheromone trails and  $\rho_l$  ( $0 \leq \rho_l \leq 1$ ) is the local evaporation rate. On account of local updating, the ants prefer those sequences that have not yet been constructed. As a result, the diversity of the solutions provided is enhanced.

After all the ants in the population have completed constructing the job permutation, global pheromone information is updated to increase the pheromone values on solution components that have been found in high-quality solutions. This is done first by lowering the pheromone trails at the global evaporation rate  $\rho_g$  and then by allowing the ants to deposit pheromone on the paths searched. A set of non-dominated solutions (NDS) are used to update pheromone trails. The set of non-dominated solutions is used to preserve all the non-dominated solutions previously found. The set is updated after each iteration. If some elements in the set are Pareto dominated by a new found solution, then they will be removed from the set. And new found non-dominated solutions are added to the set. However, the amount of pheromone deposited by ants differs in terms of solution quality. The global pheromone trail update rule is given below:

$$\tau_{\text{MK}}(j, h) = (1 - \rho_g) \cdot \tau_{\text{MK}}(j, h) + \rho_g \cdot \sum_{\text{sol} \in \text{NDS}} \Delta \tau_{\text{MK}}^{\text{sol}}(j, h) \quad (9.7)$$

$$\tau_{\text{EC}}(j, h) = (1 - \rho_g) \cdot \tau_{\text{EC}}(j, h) + \rho_g \cdot \sum_{\text{sol} \in \text{NDS}} \Delta \tau_{\text{EC}}^{\text{sol}}(j, h) \quad (9.8)$$

The amount of pheromone deposited by a solution (sol) follows these equations:

$$\Delta \tau_{\text{MK}}^{\text{sol}}(j, h) = \begin{cases} \frac{\text{val}_{\text{MK}}^{\text{best}}}{\text{val}_{\text{MK}}^{\text{sol}} - \text{val}_{\text{MK}}^{\text{best}} + 1} & \text{if } h \text{ follows } j \text{ in the solution sol} \\ 0 & \text{otherwise} \end{cases} \quad (9.9)$$

$$\Delta\tau_{EC}^{sol}(j, h) = \begin{cases} \frac{val_{EC}^{best}}{val_{EC}^{sol} - val_{EC}^{best} + 1} & \text{if } h \text{ follows } j \text{ in the solution } sol \\ 0 & \text{otherwise} \end{cases} \quad (9.10)$$

where  $val_{MK}^{sol}$  and  $val_{MK}^{best}$  are the make-span value of the solution  $sol$  and the best make-span value in the set of non-dominated solutions, respectively. Similarly,  $val_{EC}^{sol}$  and  $val_{EC}^{best}$  are the energy-consumption value of the solution  $sol$  and the best energy-consumption value in the set of non-dominated solutions, respectively.

It is possible that sometimes the algorithm might get trapped in local minima. This may happen if at each choice point, the pheromone trail is significantly higher for one choice than for all the others. In order to avoid this situation and maintain the diversity of solutions, explicit limits  $\tau_{min}$  and  $\tau_{max}$  are imposed on the minimum and maximum pheromone trails such that for all pheromone trails  $\tau(j, h)$ ,  $\tau_{min} \leq \tau(j, h) < \tau_{max}$ . After each update if  $\tau(j, h) < \tau_{min}$ , then  $\tau(j, h)$  is set to  $\tau_{min}$ ; analogously  $\tau(j, h)$  is set to  $\tau_{max}$  if  $\tau(j, h) > \tau_{max}$ . This ensures that every path has at least a small amount of pheromone, thus the probability of choosing any solution component is never 0.

The algorithm will stop after a preset number of iterations. The flowchart of the proposed PVACS algorithm is given in Fig. 9.4.

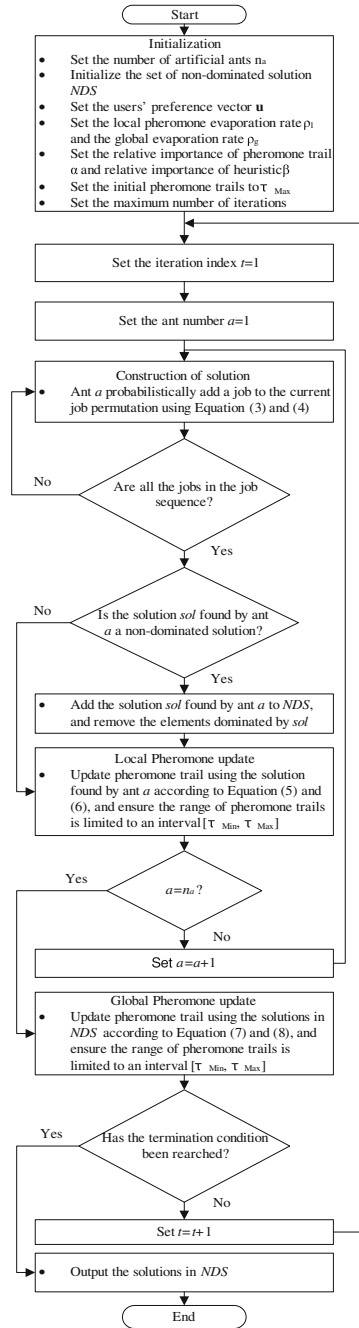
## 9.5 Computational Experiments

In the following section computational experiments are described. The experiments are carried out to compare the solution quality and performance of the proposed PVACS with two well-known algorithms NSGA-II and SPEA2. Both the algorithms use the same encoding method as PVACS, that is, solutions are represented as permutation of jobs, and Algorithm LS is then employed to obtain a valid schedule. In addition, for NSGA-II [31] and SPEA2 [29], the same genetic operators are used: binary tournament selection, single-point crossover and a single-point mutation. The descriptions of the crossover and mutation operators can be found in [69]. All the algorithms are coded in C#; a Core 2, 2 GHz computer with 2 GB RAM was used to run the experiments.

### 9.5.1 Data Generation

The experiments are implemented using randomly generated data. Different problem sizes are taken into consideration. Each test problem can be denoted by a triple  $(J, S, M)$ , where  $J$  is the number of jobs,  $S$  the number of stages and  $M$  the number of machines at each stage. The test problems proposed are (10, 2, 2), (10, 4, 4), (20, 2, 2), (20, 4, 4), (50, 2, 2) and (50, 4, 4). For all the problems the

**Fig. 9.4** The flowchart of the proposed PVACS algorithm



following assumptions are made: processing times of jobs at each stage are generated from a discrete uniform distribution of  $U(10, 50)$ ; machine speeds are from a discrete uniform distribution of  $U(2, 5)$  and machine powers are uniformly distributed over interval [10, 30].

### 9.5.2 Performance Measure

To compare the quality of three Pareto sets obtained from different algorithms, the following performance metrics are considered.

*Number of Pareto solutions metric* (NPS): this metric presents the number of Pareto-optimal solutions which is obtained by each algorithm.

*Coverage metric* ( $C$ ): this metric is a relative measure which allows clearly differentiating two sets  $A$  and  $B$  [70]. The value of  $C(A, B)$  represents the percentage of solutions in  $B$  dominated by at least one solution in  $A$ . It can be calculated by the equation:

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \succ b \text{ or } a = b\}|}{|B|} \quad (9.11)$$

where  $a \succ b$  represents that  $a$  dominates  $b$ . The closer the value of  $C(A, B)$  is to 1, which means almost all the solutions in  $B$  are dominated by some solution in  $A$ , the better the set  $A$  compared to  $B$ . However, this metric is not symmetrical, that is,  $C(A, B) = 1 - C(B, A)$  usually does not hold. Consequently, it is necessary to calculate  $C(B, A)$  and  $A$  is better than  $B$  if  $C(A, B) > C(B, A)$ .

*Users preference metric* (UP): this metric is used to measure whether the solutions in set  $A$  are interesting to users according to the predefined preference vector  $\mathbf{u} = (u_{MK}, u_{EC})$ . The UP value of set  $A$  is given as follows:

$$UP(A) = \frac{\sum_{sol \in A} \sqrt{u_{MK} \left( \frac{val_{MK}^{sol}}{val_{MK}^{best}} - 1 \right)^2 + u_{EC} \left( \frac{val_{EC}^{sol}}{val_{EC}^{best}} - 1 \right)^2}}{|A|} \quad (9.12)$$

The UP metric uses a weighted normalized distance to measure the quality of a set of non-dominated solutions. If the make-span criterion is more preferable for the users, the distance between the make-span value of solution  $sol$  and the best make-span value will be assigned a higher weight, similarly the energy-consumption criterion. The lower a UP value is, the better is the solution quality.

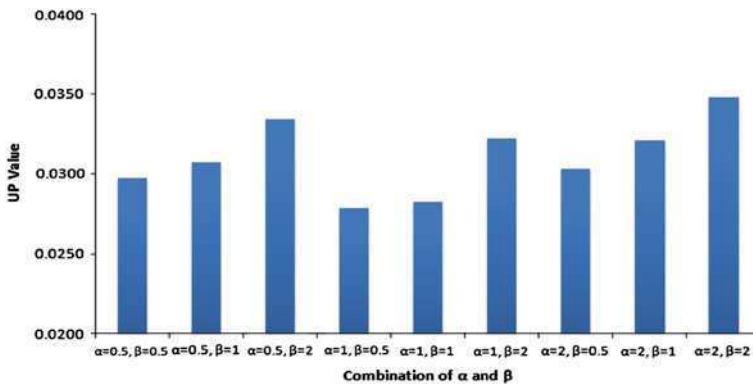
### 9.5.3 Sensitivity Analysis of the Parameters

Sensitivity analysis of the parameters is important because it can determine the influence of input parameters on the output. The study of such influence may help



**Table 9.3** Parameter selection for PVACS

Parameters	Tested values	Selected values
$n_a$	$n_a = 10\sqrt{J}$	$n_a = 10\sqrt{J}$
$\alpha$	{0.5, 1, 2}	$\alpha = 1$
$\beta$	{0.5, 1, 2}	$\beta = 0.5$
$\rho_l$	{0.05, 0.1, 0.2, 0.3}	$\rho_l = 0.1$
$\rho_g$	{0.05, 0.1, 0.2, 0.3}	$\rho_g = 0.3$
$\tau_{MK}^{\min}$	{10, 20, 30}	$\tau_{MK}^{\min} = 10$
$\tau_{MK}^{\max}$	{200, 300, 500}	$\tau_{MK}^{\max} = 500$
$\tau_{EC}^{\min}$	{20, 50, 100}	$\tau_{EC}^{\min} = 50$
$\tau_{EC}^{\max}$	{500, 1000, 2000, 5000}	$\tau_{EC}^{\max} = 2000$
<i>iter</i>	{100, 500, 1000}	<i>iter</i> = 1000

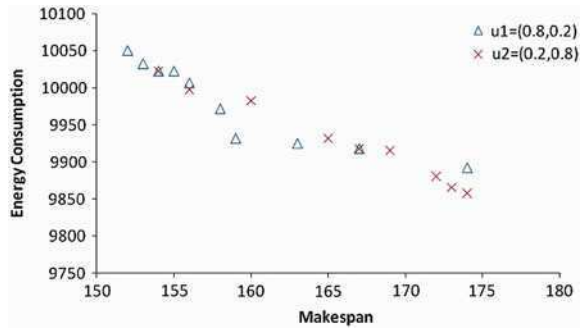


**Fig. 9.5** The best value found for the combination of  $\alpha$  and  $\beta$

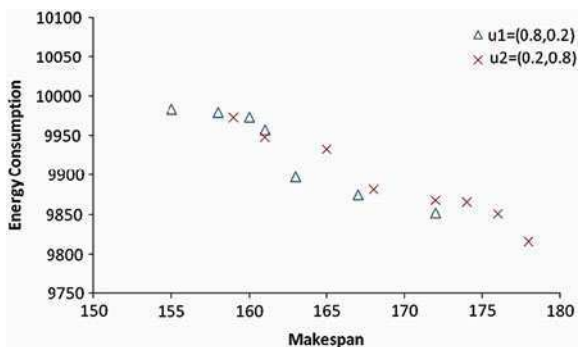
to investigate the robustness of the algorithm and improve its performance. There are a number of parameters that affect the search behaviour of the proposed PVACS. Some of the parameters are basic elements of ACO, such as the number of artificial ants  $n_a$ , the number of iterations *iter*, the relative importance of pheromone trails  $\alpha$ , the relative importance of heuristic information  $\beta$ , the local evaporation rate  $\rho_l$  and the global evaporation rate  $\rho_g$ . Some others are developed to define users' preference, i.e., preference vector  $\mathbf{u} = (u_{MK}, u_{EC})$  and the assignment strategy in Algorithm LS.

A group of values were selected and tested for each parameter. Extensive experiments were carried out by varying one parameter at each step using a trial and error approach, as described by Yagmahan and Yenisey [67]. An instance of problem (20, 2, 2) was used, and the criterion used to measure the quality of the obtained solutions is the UP metric. The preference vector was assumed to be  $\mathbf{u} = (0.5, 0.5)$ , and the Algorithm LS adopts a Makespan-First strategy. It should be noted that using a dynamic population size is preferable, therefore the number of ants was set to  $n_a = 10\sqrt{J}$ , where  $J$  is the number of jobs. The parameters and

**Fig. 9.6** The distribution of the non-dominated solutions under MF strategy



**Fig. 9.7** The distribution of the non-dominated solutions under ECF strategy



their values are shown in Table 9.3. In addition, Fig. 9.5 presents the UP metric values from combination of  $\alpha$  and  $\beta$  (the lower the better).

We next examined how the users’ preferences and the assignment strategies in Algorithm LS affect the search behaviour of PVACS. For the preference vector, two values  $\mathbf{u}_1 = (0.8, 0.2)$  and  $\mathbf{u}_2 = (0.2, 0.8)$  were considered, and two strategies Makespan-First (MF) and Energy Conservation-First (ECF) for the assignment strategy in Algorithm LS. Figure 9.6 reports the distribution of the non-dominated solutions with respect to preference vector  $\mathbf{u}_1$  and  $\mathbf{u}_2$  using the MF strategy in Algorithm LS, and Fig. 9.7 reports the results using ECF strategy. It can be observed that the distribution of non-dominated solutions is significantly affected by the predefined preference vector. If the user prefers the make-span criterion and thus gives it a larger weight, then the algorithm will be more likely to search for non-dominated solutions with better make-span values, and vice versa. The assignment strategies do not directly determine the shape of the distribution. However, using an ECF strategy will significantly reduce the energy-consumption values of all the solutions found, but result in an increase in the make-span values accordingly.

In addition, the parameters needed to determine for NSGA-II and SPEA2 include the population size, archive size, crossover probability, mutation probability and

**Table 9.4** Parameter selection for NSGA-II and SPEA2

Problem size	NSGA-II		SPEA2	
$J = 10$	Population size = 40	Crossover probability = 0.9	Population size = 40 Archive size = 20	Crossover probability = 0.9
$J = 20$	Population size = 60	Mutation probability = 0.05	Population size = 60 Archive size = 30	Mutation probability = 0.05 Iterations = 1000
$J = 50$	Population size = 100	Iterations = 1000	Population size = 100 Archive size = 50	

**Table 9.5** Comparison of the three algorithms using the NPS metric

Problem	PVACS			NSGA-II			SPEA2		
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
(10, 2, 2)	4	9	6.3	4	8	6.5	3	9	6.5
(10, 4, 4)	6	10	6.9	5	11	6.8	5	9	6.7
(20, 2, 2)	4	12	7.0	4	11	6.7	4	11	7.0
(20, 4, 4)	6	13	8.7	5	13	8.7	6	13	8.4
(50, 2, 2)	5	12	8.0	5	14	8.1	5	14	7.8
(50, 4, 4)	5	12	8.8	7	14	8.6	6	13	8.4

the number of iterations. After pilot experiments, the parameter values are summarized in Table 9.4.

### 9.5.4 Experimental Results

This section investigates the performance of PVACS against NSGA-II and SPEA2 in terms of the three metrics listed previously. For each problem, ten instances were randomly generated. We set the preference vector  $\mathbf{u} = (0.5, 0.5)$  and the Makespan-First strategy was used for PVACS.

Table 9.5 presents the number of Pareto-optimal solutions obtained by the three algorithms. The left column of each algorithm represents the minimum number of the Pareto-optimal solutions found among the ten runs, while the middle column and right column represent the maximum and average number, respectively. It can be observed that all the three algorithms have very similar performance. PVACS

**Table 9.6** Comparison of the three algorithms using the  $C$  metric

Problem	C(PVACS, NSGA-II)			C(NSGA-II, PVACS)			C(PVACS, SPEA2)			C(SPEA2, PVACS)		
	Best	Worst	Avg.	Best	Worst	Avg.	Best	Worst	Avg.	Best	Worst	Avg.
(10,2,2)	1.00	0.00	0.32	0.60	0.00	0.20	0.89	0.33	0.55	0.33	0.00	0.14
(10,4,4)	0.90	0.20	0.51	0.57	0.00	0.27	1.00	0.20	0.48	0.40	0.00	0.12
(20,2,2)	0.86	0.17	0.44	0.60	0.00	0.26	1.00	0.40	0.63	0.33	0.00	0.05
(20,4,4)	0.89	0.37	0.57	0.42	0.00	0.13	0.91	0.37	0.69	0.10	0.00	0.01
(50,2,2)	1.00	0.40	0.68	0.37	0.00	0.07	1.00	0.57	0.81	0.00	0.00	0.00
(50,4,4)	0.92	0.33	0.73	0.10	0.00	0.01	1.00	0.67	0.87	0.00	0.00	0.00

performs slightly better than the other two algorithms in terms of the average performance. However, there is generally no considerable difference between the PVACS and NSGA-II. It is also noted that all the algorithms are able to find more Pareto-optimal solutions when the number of jobs, stages and machines increases, as in this case the search space becomes much larger.

The NPS criterion is only able to evaluate the number of Pareto-optimal solutions obtained by an algorithm, however, the solution quality can be measured by the two set coverage metric. The best, worst and average values of the  $C$  metric over 10 instances are summarized in Table 9.6. As the  $C$  metric is not symmetrical, it is necessary to calculate both  $C(A, B)$  and  $C(B, A)$  to determine which algorithm is better. Algorithm  $A$  is better than  $B$  if  $C(A, B) > C(B, A)$ . When comparing PVACS with NSGA-II, it can be noticed that PVACS obtains good results in terms of the best, worst and average performance, and it clearly outperforms NSGA-II. The advantage of PVACS is even more impressive when compared to SPEA2. Such advantage of PVACS may come from the heuristic information incorporated in the algorithm, which enables the algorithm to search the solution space more effectively.

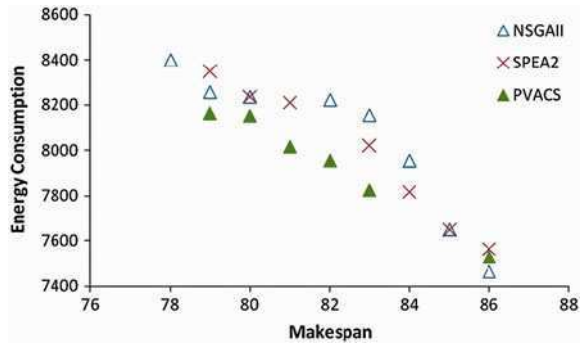
Table 9.6 reports UP metric values produced by the three algorithms. This metric is used to measure whether an obtained Pareto set are of interest to the users according to the predefined preference vector. However, as traditional NSGA-II and SPEA2 are GA-based MOEAs that are usually without preference consideration, the non-dominated solutions they found are evenly distributed in the Pareto frontier in most cases. Consequently, the UP metric values of PVACS are significantly better than both NSGA-II and SPEA2 (the lower the better), and thus are more attractive to the users.

In order to visualize the performance of the different algorithms, three instances generated from (10, 4, 4), (20, 4, 4) and (50, 4, 4) are selected to provide graphical representation for the small (Fig. 9.8), medium (Fig. 9.9) and large problems (Fig. 9.10). These figures illustrate and confirm some conclusions derived from the numerical analysis based on the metric values. As we can observe, the proposed PVACS algorithm is able to provide better solutions than NSGA-II and SPEA2 in terms of quality and distribution.

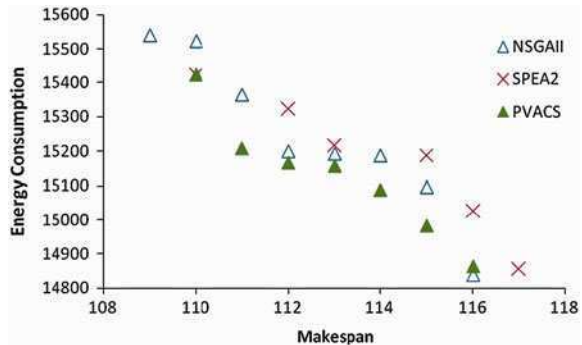
**Table 9.7** Comparison of the three algorithms using the *UP* metric

Problem	PVACS			NSGA-II			SPEA2		
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
(10, 2, 2)	0.0095	0.0164	0.0115	0.0124	0.0315	0.0195	0.0135	0.0514	0.0351
(10, 4, 4)	0.0132	0.0327	0.0283	0.0552	0.0940	0.0762	0.0538	0.0925	0.0753
(20, 2, 2)	0.0159	0.0415	0.0212	0.0296	0.0834	0.0571	0.0492	0.0817	0.0671
(20, 4, 4)	0.0217	0.0355	0.0241	0.0348	0.0952	0.0712	0.0365	0.0983	0.0856
(50, 2, 2)	0.0122	0.0326	0.0178	0.0251	0.0635	0.0454	0.0254	0.0716	0.0524
(50, 4, 4)	0.0143	0.0411	0.0223	0.0399	0.0731	0.0645	0.0481	0.0881	0.0764

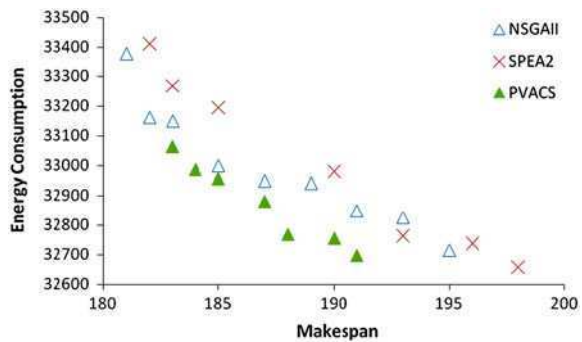
**Fig. 9.8** Non-dominated solutions for the small problem



**Fig. 9.9** Non-dominated solutions for the medium problem



**Fig. 9.10** Non-dominated solutions for the large problem



## 9.6 Concluding Remarks

In this chapter, we have discussed some important issues related to green manufacturing. Energy conservation has now become a major concern in green manufacturing. In view of the increasing importance of energy conservation, a multi-objective scheduling model considering to minimize both make-span and energy consumption has been proposed. As the two objectives are usually in conflict with each other, a PVACS has been developed to obtain a set of Pareto-optimal solutions. PVACS searches for feasible job permutations and a List Scheduling (LS) heuristic is then adopted to decode the solution and obtain a valid schedule. PVACS allows the users to define a preference vector so that the search is focused on the specific areas which are of particular interest to users. The performance of PVACS has been compared to SPEA2 and NSGA-II using three different metrics: *number of Pareto solutions*, *coverage metric* and *users preference metric*. The experimental results show that PVACS outperforms the other two algorithms.

There are a number of important directions for future research. First of all, the solutions for the proposed model are encoded as job permutations. In this coding scheme PVACS is only able to search for solutions in a small portion of the search space. Therefore, different solution encoding techniques may be considered to better explore the search space. Secondly, the heuristic information and the pheromone update rule of PVACS may be further improved. Some local search procedures can be incorporated to make the algorithm more effective and efficient. Thirdly, different heuristics for assigning jobs to machines may also improve the performance of PVACS. In addition, other multi-objective evolutionary algorithms can be developed to solve the problem. Finally, it also seems interesting to extend the model to the unrelated machine environment, or a more complex job-shop environment.

**Acknowledgments** The authors would like to thank Xiaolin Li, Qi Tan, Song Zhang for technical assistance. This work was supported by National Natural Science Foundation of China (70821001), Research Fund for the Doctoral Program of Higher Education of China (200803580024), HKSAR ITF (GHP/042/07LP), HKSAR RGC GRF (HKU 712508E), and HKU Research Committee grants.

## References

1. U.N. Environment Programme. (2010). *Green economy report: a preview*.
2. Institute, T. M. (2009). *The facts about modern manufacturing* (8th ed.).
3. US Department of Commerce, E.a.S.A. (2010). *Measuring the green economy*.
4. <http://www.eia.doe.gov/emeu/aer/consump.html>.
5. Tacconi, L. (2000). *Biodiversity and ecological economics: Participation, values, and resource management*. London: Earthscan/James & James.
6. US Energy Information Administration. (2009). *International Energy Outlook 2009*
7. Mouzon, G., Yildirim, M. B., & Twomey, J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, 45, 4247–4271.

8. Lei, D. M. (2009). Multi-objective production scheduling: a survey. *International Journal of Advanced Manufacturing Technology*, 43(9–10), 926–938.
9. Ross, M. (1992). Efficient energy use in manufacturing. *Proceedings of the National Academy of Sciences of the United States of America*, 89(3), 827–831.
10. Park, S. H., Dissmann, B., & Nam, K. Y. (1993). A cross-country decomposition analysis of manufacturing energy-consumption. *Energy*, 18(8), 843–858.
11. Fromme, J. W. (1996). Energy conservation in the Russian manufacturing industry—Potentials and obstacles. *Energy Policy*, 24(3), 245–252.
12. Golove, W. H., & Schipper, L. J. (1996). Long-term trends in US manufacturing energy consumption and carbon dioxide emissions. *Energy*, 21(7–8), 683–692.
13. Adenikinju, A. F. (1998). Productivity growth and energy consumption in the Nigerian manufacturing sector: A panel data analysis. *Energy Policy*, 26(3), 199–205.
14. Bentzen, J. (2004). Estimating the rebound effect in US manufacturing energy consumption. *Energy Economics*, 26(1), 123–134.
15. Draganescu, F., Gheorghie, M., & Doicin, C. V. (2003). Models of machine tool efficiency and specific consumed energy. *Journal of Materials Processing Technology*, 141(1), 9–15.
16. Dietmair, A., & Verl, A. (2009). Energy consumption forecasting and optimization for tool machines. *Modern Machinery Science Journal*, 62–67.
17. Herrmann, C., & Thiede, S. (2009). Process chain simulation to foster energy efficiency in manufacturing. *CIRP Journal of Manufacturing Science and Technology*, 1(4), 221–229.
18. Wolters, W. T. M., Lambert, A. J. D., & Claus, J. (1995). Sequencing problems in designing energy efficient production systems. *International Journal of Production Economics*, 41(1–3), 405–410.
19. Park, C. W., et al. (2009). Energy consumption reduction technology in manufacturing—A selective review of policies, standards, and research. *International Journal of Precision Engineering and Manufacturing*, 10(5), 151–173.
20. Fonseca, C., & Fleming, P. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1), 1–16.
21. Coello, C. A. C. (2006). Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1), 28–36.
22. Schaffer, J. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pp. 93–100
23. Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-wesley.
24. Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3), 221–248.
25. Horn, J., Nafpliotis, N., & Goldberg, D. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*. IEEE World Congress on Computational Intelligence (pp. 418)
26. Fonseca, C., Fleming, P. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 416–423
27. Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271.
28. Rudolph, G., & Agapie, A. (2000). Convergence properties of some multi-objective evolutionary algorithms. In *Proceedings of the 2000 Congress on Evolutionary Computation* (Vols 1 and 2, pp. 1010–1016)
29. Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. In *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pp. 95–100
30. Knowles, J., & Corne, D. (2000). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2), 149–172.

31. Deb, K., et al. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
32. Chaudhuri, S., & Deb, K. (2010). An interactive evolutionary multi-objective optimization and decision making procedure. *Applied Soft Computing*, 10(2), 496–511.
33. Doerner, K., et al. (2004). Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of Operations Research*, 131(1–4), 79–99.
34. Doerner, K. F., et al. (2006). Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, 171(3), 830–841.
35. Xia, W. J., & Wu, Z. M. (2005). An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 48(2), 409–425.
36. Sedenka, V., & Raida, Z. (2010). Critical comparison of multi-objective optimization methods: Genetic algorithms versus swarm intelligence. *Radioengineering*, 19(3), 369–377.
37. Berrichi, A., et al. (2010). Bi-objective ant colony optimization approach to optimize production and maintenance scheduling. *Computers & Operations Research*, 37(9), 1584–1596.
38. Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1), 1–9.
39. Tan, K. C., Lee, T. H., & Khor, E. F. (2002). Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. *Artificial Intelligence Review*, 17(4), 253–290.
40. Zitzler, E., et al. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2), 117–132.
41. Pinedo, M. (2002). *Scheduling: theory, algorithms and systems*. Upper Saddle River, NJ: Prentice-Hall.
42. Gupta, J. (1988). Two-stage, hybrid flowshop scheduling problem. *The Journal of the Operational Research Society*, 39(4), 359–364.
43. Linn, R., & Zhang, W. (1999). Hybrid flow shop scheduling: A survey. *Computers & Industrial Engineering*, 37(1–2), 57–61.
44. Ribas, I., Leisten, R., & Framinan, J. M. (2010). Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*, 37(8), 1439–1454.
45. Ruiz, R., & Vazquez-Rodriguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1), 1–18.
46. Huang, W., & Li, S. (1998). A two-stage hybrid flowshop with uniform machines and setup times. *Mathematical and Computer Modelling*, 27(2), 27–45.
47. Dessouky, M. M., Dessouky, M. I., & Verma, S. K. (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research*, 109(3), 620–631.
48. Soewandi, H., & Elmaghraby, S. E. (2003). Sequencing on two-stage hybrid flowshops with uniform machines to minimize makespan. *IIE Transactions*, 35(5), 467–477.
49. Kyparisis, G. J., & Koulamas, C. (2006). A note on makespan minimization in two-stage flexible flow shops with uniform machines. *European Journal of Operational Research*, 175(2), 1321–1327.
50. Bertel, S., & Billaut, J. C. (2004). A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research*, 159(3), 651–662.
51. Sevastianov, S. V. (2002). Geometrical heuristics for multiprocessor flowshop scheduling with uniform machines at each stage. *Journal of Scheduling*, 5(3), 205–225.
52. Kyparisis, G. J., & Koulamas, C. (2001). A note on weighted completion time minimization in a flexible flow shop. *Operations Research Letters*, 29(1), 5–11.
53. Kyparisis, G. J., & Koulamas, C. (2006). Flexible flow shop scheduling with uniform parallel machines. *European Journal of Operational Research*, 168(3), 985–997.



54. Verma, S., & Dessouky, M. (1999). Multistage hybrid flowshop scheduling with identical jobs and uniform parallel machines. *Journal of Scheduling*, 2(3), 135–150.
55. Voss, S., & Witt, A. (2007). Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *International Journal of Production Economics*, 105(2), 445–458.
56. Graham, R., et al. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5(2), 287–326.
57. Colomi, A., Dorigo, M., Maniezzo, V. (1991). Distributed optimization by ant colonies. In *Proceedings of the First European Conference on Artificial Life* (pp. 134–142)
58. Colomi, A., et al. (1994). Ant system for job-shop scheduling. *Journal of Operations Research and Statistic Computing Science*, 34(1), 39–53.
59. Dorigo, M., Maniezzo, V., & Colomi, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 26(1), 29–41.
60. Rajendran, C., & Ziegler, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 155(2), 426–438.
61. Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, 6(4), 333–346.
62. Blum, C. (2005). Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling. *Computers & Operations Research*, 32(6), 1565–1591.
63. T'Kindt, V., et al. (2002). An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. *European Journal of Operational Research*, 142(2), 250–257.
64. Jayaraman, V. K., et al. (2000). Ant colony framework for optimal design and scheduling of batch plants. *Computers & Chemical Engineering*, 24(8), 1901–1912.
65. Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2–3), 243–278.
66. Mariano, C. E., Morales, E. (1999). MOAQ an Ant-Q algorithm for multiple objective optimization problems. In *Gecco-99: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 894–901.
67. Yagmahan, B., & Yenisey, M. M. (2008). Ant colony optimization for multi-objective flow shop scheduling problem. *Computers & Industrial Engineering*, 54(3), 411–420.
68. Palmer, D. (1965). Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum. *Operations Research Quarterly*, 16(1), 101–107.
69. Damodaran, P., Manjeshwar, P. K., & Srihari, K. (2006). Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms. *International Journal of Production Economics*, 103(2), 882–891.
70. Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8(2), 173–195.

# Chapter 10

## Intelligent Optimisation for Integrated Process Planning and Scheduling

Weidong Li, Lihui Wang, Xinyu Li and Liang Gao

**Abstract** Traditionally, process planning and scheduling were performed sequentially, where scheduling was executed after process plans had been generated. Considering the fact that the two functions are usually complementary, it is necessary to integrate them more tightly so that the performance of a manufacturing system can be improved greatly. In this chapter, a multi-agent-based framework has been developed to facilitate the integration of the two functions. In the framework, the two functions are carried out simultaneously, and an optimization agent based on evolutionary algorithms is used to manage the interactions and communications between agents to enable proper decisions to be made. To verify the feasibility and performance of the proposed approach, experimental studies conducted to compare this approach and some previous works are presented. The experimental results show the proposed approach has achieved significant improvement.

---

W. Li (✉)

Faculty of Engineering and Computing, Coventry University, Coventry, CV1 5FB, UK  
e-mail: weidong.li@coventry.ac.uk

L. Wang

Virtual Systems Research Centre, University of Skövde, 541 28 Skövde, Sweden  
e-mail: lihui.wang@his.se

X. Li · L. Gao

State Key Laboratory of Digital Manufacturing Equipment and Technology,  
Huazhong University of Science and Technology, Wuhan 430074, China  
e-mail: lixinyu@mail.hust.edu.cn

L. Gao

e-mail: gaoliang@mail.hust.edu.cn

## 10.1 Introduction

In a manufacturing system, process planning and scheduling used to link product design and manufacturing are two of the most important functions. A process plan specifies what manufacturing resources and technical operations/routes are needed to produce production jobs for a product. The outcome of process planning includes the identification and specification of machines, tools and fixtures suitable for a job, and the arrangement of operations and processes for the job. Typically, a job may have one or more alternative process plans. With the process plans of jobs as input, a scheduling task is to schedule the operations of all the jobs on machines while precedence relationships in the process plans are satisfied. Although as mentioned above, there is a close relationship between process planning and scheduling, the integration of them is still a challenge in both research and applications [1]. In traditional approaches, process planning and scheduling were carried out in a sequential way. Scheduling was conducted after the process plan had been generated. Those approaches have become an obstacle to improve the productivity and responsiveness of manufacturing systems and to cause the following problems in particular [2, 3]:

- In manufacturing practice, process planner plans jobs individually. For each job, manufacturing resources on the shop floor are usually assigned on it without considering the competition for the resources from other jobs [4]. This may lead to the process planners favouring to select the desirable machines for each job repeatedly. Therefore, the generated process plans are somewhat unrealistic and cannot be readily executed on the shop floor for a group of jobs [5]. Accordingly, the resulting optimal process plans often become infeasible when they are carried out in practice at the later stage.
- Scheduling plans are often determined after process planning. Fixed process plans may drive scheduling plans to end up with severely unbalanced resource load and create superfluous bottlenecks.
- Even though process planners consider the restriction of the current resources on the shop floor, the constraints in the process planning phase may have already changed due to the time delay between the planning phase and execution phase. This may lead to the infeasibility of the optimized process plan. Investigations have shown that 20–30% of the total process plans in a given period have to be modified to adapt to the dynamic change in a production environment [2].
- In most cases, both for process planning and scheduling, a single criterion optimization technique is used to determine the best solution. However, the real production environment is best represented by considering more than one criterion simultaneously [2]. Furthermore, the process planning and scheduling may have conflicting objectives. Process planning emphasizes the technological requirements of a job, while scheduling involves the timing aspects and resource sharing of all jobs. If there is no appropriate coordination, it may create conflicting problems.

To overcome these problems, there is an increasing need for an integrated process planning and scheduling (IPPS) system. The IPPS introduces significant improvements to the efficiency of manufacturing resources through eliminating or reducing scheduling conflicts, reducing flow-time and work-in-process, improving production resources utilizing and adapting to irregular shop floor disturbances [5]. Without IPPS, a true Computer Integrated Manufacturing System (CIMS), which strives to integrate the various phases of manufacturing in a single comprehensive system, may not be effectively realized.

## 10.2 Literature Survey

In the early studies of CIMS, it has been identified that IPPS is very important for the development of CIMS [2, 6]. The preliminary idea of IPPS was first introduced in [7]. In [8], alternative process plans were used to improve the flexibility of manufacturing systems. In [9], the concept of dynamic feedback was introduced into IPPS. The integration model proposed by [9, 10] extended the concepts of alternative process plans and dynamic feedback and defined an expression to the methodology of hierarchical approach. Some earlier works of IPPS had been summarized in [6]. The most recent works related to the IPPS optimization can be generally classified into two categories: the enumerative approach and the simultaneous approach. In the enumerative approach [11–13], multiple alternative process plans are first generated for each part. A schedule can be determined by iteratively selecting a suitable process plan from alternative plans of each part to replace the current plan until a satisfactory performance is achieved. The simultaneous approach is based on the idea of finding a solution from the combined solution space of process planning and scheduling [14–17]. In this approach, the process planning and scheduling are both in dynamic adjustment until specific performance criteria can be satisfied. Although this approach is more effective and efficient in integrating the two functions, it also enlarges the solution search space significantly.

In recent years, the agent-based system used to build up the communicative collaborative framework in manufacturing planning has captured the interest of a number of researchers. In [18], the agent technology for collaborative process planning was reviewed. The focus of the research was on how the agent technology can be further developed in support of collaborative process planning as well as its future research issues and directions in process planning. In [19], a literature review on IPPS was made, particularly on the agent-based approaches for the problem. The advantages of the agent-based approach for scheduling were discussed. In [20], the research on manufacturing process planning, scheduling as well as their integration was summarized. In [21], a multi-agent system, where process routes and schedules of a part are accomplished through the contract net bids, was proposed. IDCPPS is an integrated, distributed and cooperative process planning system [22]. The process planning tasks are

separated into three levels, namely, initial planning, decision-making and detail planning. The results of these three steps are general process plans, a ranked list of near-optimal alternative plans and the final detailed linear process plans, respectively. The integration with scheduling is considered at each stage with process planning. A computerized model that can integrate the manufacturing functions and resolve some of the critical problems in distributed virtual manufacturing was proposed [23]. This integration model is realized through a multi-agent approach that provides a practical approach for software integration in a distributed environment. A multi-agent-based framework for the IPPS problem was introduced [24]. This framework can also be used to optimize the utilization of manufacturing resources dynamically as well as provide a platform on which alternative configurations of manufacturing systems can be assessed. In [25], a new methodology of distributed process planning was developed. It focused on the architecture of the new approach, using multi-agent negotiation and cooperation, and on the other supporting technologies such as machining feature-based planning and function block-based control. An online hybrid agent-based negotiation multi-agent system to integrate process planning with scheduling/rescheduling was proposed [26, 27]. With the introduction of the supervisory control into the decentralized negotiations, this approach is able to provide solutions with a better global performance. A bidding-based multi-agent system for solving IPPS was presented in [28]. The proposed architecture consists of various autonomous agents capable of communicating (bidding) with each other and making decisions based on their knowledge. A new method in IPPS was discussed in [29]. A multi-agent learning-based integration method was devised in the study to solve the conflict between the optimality of the process plan and the production schedule. In the method, each machine makes decisions about process planning and scheduling simultaneously, and it has been modelled as a learning agent using evolutionary artificial neural networks to realize proper decisions resulting from interactions between other machines. An agent-based architecture of an IPPS system for multiple jobs in flexible manufacturing systems was devised [30]. In the literature of agent-based manufacturing applications, much research applied simple algorithms such as dispatching rules which are applicable for real-time decision-making.

To identify optimal solutions in IPSS is a critical and challenging research. Some optimization approaches based on modern heuristics or evolutionary algorithms, such as genetic algorithm (GA) (for operation sequencing problem, [31–35]; for IPPS problem, [16, 36]), simulated annealing (SA) algorithm (for operation sequencing problems, [37, 38]; for IPPS problem, [11, 39]), Tabu search algorithm [40], game theory-based approach for IPPS problems [41], and particle swarm optimization (PSO) algorithm for operation sequencing problems [42], have been developed in the last two decades and significant improvements have been achieved. However, for parts with complex structures and features and multiple parts involved, these optimization processes are well known as complicated decision problems. The major difficulties include: (1) both operation sequencing and IPPS problems are NP-hard (NP: non-deterministic polynomial)

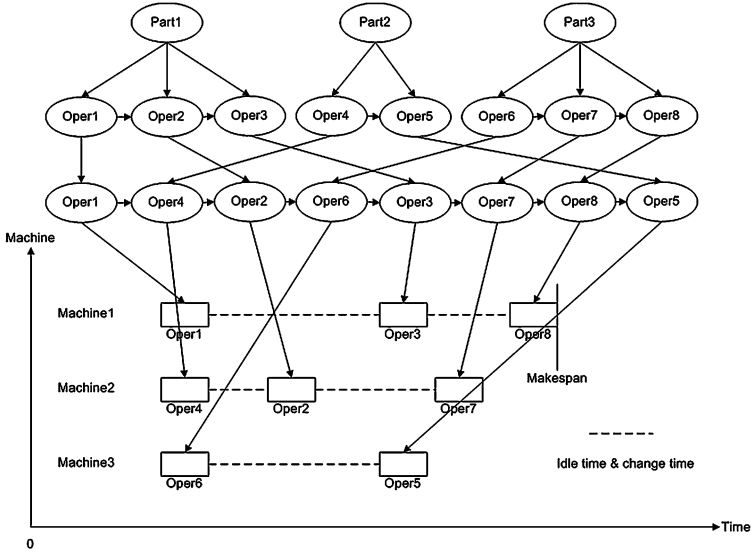


Fig. 10.1 Illustration of IPPS problems

combinatorial optimization problems. The search space is usually very large especially for IPPS problem because it involves multiple parts’ scheduling, and many previously developed methods could not find optimized solutions effectively and efficiently, and (2) there are usually a number of precedence constraints in sequencing operations and manufacturing resource utilization constraints due to manufacturing practice and rules, which make the search more difficult. Therefore, it is necessary to develop efficient models for the operation sequencing and the IPPS optimization problems and the optimization algorithms need to be more agile and efficient to solve practical cases.

### 10.3 IPPS Optimisation Formulation

The IPPS problem can be defined as follows:

Given a set of  $n$  parts that are to be processed on machines with operations including alternative manufacturing resources, select suitable manufacturing resources and sequence the operations so as to determine a schedule in which the precedence constraints among operations can be satisfied and the corresponding objectives can be achieved.

A model of IPPS is shown in Fig. 10.1.

The most popular criteria for scheduling include make-span, job tardiness and the balanced level of machine utilization, while manufacturing cost is the major criterion for process planning:

*Makespan:*

$$\text{Makespan} = \text{Max}_{j=1}^m (\text{Machine}[j].\text{Available\_time}) \quad (10.1)$$

*Total job tardiness.* The due date of a part is denoted as DD, and the completion moment of the part is denoted as CM. Hence,

$$\text{Part\_Tardiness} = \begin{cases} 0 & \text{if DD is later than CM} \\ \text{CM} - \text{DD} & \text{Otherwise} \end{cases} \quad (10.2)$$

*Balanced level of machine utilization:* the Standard Deviation concept is introduced here to evaluate the balanced machine utilization (assuming there are  $m$  machines, and each machine has  $n$  operations).

$$\text{Machine}[j] \cdot \text{Utilization} = \sum_{i=1}^n (\text{Operation}[i] \cdot \text{Mac\_T}) \quad (j = 1, \dots, m) \quad (10.3)$$

$$\chi = \frac{\sum_{j=1}^m (\text{Machine}[j] \cdot \text{Utilization})}{m} \quad (10.4)$$

$$\text{Utilization\_Level} = \sqrt{\sum_{j=1}^m (\text{Machine}[j] \cdot \text{Utilization} - \chi)^2} \quad (10.5)$$

In this problem, the following assumptions are made:

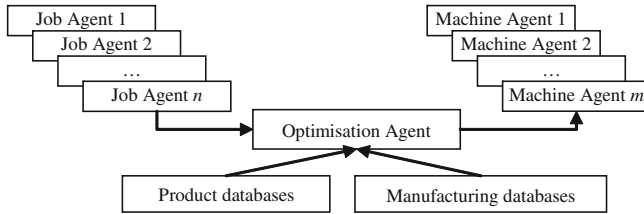
- Jobs are independent. Job pre-emption is not allowed and each machine can handle only one job at a time.
- The different operations of one job cannot be processed simultaneously.
- All jobs and machines are available at time zero simultaneously.
- After a job is processed on a machine, it is immediately transported to the next machine on its process, and the transmission time is assumed to be negligible.
- Set-up time for the operations on the machines is independent of the operation sequence and is included in the processing time.

## 10.4 A Multi-agent System for IPPS

In this research, an IPPS framework has been proposed based on the concept of a multi-agent system. A multi-agent system is a distributed artificial intelligence system that embodies a number of autonomous agents to achieve common goals.

### 10.4.1 A Multi-agent System

The architecture of the multi-agent system developed in this study and the relationships between the agents and their sub-agents are illustrated in Fig. 10.2.



**Fig. 10.2** A multi-agent system for IPPS

In this framework, there are three agents and several databases. The job agents and machine agents are used to represent jobs and machines. The optimization agent is used to optimize the alternative process plans and scheduling plans. With the consideration of the scheduling requirements and availability of manufacturing resources, these agents negotiate with each other to establish the actual process plan of every job and the scheduling plans for all jobs. The detailed description of three types of agents is provided in the following section.

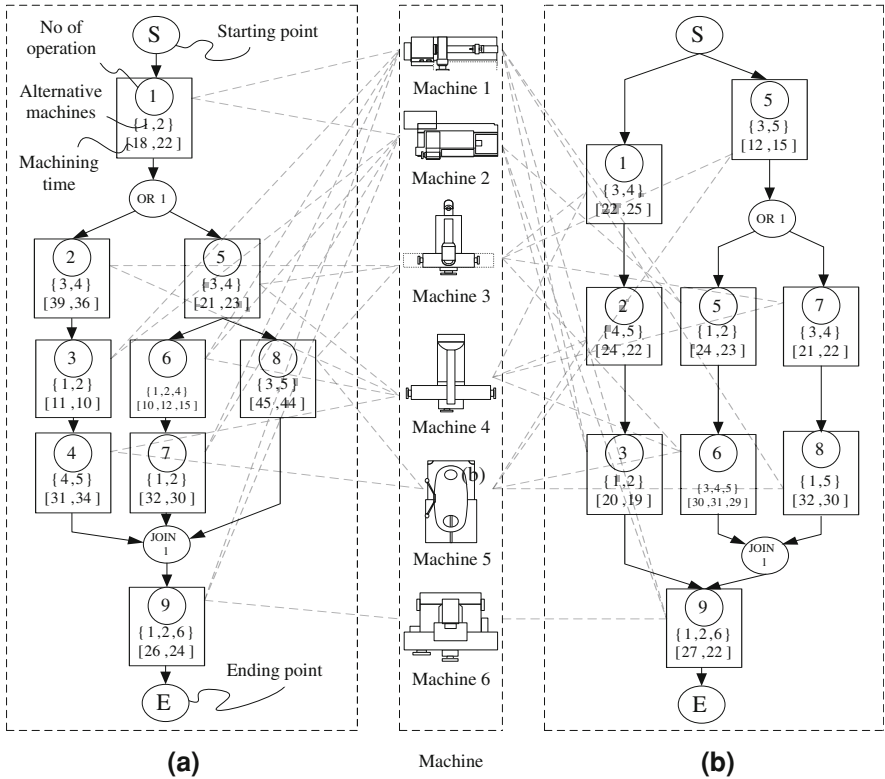
### 10.4.2 Agent Description

*Job agents* represent the jobs to be manufactured on the shop floor. Each agent contains the detailed information of a particular job, which includes job ID, job type, quantity, due date, quality requirements, CAD drawing, tolerance and surface finish requirements, etc. This agent also includes the job status. In this research, the following statuses for the job agents are considered:

- *Idle*: the job agent is idle and waiting for the next manufacturing operations.
- *Manufacturing operation*: the job agent is under manufacturing on a machine. Based on the assumption in the previous section, when the job is under a manufacturing process on a machine, it cannot be processed by other machines. The function of this agent is to provide the job's information to the multi-agent system.

The job agents use the rules from the knowledge database and negotiate with the machine agents to generate all the alternative process plans of each job. Therefore, they contain the information of alternative process plans. There are three types of flexibility considered in process plans [39]: operation flexibility, sequencing flexibility and processing flexibility [39]. Operation flexibility [14], also called routing flexibility, relates to the possibility of performing one operation on alternative machines, with possibly distinct processing time and cost. Sequencing flexibility is decided by the possibility of interchanging the sequence of the required operations. Processing flexibility is determined by the possibility of processing the same manufacturing feature with alternative operations or sequences of operations. Better performance in some criteria (e.g., production time) can be obtained through the consideration of these flexibilities. There are many methods used to describe the





**Fig. 10.3** Alternative process planning and scheduling network. **a** Alternative process plans for Job 1, **b** alternative process plans for Job 2

types of flexibility explained above, such as Petri net, and/or graphs and networks. In this research, a network representation proposed by [13, 14] has been adopted. There are three node types in the network: starting node, intermediate node and ending node. The starting node and the ending node, which are dummy ones, indicate the start and end of the manufacturing process of a job. An intermediate node represents an operation, which contains the alternative machines that are used to perform the operation and the processing time required for an operation according to the machines. The arrows connecting the nodes represent the precedence between them. OR relationships are used to describe the processing flexibility that the same manufacturing feature can be processed by different process procedures. If the links following a node are connected by an OR-connector, they only need to traverse one of the OR-links (the links connected by the OR-connector are called OR-links). OR-link path is an operation path that begins at an OR-link and ends as it merges with the other paths, and its end is denoted by a JOIN-connector. For the links that are not connected by OR-connectors, all of them must be visited. Figure 10.3 shows two jobs alternative process plan networks (job 1 and job 2). In the network of Fig. 10.3b, paths

{5, 6} and {7, 8} are two OR-link paths. For the links that are not connected by OR-connectors, such as {6, 7} and {8} in Fig. 10.3a, all of them must be visited. But they do not have precedence constraint, this means that {6, 7, 8} and {8, 6, 7} are available. The objective of the IPSS problem has been defined in the previous section.

*Machine agents* represent the machines. They read the information from the resource database. Each agent contains the information of the particular machine. The information includes the machine ID, the manufacturing features that this machine can process, the processing time and the machine status. After the job agents are created, the machine agents negotiate with the job agents and determine the jobs' operations to be processed on each machine, and the processing time of these operations is also determined at the same time. In this research, the following statuses for the machine agents are considered: (1) *idle*: the machine agent is idle and waiting for next machining operation; (2) *manufacturing operation*: the machine is processing one job and (3) *breakdown*: the machine has been broken and cannot process any jobs. Based on the assumption in the previous section, when the machine is processing one job, it cannot process other jobs. Each machine agent negotiates with the optimization agent and job agents to get the information that includes the operations' ID to be processed on them, the processing sequence of these operations and the starting time and ending time of each operation. A scheduling plan is then determined. A scheduling plan determines when and how many jobs have to be manufactured within a given period of time. Therefore, this plan has to be carried out according to the current shop floor status. If there are many changes on the shop floor and the determined scheduling plan cannot be carried out, the machine agents need to negotiate with other agents (including jobs agents and optimization agent) to trigger a rescheduling process.

*Optimization agents* are important parts of the proposed multi-agent system. They can optimize the process plans and scheduling plans to get more effective solutions. In order to accomplish this task, the optimization agent explores the search space with the aid of evolutionary algorithms, such as a hybrid SA and GA, a modified GA and a PSO algorithm. In the following section, the PSO algorithm is explained to better understand the mechanism of the optimization agent.

## 10.5 PSO Algorithm for Optimisation Agent

The IPPS problem usually brings forth a vast search space. Conventional algorithms are often incapable of optimizing nonlinear multi-modal functions. To address this problem effectively, some modern evolutionary optimization algorithms, such as PSO and GA, have been developed to quickly find a solution in a large search space through some evolutionary or heuristic strategies. A standard PSO algorithm was inspired by the social behaviour of bird flocking and fish schooling [43]. Three aspects are considered simultaneously when an individual fish or bird (particle) makes a decision about where to move: (1) its current moving direction (velocity) according to the inertia of the movement, (2) the best position

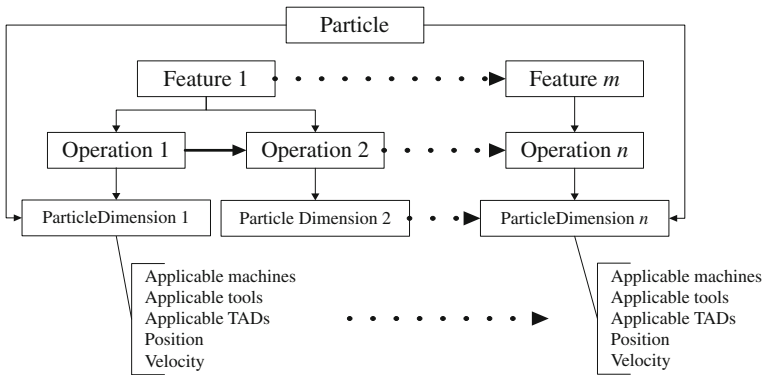


Fig. 10.4 Representation of a process plan (particle)

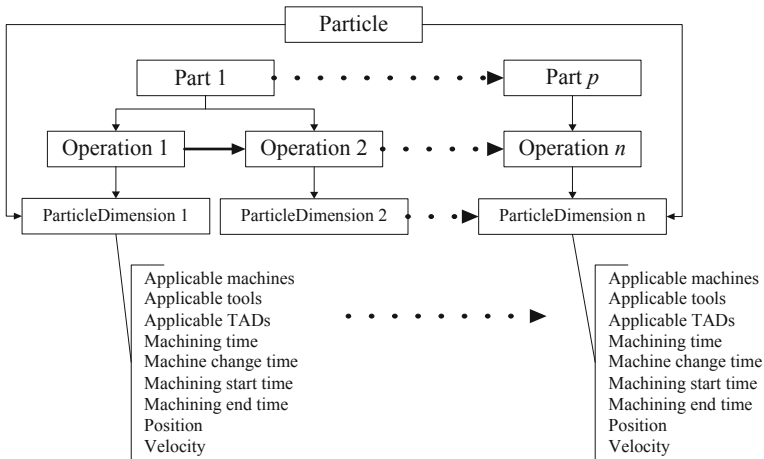


Fig. 10.5 Representation in IPPS (particle)

that it has achieved so far and (3) the best position that its neighbour particles have achieved so far. In the algorithm, the particles form a swarm and each particle can be used to represent a potential solution of a problem. In each iteration, the position and velocity of a particle can be adjusted by the algorithm that takes the above three considerations into account. After a number of iterations, the whole swarm will converge at an optimized position in the search space.

The IPPS problem can be modelled as an extension of the operation sequencing optimization problem into one of multi-parts with scheduling objectives [37, 38, 40, 44]. To achieve this, the representation of the process planning shown in Fig. 10.4 needs to be extended as that of IPPS shown in Fig. 10.5:

- In encoding process. compared to the representation of a process plan shown in Fig. 10.4, several new variables including *Mac\_time*, *Change\_time*,

**Table 10.1** Class definition of a particle dimension (an operation)

<i>Class ParticleDimension: an operation</i>	
Variable	Description
Operation_id	The id of the operation
Machine_id	The id of a machine to execute the operation
Tool_id	The id of a cutting tool to execute the operation
TAD_id	The id of a TAD (tool approach direction) to apply the operation
Machine_list[]	The candidate machine list for executing the operation
Tool_list[]	The candidate tool list for executing the operation
TAD_list[]	The candidate TAD list for applying the operation
Position	The position value of the operation
Velocity	The velocity value of the operation

*Machine\_s\_time* and *Machine\_e\_time* are added in Fig. 10.5 to record and track the time related to the execution of the operation so as to determine the time allocation on the machines. *Mac\_time*, *Change\_time*, *Machine\_s\_time* and *Machine\_e\_time* are set as 0 initially. Table 10.1 shows the extended class definition of a particle dimension (an operation).

- In decoding process. To record the machine utilization status (available time) and operations being executed on every machine (including start time, operation time and end time for each operation) at different times, a *machine* class is defined. When the sequence for all the operations is generated and the manufacturing resources are selected, the assignments of specific operations and machines are determined and therefore the schedule is obtained. By using a number of iterations to update the positions and velocities of the particle dimensions in each particle, an optimized sequence (i.e., an optimized solution) can be achieved eventually.

A traditional PSO algorithm can be applied to optimize IPPS in the following steps:

(1) Initialization

- Set the size of a swarm, e.g., the number of particles and the maximum number of iterations.
- Initialize all the particles (a particle is an IPPS solution) in a swarm. Calculate the corresponding criteria of the particles (a result is called *fitness* here).
- Set the local best particle and the global best particle with the best *fitness* (objective function).

(2) Iterate the following steps until the pre-set maximum iteration time is reached

- For each particle in the swarm, update its velocity and position values.
- Decode the particle into an IPPS solution in terms of new position values and calculate the *fitness* of the particle. Update the local best particle and the global best particle if a lower *fitness* is achieved.

### (3) Decode global best particle to get the optimized solution

However, the traditional PSO algorithm introduced above is still not effective in resolving the operation sequencing problem. There are two major reasons for it:

- Due to the inherent mathematical operators, it is difficult for the traditional PSO algorithm to consider the different arrangements of machines, tools and setups for each operation, and therefore the particle is unable to fully explore the entire search space.
- The traditional algorithm usually works well in finding solutions at the early stage of the search process (the optimization result improves fast), but is less efficient during the final stage. Due to the loss of diversity in the population, the particles move quite slowly with low or even zero velocities and this makes it hard to reach the global best solution. Therefore, the entire swarm is prone to be trapped in a local optimum from which it is difficult to escape.

To solve these two problems and enhance the ability of the traditional PSO algorithm, new operations, including mutation, crossover and shift, have been developed and incorporated in an improved PSO algorithm. Meanwhile, considering the characteristics of the algorithm, the initial values of the particles have been well planned. Some modification details are depicted below.

#### (1) New operators in the algorithm

- **Mutation.** In this strategy, an operation is first randomly selected in a particle. From its candidate machining resources (machines, tools, setups), an alternative set (machine, tool, setup) is then randomly chosen to replace the current machining resource in the operation.
- **Crossover.** Two particles in the swarm are chosen as parent particles for a crossover operation. In the crossover, a cutting point is randomly determined, and each parent particle is separated as left and right parts of the cutting point. The positions and velocities of the left part of Parent 1 and the right part of Parent 2 are reorganized to form Child 1. The positions and velocities of the left part of Parent 2 and the right part of Parent 1 are reorganized to form Child 2.
- **Shift.** This operator is used to exchange the positions and velocities of two operations in a particle so as to change their relative positions in the particle.

#### (2) Escape method

- During the optimization process, if the iteration number of obtaining the same best fitness is more than 10, then the mutation and shift operations are applied to the best particle to try to escape from the local optima.

The workflow of the improved PSO algorithm is shown in Fig. 10.6.

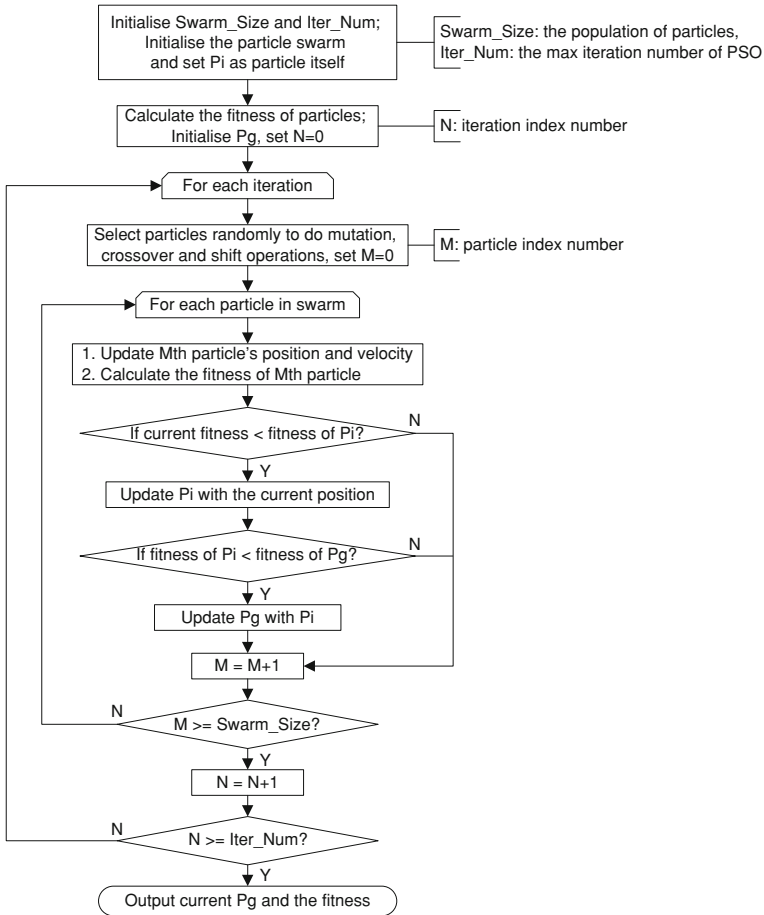


Fig. 10.6 The workflow of the PSO application for IPPS

## 10.6 Implementation and Experimental Studies

### 10.6.1 System Implementation

Microsoft Visual C++ programming language has been used to implement the multi-agent framework developed in this study, and Microsoft Access is used as the database to store information (e.g., resource and knowledge databases). The agents are executed on three hosts. The inter-agent communication is based on the point-to-point method using TCP/IP protocol, and is managed by the Knowledge Query and Manipulation Language (KQML). All messages in this research are compliant with a set of standard performatives of KQML. And, Windows Sockets which is supported by Microsoft Foundation Classes (MFC) is used to implement

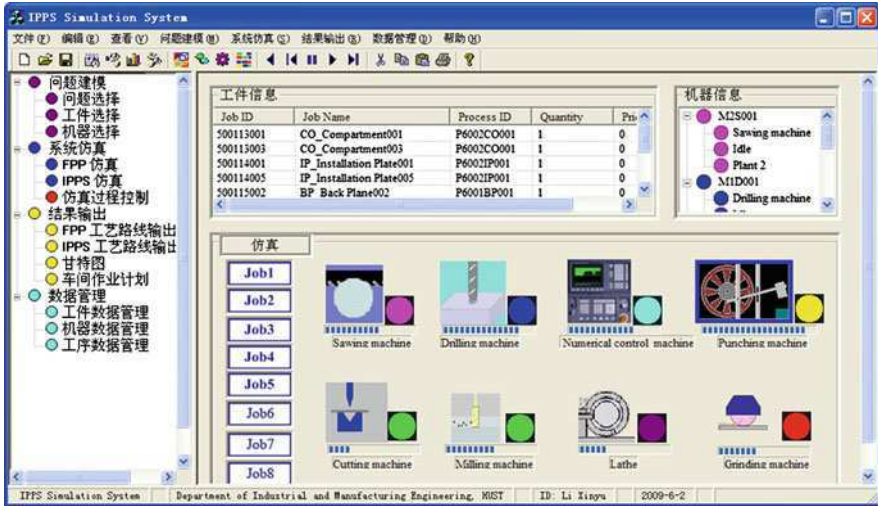


Fig. 10.7 The user interface of the developed multi-agent system

the inter-agent communication. The job agents and machine agents are used to represent jobs and machines. The optimization agent is used to optimize the alternative process plans and scheduling plans. In consideration of the scheduling requirements and availability of manufacturing resources, these agents negotiate with each other to establish the actual process plan of every job and scheduling. The user interface of the developed system is shown in Fig. 10.7.

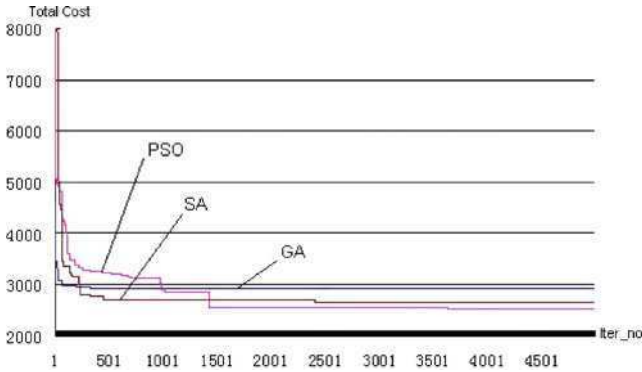
### 10.6.2 Experimental Results and Discussions on the Optimisation Agent

In order to illustrate the effectiveness and performance of the proposed optimization agent, we consider three experimental case studies. The algorithm terminates when the number of generations reaches to the maximum value.

Two experiments are used here to verify the efficiency of the PSO algorithm-based optimization agent for IPPS problems. The first experiment is designed to compare the efficiencies of the PSO, GA and SA algorithms in the application of operation sequencing optimization. The second one is used to further compare them for the IPPS optimization.

#### 10.6.2.1 Experiment 1

Three parts taken from the works of [37, 39] are used here as example parts. The GA and SA algorithms were used to compare their performance with this



**Fig. 10.8** Comparisons of PSO, GA and SA

**Table 10.2** The comparisons of GA, SA and PSO for three parts

Algorithm	Part 1		Part 2		Part 3	
	Best cost achieved	Mean cost of ten trials	Best cost achieved	Mean cost of ten trials	Best cost achieved	Mean cost of ten trials
GA	1381.0	1459.4	1228.0	1340.0	2667.0	2796.0
SA	1421.0	1447.4	1088.0	1122.0	2535.0	2668.5
PSO	1361.0	1430.0	1068.0	1103.0	2535.0	2680.5

developed PSO algorithm. As shown in Fig. 10.8, at the initial optimization stage, the GA optimizes faster than the SA and the PSO (this is shown by a more rapid fall in Fig. 10.8). However, at the middle and late stages, the GA converges while the SA and the PSO continue to decline to give better results. From Table 10.2, it can be observed that the SA and PSO algorithms outperform the GA in all the experiments of all three parts and both the SA and PSO can achieve results that are nearer the optimum.

**10.6.2.2 Experiment 2**

Two criteria are used here as the optimizing direction for IPPS problem, i.e., the make-span and the balanced machine utilization. The example parts and manufacturing resources from [39] are used here to verify the efficiencies of the PSO. Eight parts have been used to test the algorithm under more complex conditions. It can be found that the PSO can optimize the make-span after nearly 4,000 iterations and the balanced machine utilization after 3,000 iterations.

*Make-span.* As shown in Fig. 10.9 and Table 10.3, with the same time period, the PSO and the SA can achieve better results than the GA. But for 20 random consecutive trials, the SA can only proceed with optimization successfully in six trials, the PSO and the GA can proceed with optimization successfully in all 20 trials. Figure 10.10 shows the Gantt chart generated by the optimization agent.



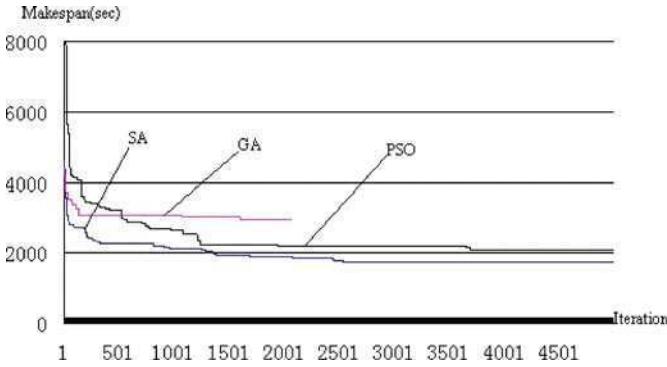


Fig. 10.9 Comparisons of PSO, GA and SA of make-span

Table 10.3 The comparisons of GA, SA and PSO of make-span

Algorithm	Time for 5,000 iterations	Robustness (successful optimization trials out of 20 trials)
GA	16 min 45 s	20
SA	45 min	6
PSO	7 min	20

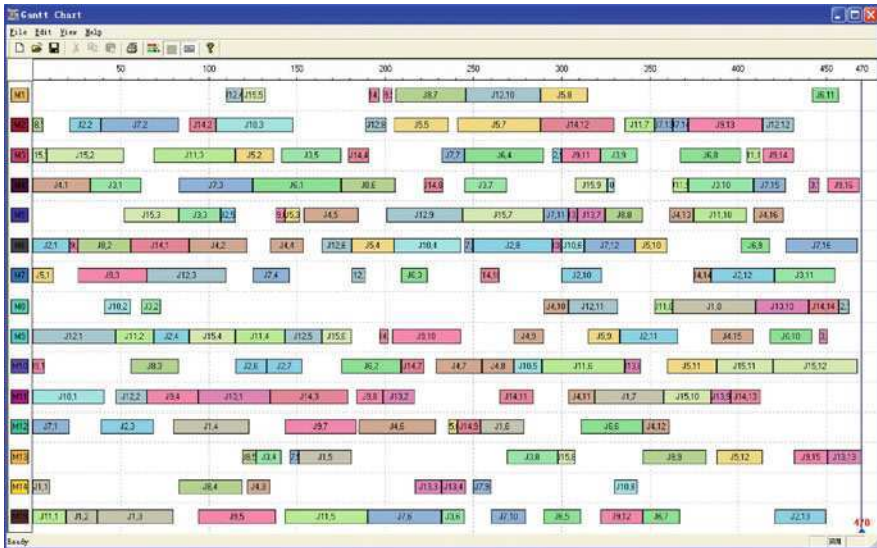


Fig. 10.10 The Gantt chart generated by the optimization agent

*Balanced machine utilization:* From Fig. 10.11 and Table 10.4, it can be observed that all of the algorithms can reach good results, while different characteristics are shown due to the inherent mechanisms of the algorithms. The SA is

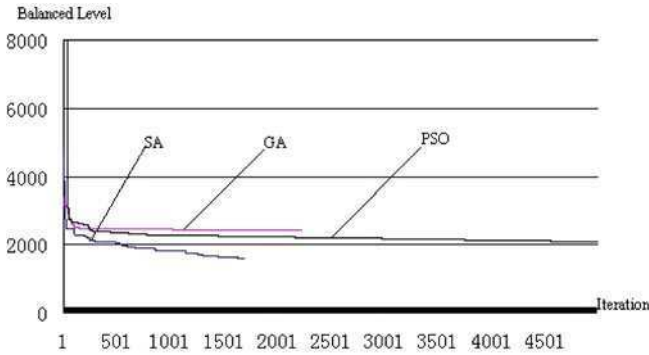


Fig. 10.11 Comparisons of PSO, GA and SA of balanced machine utilization

Table 10.4 The comparisons of GA, SA and PSO of balanced machine utilization

Algorithm	Time for 5,000 iterations	Robustness (successful optimization trials out of 20 trials)
GA	16 min 45 s	20
SA	22 min	6
PSO	7 min 30 s	20

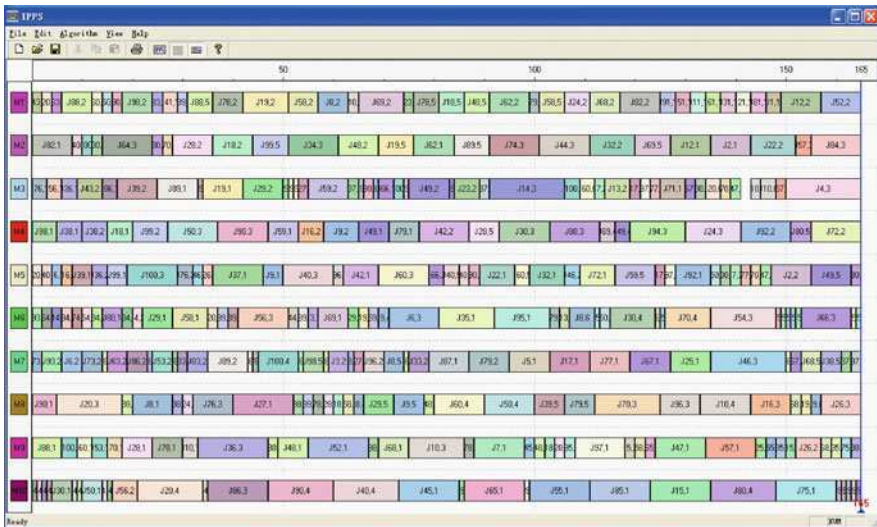


Fig. 10.12 The Gantt chart generated by the optimization agent

much ‘sharper’ to find optimized solutions than the GA and the PSO. The SA can achieve better results than the GA and the PSO. However, in 20 trials, the SA can only proceed with optimization successfully in six trials but the GA and the PSO

can proceed with optimization successfully in all 20 trials. Figure 10.12 shows the Gantt chart generated by the optimization agent.

## 10.7 Conclusions

Considering the complementary roles of process planning and scheduling, the research has been conducted to develop an agent-based approach and optimization agent to facilitate the integration and optimization of these two systems. Process planning and scheduling functions are carried out simultaneously. An optimization agent based on an evolutionary algorithm (PSO) has been developed to optimize and realize the proper decisions resulting from interactions between the agents. To verify the advantage of the optimization algorithm, experimental studies have been carried out to compare this approach with other previously developed approaches. The experimental results show that the proposed approach is very effective for the IPPS problem and achieves better overall optimization results.

With the new method developed in this work, it would be possible to increase the efficiency of manufacturing systems. One future work is to use the proposed method for practical manufacturing systems. The increased use of this approach will most likely enhance the performances of future manufacturing systems.

**Acknowledgments** The research work has been supported by collaborative grants from Coventry University, University of Skövde, the State Key Laboratory of Digital Manufacturing Equipment and Technology of the Huazhong University of Science and Technology China, and the Natural Science Foundation of China (NSFC) under Grant no. 51005088.

## References

1. Sugimura, N., Hino, R., & Moriwaki, T. (2001). Integrated process planning and scheduling in holonic manufacturing systems. In *Proceedings of IEEE international symposium on assembly and task planning, Soft Research Park* (pp. 250–254).
2. Kumar, M., & Rajotia, S. (2003). Integration of scheduling with computer aided process planning. *Journal of Materials Processing Technology*, *138*, 297–300.
3. Saygin, C., & Kilic, S. E. (1999). Integrating flexible process plans with scheduling in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology*, *15*, 268–280.
4. Usher, J. M., & Fernandes, K. J. (1996). Dynamic process planning—the static phase. *Journal of Materials Processing Technology*, *61*, 53–58.
5. Lee, H., & Kim, S. S. (2001). Integration of process planning and scheduling using simulation based genetic algorithms. *International Journal of Advanced Manufacturing Technology*, *18*, 586–590.
6. Tan, W., & Khoshnevis, B. (2000). Integration of process planning and scheduling—a review. *Journal of Intelligent Manufacturing*, *11*, 51–63.
7. Chryssolouris, G., & Chan, S. (1985). An integrated approach to process planning and scheduling. *Annals of the CIRP*, *34*(1), 413–417.

8. Beckendorff, U., Kreutzfeldt, J., & Ullmann, W. (1991). Reactive workshop scheduling based on alternative routings. In *Proceedings of a conference on factory automation and information management* (pp. 875–885).
9. Khoshnevis, B., & Chen, Q. M. (1989). Integration of process planning and scheduling function. In *Proceedings of IIE integrated systems conference and society for integrated manufacturing conference* (pp. 415–420).
10. Larsen, N. E. (1993). Methods for integration of process planning and production planning. *International Journal of Computer Integrated Manufacturing*, 6(1–2), 152–162.
11. Zhang, Y. F., Saravanan, A. N., & Fuh, J. Y. H. (2003). Integration of process planning and scheduling by exploring the flexibility of process planning. *International Journal of Production Research*, 41(3), 611–628.
12. Tonshoff, H. K., Beckendorff, U., & Andres, N. (1989). FLEXPLAN: A concept for intelligent process planning and scheduling. In *Proceedings of the CIRP international workshop* (pp. 319–322).
13. Sormaz, D., & Khoshnevis, B. (2003). Generation of alternative process plans in integrated manufacturing systems. *Journal of Intelligent Manufacturing*, 14, 509–526.
14. Kim, Y. K., Park, K., & Ko, J. (2003). A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Computers & Operations Research*, 30, 1151–1171.
15. Yan, H. S., Xia, Q. F., Zhu, M. R., Liu, X. L., & Guo, Z. M. (2003). Integrated production planning and scheduling on automobile assembly lines. *IIE Transactions*, 35, 711–725.
16. Zhang, X. D., & Yan, H. S. (2005). Integrated optimization of production planning and scheduling for a kind of job-shop. *International Journal of Advanced Manufacturing Technology*, 26, 876–886.
17. Zhang, H. C. (1993). IPPM—a prototype to integrated process planning and job shop scheduling functions. *Annals of the CIRP*, 42(1), 513–517.
18. Zhang, W. J., & Xie, S. Q. (2007). Agent technology for collaborative process planning: A review. *International Journal of Advanced Manufacturing Technology*, 32, 315–325.
19. Wang, L., Shen, W., & Hao, Q. (2006). An overview of distributed process planning and its integration with scheduling. *International Journal of Computer Applications in Technology*, 26(1–2), 3–14.
20. Shen, W., Wang, L., & Hao, Q. (2006). Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey. *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, 36(4), 563–577.
21. Gu, P., Balasubramanian, S., & Norrie, D. (1997). Bidding-based process planning and scheduling in a multi-agent system. *Computers & Industrial Engineering*, 32(2), 477–496.
22. Chan, F. T. S., Zhang, J., & Li, P. (2001). Modelling of integrated, distributed and cooperative process planning system using an agent-based approach. *Proceedings of Institution of Mechanical Engineering, Part B: Journal of Engineering Manufacturing*, 215, 1437–1451.
23. Wu, S. H., Fuh, J. Y. H., & Nee, A. Y. C. (2002). Concurrent process planning and scheduling in distributed virtual manufacturing. *IIE Transactions*, 34, 77–89.
24. Lim, M. K., & Zhang, Z. (2003). A multi-agent-based manufacturing control strategy for responsive manufacturing. *Journal of Materials Processing Technology*, 139, 379–384.
25. Wang, L., & Shen, W. (2003). DPP: An agent-based approach for distributed process planning. *Journal of Intelligent Manufacturing*, 14, 429–439.
26. Wong, T. N., Leung, C. W., Mak, K. L., & Fung, R. Y. K. (2006). Integrated process planning and scheduling/rescheduling—an agent-based approach. *International Journal of Production Research*, 44(18–19), 3627–3655.
27. Wong, T. N., Leung, C. W., Mak, K. L., & Fung, R. Y. K. (2006). Dynamic shopfloor scheduling in multi-agent manufacturing system. *Expert Systems with Applications*, 31, 486–494.
28. Shukla, S. K., Tiwari, M. K., & Son, Y. J. (2008). Bidding-based multi-agent system for integrated process planning and scheduling: A data-mining and hybrid Tabu-SA algorithm-oriented approach. *International Journal of Advanced Manufacturing Technology*, 38, 163–175.

29. Fuji, N., Inoue, R., & Ueda, K. (2008). Integration of process planning and scheduling using multi-agent learning. In *Proceedings of 41st CIRP conference on manufacturing systems* (pp. 297–300).
30. Nejad, H. T. N., Sugimura, N., Iwamura, K., & Tanimizu, Y. (2008). Agent-based dynamic process planning and scheduling in flexible manufacturing system. In *Proceedings of 41st CIRP conference on manufacturing systems* (pp. 269–274).
31. Bhaskara Reddy, S. V., Shunmugam, M. S., & Narendran, T. T. (1999). Operation sequencing in CAPP using genetic algorithms. *International Journal of Production Research*, 37(5), 1063–1074.
32. Qiao, L., Wang, X. Y., & Wang, S. C. (2000). A GA-based approach to machining operation sequencing for prismatic parts. *International Journal of Production Research*, 38(14), 3283–3303.
33. Yip-Hoi, D., & Dutta, D. (1996). A genetic algorithm application for sequencing operations in process planning for parallel machining. *IIE Transactions*, 28, 55–68.
34. Zhang, F., Zhang, Y. F., & Nee, A. Y. C. (1997). Using genetic algorithms in process planning for job shop machining. *IEEE Transactions on Evolutionary Computation*, 1, 278–289.
35. Ding, L., Yue, Y., Ahmet, K., Jackson, M., & Parkin, R. (2005). Global optimization of a feature-based process sequence using GA and ANN techniques. *International Journal of Production Research*, 43(15), 3247–3272.
36. Morad, N., & Zalzal, A. (1999). Genetic algorithms in integrated process planning and scheduling. *Journal of Intelligent Manufacturing*, 10, 169–179.
37. Ma, G. H., Zhang, Y. F., & Nee, A. Y. C. (2000). A simulated annealing-based optimization for process planning. *International Journal of Production Research*, 38(12), 2671–2687.
38. Lee, D. H., Kiritsis, D., & Xirouchakis, P. (2001). Search heuristics for operation sequencing in process planning. *International Journal of Production Research*, 39, 3771–3788.
39. Li, W. D., & McMahon, C. A. (2007). A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 20(1), 80–95.
40. Li, W. D., Ong, S. K., & Nee, A. Y. C. (2004). Optimization of process plans using a constraint-based tabu search approach. *International Journal of Production Research*, 42(10), 1955–1985.
41. Li, W. D., Gao, L., Li, X. Y., & Guo, Y. (2008). Game theory-based cooperation of process planning and scheduling. In *Proceedings of CSCWD* (pp. 841–845).
42. Guo, Y. W., Mileham, A. R., Owen, G. W., & Li, W. D. (2006). Operation sequencing optimization using a particle swarm optimization approach. *Proceedings of the Institution of Mechanical Engineers, Journal of Engineering Manufacture, Part B*, 220(B12), 1945–1958.
43. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. IV, pp. 1942–1948).
44. Li, W. D., Ong, S. K., & Nee, A. Y. C. (2002). Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts. *International Journal of Production Research*, 40(8), 1899–1922.

# Chapter 11

## Distributed Real-Time Scheduling by Using Multi-agent Reinforcement Learning

Koji Iwamura and Nobuhiro Sugimura

**Abstract** Autonomous Distributed Manufacturing Systems (ADMS) have been proposed to realize flexible control structures of manufacturing systems. In the previous researches, a real-time scheduling method based on utility values has been proposed and applied to the ADMS. Multi-agent reinforcement learning is newly proposed and implemented to the job agents and resource agents, in order to improve their coordination processes. The status, the action and the reward are defined for the individual job agents and the resource agents to evaluate the suitable utility values based on the status of the ADMS. Some case studies of the real-time scheduling have been carried out to verify the effectiveness of the proposed methods.

### 11.1 Introduction

Recently, automation of manufacturing systems in batch productions has been much developed aimed at realising flexible small-volume batch productions. The control structures of the manufacturing systems developed, such as flexible manufacturing system (FMS) and flexible manufacturing cell (FMC) are generally hierarchical. The hierarchical control structure is suitable for economical and efficient batch productions in steady state, but not adaptable to very small batch productions with dynamic changes in the volumes and the varieties of the products.

---

K. Iwamura (✉) · N. Sugimura  
Graduate School of Engineering, Osaka Prefecture University, Osaka 599-8531, Japan  
e-mail: iwamura@me.osakafu-u.ac.jp

N. Sugimura  
e-mail: sugimura@me.osakafu-u.ac.jp

Computer systems and manufacturing cell controllers have recently made much progress, and individual computers and controllers are now able to share the decision-making capabilities in the manufacturing systems. The network architectures are widely utilized for the information exchange in the design and the manufacturing.

New distributed architectures of manufacturing systems are therefore proposed to realize more flexible control structures of the manufacturing systems, which are adaptable to the dynamic changes in the volume and the variety of the products and also the unforeseen disruptions, such as malfunction of manufacturing equipment and interruption by high-priority jobs. They are so called as Autonomous Distributed Manufacturing Systems (ADMS) [1, 2], Biological Manufacturing Systems (BMS) [3, 4], and Holonic Manufacturing Systems (HMS) [5–8].

Distributed scheduling methods were proposed and applied to the real-time production scheduling problems of the ADMS, in the previous research [6]. The proposed method was adaptable to dynamic changes and unforeseen disruptions, and it was suitable for the improvement of the objective functions of the whole ADMS such as total make-span. However, there were still remaining scheduling problems from the viewpoint for the improvement of the objective functions of the individual components of the ADMS. Therefore, a real-time scheduling method based on the utility values have been proposed and applied to the ADMS, in order to improve the objective function values of the individual components of the ADMS [7].

Multi-agent reinforcement learning is newly proposed and implemented to the job agents and resource agents, in order to improve their coordination processes. In the reinforcement learning method [9], an agent must be able to sense the status of the environment to some extent and must be able to take actions that affect the status. The agent must also have a goal or goals relating to the status of the environment. The status, the action and the reward are defined for the individual job agents and the resource agents to evaluate the suitable utility values based on the status of the ADMS.

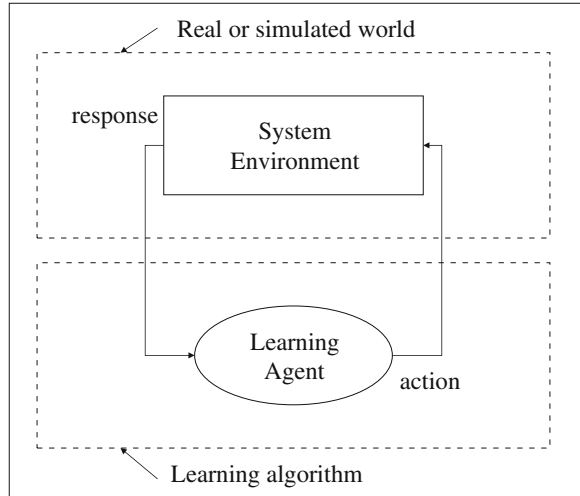
## 11.2 Reinforcement Learning and Its Application

### 11.2.1 Reinforcement Learning

Reinforcement learning is a way of teaching agents (decision-makers) about near-optimal control policies [10]. This is accomplished by assigning rewards and punishments for their actions based on the temporal feedback obtained during active interactions of the learning agents with dynamic systems. The agent should choose actions that tend to improve the measure of system performance [11]. Such an incremental learning procedure suitable for prediction and control problems was developed by Sutton [12].

Figure 11.1 shows a typical single-agent learning model containing four elements, which are the environment, learning agent, a set of actions and environmental

**Fig. 11.1** Typical reinforcement learning scheme



response. The learning agent selects an action for the system, which leads the system evolution along a unique path till the system encounters another decision-making state. At that time, the system consults with the learning agent for the next action. After a state-transition, the learning agent gathers sensory inputs representing the status from the environment, immediate reward and the time spent during the most recent state-transition. Using the information and the algorithm, the agent updates its knowledge base and selects the next action. This completes one step in the iteration process. As this process repeats, the learning agent continues to improve its performance. A simulation model of the system provides the environment component of the model.

### ***11.2.2 Reinforcement Learning Applications to Scheduling Problems***

Reinforcement learning has received some attention in recent years because it deals with the problem of how an autonomous agent learns to select proper actions for achieving its goals through interacting with its environment [13]. Following researches are typical examples of the applications of reinforcement learning to the scheduling problems.

A scheduling method using reinforcement learning was proposed for a semiconductor manufacturing system [14]. A scheduling agent with a classifier system (a reinforcement algorithm) was employed to generate product dispatching rules that put materials into the production floor according to the state of the production floor. Computer simulations were executed using data extracted from an actual semiconductor fabrication. Comparison results of the proposed reinforcement learning-based scheduling method with current scheduling algorithms, a uniform



dispatching rule (UNIF) and a constant work-in-process number dispatching rule (CONWIP) showed that the proposed method obtained similar performance to the current UNIF and CONWIP scheduling methods in terms of turnaround time, and work-in-process number.

An intelligent agent-based dynamic scheduling system was presented by using  $Q$ -III which had been developed based on the  $Q$ -learning (a reinforcement algorithm) [15]. The system is composed of the agent and the simulated environment (SE). The agent is able to perform dynamic scheduling based on the available information provided by the SE. It makes decision for selection of the most appropriate dispatching rule in real-time. It was trained by  $Q$ -III learning algorithm. The authors compared the proposed scheduling system trained by their reinforcement learning mechanism to the three dispatching rules: SPT, COVERT and CR. Their results showed that the proposed scheduling system outperformed the use of each of the three rules individually in mean tardiness on most of the testing cases.

A multi-agent approach was proposed for the dynamic scheduling of maintenance tasks of a petroleum industry production system [16]. Agents simultaneously carried out the effective maintenance scheduling and the continuous improvement of the solution quality by means of reinforcement learning, using the SARSA algorithm. The results of experiment showed that proposed approach can generate on-line scheduling solutions for predictive and corrective maintenance tasks on-line and improve their quality by minimising the variation of the outflow from the tanks.

A simulation optimization methodology using the reinforcement learning approach was proposed for the problem of scheduling of a single server on multiple products, in order to find a dynamic control policy [10]. The dynamic (state dependent) policy optimized a cost function based on the work-in-process inventory, the backorder penalty costs and the setup costs, while meeting the productivity constraints for the products. The methodology was tested on a stochastic lot scheduling problem. The dynamic policies obtained through the reinforcement learning-based approach outperformed various cyclic policies. The reinforcement learning approach was implemented via a multi-agent control architecture where a decision agent was assigned to each of the products. The multi-agent reinforcement learning scheme was able to reduce the base-stock levels in the system while keeping the product demand backorders at low levels.

## 11.3 Real-Time Scheduling Method for ADMS

### 11.3.1 Rule-Based Real-Time Scheduling Process

One of the important objectives of the ADMS is to provide the system components with the flexible and robust capability against the unforeseen disturbances of the manufacturing systems, such as failure of machining equipment and interruptions by high-priority jobs. A real-time production scheduling system has therefore been proposed for control of the components of the ADMS [6]. The real-time scheduling

means that the production schedules of the workpieces and the machining equipment are determined dynamically only when the status of the manufacturing system and its components are changed due to some events occurred in the manufacturing system. Therefore, the scheduling system only determines the schedules of the workpieces and machining equipment in the next time period. The time period means the period between the time when one event occurs and that when another successive event occurs.

The scheduling system consists of a set of ADMS components named job agents and resource agents, which represent the information processing part of the workpieces to be manufactured and that of the machining equipment, respectively. A distributed real-time scheduling method has been proposed, in the previous paper [6], to determine suitable production schedules dynamically, based on the decision-makings of the individual agents. The procedure to determine the schedule is summarized in the following.

The individual agents in the ADMS firstly modify their status, if one of the following events occurs. The status of the resource agents and the job agents are represented by ‘operating’ or ‘idling’.

1. A machining operation of a job is finished.
2. A new job is input to the ADMS.
3. A resource is broken down, or is recovered, and
4. A status of a job is changed from normal one to high-priority one.

In the second step, all the job agents which are ‘idling’ at that time select suitable resource agents, which are ‘idle’ and can carry out their machining operations in the next time period. Some collisions may occur among the selections of the job agents. For example, more than one job agent selects the same resource agent for their next machining operations, as shown in Fig. 11.2a. If a resource agent is selected by more than one job agent, the resource agent selects a most suitable job agent, as shown in Fig. 11.2b, in order to avoid the collisions, in the third step. The job agents and the resource agents select most suitable ones by applying their own decision rules.

### ***11.3.2 Real-Time Scheduling Process Based on Utility Values***

The rule-based scheduling process was adaptable to the dynamic changes and the unforeseen disruptions, and it was suitable for the improvement of the objective functions of the whole ADMS such as total make-span. However, there were scheduling problems still remaining from the viewpoint of improving the objective functions of the individual components of the ADMS.

Therefore, a real-time scheduling method based on the utility values have been proposed and applied to the ADMS, in order to improve the objective function values of the individual components of the ADMS [7]. The agents in the ADMS are divided into three classes based on their roles in the scheduling processes.

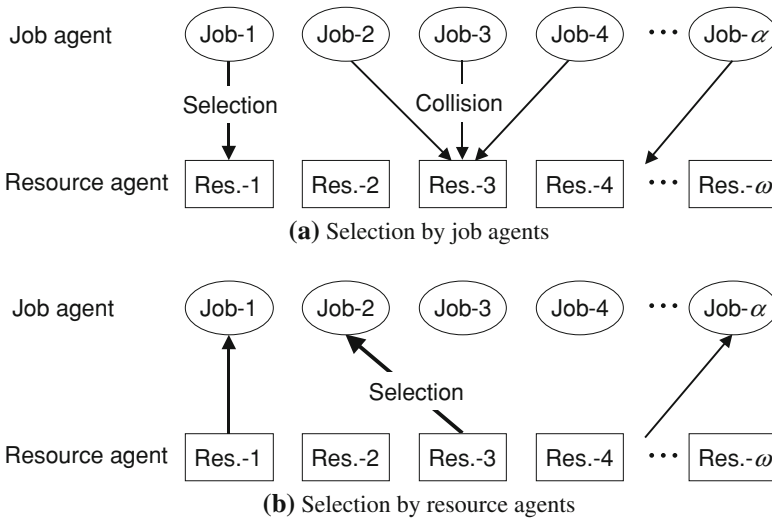


Fig. 11.2 Rule-based real-time scheduling process

- *Resource agents.* They are the information processing parts of the resources. The resources transform the jobs in the manufacturing process. In the scheduling process, the resource agents evaluate the utility values for the candidate job agents which are processed by the resources in the next time period.
- *Job agents.* They are the information processing parts of the jobs. The jobs are transformed by the resources from the blank materials to the final products in the manufacturing process. In the scheduling process, the job agents evaluate the utility values for the candidate resource agents which carry out the machining operations in the next time period.
- *Coordination agents.* It selects a most suitable combination of the resource agents and the job agents for the machining operations in the next time period, based on the utility values sent from the resource agents and the job agents.

It is assumed here that the individual job agents have the following technological information.

$M_{ik}$   $k$ th machining operation of the job  $i$ . ( $i = 1, \dots, \alpha$ ), ( $k = 1, \dots, \beta$ )

$AC_{ik}$  Required machining accuracy of machining operation  $M_{ik}$ . It is assumed that the machining accuracy is represented by the levels of accuracy indicated by 1–3, which means rough, medium high and high accuracy, individually.

$R_{ikm}$   $m$ th candidate of resource, which can carry out the machining operation  $M_{ik}$ . ( $m = 1, \dots, \gamma$ ).

$W_i$  Waiting time until the job  $i$  becomes idle if it is under machining status.

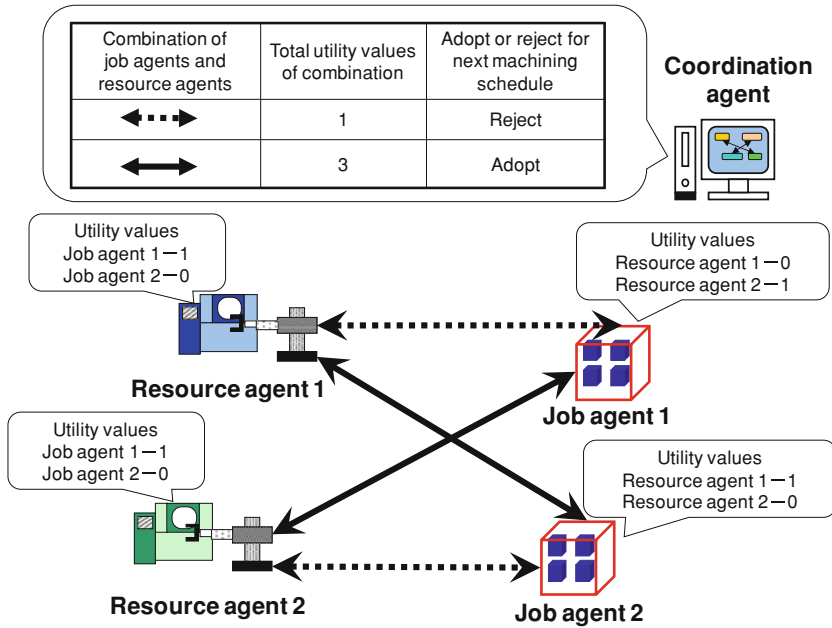


Fig. 11.3 Real-time scheduling method based on utility values

The individual resource agents have the following technological information.

- $T_{ikm}$  Machining time in the case where the resource  $R_{ikm}$  carries out the machining operation  $M_{ik}$ .
- $MAC_{ikm}$  Machining accuracy in the case where the resource  $R_{ikm}$  carries out the machining operation  $M_{ik}$ .  $MAC_{ikm}$  is also represented by the levels of 1, 2 and 3.
- $MCO_{ikm}$  Machining cost in the case where the resource  $R_{ikm}$  carries out the machining operation  $M_{ik}$ .
- $W_{ikm}$  Waiting time until resource  $R_{ikm}$  becomes idle if it is under machining status.

At the time  $t$ , all the ‘idling’ agents have to select their machining schedules in the next time period, as shown in Fig. 11.3. The following procedure is proposed for the individual agents to select their machining schedules.

1. *Retrieval of status data.* The individual ‘idling’ agents firstly get the status data from the other agents which are ‘operating’ or ‘idling’. The ‘idling’ resources and jobs can start the machining operation in the next time period.
2. *Selection of candidate agents.* The individual ‘idling’ agents select all the candidate agents for the machining operations in the next time period. For instances, the job agent  $i$  selects the resource agents which can carry out the next machining operation  $M_{ik}$ . On the other hand, the resource agent  $j$  selects all the candidate job agents which can be machined by the resource agent  $j$ .

**Table 11.1** Objective functions of agents

Objective Functions	Objective Function Values
Efficiency of resource agent	$\Sigma$ Machining time/total time
Machining accuracy of resource agent	$\Sigma$ (Machining accuracy of resources - required machining accuracy of jobs)
Flow-time of job agent	$\Sigma$ (Machining time + waiting time)
Machining cost of job agent	$\Sigma$ (Machining cost of resources)

3. *Determination of utility values.* The individual ‘idling’ agents determine the utility values for the individual candidates selected in the second step. For instance, the job agent determines the utility values, based on its own decision criteria for all the candidate resource agents which can carry out the next machining operation.

4. *Coordination.* All the ‘idling’ agents send the selected candidates and the utility values of the candidates to the coordination agent. The coordination agent determines a suitable combination of the job agents and the resource agents which carry out the machining operations in the next time period, based on the utility values. The decision criterion of the coordination agent is to maximize the total sum of the utility values of all the agents.

### 11.3.3 Evaluation of Utility Values

The utility values are evaluated based on the decision criteria of the individual agents, and various decision criteria are considered for the agents. Therefore, it is assumed that the individual agents have one of the objective functions shown in Table 11.1 for evaluating the utility values.

The following procedures are provided for the resource agents to evaluate the utility values. Let us consider a resource agent  $j$  at a time  $t$ . It is assumed that  $TT_{j,t}$ ,  $ME_{j,t}$  and  $MA_{j,t}$  show the total time after the resource  $j$  starts its operations, the efficiency, and the evaluated value of machining accuracy of the resource  $j$ , respectively. If the resource agent  $j$  selects a candidate job agent  $i$  for carrying out the machining operation  $M_{ik}$ , the efficiency and the evaluated value of the machining accuracy are estimated by the following equations.

$$ME_{j,t+1}(i) = (ME_{j,t} \cdot TT_{j,t} + T_{ikj}) / (TT_{j,t} + T_{ikj} + W_i) \tag{11.1}$$

$$MA_{j,t+1}(i) = MA_{j,t} + (MAC_{ikj} - AC_{ik}) \tag{11.2}$$

where the resource  $j$  can carry out the machining operation  $M_{ik}$  of job  $i$  ( $j = R_{ikm}$ ).

As regards the job agents, the following equations are applied to evaluate the flow-time and the machining costs, for the case where a job agent  $i$  selects a candidate resource agent  $j$  ( $=R_{ikm}$ ) for carrying out the machining operation  $M_{ik}$ . It is assumed that  $JT_{i,t}$  and  $JC_{i,t}$  give the total time after the job  $i$  is input to the ADMS and the machining cost, respectively.

$$JT_{i,t+1}(j) = JT_{i,t} + T_{ikj} + W_{ikj} \quad (11.3)$$

$$JC_{i,t+1}(j) = JC_{i,t} + MCO_{ikj} \quad (11.4)$$

The objective functions mentioned above have different units. Some of them shall be maximized and others minimized. Therefore, the utility values are normalized from 0 to 1, by applying the following equations.

1. Efficiency of resource agents:

$$RUV_j(i) = 1 - \frac{\max_{i=1,\dots,\tau} \{ME_{j,t+1}(i)\} - ME_{j,t+1}(i)}{\max_{i=1,\dots,\tau} \{ME_{j,t+1}(i)\} - \min_{i=1,\dots,\tau} \{ME_{j,t+1}(i)\}} \quad (11.5)$$

2. Machining accuracy of resource agents:

$$RUV_j(i) = \frac{\max_{i=1,\dots,\tau} \{MA_{j,t+1}(i)\} - MA_{j,t+1}(i)}{\max_{i=1,\dots,\tau} \{MA_{j,t+1}(i)\} - \min_{i=1,\dots,\tau} \{MA_{j,t+1}(i)\}} \quad (11.6)$$

3. Flow-time of job agents:

$$JUV_i(j) = \frac{\max_{j=1,\dots,\gamma} \{JT_{i,t+1}(j)\} - JT_{i,t+1}(j)}{\max_{j=1,\dots,\gamma} \{JT_{i,t+1}(j)\} - \min_{j=1,\dots,\gamma} \{JT_{i,t+1}(j)\}} \quad (11.7)$$

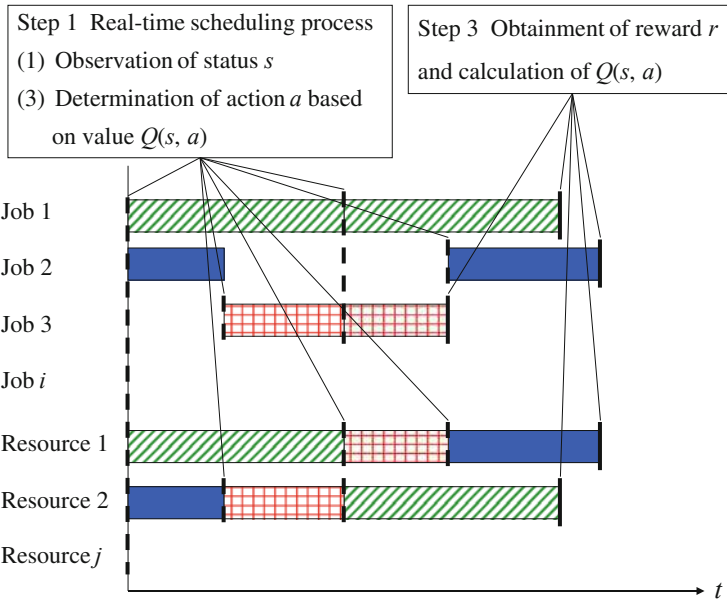
4. Machining cost of job agents:

$$JUV_i(j) = \frac{\max_{j=1,\dots,\gamma} \{JC_{i,t+1}(j)\} - JC_{i,t+1}(j)}{\max_{j=1,\dots,\gamma} \{JC_{i,t+1}(j)\} - \min_{j=1,\dots,\gamma} \{JC_{i,t+1}(j)\}} \quad (11.8)$$

where  $\max\{f(x)\}$  and  $\min\{f(x)\}$  give the maximum value and the minimum value of  $f(x)$  evaluated for all the candidates.  $\tau$  and  $\gamma$  give the number of the candidate job agents for the resource agent  $j$ , and the number of the candidate resource agents for the job agent  $i$ , respectively.

One of the important problems to be solved in the proposed system is lack of improvement capabilities of the individual agents, which have fixed decision criteria represented by the utility values. Following issues should be considered to modify the utility values by the individual agents from the view point of the property of the objective functions of the individual agents.

- The individual agents with the objective functions of efficiency or flow-time are not to be in 'idling' status, and they have to be assigned to some candidate agents for the next machining operations, in order to improve their objective functions.



**Fig. 11.4** Application of multi-agent reinforcement learning

Therefore, higher utility values are given to the candidate agents, even if the candidates are not suitable from the viewpoint of objective functions.

- The individual agents with the objective functions of machining cost or machining accuracy are to be in ‘idling’ status for the cases where they do not have any suitable candidate agents, in order to improve their objective functions. Therefore, they give lower utility values to the candidate agents not suitable from the viewpoints of the objective functions, and wait until some suitable candidate agents are found.

It is not easy to establish flexible decision criteria adaptable to the situation mentioned above. Therefore, a reinforcement learning mechanism is proposed to modify the utility values of the candidate agents based on the status of the manufacturing systems.

## 11.4 Application of Multi-agent Reinforcement Learning

A multi-agent reinforcement learning is newly proposed and implemented to the job agents and resource agents, in the present research, in order to improve their coordination processes.

Figure 11.4 summarizes the multi-agent reinforcement learning procedure proposed here. The individual job agents and resource agents carry out the following

four steps to obtain their suitable decision criteria for evaluation of the utility values by applying the multi-agent reinforcement learning.

*Step 1* The individual job agents and resource agents carry out the real-time scheduling process described in Sect. 11.3.2, when their previous machining operations are finished. The real-time scheduling processes (1) and (3) are modified as following for implementation of the multi-agent reinforcement learning.

#### 1. Retrieval of status data

The individual ‘idling’ agents get the status data from the other agents which are ‘operating’ or ‘idling’, and observe the status  $s$  of the manufacturing systems.

#### 2. Determination of utility values

The individual job agents and resource agents execute the action  $a$  based on the value  $Q(s, a)$ , to evaluate the utility values for all the candidate machining operations in the next time period, where,  $s$  and  $a$  represent the status and the actions in the reinforcement learning method, respectively.

*Step 2* The real-time scheduling processes are repeated until all the machining operations of the jobs are finished by the resources in the ADMS.

*Step 3* The individual job agents and resource agents obtain the reward  $r$  based on their own objective function values, and calculate the value  $Q(s, a)$ .

*Step 4* Step 1 to Step 3 are repeated for the new jobs to be manufactured in the manufacturing systems, in order to converge the value  $Q(s, a)$  of the individual job agents and resource agents.

In these steps, the status  $s$ , the action  $a$  and the reward  $r$  are given as follows.

#### 1. Status $s$

The status  $s$  observed by the job agents and the resource agents is represented by the following equation, in the present research.

$$s = (s_1, s_2, s_3, s_4) \quad (11.9)$$

where  $s_p$  ( $p = 1, 2, 3, 4$ ) are the number of ‘idling’ agents that have the objective functions of efficiency, machining accuracy, flow-time and machining cost, respectively. This means that the learning process of the individual agents is carried out based on the numbers and the types of the ‘idling’ agents.

#### 2. Action $a$

The individual job agents and resource agents select the parameter  $n$  ( $= 1/5, 1/3, 1, 3, \text{ or } 5$ ) in the following equation to evaluate the utility values.

$$UV' = (UV)^n \quad (11.10)$$

where  $UV$  is the utility value calculated by the individual job agents and resource agents described in Sect. 11.3.3.  $UV'$  is the modified utility value by applying the



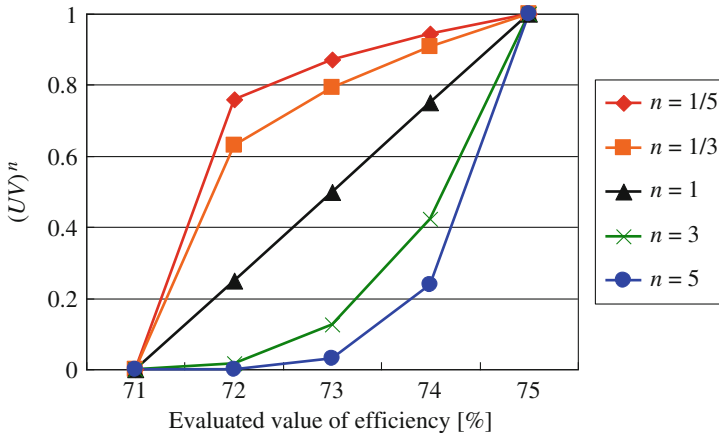


Fig. 11.5 Example of utility value obtained by action  $a$

action  $a$  based on the status  $s$ . Figure 11.5 shows an example of the modified utility values for the cases where the resource agent whose objective function is efficiency evaluate the utility values for the five candidate job agents by selecting the parameter  $n$ . In the figure, the horizontal and vertical axes show the evaluated value of the efficiency for the cases where the resource agent selects the individual candidate job agents, and modified utility values obtained by changing the parameter  $n$ , respectively. As shown in the figure, higher utility values are obtained when  $n$  is  $<1$ , and lower utility values are obtained when  $n$  is  $>1$  in comparison with  $n = 1$ .

$\epsilon$ -greedy policy is applied for the individual job agents and resource agents to determine the action  $a$ .  $\epsilon$ -greedy policy means that most of the time agents choose an action that has maximal estimated action value, but with probability they instead select an action at random [9].

### 3. Reward $r$

The individual job agents and resource agents obtain the reward  $r$  based on their own objective function values. Three different methods are considered to calculate the reward  $r$ .

Type 1 Reward calculated by the objective function values of individual agents.

The individual job agents and resource agents obtain the reward  $r_h$  given by following equations, based on their own objective functions.

- a. For the case where the objective function is efficiency

$$r_h = (a_h - b_h)/b_h \tag{11.11}$$

- b. For the case where the objective function is either machining accuracy, flow-time or machining cost

$$r_h = (b_h - a_h)/b_h \tag{11.12}$$

where  $a_h$  and  $b_h$  are the objective function values obtained by applying the proposed method with the reinforcement learning, and those obtained without the reinforcement learning.

Type 2 Reward calculated by the averaged values of the objective function of all the agents which have same objective function.

The individual job agents and resource agents obtain the reward  $r_p$  given by following equations.

$$r_p = \sum_{h=1}^{\xi} r_h / \xi \quad (11.13)$$

where  $p$  and  $\xi$  are the types of objective functions and the total number of agents with  $p$ th type of objective functions, respectively.  $r_h$  is calculated by Eqs. 11.11 and 11.12 based on the types of objective functions.

Type 3 Reward calculated by the averaged values of objective function of all the agents

The individual job agents and resource agents obtain the reward  $r_q$  given by following equations.

$$r_q = (1/4) \sum_{h=1}^4 r_p \quad (11.14)$$

where  $r_p$  is calculated by Eq. 11.13.

The value  $Q(s, a)$  is determined by applying the Monte Carlo method [9]. The individual job agents and resource agents save the  $n$  rules  $(s_t, a_t)$  ( $t = 0, 1, \dots, n - 1$ ) between the time when they obtain the reward  $r$  and the time when they obtain the new reward  $r$ . The rule  $(s, a)$  means the set of status  $s$  and action  $a$ . The value  $Q(s, a)$  is calculated by the following equations.

$$\text{Sum Reward}(s_t, a_t) \leftarrow \text{Sum Reward}(s_t, a_t) + r \quad (11.15)$$

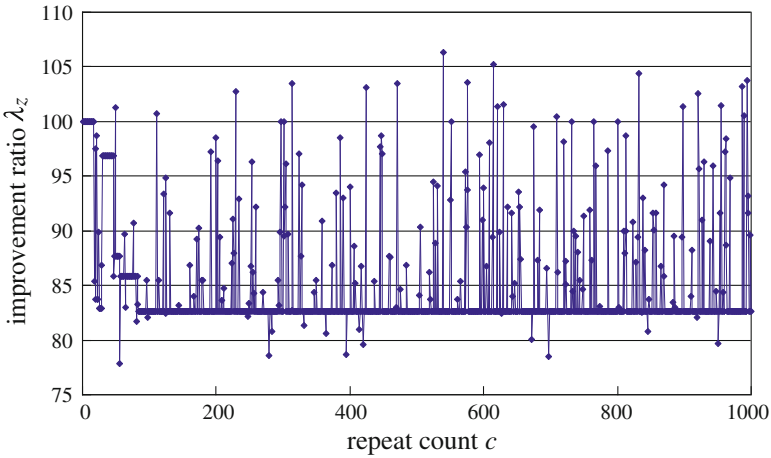
$$Q(s, a) \leftarrow \text{Sum Reward}(s, a) / \text{Reward Count} \quad (11.16)$$

where  $\text{Sum Reward}(s, a)$  is the cumulative rewards in the case where the action  $a$  is applied in the status  $s$ .  $\text{Reward Count}$  is the total number in the case where the rule  $(s, a)$  gets the reward  $r$ .

## 11.5 Case Studies

### 11.5.1 Case Without Unforeseen Event

Some case studies have been carried out to verify the effectiveness of the proposed methods. The ADMS model considered in the case studies has 10 resources. The individual resource agents have the different objective functions and the



**Fig. 11.6** Improvement ratio

different machining capacities, such as the machining time  $T_{ikm}$ , the machining accuracy  $MAC_{ikm}$  and the machining cost  $MCO_{ikm}$ .

As regards to the jobs, three cases are considered in the case study, which have 16, 20 and 30 jobs. The individual job agents have the different objective functions and the machining sequences. It is assumed that the same jobs are input to the ADMS after the resources finish all the manufacturing processes. 12 cases are considered, in the case study, by changing the machining capacities of the resources.

$\varepsilon$  is set to 0.2 for the  $\varepsilon$ -greedy policy.

Figure 11.6 shows the best result for the case where the reward is calculated by using Type3 described in Sect. 11.4. In the figure, the horizontal and vertical axes show the repeat count  $c$  and the improvement ratio  $\lambda_z$ , respectively. The repeat count  $c$  here means the number of repetitions of all the manufacturing processes of the input jobs. The improvement ratio  $\lambda_z$  means the ratio between the objective function values of all the agents obtained by the proposed method and the ones by the conventional method in the case  $z$ .  $\lambda_z$  is calculated by following equation.

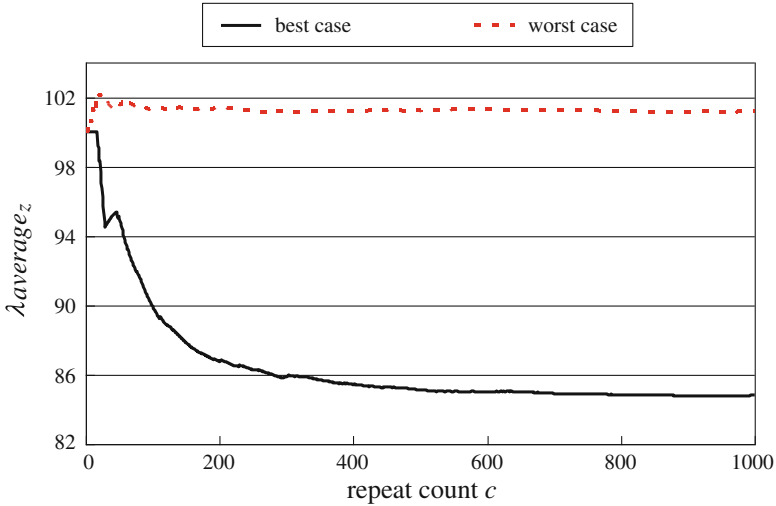
$$\lambda_z = \sum_{h=1}^v \mu_h / v \tag{11.17}$$

where  $\mu_h$  and  $v$  are the improvement ratio of the objective function values of the agent  $h$  and the total number of agents, respectively. The  $\mu_h$  is calculated by the following equation based on the type of the objective functions.

- a. For the case that the objective function is efficiency

$$\mu_h = b_h / a_h \tag{11.18}$$

- b. For the case that the objective function is either machining accuracy, flow-time or machining cost



**Fig. 11.7** Average improvement ratio

$$\mu_h = a_h/b_h \tag{11.19}$$

where  $a_h$  and  $b_h$  are the objective function values of the individual agents  $h$  obtained by the proposed method and the previous conventional method. As shown in the figure, the improvement ratio  $\lambda_z$  is converged until the episode reaches to 100.

Figure 11.7 shows the average improvement ratio  $\lambda_{average_z}$  of the best case and the worst case. Following equation gives the  $\lambda_{average_z}$  which means the average of improvement ratio  $\lambda_z$  until the episode reaches to  $\omega$  in the case  $z$ .

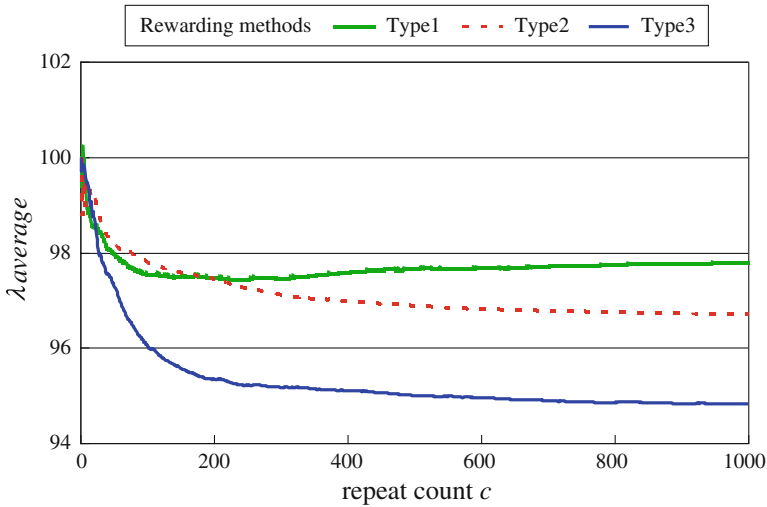
$$\lambda_{average_z} = \sum_{c=1}^{\omega} \lambda_c / \omega \tag{11.20}$$

where  $\lambda_c$  is the improvement ratio  $\lambda_z$  at the repeat count  $c$ .

Figure 11.8 shows the comparison of the cases using Types 1–3 rewarding methods described in Sect. 11.4, from the view point of  $\lambda_{average}$ .  $\lambda_{average}$  the average of  $\lambda_{average_z}$  in the all 12 cases.  $\lambda_{average}$  is calculated by the following equation.

$$\lambda_{average} = \sum_{z=1}^{12} \lambda_{average_z} / 12 \tag{11.21}$$

As shown in Fig. 11.8, all cases are effective to improve the objective function values in comparison with previous method without reinforcement learning. It means that the individual job agents and resource agents obtain the suitable decision criteria for evaluation of utility values. However, as shown in the figure, the value  $Q(s, a)$  does not converge in the case using Type 1 where the reward is



**Fig. 11.8** Comparison of rewarding methods

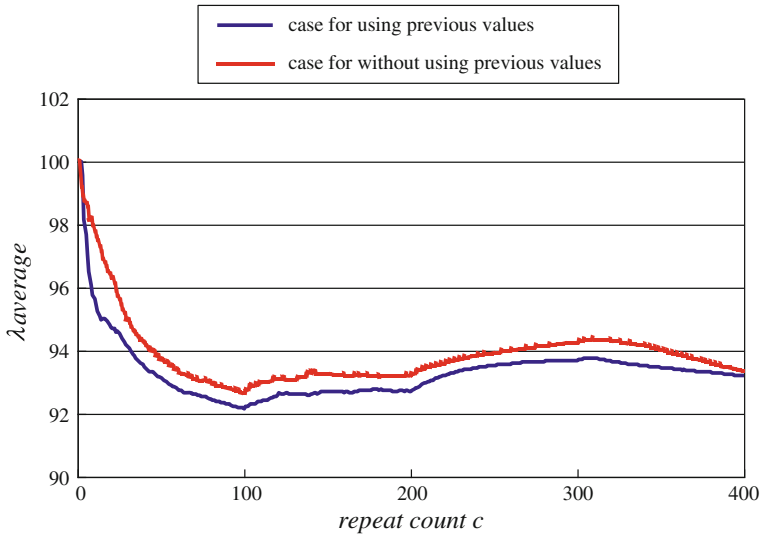
calculated by the objective function values of individual agents. The individual job agents and resource agents most improve their objective function values while using Type 3 where the reward is calculated by the objective function values of all agents.

### 11.5.2 Case with Unforeseen Event

Some case studies have been carried out to verify the effectiveness of the proposed method in the cases with unforeseen events, such as breakdown of resources and changing the number of input jobs. The individual job agents and resource agents calculate the reward by using Type3. Sixteen jobs are manufactured by 10 resources when the manufacturing processes are started in the ADMS. Following events occurred during the repetition of manufacturing processes.

1. The number of input jobs is changed to 20 when the repeat count  $c$  reaches to 100.
2. One resource is broken down when the repeat count  $c$  reaches to 200.
3. The broken resource is recovered when the repeat count  $c$  reaches to 300.

Two cases are considered to compare with and without using the previous value  $Q(s, a)$  when unforeseen events have occurred. It means that, the individual resource agents and job agents continue to use the same value after unforeseen events have occurred, in the first case. And the individual resource agents and job agents reset the value when unforeseen events have occurred, in the second case.



**Fig. 11.9** Comparison of rewarding methods

Figure 11.9 shows the results of comparison between cases with and without using previous value. As shown in the figure, both cases are effective to improve the objective function values, and the difference between the cases with and without using previous value is small.

## 11.6 Conclusions

A multi-agent reinforcement learning approach is applied to the real-time scheduling method based on the utility values for the Autonomous Distributed Manufacturing Systems. The following remarks are concluded.

1. The real-time scheduling process proposed in the previous research is modified, aimed at implementing multi-agent reinforcement learning, in order to obtain the suitable decision criteria for evaluation of utility values.
2. The status, the action and the reward are defined for the individual job agents and the resource agents to evaluate the suitable utility values based on the status of the ADMS.
3. Some case studies of the real-time scheduling have been carried out to verify the effectiveness of the proposed methods in comparison with the previous method. It was shown, through case studies, that the proposed methods are effective to improve the objective function values of the individual agents. The objective function values of individual agents are improved most effectively in the case where the reward is calculated based on the averaged objective function values of all the agents.

## References

1. Moriwaki, & T., Sugimura, N. (1992). Object-oriented modelling of autonomous distributed manufacturing system and its application to real-time scheduling. *Proceedings of the ICOOMS '92* (pp. 207–212).
2. Kadar, B., Monostori, L., & Szelke, E. (1998). An object-oriented framework for developing distributed manufacturing architectures. *Journal of Intelligent Manufacturing*, 9, 173–179.
3. Ueda, K. (1992). An approach to bionic manufacturing systems based on DNA-type information. *Proceedings of the ICOOMS '92*, (pp. 303–308).
4. Ueda, K., Hatono, I., Fujii, N., & Vaario, J. (2000). Reinforcement learning approach to biological manufacturing systems. *Annals of the CIRP*, 49, 343–346.
5. Hendrik, B., Jo, W., Paul, V., Luc, B., & Patrick, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry*, 37, 255–274.
6. Sugimura, N., Tanimizu, Y., & Iwamura, K. (2004). A Study on real-time scheduling for holonic manufacturing system. *CIRP Journal of Manufacturing Systems*, 33(5), 467–475.
7. Iwamura, K., Okubo, N., Tanimizu, Y., & Sugimura, N. (2006). Real-time scheduling for holonic manufacturing systems based on estimation of future status. *International Journal of Production Research*, 44(18–19), 3657–3675.
8. Iwamura, K., Nakano, A., Tanimizu, Y., & Sugimura, N. (2007). A study on real-time scheduling for holonic manufacturing systems -Simulation for estimation of future status by individual holons-. In M. Vladimir, V. Valeriy, & W. C. Armando (Eds.), *LNAI 4659 HoloMAS 2007* (pp. 205–214). Heidelberg: Springer.
9. Sutton, R., & Barto, A. (1998). *Reinforcement learning: an introduction*. Cambridge: The MIT Press.
10. Paternina-Arboleda, C., & Das, T. (2005). A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problem. *Simulation Modelling Practice and Theory*, 13, 389–406.
11. Kaelbling, L., Littman, M., & Moore, A. (1996). Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4, 237–285.
12. Sutton, R. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3, 9–44.
13. Wang, Y., & Usher, J. (2005). Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence*, 18, 73–82.
14. Fujii, N., Takasu, R., Kobayashi, M., Ueda, K. (2005). Reinforcement learning based product dispatching scheduling in a semiconductor manufacturing system. *Proceedings of the 38th CIRP International seminar on manufacturing systems*, CD-ROM
15. Aydin, E., & Oztemel, E. (2000). Dynamic job-shop scheduling using reinforcement learning agents. *Robotics and Autonomous Systems*, 33, 169–178.
16. Aissani, N., Beldjilali, B., & Trentesaux, D. (2009). Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. *Engineering Applications of Artificial Intelligence*, 22, 1089–1103.

# Chapter 12

## A Multiple Ant Colony Optimisation Approach for a Multi-objective Manufacturing Rescheduling Problem

Vikas Kumar, Nishikant Mishra, Felix T. S. Chan, Niraj Kumar and Anoop Verma

**Abstract** Manufacturing scheduling is a well-known complex optimization problem. A flexible manufacturing system on one side eases the manufacturing processes but on the other hand it increases the complexity in the decision making processes. This complexity further enhances when disruption in the manufacturing processes occurs or when arrival of new orders is considered. This requires rescheduling of the whole operation, which is a complex decision making process. Realizing this complexity and taking into account the contradictory objective of making a trade-off between costs and time, this research aims to generate an effective manufacturing schedule. The existing approach of rescheduling

---

V. Kumar (✉)

Department of Management, Dublin City University Business School Dublin,  
Dublin 9, Republic of Ireland  
e-mail: vikas.kumar@dcu.ie

N. Mishra

School of Management and Business, Aberystwyth University, Aberystwyth, UK  
e-mail: nim4@aber.ac.uk

F. T. S. Chan

Department of Industrial and Systems Engineering, The Hong Kong Polytechnic  
University, Hung Hom, Hong Kong, China  
e-mail: f.chan@inet.polyu.edu.hk

N. Kumar

Department of Management, School of Management, University of Bath,  
Bath, BA2 7AY, UK  
e-mail: n.kumar@bath.ac.uk

A. Verma

Computer Aided Manufacturing Laboratory, Department of Mechanical Engineering,  
University of Cincinnati, Cincinnati, USA  
e-mail: vermaap@email.uc.edu



sometimes generates entirely a new plan that requires a lot of changes in the decisions, which is not preferable by manufacturing firms. Therefore, in this research whenever a disruption occurs or a new order arrives, the proposed approach reschedules the remaining manufacturing operations in such a way that minimum changes occur in the original manufacturing plan. Evolutionary optimization methods have been quite successful and widely addressed by researchers to handle such complex multi-objective optimization problems because of their ability to find multiple optimal solutions in one single simulation run. Inspired by this, the present research proposes a multiple ant colony optimization (MACO) algorithm to resolve the computational complexity of a manufacturing rescheduling problem. The performance of the proposed MACO algorithm will be compared with the simple ant colony optimization (ACO) to judge its robustness and efficacy.

## 12.1 Introduction

Last few decades have seen rapid growth in the world economy. This growth is the result of the advancement in modern technologies and escalating demand in both the manufacturing and service sector. The rapid growth in the manufacturing sector has on one hand managed to serve the escalating demand whereas on the other hand, it has increased the complexity of handling different related manufacturing processes. Nowadays, in manufacturing environment, several new orders are introduced frequently in the market and at the same time the customer demand changes at a fast pace. Therefore, manufacturing processes need frequent adjustments to accommodate these changes and avoid any disturbances. Whenever an order arrives, a manufacturing plan is generated for that particular order and appropriate resources are then allocated. Thereafter, the corresponding manufacturing processes are decided and finally manufacturing operations are carried out to deliver the final product. Therefore, a number of decisions are made and processes need to be successfully scheduled and managed. A schedule is generated for every order and again rescheduled when a new order arrives or when any disruption occurs. Thus, rescheduling makes the process more complex, as the number of decisions needs to be re-considered such as the allocation of the raw materials, the tools and fixture requirements for the particular order and allocation of machines to perform different machining operations. Realizing these complexities, several researchers have attempted to resolve them using different optimization evolutionary algorithms (EAs). Researchers have studied various complex scenarios in different industrial context and proposed solutions accordingly under number of constraints.

The manufacturing scheduling problems commonly addressed by researchers, nowadays, mainly deal with the scheduling or planning, and focuses on minimizing make-span, generating reschedule to incorporate new product or to

resolve discrepancies. Although the research meets the aforementioned need, the issues highlighted cause frequent disturbances in the manufacturing processes. These disturbances raise a big question in the real-world application of the solution proposed by the existing research. To address the issue in this research an attempt has been made in the manufacturing plan to accommodate a new order or to address the discrepancies, with the minimum changes in the existing schedule. Trade-off is made between the objective function of minimizing the make-span and the minimizing changes in the existing schedule. This will help to smoothly run manufacturing processes and reduce the cost occurred during the material flow for the manufacturers. The reallocation of the material due to the changes in the plan also causes a chaos in the process, makes manufacturing process more complex, incurs additional cost to the manufacturers and fails to optimally utilize the manufacturing resources. If the manufacturing process is handled manually then this issue becomes more severe whereas if it is an automated process then it increases the workload of the planner. Thus, addressing this issue will also reduce the material flow processes and resolve reallocation costs. Therefore, this study will fill the existing gap of research of efficiently handling a multi-objective rescheduling problem in the manufacturing environment.

The chapter is organized as follows. [Section 12.2](#) reviews the literature in the area of manufacturing rescheduling which is followed by the mathematical formulation of this research in [Sect. 12.3](#). Several decision variables, constraints and objective functions are discussed in [Sects. 12.3.1–12.3.3](#). The detailed explanation of the proposed MACO algorithm is presented in [Sect. 12.4](#). A case study example is elaborated in [Sect. 12.5](#) and results and discussions are provided in [Sect. 12.6](#). Finally, [Sect. 12.7](#) concludes this research and provides some future research directions.

## 12.2 Literature Review

Manufacturing rescheduling is of prime importance for researchers because of the dynamic nature of the manufacturing operations and high probability of the occurrence of the unexpected events. The empirical research findings shows that firms that frequently reschedule perform better. Yamamoto [1] proposed a rescheduling procedure for real-time control of a computerized manufacturing facility managed by a central manufacturing operating system. His study showed that rescheduling is more advantageous compared to fixed sequencing and priority despatching procedures. Wu et al. [2] developed one machine rescheduling heuristics to resolve the unforeseen disruption in manufacturing environment. Their study aimed at the minimization of the make-span and the impact of the schedule change. Abumaizar and Svestka [3] performed a factorial experiment on benchmark scheduling problems to study the effect of different rescheduling methods, various problem characteristics and disruption scenarios on the performance of the new schedules. They proposed an algorithm to resolve the

rescheduling problem of affected operations and the result indicated that the proposed algorithm overcomes the disadvantages associated with other rescheduling methods. Jain and ElMaraghy [4] pointed out the necessity of generation of new and modified production schedules in the complex manufacturing environment. They proposed using genetic algorithm (GA) to revise only those operations that must be rescheduled and can, therefore, be used in conjunction with the existing scheduling methods to improve the efficiency of flexible manufacturing systems. Fang et al. [5] also studied the manufacturing rescheduling problem and proposed GA to resolve the problem.

Vieira et al. [6] describe a framework for understanding rescheduling strategies, policies and methods. In their study they highlighted the significance of understanding rescheduling that addresses some aspects of scheduling theory and practice. Silva et al. [7] presented a comparative study of GA and ACO applied to the online re-optimization of a logistic scheduling problem. Their study indicated that although GA converges faster; however, in dynamic environment, it fails to cope with the disturbances unless they re-optimize the problem. On the other hand, the ant colonies are able to find new optimization solutions without re-optimizing the problem, through the inspection of the pheromone matrix. This study signifies the efficacy of the ACO in resolving manufacturing rescheduling problem. Hozak and Hill [8] highlighted some of the issues and opportunities regarding re-planning and rescheduling frequencies. Potthoff et al. [9] studied the railway crew rescheduling problem of the Dutch railway network when a disruption occurs and proposed an algorithm based on column generation techniques combined with Lagrangian heuristics. Their findings indicated that the proposed algorithm was efficient in rescheduling. These studies indicate that the rescheduling problem has been addressed by many researchers and they are still developing new robust methods to resolve these problems more efficiently. The present study will also address the rescheduling problem and proposes a MACO algorithm which will be discussed later in the upcoming section.

A great amount of literature exists that deals with the multi-objective optimization (MOO) problems in the manufacturing context. Several researchers advocate the use of EAs in resolving the complexity of the MOO problems such as the use of particle swarm optimization (PSO), GA, ACO and bee colony optimization (BCO) [10–12]. Moreover, several researchers have also proposed hybrid evolutionary optimization algorithms to deal with specific optimization problems. Zitzler et al. [13] point out that popularity of the EAs in resolving the multi-objective problems is due to their inherent parallelism and their capability to exploit similarities of solutions by crossover. Tan et al. [14] proposed an evolutionary artificial immune system algorithm to solve the MOO problem. They examined the effectiveness of the proposed algorithm based upon seven benchmark problems characterized by different difficulties in local optimality, non-uniformity, discontinuity, non-convexity, high-dimensionality and constraints. Wei and Yuying [15] applied the Pareto-based multi-objective GA to optimize sheet metal forming process. Sbalzarini et al. [16] also favour the use of EAs for the MOO problem. A multi-objective vehicle routing problem (VRP) was studied

by Jozefowicz et al. [17]. Coello [18] provided a general overview of the work that has been done in the last twenty years in evolutionary MOO and highlighted some of the methodological issues related to the use of the multi-objective evolutionary algorithms (MOEA). He also suggested some of the general research trends in this area.

Schaffer [19] proposed a vector evaluated GA (VEGA) method that consists of a simple GA with a modified selection mechanism to solve the multi-objective optimization problem. This approach was criticized due to number of deficiencies such as its inability to retain solutions with acceptable performance that could be possibly good candidates for becoming non-dominated solutions. However, Zitzler et al. [13] emphasized, in their study of the comparison of the different MOEA, that VEGA were superior to other EAs such as Hajela's and Lin's weighted sum-based approach (HLGA) [20] and the Niche Pareto GA (NPGA). Deb and Jain [21] suggested a couple of running metrics for measuring the convergence to a reference set and for measuring the diversity in population members at every generation to reveal important insights and dynamics of the working of an MOEA. Loetamonphong et al. [22] proposed a genetic-based algorithm to find the "Pareto optimal solutions" for the multi-objective problems. In particular, they studied a new class of optimization problems which have multiple objective functions subject to a set of fuzzy relation equations. Srinivas and Deb [23] investigated Goldberg's notion of non-dominated sorting in GAs along with a niche and speciation method to simultaneously find multiple Pareto-optimal points. Konak et al. [24] also used GA with specialized fitness functions for solving problems with multiple objectives.

The studies discussed above shows the popularity of the EAs in multi-objective optimization problems. Earlier, Silva et al. [7] pointed out that ACO algorithm was better in resolving the rescheduling problems as compared to the GA. Dorigo et al. [25] in his study suggested how best to apply ACO to dynamic and stochastic variations and highlighted the importance of having a better understanding of the theoretical properties of ACO algorithm. Realizing the popularity of the EAs in resolving the MOO problems, this research proposes a MACO algorithm. The ACO algorithm takes inspiration from the foraging behaviour of the ant species. These ants deposit a substance known as 'pheromone' while travelling on the ground in order to mark some favourable path that should be followed by other members of the colony. If there are multiple paths to the destination, the ants follow the path that has the highest concentration of the pheromones deposited by the other ants. ACO algorithm uses a similar mechanism for solving optimization problems. Several researchers have successfully applied the ACO algorithm to solve the multi-objective optimization problems such as Dorigo et al. [26], Grave et al. [27], García-Martínez et al. [28], and Yagmahan and Yenisey [29]. The proposed MACO algorithm amalgamates the property of the multiple ant colony system (ACS) to meet the objective of minimizing the make-span and the minimum changes in the manufacturing plan. The next section discusses the problem formulation part of this research.

## 12.3 Problem Formulation

In the manufacturing scenario studied in this research, the introduction of a new order or disruption causes the change in the manufacturing strategy and thus the manufacturing plans needs to be rescheduled to accommodate these changes. These changes make the manufacturing process more complex. Although, the introduction of the flexible manufacturing system somehow eases such type of problems, this problem is still complex because of the frequent changes in the manufacturing plan. Whenever, a manufacturing plan needs to be rescheduled, the parts and raw materials are required to move from one machine to another using an automated guided vehicle. Thus, it causes chaos in the manufacturing process plan. To avoid the aforementioned chaos the processes need to be altered as minimum as possible.

Therefore, to address the manufacturing rescheduling problem in this research the objective is to achieve a trade-off between the minimization of the total make-span and the minimum changes in the manufacturing schedule/plan. The mathematical formulation of the model studied in this research is explained in the following subsections. The notations used are presented in Appendix (12.A). The mathematical formulation of the model is presented below which discusses the decision variables, constraints and the objective function in detail.

### 12.3.1 Decision Variables

Several decisions variables used in the study are characterized by binary or 0–1 integer values as shown below:

$$X_{cnim} = \begin{cases} 1, & \text{if operation } i \text{ of part } n \text{ is assigned on machine } m \text{ for the} \\ & \text{customer order } c \\ 0, & \text{otherwise} \end{cases} \quad (12.1)$$

$$Z_{cnim} = \begin{cases} 1, & \text{if predecessor of operation } i \text{ of part } n \text{ processed for customer} \\ & \text{order } c \text{ on the machine } m \\ 0, & \text{otherwise} \end{cases} \quad (12.2)$$

$$\gamma_{ijm} = \begin{cases} 1, & \text{if operation } i \text{ precedes operation } j \text{ on the machine } m \\ 0, & \text{otherwise} \end{cases} \quad (12.3)$$

$$Y_{cnij} = \begin{cases} 1, & \text{if there is a precedence relation between operation } i \text{ and } j \\ & \text{for the part type } n \text{ of the customer order } c \\ 0, & \text{otherwise} \end{cases} \quad (12.4)$$

### 12.3.2 Constraints

Apart from the decision variables considered in this research, a number of constraints were also taken into account such as precedence constraint, machine constraint and operational time constraint. The explanation and mathematical expressions of these constraints are expressed below.

*C1: Precedence constraints.* This constraint signifies that the precedence relationship between operation  $i$  and operation  $j$  for the part type  $n$  of the customer order  $c$  is feasible if:

$$Y_{cnij} (X_{cnim} S_{cnim} + X_{cnim} PT_{cnim}) \leq X_{cnjm} S_{cnjm} \quad \forall c, n, i, j, m \quad (12.5)$$

*C2: Machine constraints.* This constraint implies that the machine will start a new operation only after the completion of the previous operation. This constraint can be expressed as

$$\xi(1 - \gamma_{ijm})(X_{cnjm} S_{cnjm} + X_{cnim} S_{cnim}) \leq PT_{cnim} X_{cnim} \quad \forall c, n, i, j, m \quad (12.6)$$

where  $\xi$  is a very large positive number.

*C3: Operational Time Constraint.* This constraint signifies that the completion time of each operation should be either positive or zero, i.e.,

$$PT_{cnim} \geq 0 \quad (12.7)$$

*C4: Operation Constraint.* This constraint implies that operation is performed only on one machine and is expressed as:

$$\sum_{m=1}^M X_{cnim} = 1 \quad (12.8)$$

### 12.3.3 Objective Functions

This research deals with a multi-objective optimization problem aiming to achieve the trade-off between the minimization of the total make-span and the minimum changes in the manufacturing plan. Therefore, this section outlines the mathematical formulation of the objectives of this research.

The total time required to process all the parts for all the customer orders can be calculated as:

$$TTR = \sum_{c=1}^C \sum_{n=1}^N \sum_{i=1}^I \sum_{m=1}^M PT_{cnim} X_{cnim} \quad (12.9)$$

Since parallel processing of parts takes place, the working time for each machine can be calculated as:

$$WT_{cm} = \sum_{n=1}^N \sum_{i=1}^I PT_{cnim} X_{cnim} \quad (12.10)$$

The first objective of the proposed model is to minimize the make-span; hence, the first objective function can be mathematically expressed as:

$$\text{Objective Function } A = \text{Minimize } [Max (WT_{cm})] \quad (12.11)$$

The second objective function deals with the minimization of number of changes in the manufacturing plan which is expressed as

$$\text{Objective Function } B = \text{Minimize } (\text{CHNG}) \quad (12.12)$$

Since both the objectives contradict each other, it is a multi-objective optimization problem. In order to resolve these objectives in this research a weighted sum method has been used. Hence, the overall objective is

$$\text{Overall Objective} = \text{Minimize } (w1 * WT_{cm} + w2 * \text{CHNG}) \quad (12.13)$$

where  $w1$  and  $w2$  are weights corresponding to objective function A and B. These weights are assigned according to the preference by the decision maker. The next section elaborates the proposed MACO algorithm in detail.

## 12.4 MACO Algorithm

The nature has always attracted the attention of the researchers worldwide to resolve the real-world optimization computational problems. This is due to the capability of the nature-inspired algorithms to handle the increasing size and complexity of the real-world problems. EAs, a subset of the evolutionary computation have emerged as popular method that has been widely applied by researchers to resolve the MOO problems. This popularity of EAs has led to the development of a number of nature-inspired EAs such as PSO [10, 30], BCO [12, 31], and ACO algorithm [11, 32].

All these EAs mimic certain properties of the nature while resolving complex problems. For example, the PSO method is inspired from the flocking of birds. Although all the algorithms are capable of resolving the complex optimization problems, ACO was adopted to resolve the manufacturing rescheduling problem for the purpose of this research. Deneubourg et al. [33] comprehensively examined the pheromone laying and following behaviour of ants. They studied the behaviour of Argentine ants in an experiment known as the “double bridge experiment”. The experiment showed that the ants favoured the path where the highest concentration of the pheromones was found. The proposed algorithm is the modified version of the simple ACO algorithm and has been termed as MACO

algorithm. This proposed MACO algorithm incorporates the concept of multiple ant colonies system [34] and this will be discussed later in this chapter. The newly added properties to the original ACO make it more capable of efficiently resolving the multi-objective optimization problem. This will be also justified later in the results and discussions section while comparing the results with the other EAs.

The ACO algorithm firstly proposed by Dorigo [11] was inspired from the behaviour of real ants seeking a path between their colony and a source of food, to search for an optimal path. Dorigo [11] stated that ants are social insects, that live in colonies and whose behaviour is directed more to the survival of the colony as a whole than to a single individual component of the colony. Ants accomplish the task of searching the food by travelling randomly and then returning to the colony once being successful. The other ants, following the one, smell the pheromones and apply a probabilistic approach in selecting the node with the highest pheromone trail on the paths. Following the pheromone trail ants reach to the food source or back to the nest and vice versa. Using this concept in the real-world problem, the ants acting as an agent are initially randomly generated on nodes, which stochastically move from a start node to feasible neighbour node [34]. The agents collect and store information in pheromone trails while looking for the feasible solution. The pheromone evaporates during the search process to avoid local convergence and to explore more search areas. The next moving ant in turn leaves new pheromone that is added to the already existing one and the probability to choose a node depends on intensity of pheromone trail perceived. Therefore, the ants move in an autocatalytic process, favouring the path along which more ants have travelled and by traversing all the nodes [32]. This understanding is developed as an algorithm or heuristics and applied to resolve a number of different computational problems. Kawamura et al. [35] in their study of the colony level interactions of ants therefore defines ACS as a constructive population based search technique to solve optimization problems by using the principle of pheromone information. In the proposed MACO algorithm the pheromone trail behaviour of the ants will be used to find the optimal solution.

The proposed MACO algorithm was developed earlier by Chan and Kumar [34] to design a balanced and efficient supply chain network that maintains the best balance of transit time and customer service. In the proposed MACO algorithm, ants are defined as simple computational agents having some memory. The ants are assumed to be living in an environment where time is discrete and finds an optimal solution using a dynamic memory structure incorporating information on the effectiveness of previously obtained results. Chan and Swarnkar [32] highlighted the characteristics of the ACO that includes:

- ACO is a method to construct solutions which balances pheromone trails (characteristics of past solutions) with a problem-specific heuristic (normally, a simple greedy rule).
- It is a method to both reinforce and evaporate pheromone, and
- Local (neighbourhood) search to improve solutions.

Therefore in the ACS the ants which act as agents, iteratively construct solutions to combinatorial optimization problems. The different optimization problems can be



resolved using the ACS such as the VRP [36], travelling salesman problem (TSP) [37–39], quadratic assignment problem (QAP) [40] and production scheduling problem (PSP) [41]. The solution generation by ants is guided by pheromone trails and the problem specific heuristic information. The efficiency of the ACS in handling the discrete optimization problems is evidence in the literature [42–47]. Chan and Kumar [34] in their earlier work identify that ACS works mainly in four phases:

1. *Initialization*. This phase involves providing initial guidelines to the ants to pursue the movement in upcoming phases. These guidelines include the laying of initial pheromone trails on the paths and to diversify the search at preliminary stages by exploring maximum possible alternative paths.
2. *Node Transition*. The second phase is the most crucial phase characterized by the movement of ants on different nodes using a probabilistic approach that is based on a trade-off between visibility and pheromone trails. The quality of solution depends upon its working strategy. The nodes in this chapter refer to the machines.
3. *Updating*. The third phase is marked by the pheromone update on different paths exploiting the tours travelled by ants. In the proposed MACO algorithm the fuzzy function will be used during the updating process. The intensity of pheromone trails decides the motion of ants in the following cycles.
4. *Stopping Criteria*. The final phase deals with the termination of the algorithm which specifies explicit number of cycles to be completed after which the algorithm stops and the tour with the best result is given as the output.

Although the ACO algorithm is capable of resolving different combinatorial optimization problem because of its flexible nature, often these algorithms face the problem of stagnation and quick convergence. The nature of the ACO algorithm to get entrapped in the local minima is because it depends only on a positive feedback mechanism by the use of pheromone. To overcome this demerit of the ACO algorithm in the proposed MACO algorithm several ant colonies are used irrespective of the general ant system. These different colonies of the ants interact with each other to find a global optimal solution. The colony-level interactions also results in both positive and negative feedback that can be controlled and hence it works better with respect to the general ACS algorithm. This algorithm works with multi-dimensional data set, in which each dimension represents a definite collection of nodes with similar attributes. Each node has its characteristics dimension and a different ant colony moves in each dimension with the mutual cooperation of its “clone ants” in other colonies. In this system, each colony has the same number of ants and every colony contains the same number of clones of the ants so that they can share information from the other colonies. Thus, one colony of ants is totally devoted to find the optimal solution and once it achieves its objective the results are shared with the other colonies to avoid the repetition.

The proposed MACO algorithm uses  $M$  colonies and each colony is comprised of  $N$  ants. The ants in the colony have clone ants in other colonies that work together to find the best optimal solution. Each colony of ants associates with different objective, each colony having its own pheromone structure. During the

initialization phase i.e., at  $t = 0$ , ants are positioned on different nodes; however, the ants can freely move in any direction. The  $n^{\text{th}}$  ant belonging to the  $m^{\text{th}}$  colony is denoted as  $(m, n)$ . At each time  $t$ ,  $M \times N$  ants move between the nodes to search the optimal path. Initially ants are moved randomly for a few cycles without any visibility or pheromone trail identification capacity [34]. This takes the algorithm out of initial stagnation and provides initial guidelines for the ants to start with. The notations are presented in Appendix (12.B).

Let  $\tau_{ij}^m(t)$  be the intensity of the pheromone on the edge  $(i, j)$  i.e., a path from node  $i$  to node  $j$  in the  $m^{\text{th}}$  colony at time  $t$ . Initial value of  $\tau_{ij}^m(t)$  for trail intensity on edges are set as zero. After each ant has completed its tour by  $\Gamma$  time intervals, the intensity of the pheromone  $\tau_{ij}^m(t)$  becomes,

$$\tau_{ij}^m(t + \Gamma) = \rho \cdot \tau_{ij}^m(t) + \sum_{n=1}^N \Delta\tau_{ij}^{mn} \quad (12.14)$$

where  $\rho$  is a coefficient such that  $(1 - \rho)$  represents the evaporation rate of the pheromone between time  $t$  and  $t + \Gamma$ . The value of  $\rho$  must be set to a positive value less than 1 ( $\rho < 1$ ) to avoid unlimited accumulation of the pheromone. The value of  $\Delta\tau_{ij}^{mn}$  represents the intensity per unit of length of the edge  $(i, j)$  for ant  $(m, n)$  and this is given as

$$\Delta\tau_{ij}^{mn} = \begin{cases} Q/T_{ij}^{mn}, & \text{if ant } (m, n) \text{ uses edge } (i, j) \text{ between time } t \text{ and } (t + \Gamma) \\ 0, & \text{otherwise} \end{cases} \quad (12.15)$$

$Q$  is a constant and scarcely affects the behaviour of the algorithm.  $T_{ij}^{mn}$  is the total transit time of ant  $(m, n)$  i.e., the ant generating a tour with minimum time can lay a larger intensity of the pheromone on its tour. Two tabu lists are associated with each ant to do away the repetition of the nodes in an ant's path. These tabu lists are simply a data structure and it is different from the tabu list uses in tabu search.  $\text{Tabu list}_1^{mn}$  keeps the track of the nodes encounters in the whole path by the ant  $(m, n)$  and it is emptied after one cycle whereas the  $\text{Tabu list}_2^m$  stores the information about the optimal path found by colony and this can be used by the ants of different colonies to leave the nodes which is encountered by first colony and thus best possible tour of every colony can be determined separately.

The transition probability from node  $i$  to  $j$  for ant  $(m, n)$  is defined below:

$$P_{ij}^{mn}(t) = \begin{cases} \frac{\pi_{ij}^m(t)}{\sum_{r \notin \text{tabu}^{mn}(t)} \pi_{ij}^m(t)} & \text{if } j \notin \text{tabu}^{mn}(t) \\ 0, & \text{otherwise} \end{cases} \quad (12.16)$$

$$\pi_{ij}^m(t) = \{\pi_r [\tau_{mn}^r(t) + c(m)]^\alpha\} [\eta_{ij}]^\beta \quad (12.17)$$

Here,  $\text{tabu}^{mn}(t)$  indicates the tabu list of ant  $(m, n)$  at time  $t$ . This list consists of nodes that have already been visited until time  $t$ . The tabu list is emptied after one

cycle and ants can again freely choose any node where,  $r$  denotes the node different from  $j$ .  $\pi_{ij}^m(t)$  means the degree of preference for an edge  $(i, j)$  connected to node  $j$  and if this value is large then the ant  $(m, n)$  tends to choose node  $j$  as the next one to visit. The value of  $\pi_{ij}^m(t)$  depends on the objective function. The parameters  $\alpha$  and  $\beta$  must be set in advance that controls the relative importance of trail versus visibility. In addition  $\eta_{ij}$  is defined as visibility which is the inverse of difference between the processing time and the machine node  $(i, j)$ .

After completion of a tour, pheromone trail on each edge is updated using Eqs. 12.16 and 12.17. A variable array “best tour” keeps the track of the overall best result performed. An allocation is considered the best one which associated with the minimum transit time between nodes. The proposed algorithm is repeated until the required stopping criterion is fulfilled. The steps of the complete algorithm are described in Appendix (12.C). The next section discusses a small case study to demonstrate the efficacy of the proposed MACO algorithm.

## 12.5 Case Study

To demonstrate the efficacy and robustness of the proposed MACO algorithm on the mathematical model studied in this research, a randomly generated constrained example is considered. The numerical has two objectives; to minimize the make-span and to have minimum changes in the manufacturing schedule. A number of constraints were taken into account that has been discussed earlier in the problem formulation section. The problem generation scheme used in the research is demonstrated below.

In this example total eight machines are considered to manufacture six parts. These are total 17 different operations to be performed on the parts. Each operation can be carried out on more than one machine. The data for the alternative machines and corresponding processing times for the different operations and part types are presented in Table 12.1. There is also a precedence relationship between the different operations and these relationships are shown in Table 12.2. In this problem it is assumed that after 310 min, a new part i.e., part no. 6 is introduced which consists of total two operations. The main objective is to accommodate this new order/part type and simultaneously minimize the total make-span with the minimum changes in the manufacturing plan. In this case study equal weights have been assigned for both the objectives.

## 12.6 Results and Discussions

The MACO algorithm achieves the optimal or near-optimal solutions for the objective considered in this research and emerges as a powerful EA. The make-span before the introduction of the new order was 605 min (Table 12.3). In the

**Table 12.1** Data for the case study

Part	Operation	Alternative available machines
1	1	3(210),7 (215)
2	2	1(380), 2 (310)
2	3	2 (140), 5 (130)
2	4	1(216), 8 (220)
2	5	2 (234), 3 (140), 6 (315)
2	6	1 (261), 6 (165)
3	7	1 (321), 7 (318), 8 (390)
3	8	2 (198), 3 (234), 8 (114)
4	9	5(235)
4	10	4 (155), 8(123)
5	11	1 (321), 8 (30)
5	12	2 (43),5 (25)
5	13	3 (122), 7(104)
5	14	4 (38)
4	15	1 (22), 7 (52)
6	16	3(152),4(138), 5(140)
6	17	1 (40)

**Table 12.2** Precedent relationship between operations

2 → 3
3 → 6
4 → 5; 5 → 6
7 → 8
9 → 10
11 → 12; 13 → 14; 12 → 15; 14 → 15
16 → 17

**Table 12.3** Initial manufacturing sequence

	Operation		Operation		Operation	
	Start time (min)	End time (min)	Start time (min)	End time (min)	Start time (min)	End time (min)
M1	4		15			
	0	216	487	509		
M2	2					
	0	310				
M3	1		5			
	0	210	216	356		
M4	10		14			
	235	390	422	460		
M5	9		3		12	
	0	235	310	440	462	487
M6	6					
	440	605				
M7	7		13			
	0	318	318	422		
M8	8		11			
	318	432	432	462		

**Table 12.4** Final manufacturing sequence generated by MACO

	Operation		Operation		Operation	
	Start time (min)	End time (min)	Start time (min)	End time (min)	Start time (min)	End time (min)
M1	4		17		15	
	0	216	375	415	487	509
M2	2		3			
	0	310	310	450		
M3	1		5			
	0	210	216	356		
M4	10		14			
	235	390	422	460		
M5	9		16		12	
	0	235	235	375	462	487
M6	6					
	450	615				
M7	7		13			
	0	318	318	422		
M8	8		11			
	318	432	432	462		

case study discussed earlier, when a new order is introduced the proposed MACO algorithm accommodates the new order and generates a new manufacturing plan. During the generation of the new schedule by MACO algorithm, only one part was required to be shifted from the previous schedule and the total new make-span was found to be 615 min (Table 12.4). However, when the same problem was tested using the simple ACO algorithm, it resulted in two changes in the manufacturing plan and the make-span was found to be same i.e., 615 min.

Thus, the proposed MACO algorithm causes minimum changes in the manufacturing schedule and at the same time minimizes the make-span. Hence, MACO algorithm was found to be more efficient than the normal ACO. The MACO algorithm was coded in the C++ programming language and the outcome shows the potential of the proposed algorithm in resolving the manufacturing rescheduling problem.

## 12.7 Conclusions

In this research a multi-objective manufacturing rescheduling problem has been tackled using MACO algorithm. During the rescheduling process the implementation of the manufacturing plan faces many difficulties. Hence, the main aim of

this research is to overcome the difficulty during the rescheduling of the manufacturing processes. The objective of the research is twofold; to minimize the make-span and to simultaneously reduce the number of changes in the manufacturing plan. This will help the manufacturing firm to accommodate the new order without making significant changes in the manufacturing process. Thus, this paper makes significant contribution to the existing literature. Use of MACO algorithm to resolve the rescheduling problem further helps to overcome the shortcomings of the general ACS since the general ACS works on the positive feedback mechanism by the use of pheromone. Thus, the general ACS has a tendency to get trapped in the local minima. However, in the MACO algorithm, the different colonies of the ants interact with each other to find a global optimal solution. Therefore, with the help of colony-level interactions, both positive and negative feedback can be controlled and thus has better performance. Nevertheless, the MACO algorithm performs better than the general ACS, as is evident from the outcome of the analysis.

In the future several other factors can be taken into account during the rescheduling of the manufacturing processes such as the machine reliability, outsourcing and vehicle routing functions. Thus, the research has an adequate scope for further extension by making the problem more complex and adding more objective functions. Future research may also involve testing the efficacy of the MACO algorithm under diverse manufacturing scenarios.

## Appendix 12.A

---

*C*: Customer demand index,  $c = \{1, 2, 3, 4, \dots, C\}$

*N*: Part number,  $n = 1, 2, \dots, N$ .

*I*: Operation index,  $I = 1, 2, \dots, I$ .

*M*: Machine index,  $m = 1, 2, \dots, M$ .

Scinm: Start time of operation *i* for part *n* on machine *m* for customer demand *c*.

MTc: Make-span time for customer demand *c*.

PTcnim: Processing time for operation *i* of part *n* assigned on machine *m* for Customer demand *c*.

WTcm: Working time of machine *m* for completing customer demand *c*.

TTRc: Total time required for processing of all parts for customer demand *c*.

CHNG: Number of changes in manufacturing plan

---

## Appendix 12.B

---

$M$	Number of colonies
$N$	Number of ants to each colony
$Q$	A constant
$R$	Nodes still to be visited by ant
$i, j$	Machine nodes
$T_{ij}^m(t)$	Intensity of the pheromone on the edge $(i, j)$ in the $m$ th colony at time $t$ .
$\rho$	A coefficient
$(1 - \rho)$	Evaporation rate of the pheromone
$\text{Tabu list}_1^{mm}$	Intensity of the pheromone per unit of length of the edge $(i, j)$ for ant $(m, n)$
$\text{Tabu list}_2^m$	Total transit time of ant $(m, n)$ in travelling edge $(i, j)$
$\text{Tabu list}_1^{mm}$	List of nodes encounters in the path of ant $(m, n)$
$\text{Tabu list}_2^m$	List of nodes which creates the optimum path in colony $m$
$\text{Tabu}^{mm}(t)$	Tabulist of ant $(m, n)$ at time $t$
$P_{ij}^{mm}(t)$	Transition probability from node $i$ to $j$ for ant $(m, n)$ at time $t$
$\Pi_{ij}^m(t)$	Degree of preference for an edge $(i, j)$ connected to node $j$ at time $t$
$\eta_{ij}$	Visibility from node $i$ to $j$
$\alpha$	Factor that controls the importance of trail
$\beta$	Factor that controls the importance of visibility
$N_C$	Counter for number of cycles
$N_{C,max}$	Maximum number of cycles

---

## Appendix 12.C: Steps of MACO algorithm

---

### Step 1: Initialization

Set  $t = 0$ ; /\* time Counter\*/

Set  $N_C = 1$ ; /\* number of iterations/number of cycles counter\*/

Set  $\tau_{ij}^m(0) = c$ ; on each node/\* this is the initial pheromone trail on the edge  $(i, j)$  and  $c$  is a small positive number\*/

Set  $\Delta\tau_{ij}^{mm} = 0$ ; on each node. /\* this is the increase in the trail level on edge  $(i, j)$ \*/

Set  $\text{tabu list}_1^m = 0$ ; /\*  $\text{tabu list}_1^{mm}$  gives the list of nodes traversed by ant  $(m, n)$ \*/

Set  $\text{tabu list}_2^m = 0$ ; /\*  $\text{tabu list}_2^m$  gives the list of optimum paths traversed in colony  $m$ \*/

### Step 2

Set all ants at the starting node

Set  $s = 0$ /\*  $s$  is the index of  $\text{tabu list}_1^{mm}$ \*/

For  $m = 1$  to  $M$  do/\*  $M =$  total number of colonies\*/

For  $n = 1$  to  $N$  do/\*  $N =$  total number of ants in each colony\*/

Place the starting node of the  $(m, n)$  and insert it in  $\text{tabu list}_1^{mm}(t)$

### Step 3

Generate the path based on pheromone feedback

For  $m = 1$  to  $M$  do

For  $n = 1$  to  $N$  do

---

(continued)

(continued)

---

For  $s = 1$  to  $S - 1$  do

Choose the node  $j$  to move with the transition probability  $P_{ij}^{mm}(t)$ , given in Eq. 12.15.

Move the  $(m, n)$  ant to the node  $j$

Insert the node  $j$  in  $tabu\ list_1^{mm}(t)$

Step 4

For  $m = 1$  to  $M$  do

For  $n = 1$  to  $R$  do/\*  $R$  is the max number of nodes in a colony  $m$  which is to be visited by ants, the value of  $R$  can be find by the rule discussed in [34]

Repeat the following until failure is obtained.

For every node  $i = 1$  to  $I$  do

Examine the balancing criteria and the objective function for edge  $(i, j)$  of ant  $(m, n)$ .

Update the shortest path find and place it in  $tabu\ list_2^m$  for every colony.

Step 5

Compute changes of the pheromone in every colony

For  $m = 1$  to  $M$  do

For every edge  $(i, j)$

Update the intensity of pheromone by Eqs. 12.15 and 12.16.

Set  $t = t + n$

Set  $N_C = N_C + 1$

Step 6: Checking stopping criteria

If  $(N_C < N_{C, \max})$

Then

Empty  $tabu\ list_1^{mm}$  and Goto step 2.

Else

Update  $tabu\ list_2^m$  for each colony

Print the Output

STOP

---

## References

1. Yamamoto, M. (1985). Scheduling/rescheduling in the manufacturing operating system environment. *International Journal of Production Research*, 23(4), 705–722.
2. Wu, S. D., Storer, R. H., & Chang, P. C. (1993). One-machine rescheduling heuristics with efficiency and stability as criteria. *Computers and Operations Research*, 20(1), 1–14.
3. Abumaizar, A. J., & Svestka, J. A. (1997). Rescheduling job shops under random disruptions. *International Journal of Production Research*, 35(7), 2065–2082.
4. Jain, A. K., & ElMaraghy, H. A. (1997). Production scheduling/rescheduling in flexible manufacturing. *International Journal of Production Research*, 35(1), 281–309.
5. Fang, H.L., Ross, P. & Corne, D. (1993). A promising genetic algorithm approach to job-shop scheduling, rescheduling, and open-shop scheduling problems. In S. Forrest (Ed.), *Proceedings of the 1st Annual Conference on Genetic Algorithms* (pp. 375–382) San Mateo: Morgan Kaufmann.
6. Vieira, G. E., Herrmann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1), 39–62.
7. Silva, C. A., Sousa, J. M. C., & Runkler, T. A. (2008). Rescheduling and optimization of logistic processes using GA and ACO. *Engineering Applications of Artificial Intelligence*, 21(3), 343–352.



8. Hozak, K., & Hill, J. A. (2009). Issues and opportunities regarding replanning and rescheduling frequencies. *International Journal of Production Research*, 47(18), 4955–4970.
9. Pothhoff, D., Huisman, D. & Desaulniers, G. (2010). Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, published online in Articles in Advance, May 25, 2010.
10. Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*. Vol. 4. (pp. 1942–1948).
11. Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*, PhD Thesis, Politecnico di Milano, Italie.
12. Pham, D.T. & Ghanbarzadeh, A. (2007). Multi-objective optimization using the Bees Algorithm. *Proceedings of IPROMS 2007 Conference*.
13. Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionary Computation*, 8(2), 173–195.
14. Tan, K. C., Goha, C. K., Mamuna, A. A., & Ei, E. Z. (2008). An evolutionary artificial immune system for multi-objective optimization. *European Journal of Operational Research*, 187(2), 371–392.
15. Wei, L., & Yuying, Y. (2008). Multi-objective optimization of sheet metal forming process using Pareto-based genetic algorithm. *Journal of Materials Processing Technology*, 208(1–3), 499–506.
16. Sbalzarini, I.F., Müller, S. & Koumoutsakos, P. (2000). Multiobjective optimization using evolutionary algorithms. *Proceedings of the Summer Program*, Center for Turbulence Research, NASA.
17. Jozefowicz, N., Semet, F., & Talbi, E. G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2), 293–309.
18. Coello, C. A. (2006). Evolutionary multiobjective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1), 28–36.
19. Schaffer, J.D. (1984). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD Thesis, Vanderbilt University.
20. Hajela, P., & Lin, C. Y. (1992). Genetic search strategies in multi-criterion optimal design. *Structural Optimization*, 4, 99–107.
21. Deb, K. & Jain, S. (2002). Running performance metrics for evolutionary multi-objective optimization. Technical Report, KanGAL, Indian Institute of Technology, Kanpur 208016, India.
22. Loetamonphong, J., Fang, S. H., & Young, R. E. (2002). Multi-objective optimization problems with fuzzy relation equation constraints. *Fuzzy Sets and Systems*, 127(2), 141–164.
23. Srinivas, N., & Deb, K. (1994). *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*. MIT Press, 2(3), 221–248.
24. Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: a tutorial. *Reliability Engineering and System Safety*, 91(9), 992–1007.
25. Dorigo, M., Birattari, M. & Stützle, T. (2006). Ant colony optimization: artificial ants as a computational intelligence technique. *IRIDIA—Technical Report Series*, Technical Report No. TR/IRIDIA/2006-023.
26. Dorigo, M., Maniezzo, V., & Colomi, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, Cybernetics—Part B: Cybernetics*, 26(1), 29–41.
27. Gravel, M., Price, W. L., & Gagné, C. (2002). Scheduling continuous casting of aluminium using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, 143(1), 218–229.
28. García-Martínez, C., Cordon, O., & Herrera, F. (2007). A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*, 180(1), 116–148.
29. Yagmahan, B., & Yenisey, M. M. (2008). Ant colony optimization for multi-objective flow shop scheduling problem. *Computers and Industrial Engineering*, 54(3), 411–420.

30. Chan, F. T. S., Kumar, V. & Mishra, N. (2007). A CMPSO algorithm based approach to solve the multi-plant supply chain Problem. In Felix T.S. Chan & Manoj Kumar Tiwari (Ed.), *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*. Vienna, Austria: I-Tech Education and Publishing, ISBN: 978-3-902613-09-7.
31. Chong, C. S., Low, M. Y. H., Sivakumar, A. I. & Gay, K. L. (2006). A bee colony optimization algorithm to job shop scheduling. *Proceedings of the 2006 Winter Simulation Conference*. December 3–6, 2006. (pp. 1954–1961) Monterey, CA USA.
32. Chan, F. T. S., & Swarnkar, R. (2006). Ant colony optimization approach to a fuzzy goal programming model for a machine tool selection and operation allocation problem in an FMS. *Robotics and Computer-Integrated Manufacturing*, 22, 353–362.
33. Deneubourg, J. L., Aron, S., Goss, S., & Pasteels, J. M. (1990). The self organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3, 159–168.
34. Chan, F. T. S., & Kumar, N. (2009). Effective allocation of customers to distribution centres: a multiple ant colony optimization approach. *Robotics and Computer-Integrated Manufacturing*, 25, 1–12.
35. Kawamura, H., Yamamoto, M., Suzuki, K. & Ohuchi, A. (2000). Multiple ant colonies algorithm based on colony level interactions. *Publication in the IEICE Transactions, Fundamentals*, E83-A (Vol. 2, pp. 372–379).
36. Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999a). Applying the ant systems to the vehicle routing problem. In S. Voss, S. Martello, I. H. Osman, & C. Roucaïrol (Eds.), *Meta-Heuristics: Advances and Trends in Local search Paradigms for Optimization*. (pp. 285–296), Dordrecht, Netherlands, Kluwer Academic Publishers.
37. Golden, B. & Stewart, W. (1985). *Empiric Analysis of Heuristics in the Travelling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy-Kan & D.B. Shmoys (Eds.), New York: Wiley.
38. Lawler, E. L., Lenstra, J. K., Rinnooy-Kan, A. H. G., & Shmoys, D. B. (1985). *The Travelling Salesman Problem*. New York: Wiley.
39. Dorigo, M., & Gambardella, L. M. (1997). Ant Colonies for the travelling salesman problem. *BioSystems*, 43, 73–81.
40. Maniezzo, V., & Colorini, A. (1999). The ant system applied to the quadratic assignment problem. *IEEE Transactions on Knowledge and Data Engineering*, 11(5), 769–778.
41. Ying, K. C., & Liao, C. J. (2003). An ant colony system approach for scheduling problems. *Production Planning and Control*, 14(1), 68–75.
42. Goss, S., Beckers, R., Deneubourg, J. L., Aron, S. & Pasteels, J. M. (1990) How trail laying and trail following can solve foraging problems for ant colonies. In R.N. Hughes (Ed.). *Behavioural Mechanisms of Food Selection*, NATO-ASI Series, (Vol. G 20, pp. 661–678) Berlin: Springer
43. Gambardella, L. M. & Dorigo, M. (1996). Solving symmetric and asymmetric TSPs by ant colonies. In *Proceedings of the IEEE Conference on the Evolutionary Computation* (pp. 622–627).
44. Dorigo, M., Maniezzo, V. & Colorini, A. (1991). *Positive Feedback as a Search Strategy*, Technical report (pp. 91–106), Dipartimento di Elettronica, Politecnico di milano, Italy.
45. Colorini, A., Dorigo, M. & Maniezzo, V. (1991). Distributed optimization by ant colonies. In F. Vareladn & P. Bourguine (Eds.), *Proceedings of European Conference on Artificial Life*. (pp. 134–142) Paris, France: Elsevier Publishing.
46. Colorini, A., Dorigo, M. & Maniezzo, V. (1992). An investigation of some properties of an ant algorithm. R. Manner & B. Manderick (Eds.), In *Proceedings of Conference on Parallel Problem Solving from Nature* (pp. 509–520). Brussels, Belgium: Elsevier Publishing.
47. Gambardella, L.M., Dorigo, M. (1995). Ant-Q: A reinforcement learning approach to the travelling salesman problem. In *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 252–260).

**Part IV**  
**Systems Design and Analysis**

# Chapter 13

## Reconfigurable Facility Layout Design for Job-Shop Assembly Operations

Lihui Wang, Shadi Keshavarzmanesh and Hsi-Yung Feng

**Abstract** Highly turbulent environment of dynamic job-shop operations affects shop-floor layout as well as manufacturing operations. Due to the dynamic nature of layout changes, essential requirements such as adaptability and responsiveness to the changes need to be considered in addition to the cost issues of material handling and machine relocation when reconfiguring a shop floor's layout. Here, based on the source of uncertainty, the shop-floor layout problem is split into two sub-problems and dealt with by two modules: re-layout and find-route. Genetic algorithm is used where changes cause the entire shop re-layout, while function blocks are utilized to find the best sequence of robots for the new conditions within the existing layout. This chapter reports the latest development to the authors' previous work.

### 13.1 Introduction

A facility layout problem is to find a one-to-one mapping between machine types and their locations on a shop floor based on the operation routings of products. It is one of the key areas that significantly affect the manufacturing productivity in

---

L. Wang (✉)

Virtual Systems Research Centre, University of Skövde, 541 28 Skövde, Sweden

e-mail: lihui.wang@his.se

S. Keshavarzmanesh

Department of Mechanical and Materials Engineering, The University of Western Ontario, London, ON N6A 5B9 Canada

e-mail: skeshava@uwo.ca

H.-Y. Feng

Department of Mechanical Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada

e-mail: feng@mech.ubc.ca

**Table 13.1** Choice of a layout type [4]

Cost of re-layout	Uncertainty of future production requirement	
	Low	High
Low	<i>Dynamic layout</i>	<i>Reconfigurable layout</i>
High	<i>Robust layout</i>	<i>Distributed layout</i>

terms of cost and time. An effective facility layout can reduce the operating cost of an industry from 10 to 30% [1] by minimising material handling cost, which is the ultimate goal of a facility layout design.

Since production uncertainty becomes one of the most challenging aspects of the manufacturing environments in the 21st century, the success of next generation of intelligent manufacturing depends on its capability of dynamic responsiveness to the production requirements. In such an environment, frequent changes in product design, product mix, production volume and process affect the facility layout as well as other areas [2].

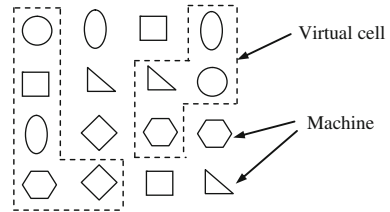
Generally, there exist two types of manufacturing uncertainties: internal and external. The former is due to internal disturbances such as equipment breakdown, job delay, reject and rework, while the latter is caused by external forces such as product demand (volume), product price, product mix and urgent job [3]. From practical point of view and depending on the degree of uncertainty and the cost of re-layout, designers can choose one among the following layout types (Table 13.1).

1. *Dynamic layout*. Considers several production periods, and layouts are determined for each period by balancing material handling costs over all periods and the overall cost of relocating facilities in consecutive layouts.
2. *Robust layouts*. Behaves well over multiple production periods and in different scenarios with low uncertainty.
3. *Distributed layout*. Allows a facility to conform future fluctuations in flow-shop patterns and volumes, particularly when demands fluctuate too much to make facility reconfiguration cost-effective, and especially when there is a large number of machines and machine types. This type of layout can be used to quickly form a temporary (virtual) cell.
4. *Reconfigurable layout*. Aligns itself with the notion of real-time enterprise in which the changes to layout context are readily available, and it keeps operating on the edge by doing real-time layout adjustment with live data [5].

Within the context of a dynamic manufacturing environment, a job shop is characterized by multiple machine types and multiple part types and thus experiences a high degree of variability in material flow due to growing internal and external uncertainty factors in today's manufacturing environment. The choice of layout for a job shop significantly impacts its performance. Some traditional job-shop layouts that are commonly found in the literature are as follows [6].

1. *Functional layout*. Groups the same machine types in a single workcentre that may process parts of different part families. It results in complicated routes

**Fig. 13.1** Virtual cells can be quickly formed in a distributed layout



between workcentres, long throughput time, high work-in-process level and high material handling cost.

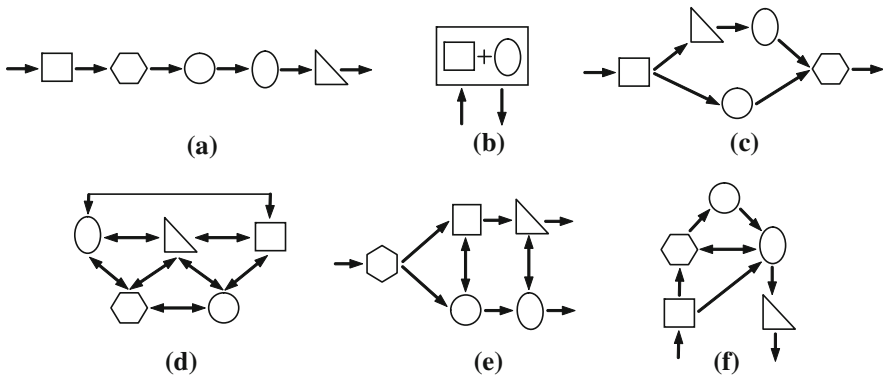
2. *Flow-line layout.* Arranges machine types along a production line. It is usually infeasible for a job shop due to multiple flow routes caused by the diverse part mix.
3. *Cellular layout.* Groups the machines required for the parts with common or similar operation sequences to form manufacturing cells for part families. This leads to limited flexibility in case of machine breakdown or change in product mix.

To overcome the drawbacks of traditional job-shop layouts, the concept of virtual cell formation and hybrid cellular layout have been developed.

Virtual cells are dynamically formed cells in which machines are configured logically and temporarily [7]. A virtual cell is a group of machines only in the system control software (see Fig. 13.1). It allows the time-sharing of workstations physically distributed in different cells belonging to different part families.

Other factors and design issues may be involved in a facility layout problem: the type of workshop and production variables, material handling systems, the number of floors on which the machines can be assigned, workshop shape and size, facility shape and size, the pick-up and drop-off locations, etc. Considering the different aspects of facility layout, it is known to be complex and NP-hard [8]. The facility layout problems addressed in the literature are strongly dependent on the factors that differentiate the nature of the problems.

Different approaches to solving the facility layout problems can be classified into four groups: exact methods, heuristics, meta-heuristics and hybrid approaches. Early attempts have mainly incorporated exact methods such as dynamic programming [9] and usually consider only material handling and rearrangement costs. Bounding procedure is employed to decrease the number of possible states, but it may become too complex when there are a large number of facilities. Heuristics, such as steepest-descent pairwise-interchange [10] were incorporated to overcome the intensive computations of the exact methods. The well-known meta-heuristics, i.e., genetic algorithm [11], ant colony optimization [12], and simulated annealing [13], are suitable for larger-size facility layout problems with multi-objective functions including material handling cost, rearrangement cost, work-in-process, cycle time, etc. Some hybrid methods combining the above-mentioned methods have also been developed to deal with more difficult scenarios such as hybrid assembly lines [14].



**Fig. 13.2** Various layout modules. **a** Flowline module, **b** machining centre module, **c** branched flowline module, **d** functional layout module, **e** patterned flow module, **f** cell module

A hybrid layout is a combination of functional and cellular layouts, which is similar to the concept of virtual cell. It brings the identical machine types next to each other without destroying the allocation of cells to part families. Recently, the concept of hybrid cellular layout is extended by combining different layout modules [6]. The layout modules are shown in Fig. 13.2.

However, dealing with job-shop uncertainty, attempts still need to be made for distributed decision-making at runtime based on environmental changes. Focusing on the turbulent and distributed manufacturing environments, this research proposes to incorporate *function block* technology to increase the adaptability and proactive responsiveness, so that the system is able to autonomously suggest alternative routes among robots based on the changes in a robotic assembly shop floor. This chapter presents the latest development on the assembly shop layout problem as the continuation of the authors' previous work [15].

## 13.2 Assembly Shop Layout Planning

Due to the highly turbulent environment of today's manufacturing environment, the potential to frequently alter layout has transformed the shop-floor layout problem from only considering long-term material handling and machine relocation costs to also considering other essential requirements such as adaptability and proactive responsiveness to the dynamic changes when reconfiguring the shop floor from one layout to another. As shown in Table 13.1, reconfigurable layout is suitable for a highly turbulent manufacturing environment. It has the primary advantage of minimising the material handling cost by reconfiguring a layout when warned by changes. Of course, this cost must not be more than the cost of relocating the equipments.

It is unlikely to solve a complete shop layout problem with all the details yet in an efficient way [6]. Therefore, researchers normally make several assumptions

and simplifications in their models without missing the important underplaying structure. Considering the assumptions below, reconfigurable layout is considered to deal with dynamic assembly shop floors in this research.

- An assembly system is a fully automated robotic system. There is no time variation due to unsteady human operations.
- All equipments are 100% reliable except when failure is considered.
- All the robots work continuously until there is no more part to produce.
- Certain places on the shop floor are specified for locating robots.
- There is no space limitation.
- The shape and size of the equipments are not of concern.

It is quite common to consider equal-size area utilization and ignore the shape and size of the workstations for a robotic assembly shop where it is not an unrealistic assumption [16].

Moreover, the source of uncertainty could be a change in product mix/volume, or an unexpected event such as a machine breakdown or an urgent job. A reasonable change in product mix or volume should happen to make it worthy of shop-floor re-layout. However, on the other hand, it would be good enough to find an optimal robot sequence route within the existing layout for an urgent job or finding the best alternative route in case of machine failure. This splits the shop-floor layout problem into two parts: *re-layout* and *find-route*. To deal with the re-layout issue, any meta-heuristic method can be incorporated to find an optimal/near-optimal new layout for the shop floor. However, for the find-route, function blocks are incorporated to suggest the best route among the robots for the new conditions. Figure 13.3 depicts the basic idea of the reconfigurable assembly shop layout problem in this research using both genetic algorithm (GA) and function block (FB).

### 13.3 Shop-Floor Re-Layout Using GA

GA is one of the most commonly used meta-heuristic methods, which does not rely on the analytical properties of the function to be optimized and thus is suitable to deal with a large class of optimization problems [17]. It has been successfully incorporated in the facility layout problem [12, 16, 18]. GA starts with an initial set of random solutions for the problem under consideration. The set of the solutions is also known as the population of chromosomes. A chromosome may be comprised of individuals called genes. The chromosomes are evaluated according to a specified fitness function, and evolve through an iterative process under GA operators to generate new populations. This iteration stops upon satisfying a specific stopping criterion.

In this section, the steps of implementing GA to deal with the re-layout issue of the prototype shown in Fig. 13.3 are presented. Details of this work are explained through an illustrative example.



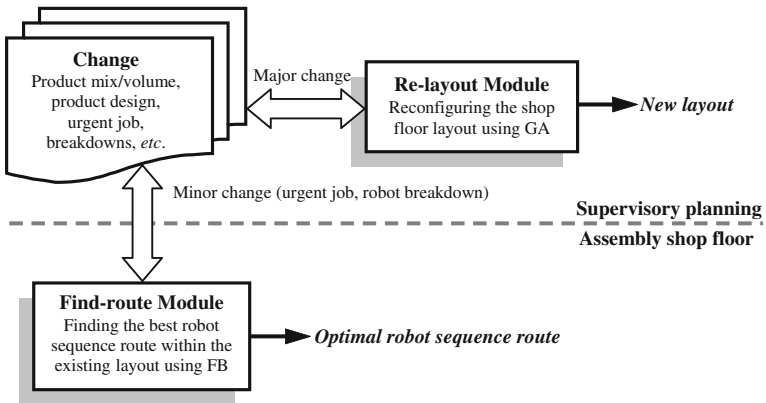


Fig. 13.3 Reconfigurable assembly shop layout approach

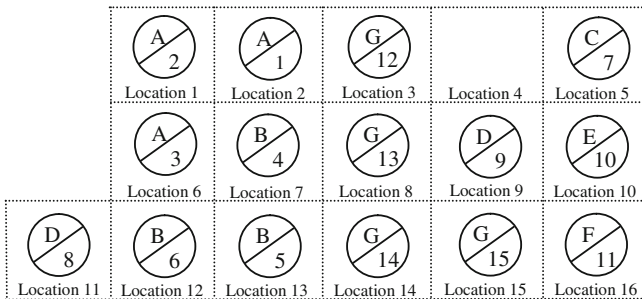


Fig. 13.4 Shop floor layout—letters and numbers inside the circles represent robot types and identities, respectively

### 13.3.1 Chromosome Representation

GA requires a representation scheme for the chromosomes. This is also known as chromosome encoding. In this study, a form of direct string representation is used as chromosome. Each chromosome has as many genes as the number of the locations assigned to the robots on a shop floor. To illustrate this, an example of a shop-floor layout with 15 robots distributed over 16 designated locations is depicted in Fig. 13.4, which is simplified according to the assumptions in Sect. 13.2. The encoded chromosome representation for this shop-floor layout is (2 1 12 0 7 3 4 13 9 10 8 6 5 14 15 11), where 0 indicates an empty location.

### 13.3.2 GA Operators

Four genetic operators are incorporated in this research: cut-and-paste, crossover, mutation and reproduction. These genetic operators and their operation results are

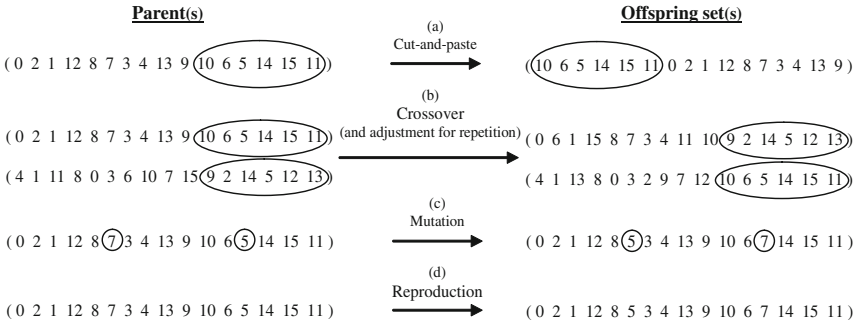


Fig. 13.5 Genetic operators

shown in Fig. 13.5. The chromosomes are selected for the genetic operators based on their fitness.

In order to select parent chromosomes for implementing crossover operator, the Roulette wheel selection technique is applied. In this research, a genetic operator combining cut-and-paste and simple crossover is employed to increase the possibility of examining more variety of different shop-floor layouts. First, a random number  $r \leq 16$  is generated for each parent separately. The genes whose ranks in the chromosome string are greater than  $r$  are cut-and-paste to the beginning of the chromosome as shown in Fig. 13.5a. Then, another random number  $r' \leq 16$  is generated for the implementation of simple crossover. As shown in Fig. 13.5b, the genes whose ranks are greater than  $r'$  in the parents' chromosome strings are swapped. Finally, a backward replacement procedure is carried out to eliminate the repeated genes outside the cutting section and retrieve the missing genes instead of them in the offspring chromosomes. The rest of the chromosomes remain identical to their parents.

The probability of applying mutation over chromosomes (mutation rate) within a population is usually a small number; otherwise, the algorithm may not be able to converge. The chromosome is chosen randomly and the mutation operator selects two random genes of the chromosome to swap their positions. Mutation helps to increase the searching power by avoiding premature convergence and escaping from local optima where reproduction or crossover may not produce a good solution to the problem. The procedure is terminated when the number of generations is reached to a predetermined value.

### 13.3.3 Fitness Function

The fitness function to evaluate a solution is defined as follows:

$$\text{Fitness function} = \sum_{j=1}^M \left( \sum_{i=1}^N C_{i,i+1} d_{i,i+1} f_{i,i+1} + R_i d_r \right)_j, \quad (13.1)$$

where

$d_r = 0$	if robot $i$ is not relocated
$R$	robot relocation cost
$f$	product flow
$d$	the distance between two robots $i$ and $i+1$
$C$	material handling cost per unit distance
$N$	number of robots required to assemble a product
$M$	total number of products

### 13.3.4 Searching Algorithm

The genetic search process used in this research is as follows. This is also depicted in the flowchart as shown in Fig. 13.6.

1. Randomly generate the initial set of chromosomes with a predetermined population size.
2. Evaluate the cost of each chromosome in the population according to the fitness function.
3. Calculate the average fitness of the population.
4. Use the elitist strategy, i.e., keep the potentially best chromosomes to the following generation by eliminating the same number of the worst members of the population. For the sake of computation and efficiency, the population size is kept constant. Any chromosome whose fitness in proportion to the average fitness of the population is greater than a pre-specified value is omitted, and the same number of the best chromosomes with the lowest costs gets reproduced to the next generation.
5. Apply the Roulette wheel selection technique to select the parent chromosomes from the current population. Implement the cut-and-paste and the simple crossover operators as explained in Sect. 13.3.2 to generate the new population.
6. Apply the mutation operator based on the mutation rate.
7. The rest of the chromosomes are reproduced from the current population to the new generation.
8. Check the termination condition. Stop if the number of iterations reaches the pre-specified value. Otherwise, proceed to the next generation and go back to step 2.

## 13.4 FB-Enabled Assembly Routing Planning

As mentioned in Sect. 13.2, a significant change in product volume/mix should occur to trigger the re-layout of an assembly shop floor. However, in case of an urgent job arrival or a robot breakdown, it may not be necessary to change the whole shop-floor layout. Defining an urgent job as a product with main assembly route A–B–A–D and secondary assembly route F–G–A as shown in Fig. 13.7, an

Fig. 13.6 GA procedures

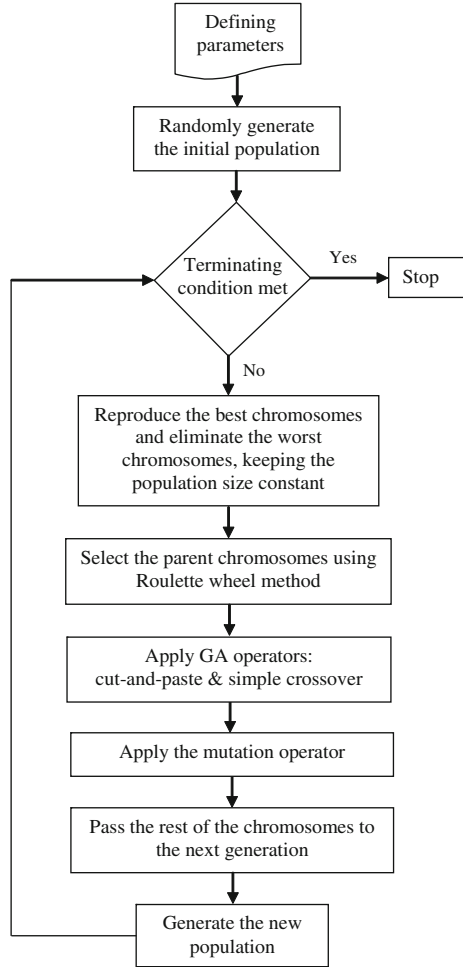
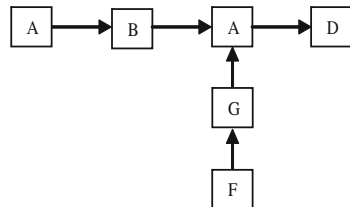
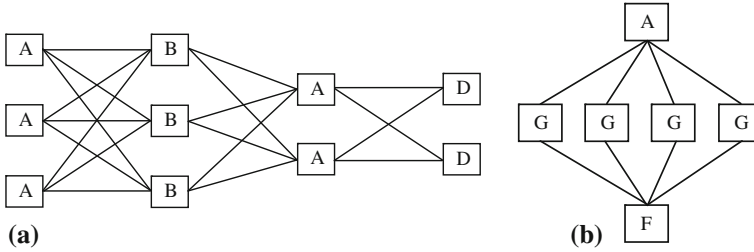


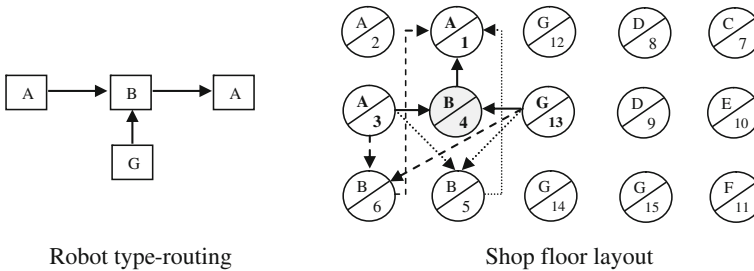
Fig. 13.7 Robot type-routing of an urgent job, with the main assembly route of A–B–A–D, and the secondary assembly route of F–G–A



optimal/near-optimal sequence of robots to visit (or assembly routing) can be found within the existing shop layout without carrying out any actual layout reconfiguration. If there are  $i$  robots of type A,  $j$  robots of type B and  $k$  robots of type A, there are  $p = i \times j \times (i - 1) \times k$  possibilities of robot sequences for the



**Fig. 13.8** Possible assembly routings for the urgent job shown in Fig. 13.7 on the shop floor of Fig. 13.4. **a** Main assembly routing, **b** secondary assembly routing



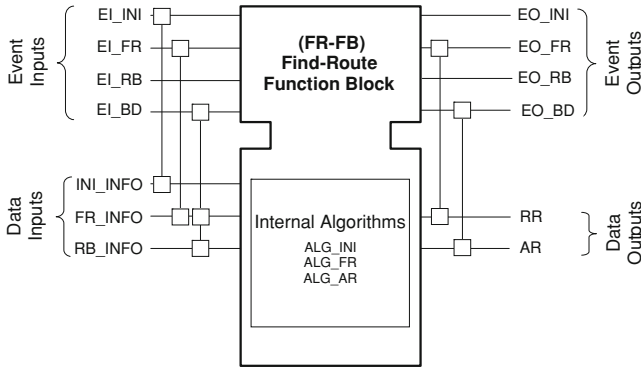
**Fig. 13.9** Alternative assembly routings on the shop floor in case of the failure of robot 4 of type B

main assembly line. The optimal/near-optimal robot sequence in terms of material handling cost should be identified among these possible routings. The same process needs to be repeated for the secondary assembly route. If there are  $n$  robots of type F and  $m$  robots of type G, there are  $q = m \times n$  possibilities of robot sequences for the secondary assembly route. Therefore, there are  $q \times p$  possible assembly routings in total for this product. These possibilities are shown in Fig. 13.8.

To select the optimal yet feasible route among the possible ones, the evaluation criteria must also consider the time delay due to in-process productions involving the robots of the same types. By allocating a designated FB to each robot, the FB can be used to find the optimal robot route on the shop floor by means of the FB's embedded algorithms. The FB can also be incorporated to find the best alternative robot/route in case of robot breakdown (Fig. 13.9).

### 13.5 Designing a Find-route Function Block

This section extends the authors' previous research [15] on incorporating FB methodology for dynamic assembly planning and control so as to increase a system's adaptability to dynamic changes. Particularly, a new FB is designed and



**Fig. 13.10** The find-route function block FR-FB

added to the existing FB network for each robot. It deals with the find-route of the prototype as shown in Fig. 13.3. The new FB, named *find-route function block* (FR-FB), is depicted in Fig. 13.10. The embedded algorithms in this function block are responsible for the issues explained in Sect. 13.4, i.e.:

- Finding a robot sequence within the existing layout based on the robot type-routing of an urgent job.
- Finding the best alternative robot in case of a robot breakdown.

In what follows, the details of the input, output and internal algorithms of this function block are explained.

### 13.5.1 Input

The input data to an FR-FB are: INI\_INFO, FR\_INFO and RB\_INFO.

INI\_INFO (initialization information) is coupled with EI\_INI, an initialization event, to receive the latest updates and information from the upper-level supervisory planning, including:

- Robot’s location.
- Out-of-order devices.
- Material handling cost per unit distance.
- Type of material handling equipment between this robot and other robots.
- Speed of material handling equipments.

Upon receiving all the initialization information, an event output EO\_INI is fired to signal the completion of the initialization process.

FR\_INFO (find-route information), coupled with the event input EI\_FR, contains the following information used by the find-route algorithm ALG\_FR:

- Robot sequence to assemble a product (main or secondary assembly route).
- Product flow.
- Type of material handling equipment receiving at this robot.
- Time left to complete the current job by the robot.

In case of a robot breakdown, the situation is informed to the function block by the input event EI\_RB, which in turn triggers the output event EO\_RB broadcasting the situation to other robots of the same type, as well as the previous and the next robots in the routing. Any reply is received by the FR-FB via event EI\_BD together with the RB\_INFO data input.

RB\_INFO (robot breakdown information), used by the alternate-route algorithm ALG\_AR, includes:

- Time left on the same-type robot to complete its current job.
- The robots' locations.
- The type of material handling equipment to these robots.

### ***13.5.2 Output***

On the other side, the output data, namely RR and AR, are produced by the embedded algorithms ALG\_FR and ALG\_AR, respectively.

RR (robot routing):

- Accumulated material handling cost.
- A string representing the robot sequence (see [Sect. 13.5.3](#) for details).
- Suggesting the best sequence of robots if the robot is the last one in the chain.

AR (alternative route):

- Evaluation of the routing with other robots in case of breakdown.
- Suggestion of the best alternative robot for substitution.

Upon completion of algorithms ALG\_FR and ALG\_AR, output events EO\_FR and EO\_BD are fired, respectively. Details of the two algorithms embedded in an FR-FB are described in the next section.

### ***13.5.3 Embedded Algorithms***

ALG\_FR (find-route algorithm) is responsible for finding the best robot sequence among many possible ones for an urgent job as illustrated in [Fig. 13.8](#). The routing evaluation is divided into the main and the secondary assembly routings. A double-row string is utilized to represent the types and IDs of the robots to form a robot sequence. This string is passed to the robots for processing, starting from

Out of order/in-use robots					Robot type and ID								
R1	R2	R3	R4	R5	A	B	A	D	$T_{max}$	CT1	CT2	MHC	TC
0	0	0	0	0									

**(a)**

R1	R2	R3	R4	R5	F	G	A	$T_{max}$	CT1	CT2	MHC	TC
0	0	0	0	0								

**(b)**

**Fig. 13.11** Function block’s routing strings: **a** main assembly routing, and **b** secondary assembly routing of Fig. 13.8

the first robot type in the string. The cost evaluation is based on Eqs. 13.2–13.6. Since different type of material handling equipment (MHE) serves in different routes between the robots, Eqs. 13.3–13.5 are defined to consider the cost of time due to the different possible combinations of MHEs and robots.

$$MHC = C_{j-1,j}d_{j-1,j}f_{j-1,j}, \tag{13.2}$$

$$\text{Cost of Time (CT)} = CT1 + CT2 \tag{13.3}$$

$$CT1 = L \times T_{max}, \quad \text{where } T_{max} = \max \{ T_j \} \text{ and } j \in [1, N] \tag{13.4}$$

$$CT2 = L \times \sum_{j=2}^N \frac{d_{j-1,j}f_{j-1,j}}{v_{j-1,j}} \tag{13.5}$$

$$\text{Total Cost (TC)} = MHC + CT \tag{13.6}$$

where

$L$  cost per unit time.

$N$  number of robots required for assembling the product.

$T$  time left on robot  $j$  to finish its in-process job.

$d$  the distance between two robots  $j - 1$  and  $j$ .

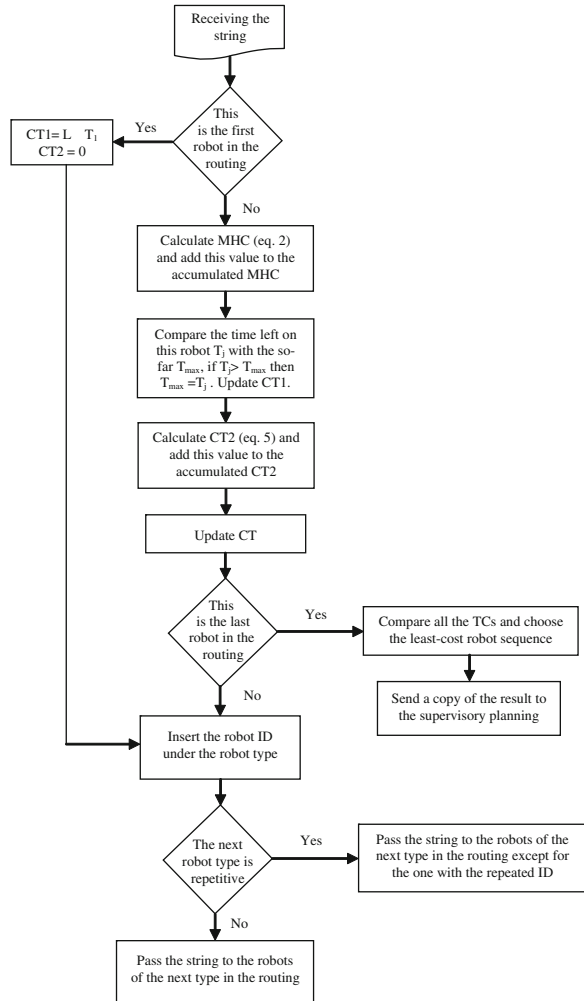
$v$  speed of the MHE receiving by robot  $j$  from robot  $j - 1$ .

As shown in Fig. 13.11, each FR-FB receives the robot routing, ID numbers of the previous robots, accumulated MHC, accumulated CT2, CT1 and TC up to this point for the main and secondary assembly routings of a product. It then adds its ID to the string under its type and calculates the robot’s associated MHC, CT1 and CT2 and updates these values in the string. The algorithm eliminates repetitive robot IDs to avoid using the same robot for the evaluation. It also checks the ID of the out-of-order robots to ignore them for the evaluation. The last robot in the chain also compares the total costs and chooses the best sequence of robots with the least cost. Figure 13.12 depicts the procedure of the algorithm.

ALG\_AR (alternate-route algorithm) is the next algorithm embedded in the FR-FB. It is used to find the best alternative robot in case of a robot breakdown. It communicates with the previous and the next robots as well as other robots of



**Fig. 13.12** Procedures of find-route algorithm

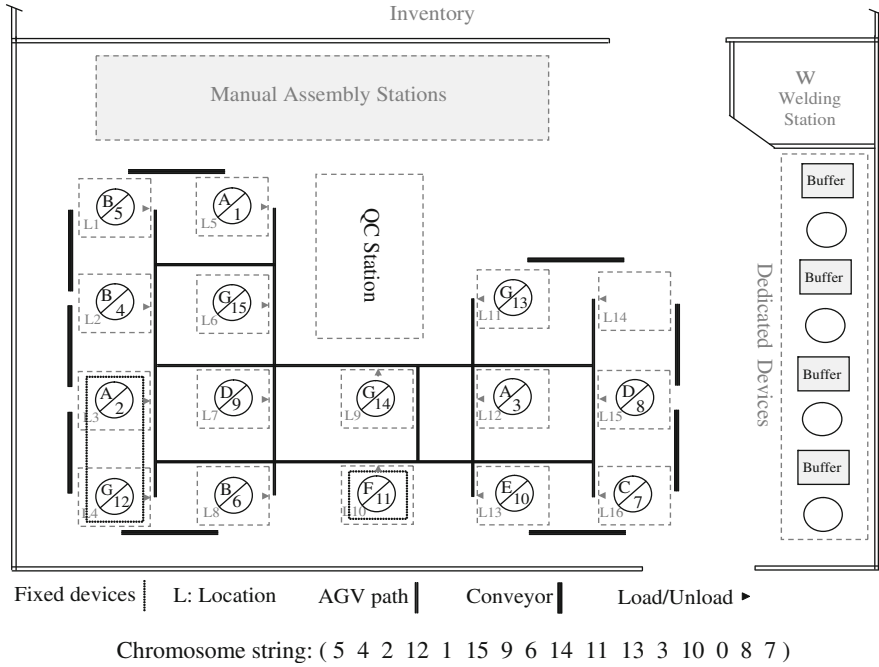


the same type, and then calculates the total cost including MHC for a robot sequence consisting of the previous robot, the alternative robot and the next robot in the routing.

### 13.6 Case Study

The input data required for a facility layout study is usually comprised of:

1. Set of products.
2. Operation sequence for each product.
3. Production quantity for each product.



**Fig. 13.13** Example of an assembly shop floor

Figure 13.13 shows a hypothetical but typical example of a job-shop assembly environment. The chromosome string of this layout is shown in the figure as well. Table 13.2 presents the current product routings for this shop floor including the new products (shaded). In the supervisory planning, according to the operation sequence and based on the capacity and the capability of the robots, the sequence of robot types required for the operations have been specified and presented as robot routings in the table.

Incorporating the elitist strategy during GA implementation, the chromosomes with a fitness value more than 1.5 times of the generation’s average cost are replaced with the same number of the chromosomes with the least costs. Other parameters used in the GA are as follows.

- Conveyor:  $C = 0.1$  (cost unit)/(distance unit).
- AGV:  $C = 1$  (cost unit)/(distance unit).
- Relocation:  $R = 0.75$  (cost unit)/(distance unit).
- Initial population: 150 (chromosomes).
- Mutation rate: 1/150.
- Termination criterion: 800 (generations).

The output of GA calculation is presented in Fig. 13.14 with two best layouts. Choosing solution 1 as shown in the figure, the optimized chromosome string resulting from the re-layout module is decoded in Fig. 13.15.

**Table 13.2** Product routings (new products are shaded)

Product	Robot routings	Product demand
1	A → B → A ↑ G	54
2	B → D → C	75
3	G → F E → F	81
4	A	69
5	B → G → D ↑ F	90
6	A → G → D → B ↑ E ↑ C	77
7	A → G	100
8	E → G → A ↑ ↑ B D	70

The urgent job of Fig. 13.7 is considered to be carried out within this new layout. Table 13.3 shows the current situation of the robot types involved in the robot type-routing of the urgent job. The total time left on a robot to finish producing the in-process product is shown as well.

As discussed in Sect. 13.2, the find-route module is in charge of finding the robot sequence for the urgent job by means of FR-FB as well as finding the best robot to substitute the robot that breaks down during production. The two embedded algorithms of this function block are developed in MATLAB®, which deals with the following two cases for the same example:

1. Finding the best robot sequence for the routing of Fig. 13.7 in the layout of Fig. 13.15.
2. Finding the best robot alternative in case of breakdown of robot G.

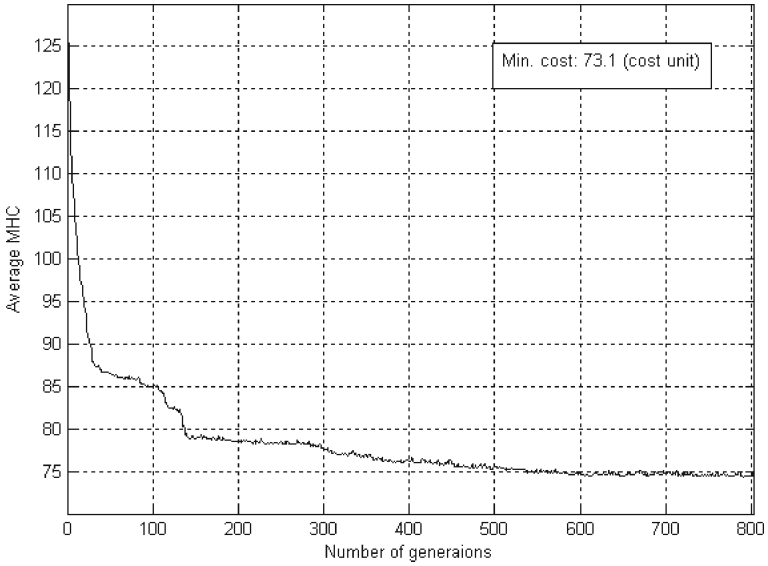
The results are calculated using the following parameters.

Conveyor:  $v = 0.2$  (distance unit)/(time unit).

AGV:  $v = 0.3$  (distance unit)/(time unit).

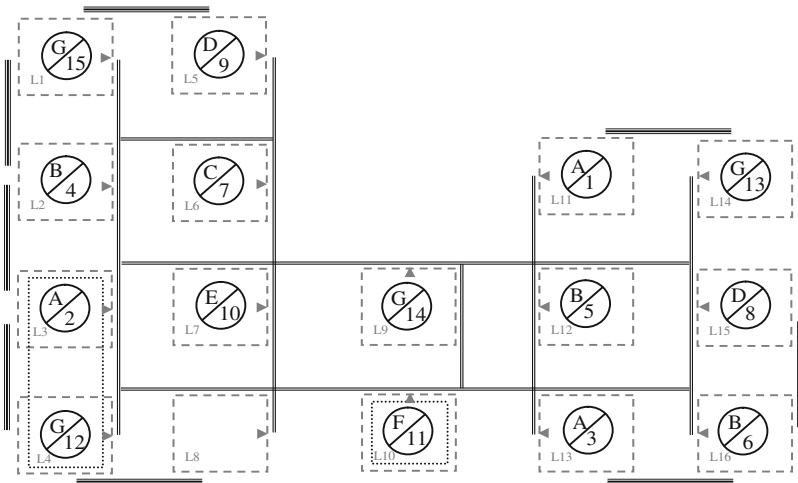
Cost per unit time:  $L = 0.15$  (cost unit)/(time unit).

*Case 1* The first algorithm ALG\_FR searches for the optimal sequence of robots for the main assembly routing, i.e., A–B–A–D. As shown in Fig. 13.16a, the least-cost robot sequence is found as 3–5–1–8, with the total cost of 196.6 (unit cost).



**Solution 1:** (15 4 2 12 9 7 10 0 14 11 1 5 3 13 8 6)  
**Solution 2:** (15 9 2 12 1 4 3 0 14 11 5 10 7 6 13 8)

**Fig. 13.14** GA results of assembly shop floor re-layout at a cost of 73.1 cost unit



**Fig. 13.15** Optimized assembly layout

For the secondary assembly routing, i.e., F–G–A, where the robot of type A is already chosen as in the main assembly routing (robot 1), the least-cost robot sequence is 11–14–1 as shown in Fig. 13.16b with the total cost of 92 (unit cost).

**Table 13.3** Remaining time on robots involved in robot type-routing of the urgent job

Robot type	Robot ID	Time left on in-process job (h)	In-use/ out-of-order
A	1	5	
	2	8	×
	3	5	
	4	5	
B	5	6	
	6	6	
D	8	6	
	9	7	
	12	10	×
G	13	5	
	14	7	
	15	6	
F	11	7	

R1	R2	R3	R4	R5	A	B	A	D	T <sub>max</sub>	CT1	CT2	MHC	TC
2	12	0	0	0	3	5	1	8	5	50	90.6	56	196.6

(a)

R1	R2	R3	R4	R5	F	G	A	T <sub>max</sub>	CT1	CT2	MHC	TC
2	12	0	0	0	11	14	1	7	70	11	11	92

(b)

**Fig. 13.16** Search result for case 1. **a** Main assembly routing, **b** secondary assembly routing

R1	R2	R3	R4	R5	F	G	A	T <sub>max</sub>	CT1	CT2	MHC	TC
2	12	0	0	0	11	13	1	7	70	21	15.4	116.4

**Fig. 13.17** Search result for case 2

*Case 2* The second embedded algorithm ALG\_AR is triggered to deal with the robot breakdown. Assuming a failure of robot G in this example, the best alternative robot for substitution is found to be robot 13 with a total cost of 116.4 (Fig. 13.17).

### 13.7 Conclusions

This chapter presents a hybrid approach for the assembly shop-floor layout problem by incorporating both genetic algorithm and function blocks. This combined approach is particularly useful for a turbulent job-shop assembly environment where function blocks can deal with configuration changes due to dynamic operations. It consists of two different modules: (1) re-layout module, and

(2) find-route module. The former deals with major shop-floor changes and derives an alternative layout using GA when the reconfiguration cost can be properly justified against material handling cost, etc., whereas the latter utilizes a specialized function block to deal with soft changes, such as an urgent job and/or a robot breakdown, when alternative assembly routes can be determined within the existing layout.

The novelty of this approach is incorporating the function block methodology in finding alternative assembly routes where the quantity of the assembly operations is low and the unavailability of a robot is of temporary nature. In addition to GA-based global optimization of shop-floor re-layout, the FB-based methodology applies embedded algorithms for proactive and adaptive decision-making according to the current assembly plan and assembly sequence. As function blocks can also be used for process control, it is possible to make decision at run-time and continue the current assembly operation with the least interruption but using alternative resource in alternative route.

For proof of concept, the proposed method has been implemented and tested through a case study in the simulated environment of MATLAB. As demonstrated by the case study, the hybrid approach can enhance the adaptability of an assembly shop against disturbances by providing (near) optimal layout and routing solutions effectively. It is expected that this approach can also contribute to factory automation in the next-generation adaptive manufacturing systems. Conducting real-world tests to further validate this approach is the future work of this study.

## References

1. Raman, D., Nagaliangam, S. V., & Lin, G. C. I. (2009). Towards measuring the effectiveness of a facilities layout. *Robotics Computer-Integrated Manufacturing*, 25, 191–203.
2. Webster, D. B., & Tyberghein, M. B. (1980). Measuring flexibility of job-shop layouts. *International Journal of Production Research*, 18(1), 21–29.
3. Kulturel-Konak, S. (2007). Approaches to uncertainties in facility layout problems: perspective at the beginning of 21st century. *Journal of Intelligent Manufacturing*, 18, 237–284.
4. Benjafar, S., Heragu, S. S., & Irani, S. A. (2002). Next generation facility layout: research challenges and recent progress. *Interfaces*, 32(6), 58–76.
5. Meng, G., Heragu, S. S., & Zijm, H. (2004). Reconfigurable layout problem. *International Journal of Production Research*, 42(22), 4709–4729.
6. Irani, S.A., & Huang, H. (1998). Layout modules: a novel extension of hybrid cellular layout. *Proceedings of 1998 ASME International Engineering Congress and Exposition*, Anaheim, CA, November 15–20
7. Parveen, P., Chowdary, B. V., Deshmukh, S. G., & Prasant, P. (2009). A new approach for formation of virtual cells. *International Journal of Manufacturing Research*, 4(2), 171–188.
8. Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: a survey. *Annual Review in Control*, 31, 255–267.
9. Rosenblatt, M. J. (1986). The dynamics of plant layout. *Management Science*, 32(1), 76–85.
10. Urban, T. L. (1993). A heuristics for the dynamic facility layout problem. *IIE Transactions*, 24(4), 57–62.

11. El-Baz, M. A. (2004). A genetic algorithm for facility layout problems of different manufacturing environment. *Computers and Industrial Engineering*, 47, 233–246.
12. Jain, P., & Sharma, P.K. (2005). Solving job shop layout problem using ant colony optimization technique. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp. 288–292
13. Defersha, F., & Chen, M. (2009). A simulated annealing algorithm for dynamic system reconfiguration and production planning in cellular manufacturing. *International Journal of Manufacturing Technology and Management*, 17(1–2), 103–124.
14. Qin, Y.F., & Zhao, M.Y. (2004). Research on optimization method for hybrid assembly line design. *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision*, pp. 509–514
15. Wang, L., Keshavarzmanesh, S., & Feng, H.-Y. (2008). Design of adaptive function blocks for dynamic assembly planning and control. *Journal of Manufacturing Systems*, 27(1), 45–51.
16. Wang, M. J., Michael, H. H., & Meei, Y. K. (2004). A solution to unequal area facilities layout problem by genetic algorithm. *Computers in Industry*, 56, 207–220.
17. Talbi, El-Ghazali. (2009). *Metaheuristics, from design to implementation*. USA: Wiley.
18. Islier, A. A. (1998). A genetic algorithm approach for multiple criteria facility layout design. *International Journal of Production Research*, 36(6), 1549–1569.

# Chapter 14

## A Simulation Optimisation Framework for Container Terminal Layout Design

Loo Hay Lee, Ek Peng Chew, Kee Hui Chua, Zhuo Sun and Lu Zhen

**Abstract** Port designers are facing challenges in choosing appropriate terminal layouts to maximise operational efficiencies. This study aims to address this problem by providing a simulation optimisation framework for container terminal layout design. This framework consists of three main modules which are automated layout generator (ALG), the multi-objective optimal computing budget allocation (MOCBA) algorithm and the genetic algorithm (GA). ALG is to automatically generate a simulation model for a set of given design parameters; MOCBA is to intelligently determine the simulation replications to different designs for identifying promising designs; GA is to help generate new design parameters for optimisation. Numerical examples are used to demonstrate the applicability of this framework.

---

L. H. Lee (✉) · E. P. Chew · K. H. Chua · L. Zhen

Department of Industrial and Systems Engineering, National University of Singapore,  
10 Kent Ridge Crescent, Singapore, Singapore  
e-mail: iselee@nus.edu.sg

E. P. Chew  
e-mail: isecep@nus.edu.sg

K. H. Chua  
e-mail: u0508477@nus.edu.sg

L. Zhen  
e-mail: isezi@nus.edu.sg

Z. Sun  
Centre for Maritime Studies, National University of Singapore,  
12 Prince George's Park, Singapore, Singapore  
e-mail: sunzhuo@nus.edu.sg



## 14.1 Introduction

In the past decades, the container shipping industry has been growing rapidly. In order to capture this growing market, many governments and private operators increase their investment on port infrastructures, e.g., building new terminals, employing new port technologies or enlarging the size of the existing terminals. This has resulted in intense competition among these ports. Hence it is important for these port operators to improve the efficiency of container terminal operations, especially when they set up their new terminals.

When the port operators decide to build a new terminal, they need to first decide the layout skeleton (e.g., horizontal stack layout versus vertical stack layout) and the operation logic that governs the processes in this new terminal (e.g., how vessels are going to moor given limited number of available berths). There are some design parameters in these layouts and operation logics need to be determined, and usually port operators will choose few different sets of these parameters based on their experiences, for examples the number of blocks, the dimension of each block, the number of resources used, such as quay cranes, yard cranes and vehicles. Then, they will develop simulation models for this new terminal by using some simulation software (e.g., Automod and eM-Plant). After that, they will run the simulation models with these few sets of parameters to determine which set of design parameters they should choose based on the estimated performance. Simulation models are used in this case because they can capture many real constraints and uncertainties in ports which analytical models cannot do.

Simulation is a powerful tool and is often used to evaluate alternative designs and explore possibilities. A vast variety of research topics have utilised simulation to improve port processes. Steenken et al. [1] and Vis and Koster [2] both gave a comprehensive review and classification of the current research and future direction on the container terminal operation. Many research works are focusing on port-related decision support systems. Kozan [3] conducted a comparison between the analytical and simulation planning models for a container terminal. Bruzzone et al. [4] showed the advantages and effectiveness of simulation approach for managing complex container port. By using simulation models, Yun and Choi [5] analysed the performance of a container terminal system in Pusan; Nam et al. [6] determined the optimal number of berths and quay cranes used in a terminal in Pusan; Shabayek and Yeung [7] predicted the performance of the operations in a terminal in Hong Kong; Sgouridis et al. [8] analysed the inbound container handling in the “All-Straddle-Carrier” system; Yang [9] analysed the effect of the increase in the number of automated lifting vehicles on the productivity of the terminal.

All the above works assumed a fixed layout to work on, and do not consider a large number of design alternatives. If the port operators want to use the simulation approach in finding an optimal port layout design they will face three challenges. First, it is not easy to modify the simulation models when the design parameters are changed. For example, when the number of blocks or the width

and the length of the block are changed, we need to redraw the layout and possibly to redefine the logic used in the model [10]. This may require a substantial amount of time and effort to create these new simulation models. Second, in order to estimate an accurate performance through simulation, we usually need to run the simulation with many replications. This means that the simulation time can be quite long. Third, there might be a lot of possible design parameters, and to enumerate all of them might be computationally infeasible. Due to these reasons, the port operators can only test a limited set of design parameters.

To improve the port operators' layout design capability, we need to address the above three challenges. For the first challenge, we should develop a program that can take any design parameters and easily create simulation models without any human intervention. The second and third challenges belong to the domain of simulation optimisation. Simulation optimisation is defined as the process of finding the best values of some decision variables for a system where the performance is evaluated based on the output of a simulation model for this system [11]. Fu et al. [12] has given a comprehensive review on the approaches used in simulation optimisation. Many real life problems can be solved using simulation optimisation technique, such as inventory control problems [13], cross-docking problems [14] and aircraft spare part problems [15]. For multi-objective simulation optimisation problems, Lee et al. [16] propose a framework which integrates the multi-objective computing budget allocation algorithm (MOCBA) [17] with the search method, where MOCBA can handle the second challenge while the search method can deal with the third challenge. In multi-objective optimisation problems, we might not be able to find a single best solution (or design) that simultaneously optimises all the objectives. In this case, we may want to find solutions for which their objectives have been optimised to the extent that if we try to optimise a subset of these objectives any further, then the remaining objective(s) will become worse. These designs are called non-dominated designs or Pareto designs. Hence, for multi-objective problems, instead of finding a unique single best solution to the problem, we aim to find a set of non-dominated designs. This set is also known as Pareto set. MOCBA aims to allocate simulation runs effectively to all design alternatives so as to maximise the probability to correctly identify the Pareto set. However, MOCBA only deals with a finite number of design alternatives. If we want to explore a larger feasible space, we need to integrate it with a search method, and Lee et al. [16] provide a framework on how this can be done.

In this study, we will develop a simulation optimisation framework based on Lee et al. [16] to solve the port layout design problem. The port layout design problem is defined as follows: given a fixed dimension of land space, we want to find promising port layout design parameters which consider few objectives at the same time. The objectives will be the performance measures determined by port operators which might include quay crane productivity, vehicle utilisation, vessel turnaround time, etc.

In this simulation optimisation framework, we will first create an automated layout generator (ALG) which automatically generates a new simulation model

given a set of design parameters. Then a simulation optimisation algorithm that combines the genetic algorithm (GA) and MOCBA is developed. GA is to generate new design parameters for optimisation while MOCBA attempts to efficiently allocate computing resources to different designs to identify the Pareto optimal design parameters.

In Sect. 14.2, we will describe the simulation optimisation framework. Section 14.3 will provide the numerical experiments. Finally, we give conclusions in Sect. 14.4.

## 14.2 Simulation Optimisation Framework

This study is motivated by the actual port layout design problem faced by port operators. Generally, it is very time-consuming to find an optimal design for container terminals. Port operators usually choose a few sets of design parameters according to their experience, and then evaluate them through simulation to identify the promising design. Our aim here is to automate this process and help port operators to explore more design alternatives so that they can select better designs.

### 14.2.1 General Framework

Simulation optimisation is the process of finding the best values of some decision variables for a system where the performance is evaluated based on the output of a simulation model for this system [11]. In this chapter, a simulation optimisation platform is developed for facilitating port layout design. The general framework of the platform is illustrated in Fig. 14.1.

The framework comprises of two main modules: an ALG and a simulation optimisation module. The ALG module is developed to allow the port designer to change the port simulation model parameters and generate multiple designs. The simulation optimisation module is developed to address two main issues in the port design process. First, the number of possible design alternatives grows exponentially as more parameters are considered or when the range of possible values for parameters increases. This means that the search space for all possible designs can be very large, which makes it computationally infeasible or costly to do an exhaustive search. Second, instead of only one single performance measure, users may require a port design that optimises more than one performance measure.

The general flow of the framework is as follows.

Port operators will first decide the performance measure they want to optimise for the port. They should define the port design space, which includes (1) port parameters i.e., the number of parameters to consider and their ranges; (2) port operation logic; (3) port layout skeleton. Based on the information from the

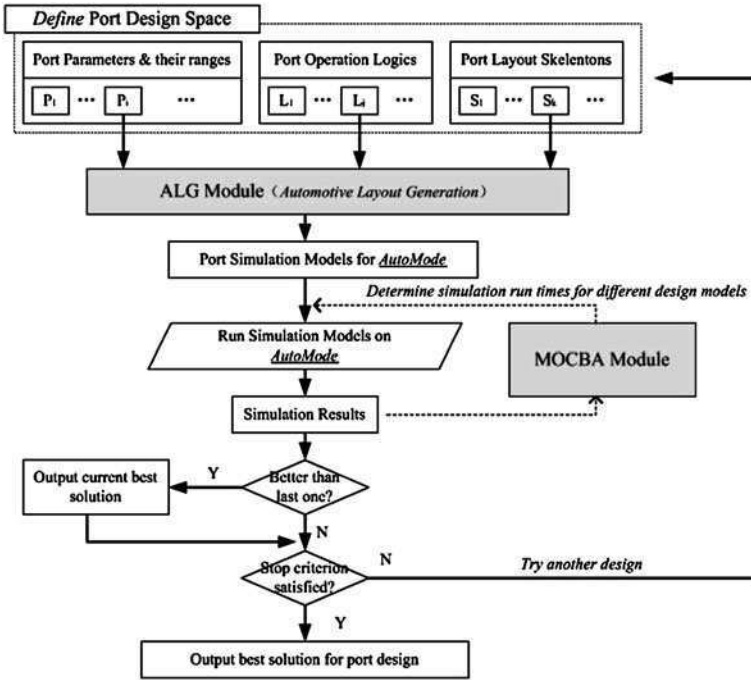


Fig. 14.1 The flowchart of the simulation optimisation for port layout design

port design space, the simulation optimisation module will first generate an initial set of design parameters. The ALG module of the program will subsequently generate a simulation model for each of these design parameters. In this study, Automod is chosen as the simulation tool. The simulation optimisation module will allocate the computing budget (or number of simulation runs) among the different simulation models (or different design alternatives). Based on the simulation outputs, it will identify all the non-dominated designs and will be stored in the elite set. Fitness values are computed for all the designs based on the probability of non-dominating which can be obtained when we run MOCBA algorithm. GA will use these fitness values to generate a new population which consists of different sets of design parameters. These new sets of design parameters are then fed into the ALG module to generate new simulation models. Then simulation optimisation module will determine the number of simulation replications to be allocated to each design and also identify non-dominated designs from this new population. These non-dominated designs will then be placed in the elite set. This process will repeat until a stopping criterion is met. Eventually we will run the MOCBA algorithm again on the elite set to get the final non-dominated designs from the same set. These designs will be candidates for port operators to select their designs from.

The following sections will introduce the key modules contained in the proposed framework.

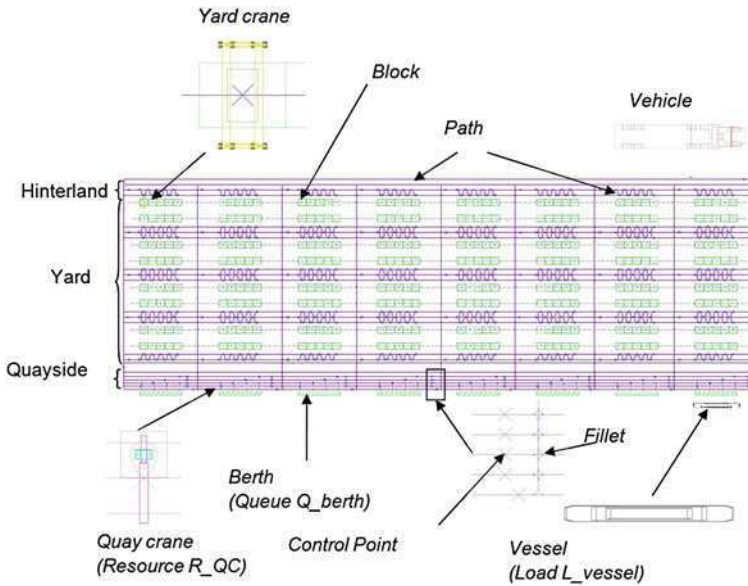


Fig. 14.2 An example of port layout model built in Automod

### 14.2.2 Automod Model of Port Layout

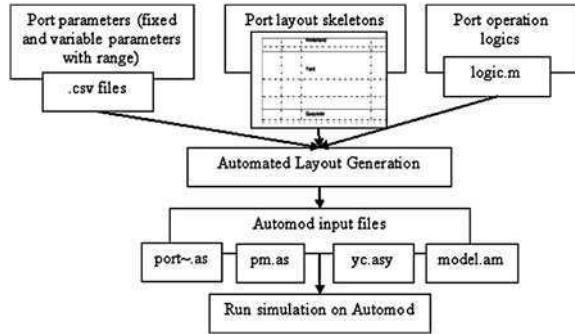
As mentioned earlier, a library of port skeleton layouts and port logic needs to be defined. In the framework that we develop, only one port skeleton layout and one port logic are defined. It is envisioned that additional port skeleton layouts and port operation logic can be added on at a later time without making major changes to the framework. The current Automod base model of port layout is described as follows:

This is the base model of the port whose modifications will be carried out by the ALG automatically to reflect the user’s requirements. In this type of layout, the containers are stacked horizontally and parallel to the berth (Fig. 14.2), a popular layout mainly used in Asian ports such as port Bander Abbas in Iran and the Jebel Ali port of Dubai.

In the simulation of port, users are interested in the long run performance of the system. This type of simulation is more commonly known as steady-state simulations. Autostat is a complementary software tool of Automod used to study the simulation runs and determine the appropriate warm-up period. However, given that the proposed framework is to be fully automated, conducting warm-up analysis manually using Autostat is not appropriate. Therefore, we develop an automated warm-up analysis algorithm in the logic file of Automod model.

In selecting the warm-up analysis method, it is decided to choose the method mainly based on two criteria: ease of automation and computer time taken. Ease of automation is necessary as the method chosen must not require any additional

**Fig. 14.3** Framework of the ALG module



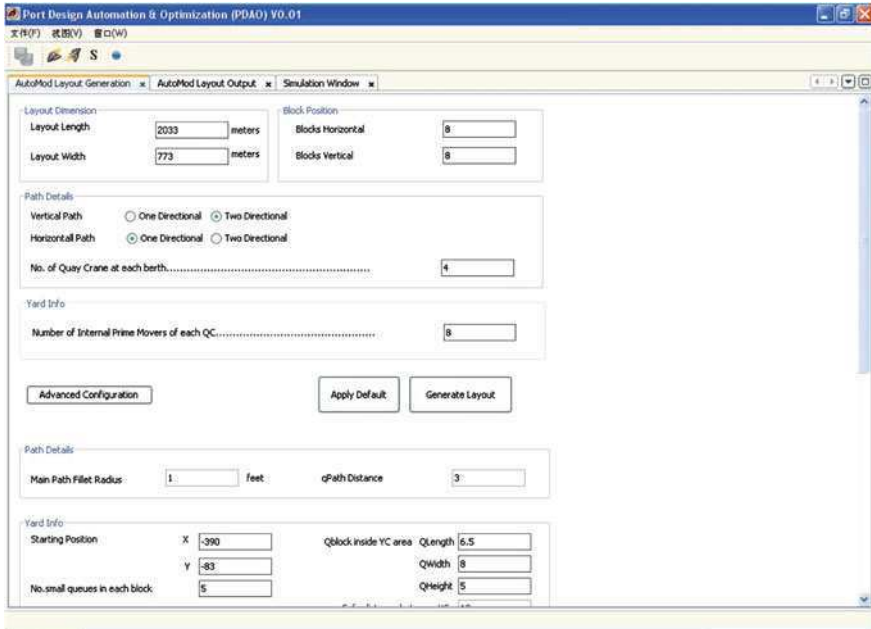
human intervention during the runs, while computer time taken must be minimised as a lot of simulation models with different parameters will be run, each requiring its own warm-up analysis. It is noted that the warm-up analysis method chosen may not be the best available but it serves our objectives in developing the framework and can be easily changed in the future.

### 14.2.3 Automated Layout Generation

As mentioned earlier, many simulation programs do not have the capability of adjusting the positions of the layout when any of the parameters needs to be changed. The ALG module (Fig. 14.3) is developed to ease human efforts in changing parameters of the simulation model. The ALG program can effectively convert user desired specifications into Automod system files to build a new Automod model based on the skeleton layout and the port operation logic that are developed earlier.

First the initial design parameters are stored in a “.csv” input file (Microsoft office excel comma separated values file). Some of the parameters in the input file are fixed while others can be varied. Parameters that can be varied (which define the difference in designs) will be generated externally and then will be written in the input file while keeping the fixed parameters constant. Upon reading the information from the input file, the ALG module will generate several Automod model files, which currently are port ~.asy, pm.asy and yc.asy. The first file is the main process system which defines the simulation logic, resource definition, etc., while the latter two files define the path mover systems for the prime movers and the yard cranes, respectively. The last thing that needs to be done is to generate the model.amo file, which is the executable file for Automod models. The random number set is changed in the model.amo file. With these system files, the Automod model is ready.

With the ALG module as shown in Fig. 14.4, the Automod system files can be created in a few seconds. Changing some design parameters and creating a new simulation model becomes far more efficient than the manual process. Current features of ALG include:



**Fig. 14.4** Interface for port layout generation

1. The ALG code is divided into sections corresponding to each system file that the Automod model needs. The developer can locate parts of ALG to make changes easily if there are any changes to the Automod base model logic or layout.
2. All variables used in the different sections of ALG are located in the start of the sections. This allows the developer to access the variables they need easily.
3. All the parameters pertaining to port layout or operation logic are coded as variables, allowing the developer to choose the variables which the user may want to vary.

We have developed an interface for user to input the design parameter, and it is illustrated in Fig. 14.4.

#### ***14.2.4 Simulation Optimisation***

The simulation optimisation framework integrates the MOCBA algorithm and the GA where MOCBA is controlling the assignment of simulation replications for each design alternative and GA is in charge of generating promising design alternatives.

MOCBA is an algorithm that is used to tackle the multi-objective ranking and selection problem. It aims at allocating simulation replications to design alternatives so as to minimise type 1 and type 2 errors. Type 1 error is defined as the

probability of missing non-dominated designs in the Pareto set while the type 2 error refers to the probability of including dominated designs in the Pareto set [17]. These two errors are also related to the probability of correct selection. The algorithm of implementing MOCBA is as follows:

1. Running the simulation model for each design alternative with an initial number of simulation replications.
2. Compute the sample mean and the sample variance based on the simulation outputs, and then select the designs into the Pareto set based on these values.
3. Compute the type 1 and type 2 errors, if these errors are less than preset tolerance levels or the computing budget has been exhausted, we will terminate the algorithm. Otherwise, go to step 4.
4. Allocate additional simulation replications to design alternatives according to the asymptotic allocation rule of MOCBA (the designs which play the dominating role will be assigned according to the square root rule while the designs playing the non-dominated role will be assigned according to the noise-to-signal ratio), go to step 2.

The details of the algorithm as well as the asymptotic MOCBA allocation rules can be found in [17].

GA is introduced as a computational analogy of adaptive systems. It is modelled loosely on the principles of the evolution i.e., natural selection, in which a population of individuals undergo selection in the presence of variation-inducing operators such as mutation and recombination (crossover) operators. A fitness function is used to evaluate individuals, and the reproductive success varies with fitness.

GA is chosen as the search engine to find the best design for two main reasons. The first reason is that GA is commonly used in industries and is proven as an effective search heuristic. Institutions such as National Aeronautics and Space Administration (NASA) have employed GA in their research. The second reason is that GA is able to help the search escape from local optimum.

When we implement the simulation optimisation framework to solve the port layout design problem, we use a standard type of GA. For chromosome representation, we represent each decision variable as a gene. In our case studies, there are six decisions to be decided, and therefore there are six genes in each chromosome. These six decisions are the number of blocks, the length and the width of a block, the number of quay cranes, the number of yard cranes and the number of vehicles. We use arithmetic crossover as the crossover operator. For mutation operator, we use a generic operator to alter one or more genes in the chromosome at one time. Tournament selection is used as the selection mechanism.

We have integrated MOCBA with GA algorithm in the following way:

1. In each generation of GA, we will use MOCBA algorithm to determine the simulation replications allocated for design alternatives in the population.
2. We use the approximated probability of non-dominating as the fitness of the designs. This probability of non-dominating can be estimated when we run



the MOCBA algorithm. This is also the fitness value chosen by Lee et al. [10] when they utilise MOCBA in their research.

3. In each generation of GA, all the non-dominated designs will be put into an elite set. When the GA terminates, MOCBA algorithm will be run on all the designs in the elite set again in order to identify the true Pareto designs.

The GA code is developed in-house to facilitate the link between the ALG and MOCBA modules, and to allow future developers easily modifying the code if necessary.

The GA module is coded such that users can input the settings required for GA. Users will be able to define the number of designs in one generation; the number of designs to be carried over to the next generation; the number of designs to be designated as parent candidates; the number of parent candidates to be selected as the mating parents in the tournament selection phase; the mutation probability; and the number of generations to evaluate before the stopping criteria are met.

### ***14.2.5 The Overall Procedure of the Framework***

The input from the port operators will be directly entered into .csv files, which will be served as an input file for the different modules in the program. Another set of input with regard to the port usage, such as vessel arrival and container counts, will have to be input into text files to use with the Automod simulation models. The general steps of the framework are presented below:

1. Initialisation: users define settings for GA and MOCBA. Users define port's fixed parameters and variable parameters range. Users define the following files for Automod simulation: vessel inter-arrival time; vessel's quay crane requirements and number of containers in a vessel; each container type (import, export or transshipment), size (20 or 40 ft), their stay in the yard and destination; containers sizes and inter-arrival times coming from the hinterland for export; quay crane and yard crane parameters (loading and discharging times).
2. Generate initial set of port designs based on user inputs.
3. Run the MOCBA module to determine the number of simulation replications to be allocated to each design. Output the performance index for each design and place the non-dominated designs into an elite set.
4. Run the GA module to generate a new set of designs based on the performance index from MOCBA.
5. If termination criterion is not met, repeat from step 3. Else, run MOCBA on the elite set to delete all the dominated designs from the set.
6. Output the non-dominated designs from the Pareto set.

The details of the framework can be found in the Fig. 14.5.

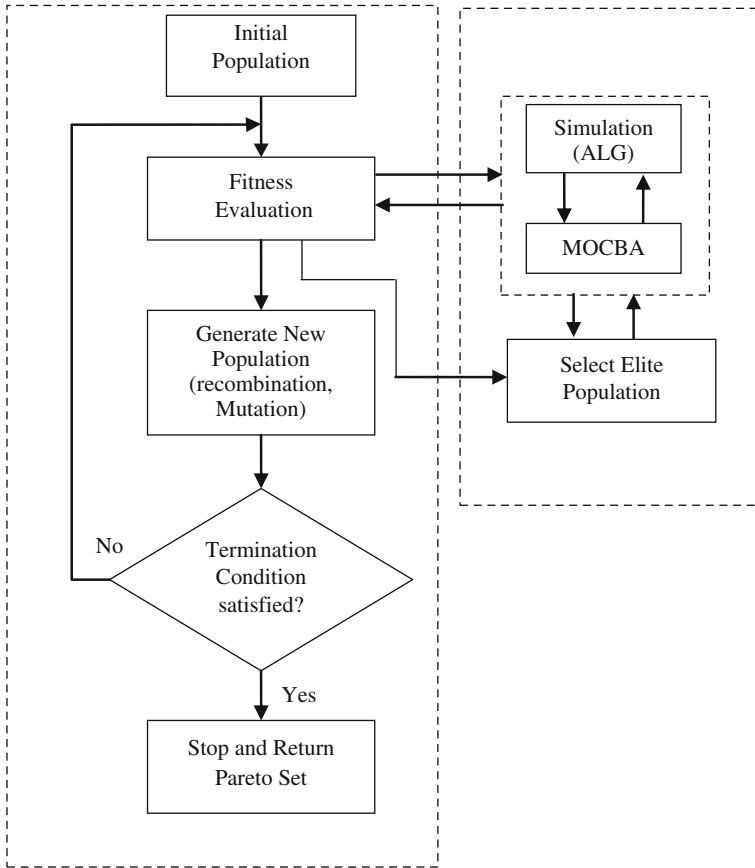


Fig. 14.5 Flow chart of the multi-objective simulation optimisation framework

### 14.3 Numerical Experiments

We use some case studies created based on hypothetical example with real port parameters to demonstrate the application of this framework. We also compare the performances of different algorithms.

#### 14.3.1 Specifications

A port operator decides to build two container terminals (Ports A and B) given two lands with fixed dimension. The proposed platform will be used to perform the simulation optimisation and offer recommendations on the layout and equipments requirements for these two pieces of lands.

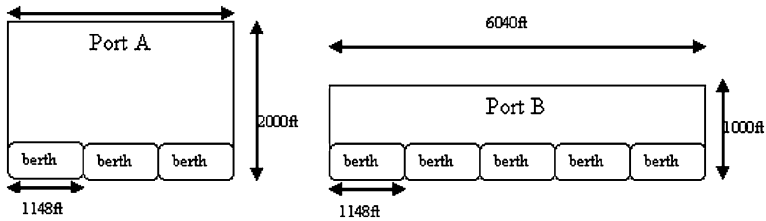


Fig. 14.6 Land specifications for Port A and Port B (not to scale)

Port A uses a land with a dimension of 3,744 ft by 2,000 ft and Port B uses a land with a dimension of 6,040 ft by 1,000 ft. For both lands, the longer side faces the sea. Using the standard berth size of 1,148 ft, Ports A and B can hold 3 and 5 berths, respectively (refer to Fig. 14.6). It is decided that they would use cranes, which have a width of 88.5 ft. Each quay crane will be placed 130 ft apart. Their budget allows them to purchase up to a maximum of 15 quay cranes, while they can afford a maximum of 8 prime movers per quay crane.

Based on the past experiences and forecasts, the vessel type, inter-arrival timings and the number of containers each vessel carries, loading and discharging time of the yard and quay cranes are given.

In order to attract more vessels to the container terminal, it is required that the layout minimises vessel turnaround time. Another additional requirement is to maximise the quay crane utilisation, which saves cost in the long-run as quay cranes are the most expensive equipment of the container terminal.

### 14.3.2 Experiment Results

Figures 14.7 and 14.8 shows the results of Ports A and B, respectively. We could see that the QC utilisation of both ports is about 70–90%, and the Port A's turnaround time is a slightly lower than Port B.

To illustrate the efficiency of the MOCBA module that was suggested in the proposed platform, we ran the same design parameters using UCBA (uniform computing budget allocation algorithm, i.e., the computing budget will be allocated equally among all the designs), which is another commonly used algorithm when the simulation budget is abundant. In order to have a fair comparison, MOCBA and UCBA are allocated with the same total computing budget.

We compare the results from MOCBA and UCBA. From the results in Figs. 14.7 and 14.8, we could see that the MOCBA is better than the UCBA.

The non-dominated points of MOCBA and UCBA for Port A and B are illustrated in Figs. 14.9 and 14.10. Although those result points are close to each other we still could see the MOCBA is better than UCBA. By combining the results obtained from MOCBA and UCBA, we analysed the percentages of non-dominated results from MOCBA and UCBA, respectively. For Port A, the

Fig. 14.7 Results of Port A

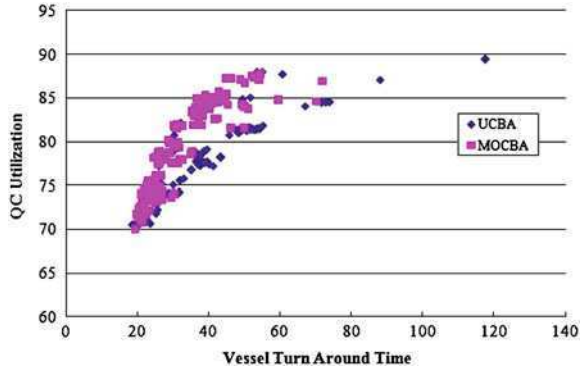


Fig. 14.8 Results of Port B

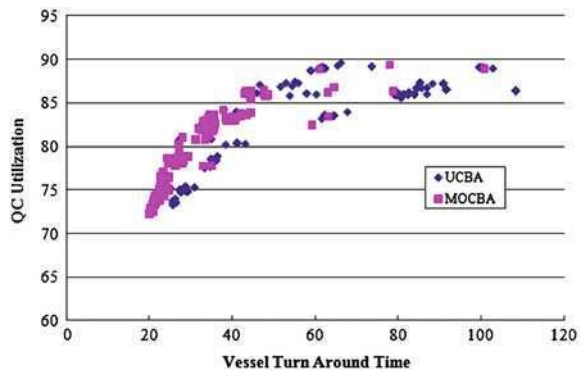
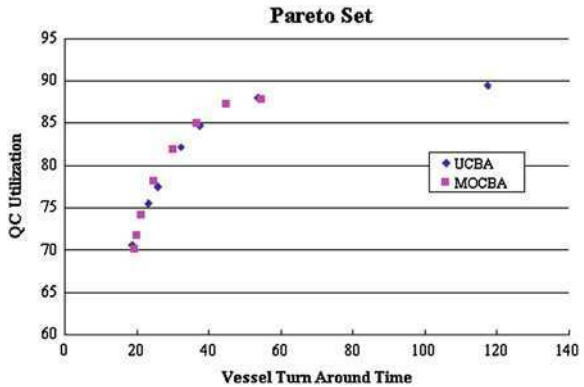


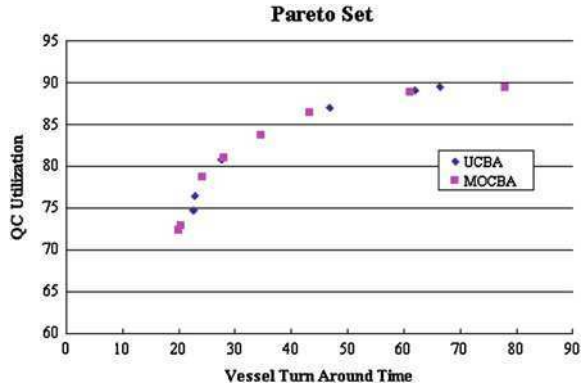
Fig. 14.9 The non-dominated results of Port A



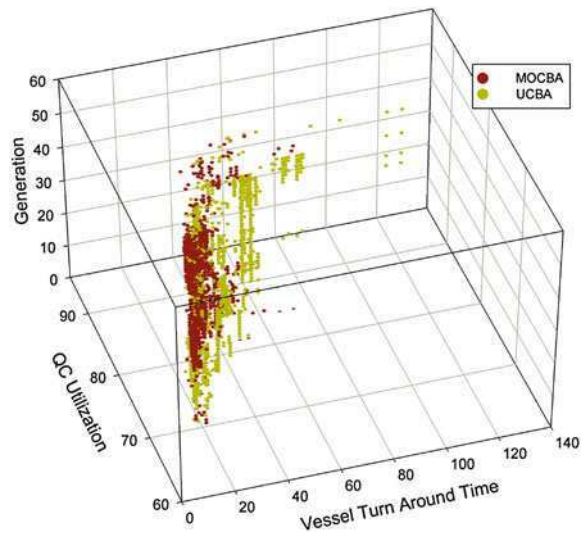
percentage ratio of non-dominated results (MOCBA: UCBA) is 5:3; for Port B, the ratio is 5:4. The results validate that MOCBA outperform the UCBA.

Figure 14.11 shows the evolution process of results of MOCBA and UCBA in different generations for Port A. We use CDE (closest distance to efficient frontier) as a measure to compare MOCBA and UCBA. Here, CDE is weighted sum of the closest distances between the results to the Pareto front. The lower CDE value is

**Fig. 14.10** The non-dominated results of Port B



**Fig. 14.11** The results of MOCBA and UCBA in different generations (Port A)



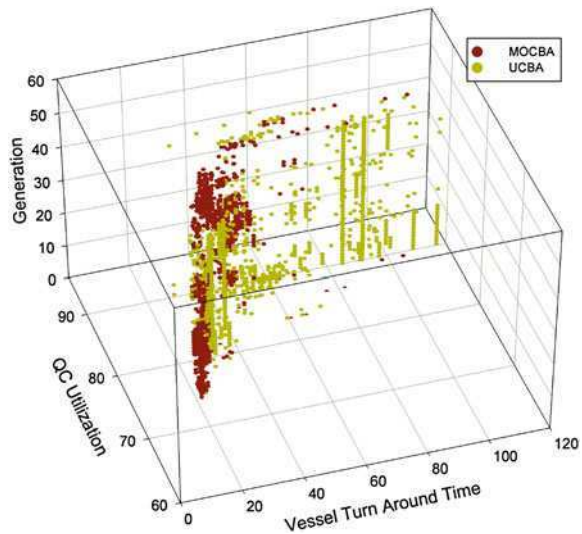
better. For Port A, the CDE of MOCBA and UCBA are 87.22 and 115.79, respectively, which demonstrates that MOCBA is better than UCBA.

Figure 14.12 shows the evolution process of results of MOCBA and UCBA in different generations for Port B. We also use CDE (closest distance to efficient frontier) as a measure to compare MOCBA and UCBA. For Port B, the CDE of MOCBA and UCBA are 53.69 and 338.95, respectively, which demonstrates that MOCBA is better than UCBA.

### 14.4 Conclusions

In this chapter, the port layout design problem is solved using the proposed simulation optimisation framework. It is built to ease human effort in manual port design process and utilise the power of optimisation algorithms in selecting the

**Fig. 14.12** The results of MOCBA and UCBA in different generations (Port B)



promising port designs. We use Automod software to illustrate the implementation of the framework, and the ALG is developed to create Automod simulation models given the design parameters. This helps to make simulation optimisation possible. We show in our case studies that the proposed simulation optimisation algorithms are effective in allocating computing resources and able to find better non-dominated designs.

The framework is envisioned to be a “plug and play” program where developers can add in different modules when they want to try new designs, different operation strategies and other simulation optimisation techniques. On the other hand, even though Automod is used in this framework, these concepts are general enough to be applied to other commercial software if they can be effectively communicated with some external programs.

## References

1. Steenken, D., Voß, S., & Stahlbock, R. (2004). Container terminal operation and operations research—a classification and literature review. *OR Spectrum*, 26, 3–49.
2. Vis, I. F. A., & de Koster, R. (2003). Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research*, 147, 1–16.
3. Kozan, E. (1997). Comparison of analytical and simulation planning models of seaport container terminals. *Transportation Planning and Technology*, 20, 235–248.
4. Bruzzone, A.G., Giribone, P., & Revetria, R. (1999). *Operative requirements and advances for the new generation simulators in multimodal container terminals: Proceedings of the 1999 Winter Simulation Conference* (pp. 1243–1252).
5. Yun, W. Y., & Choi, Y. S. (1999). A simulation model for container-terminal operation analysis using an object-oriented approach. *International Journal of Production Economics*, 59, 221–230.

6. Nam, K., Kwak, K., & Yu, M. (2002). Simulation study of container terminal performance. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 128, 126–132.
7. Shabayek, A. A., & Yeung, W. W. (2002). A simulation model for the Kwai Chung container terminals in Hong Kong. *European Journal of Operational Research*, 140, 1–11.
8. Sgouridis, S. P., Makris, D., & Angelides, D. C. (2003). Simulation analysis for midterm yard planning in container terminal. *Journal of Waterway, Port Coastal and Ocean Engineering*, 129, 178–187.
9. Yang, C. H., Choi, Y. S., & Ha, T. Y. (2004). Simulation-based performance evaluation of transport vehicles at automated container terminals. *OR Spectrum*, 26, 149–170.
10. Lee, L.H., Chew, E.P., Cheng, H.X. & Han Y.B. (2008). *A study of port design automation concept: Proceedings of the 2008 Winter Simulation Conference*, 1–5 (pp. 2726–2731).
11. Olafsson, S., & Kim, J. (2002). *Simulation optimization: Proceedings of the 2002 Winter Simulation Conference*.
12. Fu, M.C., Glover, F.W. & April, J. (2005). *Simulation optimization: A review, new developments, and applications: Proceedings of the 2005 Winter Simulation Conference*, 1–4 (pp. 83–95).
13. Ramaekers, K. (2009). A simulation optimization approach for inventory management decision support based on incomplete information. *4OR: A Quarterly Journal of Operations Research*, 7, 93–96.
14. Adewunmi, A., & Aickelin, U. (2007). Noise Reduction Technique for a Simulation Optimization Study, 2007.
15. Lee, L. H., Chew, E. P., Teng, S. Y., & Chen, Y. K. (2008). Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem. *European Journal of Operational Research*, 189, 476–491.
16. Lee, L.H., Chew, E.P., & Teng, S.Y. (2006). *Integration of statistical selection with search mechanism for solving multiobjective simulation-optimization problems: Proceedings of the 2006 Winter Simulation Conference*, 1–5 (pp. 359–368).
17. Lee, L. H., Chew, E. P., Teng, S. Y., & Goldsman, D. (2010). Finding the non-dominated Pareto set for multi-objective simulation models. *IIE Transactions*, 42(9), 656–674.

# Chapter 15

## Simulation-Based Innovization Using Data Mining for Production Systems Analysis

Amos H. C. Ng, Catarina Dudas, Johannes Nießen  
and Kalyanmoy Deb

**Abstract** This chapter introduces a novel methodology for the analysis and optimization of production systems. The methodology is based on the innovization procedure, originally introduced for unveiling new and innovative design principles in engineering design problems. Although the innovization method is based on multi-objective optimization and post-optimality analyses of optimised solutions, it stretches the scope beyond an optimization task and attempts to discover new design/operational rules/principles relating to decision variables and objectives, so that a deeper understanding of the problem can be obtained. By integrating the concept of innovization with discrete-event simulation and data mining techniques, a new set of powerful tools can be developed for general systems analysis, particularly suitable for production systems. The uniqueness of the integrated approach proposed in this chapter lies on applying data mining to the data sets generated from simulation-based multi-objective optimization, in order to automatically or semi-automatically discover and interpret the hidden relationships and patterns for optimal production systems design/reconfiguration. After describing the simulation-based innovization using data mining procedure and its difference from conventional simulation analysis methods, results from an

---

A. H. C. Ng (✉) · C. Dudas · J. Nießen · K. Deb  
Virtual Systems Research Centre, University of Skövde,  
PO Box 408, 541 28 Skövde, Sweden  
e-mail: amos.ng@his.se

C. Dudas  
e-mail: catarina.dudas@his.se

K. Deb  
e-mail: deb@iitk.ac.in

K. Deb  
Department of Mechanical Engineering, Indian Institute of Technology Kanpur,  
Kanpur, India



industrial case study carried out for the improvement of an assembly line in an automotive manufacturer will be presented.

## 15.1 Introduction

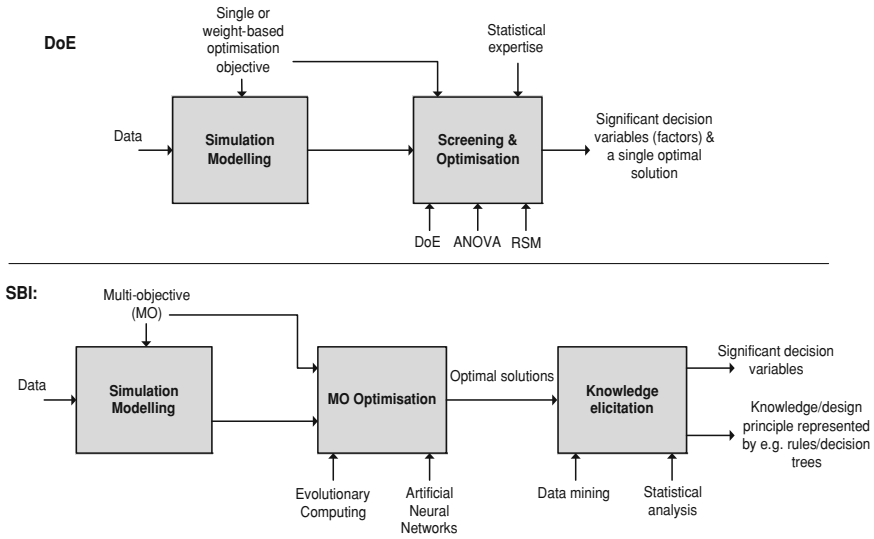
A production system can be defined as the arrangement and operation of material, machines, tools and human resources and information to produce the value-added physical or service products to satisfy certain customer/market needs [1, 2]. In practice, designing a production system involves a series of complex decisions over time to satisfy the strategic objectives of the company [3]. The decisions on, e.g., equipment sizing, layout, level of automation, workload allocations, material and information flow, for a new production system or for the re-configuration of an existing production line to cope with new product variants, can pose big challenges to the designer/manager because of the complex combinations and interactions among the system entities. Furthermore, to select the optimal parameters of the system entities so as to achieve the desired overall performance of the production system is a very complex task that has been proven to be difficult for the decision-maker in the design process. Take an example from the automotive industry. In connection to the current adaption to more CO<sub>2</sub> efficient powertrains and vehicles, the automotive industry must change-over to the production of new fuel-saving products, including other variants and components than in current production. As a result from this, gaining profitability is not just a matter of simply running current production in a more efficient way. Automotive manufacturers worldwide are facing many important decisions in designing or re-configuring production facilities to accommodate this increased number of variants. These decisions are extremely important since they tend to lock around 80% of cost of the investment and operation costs. In other words, if the optimal alternatives are not explored and considered so that non-optimal decisions have been made in the early stages, then the investment cost will be significantly higher and the operational costs of the production system will be affected throughout its whole life cycle. The aim of production systems analysis, particularly in the design, re-configuration and/or improvement phases, is therefore to provide the advanced methods and tools to aid the production designers/managers to have a deeper understanding of the problem in hand, systematically explore and evaluate different alternatives and then generate the essential information/knowledge to support them to make the right decisions in order to optimise the performance of the production system as a whole.

Despite the essential role that production systems analysis can play, the common industrial practice today seems to make important decisions based mainly on the experience from existing processes and static estimation tool. With the abundance of data collected and saved in industry today, it is possible to perform detailed analysis on an existing process. Nevertheless, when it comes to making important decisions for the design/re-configuration or improvement in the system

level, then the decision-makers are very often caught into the problem of shifting the right and accurate information out of the data ocean—the so-called data haystack syndrome [4]. One of the barriers for more efficient production is that while there is in principle abundant data about both the productivity at different level of the factory, these data need to be organised and transferred into knowledge suitable for decision-making support. As an example, unravelling or discovering relationships between input parameters and output parameters such as productivity and product quality requirements in manufacturing is seen as an important task. In general, the term “knowledge discovery” refers to a higher-level task in which in addition to solving the current problem, important insights about solving similar problems are also gained. This makes the user aware of the interactions among problem parameters and their combined effect on the overall performance of the system.

Many books and research papers have been written on the issue of knowledge discovery, but most literature focuses on the importance of the knowledge discovery task rather than suggesting any pragmatic and realisable procedure of discovering hidden knowledge in a systematic manner. Recently, Deb [5] proposed an ‘innovization’ task (the term comes from the task of creating innovative design principles through optimization) for this purpose and demonstrated its use in many engineering design and process optimization problems by performing a manual knowledge retrieval procedure. By integrating the concept of innovization with discrete-event simulation (DES), we believe that a new set of powerful tools can be developed for general systems analysis, particularly suitable for production systems analysis in order to support optimal decision-making in design and improvement activities. This method is so-called simulation-based innovization (SBI). The proposed SBI method can be divided into three main tasks:

1. Gathering high-performing solutions through multi-objective optimizations (MOO) via simulations: by considering at least two conflicting objectives of design, an evolutionary multi-objective optimization (EMO) procedure creates multiple trade-off optimal (or high-performing) solutions to a problem [6]. This procedure will generate a set of variable-objective data set in which each solution is an optimal or near-optimal solution and there exists a clear trade-off among objectives from one solution to the other. The user is thus able to take an informed decision, i.e., selecting a solution that is best for the situation at hand. For production systems design in the conceptual phase, this step can be fully supported by using FACTS analyzer, an internet-enabled DES tool with built-in EMO capability [7].
2. Retrieving hidden knowledge: it is argued that since the obtained solutions are all optimal (or close to being Pareto-optimal solutions), they are bound to follow and exhibit certain relationships among variables vis-à-vis objectives. For example, in one type of problem, it may be observed that all trade-off optimal solutions require a particular design/operation variable to take a fixed or almost fixed value whereas other variables must be changed linearly or exponentially with an expected change in an objective value. Insight in such intricate relationships provides useful knowledge about ‘how to solve the



**Fig. 15.1** Comparing SBI method with DoE in simulation analysis

problem optimally’, and moreover, it results in a better expertise in solving the problem (or related problems) in the future with different data or parameters.

3. Developing a knowledge-base for future use: once such knowledge as mentioned above has been obtained, it has to be documented/stored so that meaningful generic and specific implications of this knowledge can be understood and disseminated within various divisions/levels of the organization in order to achieve a better/improved design/operation of the production system.

The solution of using innovization for knowledge extraction, in forms of rules, will be an important scientific achievement and would be a unique attempt to find such relationships in an automated manner. This should have a long-term impact to the scientific community and industrial practice. The developed rule bases will be investigated for their validity from the theory of optimization. Thereafter, the rule bases will be verified using either the real system or its simulation model for their contextual validity and usefulness. If needed, more solutions can be created until a set of rules which describe the relationships among variables and objectives of trade-off optimal solutions is obtained. In other words, SBI is an iterative process that may require frequent interactions with the decision-maker. The knowledge generated from SBI will be most valuable for better modelling and understanding the good design/operational principles for the production system under study. Schematically, the proposed approach and the conventional approach in using simulation for optimization can be distinguished by comparing the system analysis and design technique (SADT) diagrams in Fig. 15.1. It is important to emphasise that while the concept of SBI is a research challenge, it is targeted to be an automated or semi-automated procedure. In other words, we are aiming at developing a SBI-based decision- support system in which users have no need to

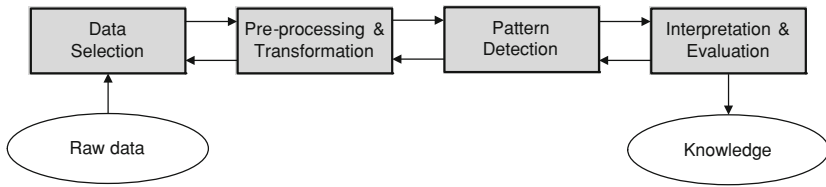
possess any specific expertise to use it when compared to running experiments and performing statistical analysis using design of experiments. With this in mind, state-of-the-art data mining (DM) techniques like clustering [8] and decision trees [9], which can be used to automatically or semi-automatically discover and decipher hidden properties, relationships or patterns of the optimal solutions are hence very suitable for the purpose of SBI. On one side, DM techniques, particularly decision trees, can offer many benefits, like generating easy-to-follow and self-explanatory ‘rules’ [10], not merely in the knowledge/pattern extraction as required in the second step of SBI, but also for step 3—developing a knowledge-base. On the other side, MOO facilitates the generation of a set of wide-spread and diversified optimal solutions as the training data set required for a reliable DM process, which in contrast cannot be obtained with a singleton of optimal solution in a single objective optimization scenario [8].

The aim of this chapter is to introduce such a SBI using DM, or SBI-DM, procedure for production systems analysis. The rest of the chapter is organised as follows: we begin with a literature review of DM for knowledge discovery so that the terminology and concepts of general DM techniques related to SBI-DM are defined and briefly explained (Sect. 15.2). Description of the SBI-DM procedure is provided in Sect. 15.3. A case study and results from applying SBI-DM to an assembly line at an automotive manufacturer are presented in Sect. 15.4. Conclusions and our current plan in applying SBI-DM for more complex production scenarios/case studies can be found in Sect. 15.5.

## 15.2 Data Mining for Knowledge Discovery: A Literature Review

A widely accepted definition of DM is ‘the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data’, given by Fayyad et al. [11]. Pattern in this case can be any relationship between data sets, data fields and values or regularity in the data. DM is usually associated with the term knowledge discovery in databases (KDD) because relevant data sets are mainly found and searched in different databases [12] and to identify the important patterns in the data stored in databases is still the key application of DM techniques.

The term KDD was introduced in the early 1990s [13]. Just a short period of time later the idea to see the discovering as a guided process for extracting useful knowledge from huge data sets was developed. To some authors, like Fayyad et al. [11], the KDD process enlarges the DM process with some pre- and post-processing tasks which are necessary to be more problem-specific. This is important to prevent negative impacts on the validity of the analysis results. To acknowledge a data analysis task as an overall process has quickly become a wide-spread standard so that the KDD process model is nowadays applied in different domains like crime prevention and clear up, forensic and judicial environment and



**Fig. 15.2** An ordinary KDD process model

especially in the business world where KDD helps with credit scoring or the realization of direct marketing [14].

The quality of the discovered knowledge in many cases is influenced by the effectiveness of the DM technique used in the pattern detection step and the size as well as the quality of the data being investigated. ‘If users select the wrong data, choose inappropriate attributes, or transform the selected data inappropriately, the results will likely suffer’ [15]. Because of this, knowledge discovery has to be understood as a process where each single step is paramount. As illustrated in Fig. 15.2, an ordinary KDD process consists of four consecutive steps: data selection, preprocessing and transformation, pattern detection as well as interpretation and evaluation.

Although the first step is data selection and the last step is interpretation and evaluation, the KDD process cannot be seen as a linear process. In the pre-processing step it might come out that there is some missing data which require some modification in the selection step. This applies for the pattern detection and interpretation/evaluation steps as well. When it comes out during the interpretation that the found patterns are not applicable or useless, some changes of the input parameters of the used DM method might enhance the situation. In the worst case, the entire process must be restarted from the beginning. Feedback loops are essential for the whole analysis process [15]. Each step of the KDD process will be described with more details in the following sections.

### ***15.2.1 Data Selection***

In general, data selection in a KDD process is defined as the extraction of a subset from all accessible data which is related to the underlying problem area. This implies that there is an elimination process for some data in the data set which are considered to be not sufficiently important to the data analysis process [16].

It is sometimes overlooked that the process idea of KDD is also valuable for artificial data sets generated from simulation data. Those data sets usually do not have their origin in the real-world but have an experimental character. The theoretical accessible data in this case might be the complete permutation of all input variables which can be evaluated to gather some output. To evaluate all different permutations might not be possible due to computational limitations. Therefore,

the focus has to be set on a smaller subset which has predefined characteristics (e.g., being a Pareto-optimal solution). In other words, the data selection process can also be seen as a search (optimization) process.

### ***15.2.2 Pre-Processing and Transformation***

The task to be done in the pre-processing and transformation step can be very different and depends on the characteristics of the underlying data set selected. Because of this, there are several different terms connected to this step in the literature:

- *Data cleaning* sets up priorities on incorrect data and is necessary whenever there are mistakes in the data acquisition process [16]. Incorrect data sets make representation through missing values which have to be filled as the case may be ignored or noisy data which have to be corrected.
- *Data integration* is relevant if data sets from different sources are used/fused. Different sources (databases, websites, flat files, etc.) often stand for incompatible data models and formats as well as for incomprehensible naming convention [15]. Data integration prevents the pattern detection step from using redundant data sets and increases the quality of the data set by supplementing missing attributes, removing duplicate instances and resolving data inconsistencies [17].
- *Data reduction* is needed to downsize the data sets. This should be done for two reasons: (1) there might be a computational bottleneck due to data volumes of several giga- or terabytes; (2) the pattern detection algorithm might find more interesting pattern if it works with reduced data sets using downsize procedures [18].

Regardless of which DM operation is applied, the purpose of the pre-processing and transformation step is to prepare the data for the subsequent mining process. It is important because appropriate data sets can improve the quality of the mining results and can also decrease the time required for running the DM algorithm [12].

### ***15.2.3 Pattern Detection/Data Mining***

Pattern detection is the step where the prepared data is analysed by the application of specific algorithms for extracting patterns from data [11]. Since the techniques to find pattern are originated from DM, both terms (pattern detection and DM) are used synonymously. In the present paper it is assumed that DM is a step in the overall KDD process which is surrounded with some previous and subsequent steps. Besides the term DM, there appear some other expressions less frequently used in the literature, e.g., knowledge mining, knowledge extraction, data analysis, pattern analysis, data archaeology and data dredging [12]. While the emphasis can be slightly different, but in general it is assumed that all these terms can be covered by the term DM.

There are many different interpretations of data mining in general. The question whether an operation is a DM method or not can be simplified by the analysis of the goals. Basically there are two different goals to be reached by analysis—pattern detection and hypotheses verification:

- *Verification-driven data mining* refers to the goal hypotheses verification. Usually it extracts information in the process in order to validate a hypothesis postulated by a user. Predominant techniques in this field are multi-dimensional analysis.
- *Discovery-driven data mining* refers to the goal pattern detection. It automatically extracts knowledge from data sets when there is not much known about it. Patterns are derived or functions are learned, which are valid for the underlying data set [17].

Other authors like Neckel and Knobloch [19] assign only methods for pattern detection to DM. This chapter takes this view because the discovery of new knowledge is in the foreground. To verify a hypotheses means there is already an understanding about the investigated field and there already exist some assumptions.

Different DM algorithms have different characteristics for handling the analysed data. A wide-spread differentiation is to classify in supervised learning and unsupervised learning:

- *Supervised Learning*—a supervised learning method differs between input attributes and target attributes (also dependent and independent attributes, see [20]). Methods which follow the supervised learning paradigm attempt to discover the relationship between these attribute types. The structure of the discovered relationship is covered in a model which can be used for further operations [10]. During the learning phase different samples of a training set are given as an input. A pattern in the training set on which the transformation from input variables to also target variables is based on is occupied with costs. These costs usually are a figure that represents how well a pattern is able to transform input variables to target variables correctly. The goal is to find a model which minimises the sum of costs for all training samples [21]. Because of the evaluation part where bad solutions are punished with high costs it can be spoken about a supervisor or teacher which leads to the name supervised learning. Since supervised learning improves the ability of the trained method to transform input variables to target variables correctly, it is commonly used for prediction methods [20]. For good predictions there must be enough training cases, otherwise there is a risk that some pattern will not be found. On the other hand, some prediction methods might be unstable which mean that more training cases lead to weaker results [18].
- *Unsupervised learning*—in contrast to supervised learning where the relationship between input variables and a target variable is important, unsupervised learning refers to ‘modelling the distribution of instances in a typical, high-dimensional input space’ [20]. There is no dependent attribute like the target variable in supervised learning methods and there is also no explicit teacher [21]

which means that it is free from user influences. Unsupervised learning methods detect similarities and differences in a data set and is also able to group similar subsets into clusters or segments [22]. Clustering and dependency analysis are usually done by an unsupervised learning method.

### 15.2.4 Interpretation and Evaluation

The final step is the interpretation and evaluation of the found patterns. Since there can be too many different solutions or patterns found which might not represent certain knowledge due to uncertainties, incomplete data or faulty data capturing, it is important to have an instrument to separate a good solution from a bad solution [23]. All calculations which are done for this can be framed by the term interestingness measures [12].

The goal of interestingness measures is the identification of the interesting patterns which represent interesting knowledge. There exists a vast amount of interestingness measures due to different perspective on interestingness. Liu and Özsü [17] state that those patterns are interesting which deliver useful knowledge for a given application and have a certain degree of validity. Usefulness and validity are also important for Han and Kamber [12] and they demand novelty and easily understandable solutions. Bissantz and Hagedorn [23] remind especially that the definition of DM demands novelty and non-triviality; found patterns are trivial and therefore not interesting if:

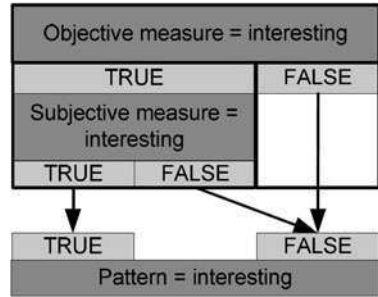
- They are tautological (e.g., all pregnant patients were female).
- Multiple rules describe the same context.
- The found statement refers only to a single element of the whole data set.
- An expert has known this statement before.
- A simple data operation like multi-dimensional query could produce the same result.

The absence of a wide-spread agreement on a formal definition, interestingness is defined as a broad concept in [24]. On top level they differ between three measurement groups: objective measures, subjective measures and semantics-based measures:

- *Objective measures* are the most common measurement group where neither knowledge about the user nor knowledge about the application is needed. The measures are only based on the raw data so this is why they are designated as objective. The used measures often have their origin in theories from probability and statistics [24]. Terms and measures which fit to this group are accuracy, conciseness, generality, reliability, peculiarity and diversity.
- In contrast to objective measures, *subjective measures* consider explicit knowledge or expectation of the user or offer an interacting process with the user [24]. This causes alternative result patterns for different users with different



**Fig. 15.3** Relationship of subjective and objective measures



characteristics and therefore can be designated as subjective because they reflect the needs and interests of a particular user [12]. A difficulty in this measurement group is the representation of user knowledge which is required. Terms and measures which fit to this group are novelty and surprise.

- For *semantics-based measures*, domain knowledge of the investigated area is needed. Since this knowledge can be gained from an expert user it can also be seen as a special case of subjective measures. The difference is that domain knowledge in this case is not about the data itself like in subjective measures, but represents a kind of utility function where user goals are reflected [24]. DM results should be patterns that optimise this utility function. The functions of the different measurement groups are not equal. Good values for objective measures are an imperative condition because an adequate objective measure which shows that a found pattern is not interesting from an objective perspective cannot become interesting from a certain perspective of an individual user. Subjective measures (also semantics-based measures) can be seen as a sufficient condition. Figures of those measures are alone not able to show whether a pattern is interesting or not but without them it is not possible to say if a pattern is interesting for a certain application area.

The relationship of subjective and objective measures for supporting the discovery of interesting patterns is summarised in Fig. 15.3 [25]. For all measuring methods count they are associated with a threshold [12]. This threshold is controlled by the user and finally is decisive for labelling a pattern as interesting or not interesting. For example, found rules that do not reach a confidence threshold of 70% might be designated as not interesting by the user. A rule which does not reach this threshold is more likely to reflect exceptions or noise and give no helpful information.

All interestingness measures usually appear in the last step of the KDD process model. They make it possible to rank different solutions—which mean that some kind of comparison of two solutions is also possible—and it is possible to filter interesting solutions from not interesting solutions. Besides those helpful actions sometimes, interestingness measures are used during the mining step to avoid the waste of computational capabilities for the discovery of uninteresting patterns.

### 15.3 A Data Mining Procedure for SBI

The research methodology proposed in this chapter is driven by the assumption that Pareto or near Pareto-optimal solutions generated by using SMO have some common properties, which make them outstanding than other solutions. Therefore, the overall goal can more precisely be stated as the extraction and translation of those common attributes to useful pattern which can be presented in form of knowledge. The necessary steps to reach this goal do not deviate significantly to a standardised process model of an ordinary KDD process. Therefore, based on the general DM procedure reviewed in the previous section, we introduce a novel DM procedure for SBI to extract knowledge from a simulation model. This procedure is referred to as SBI-DM and is illustrated in Fig. 15.4. As mentioned, the major deviation, when comparing SBI-DM with a KDD process, lies on the data used for the pattern detection which is not from a data source with historical data, but from experimental data generated from SMO. On the other hand, as it will be seen in Sect. 15.3.2, mapping the extracted rules with the colour-coded visualization of the Pareto-optimal solutions in the objective space is another uniqueness associated with SBI-DM.

#### 15.3.1 Data Selection and Pre-Processing Using SMO

The first step in any DM studies involves data gathering to form a data set. In the proposed SBI-DM procedure, the data set is generated from a SMO process where both input and output data are collected. Since DM techniques are used to discover pattern in the data, it is advantageous to have a good diversity of solutions from the Pareto front. Apart from using an efficient algorithm like NSGA-II [26], it is important to note that uncertainty handling is a critical issue if stochastic simulation is used to generate the data set. Uncertainty due to the randomness of stochastic simulation output can be regarded as the noise for EMO algorithms. In production simulation, this kind of noise may be posed by disturbances (machine breakdowns) in the operations or due to their natural variations. Solutions that are statistically superior (or inferior) might be discarded (selected) and wrongly ordered if the selection operation and non-dominating sorting (NDS) procedure in an EMO algorithm, like NSGA-II, do not take into account the deviations of the output data. Because of this, a novel definition of dominance, called confidence-based significant dominance (CSD), has been used in the MA-NSGA-II implementation in FACTS Analyzer (for more details of CSD, see [27]). Additionally, for the data set to contain a representative set of solutions, Latin hypercube design (LHD) [28] is more suitable than a random data selection. LHD algorithms following two characteristics which are important to provide a good set of initial solutions to both the optimization search as well as DM process: (1) the correlation between input factors can be minimised; (2) theoretically, the selected samples are

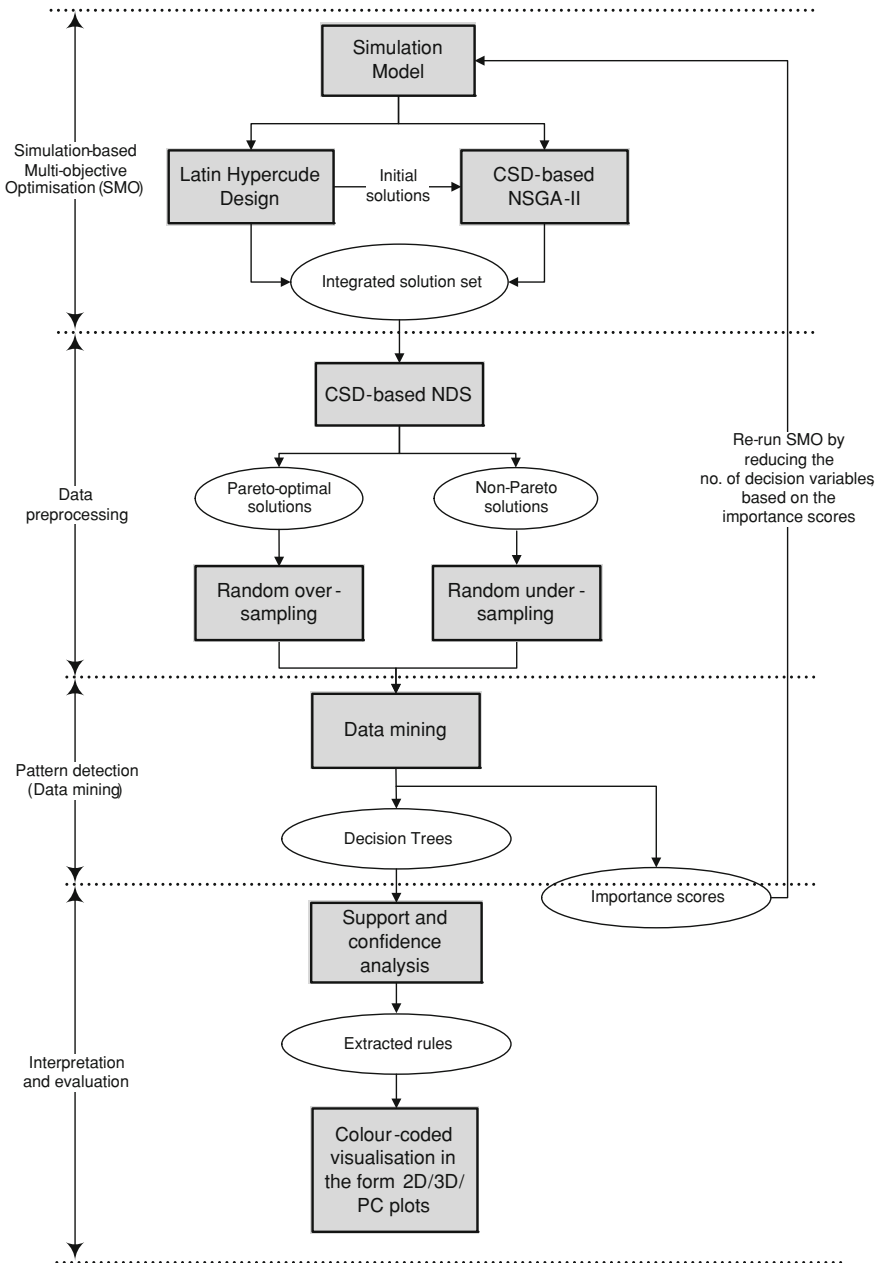


Fig. 15.4 The SBI-DM process

spread out across the whole input space. In the proposed SBI-DM procedure, all decision variables which shape the dimensions of the input space are used as the input for the LHD. The solution set generated by LHD can then be selected by the user as the initial solutions for MA-NSGA-II. After the optimization process, the two sets of data are combined into an integrated data set, sorted with CSD-based NDS again so that classification can be performed in a later step using DM.

Unlike an ordinary DM process in which data cleaning, integration and reduction are most important, the main purpose of the data pre-processing phase is to handle any imbalance of the data set. Actually, imbalanced data sets occur frequently both for prediction and description in any classification tasks. A data set is called imbalanced when the analysis task deals with a classification problem in which the allocation of the samples to the classes is imbalanced. Most samples of the data sets refer to one class which is also called the majority class and a small subset of the samples refers to the other classes which are called minority classes. Usually the minority class is the more important one. Unfortunately, many algorithms used for generating classification models are driven by accuracy which means that they try to minimise an overall error. The problem is that the minority class contributes only a little bit to this error. High-accuracy values are also possible when all samples of the minority class are misclassified. This implies some specific pre-processing has to be done to deal with the imbalance of Pareto-optimal and non-Pareto solutions. In SBI-DM, imbalanced data sets are likely to occur because the amount of Pareto-optimal solutions is very often smaller than the amount of non-Pareto solutions.

It is possible to reduce the samples of the majority class, or so-called under-sampling, or to increase the amount of samples in the minority class using over-sampling. While there are advanced algorithms for performing under-sampling, the simplest approach is random under-sampling in which an element of the majority class is randomly chosen and deleted until the majority class has reduced to the desired size. Similarly, in a random over-sampling approach, elements of the minority class are randomly selected chosen and duplicated with the same probability until a desired ratio between the Pareto-optimal and non-Pareto solutions is reached. In contrast to random over-sampling is the generation of new samples to enlarge the minority set. Generally speaking, the problem of the under-sampling approach is that some important information will be lost with the deleted solutions, especially if the random under-sampling approach is used which can lead to an uncontrolled waste of information [29]. As the LHD algorithm is used in SBI-DM to represent properly the whole input space, deleting some of these solutions would lead to a worse representation of the solution space. Therefore, an over-sampling seems to be more suitable. In contrast to random over-sampling, there are some generative over-sampling approaches that re-model the natural distribution of the minority class in order to generate new samples based on this distribution model [29]. Another popular method is the synthetic minority over-sampling technique (SMOTE) developed by Chawla et al. [30] which uses interpolation to create new samples. There are two major problems of creating new solutions using methods like SMOTE or generative over-sampling: (1) all the solutions generated

need to be evaluated with the simulation model; (2) more importantly, against the intention of the re-sampling, the generated samples do not lead to any new Pareto-optimal points in most cases. These mean creating new Pareto-optimal solutions will be both difficult and computationally expensive for any over-sampling algorithms. Therefore, random over-sampling is more suitable for the SBI-DM to artificially increase the set of Pareto-optimal solutions.

### ***15.3.2 DM for SBI Using Decision Trees***

The mining process involves choosing appropriate method(s) to be used for searching patterns in data depending on the problem at hand. For the modelling process the input data is used in order to estimate the outcome of a given output value, i.e., to build a predictive model that can predict an unseen observation and be able to tell its outcome value. The outcome value can either be nominal (e.g., positive/negative) or numerical (e.g., time lengths) and the first case of modelling is referred to as classification. Hence one wants to find the correct class among a limited number, and for numerical outcomes the modelling is called regression. Although the methods used for the modelling can be identical, it is important to distinguish a predictive model from a descriptive model in terms of the application purpose. One of the most important features of a predictive model is to be as accurate as possible when evaluated on independent data and there are several different measurements. Two measurements for classification models are accuracy, i.e., the amount of correctly classified, and the area under the ROC curve (AUC), the probability that a test example belonging to a class is ranked as being more likely belonging to the class than a test example not belonging to the class. For regression models, mean squared error measurement, i.e., the sum of all errors between the correct value and its corresponding estimated value is typically used. In addition to making correct predictions, to have a comprehensive or descriptive model is needed in most applications. For descriptive models, the reason behind a specific prediction can be identified and the decision-maker can also gain information about the importance of input variables and in the final analysis this can be visualised in a lucid way. Apparently, descriptive models are of great interest in the SBI-DM procedure because rule sets are required for the explanation of how to solve the problem optimally and as a final step these rules will also be visualised.

Decision trees are particularly appealing descriptive models due to their ability to provide comprehensible models and at the same time have high-predictive performance. The main principle when generating decision trees is a special case of a recursive partitioning algorithm, which the algorithm searches among the input variables in order to find, in some sense, the best one to split the data set. After each split there is a new subset of observations waiting to be assigned as a final leaf or if the process of finding the best split should be repeated on this subset, hence the term recursive partitioning. The final tree is a graph-like tree where the root contains the initial set of observations and all other nodes and leaves are

linked from this first node by edges labelled with the chosen variable and its corresponding splitting value. Each leaf contains a path description on the how to go there, i.e., a set of rules, and the predicted value for this specific leaf. For classification problems, this is a percentage of how likely it is to belong to each class and for regression problems this is a predicted mean value. In the proposed SBI-DM procedure, the aim of the DM is to distinguish the properties of decision variables on Pareto-optimal solutions from non-Pareto solution. Hence, unlike ordinary regression analysis for a single variable, a multi-variable analysis problem can be converted to a simple classification problem. The novelty of this method is that the study of multiple dependent variables can be converted into a simple output classification problem so that existing DM can be directly used.

The predictive performance can be improved by combining a large number of decision trees into so-called ensembles, which are used to form a joint vote on the classification or regression value for the output value [31]. Although the predictive performance is considerably improved, the lack of interpretation of the classifications is instead eliminated and hence also the descriptive capability. The path description, or rule sets, which will lead to a specific value is missing for ensembles. Nevertheless, advanced data mining tools like Rule Discovery System<sup>TM</sup> (RDS) [32], which is also used in this study, provides some guidance on how the classifications are done by offering the insight to the importance of each input variable. Consequently, the results from ensembles and single decision trees can be combined, where the former indicates the most important variables and the latter is used for generating rule sets. RDS uses a measure called *importance score* to quantify how influencing the input/decision variables to the classification/regression problem. This value stands for the impact of a variable on the reduction of the classification error in a decision tree. After the growing phase of the decision tree the squared error for each node and leaf can be calculated. These values are the basis for calculating the contribution of the variables for the overall squared error reduction. The overall squared error reduction is defined as the difference of the original squared error and the total resulting squared error which is left after building the complete decision tree. The contribution of one variable for the overall error reduction can be calculated as the sum of all differences between the squared errors of the parent nodes and the sum of the squared errors for its children. This contribution of the overall error reduction is afterwards transformed into the importance score by dividing it by the overall error reduction to get a normalised figure. Using the important scores, after the first run of data mining, unimportant variables may also be removed in order to get a crisper model with shorter rules and consequently less noise, as illustrated in Fig. 15.4.

The final step in the DM process is about how to present the results of the prior processes in a suitable way. Although the decision trees are comprehensive the results may need some refinement and adjustment in order to be interpretable for a decision-maker. This can be done by different visualization techniques, such as plotting the most important input variables versus the output variables in form of Parallel Coordinates. The SBI-DM procedure proposes a method to map the rules that are most interesting to the decision-makers to their associated points on the

objective space using different colours, so-called colour-coded rule visualization. To understand how this works, it is important to understand how we define and select ‘interesting’ rules in SBI-DM.

A classification rule is constructed by two parts, an antecedent set of conditions and the consequent class ( $c_1, \dots, c_j$ ). Each element in the antecedent set of conditions (A) consists of a variable ( $x_1, \dots, x_n$ ) and its corresponding value ( $v_1, \dots, v_n$ ) which is linked by an operator ( $op_1, \dots, op_n$ ):

$$\text{Rule } r_k : \text{If}(x_1 \text{ op}_1 v_1)\text{AND} \dots \text{AND}(x_i \text{ op}_i v_i)\text{THEN class} = c_i$$

One needs a way to determine the objective interestingness of these rules and the evaluation measures used are support and confidence which originate from the field of association rule mining [33, 34]. The support count (denoted as  $SupC(A \Rightarrow c_j)$ ) is defined as the number of patterns that fulfill both the antecedent set of conditions and the consequent class, and is used in the calculation of both support and confidence. Support is the ratio of the support count and the total number of observations (N) [34]:

$$\text{support}(A \Rightarrow c_j) = \frac{SupC(A \Rightarrow c_j)}{N} \quad (15.1)$$

Confidence is the ratio of the support count and the number of observations where the antecedent set of conditions (A) is true [34]:

$$\text{confidence}(A \Rightarrow c_j) = \frac{SupC(A \Rightarrow c_j)}{SupC(A)} \quad (15.2)$$

Support can be seen as an indicator of how frequent a rule is within the data set and as a consequence how significant that specific rule is. For a rule to be interesting its support has to be convincingly high since the decision support should disregard infrequent rules. The confidence should also be rather high since it reveals the strength of the rule among all observations in the data set. In SBI-DM, the user should set the decisive threshold to an appropriate value for the confidence level which means that rules with less confidence should be ignored. On the other hand, if a rule has confidence close to 1 but with very low support would most likely be labelled as an uninteresting rule due to the infrequency of the rule in the data set.

## 15.4 An Industrial Case Study

An industrial case study has been conducted for an engines assembly line in an automotive manufacturer. The aim of this case study was to investigate how the SBI-DM procedure can be applied to improve the performance of the assembly line through locating the critical area to improve, as well as identifying the key influencing parameters and their optimal values. Specifically, the SBI-DM

procedure has been applied for finding useful knowledge for the three inherently conflicting objectives of the production line, i.e., maximising production rate (throughput or TP), minimising average cycle time<sup>1</sup> (CT) and minimising average work-in-process (WIP).

As mentioned, the efficient SMO algorithm, namely MA-NSGA-II, has been used to find a number of trade-off high-performing solutions to this industrial problem. Thereafter, DM has been used to find rules related to variables and objectives, which are inherent to the obtained Pareto-optimal solutions. This involves using RDS to cluster the data set and then generate a decision tree-based predictive DM model using the SMO explored solutions as the data sets.

### ***15.4.1 The Assembly Line and its Simulation Model***

A DES model has been built for the assembly line using FACTS Analyzer, or simply FACTS. The model is an ‘abstracted’ model in the sense that some detailed operational logic and material handling between the workstations are not modelled and simulated. With the current configuration of the real line, a simulation (10 replications) has been run with a 6-day simulation horizon and 1 day warm-up period. Simulation output of the FACTS model gives:  $TP_0 = 67.5$  parts/h,  $CT_0 = 6458.2$  s and  $WIP_0 = 189.48$ . Simple validation has shown the standard error to be within  $\pm 3\%$  in TP and  $\pm 5\%$  in CT (due to disturbances, collected TP and CT figures of the real line are subjected to variances).

Today, there are 110 pallets being circulated in the real line. The number of pallets (denoted as  $N_p$  hereafter) has limited the maximum level of WIP that can be stayed in the main line. In other words, the main line is practically operated as a CONWIP (CONstant WIP) line, introduced in Spearman et al. [37]. With an intention to increase the throughput of the whole line, the production engineers working in the company have considered to increase  $N_p$ . By intuition, increasing  $N_p$  will directly increase TP. But how will this decision affect the other key performance measures of the production line, namely, CT and WIP? Little’s Law [35], written in the form  $TP = WIP/CT$ , has suggested that the same throughput can be achieved with a large WIP and long CT or with a low WIP and short CT [36]. It is therefore an important question to investigate what factors can make the system to reach a high level of TP, accompanied with a low WIP and short CT. The answer to this question will have a significant impact on the operational decisions if the production line at the company has to be improved. In the context of SMO, this implies the need of finding the multiple best or “optimal” tradeoffs

---

<sup>1</sup> Cycle time, which is also called variously as manufacturing lead time, throughput time or sojourn time, is used in this paper to refer to the time from a job is released at the beginning of the line/system until it reaches its end (i.e., the time a part spends as WIP). This terminology follows the definition found in standard textbooks for manufacturing systems analysis, e.g., [36].



between the maximization of TP and the minimization of both average CT and total WIP (main parts plus sub-assemblies in this study).

While there are small buffers, in form of small storages or conveyors, between all workstations, there are 11 bigger storage areas between several workstations. The sizes of the big buffers, along with  $N_p$ , represent the major decision variables of the entire assembly line. Therefore, there are totally 12 input variables used in the simulation-based optimizations under this study. It is very interesting to seek, through adjusting the 12 decision variables through SMO, the existence of any other configuration(s) which could improve the TP of line, without producing significant impact to CT and WIP. In terms of optimization, this is only a simple optimal buffer allocation problem. Nevertheless, the insights that have been gained in this study are very useful to the line improvement. For example, surprisingly, the results from applying SBI-DM to the FACTS model have revealed that increasing  $N_p$  will not only deteriorate CT and WIP of the line as suspected, but also not improving TP. These will be explained in the following sub-sections with the details of the SBI-DM steps.

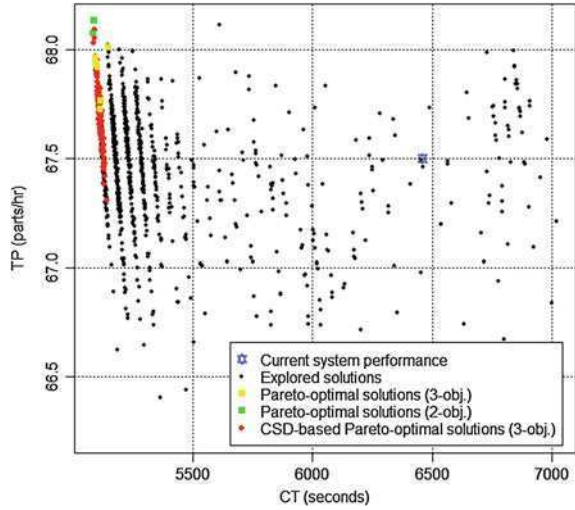
### 15.4.2 Results from SMO

All SMO results presented in this paper are generated by using the integrated EMO algorithm in FACTS Analyzer with the optimization ran on the OPTIMISE platform [38]. Specifically, the algorithm used for all the SMO runs is MA-NSGA-II, as a variant of NSGA-II, which uses artificial neural networks to filter out candidate solutions which are likely to be inferior in the EMO process and CSD to handle simulation output noise. SMO runs have been carried out with three optimization objectives (maximise TP, minimise CT and minimise WIP) that subject to the constraints of the 12 decision variables, e.g.,  $N_p \in [100, 135]$ . Figures 15.5 and 15.6 show the scatter plot of the explored solutions in the CT-TP space and WIP-CT space, respectively. The current system performance is also plotted for the ease of comparison. By using an ordinary NDS, Pareto-optimal solutions from SMO with both the 3 objectives and 2 objectives (maximising TP and minimising CT), are highlighted as yellow and green squares in Fig. 15.5. We put focus on the Pareto front explored in the 3-objective SMO run and call it PF. On the other hand, the red dots in Figs. 15.5 and 15.6 are Pareto-optimal solutions generated with the CSD-based NDS, which was mentioned earlier in this paper.

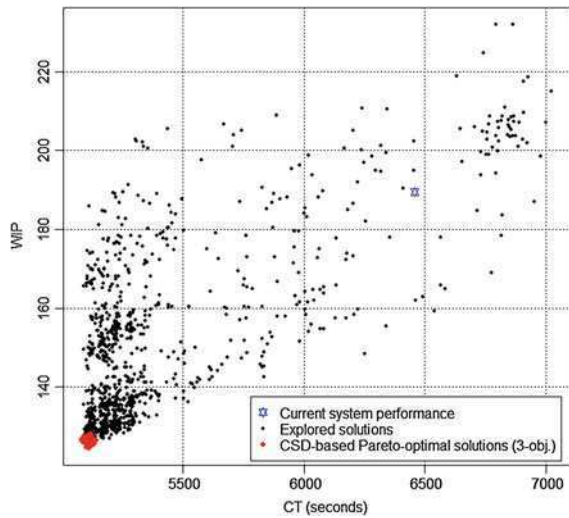
The large number of CSD-based Pareto-optimal solutions generated from the SMO (see Fig. 15.5) has indicated that there exist many configurations which can produce the same level of system performance. In the following, a Pareto front generated with the CSD technique will be represented as  $PF_s$ .

Are the  $PF_s$  solutions significantly better than the current system configuration? Statistical confidence interval tests [39] with 99 and 97.5% confidence have been conducted to verify the difference between the current system performance and the outputs of the solution which produces the best TP in PF (the topmost green square in Fig. 15.5). Test results are listed in Tables 15.1 and 15.2 with TP, CT and WIP

**Fig. 15.5** SMO explored solutions in the CT-TP space



**Fig. 15.6** SMO explored solutions in the CT-WIP space



of the current system performance denoted as  $TP_0$ ,  $CT_0$  and  $WIP_0$  in contrast with the optimal values  $TP'$ ,  $CT'$  and  $WIP'$ , respectively.

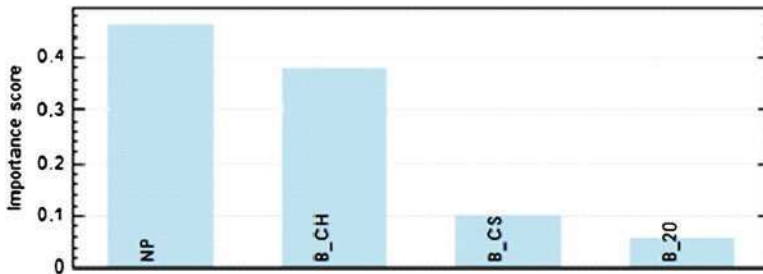
While significant reduction in CT and WIP can be verified by the negative ranges detected in the difference tests, the hypothesis that  $(TP' - TP_0) > 0$  (i.e.,  $TP'$  is higher than  $TP_0$ ) has to be rejected if the confidence level selected is 99% (99.33% is required if the confidence level for each objective has to be 95%, according to the Bonferroni inequality [39]). This implies that significant improvement on the current system performance can be made, in terms of CT and WIP, but not TP, by adjusting the decision variables.

**Table 15.1** Difference test with 99% confidence

$TP' - TP_0$	$CT' - CT_0$	$WIP' - WIP_0$
$[-0.0135, 1.3385]$	$[-1429.6, -1312.9]$	$[-69.74, -57.18]$

**Table 15.2** Difference test with 97.5% confidence

$TP' - TP_0$	$CT' - CT_0$	$WIP' - WIP_0$
$[0.0887, 1.2363]$	$[-1420.4, -1321.0]$	$[-68.72, -58.2]$



**Fig. 15.7** Importance scores of the PF classification analysis

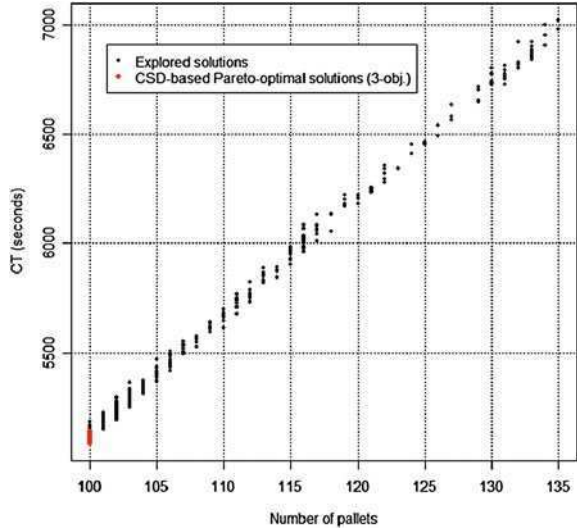
While the SMO results have illuminated that there exist many possible configuration settings which may outperform the current system with respect to CT and WIP. The basic question on what factors are contributing the ‘optimal’ performance cannot be answered by simply examining the objective space visually (e.g., looking at the CT-TP plot). In order to identify the key influencing decision variables and their optimal values as well as exploring their underlying relationships, DM methods have been applied to extract knowledge from the SMO data sets, as suggested by the SBI-DM procedure.

### 15.4.3 Results from the SBI-DM

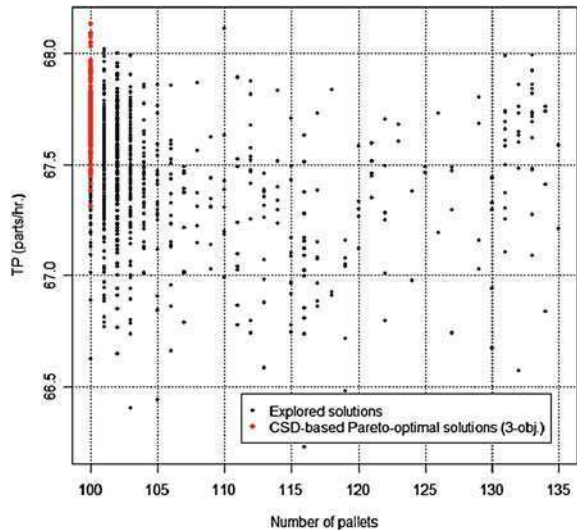
The SBI-SM procedure has been applied to identify if there are any key factors and their relationships on contributing the solutions to lie on the Pareto front. In this case, the output data is classified into only two groups: solutions lying on the  $PF_s$  ( $PF_s = \text{TRUE}$ ) and those that are not ( $PF_s = \text{FALSE}$ ). The results obtained in this analysis are of surprisingly high accuracy and usefulness—the two most important quality measures in any data mining procedures.

Figure 15.7 is the importance score chart of the PF classification, showing the key contributing factors for  $PF_s$  solutions, in the order of importance:  $N_p$ , capacity of  $B_{CH}$  (the buffer for cylinder head),  $B_{CS}$  (the buffer for camshaft) and  $B_{20}$  (the buffer for OP20). There are five rules generated in the decision tree, with accuracy

**Fig. 15.8** Linear correlation between  $N_p$  and CT; all PFs solutions lie on  $N_p = 100$



**Fig. 15.9** Plotting  $N_p$  again set TP; all PFs solutions at  $N_p = 100$

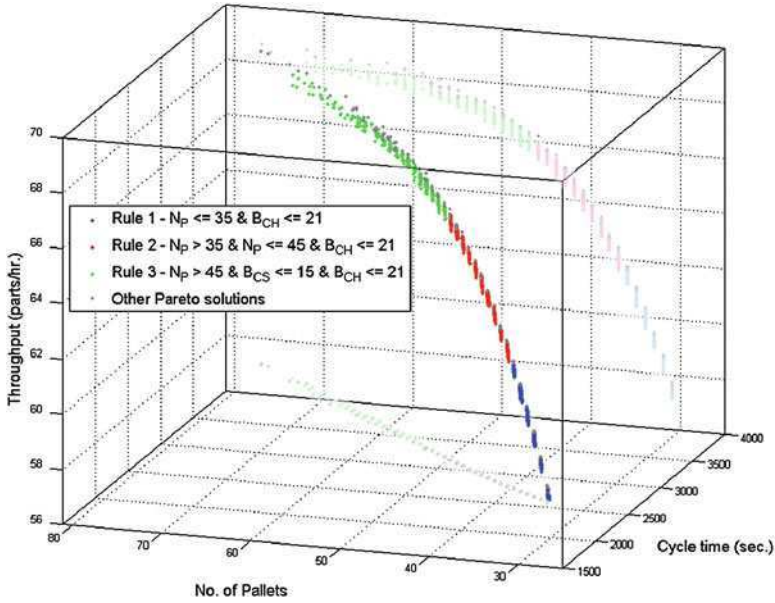


99.083% and total AUC = 0.995, indicating very high-prediction accuracy. Following are the two most important rules extracted from DM:

Rule 1: IF  $N_p > 100$  THEN  $PF_s = \text{FALSE}$

Rule 2: IF  $N_p = 100$  AND  $B_{20} > 8$  AND  $B_{CS} \leq 11$  AND  $B_{CH} \leq 22$  THEN  $PF_s = \text{TRUE}$

Rule 1 can be verified by simply plotting  $N_p$  against CT and TP using the solutions explored in the SMO, as shown in Figs. 15.8 and 15.9, respectively. While a linear correlation between  $N_p$  and CT may be conjectured, there appear to



**Fig. 15.10** 3-D scatter plot for observing the correlation between  $N_p$ , TP and CT

be no direct relationships between  $N_p$  and TP. The most interesting insight that can be gained from these two data plots is that all  $PF_s$  solutions have one common attribute:  $N_p = 100$ , which is consistent with the rules extracted from data mining.

**15.4.4 SBI-DM Results from Expanded Pallet Range**

The first set of SBI-DM results has made a clear indication that the number of pallets can be reduced because (1) all Pareto solutions lie on  $N_p = 100$ , the lower limit during the SMO experiment; (2) no direct relationship between  $N_p$  and TP simply indicate the range of  $N_p$  that determine the TP is much lower than the tested range [100, 135].

Contrary to the original intention of the production engineers to verify the effect of increasing  $N_p$ , a new SBI-SM run was carried out with  $N_p \in [30, 135]$  to test the effect of significantly reducing  $N_p$ . The SBI-DM results presented in form of colour-coded 3-D plot, is shown in Fig. 15.10. Note, there are only three rules generated from RDS that have confidence over 0.7 (see Table 15.3), the threshold we set for this analysis required by the SBI-DM procedure. Several important observations can be made with the 3-D data plot and the rules extracted:

- CT correlates linearly with  $N_p$ , as indicated by the shadow on the CT- $N_p$  plane.
- While TP increases proportionally with  $N_p$  in the range [30, 90], TP stops to increase when  $N_p > 90$ . The maximum TP that can be attained by the system is

**Table 15.3** Support and confidence for DM generated rules

Rule	Supp. count	Conf. count	Total no.	Support	Confidence
$N_p \leq 35$ and $B_{CH} \leq 21$	52	53	3232	0.016	0.981
$N_p > 35$ and $N_p \leq 45$ and $B_{CH} \leq 21$	53	73	3232	0.016	0.726
$N_p > 45$ and $B_{CS} \leq 15$ and $B_{CH} \leq 21$	54	73	3232	0.017	0.740

68 parts/h, with the optimal buffer setting at  $N_p = 80$ . Further increasing  $N_p$  will not improve TP as intended but only elevate CT and WIP.

- Apart from the effect of  $N_p$ , Pareto-optimal solutions, particularly those in the high TP region, illustrated by the green data points, have the common properties of limited buffer level in  $B_{CS} (\leq 15)$  and  $B_{CH} (\leq 21)$ .

To further investigate the key influencing factors for TP and WIP, two DM predictive models have been built using the same data set from the SMO of expanded pallet range. Figures 15.11 and 15.12 shows the decision trees and importance scores generated for TP and WIP analysis, respectively. While the decision tree and importance scores generated for the TP simply confirm the effects of  $N_p$  on TP, the decision tree and importance scores for WIP have given some additional information:  $B_{CS}$  and  $B_{CH}$  are the key influencing factors for WIP. This explains why it is important to keep these two buffers at their optimal level if the optimal trade-off between TP and WIP are desired.

### 15.4.5 Analyzing the Effects of Process Improvement Using SBI-DM

If neither buffer optimization nor increasing  $N_p$  is the right approach to improve the TP of the line, what can be done to do so? Following the philosophy introduced in the theory of constraints (TOC) [40], we use FACTS Analyzer to locate the bottleneck (or constraint, using the terms of TOC) that restrains the current production rate of the line. FACTS Analyzer provides two output data plots for doing this: (1) utilization (% working) and (2) the more advanced shifting bottleneck detection method developed by researchers at Toyota [11].

Because of its low reliability (currently 80% based on the data collected by the company), OP10 at the beginning of the line is believed to be the bottleneck, not by measuring the machine’s working utilization but in terms of the total active time,<sup>2</sup> as listed in Table 15.4. Two more SMO runs were conducted to investigate

<sup>2</sup> Defined in the shifting bottleneck detection method [41], active time is the time when the machine is working, changing tools or in repair and inactive time includes starving, blocked or waiting.

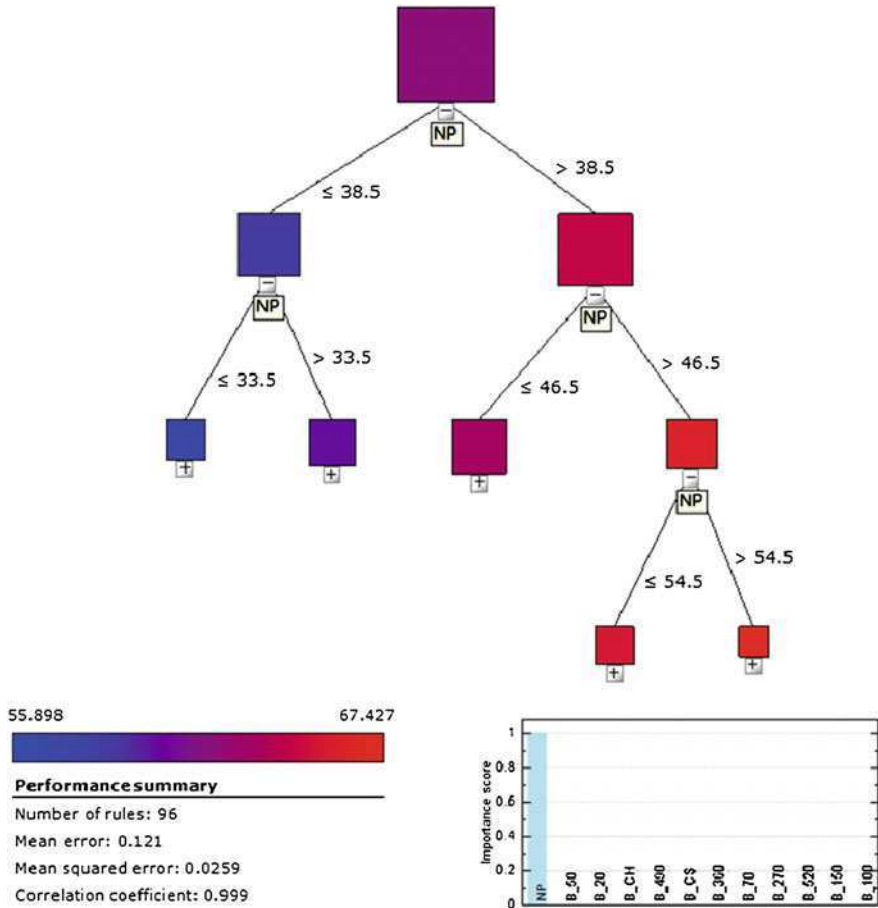


Fig. 15.11 Decision tree and importance score for TP analysis for  $N_p = [30, 135]$

the effect of improving OP10 on the objective space by making the following changes to the original FACTS model: (1) availability of OP10 changed from 80 to 95%; (2) availability of OP10 altered to 95% as well as reducing its processing time by 7%. CSD-based Pareto fronts generated in these two SMO runs are denoted as  $PF_{s1}$  and  $PF_{s2}$ , respectively. Figure 15.13 shows the data plots to compare  $PF_s$ ,  $PF_{s1}$  and  $PF_{s2}$ . As the Pareto-optimal solutions of the improved line (either  $PF_{s1}$  or  $PF_{s2}$ ) significantly outperform  $PF$  of the existing line in the CT-TP plot, it may be concluded that improving the availability of OP10 has produced improvement to the production rate of the whole line. On the other hand, it is interesting to observe that solutions in  $PF_{s1}$  and  $PF_{s2}$  are non-inferior to each other. This implies further improving the workstation OP10 by cutting its processing time will not give any additional enhancement to the performance of the whole line.

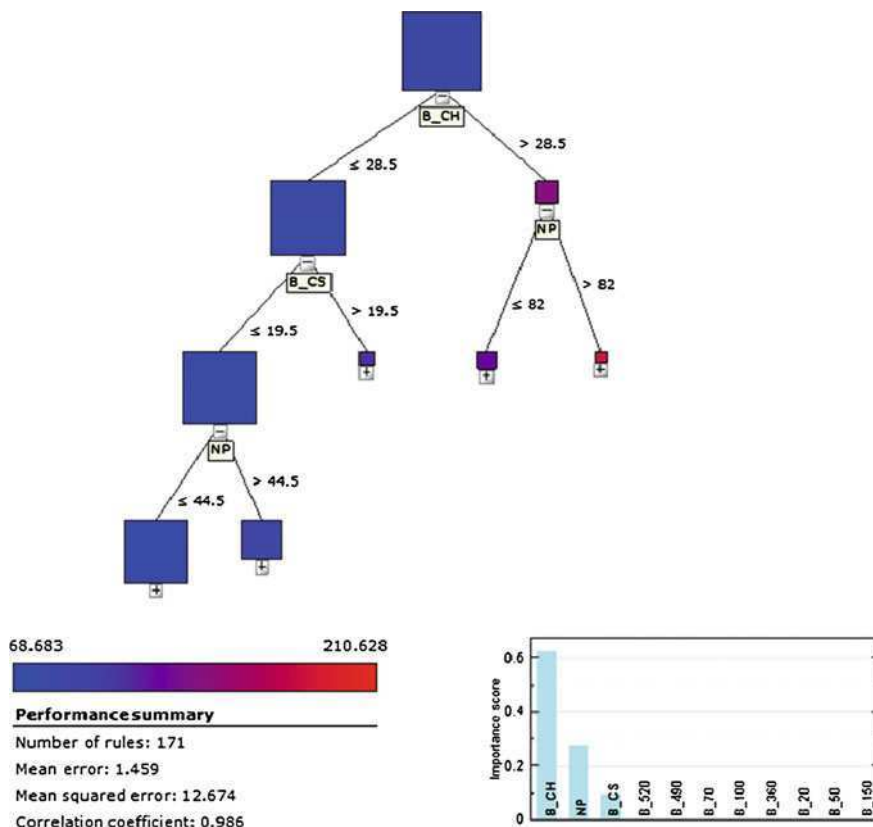


Fig. 15.12 Decision tree and importance score for WIP analysis for  $N_p = [30, 135]$

Table 15.4 Bottleneck analysis

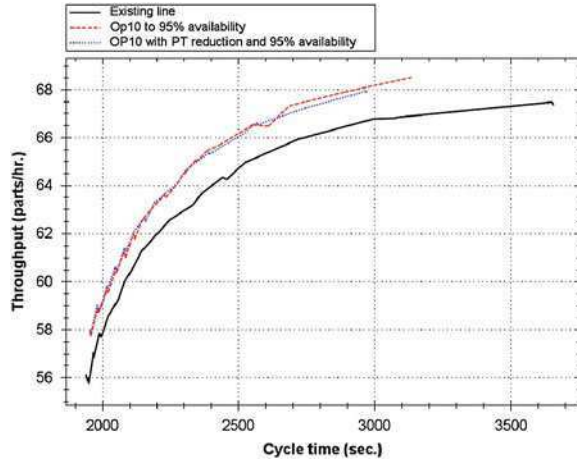
Workstation	Utilization (working %)	Sole (%)	Sole + Shifting (%)
OP10	76.85	39.24	64.66
OP600	80.57	15.99	41.82
OP430	81.68	2.277	9.975
OP510	80.59	1.504	5.973
OP340	83.91	0.797	3.81

To summarise, based on these SBI-DM analyses, several important conclusions can be made that will aid the production manager to make decisions on the improvement for the flow line under study:

- The current system is not configured with an “optimal” setting. Significant improvement to the line, in terms of the CT and WIP performance measures, is possible by simply tuning the decision variables, including the number of pallets and major buffer capacities.



**Fig. 15.13** Comparing solutions from  $PF_s$ ,  $PF_{s1}$  and  $PF_{s2}$



- Instead of giving any significant improvement in the system throughput, increasing the number of pallets can only elevate CT and WIP. As a matter of fact, there is a linear correlation between the number of pallets and CT. In contrary to the original intention of the production engineers to increase the number of pallets, reducing the number of pallets to the level that produce the maximum production rate and optimal trade-off with CT and WIP is highly recommended.
- While the key influencer of the TP and CT is the number of pallets, to have the best trade-off between TP and WIP, it is recommended by the SBI-DM analysis that  $B_{CH}$  should be  $<22$  and  $B_{CS} < 16$ .
- The production line can be effectively increased by improving the availability of the current bottleneck workstation, OP10, from 80 to 95%. Interestingly, further reducing the processing time of OP10 by 7% does not produce additional improving effect as observed by comparing the CT-TP plots generated by the SMO runs.

## 15.5 Conclusions and Outlook

By integrating the concept of innovization with simulation and DM techniques, a new set of powerful tools has emerged for production systems design, analysis, optimization as well as improvement. As a method for retrieving the relationships between input parameters (decision variables) and output parameters (performance measures) for production systems, SBI-DM can be regarded as a methodology applied for continuous improvements within the context of Lean production, TOC and Six Sigma as well. While this claim needs to be proved with more scientific research in the future, this paper has presented a successful case study, carried out in an automotive manufacturer, on how good design/operational principles can be

retrieved by following the SBI-DM procedure. Useful insights have been gained from the decision trees generated by applying DM technologies to the Pareto-optimal solutions acquired from SMO, as suggested in the SBI-DM procedure. The extracted knowledge is valuable to aid the decision-maker to make the right decisions if the production line at the company has to be improved.

Although in this chapter we focus on stochastic production simulations using DES, it is believed such a SBI-DM procedure is also applicable for other stochastic simulations in other areas, like supply chain, transportation, etc. Specifically, applying SBI-DM to supply chain simulation models developed with system dynamics, has been found to be promising to decipher insights for real-world complex supply chain networks which cannot be obtained with other classical optimization/mathematical methods. On the other hand, the optimization problem addressed in this paper is in fact a simple one. Some industrial case studies involving more complex combination of product mix, buffer allocation, workload allocation/balancing, dispatching/sequencing/scheduling logic, flow control as well as quality control policy are now underway and will be published in the future.

**Acknowledgments** Results presented in this case study are based on parts of the research outcomes of the Factory Analyses in ConcepTual phase using Simulation (FACTS) project (2006–2008) and the FFI-HSO (Holistic Simulation Optimization) project (2009–2012). The authors gratefully acknowledge VINNOVA, Sweden, for the provision of research funding for these two projects.

## References

1. Chryssolouris, G. (1992). *Manufacturing systems: Theory and practice*. New York: Springer.
2. Wu, B. (1992). *Manufacturing systems design and analysis* (2nd ed.). London: Chapman and Hall.
3. Cochran, D. S., Arinez, J. F., Duda, J. W., & Linck, J. (2002). A decomposition approach for manufacturing system design. *Journal of Manufacturing Systems*, 20(6), 371–389.
4. Goldratt, E. M. (1991). *Haystack Syndrome*. Great Barrington, MA: North River Press.
5. Deb, K., & Srinivasan, A. (2006). Innovization: Innovating design principles through optimization. *Proceedings of the Genetic and evolutionary Computation Conference (GECCO-2006), The Association of Computing Machinery (ACM), New York*, (pp. 1629–1636).
6. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (3rd ed.). Wiltshire: Wiley.
7. Ng, A. H. C., Urenda, M., & Svensson, J. (2008). Multi-objective simulation optimization for production systems design using FACTS analyzer. *Proceedings of the 2<sup>nd</sup> Swedish Production Symposium (SPS'08), Stockholm*, November 18–20, 2008.
8. Bandaru, S., & Deb, K. (2010). Automating discovery of innovative design principles through optimization. KanGAL Technical Report No.2010001.
9. Dudas, C., Ng, A. H. C., & Boström, H. (2008). Knowledge extraction in manufacturing using data mining. *Proceedings of the 2<sup>nd</sup> Swedish Production Symposium (SPS'08), Stockholm*, November 18–20, 2008.

10. Rokach, L., & Maimon, O. Z. (2008). *Data mining with decision trees: Theory and applications*. Hackensack, NJ: World Scientific.
11. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39, 27–34.
12. Han, J., & Kamber, M. (2004). *Data mining: Concepts and techniques* (7th ed.). San Francisco, Calif: Kaufmann.
13. Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1992). Knowledge discovery in databases: An Overview. *AI Magazine*, 13, 57–70.
14. Vedder, A. (1999). KDD: The challenge to individualism. *Ethics and Information Technology*, 1, 275–281.
15. Sumathi, S., & Sivanandam, S. N. (2006). *Introduction to data mining and its applications*. Berlin: Springer.
16. Pachón, V., Jacinto, M., & Maña, M. J. (2009). Practical application of a KDD process to a sulphuric acid plant. In S. Omatu (Ed.), *Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living: Proceedings of the 10<sup>th</sup> International Work-Conference on Artificial Neural Networks, IWANN 2009 Workshops, Salamanca, Spain, part II*, June 10–12, 2009. Berlin: Springer.
17. Liu, L., & Özsu, M. T. (2009). Encyclopedia of database systems. <http://dx.doi.org/10.1007/978-0-387-39940-9>.
18. Weiss, S. M., & Indurkha, N. (2002). *Predictive data mining: A practical guide*. San Francisco, CA: Morgan Kaufmann.
19. Neckel, P., & Knobloch, B. (2005). *Customer relationship analytics: Praktische anwendung des data mining im CRM* (1st ed.). Heidelberg: dpunkt-Verl.
20. Kohavi, R., & Provost, F. (1999). Glossary of terms. *Machine Learning* 30, 271–274, <http://ai.stanford.edu/~ronnyk/glossary.html>.
21. Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York, NY: Wiley.
22. Groth, R. (1998). *Data mining: a hands-on approach for business professionals*. Upper Saddle River, NJ: Prentice-Hall.
23. Bissantz, N., & Hagedorn, J. (2009). Data mining. *Business & Information Systems Engineering*, 1, 118–122.
24. Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38, 1–32.
25. Nießen, J. (2010). Discovering knowledge from simulation-based evolutionary multi-objective optimization through data mining. MSc dissertation, School of Informatics and Communication, University of Skövde, Sweden.
26. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, 6(2), 181–197.
27. Ng, A. H. C., Syberfeldt, A., Grimm, H., & Svensson, J. (2008). Multi-objective simulation optimization and significant dominance for comparing production control mechanisms. *Proceedings of the 18<sup>th</sup> International Conference on Flexible Automation and Intelligent Manufacturing (FAIM'08)*, Skövde, Sweden (pp. 1210–1219).
28. Joseph, V. R., & Ying, H. (2008). Orthogonal-maximin Latin hypercube designs. *Statistica Sinica*, 18, 171–186.
29. Liu, A., Ghosh, J., & Martin, C. (2007). Generative oversampling for imbalanced datasets. *Proceedings of the 3<sup>rd</sup> International Conference in Data Mining* (pp. 66–72).
30. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 341–378.
31. Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1–2), 105–139.
32. Rule Discovery System, v. 2.6.0, Compumine AB. Retrived April 2010, from <http://www.compumine.com/web/public/rds>.
33. Lin, W., Alvarez, S. A., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6, 83–105.

34. Ishibuchi, H., Kuwajima, I., & Nojima, Y. (2008). Evolutionary multiobjective rule selection for classification rule mining. In A. Ghosh, S. Dehuri, & S. Ghosh (Eds.), *Multi-objective evolutionary algorithms for knowledge discovery from databases*. Berlin: Springer.
35. Little, J. D. C. (1992). Are there 'Laws' of manufacturing. In J. A. Heim, & W.D. Compton (Eds.), *Manufacturing systems: Foundations of world-class practice*. Washington, DC: National Academy Press (pp. 180–188).
36. Hopp, W. J., & Spearman, M. L. (2000). *Factory physics: foundations of manufacturing management* (2nd ed.). Burr Ridge, IL: Irwin McGraw-Hill Higher Education.
37. Spearman, M. L., Woodruff, D. L., & Hopp, W. J. (1990). CONWIP: A pull alternative to Kanban. *International Journal of Production Research*, 28(5), 879–894.
38. Ng, A.H.C., Grimm, H., Lezama, T., Persson, A., Andersson, M., & Jägstam, M. (2008). OPTIMISE: An internet-based platform for metamodel-assisted simulation optimization. In: X. Huang, Y-S. Chen, & S-L. Ao (Eds), *Recent Advances in Communication Systems and Electrical Engineering*. Heidelberg: Springer (pp. 281–296).
39. Law, A. M., & Kelton, W. D. (2000). *Simulation Modeling and Analysis* (3<sup>rd</sup> ed.). New York: McGraw-Hill Higher Education.
40. Goldratt, E. M., & Cox, J. (1986). *The goal: a process of ongoing improvement* (revised edition ed.). Croton-on-Hudson, NY: North River Press.
41. Roser, C., Nakano, M., & Tanaka, M. (2002). Shifting bottleneck detection. *Proceedings of the 2002 Winter Simulation Conference, San Diego, CA, USA* (pp.1079–1086).

# Chapter 16

## Multi-objective Production Systems Optimisation with Investment and Running Cost

Leif Pehrsson, Amos H. C. Ng and Jacob Bernedixen

**Abstract** In recent years simulation-based multi-objective optimisation (SMO) of production systems targeting e.g., throughput, buffers and work-in-process (WIP) has been proven to be a very promising concept. In combination with post-optimality analysis, the concept has the potential of creating a foundation for decision support. This chapter will explore the possibility to expand the concept of introducing optimisation of production system cost aspects such as investments and running cost. A method with a procedure for industrial implementation is presented, including functions for running cost estimation and investment combination optimisation. The potential of applying SMO and post-optimality analysis, taking into account both productivity and financial factors for decision-making support, has been explored and proven to be very beneficial for this kind of industrial application. Evaluating several combined minor improvements with the help of SMO has opened the opportunity to identify a set of solutions (designs) with great financial improvement, which are not feasible to be explored by using current industrial procedures.

---

L. Pehrsson (✉)  
Volvo Car Corporation, 405 31 Göteborg, Sweden  
e-mail: leif.pehrsson@his.se

L. Pehrsson · A. H. C. Ng · J. Bernedixen  
Virtual System Research Centre, University of Skövde,  
541 28 Skövde, Sweden  
e-mail: amos.ng@his.se

J. Bernedixen  
e-mail: jacob.bernedixen@his.se

## 16.1 Introduction

It is widely accepted that simulation is the only general purpose and generally applicable modelling tool for truly complex systems [1]. Particularly, it is often said that discrete-event simulation (DES) is the most promising tool to support decision-making in production systems design and analysis. While DES can be used to test various scenarios under given sets of parameters in order to evaluate the specific solutions performance, trying to find optimal solutions using DES would many times require unrealistic effort of time. Simulation-based optimisation (SBO), the technology that connects meta-heuristics search methods to simulation models, can be used to address such issues [2]. SBO for production system and production line analysis has evolved to incorporate multi-objective optimisation (MOO), using genetic algorithms [3]. An example of utilising MOO for production system analysis is FACTS Analyser, an Internet-enabled SBO toolset developed specifically for factory design, analysis and optimisation in the conceptual phase [4].

The integration of SBO and MOO, or Simulation-based Multi-objective Optimisation (SMO), has opened the opportunity to find the optimal or near-optimal solutions considering several objectives within certain constraints. So far SMO, applied on production systems, has been used in targeting traditional production system objectives such as throughput, work-in-process (WIP) and lead time. The industry often relies on lean methods to solve production issues [5] and lean is a necessary but not sufficient approach for analysing production system issues [6]. In combination with post-optimality analysis, the concept of SMO has the potential of creating a foundation for decision support, introduced by Deb using the term “innovization”, meaning the task of creating innovative design principle through optimisation [7, 8].

According to international good practice guidance for accounting, investment (project) appraisals and capital budgeting, involving the assessment of a project's financial feasibility should use Discounted Cash Flow (DCF) analysis as a supporting technique in order to compare costs and benefits in different time periods and to estimate the net present value (NPV) [9]. This is dependent on an estimation of the expected cash flows related to the investment or the project, the life of the investment and the opportunity cost of investing in a project of similar risk profile. It is also considered that costing contributes to an understanding of profits and value creation and the efficiency and effectiveness of input to output transformation in an operational process [10]. Costing for decision support is also useful for the improvement of performance, value creation, scenario analysis and the effective and efficient application of resources and processes within an enterprise [11]. In order to support financial decision-making, the traditional simulation objectives might not be relevant to provide the needed information on, e.g., cash flow, without additional calculations. However, there are examples of the merger of DES and methods for cost estimation connected to activity-based costing (ABC) [11] and the use of DES as the means for cost reduction and performance improvement [12].

Some cost models, such as cost deployment [13] and a general economic model for manufacturing cost simulation [14], require in-depth data mapped bottom-up in the production system. There is also an approach to overcome pitfalls due to data complexity by using capacity cost rate within time-driven ABC [15]. The concept of measuring the cost of capacity [16] includes several tools and techniques within a framework for analysing capacity cost management issues, e.g., the time-based resource effectiveness model, and the combination of operational and financial data creates a strong basis for analysis and decision-making that can be merged with capital investment models. The time-based analysis results can be transferred to financial data with a process costing model. The resource effectiveness model shows similarities to loss models used in industry [5, 17], and is corresponding to industrial production data.

The aim of this chapter is to propose a novel SMO-based decision-making support method for production systems design and/or improvement. Such a method is based on the incorporation of investment and running cost modelling into SMO so that the best trade-off between cost and other productivity measures can be sought and analysed effectively. The method is fully tested in an industrial case study within automotive industry with very promising results. The rest of this chapter is organised as follows: Sect. 16.2 covers a general overview of the cost optimisation method. The running cost and investment functions are presented in Sects. 16.3 and 16.4, respectively. Sections 16.5 and 16.6 introduce the simulation method and optimisation objectives supporting the cost-based SMO-framework, respectively. The context, experimental setting, results and analyses of the industrial case study are presented in Sect. 16.7. Conclusions are given in Sect. 16.8.

## 16.2 The Cost Optimisation Method

Production efficiency is one paramount factor for the survival of any industrial companies, particularly for those in the automotive industry wherein competition becomes more and more intensified. The implementation of SMO could be one contributor of great importance to enhance the production efficiency within the automotive industry as the technique has been proving very promising [7].

Decision-making in industry is to a great extent based on financial information due to the fierce competition. However, decision-making regarding production systems on financial basis may not always be based on a satisfying analysis of available options and data. How do we find the best trade-off between investment and running cost? Is it possible to satisfy customer demand and reduce production cost simultaneously with less investment than planned? What about buffer allocation and complete system cycle time in combination with various investment alternatives? One way of trying to answer such questions and improve decision-making within industry would be to combine the optimisation of traditional production system objectives with financial objectives. Doing that shall enable finding Pareto-optimal or near Pareto-optimal solutions or trade-offs between conflicting

financial objectives and production system properties. In that way, decision-making could benefit from a transparent, well-prepared analysis of the interaction between decision parameters. It will also help the decision-maker to select solutions from an area of the objective space that provides the best known trade-off between conflicting objectives. Another advantage is that relating the near-optimal solutions back to the parameter settings opens the opportunity to find knowledge about the analysed system. Are there any specific patterns in the optimised data that we can benefit from? Are some of the available options more important than others? Is there a specific order in which to introduce updates in the production system that results in a better fulfilment of the objectives during implementation? Any decision-maker would probably benefit from having answers to this kind of questions.

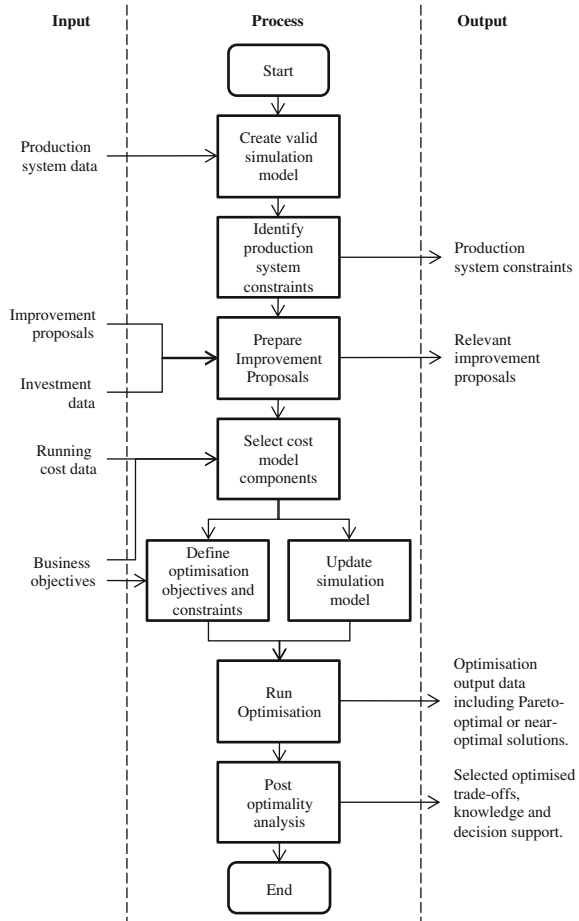
In order to meet the need for enhanced decision-making within manufacturing operations, theories required to create method for SMO applied on production systems including cost aspects as investment against running cost has been developed. An important part of the development has been to formulate functions for mapping production system properties into the financial domain by following a standard cost model. Another part has been integration with simulation and post-optimality analysis. The preparation of decision scenarios has also been given some analyses, resulting in an opportunity to identify the major constraints of the production system. The idea is to prepare the prerequisites for a SMO-framework targeted at decision-making within manufacturing management and manufacturing engineering.

Since different techniques and tasks are used as the foundation of the method, it is vital to arrange them in a process flow. By doing so, the method can be automated to a certain level in order to facilitate industrial implementation. The major inputs and outputs connected to the process are illustrating the methods requirements and its resulting outcomes. Decision-support information is created in several steps and input can be adjusted accordingly, indicating that the method should be applied as an interactive flow of events. That is, some decisions might be made already during the course of the analysis and not after it is finished. Sometimes, it might be enough to identify the major constraints of the production system and some relevant improvement proposals to make an acceptable decision. In other cases, simple plots of the optimised Pareto front can provide enough information for making decisions. However, when there are complex relations and several dimensions due to many objectives, or when extended new knowledge is required, post-optimality analysis may be necessary to be applied to reveal sufficient information to the decision-maker. Depending on the decision situation, the process may be exited when suitable output information is acquired. The main process steps with their input and output are described in Fig. 16.1.

The initial input to the process is production system data mainly consisting of processing times, availability or up-time, mean time to repair (MTTR)—in this case equivalent to mean down time (MDT)—and buffer capacity. Later on improvement proposals, investment data, running cost data and business objectives are required when preparing for optimisation.



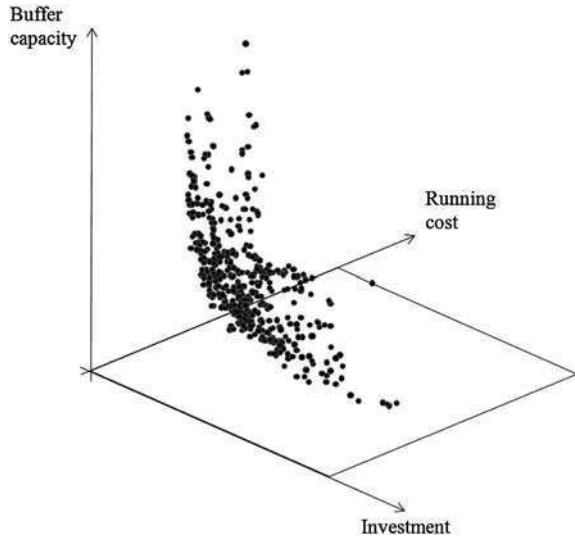
**Fig. 16.1** The cost optimisation process



The initial input in combination with information about the flow is used to create and validate a simulation model of the production line. The first task to perform with the model is to identify the major constraints of the production system as input to the selection of relevant improvement proposals. Investment data is connected to the selected improvement proposals and introduced as input parameters to the simulation model. Running cost data is collected and cost model components are selected depending on the scenario to be studied, the business objectives and the characteristics of the improvement proposals. The simulation model is updated and prepared for optimisation with objectives and applicable constraints. After choosing optimisation algorithm and setting of related parameters the actual optimisation phase can be started.

In order to evaluate certain patterns in the resulting data, especially among optimal or near-optimal non-dominated data, post-optimality analysis is conducted e.g., by sorting and data mining. The result is a selection of the optimised trade-

**Fig. 16.2** Three-dimensional plot of optimisation results



offs, knowledge and decision support to be used for production system design and investment decision-making. Initial validation has been carried out by application of the method in a real-life industrial case described later in this chapter.

In case of complex relations within the optimisation problem and when there are several dimensions to consider due to many objectives, post-optimality analysis might be required in order to produce sufficient decision support. It is also likely that such analysis can reveal hidden information within the optimised data and create new knowledge according to the concept of innovization [7]. Suggested methods to be used depend on the situation but some general methods such as sorting, clustering and/or data mining using decision trees are some examples. Simple visual analysis method like colour-coding of results can be a generic method to visualise decision parameters.

Combinations of analysis tools and presentation techniques may reveal useful information to the decision-maker. Three-dimensional plotting and colour-coding enables presentation of four dimensions and clustering of data based on data mining might help to highlight areas of interest. An example of three-dimensional plotting of optimised data is found in Fig. 16.2. Some results from innovization might be possible to present as design principles valid for a certain type of system.

### 16.3 Running Cost Estimation

There are several cost models available for production system analysis and improvement. Examples of such models are cost deployment [13] and a general economic model for manufacturing cost simulation [14]. There are also examples

of merging DES with ABC in order to consider process variation [11]. However, these methods are very detailed requiring in-depth bottom-up data mapping in the production system. On many occasions the available time span or lead time to analyse the options at hand and make a decision is strictly limited due to project management systems and time-to-market requirements. In the situation of conceptual production system design and improvement prioritisation in industry, the detailed models tend to be far too impractical to be used.

The above-mentioned methods, cost deployment and the general model for manufacturing cost simulation, were considered to be used during the development of the cost modelling method. After trying to collect data for case studies the conclusion was that data with the very detailed level required by these methods was rare to be found in industry, especially within a time frame coherent with the decision-making process demands. The conclusion is that in this case a fast, accurate model with reduced data complexity is needed.

Are there then any alternatives available that can meet the speed and data quality requirements, if we consider a need for constantly updated models with sufficient accuracy to make the right decisions? A time-based method like the resource effectiveness model [16] can be used as an information source that can easily be translated into financial figures through a process costing model defined on process time rather than units of production. This kind of model also corresponds to loss models used in industry [5, 17]. The industrial loss models are describing the utilisation of production time and man-hours focusing on providing information on how much of the available resources are used for value adding activities. Another interesting factor is that many of the key performance indicators used within an industrial loss model are used as sources for decision-making. In addition to time utilisation these kinds of models also focus on providing information for minimisation of material and energy consumption.

A similar approach for ABC, referred to as time-driven ABC, is described by Kaplan and Andersson [15] in order to overcome pitfalls due to data complexity by focusing on capacity cost rate. The capacity cost rate can be calculated as:

$$CCR = \frac{CCs}{PCrs} \quad (16.1)$$

where,  $CCR$  = capacity cost rate,  $CCs$  = cost of capacity supplied and  $PCrs$  = practical capacity of resources supplied.

### 16.3.1 Running Cost Function

Would it be possible to compose a running cost function for production system optimisation relying on time-based cost theories? In order to answer this question and try to estimate the effect of improvements from a cost perspective, a function for translating the effect of production line performance and production system

improvements to cost has been composed, which is derived from the resource effectiveness model and time-driven ABC. One basic principle for the suggested cost model is that reference data from a current running production system is used as the initial cost base. Then differences appearing due to the introduction of various investments or improvement options are added in the form of delta costs.

The prerequisite for this approach is to use an aggregated level of information and then add or subtract relevant deltas induced by changes in the model during optimisation. In this case the aggregated information used is the annual running cost for the initial state of the production studied. Annual data is used in order to correspond with forecasts, budget and accounting values. When the method is used for analysis of conceptual phase production systems, an estimation of the initial running cost for the complete system must be made relying on information about the initial production setup. Then the delta effects corresponding to various options and scenarios can be studied and analysed.

In its most basic form, the cost model principle is described by:

$$Cr = Ci + \Delta C \quad (16.2)$$

where,  $Cr$  = running cost per year (annual running cost),  $Ci$  = initial total running cost per year and  $\Delta C$  = delta cost.

Based on the cost of resources for running the production, a cost per hour is calculated and the reduced need of time for production is calculated from the increased throughput, defined step by step below.

$$\Delta Ct = \Delta H \times Ch \quad (16.3)$$

where,  $\Delta Ct$  = throughput delta cost,  $\Delta H$  = difference in need of production time and  $Ch$  = average cost per hour (additional time) corresponding to "capacity cost rate".

$$\Delta H = H - Hi \quad (16.4)$$

where,  $H$  = time required for production of annual production volume,  $Hi$  = initial time required for production of annual production volume.

$$\Delta H = \frac{Vp}{T} - \frac{Vp}{Ti} \quad (16.5)$$

where,  $Vp$  = annual production volume.

$$\Delta H = Vp \left( \frac{1}{T} - \frac{1}{Ti} \right) \quad (16.6)$$

where,  $Ti$  = Initial Throughput and  $T$  = Throughput.

The average cost per hour can be based on several components including energy consumption, coolant consumption, labour costs, etc. This cost is dependent on the operation to be analysed and can be described through a delta throughput to cost function:

$$\Delta Ct = Ch \times Vp \left( \frac{1}{T} - \frac{1}{Ti} \right) \quad (16.7)$$

It is important to remember that a certain  $\Delta Ct$  might only be valid within a specific interval and it might be needed to introduce several  $\Delta Ct$ -functions in order to create a complete model due to constraints in the production setup, e.g., balancing of manning or different costs on various shifts.

$$\Delta Ct_i = Ch_i \times Vp \left( \frac{1}{T} - \frac{1}{Ti_i} \right) \quad (16.8)$$

valid for  $a_i < Ti_i < b_i$

The complete throughput delta cost is then:

$$\Delta Ct = \sum_{i=1}^m \Delta Ct_i \quad (16.9)$$

However there is also likely to be a change in the annual cost for a certain part of the process induced by the selected improvement. Each improvement can have a number of such delta costs attached and could reflect e.g., maintenance cost deltas.

*Delta annual cost function:*

$$\Delta Ca = \sum_{j=1}^n \Delta Ca_j \quad (16.10)$$

There may also be a change in the cost per produced item due to the performed changes, e.g., consumption of cutting tools, welding tips, incoming material and energy consumption.

*Delta cost due to changed cost per produced unit function:*

$$\Delta Cu = \sum_{k=1}^o \Delta Cu_k \times Vp \quad (16.11)$$

The combination of the cost effects induced by an improvement formulates the total running cost function. In order to enable tailored applications, a customised cost component is added to the resulting expression.

*The complete annual running cost function:*

$$Cr = Ci + \sum_{i=1}^m \Delta Ct_i + \sum_{j=1}^n \Delta Ca_j + \sum_{k=1}^o \Delta Cu_k \times Vp + \Delta Cc \quad (16.12)$$

where,  $\Delta Cc$  = user definable custom cost component.

The annual running cost expression in its simplified form becomes:

$$Cr = Ci + \Delta Ct + \Delta Ca + \Delta Cu + \Delta Cc \quad (16.13)$$

## 16.4 Investment Cost Parameters

With traditional industrial methods, it is difficult to estimate and analyse the performance and the resulting running cost induced by combinations of minor investments and variations in various dynamic system parameters of a production system. Based on experience from data collected to case studies within machine-intensive automotive component manufacturing, the common level of industrial data available for scenario description is up-time, processing time and investment cost.

The local effect on, e.g., the processing time achieved by a specific improvement with a defined cost is rather well known within a mature automotive manufacturing organisation. However, in many times, improvements are not introduced due to the inability to analyse and optimise the impact from a combination of activities. The running cost function alone cannot solve this issue without investment data mapped to relevant scenarios in a simulation model. In order to be able to optimise investments and running cost simultaneously, a number of parameters must be incorporated in the simulation model.

The suggested solution is to map and index changes in processing time and up-time to certain objects in the simulation model in a discrete number format. The index number for each improvement linked to an object can then be related to a certain investment and the complete investment can be calculated in order to be used as an optimisation objective.

The impact of investments related to processing time can be written:

$$Ip = \sum_{i=1}^m Ip_i \quad (16.14a)$$

where,  $Ip$  = processing time-related investments.

The impact of investments related to up-time can be written:

$$Iu = \sum_{j=1}^n Iu_j \quad (16.14b)$$

where,  $Iu$  = up-time-related investments.

Another factor with potential effect on the production system performance is buffer capacity and buffer allocation. If there are certain investments related to changes in buffer capacity they can be modelled in the same way as processing time or up-time-related investments:

$$Ib = \sum_{k=1}^o Ib_k \quad (16.15)$$

where,  $Ib$  = buffer capacity-related investments.

In order to enable tailored applications a custom investment component is added to the resulting expression. The complete investment function for integration in a simulation model can be expressed as:

$$I = \sum_{i=1}^m Ip_i + \sum_{j=1}^n Iu_j + \sum_{k=1}^o Ib_k + Ic \quad (16.16)$$

where,  $I$  = total investment,  $Ic$  = user definable customised investment component.

Following is the total investment expression in its simplified form:

$$I = Ip + Iu + Ib + Ic \quad (16.17)$$

## 16.5 Simulation Model

In order to perform optimisation of the financial impact from investments in a production system with the suggested method, a valid simulation model of the production system is required. It is highly recommended that the model abstraction level is aligned with available data and the prerequisites of the decision situation. There are a few questions to answer when considering the model abstraction level. Is the analysis to be made in the conceptual phase, during implementation of production updates or is it made during running production? What level of decision is to be made and which are the main objectives that should be met? Are there any limitations in lead time, available resources or data until the point of decision? Most likely there are such limitations in most situations within common industrial operations.

In order to avoid pitfalls regarding too complex modelling and requirements for very detailed data which are difficult to collect, an easy-to-use modelling environment with built-in data complexity reduction, targeting at the system level of production operations, will facilitate creation of models with a balanced model abstraction trade-off.

### 16.5.1 Simulation Model Integration

During optimisation, the input parameters must be altered in the simulation model in each evaluation iteration. In a SBO application, these parameter changes can be made automatically by integrating an optimisation algorithm to the simulation model. The simulation model integration is performed by enabling setting of scenario data from the optimisation engine depending on new parameter settings generated by the optimisation algorithm. Every object that will be directly affected by investments or improvements is the subject for the connection to a scenario selection. The principle of a scenario table for cost and parameter values is shown in Table 16.1.

The proper parameter value can be selected from the cost parameter value table by adding an optimisation parameter for the actual scenarios to be simulated in specific optimisation iterations. In the simulation software, the parameter value can be set from the table based on an integer value for a specific scenario controlled by

**Table 16.1** Scenario table for an object in the simulation model

Scenario	Parameter $X$	Parameter $Y$	Ip	Iu	Ib	$\Delta Ca$	$\Delta Cu$
0	$A$	$\alpha$	$Ip_a$	$Iu_a$	$Ib_a$	$\Delta Ca_a$	$\Delta Cu_a$
1	$B$	$\beta$	$Ip_b$	$Iu_b$	$Ib_b$	$\Delta Ca_b$	$\Delta Cu_b$
...	...	...	...	...	...	...	...
$n$	$N$	$v$	$Ip_n$	$Iu_n$	$Ib_n$	$\Delta Ca_n$	$\Delta Cu_n$

the optimisation engine. Scenario 0 is corresponding to the initial solution with no improvements applied.

An essential factor to be considered is the presence of any constraints concerning the order in which improvements and changes can be introduced in the production system. Any given scenario might be dependent on the implementation of another scenario.

One way of handling such constraints is to sort the scenarios for a simulation model object before optimisation and define an implementation order. Another way is to include a previous scenario in further scenario definitions. If scenario  $i$  is dependent on the previous scenario then apply scenario  $i - 1$  and scenario  $i$  together.

Together with the scenario settings each buffer capacity in the simulation model may be varied simultaneously by the optimisation engine in order to combine the scenarios with buffer capacity optimisation.

It is not merely a matter of setting values in the simulation model—data must be acquired as well. Output data is read from the model after each simulation run. In order to estimate the resulting running cost for a simulation the throughput is monitored and used as the input to the throughput delta cost component in the running cost function. Before optimisation a reference value for the initial throughput is decided by a simulation representing the initial production status. During optimisation the delta annual costs and the delta costs due to changed cost per produced unit are taken from the scenario tables and summed together in the running cost function.

Shifting bottleneck technique [18] is used for the process step of identifying production system constraints (bottlenecks). This requires the changes in state of all machines in the model to be logged while simulating. Their active periods are divided into bottleneck and non-bottleneck periods and then further into sole and shifting bottleneck periods.

## 16.6 Optimisation Objectives

The main objectives addressed by the cost optimisation method are running cost and investment. In order to evaluate the effect of various improvements and their potential it is also essential to consider optimisation of buffer allocation and buffer capacity.

The first objective is to minimise the running cost function:

$$\min \left( Ci + \sum_{i=1}^m \Delta Ct_i + \sum_{j=1}^n \Delta Ca_j + \sum_{k=1}^o \Delta Cu_k \times Vp + \Delta Cc \right) \tag{16.18}$$



The second objective is to minimise the investment function:

$$\min \left( \sum_{i=1}^m I p_i + \sum_{j=1}^n I u_j + \sum_{k=1}^o I b_k + I c \right) \quad (16.19)$$

Simultaneously with the minimisation of the cost and investment objectives the summation of the capacity of all buffers in the production system is minimised:

$$\min \left( \sum_{i=1}^m B c_i \right) \quad (16.20)$$

where,  $B c_i$  = capacity of buffer number  $i$  in the production system.

Depending on the business goals, there are several other production system properties that might be the optimisation objective in combination with the financial objectives. Some other objectives that might be of interest to combine with the financial objectives are throughput, complete system cycle time, WIP and the number of pallets, in a part of, or in the complete system. In assembly operations, the number of workers in the production teams is an essential parameter to optimise.

## 16.7 Case Study

In order to verify the proposed method, a case study was conducted within the automotive industry. A production line for automotive components with capacity constraints is required to be operated on overtime (e.g., night/weekend shifts) in order to meet the forthcoming increase in customer demand. At the same time, major product changes will be introduced in the line. There are a number of potential improvements with various investments attached that could reduce the capacity constraints. There should be an opportunity to avoid operating the line on overtime. However, there are not enough data for making a decision to invest and reduce the operating time and the cost for labour and production resources.

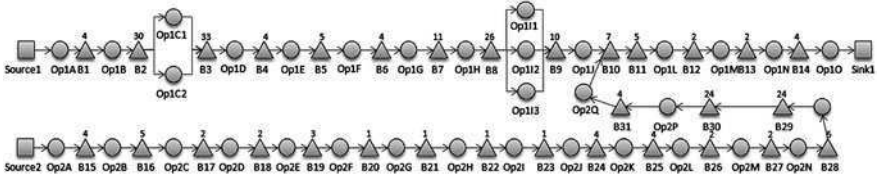
The initially forecasted annual running cost is \$4.9 million (M) and the main objective is to achieve a 20% cost reduction. The challenge is to identify the optimal investment alternatives that can reduce running cost as much as possible, minimise total investment cost for improvement, maximise throughput and simultaneously minimise inter-workstation buffers.

### 16.7.1 Simulation Model and Validation

In order to analyse and optimise the production line a simulation model was created by using FACTS Analyser [4], a software tool developed for supporting factory design, analysis and optimisation during the conceptual design phase.



**Fig. 16.3** Initial simulation model for validation



**Fig. 16.4** Conceptual line configuration model for analysis and optimisation. Allocated buffer capacity is shown in figures above the triangular buffer symbols

Initially the existing production line was modelled for validation against throughput. The model is visualised in Fig. 16.3.

The model shows great correspondence to the real production line with an average model throughput of 17 pieces per hour with a standard deviation of 0.21 in comparison with the real average throughput of 17 pieces per hour. The validation was based on a simulation horizon of 6 days including 1 day warm-up time. Five replications were used to get an indication on the standard deviation.

### 16.7.2 Introduction of Conceptual Changes in the Simulation Model

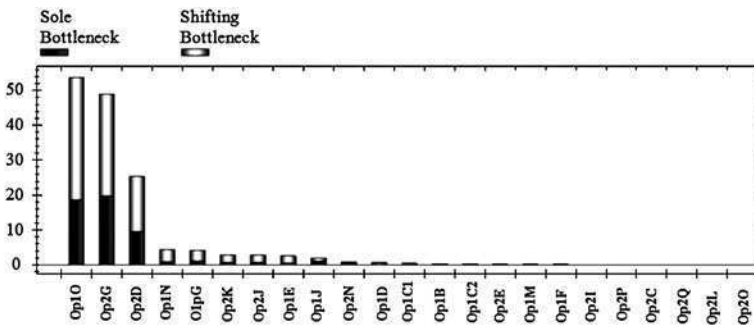
Major changes are planned to be introduced in the studied production line due to product changes and increasing customer demand, including partly parallelised flow and a combination of additional new and re-used equipment as well as removal of old equipment. The conceptual line was modelled in FACTS Analyser and is illustrated in Fig. 16.4.

### 16.7.3 Simulation with Shifting Bottleneck Detection Analysis

In order to predict the performance of the conceptual production line and in order to identify major constraints a simulation with shifting bottleneck detection analysis [18] was carried out using FACTS Analyser. The simulation horizon used was 8 days including 1 day warm-up with 5 replications. The expected throughput performance was approximately 30 pieces per hour and the simulation result was 28.7 pieces per hour with a standard deviation of 0.61 and 189 pieces of average

**Table 16.2** Simulation results

Variable	Mean	Std. dev.
Parts produced	4,814.40	101.89
Throughput	28.66	0.61
Lead time (complete system cycle time)	23,542.36	696.44
WIP	188.56	1.98
S1 (Parts produced)	2,584.40	117.95
S1 (Throughput)	15.38	0.70
S1 (Lead time)	33,756.04	1,687.48
S2 (Parts produced)	2,230.00	27.69
S2 (Throughput)	13.27	0.16
S2 (Lead time)	11,750.50	229.49



**Fig. 16.5** Sole and shifting bottlenecks in the conceptual production line

WIP. The results were also divided into the two connecting parts of the line as seen in Table 16.2, in which *S1* represents products from Source 1 and *S2* represents products from Source 2.

The production volume forecasts indicate that an average throughput of at least 34.5 pieces per hour is required to meet the customer demand. This is where the shifting bottleneck analysis, illustrated in Fig. 16.5, becomes very useful. By analysing the major constraints of the line, precise targeted actions can be suggested in order to improve the production system performance.

### 16.7.4 Production Process Improvement Proposals with Investments

Based on the shifting bottleneck analysis a number of potential improvement proposals were collected from the organisation connected to the line. Emphasis was on the part of the line supplied from source 1 as the throughput potential was considered to be greater compared to the part supplied from source 2. In this case the improvement actions are required to be introduced in a certain order when

**Table 16.3** Relevant improvement proposals

Operation and improvement	0	1	Cost 1 (\$)	2	Cost 2 (\$)	3	Cost 3 (\$)	4	Cost 4 (\$)
<b>Op1E</b>									
<i>Iu</i>	91%	95%	5 443	96%	2 500	97%	4 500	-	-
<i>Ip</i>	145 s	143 s	10 000	133 s	10 000				
<b>Op1G</b>									
<i>Iu</i>	91%	95%	5 000	95.5	2 500	96%	25 000	97%	2 500
<i>Ip</i>	145.3 s	138.3 s	20 000	-	-	-	-	-	-
<b>Op1H</b>									
<i>Iu</i>	-	-	-	-	-	-	-	-	-
<i>Ip</i>	153 s	115 s	2 500	-	-	-	-	-	-
<b>Op1 J</b>									
<i>Iu</i>	-	-	-	-	-	-	-	-	-
<i>Ip</i>	145 s	130 s	20 000	-	-	-	-	-	-
<b>Op1 N</b>									
<i>Iu</i>	93%	95%	5 000	-	-	-	-	-	-
<i>Ip</i>	95 s	80 s	20 000	-	-	-	-	-	-
<b>Op1O</b>									
<i>Iu</i>	-	-	-	-	-	-	-	-	-
<i>Ip</i>	125 s	85 s	20 000	-	-	-	-	-	-

applied. That is up-time improvement 1 has to be introduced before up-time improvement 2 and processing time improvement 1 has to be introduced before processing time improvement 2, in each operation. The collection of relevant improvement proposals can be seen in Table 16.3.

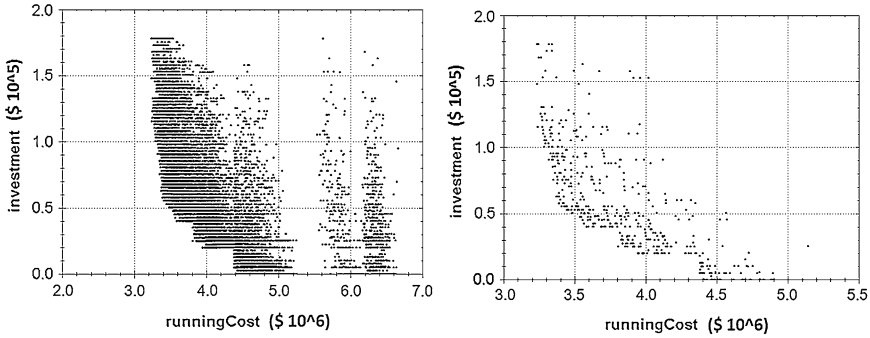
### 16.7.5 Optimisation

The cost variables and the running cost function were included in the virtual model of the conceptual production line. The annual running cost function was in this case used with the initial cost,  $C_i$ , and the delta throughput to cost,  $\Delta Ct$ . Investment alternatives combined with lean buffer configuration are the inputs to control during optimisation. All buffer capacities were subject for optimisation between 1 and 40 entries in steps of 1, except for three buffers where one is a conveyor with a size constraint limiting the maximum number of entries to 5 and the other two having a constraint caused by a minimum number of one pallet containing 24 items.

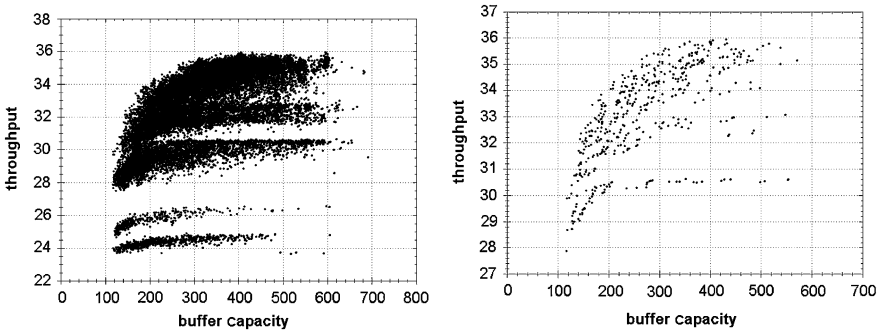
Objectives for optimisation were:

$$\min(Cr), \min(I), \min(Bc) \tag{16.21}$$

The NSGA-II algorithm [5] for multi-objective optimisation was used. The optimisation was run with 20,000 simulation iterations, each based on 5 replications with 8 days as simulation horizon, including 1 day for warm-up.



**Fig. 16.6** Investment versus running cost, complete data set to the left and only non-dominated solutions to the right



**Fig. 16.7** Throughput versus buffer capacity, complete data set to the left and only non-dominated solutions to the right

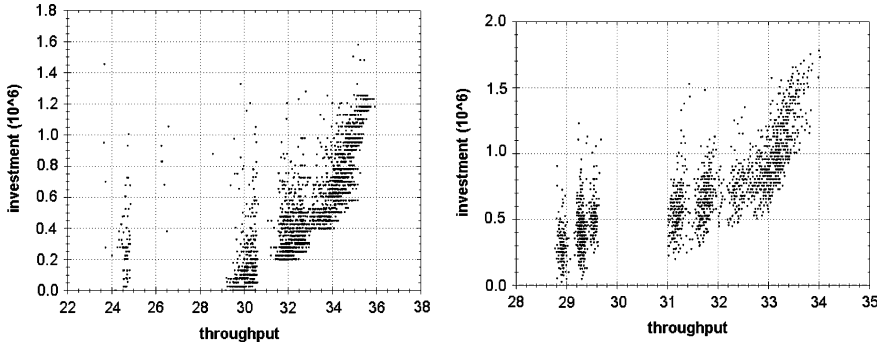
### 16.7.6 Optimisation Results

The result of the optimisation can be plotted by forming Pareto fronts considering the conflicting objectives, and the most interesting objectives are investment against running cost as shown in Fig. 16.6.

Since there are three objectives in the optimisation, the two-dimensional plot of non-dominated solutions does not reflect the typical two-objective Pareto front appearance.

An interesting conclusion is that the cost performance could be improved from M\$4.9–M\$4.4 by re-configuring the buffer capacity only. That is  $(4.9 - 4.4) / 4.9 = 10\%$  improvement. However, this requires the buffer capacity combinations to be realistic for implementation. By plotting the throughput against the total buffer capacity in the line, as shown in Fig. 16.7, some more interesting properties can be discovered.

The maximum throughput with the suggested changes in the line is approximately 36 pieces per hour and requires total buffer capacity at 400 entries. The



**Fig. 16.8** Left: experiment run with buffer capacity optimisation; right: experiment run without buffer capacity optimisation

total buffer capacity of the initial line was only 238 entries, limiting the maximum throughput to somewhat over 34 pieces per hour. By running an optimisation with the same improvement and investment parameters without buffer capacity optimisation this becomes even clearer as shown in Fig. 16.8.

The best solutions for throughput are just exceeding 34 pieces per hour without buffer optimisation and the best results with buffer optimisation is very close to 36 pieces per hour. This indicates that the initial line buffer configuration is a capacity constraint also negatively affecting the running cost when considering the throughput to cost model. Another conclusion is that the throughput objective is not possible to reach, given the proposed equipment improvements, without re-configuring buffer capacity.

### 16.7.7 Post-optimality Analysis

In order to find a relevant trade-off between investment and running cost the data must be analysed with these two objectives in mind. Finding the right solution within 20,000 data records can be difficult. To reduce the effort, some essential objective attributes have to be focused. After a discussion with the production engineers and the line supervisors, it was agreed that the throughput must be at least 34.5 in order to completely remove the need of production on additional time.

The first analysis, sorting the data for throughput over 34.5 and minimum investment shows that it would be theoretically possible to reach the throughput objective with an investment of \$50,000. When further looking into this solution, it contains major changes in buffer capacity, not feasible to implement in the short run requiring over 400 buffer entries in the line. It was agreed to add a constraint limiting the maximum buffer capacity to 300 and sort the data accordingly. After this 171 records of the initial 20,000 fulfilled the requirements. The minimum investment among these solutions was \$65,000, requiring 19 buffers to be increased.

**Table 16.4** Selected line configuration

Investments								
<i>Iu</i> Op1E	<i>Ip</i> Op1G	<i>Ip</i> Op1 J	<i>Ip</i> Op1 N	<i>Ip</i> Op1O	<i>Iu</i> Op1G	<i>Ip</i> Op1H	<i>Iu</i> Op1 N	<i>Ip</i> Op1E
2	1	1	1	1	3	1	1	2
Parameter settings								
96%	138,3 s	130 s	80 s	85 s	96%	115 s	95%	133 s
\$7 942	\$20 000	\$20 000	\$20 000	\$20 000	\$32 500	\$2 500	\$5 000	\$20 000
Sum								\$147943

**Table 16.5** Selected line configuration simulation results

Variable	Mean	Std. Dev.
Parts produced	5,831.20	22.64
Throughput	34.71	0.13
Lead time (complete system cycle time)	17,018.44	292.76
WIP	164.78	3.52
S1 (Parts produced)	3,578.20	26.09
S1 (Throughput)	21.30	0.16
S1 (Lead time)	18,520.25	695.74
S2 (Parts produced)	2,253.00	12.37
S2 (Throughput)	13.41	0.07
S2 (Lead time)	14,636.73	384.83

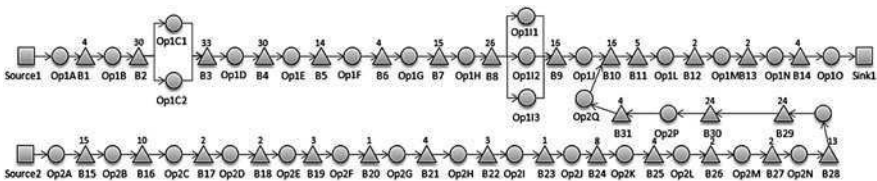
After a discussion with the engineering team, these buffer changes were also considered not to be feasible to implement in a short term. It was agreed to search for a solution with as few buffer increases as possible. After looking into the best solutions considering the objectives a new solution was configured manually based on the optimisation results, including 11 buffer increases and a throughput of 34.7 pieces per hour with a standard deviation of 0.13, verified by simulation in the same model used for the optimisation, see Tables 16.4 and 16.5.

The new buffer configuration derived from one of the optimised solutions was achieved by setting the capacity for a number of buffers to the original values. The selection criteria were to leave buffers with capacity larger than the optimised levels unchanged and at the same time leave the capacity of buffers with small optimised increases at original values, as shown in Table 16.6. Despite increasing the buffer capacity, the selected solution actually performs better than the initial line configuration in terms of WIP with an average of 165 pieces compared to 188 pieces. The investment for this solution is \$150,000 and the annual running cost would be reduced by approximately \$1.4 M.

After taking into consideration that the shift from corresponding to the actual reduced time is valid for throughput figures between 26.5 and 34.0, the actual annual saving was recalculated to \$1.29 M by using the throughput to cost function. The delta annual cost and the delta cost due to changed cost per produced unit were set to zero in this case.

**Table 16.6** Original, optimised and finally configured buffers

Buffer configuration									
B1	B2	B3	B4	B5	B6	B7	B8	B9	
4	30	33	4	5	4	11	26	10	Original
3	6	15	30	14	5	16	10	16	Optimised
4	30	33	30	14	4	16	26	16	Configured
B10	B11	B12	B13	B14	B15	B16	B17	B18	
7	5	2	2	4	4	5	2	2	Original
7	1	2	16	2	15	10	2	2	Optimised
7	5	2	16	4	15	10	2	2	Configured
B19	B20	B21	B22	B23	B24	B25	B26	B27	
3	1	1	1	1	4	4	2	2	Original
4	2	4	3	2	8	1	1	1	Optimised
3	1	4	3	1	8	4	2	2	Configured
B28	B29	B30	B31	Throughput mean			Total sum		
5	24	24	4	28,66			236		Original
13	26	25	3	35,20			265		Optimised
13	24	24	4	34,71			329		Configured



**Fig. 16.9** Resulting line with re-configured buffer capacity. Allocated buffer capacity is shown in figures above the triangular buffer symbols

$$\Delta C = \Delta C_t + \Delta C_a + \Delta C_u = \Delta C_t + 0 + 0 = \Delta C_t \tag{16.22}$$

$$\Delta C_t = Ch \times V_p(1/T - 1/T_i) \text{ valid for } a_i < T < b_i, 26.5 < T < 34.0 \tag{16.23}$$

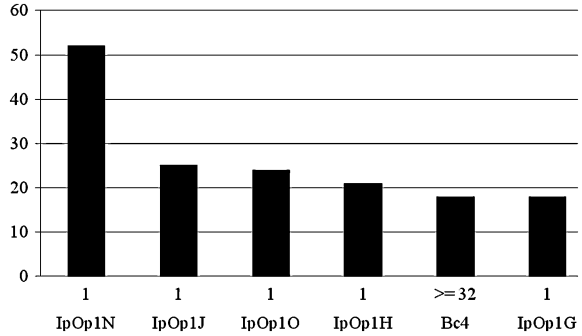
$$\Delta C_t = 1450 \times 164000(1/34.0 - 1/28.7) = M\$ - 1.29 \tag{16.24}$$

$$C_r = C_i + \Delta C = 4.90 + (-1.29) = M\$3.61 \tag{16.25}$$

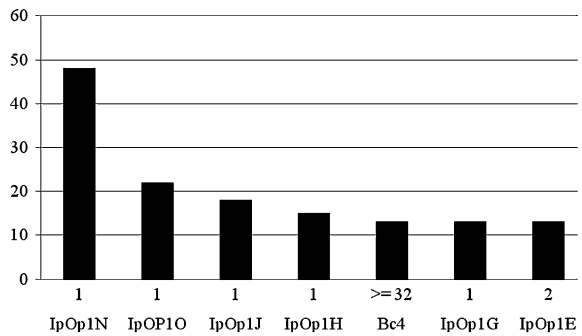
The team on the line added 0.5 pieces per hour as a safety precaution when the objective 34.5 was set. The throughput performance improvement is  $34.7/28.7 = 20.9\%$ , the annual cost performance improvement is  $1.29/4.9 = 26.3\%$  and the cost saving year one including investment is  $(1.29 - 0.150)/4.90 = 23.3\%$ . The reason for not selecting a solution with lower investment in this case is the extremely good business case revealed by the optimisation. The opportunity to choose a solution with fewer buffer changes than one of the solutions on the actual Pareto front was convenient for the engineering team carrying out the changes. The resulting line configuration is shown in Fig. 16.9.



**Fig. 16.10** Factors favouring high throughput



**Fig. 16.11** Factors favouring low running cost



### 16.7.8 Knowledge Extraction Through Data Mining

With the purpose of further investigating the possibilities of extracting knowledge to be used as an enhancement for decision-making, data mining was applied on the data from the optimisation. Microsoft SQL Server 2005 Data Mining Add-ins for Microsoft Office 2007 was used to perform data analysis searching for factors favouring high throughput and low running cost. Part of the report from the data mining is shown in Figs. 16.10 and 16.11.

The importance of implementing the improvements IpOp1N, IpOp1J and IpOp1O can clearly be seen as well as the significance of a capacity in buffer number 4 being equal to or larger than 32. Overall, the selected solution reflects the characteristics found in the data mining, indicating that the technique can be useful for providing decision support.

## 16.8 Conclusions

The potential of applying SMO, taking into account financial objectives, like investment and running cost, for decision-making support in designing/re-configuring production systems, has been explored in this chapter. Evaluating several

combined minor improvements with the help of multi-objective optimisation has opened the opportunity to identify a set of solutions revealing great financial improvement, which cannot be sought by applying any current industrial procedures. In the production line studied in this chapter the throughput can be improved by 20.9% and the cost performance on annual basis can be improved by 26.3% by applying SMO including financial objectives. The important results from the SMO study are briefly summarised below:

One finding by the SMO is that a 10% annual running cost reduction would be achievable by re-configuring the buffer capacities only.

Given the proposed equipment improvements, the target capacity is not possible to reach without re-configuring buffer capacity. In other words, investment for improvements and buffer capacities cannot be optimised separately but need to be considered simultaneously.

By utilising the knowledge created by SMO a feasible solution with a very limited need for investments could be selected in order to improve the throughput by 20.9% and the annual running cost performance by 26.3%. That is, SMO including investment and running cost objectives has proven to be a very promising concept for production system improvement and development.

WIP could be reduced despite increased buffer capacity, considering a combination of improvements.

Data mining seems to be a useful tool for finding key influencing factors from SMO as a support for creating knowledge to be applied within production system design.

Including cost parameters and cost objectives in SMO enhances the capability of the method as a decision-support instrument within the industry.

In summary, this case study has adequately proven that such a financial-based SMO method can be very valuable for practical industrial applications.

## References

1. Fu, M. C., Andradóttir, S., Carson, J. S., Glover, F., Harell, C. R., Ho, Y.-C., Kelly, J. P., & Robinson, S. M. (2000). Integrating optimisation and simulation: research and practice. In *Proceedings of the 2000 Winter Simulation Conference* (pp. 610–616). December 9–12, IEEE, Arlington, VA.
2. Law A. M., & McComas, M. G. (2002). Simulation based optimisation. *Proceedings of the 2002 Winter Simulation Conference* (pp. 41–44). December 8–11, 2002. WSC2002, San Diego, CA.
3. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (3rd. ed.). Wiltshire, UK: Wiley.
4. Ng, A., Urenda, M., Svensson J., Skoogh, A., & Johansson, B. (2007). FACTS analyser: An innovative tool for factory conceptual design using simulation. In *Proceedings of Swedish Production Symposium*, August 28–30, 2007, Gothenburg.
5. Pehrsson, L. (2009). *Simulation-based optimisation in relation to current production system data and evaluation models*, Master Thesis in Manufacturing Engineering, University of Skövde, School of Technology and Society, Skövde.

6. Standridge, C. R., & Marvel, J. H. (2006). Why Lean needs simulation. *Proceedings of the 2006 Winter Simulation Conference* (pp. 1907–1913). December 3–6, 2006. WSC 2006, Monterey, CA.
7. Deb, K., & Srinivasan, A. (2006). Innovization: Innovating design principles through optimization. *Proceedings of the Genetic and evolutionary Computation Conference (GECCO-2006)*(pp. 1629–1636). The Association of Computing Machinery (ACM), New York.
8. Ng, A., Deb, K., & Dudas, C. (2009). Simulation-based Innovization for production systems and improvement: An industrial case study. *Proceedings of the International 3'rd Swedish Production Symposium (SPS'09)*, December 2–3, 2009. Göteborg, Sweden .
9. Professional Accountants in Business Committee (2008). *International good practice guidance, project appraisal using discounted cash flow*. Professional Accountants in Business Committee, International Federation of Accountants, 545 Fifth Avenue, 14th Floor, New York, NY 10017, USA.
10. Professional Accountants in Business Committee (2009). *International good practice guidance, evaluating and improving costing in organisations*. Professional Accountants in Business Committee, International Federation of Accountants, 545 Fifth Avenue, 14th Floor, New York, NY 10017, USA.
11. Von Beck, U., & Nowak, J.W. (2000). The merger of discrete event simulation with activity-based costing for cost estimation in manufacturing environments. *Proceedings of the 2000 Winter Simulation Conference*, December 10–13, 2000. WSC 2000, Wyndham Palace Resort & Spa, Orlando, FL, USA, ACM.
12. Brown Ethan, J., & Sturrock, D. (2009). Identifying cost reduction and performance improvement opportunities through simulation. *Proceedings of the 2009 Winter Simulation Conference*, WSC 2009, December 13–16, 2009. Austin, TX.
13. Yamashina, H., & Kubo, T. (2002). Manufacturing cost deployment. *International Journal of Production Research*, 40(16), 4077–4091.
14. Jönsson, M., Andersson, C., & Ståhl, J-E. (2007). *A general economic model for manufacturing cost simulation*. Presented at the 41st CIRP Conference on Manufacturing Systems, Tokyo 2008.
15. Kaplan, R. S., & Andersson, S. R. (2007). *Time-driven activity-based costing – A simpler and more powerful path to higher profits*, Harvard Business School Publishing Corporation, ISBN-13:978-1-4221-0171-1.
16. CMA Canada (1999). *Strategic management series, strategic cost management, measuring the cost of capacity*. The Society of Management Accountants of Canada, Mississauga Executive Centre, One Robert Speck Parkway, Suite 1400, Mississauga, ON Canada L4Z 3M3.
17. Nord, C., & Pettersson, B. (1997). *Total Productive Maintenance med Erfarenhet från Volvo*, IVF Industriforskning och utveckling AB, 1997.
18. Roser, C., Nakano, M., & Tanaka, M. (2002). Shifting bottleneck detection. *Proceedings of the 2002 Winter Simulation Conference* (pp. 1079–1086). San Diego, CA.

# Chapter 17

## Supply Chain Design Using Simulation-Based NSGA-II Approach

Lyes Benyoucef and Xiaolan Xie

**Abstract** This chapter addresses the design of supply chain networks including both network configuration and related operational decisions such as order splitting, transportation allocation and inventory control. The goal is to achieve the best compromise between cost and customer service level. An optimisation methodology that combines a multi-objective genetic algorithm (MOGA) and simulation is proposed to optimise not only the structure of the network but also its operation strategies and related control parameters. A flexible simulation framework is developed to enable the automatic simulation of the supply chain network with all possible configurations and all possible control strategies. To illustrate its effectiveness, the proposed methodology is applied to two real-life case studies from automotive industry and textile industries.

### 17.1 Introduction

#### 17.1.1 Context

The global economy and the recent developments in information and communication technologies (ICT) have significantly modified the business organisation of enterprises and the way they do business. New forms of organisations such as

---

L. Benyoucef (✉)  
INRIA, COSTEAM Project, ISGMP Bat. A, Ile Du Saulcy, 57000 Metz, France  
e-mail: lyes.benyoucef@inria.fr

X. Xie  
ENSM.SE, 158 Cours Fauriel, 42023 Saint-Etienne Cedex 2, France  
e-mail: xie@emse.fr

extended enterprises and networked enterprises (also called supply chain networks) appear and they are quickly adopted by most leading enterprises. It is well known that “competition in the future will not be between individual organisations but between competing supply chains” [1]. Thus, business opportunities are captured by groups of enterprises in the same network. The main reason for this change is the global competition that forces enterprises to focus on their core competences (i.e., to do what you do the best and let others do the rest). According to a visionary report of Manufacturing Challenges 2020 conducted in the USA [2], this trend will continue and one of the six grand challenges of this report is the ability to *reconfigure networked enterprises rapidly in response to changing needs and opportunities*. Although the resulting supply chain networks are more competitive, the tasks for planning, managing and optimising are much more difficult and complex.

A supply chain is a network of facilities, such as suppliers, plants, distributors, warehouses, retailers which performs a set of operations including procurement of components and raw materials, assembling of products, storage and handling of semi-finished and finished products, transportation and delivery of products. Supply Chain Management (SCM) has become recognised as a critical aspect in today’s fiercely competitive business environment. In 2002, American companies spent \$910 billion, or about 8.7% of the United States gross domestic product (GDP), on business logistics systems, which contained the warehousing costs, transportation costs, shipper related costs and logistic administration costs. In Singapore, the transport and communication industry sector contributed about 10.8% of the GDP in year 2003. Considering the importance and the influence of SCM, manufacturers and retailers have paid great efforts to handle the flow of products efficiently and coordinate the management of supply chain smoothly.

While alliance-like enterprise networks with the underlying supply network represent tremendous business opportunities, they also make the involved enterprises face greater uncertainties and risks. Firstly, networks or supply chains have to be modified or dissolved once the business opportunities evolve or disappear. Secondly, changes or major perturbations at one enterprise may propagate through the whole network to other enterprises and hence influence their performance. The evolution from single enterprise with a high vertical range of manufacture towards enterprise networks offers new business opportunities especially for small and medium enterprises that are usually more flexible than larger companies. However, in order to be successful, performance and expected benefits have to be carefully evaluated and balanced in order to become a partner of the right supply chain network for the right task.

### ***17.1.2 Motivations***

States-of-the-art on supply chain modelling and optimisation approaches are presented in [3] and [4]. Schmidt and Wilhelm [3] pointed out that interactions of decisions at different levels (strategic, tactical and operational)

should be considered. Goetschalckx et al. [3] presented an overview of the application of mathematical programming models in the strategic design and improvement of global logistics systems/supply chain network. They summarised the international characteristics of published strategic global logistics models. *Among the international characteristics of these global logistics systems, stochastic features make up an important set.*

Meixell and Gargeya [5] reviewed the model-based literature for the global supply chain design problem by using dimensions related to ongoing and emerging issues in supply chain globalisation. Overall, they realised that although the research community has tackled some of the most difficult global supply chain issues, only few models among them comprehensively address outsourcing, integration and strategic alignments in global supply chain design. They concluded that *global supply chain models need to address the composite supply chain design problem by extending models to include both internal manufacturing and external supplier locations, global supply chain models need broader emphasis on multiple production and distribution tiers in the supply chain, the performance measures used in global supply chain models need to be broadened in definition to address alternative objectives, and more industry settings need to be investigated in the context of global supply design.*

Klose and Drexel [6] presented an extensive state of the art dedicated to facility location models for distribution system design. Model formulations and solution approaches vary widely in terms of fundamental assumptions, mathematical formulation, computational complexity and performance. They focused in particular on continuous location models, network location models, mixed-integer programming models and applications. Revell and Eiselt [7] also surveyed a number of the important decisions problems in facility location. They stated that *the field is very active with many interesting problems still being investigated, both from a problem statement/formulation and algorithmic point of view. Although the field is active from a research perspective, when it comes to applications, there appears to be a significant deficit, at least as compared to other, similar, fields.*

After a comprehensive literature review and interviews of industries, we have identified three keys facts that should be taken into account when designing a supply chain network.

- Supply chain network is a dynamic, stochastic and complex system. The performance of any particular facility in the system depends to a large extent on the behaviour of other facilities. For improving the overall performance of a network, it is necessary to view the system as a whole. Strategic decisions, such as facility location, should be optimised simultaneously with other decisions at the tactical/operational level related to production planning, transportation, inventory control, etc.
- Existing literature on modelling supply chains is very rich, which includes techniques like Petri nets [8], fuzzy logic [9], multi-agent systems [10], mathematical programming [11], etc. Nevertheless, an overwhelming majority of the literature formulates production–distribution network design problem, as

a particular case of supply chain design, in terms of mixed-integer programming models. Due to the complexity and tractability, supply chain uncertainties and dynamics, such as demand fluctuation, production uncertainty and transportation instability, are either absent or over-simplified in most of these models. In particular, these methods fall short when qualitative optimisation variables are involved, such as the selection of inventory control and production policies.

- The total cost of all supply chain activities is often used as the unique optimisation criterion. While in current competitive environment, the consideration of only one objective is not sufficient to derive good decisions. Customer satisfaction related issues should be taken into account at the stage of supply chain design. True multi-objective optimisation is necessary to avoid the inappropriate transformation of customer service level to some costs.

### ***17.1.3 Contributions***

In this chapter, we present a new hybrid approach to support decision makers for the assessment, design and improvement of such supply chain networks. The approach consists of an optimiser and a simulator. The optimiser, based on an NSGA-II algorithm, is used to find best-compromised solutions with respect to various criteria, such as cost and customer service level. Candidate solutions suggested by the optimiser are evaluated through simulation, which enables realistic evaluation taking into account uncertainties and dynamics along the whole supply chain. The simulation model builder is developed to facilitate automatic model creation, which is a challenging issue as decision variables describe key aspects of supply chain network structure and its operating rules. To validate the proposed approach, two case studies, proposed by partners from automotive and textile industries, are presented and the computational results analysed.

Our approach differs significantly from existing simulation-based optimisation approaches for supply chain optimisation problem which to the best of our knowledge, not only focus on the optimisation of quantitative control parameters or system parameters such as buffer capacities but also for fixed supply chain structure and given control strategies. Our approach optimises at the same time the structure of the network, the set of control strategies and the quantitative parameters of the control strategies. This would not be possible without the properly defined generic modelling and simulation framework that allows the evaluation of a supply chain network for all possible configurations and control strategies.

In the remainder part of the chapter, [Sect. 17.2](#) sets the problem under consideration. [Section 17.3](#) gives a brief literature review of existing models and methods for deterministic and stochastic production–distribution network design problem, and supplier selection problem. [Section 17.4](#) discusses the architecture of

the proposed simulation-based NSGA-II and describes in detail the developed simulation framework and different operational rules. Section 17.5 presents the first case study from automotive industry. Section 17.6 presents the second case study from textile industry. The two sections demonstrate how the case studies are handled by applying the proposed simulation-based optimisation approach. Section 17.7 concludes the chapter with some perspectives.

## 17.2 Problem Setting

### 17.2.1 Supply Chain Network Design

Given a general supply chain network with all existing and potential facilities comprising four stages: supply stage, production stage, distribution stage and customer stage (Fig. 17.1), the network design problem consists in selecting the facilities to open/operate in order to form a network with minimal overall costs and highest customer service levels.

Due to the complexity and dynamic nature of the problem under consideration, we present the network design problem in a descriptive manner, i.e., principal characteristics of facilities, operations and processes are described to clearly set the problem.

The supply stage contains  $P$  potential suppliers. All suppliers provide the same type of products, but at different prices, duties, supply lead times, etc. Price and duties are financial attributes that determine the purchasing cost. Supply lead time is defined as the time span from order reception by the supplier to the moment when products are ready for transportation. There are several links available for a supplier to ship products to the plants. In case of multiple links, transportation allocation rules are used for link selection and volume allocation among selected links. Transportation lead time could be constant or random.

The production stage is composed of  $K$  plants that produce different final products. Each plant has a limited production capacity. We assume that raw materials and components are always available. Whenever a production order is assigned to a plant corresponding products will be available after a period of production lead-time. The production lead-time could be constant or random. The number of products that can be dispatched into a plant is limited by the plant production capacity. Each plant has limited finished goods inventory (FGI). There is no transportation link between plants. All final products are delivered to customers via distribution centres.

The customer stage contains all the customers who are served by different distribution centres (DCs). Different customers generate independent random demands for multiple types of products. For each customer and each type of product, the demand quantity and frequency could be constant or random. No transportation link exists between customers. In this study, products demanded by



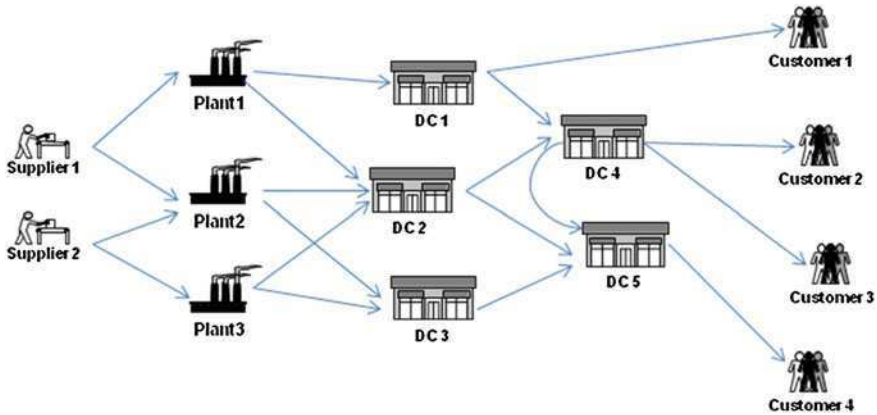


Fig. 17.1 Multi-echelon supply chain network model

customers are managed in two manners, either make-to-stock (MTS) or make-to-order (MTO), according to their characteristics. Demands for MTS products are served by distribution centres, while demands for MTO products are forwarded to plants directly.

The intermediate stage is the distribution network, comprising a number of DCs with links between DCs. The MTS inventory held in a DC is replenished, according to some given rules such as  $(R, Q)$  or  $(s, S)$ , from plants or other upstream DCs. In case of stock-out at a DC, the MTS demands assigned to the DC are backlogged for future fulfilment.

All facilities in the network are connected by transportation links. Each link could be a combination of various transportation modes such as railway, road and sea. There could be multiple links available between two facilities. Each transportation link has its carrier departure rule which determines the carrier departure condition and frequency. Each carrier has a limited transportation capacity. The products sent out from one end of a link are considered available at the other end after a period of transportation lead-time, which could be constant or random. More modelling details will be given in Sects. 17.4 and 17.5.

### 17.2.2 Decisions and Performance Indicators

Three types of decisions are to be made in the model under consideration. First, an “open or close” decision should be made for each candidate site in order to locate plants and/or DCs. Such decisions are considered strategic and have major impact on a system’s performance. Second, appropriate operational rules are to be selected to manage various operations in the network. For example, for each DC and each type of products of the DC, it is necessary to determine the corresponding

inventory control policy. Accordingly, the control parameters associated with selected operational rules represent the third type of decisions.

Given a combination of these decisions, the quality of the corresponding network configuration depends on its profitability and operational efficiency. To measure the profitability and operational efficiency of a candidate production–distribution network, it is indispensable to define relevant performance indicators. Two types of indicators are taken into consideration in this study: financial and logistics indicators. Various costs are considered as financial indicators, including investment costs, production costs, transportation costs and inventory holding costs. Logistics indicators include average demand fill-rate, average demand cycle time, probability of on-time delivery, etc.

The two types of performance indicators are often contradictory and the balance between them is critical issue that should be handled directly by the decision-maker himself. As a result, it is necessary to perform multi-objective optimisation, instead of transforming them into a single cost function.

## 17.3 Literature Review

The literature dedicated to supply chain management problems is very rich. In this section, we restrict our literature review to two classes of problems namely “production–distribution network design” and supplier selection.

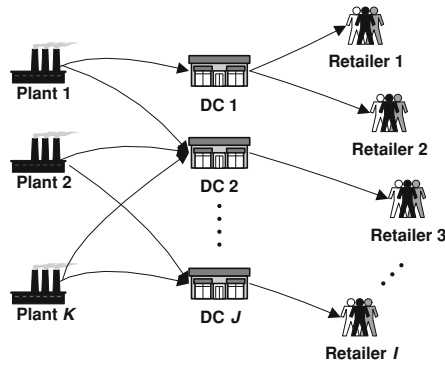
### 17.3.1 *Production–distribution Network Design*

The production–distribution network design problem has been extensively studied in the literature. We summarise in the following major literature reviews, and existing models and methods developed for production–distribution network design.

#### 17.3.1.1 **Deterministic Models**

Many dedicated models have been proposed in the past for production–distribution network design. Due to the complexity, most existing models are deterministic. In fact, uncertainties and dynamics along a production–distribution network are simply omitted or not realistically addressed.

Different deterministic facility location models are proposed in [12] which enable the determination of distribution centre locations and their service areas. The location problem is formulated using linear mixed-integer programming for minimisation of investment costs. These models are important basis for later research. Based on these models, Cohen and Lee [13] proposed a 4-stage



**Fig. 17.2** The network considered in [16]

production–distribution network model “PILOT”. PILOT is a multi-period deterministic model for minimising a nonlinear cost function. Another linear mixed integer programming model was proposed in [14] for international production–distribution network design. This deterministic model is a single period model, which consists in maximising the total after-tax profit.

The model proposed by Arntzen et al. [15] is a deterministic multi-period multi-product model initially built for DEC global logistics chain design. This model is considered as one of the most complex and complete models in the literature.

A representative model of Pirkul and Jayaraman [16] is described in details to give a better idea on existing deterministic models. The model concerns a 3-stage production–distribution network in which different products flows from plants to distribution centres and then to retailers. The plants and retailers are geographically dispersed in a region. Each retailer faces demands for a variety of products that are manufactured at the plants. The network under consideration is illustrated in Fig. 17.2.

The costs considered in this model include: (1) site-dependent distribution centre opening cost, (2) site-dependent plant opening cost, (3) unit transportation cost of product from plant to distribution centre and (4) unit transportation cost of product from distribution centre to retailer. Three types of constraints are taken into account:

- Each plant has a maximum production capacity, expressed in terms of the maximum operating time units available at the plant. Each unit of product takes some time units at each plant.
- Each distribution centre has a maximum handling capacity, expressed in terms of the maximum number of product units that the distribution centre can handle.
- Each retailer is assigned to exactly one selected distribution centre for each type of products.

The problem consists of selecting  $W$  distribution centres and  $P$  plants to serve  $n$  retailers with  $p$  product types such that the total cost is minimised and the

above-mentioned constraints are satisfied. A Lagrangian relaxation-based method is used to solve the problem.

Pirkul and Jayaraman [17] discussed an identical Lagrangian method for a revised production–distribution model named “PLANWAR”. The proposed model is slightly different in that the DC-Retailer flow is expressed in continuous variables, there is no single sourcing requirement for retailers and the requirement on the number of plants and distribution centres to be located is changed to an inequality constraint. Furthermore, one more variation is related to the capacity at plants in “PLANWAR”, which is a different resource than the storage space.

Pirkul and Jayaraman [18] presented a 4-stage capacitated multi-product supply chain system models. From the modelling perspective, the new model differs from the earlier proposed models in several ways. Suppliers with limited capacity are taken into account. The model also considers the plant production decisions, raw material requirements from the suppliers and bill-of-materials. The single sourcing condition is changed into the sole sourcing of each retailer from the same DC for all products. A Lagrangian relaxation approach was proposed.

Vila et al. [19] presented a generic methodology to design production–distribution networks of divergent process industry companies in a multinational context. The proposed methodology uses a mathematical programming model to map the industry manufacturing process onto potential production–distribution facility locations and capacity options. Each facility may use different layouts and the plant capacity is specified by selecting appropriate technological options. The objective is to maximise global after tax profit in a predetermined currency. The proposed methodology was applied to a case study from softwood lumber industry by using commercial optimisation software.

Martel [20] proposed a mathematical programming approach to design international production–distribution networks for make-to-stock products with convergent manufacturing processes. Various formulations of the elements of production–distribution network design models are discussed. Special attention is paid to modelling issues encountered in practice, which have significant impacts on the quality of the designed logistics network. The discussed issues include the choice of the objective function, definition of the planning horizon, manufacturing process and product structures, logistics network structure, demand and service requirements, facility layouts and capacity options, product flows and inventory modelling, as well as financial flows modelling. A typical mixed-integer programming model is presented and solved with commercial solvers.

### 17.3.1.2 Stochastic Models

Literature dedicated to the stochastic facility location problem also becomes rich. However, limited models have been proposed for the stochastic production–distribution network design problem.

Snyder [21] presented a comprehensive state-of-the-art review of existing stochastic models for the facility location problem. Many of these models

minimise the expected cost or maximise the expected profit of the system under consideration. Others take a probabilistic approach, e.g., maximising the probability that the solution is in some sense “good”. Part of the models are solved using algorithms designed specifically for the problem, where others are solved using more general stochastic programming techniques.

Louveaux [22], presented stochastic versions of the capacitated P-median problem (CPMP) and capacitated fixed charge location problem (CFLP) were presented, where customers’ demands, production costs, transportation costs and selling prices are random variables. The goal is to choose facility locations, determine their capacities and decide which customers to serve and from which facilities in order to maximise the expected utility of profit.

Ricciardi et al. [23] proposed a facility location model with random throughput cost considering DCs. The objective is to minimise the deterministic transportation cost (plant-to-DC and DC-to-customer) plus the expected throughput cost at the DCs. The authors first consider the network flow aspect of the problem (assuming the DC locations are given) and develop a model for the expected flows. Then they embed the expected cost model into a nonlinear integer program. For each candidate solution to the location problem, a Lagrangian problem is solved to compute the expected flows.

Stochastic versions of the joint inventory-location model are presented in [24–28]. The models make decisions on DCs location while minimising fixed investment costs, transportation costs and inventory costs at the DCs given stochastic customer demands. The demand means and variances could be stochastic, as well as costs, lead-times and other parameters. They formulated the problem as a nonlinear integer program and presented a Lagrangian relaxation approach to solve it. Several computational experiments attested the effectiveness of the proposed approach.

Erlebacher and Meller [24] formulated a highly nonlinear integer inventory-location model. The customer demands are stochastic and rectilinear distances are used to represent the distances between the locations. Each DC operates under a continuous review inventory system. The problem consists in the determination of the number of DCs and their locations, as well as the customers they serve in order to minimise the fixed operating costs at DCs, inventory holding costs and transportation costs. Since the general version of the problem is NP-hard, they developed analytical models and proposed heuristic procedures for special cases obtained under some simplified assumptions.

Shen [28] proposed a nonlinear integer-programming model for the multi-product supply chain design problem. The model determines the location of facilities and the assignment of retailers to the facilities in order to minimise a nonlinear objective function that includes the economies of scale costs at the facilities. It is the first multi-product supply chain design model that incorporates supply chain costs exhibiting economies of scale. It generalises many well-studied models. A Lagrangian relaxation solution algorithm is proposed, and compared with existing algorithms for different special cases of the proposed model.

Tanonkou et al. [29] considered a single-product distribution network design problem where facility location and supplier selection decisions are integrated. The retailers' demands are random and supply lead-times constants. A nonlinear integer-programming model is proposed, for which the determination of exact solutions is a NP-hard problem. A Lagrangian relaxation solution algorithm is developed allowing the determination of distribution centres locations, assignment of distribution centres to suppliers, and assignment of retailers to the distribution centres to minimise the total fixed distribution centres location costs, running inventory and safety stock costs at the distribution centres and transportation costs.

Tanonkou et al. [30] presented a Lagrangian relaxation-based approach to solve a single-product distribution network design problem with random demands and random supply lead-times. A nonlinear integer-programming model is proposed. The model determines the location of distribution centres and the allocation of retailers to the distribution centres. The goal is to minimise the total fixed distribution centres location costs, running inventory and safety stock costs at the distribution centres and transportation costs through the network, while ensuring a given retailer service level. The resulting problem is difficult since it incorporates nonlinear working-inventory costs and nonlinear safety stock inventory costs. Computational results are presented and analysed by validating the effectiveness of the proposed approach. The multi-product version of the problem is considered in [31], where a similar approach is used.

For joint transportation-location problem, França and Luna [32] used Benders decomposition to solve a problem, which combines the CFLP and the stochastic transportation problem with random customer demands.

### ***17.3.2 Supplier Selection***

Motivated by uncertainty reduction and customer service improvement, more and more companies are paying attention to multiple sourcing, i.e., to engage with more than one supplier at the same time. The studies by Moinzadeh and Nahmias [33], Sculli and Shum [34], Ramasesh et al. [35], Lau and Zhao [36] and Ganeshan et al. [37] demonstrate the interest of companies to adopt the multiple sourcing strategies when managing their inventories. Advantages of using multiple sourcing (i.e., multiple suppliers) to replenish one inventory item include stocking efficiency, supplier reliability, pricing and quality competitiveness, etc. However, most of the studies assume that items/products of different suppliers are identical, namely that suppliers provide items/products at the same price and quality [37]. The resulting problem is turned to an “inventory-only” problem. Lau and Zhao [36] developed computational methods to compute the optimal order-splitting ratio, order quantity and reorder point with only two suppliers.

Sedarage et al. [38] developed an optimisation model to determine both the reorder level and the order split quantities simultaneously for general  $n$ -supplier

systems in which the unit time demand and supplier lead times are random variables. The model minimises the expected total cost, including the ordering cost, procurement cost, inventory holding cost and shortage cost. Extensive numerical experiments were performed to analyse the advantages and distinct characteristics of multiple-supplier systems versus single and dual sourcing systems.

Ghodsypour and O'Brien [39] presented a nonlinear integer programming model to solve the multiple sourcing problems, which takes into account the total cost of logistics, including net price, storage, transportation and ordering costs. An algorithm is proposed to solve the model and numerical experiments are presented to illustrate its efficiency.

Qi [40] studied an integrated decision-making model for a supply chain system where a single manufacture faces pricing, production and procurement constraints. The market demand is price-sensitive, and the manufacture supply capacity has to be acquired from a set of capacitated suppliers. The problem consists in simultaneously determining the selling price and the production quantity, as well as the supplying capacity from the suppliers in order to maximise the total profit. The problem is proved to be NP-hard in the ordinary sense and a heuristic algorithm and an optimal dynamic programming algorithm were developed. To demonstrate the efficiency and effectiveness of the algorithms, some experimental studies are presented.

Wang et al. [41] considered a  $n$ -capacity supplier, single item inventory system, where the suppliers have different lead times and purchase prices. An integer linear programming model is proposed to help managers select the optimal suppliers and determine both the reorder level and split suborders of each selected supplier for a given order quantity so that the total average inventory cost is minimum and constraints of supplier ability, quality and demand are considered. An approach combining the branch-and-bound algorithm and the enumeration algorithm is developed to solve the problem.

Ding et al. [42] presented a simulation-based evolutionary multi-objective optimisation approach for integrated decision-making including supplier selection, order splitting, transportation allocation and inventory control. The approach developed by them includes an optimiser and a simulator. The optimiser, based on a multi-objective genetic algorithm, is used to find best-compromised solutions with respect to various criteria, such as the total cost and customer service level. Candidate solutions suggested by the optimiser are evaluated through simulation, which enables realistic evaluation taking into account uncertainties and dynamics along the whole supply chain. The simulation model builder is developed to facilitate automatic model creation, which is a challenging issue as decision variables describe key aspects of supply chain network structure and its operating rules.

For more state-of-the-art analysis including deterministic and stochastic models of supply chain design, the reader may refer to Slats et al. [43], Beamon [44], Sarmiento and Nagi [45], and Snyder [21, 27].

## 17.4 Simulation-Based NSGA-II Approach

### 17.4.1 Approach Overview

For supply chain optimisation practitioners, one major obstacle is uncertainty, which represents the supply chain dynamics. Its stochastic nature makes most analytical models either over simplistic or computationally intractable. Computer simulation, with a strong capability in handling supply chain dynamics, is regarded as the most popular analysis tool for these systems. In particular, discrete-event simulation is often used to facilitate “what-if” analysis. Simulation-based optimisation is thus regarded as an effective method that adapts simulation to applications requiring optimisation.

A general simulation-based optimisation method consists of two essential components: an optimisation module that guides the search direction and a simulation module used to evaluate performances of candidate solutions (*network configuration + operational rules and parameters*). Compared with mathematical programming techniques, simulation-based optimisation methods replace the analytical objective function and constraints by one or more simulation models. The decision variables are the conditions under which the simulation is run. Iteratively the output of the simulation is used by the optimisation module to provide feedback on progress of the search for the optimal solution.

Existing literatures related to simulation-based optimisation methods can be arranged under four major categories: gradient-based search methods, stochastic optimisation, response surface methodology and heuristic methods [46]. For industrial applications, several search algorithms have been linked with simulation, including pattern search, simplex, simulated annealing and genetic algorithm. These search algorithms intelligently guide the simulation model to near-optimal solutions. According to an empirical comparison of these four algorithms [47], genetic algorithm showed the capability to robustly solve large problems and problems with non-numeric variables. It performed well over the others in solving a wide variety of simulation problems.

In this study, a simulation-based multi-objective optimisation method has been developed and integrated for joint optimisation of supply chain network structure and operational parameters (inventory control parameters, transportation allocation, etc.). More specifically, a non-dominated sort genetic algorithm-II (NSGA-II) is adapted to perform stochastic search for solutions (network structure and/or operational rules parameters), which achieves a trade-off regarding conflicting criteria, e.g., costs and customer service level. Decisions are incorporated into discrete-event simulation models for the evaluation of KPIs. The structure of the proposed simulation-based optimisation approach is shown in Fig. 17.3.

The uniqueness of the proposed approach is that it not only makes decision at the strategic level, but more importantly, it addresses the operational aspects of each solution through simulation. In the following sub-sections, we present in more detail the modelling and simulation framework and the adapted NSGA-II algorithm.



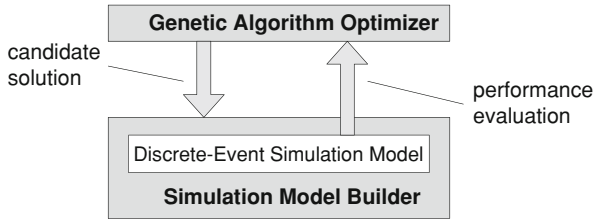


Fig. 17.3 The simulation-based optimisation framework

### 17.4.2 A Supply Chain Modelling and Simulation Framework

The use of simulation instead of analytical evaluation of a supply chain network is motivated by our objective to make network design decisions based on true operational performances of the related supply chain. *This requires realistic modelling of uncertainties and supply chain dynamics that cannot be fulfilled with analytical performance evaluation methods.*

For most simulation evaluation approaches, supply chain processes are modelled to perform “*what-if*” analysis. Simulation plays a different role in our approach. In this approach, discrete-event simulation is used to estimate the operational performance of all solutions suggested by the optimiser. Moreover, unlike most simulation-based optimisation methods, in which decision variables are only quantitative parameters for system control, our optimisation targets incorporate *structural, qualitative and quantitative variables*. Different combinations of facilities and transportation links result in different network structures. Correspondingly, information flows and material flows in the simulation model are different from one to another. The simulation model has to be regenerated each time according to the selected network structure, and operational rules should be adapted according to both the network structure and control parameters.

Due to the numerous combinations of decision variables, *a flexible simulation-modelling framework is indispensable to facilitate the automatic creation of simulation models*. This difficulty is addressed in our approach by two means. We first develop an object-oriented modelling framework dedicated to production–distribution network simulation. Principal facilities, such as plant and distribution centre (DC), are modelled as C++ classes. Further, we have defined and implemented main operation rules for decision making during simulation to ensure the connectivity of information and material flow when simulating a network with varying structures.

#### 17.4.2.1 Facility Modelling

One advantage of using simulation for performance evaluation is that we could model complex network operations with respect to their dynamic nature. Four

supply chain facilities respectively customer, distribution centre, plant and supplier and transportation link are abstracted and each is implemented as a C++ class. We focus in this part on the presentation of the dynamic characteristics of each facility and its associated operations considered in this framework.

- *Customer*: Customers generate demands for final products. Each customer could generate demands for multiple types of products, with random or constant quantity and frequency. The customer object contains information on expected lead-time, used to measure the on-time delivery. All customer demands are collected and assigned respectively to DCs or plants according to its corresponding production strategy. During the discrete-event simulation, this object interacts with the whole network through two events respectively for demand generation and products receiving. Customer demands for products are served by *distribution centres* (DC) directly. The inventory held in a DC is replenished according to some given inventory control rules. It places replenishment orders to its upstream DCs or directly to plants. Storage capacity is introduced in the sense that over-capacity products can still be received and stored in the DC, while an extra over-capacity penalty cost is charged for the excessive volume. A DC is characterised by four events, (i) inventory check, (ii) ordering, (iii) goods receiving and (iv) goods dispatching.
- *Distribution centre*: The inventory held in the DC is replenished according to some given inventory control rules. It places purchasing orders to plants and receives deliveries from plants through transportation links. Storage capacity is introduced in the sense that over-capacity products can be still received and stored in the DC, while an extra over-capacity cost is charged for the excessive volume. Relevant costs are inventory holding cost, ordering cost and over-capacity cost.
- *Plant*: Plants plays an important role. The maximum production capacity  $Q_{\max}$  of a plant is constant or variable following a given distribution. All types of products share the capacity. Production lead-time is the period from the moment when production is started for a certain product until the moment when the corresponding final product is available. A minimum production quantity  $Q_{\min}$  is introduced for economics of scale, i.e., production will not be started if the total waiting order quantity is not enough.

For a better understanding of the production process modelled in this framework, Fig. 17.4 shows different events observed during the production process.

More specifically, as soon as a production order is assigned to and then received by a plant at time  $t_0$ , the order is put in the waiting order queue. According to the bill-of-material, the order will be confirmed and ready for production at time  $t_1$  if all components are available. In this study, time  $t_0$  is equal to  $t_1$  since we assume that components and raw materials are always available. The total waiting order quantity  $Q_{\text{total}}$  is checked at a given frequency. We note the interval between two consecutive moments by  $T_{\text{intvl}}$ . Then we have  $Q_{\text{total}} = N \times Q_{\max} + Q_{\text{res}}$ , where integer  $N \geq 0$  and  $Q_{\text{res}} < Q_{\max}$ . If the quantity  $Q_{\text{total}}$  is enough ( $Q_{\text{total}} \geq Q_{\min}$ ),

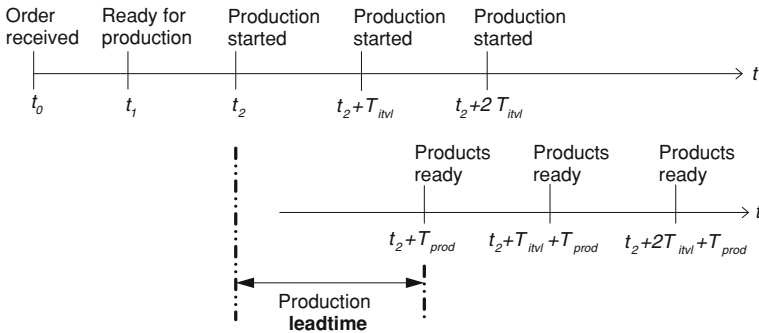


Fig. 17.4 Production process modelling in a plant

then the production is started at time  $t_2$ . Two cases are possible then, respectively  $N = 0$  and  $N > 0$ :

- (1) In case that  $N = 0$ , a volume of  $Q_{res}$  will be produced and corresponding products will be available at time  $t_2 + T_{prod}$ .
- (2) In case that  $N > 0$ ,  $N$  times production of a volume  $Q_{max}$  will be started consecutively at each possible starting moment. For the rest part  $Q_{res}$ , the production will be started if  $Q_{res} \geq Q_{min}$ . Otherwise, the rest part will keep waiting until the arrival of new production orders. For those treated orders, the final products will be available respectively at  $t_2 + T_{prod}$ ,  $t_2 + T_{intvl} + T_{prod}$ ,  $t_2 + 2T_{intvl} + T_{prod}$  and so on.

- *Supplier*: Manufacturing details are excluded in this study. Rather, important information related to the FOB (free on board) price, charged duties for export and supply lead-time are modelled as attributes. A minimum order size is introduced to represent the manufacturing economies of scale. A supplier accepts only those purchasing orders with a quantity larger than the minimum order size. More precisely, during the simulation, the orders generated by plants with a quantity less than the minimum order size are accumulated and sent out as a single order when the total quantity is sufficient. This is not explicit capacity limitation for suppliers, however we use “supply lead time” to reflect the limited supply capacity. Supply lead-time is a variable attribute regarding the purchasing quantity. Supplier engagement cost is incurred if a new supplier is included in the supplier portfolio, representing the costs for contract negotiation.
- *Transportation link*: When products are ready for transportation, they are transported from one facility to another through *transportation links*. Products assigned to each transportation link wait in a queue. Transportation lead-time and carrier capacity are modelled. An order could be divided into sub-orders and transported in several times. Two types of rules are associated with this building block to manage issues related to carrier loading and departure. Transportation cost is calculated depending on unit shipment cost

**Table 17.1** Supply chain facilities' attributes

Customer	Distribution centre	Plant	Supplier	Transportation link
Demand quantity*	Storage capacity	Maximum production capacity*	FOB price*	Transportation lead-time*
Demand interval*	Over-capacity cost	Minimum production quantity*	Duties*	Carrier capacity
Behaviour type	Holding cost	Production lead-time*	Supply lead-time*	Unit shipment cost*
Expected lead-time	Ordering cost		Minimum order size*	Batch shipment cost*
Service priority			Engagement cost	

and batch shipment cost. The unit shipment cost is applied to the shipped product quantity, while batch shipment cost occurs once for each shipment. During the simulation, a transportation link is characterised by two events: (i) batch size check, and (ii) carrier departure.

Table 17.1 lists the principal attributes of the four facilities plus the transportation link. The attributes marked “\*” could be either a constant or a random number, following some historical data or a given distribution law.

### 17.4.2.2 Operation Rules

The principal operation rules considered in this study are introduced as follows:

*Demand fulfilment rules* concern how customer demands are handled at a DC. Two non-parameterised local rules are used including first-come-first-service (FCFS) and Priority Service (PS). With FCFS, customer demands are treated sequentially according to their arrival date. With PS, the customer with the highest priority is served first. These two rules are also used for *production order scheduling*.

Three *inventory control rules* commonly used for inventory replenishment are implemented respectively, (i) base stock: if the inventory position  $I_t$  is lower than the base stock level  $B$ , the order quantity is  $(B - I_t)$ ; (ii)  $(R, Q)$ : if the inventory position  $I_t$  is lower than the reorder point  $R$ , the order quantity is  $Q$ ; (iii)  $(s, S)$ : if the inventory position  $I_t$  is lower than the reorder point  $s$ , order  $(S - I_t)$ .

*Order assignment rules* are introduced for the assignment of production/replenishment orders, generated either by customers or by DCs, to different plants and/or DCs. We defined two global rules to deal with this case. The first consists in placing the whole order to one plant, according to plants' and DCs' performance characteristics such as production capacity or length of the waiting order queue. The second choice consists of splitting each order into several sub-orders, one for each plant/DC, and then placing a sub-order at the same time to each plant/DC, according to the order assignment weight  $w_i$  for plant  $i$  in our case.

*Transportation allocation rules* are necessary since there may be multiple links between two nodes. This type of rule facilitates decision-making on which link to use and how much to ship. Transportation allocation rules used in this chapter are similar to order assignment rules. Either products are sent out using a unique link determined according to transportation cost and lead-time or products are allocated to all possible outgoing links proportional to the allocation weight associated with each link. In the case of ratio-based rule, the weights are determined by the optimiser if no appropriate parameter setting is available.

*Carrier loading rules* are used if a carrier capacity is specified for a transportation link. If the FCL (full carrier load) rule is applied, the carrier cannot depart unless the quantity of the products for transportation reaches the carrier capacity, while such a condition is omitted in the case of LCL (less than carrier load). These rules are non-parameterised local rules.

*Carrier departure rules* define the conditions to trigger the shipment process. Three non-parameterised local rules are used. Carrier departure can be scheduled upon a regular base, i.e., “periodic”, or following any “given schedule”. In the case of a “ready to go” rule, the shipment process can be triggered at any moment. These rules are generic and flexible in the sense that they guarantee all simulation models are meaningful and executable with respect to any network structures generated by the optimiser. Orders could be forwarded to the desired facility with an appropriate quantity and products are reasonably allocated and delivered through corresponding transportation links. Simulation models with various network structures could thus be created automatically without human intervention and successfully run.

The object-oriented structure of the simulation module enables a relatively easy extension for various applications. More specifically, new attributes could be added into facility building blocks and new operational rules could also be designed and added into the simulation module. The utilisation of such operation rules is further demonstrated in the next section with an application in the automotive industry.

### ***17.4.3 Multi-Objective Evolutionary Optimisation Methods***

Classical multi-objective optimisation methods, such as weighted sum and goal programming, suggest converting a multi-objective optimisation problem to a single-objective optimisation problem by emphasising one particular Pareto-optimal solution at a time. The weighted sum approach requires the appropriate weights, which are often hard to set. To obtain the Pareto-optimal front, such methods should be applied a large number of times with different weights.

Besides the classical multi-objective optimisation methods, a number of multi-objective optimisation genetic algorithm (MOGA) variants have been developed in the past decade. In a pioneering work in the field of Pareto-based MOGA, Fonseca and Fleming [48] developed an approach that is relatively easy to implement. But its performance is highly dependent on a parameter named “niche size”, which is

hard to define. Horn et al. [49] proposed a Niche Pareto GA (NPGA) that does not use a ranking method. Rather, Pareto domination tournaments are used to select individuals for crossover. NPGA runs very fast but its performance also depends on a specific parameter that is hard to set.

Recently, Deb et al. [50] presented an improved elitist genetic algorithm named Non-dominated Sorting Genetic Algorithm II (NSGA-II). The NSGA-II outperforms other MOGA variants due to three advanced operations: an improved non-dominated sorting approach for ranking solutions of a population, a crowded-comparison approach used in solution selection for diversity preservation and an elitism selection procedure that combines parent and offspring population. These operations are summarised in the following.

- *Ranking*: NSGA-II uses the ranking definition devised by to sort the solutions in the current population. All the non-dominated individuals in the current generation are assigned rank 1. These points are then removed from the generation temporarily and the next set of non-dominated individuals is identified and assigned rank 2. This process continues until the entire population is ranked. After the ranking procedure, fitness is assigned accordingly. From the implementation point of view, NSGA-II proposes a fast non-dominated sorting approach, which reduces the sorting computation complexity from  $O(MN^3)$  to  $O(MN^2)$ , where  $M$  denotes the number of optimisation objectives and  $N$  denotes the population size.
- *Crowded-comparison operator*: It is desired that a MOGA maintains a good spread of solutions on the Pareto front. To preserve diversity, most MOGA variants use fitness sharing for this aim, which involves a sharing parameter  $\sigma_{\text{share}}$ . The optimisation performance largely depends on the parameter setting. However it is a challenging issue to set the parameter for fitness sharing appropriately. NSGA-II replaces the sharing function with a parameter-free crowded-comparison operator  $<_n$ . This operator guides the selection process towards a uniformly spread-out Pareto-optimal front. Assume that every individual  $i$  in the population has two attributes: non-domination rank ( $i_{\text{rank}}$ ) and crowding distance ( $i_{\text{distance}}$ ). The partial order ( $<_n$ ) is defined as:

$$i <_n j, \text{ if } (i_{\text{rank}} < j_{\text{rank}}) \text{ or } ((i_{\text{rank}} = j_{\text{rank}}) \text{ and } (i_{\text{distance}} > j_{\text{distance}})) \quad (17.1)$$

That is, between two solutions with different non-domination ranks, the solution with the lower (better) rank is preferred. Otherwise, if both solutions belong to the same front, the solution that is located in a less crowded region is preferred.

- *Elitism selection procedure*: Given the current population  $P_t$  and its offspring population  $G_t$  which is generated from  $P_t$  and both of size  $N$ , the elitism selection procedure is used to create the next population  $P_{t+1}$  (see [50]). First, a combined population  $R_t = P_t \cup G_t$  is formed. The population  $R_t$  (size  $2N$ ) is sorted according to non-domination. Since all previous and current population members are included in  $R_t$ , elitism is ensured. The set  $F_1$  called non-dominated set contains all non-dominated solutions in  $R_t$ . If the size of  $F_1$  is smaller than  $N$ ,

we definitely choose all members of the set  $F_1$  for the new population  $P_{t+1}$  and remove these solutions from  $R_t$ . The remaining members of the population  $P_{t+1}$  are chosen from the new non-dominated set. Thus, solutions from the set  $F_2$  are chosen next, followed by solutions from the set  $F_3$ , and so on. This procedure is continued until no more sets can be accommodated. Say that  $F_k$  is the last non-dominated set beyond which no other set can be accommodated. To choose exactly  $N$  population members, the solutions of the last front  $F_k$  are sorted using the crowded-comparison operator  $<_n$  in descending order and choose the best solutions needed to fill all population slots. For more details about GA and different MOGAs, reader is directed to [50–52].

### 17.4.4 Adapted NSGA-II Algorithm

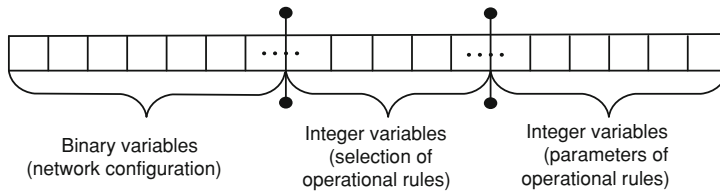
The main steps of our adapted NSGA-II algorithm can be summarised as follows:

- Step 1: Generate the initial population  $P$  of size  $N$ ;
- Step 2: Evaluate solutions in  $P$  using simulation;
- Step 3: Rank solutions in  $P$  by the non-dominated sorting approach;
- Step 4: Update the current best-so-far Pareto front  $P^*$ , i.e., Pareto filtering;
- Step 5: Select two parents for crossover by applying twice the binary tournament selection. This type of selection consists in first selecting randomly two solutions from  $P$  and then picking up the best-ranked one;
- Step 6: Perform crossover with probability  $p_c$  to generate two offspring;
- Step 7: Perform mutation with probability  $p_m$  for each offspring;
- Step 8: Perform feasibility check and repair for each offspring;
- Step 9: Add the two offspring in the offspring population  $G$ ;
- Step 10: Repeat steps 5–9 for obtain  $N$  offspring in  $G$ ;
- Step 11: Evaluate by simulation solutions in  $G$ ;
- Step 12: Generate the next population by the elitism selection procedure from  $P$  and  $G$ ;
- Step 13: Repeat steps 4–11 till termination condition is reach.

We describe various sub-modules of NSGA-II in the following subsections for solving our network design problem.

#### 17.4.4.1 Solution Encoding

Given a supply chain network with all existing and potential facilities (suppliers, plants and DCs), the network configuration consists in selecting a set of potential facilities to open and existing facilities to consider. More specifically, each potential facility is associated with a binary variable. These variables are coded using binary genes in the *chromosome* (a GA term representing the set of



**Fig. 17.5** The three-segment chromosome in the model

optimisation variables). A gene of value “1” indicates that the corresponding supplier or plant or DC is decided to be opened or considered. “0” means the corresponding facility is excluded in the present configuration.

One or more operation rules are associated with each existing/potential facility to sort different decision-making issues encountered in the operation of the facility. The selection of the operation rules to use and the setting of the control parameters of related operation rules are also encoded in the chromosome.

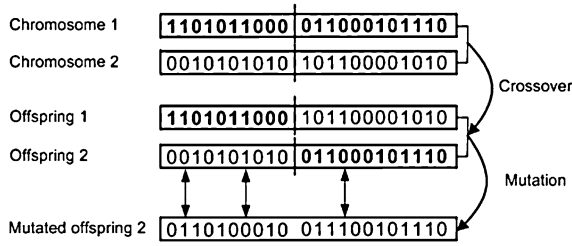
As a result, the solution is encoded in a 3-segment chromosome and designed with respect to the decisions (see Fig. 17.5): (i) a binary chain indicating the open/close decision of facilities (ii) a chain of integers indicating the operation rule for each potential decision-making issue, (iii) a chain of integers for all control parameters of all potential operation rules. Note that for a network configuration, segments 2–3 related to closed facilities are not relevant and do not have any impact on the performance of the network. Similar remark applies for segment 3 related to unselected operation rules.

#### 17.4.4.2 Crossover and Mutation

Due to the different natures of the three segments in the chromosome, crossover and mutation operations are performed independently on each segment. There are two basic parameters of GA-crossover probability and mutation probability. Crossover probability says how frequently crossover will be performed. If there is no crossover, offspring (child) is exact copy of parents. If there is a crossover, offspring is made from parts of parents’ chromosome. Crossover is made in hope that new chromosomes will have good parts of old chromosomes and the new chromosomes will be better. Mutation probability indicates how often will be parts of chromosome mutated. If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed. *Mutation is made to prevent the algorithm from falling into local extreme. The mutation rate should not be set too high; otherwise the algorithm will become random search in the extreme case.*

Classic crossover operators are implemented, such as one-point, two-point and uniform crossover. A unique mutation method is used, which flips each bit in the chromosome either from “0” to “1” or from “1” to “0” with given probability. For this problem, three crossovers are used for this 3-segmented chromosome.





**Fig. 17.6** Illustration of GA operators

Crossover operates on two chromosomes at a time and generates offspring by combining both chromosomes features. After the crossover operation, the resulting individual might fall into a local optimum. Hence, mutation is introduced as an operator to interfere the chromosome with certain possibility. Figure 17.6 illustrates two genetic operations: one-point crossover and mutation.

### 17.4.4.3 Chromosome Feasibility Check and its Repairing Strategy

As the network structure changes during genetic operations, it is necessary to introduce a network feasibility check and repair procedure. This procedure is used to verify the feasibility of candidate networks and repair the infeasible ones if some infeasible genes are occasionally generated (i.e., genes that violate any of the restrictions) during crossover and mutation. More specifically, in this chapter, we define the concept of network feasibility from the connectivity point of view. A *feasible* network should satisfy two conditions: (i) for each site in the network and for each type of input product, there is at least one valid upstream site which supplies the product; (ii) for each site in the network and for each type of output product, there is at least one valid downstream site which demands the product. To better understand the concept of feasibility, consider the production–distribution network design illustrated in Fig. 17.7.

Open or close decisions on three plants and DCs should be made. Given a chromosome [0, 1, 0, 0, 1, 0], only plant 2 and DC2 are valid, drawn by solid lines (non valid ones are drawn by dashed lines). As client1 demands for two types of products (T1, T2) and no plant is available to produce product T1, the network is considered infeasible. The repair procedure takes action by flipping the gene of plant 1 from 0 to 1, such that the supply of product T1 for client1 is guaranteed by plant 1. The corresponding chromosome is changed to [1, 1, 0, 0, 1, 0].

## 17.5 Automotive Real-life Case Study

In this section, we present a real-life case study from automotive industry to illustrate the effectiveness of the proposed approach.

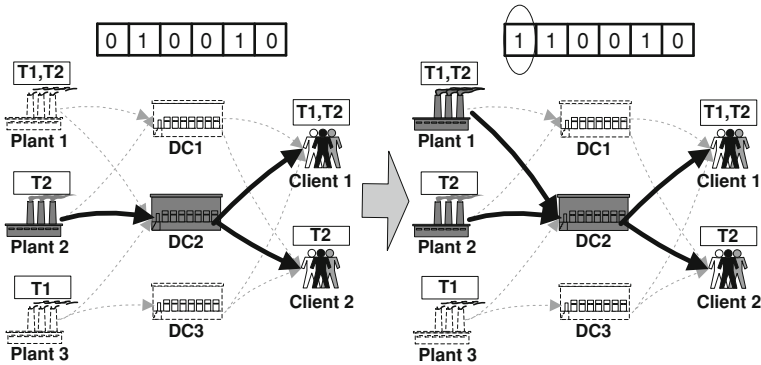


Fig. 17.7 Feasibility check and repair: case of production–distribution network

### 17.5.1 The Case Study

The automotive company is taking a strategic transformation effort and would like to improve the profitability and efficiency of its supply chain. The production–distribution network rationalisation project is one of the initiatives. As shown in Fig. 17.8, the network is composed of three plants, five distribution centres and six customer zones. More specifically, six customer zones generate independent stochastic demands for vehicles. Four regional distribution centres respectively RDC1, RDC2, RDC3 and RDC4 provide service to the six customers. Truck is the major transportation mode for vehicles delivery. All the vehicles are produced by the three plants (Plant 1, Plant 2 and Plant 3) and then consolidated in a nearby distribution centre (CDC). From the CDC, vehicles are transported to the four RDCs via two different transportation modes respectively boat and train. Stocks are only held in CDC and RDC1. The other three RDCs (RDC2, RDC3 and RDC4) are used as cross-dock points with temporary stock holding.

In order to keep the visibility of Fig. 17.8 to readers, intentionally we have hidden some links connecting RDCs and customers. In fact, each RDC is connected to each customer by exactly one transportation link using truck. There are 35 transportation links for which the characteristics are reported in Table 17.2.

In this study, each customer zone demand is divided into two parts: MTS demand and MTO demand. Each MTO demand is forwarded directly to the plant, which has the lowest workload. MTS demands are served by the RDCs. In case of stock-out at RDC1, MTS demands are backlogged and put in the waiting queue until the next replenishment. The MTS replenishment orders generated by RDC1 are sent to CDC, which replenishes its stock by sending production orders to the plants. It is a multi-echelon inventory system.

Since the network is managed in a “pull” manner, all vehicles produced in the plants already have their destination. There is no routing problem in this case. All vehicles produced in the three plants are first accumulated at CDC by train.

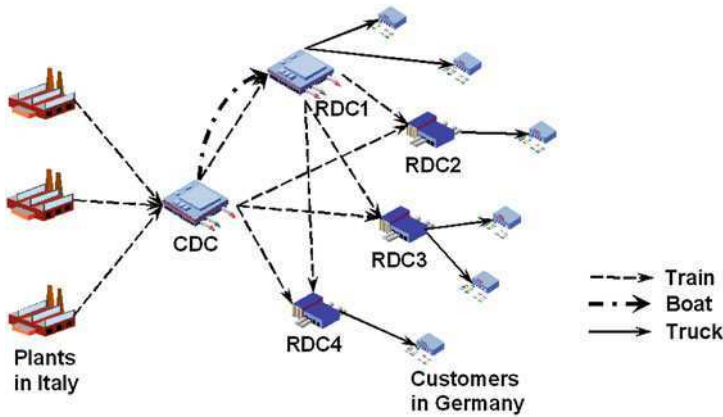


Fig. 17.8 The studied automotive production–distribution network

From CDC, MTO vehicles are sent to corresponding RDCs by train in order to reduce the transportation time. When inventory replenishment orders arrive at CDC, MTS vehicles are conveyed to RDC1 by boat for transportation cost saving. MTS vehicles in RDC1 can be delivered directly to its serving customers or transported to other RDCs (RDC2, RDC3 and RDC4) regarding the MTS demands generated by their customers.

The attributes of each facility considered in the modelling and simulation framework are listed respectively in Tables 17.3, 17.4 and 17.5. The data setting used for the numerical experiments is also presented in these tables. Note that, in Table 17.5, L1, L2 and L3 correspond to the transportations links between the three plants and CDC. Links L4 is the sea transportation link between CDC and RDC1. Links L5, L6, L7 and L8 represent the railway links between CDC and four RDCs, respectively. Links L9, L10 and L11 are inter-RDC links which connect RDC1 and other three RDCs for MTS vehicles forwarding. The rest are service links, which connect four RDCs and all the six customers. Note that the lead-time of link L4 follows a normal distribution with a standard deviation of 3 days. The random lead-time is due to the long distance and the use of boat as major transportation mode. All other transportation lead-times are set constant because of the related short transportation distance.

### 17.5.1.1 Operation Strategies

The overall operation mechanism is described as following. Six “customer” are modelled to generate weekly independent demands. The demand quantity follows a normal distribution with different parameters for different customers. Each customer demand is divided into MTS demand and MTO demand according to a given ratio (given by the company). Regarding the order assignment rule, each

**Table 17.2** Transportation links

ID	Transportation lead-time			Unit cost (€)	Min. batch size	Max. batch size	Departure interval (day)
	Distribution type	Mean (day)	Std. dev. (day)				
L1	Constant	1.0		10	500	500	1
L2	Constant	2.0		20	240	240	1
L3	Constant	3.0		30	240	240	1
L4	Normal	15.0	3	10	1000	5000	10
L5	Constant	5.0		100	240	240	3
L6	Constant	4.0		80	120	240	3
L7	Constant	3.0		70	120	240	3
L8	Constant	2.5		60	120	240	3
L9	Constant	1.0		20	120	240	3
L10	Constant	1.5		25	120	240	3
L11	Constant	2.0		30	120	240	3
L12	Constant	0.5		10			
L13	Constant	0.5		10			
L14	Constant	0.5		10			
L15	Constant	1.0		20			
L16	Constant	1.0		20			
L17	Constant	1.5		30			
L18	Constant	0.5		10			
L19	Constant	0.5		10			
L20	Constant	1.0		20			
L21	Constant	0.5		10			
L22	Constant	1.5		30			
L23	Constant	1.5		30			
L24	Constant	1.0		20			
L25	Constant	1.0		20			
L26	Constant	0.5		10			
L27	Constant	0.5		10			
L28	Constant	1.0		20			
L29	Constant	1.0		20			
L30	Constant	1.5		30			
L31	Constant	1.5		30			
L32	Constant	0.5		10			
L33	Constant	1.0		20			
L34	Constant	0.5		10			
L35	Constant	0.5		10			

**Table 17.3** Plants

ID	Production capacity	Production lead- time (day)	Unit production cost (€)	Minimum production lot size	Production frequency
P1	400	12.0	5000	200	Daily
P2	400	11.0	5500	200	Daily
P3	220	14.0	5500	100	Daily

**Table 17.4** Distribution centres

ID	Inventory check interval (day)	Unit inventory cost (€/day)	Unit ordering cost (€)	Storage capacity
CDC	2.0	10.0	10.0	5000
RDC1	7.0	10.7	10.0	1500
RDC2	None	34.0	10.0	200
RDC3	None	15.2	10.0	300
RDC4	None	32.3	10.0	300

**Table 17.5** Customers

ID	Demand		Demand interval (day)	MTO ratio (%)
	Mean	Std. dev		
C1	160	30		
C2	160	40	7.0	70
C3	160	10	7.0	70
C4	160	50	7.0	70
C5	160	10	7.0	70
C6	160	50	7.0	70

MTO demand is forwarded directly to a plant which has the *lowest workload*, while MTS demands are served by the RDCs with respect to the *on-hand inventory level* at RDC1. In case of stock-out at RDC1, MTS demands are backlogged until the next replenishment. We assume that it is possible to serve any customer from any RDC. But for each specific scenario encountered during the optimisation process, each customer is served by only one RDC for economics of scale. The service relationship between customer and RDC is determined according to the distance matrix between the customers and RDCs. Each customer is always served by the closest RDC available.

Two inventory control policies,  $(R, Q)$  and  $(s, S)$ , are considered for inventory replenishment of MTS vehicles in CDC and RDC1. The rule selection is considered as an optimisation option. As replenishment orders of CDC are usually of high volume, production orders of CDC are split into sub-orders according to the order assignment weight associated with each plant. Thus, the order assignment weight is also an optimisation variable. Then each plant is assigned its own production order for MTS vehicles and for MTO vehicles. A production capacity is set for each plant. FIFO (first-in-first-out) production policy is used to handle MTS and MTO orders of each plant, i.e., all the orders are treated sequentially according to their arrival time at the plant. Production lead-time is constant in this case. For all transportation links, transportation lead-time is modelled as random or constant parameters according to the scenario design.

To summarise, the decisions to be optimised include the open/close decisions on three plants and three RDCs (RDC2, RDC3 and RDC4). The production order assignment ratios are also to be optimised, which are indispensable in case with

multiple plants. Moreover, both qualitative and quantitative parameters are to be optimised for inventory control policies at CDC and RDC1.

### ***17.5.2 Instantiation of the Simulation-based NSGA-II***

The simulation scenario is developed based on the simulation framework presented in Sect. 17.4. No real data is presented in this chapter due to confidentiality. Instead, we focus on the description of important simulation-modelling features.

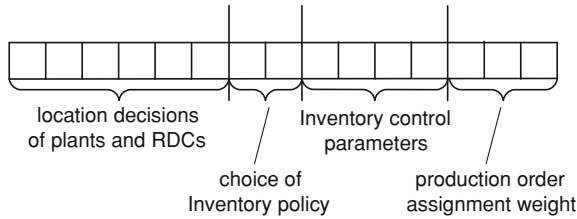
As described in Sect. 17.4, each simulation run would evaluate a scenario comprising a set of decision variables. In our case, the decision variables include the open/close decision for each plant (Plant 1, Plant 2 and Plant 3) and the three RDCs (RDC2, 3, 4). A production order assignment weight is associated with each open plant. Besides the strategic decisions, some qualitative variables, such as the inventory control parameters for CDC and RDC1, are also taken into account. Given the decision variables, we code the corresponding four-segment chromosome (Fig. 17.9). The first part contains six binary genes for open/close decisions of the six facilities. Two qualitative variables are used in the second segment indicating the choice of inventory policy in CDC and RDC1, of which the parameters are included in the third segment with four integers. At the end of the chromosome, three integers represent the production order assignment weight for each plant for MTS replenishment orders issued by CDC.

After the simulation run, a number of performance indicators are evaluated to measure the effectiveness and efficiency of the candidate network scenario. A variety of costs are evaluated, including investment cost for facility open/close, production cost, transportation cost, inventory holding cost and order handling cost. Besides the evaluation of different costs, three service level indicators are evaluated by the simulation model. The demand cycle time is the time period from the moment when a customer order is placed to the moment when corresponding vehicle is delivered to the customer. The on-time delivery rate is the percentage of orders that are delivered within a preset lead-time. The fill-rate for MTS demands is the percentage of MTS demands that are met directly from stock without backlogging. The three indicators are key measurements of the service level of a candidate network and should be taken into consideration together with the costs.

### ***17.5.3 Experimental Results and Analysis***

Some numerical experiments are conducted using a PC Pentium IV, 1.5 GHz and 512 Mo of RAM. The computational time is estimated to be 17.2 h. Note again that no real data is presented in this chapter due to confidentiality.

Regarding the strategic nature of production–distribution network design, the simulation horizon is set as 3 years in order to study more network dynamics.



**Fig. 17.9** Decision variables in the chromosome for the automotive case study

The warm-up period, required for the simulation model to reach a steady state, is set as 3 months. For each set of decision variables, 5 replications are simulated using different random number streams to smooth out residual randomness. Two optimisation criteria are considered: (i) minimisation of the average total cost for each vehicle unit; and (ii) minimisation of the average demand cycle time. Initial population is randomly generated. We run the optimiser for 2000 GA generations with a population size of 100. One-point crossover is employed, as the chromosome length is relatively short. The probability of crossover and the probability of mutation is set as 0.9 and 0.1 respectively. Other genetic operations are those of NSGA-II, such as the Pareto dominance ranking procedure and the elitist selection.

Figure 17.10 shows the best-so-far Pareto-optimal solution obtained. Each point in the figure represents a solution with a best-so-far compromise between the total unit average cost and the average customer demand cycle time. Each solution is with a different network configuration, selection of operation rules and parameter setting.

From left to right, we observe that the cycle time is reduced intensively while with very limited cost increase. This is the area that real supply chain operation managers look to improve. While with further reduction of the demand cycle time, the network cost increases tremendously. This is the area that draws attention in practice. The decision-maker should choose the most appropriate solution regarding the company's specific requirements on cost and demand cycle time. Furthermore, when we look in detail, we find that the solutions consist of mainly two types of network configurations, respectively Plant 1-Plant 2-CDC-RDC1 and Plant 1-CDC-RDC1. Two plants are used for more production capacity when short demand cycle time is required. Otherwise, only plant 1 is needed which is more cost-effective. All three RDCs (RDC2, RDC3 and RDC4) are closed. This centralised distribution network configuration with only RDC1 is expected to benefit from economics of scale.

## 17.6 Textile Real-life Case Study

A real-life case study is presented in this section to demonstrate how the proposed approach is applied to solve a complex problem from textile industry. Experimental results are analysed for validation.

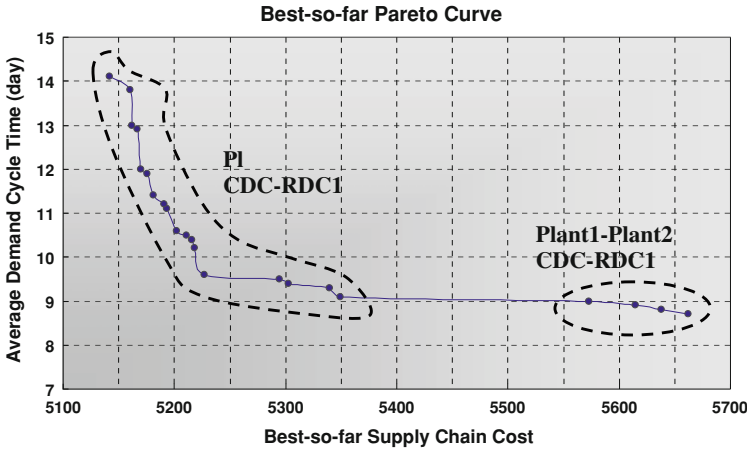


Fig. 17.10 Best-so-far curve comprising 22 different network solutions

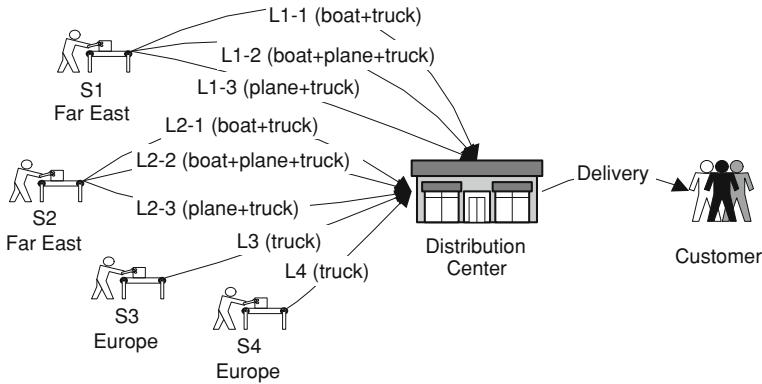
### 17.6.1 The Case Study

The supply chain studied in the case is operated by a European company in the textile industry. The company outsources its production to outside contractors and it focuses only on product design, marketing and distribution issues. In this case study, we consider one part of its global supply chain, which distributes a single type of product, “classic boot”, around Europe (Fig. 17.11). Due to confidentiality, no real data of this case study is presented in this chapter.

Actually, the company operates a central DC to hold stock of boots and to meet customer demands. A unique customer is considered representing the market. According to the inventory control policy, the DC places replenishment orders periodically. A unique supplier (S1 in Fig. 17.11) in the Far East is employed for stock replenishment. All purchasing orders are forwarded to S1 directly. There is only one transportation link (L1-1 in Fig. 17.11) that connects the DC and the supplier. After a period of supply lead-time, required boots are collected into containers and transported by boat from the Far East to a European harbour and then to the DC by trucks.

The motivation of the company’s effort on supplier portfolio optimisation is two-fold. Firstly, the current order-to-delivery lead-time (the period from the moment when the DC places a purchasing order to the moment when the DC receives required products) is relatively long, due to the long distance between the Far East and Europe, and the utilisation of boats as the principal transportation mode. Secondly, demands for “classic boots” have high variability and stock-outs frequently occur. Hence, the company selects three other suppliers, besides the existing one, as candidates to form a new supply portfolio. One potential supplier, denoted by S2, is also located in the Far East that provides the same type of boot at the lowest price. In order to reduce the order-to-delivery lead-time, the company





**Fig. 17.11** Supply chain network structure of the case study

also considers two additional transportation links for both suppliers from Asia. One transportation link is that products are conveyed by plane from Asia to a European airport, from where they are moved to the DC by truck. This is the fastest but also the most expensive way. An intermediate transportation solution is that products are transported by boat from the supplier to another Asian transit. Then the products are conveyed by plane to a European airport, from where they are finally moved to the DC by trucks. Two other potential suppliers are located in Europe, denoted by S3 and S4 respectively. These two suppliers are more responsive because they are closer to the DC. Only one transportation mode, using truck, is provided for each of them. Their lead-times are shorter, but on the other hand, their prices are higher than that of the two suppliers from the Far East.

The objective is to redesign the supply chain by possibly selecting new suppliers. Decisions related to order splitting; transportation allocation and inventory control should also be optimised. The decision-making is complicated by three issues:

- Multiple suppliers offer the same type of products at different prices and with different supply lead times. A supplier that offers a lower price provides products with a longer lead-time. The trade-off between price and lead-time should be studied.
- Some suppliers have more than one transportation link. Decisions incorporate transportation link selection and transportation volume allocation, besides the supplier selection and inventory control issues.
- Customer demand and transportation lead-time are stochastic. The uncertainties should be well addressed during the optimisation process.

### ***17.6.2 Instantiation of the Simulation-Based NSGA-II***

Given the features of the case study, we customise the simulation model accordingly. The attributes of each supply chain facility and specifications of each

**Table 17.6** Suppliers features

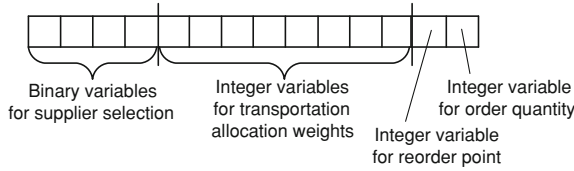
ID	Engagement cost (€)	Price (€/pair)	Duties (%)	Supply lead-time		Minimum order size (pair)
				Mean (day)	Std. dev. (day)	
S1	0	12.0	10	15.0	3.5	1000
S2	100000	10.0	20	20.0	4.0	1000
S3	80000	14.0	0	10.0	2.5	500
S4	100000	16.0	0	8.0	2.0	500

**Table 17.7** Transportation links features

ID	Transportation lead-time			Unit cost (€/pair)	Batch cost (€/batch)
	Distribution type	Mean (day)	Std. dev. (day)		
L1-1	Normal	20	4.0	0.5	500
L1-2	Normal	8	2.5	2.0	800
L1-3	Normal	5	1.5	4.0	500
L2-1	Normal	25	5.0	0.5	500
L2-2	Normal	10	2.5	2.0	800
L2-3	Normal	5	1.5	4.0	500
L3	Constant	4	1.0	1.0	300
L4	Constant	2	1.0	0.2	300

operational rule are filled in as input data. More precisely, the customer is assumed to generate daily demand. The demand quantity follows a normal distribution with a mean of 300 and a standard deviation of 50. The customer is defined as “impatient”, i.e., backorders are not allowed. Expected lead-time and service priority are not needed for the unique impatient customer case. The daily inventory holding cost is 0.25€ per pair of boots. The DC has a storage capacity of 3000 pairs of boots, with an over-capacity penalty cost of 10€ per pair per day. Each replenishment order incurs a fixed ordering cost of 100€. The parameters of the 4 candidate suppliers are summarised in Table 17.6. Supplier lead-times are assumed to be independent of the purchasing order quantity in this study due to the large production capacity of these candidate suppliers. Table 17.7 lists the features of all transportation links.

The operational rules are also specified according to the case study features. Customer demands are managed in a FCFS manner. The DC uses a weekly-review ( $R, Q$ ) rule for inventory control. In case of multiple suppliers, each replenishment order is split into sub-orders according to the order assignment weight associated with each supplier. Each sub-order is sent to the corresponding selected supplier. When products are ready, transportation volume is allocated among the transportation links according to the allocation weight of each link. Considering that the transportation is completely outsourced to a third-part logistics company, we assume that there is no limitation for transportation capacity. The allocated products are loaded as a whole and the carrier departs immediately.



**Fig. 17.12** Decision variables in the chromosome for the textile case study

In this study, the supplier portfolio is the focal decision to make. Simultaneously, parameters of the selected operational rules should be determined, including the order assignment weight for each supplier, the transportation allocation weight for each transportation link and two parameters of the  $(R, Q)$  inventory control rule. More specifically, the order assignment and transportation allocation are handled as follows. A weight  $w_j$  is assigned to each transportation link  $j$  and transportation volume is assigned to different links according to weights  $w_j$ . No weight is explicitly associated with a supplier  $i$  and we use the sum of weights of transportation links outgoing  $i$  as its weight  $W_i$ . Each inventory replenishment order is then split among all selected suppliers according to their weight  $W_i$ . Note that this is equivalent to assigning separately a weight to each supplier and other weights for transportation links. Figure 17.12 shows the structure of the customised chromosome comprising three segments.

The first segment of binary variables represents the decisions for supplier selection. The eight integers in the second segment represent the transportation allocation weight  $w_j$  assigned to each transportation link  $j$ . The weight can be an integer between 0 and 31. In the third part, there are two integers representing the reorder point  $R$  and order quantity  $Q$  of the distribution centre, respectively.  $R$  and  $Q$  can be an integer within [5000, 12000] and [500, 4000], respectively. These intervals are determined based on historic data and preliminary tests. Suppose that given a candidate solution, coded as  $(0,1,0,1)-(0,0,0,30,0,15,0,15)-(6200,2000)$ , the model builder will decode the string, and build up a simulation model correspondingly (Fig. 17.13). In this case, if the inventory position is discovered to be below 6200, a replenishment order of 2000 pairs of boots will be placed. 75% of the required boots will be supplied by S2 as the weights of the three transportation links of S2 are respectively 30, 0 and 15. Since the weight associated with link L2-2 is equal to 0, the transportation combination of plane and truck from S2 to DC is not utilised. Hence, 1000 pairs of boots (50%) are transported through link L2-1 and 500 pairs (25%) are transported through link L2-3. The rest (25%) are purchased from S4 and delivered through the unique link L4.

After each simulation run, the KPIs related to various costs are retrieved, together with the total demand quantity ( $Q_{\text{dmd}}$ ) and the total lost demands ( $Q_{\text{lost}}$ ). The optimisation objectives are defined as: (i) minimisation of the average unit cost per filled demand  $\mu_1 = C_{\text{total}}/(Q_{\text{dmd}} - Q_{\text{lost}})$ ; and (ii) maximisation of the demand fill-rate  $\mu_2 = (Q_{\text{dmd}} - Q_{\text{lost}})/Q_{\text{dmd}}$ .

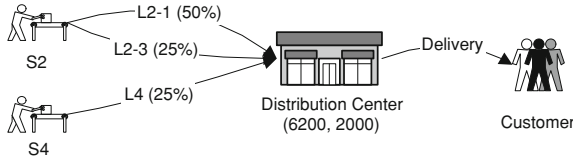


Fig. 17.13 Network representation of a candidate solution

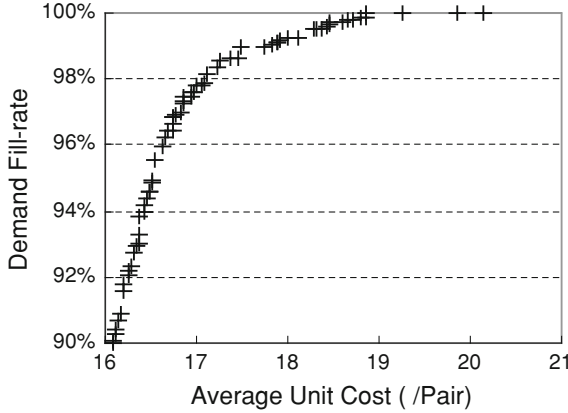


Fig. 17.14 The best-so-far Pareto front

### 17.6.3 Experimental Results and Analysis

The simulation horizon is set to 3 years, with a warm-up period of 3 months. For each set of decision variables, 10 independent simulation replications are performed. The performance indicators are averaged across simulation runs. These parameters are determined based on a series of preliminary tests. We further verify the simulation quality of the obtained Pareto-optimal solutions. A 95% confidence interval is constructed for the mean of average unit cost  $\mu_1$  and the mean of customer demand fill-rate  $\mu_2$ . As a result, the intervals obtained are around  $[\mu_1 - 0.1, \mu_1 + 0.1]$  and  $[\mu_2 - 0.01\%, \mu_2 + 0.01\%]$ , respectively. This indicates that the Pareto-optimal solutions are statistically reliable.

The NSGA-II-based optimiser is run for 2000 generations with a population size of 100. The probabilities of crossover and mutation are set as 0.9 and 0.1, respectively. Binary tournament selection and one-point crossover are employed. It takes 18.5 h to finish the computation on a machine with a CPU Pentium IV 1.5 GHz.

Considering that customer service level is critical for the company, we focus on the analysis of solutions that keep the demand fill-rate beyond 90%. Figure 17.14 shows the distribution of the best-so-far Pareto front for solutions with a demand fill-rate higher than 90%. Each of the 70 points represents a specific solution that is best-so-far Pareto-optimal. For further analysis, we look into the details of the best-so-far Pareto set.

**Table 17.8** Best-so-far Pareto solutions

$\mu_1$	$\mu_2$ (%)	Supplier portfolio	Transportation allocation weight	$R$	$Q$
20.13	100	S2 + S3	L2(19,16,0) + L3(20)	7239	1619
19.86	99.99	S2 + S3	L2(18,17,0) + L3(22)	7100	1661
...	...	...	...	...	...
18.32	99.51	S2 + S3	L2(22,19,0) + L3(22)	6813	1623
18.11	99.28	S2	L2(8,11,0)	8250	1308
18.01	99.24	S2	L2(16,18,0)	8429	1337
...	...	...	...	...	...
16.13	90.39	S2	L2(23,20,0)	6932	1555
16.12	90.25	S2	L2(13,0,0)	9821	1106
16.10	90.09	S2	L2(10,0,0)	10048	1104
16.08	90.01	S2	L2(13,0,0)	10446	1103

Table 17.8 summarises several important solutions. We observe that the solutions in the best-so-far Pareto set are composed of basically two categories of supplier portfolios. Solutions with a demand fill-rate higher than 99.5% have a common supplier portfolio that combines the cheapest supplier S2 with a European supplier S3. Solutions with a demand fill-rate lower than 99.5% use a unique supplier “S2” that is the cheapest supplier. Note that the most expensive transportation link L2-3, which uses plane as the major mode, is excluded in both types of supplier portfolio.

A number of tests are performed for sensitivity analysis. When the standard deviation  $\sigma_L$  of the supply lead-time of supplier S2 is changed from 4 to 6 days, the best-so-far solutions with a demand fill-rate higher than 99.63% turn to choose only one supplier S1 using all three transportation links. To obtain the demand fill-rate of 99.60%, the solution with  $\sigma_L = 4$  and the solution with  $\sigma_L = 6$  both choose the same supplier portfolio S2 + S3 and the transportation solution with links L2-1, L2-2 and L3. The reorder point  $R$  and the order quantity  $Q$  are respectively 6699 and 1653 if  $\sigma_L = 4$ , 7086 and 1582 if  $\sigma_L = 6$ . The average unit cost is 18.43 if  $\sigma_L = 4$  and 19.25 if  $\sigma_L = 6.7$ .

## 17.7 Conclusions and Future Work

We have presented a simulation-based multi-objective optimisation approach for design of supply chain networks by integrating the strategic network configuration decisions and the selection of best-suited operation strategies. The approach relies on the one hand on a multi-objective genetic algorithm and a generic modelling and simulation framework. The latter is designed in such a way to allow evaluation of a network with all possible configurations and all possible operation strategies. Stochastic phenomena along the whole supply chain (demand fluctuation, transportation uncertainty, etc.) are taken into account during the optimisation process.

As perspectives, more supply chain facilities are to be defined and implemented. Most importantly, a more comprehensive supply chain simulation framework should be established in order to apply the method for more cases. Some benchmarking works are also to be done for method performance assessment, comparing to the results obtained by analytical methods for some simplified cases. Another important research direction is to take into account risk related issues in the supply chain design. The approach presented in this chapter cannot be easily extended, as the computation time would be prohibitive.

## References

1. Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2003). *Designing and managing the supply chain: Concepts, strategies and case studies*. New York: McGraw-Hill.
2. National Research Council, Visionary manufacturing challenges for 2020 (1998). *Committee on visionary manufacturing challenges, board on manufacturing and engineering design, commission on engineering and technical systems*. Washington, DC: National Academy Press.
3. Schmidt, G., & Wilhelm, E. (2000). Strategic, tactical and operational decisions in multinational logistics networks: A review and discussion of modeling issues. *International Journal of Production Research*,39(7), 1501–1523.
4. Goetschalckx, M., Vidal, C. J., & Dogan, K. (2002). Modeling and design of global logistic systems: A review of integrated strategic and tactical models and design algorithms. *European Journal of Operational Research*,143, 1–18.
5. Meixell, M. J., & Gargeya, V. B. (2005). Global supply chain design: A literature review and critique. *Transportation Research Part E*,41, 531–550.
6. Klose, A., & Drexel, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research*,162, 4–29.
7. ReVelle, C. S., & Eiselt, H. A. (2005). Location analysis: A synthesis and survey. *European Journal of Operational Research*,165, 1–19.
8. Jain, V., Wadhwa, S., & Deshmukh, S. G. (2006). Modeling and analysis of supply chain dynamics: a high intelligent time petri net based approach. *International Journal of Industrial and Systems Engineering*,1(1/2), 59–86.
9. Zarándi, M. H. F., Turksen, I. B., & Saghiri, S. (2002). Supply chain: Crisp and fuzzy aspects. *International Journal of Applied Mathematics and Computer Science*,12(3), 423–435.
10. Swaminathan, M. J., Smith, S. F., & Sadeh, N. M. (1998). Modeling supply chain dynamics: A multiagent approach. *Decision Sciences*,29(3), 607–632.
11. Melo, M. T., Nickel, S., & Saldanha da Gama, F. (2006). Dynamic multi-commodity capacitated facility location: A mathematical modeling framework for strategic supply chain planning. *Computers and Operations Research*,33, 181–208.
12. Geoffrion, A. M., & Graves, G. W. (1974). Multi-commodity distribution system design by Bender's decomposition. *Management Science*,20, 822–844.
13. Cohen, M. A., & Lee, H. L. (1985). Manufacturing strategy: concepts and methods. In P. R. Kleindorfer (Ed.), *The Management of Productivity and Technology in Manufacturing* (pp. 153–188). New York: Plenum.
14. Cohen, M. A., & Lee, H. L. (1989). Resource deployment analysis of global manufacturing and distribution networks. *Journal of Manufacturing and Operations Management*,2, 81–104.
15. Arntzen, B. C., Brown, G. G., Harrison, T. P., & Trafton, L. L. (1995). Global supply chain management at digital equipment corporation. *Interfaces*,25, 69–93.
16. Pirkul, H., & Jayaraman, V. (1996). Production, transportation, and distribution planning in a multi-commodity tri-echelon system. *Transportation Sciences*,30(4), 291–302.

17. Pirkul, H., & Jayaraman, V. (1998). A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution. *Computers and Operations Research*, 25(10), 869–878.
18. Pirkul, H., & Jayaraman, V. (2001). Planning and coordination of production and distribution facilities for multiple commodities. *European Journal of Operational Research*, 133, 394–408.
19. Vila, D., Martel, A., & Beaugard, R. (2006). Designing logistics networks in divergent process industries: A methodology and its application to the lumber industry. *International Journal of Production Economics*, 102(2), 358–378.
20. Martel, A. (2006). The design of production-distribution networks: A mathematical programming approach. In J. Geunes & P. M. Pardalos (Eds.), *Supply chain optimization 98* (pp. 265–305). Springer series: applied optimization. Berlin: Springer
21. Snyder, L. V. (2006). Facility location under uncertainty: A review. *IIE Transactions*, 38(7), 547–564.
22. Louveaux, F. V. (1986). Discrete stochastic location models. *Annals of Operations Research*, 6, 23–34.
23. Ricciardi, N., Tadei, R., & Grosso, A. (2002). Optimal facility location with random throughput costs. *Computers and Operations Research*, 29, 593–607.
24. Erlebacher, S. J., & Meller, R. D. (2000). The interaction of location and inventory in designing distribution systems. *IIE Transactions*, 32, 155–166.
25. Daskin, M. S., Coullard, C. R., & Shen, Z.-J. M. (2002). An inventory-location model: formulation, solution algorithm and computational results. *Annals of Operations Research*, 110, 83–106.
26. Shen, Z.-J. M., Coullard, C. R., & Daskin, M. S. (2003). A joint location-inventory model. *Transportation Science*, 37(1), 40–55.
27. Snyder, L. V. (2004). *Supply chain robustness and reliability: Models and algorithms*. Ph.D. Dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA.
28. Shen, Z.-J. M. (2005). A multi-commodity supply chain design problem. *IIE Transactions*, 37, 753–762.
29. Tanonkou, G. A., Benyoucef, L., & Xie, X. (2006). Integrated facility location and supplier selection decisions in a distribution network design. *Proceedings of the 2nd IEEE International Conference on Service Operations and Logistics, and Informatics* (pp. 399–404). June 21–23, Shanghai (China).
30. Tanonkou, G. A., Benyoucef, L., & Xie, X. (2007). Design of multi-commodity distribution network with random demands and supply lead-times. *Proceedings of the 3rd IEEE International Conference on Automation Science and Engineering* (pp. 698–703). September 22–25, Scottsdale, AZ, USA.
31. Tanonkou, G. A., Benyoucef, L., & Xie, X. (2008). Design of stochastic distribution networks using Lagrangian relaxation. *IEEE Transactions on Automation Science and Engineering (TASE)*, 5(4), 597–608.
32. França, P. M., & Luna, H. P. L. (1982). Solving stochastic transportation-location problems by generalized Benders decomposition. *Transportation Science*, 16(2), 113–126.
33. Moynzadeh, K., & Nahmias, S. (1988). A continuous review model for an inventory system with two supply modes. *Management Science*, 34, 761–773.
34. Sculli, D., & Shum, Y. W. (1990). Analysis of a continuous review stock-control model with multiple suppliers. *Journal of Operational Research Society*, 41, 873–877.
35. Ramasesh, R. V., Ord, J. K., Hayya, J. C., & Pan, A. (1991). Sole versus dual sourcing in stochastic lead-time ( $s, Q$ ) inventory models. *Management Science*, 37, 428–443.
36. Lau, H. S., & Zhao, L. G. (1994). Dual sourcing cost-optimization with unrestricted lead-time distributions and order-split proportions. *IIE Transactions*, 26, 66–75.
37. Ganeshan, R., Boone, T., & Stenger, A. J. (2001). The impact of inventory and flow planning parameters on supply chain performance: An exploratory study. *International Journal of Production Economics*, 71, 111–118.

38. Sedarage, D., Fujiwara, O., & Luong, H. T. (1999). Determining optimal order splitting and reorder level for N-supplier inventory systems. *European Journal of Operational Research*, *116*, 389–404.
39. Ghodyspour, S. H., & O'Brien, C. (2001). The total cost of logistics in supplier selection, under conditions of multiple sourcing, multiple criteria and capacity constraint. *International Journal of Production Economics*, *73*, 15–27.
40. Qi, X. (2007). Order splitting with multiple capacitated suppliers. *European Journal of Operational Research*, *178*, 421–432.
41. Wang, G., Jiang, Z., Li, Z., & Liu, W. (2008). Supplier selection and order splitting in multiple-sourcing inventory systems. *Frontiers of Mechanical Engineering in China*, *3*(1), 23–27.
42. Ding, H., Benyoucef, L., & Xie, X. (2008). Simulation-based evolutionary multi-objective optimization approach for integrated decision-making in supplier selection. *International Journal of Computer Applications in Technology*, *31*(3/4), 144–157.
43. Slats, P. A., Bhola, B., Evers, J. J. M., & Dijkhuizen, G. (1995). Logistic chain modeling. *European Journal of Operational Research*, *87*, 1–20.
44. Beamon, B. M. (1998). Supply chain design and analysis: models and methods. *International Journal of Production Economics*, *55*, 281–294.
45. Sarmiento, A. M., & Nagi, R. (1999). A review of integrated analysis of production-distribution systems. *IIE Transactions*, *31*, 1061–1074.
46. Azadivar, F. (1999). Simulation optimization methodologies. *Proceedings of the 1999 Winter Simulation Conference*, *1* (pp. 93–100).
47. Lacksonen, T. (2001). Empirical comparison of search algorithms for discrete event simulation. *Computers and Industrial Engineering*, *40*(1/2), 133–148.
48. Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithm for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 416–423). Morgan Kaufmann: San Mateo, CA.
49. Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. *Proceedings of the 1st IEEE International Conference on Evolutionary Computation* (pp. 82–87). Piscataway, NJ.
50. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197.
51. Coello, C. A. C. (2000). An updated survey of ga-based multiobjective optimization techniques. *ACM Computing Surveys*, *32*, 109–143.
52. Goldberg, D. E. (1986). Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.



# Index

- A**
- Accuracy, 258, 330, 335–336, 409, 414
  - Adaptability, 365, 368, 374, 383
  - Aggregation, 191, 251, 255, 275
  - AHP, 44, 51
  - Algorithm, 38, 43, 45, 47, 49–50, 53, 58, 65–67, 71–72, 74–76, 82–83, 96–98, 108, 117, 121–122, 124, 127–128, 132–133, 139–140, 146, 148–150, 155–156, 158, 202, 213–214, 216–217, 263, 266, 284, 287–290, 293–295, 297–299, 301, 308, 315–316, 318–319, 327–328, 343, 345–347, 350–354, 356, 358–359, 361, 365, 373, 376–378, 380, 384–385, 387, 389–390, 393, 395, 398, 407, 413–414, 418, 436, 441, 447, 458, 465, 467–468, 475
    - evolutionary algorithm, 3, 27, 29, 53, 67, 74, 127–128, 132–133, 139, 158–159, 167, 186, 189, 191, 215, 217, 219–220, 222, 228, 230, 246–247, 282, 284, 301–305, 308, 313, 322–323, 344, 347, 360, 400, 427, 429, 452
    - genetic algorithm, 43, 49, 67, 98, 108, 128, 159, 190, 212, 217, 254, 284, 308, 346–347, 367, 369, 432, 455, 467–468, 473
    - statistical algorithm, 72
    - stochastic algorithm, 127
  - Analysis, 67–70, 79–80, 87, 92–94, 96–97, 104, 109, 114, 125–126, 128, 139–140, 147–148, 150, 153, 159, 194, 217, 251–254, 260, 264, 273, 283, 295, 299, 390, 401–406, 409, 422, 424–426, 432–434, 436, 438, 441, 443, 467, 468
    - data analysis, 252, 407
    - dependency analysis, 409
    - tolerance analysis, 252–255, 277
    - eigen-frequency analysis, 194
  - Ant colony optimisation, 251, 263, 285, 287, 347, 367
    - ACO, 291, 295, 346, 351
    - Daemon action, 264
    - decay, 263–264
    - encoding scheme, 288
    - evaporation rate, 268, 292, 294, 296, 353, 358
    - pheromone, 293, 352
  - Approximation, 32, 34, 72, 105, 132, 140, 142, 146, 148–149, 230, 244, 285, 304
  - Assembly, 47
  - Asymptotic convergence, 6
  - Attainment indicator, 21
  - Automation, 402
  - Automotive, 49, 127, 131, 139, 157, 159, 269, 402, 405, 416, 426, 433, 440, 443, 455, 458–459, 472, 476–478, 482
  - Availability, 52, 71, 76–77, 252, 263, 311, 318, 383, 424, 426, 434
- B**
- Backorder, 328, 485
  - Batch production, 325
  - Batch sequencing, 52, 58
  - Bill-of-material, 469
  - Binary gene, 474, 481

- B (cont.)** 328, 345, 367, 387, 446, 455, 458, 461, 466–469, 483–486
- Bloating, 23
  - Bonferroni Inequality, 419
  - Bottleneck, 426
  - Boundary condition, 83, 86
  - Branch-and-bound, 41–42, 466
  - Buffer, 66, 379, 418, 420, 423, 425, 427, 431, 433–434, 440, 442–444, 446–452, 458
  - Build-to-order, 40, 47, 68, 156
  - Bullwhip effect, 62
- C**
- Capacity, 43
  - Cash flow, 432
  - Casting feature, 256
  - Cellular layout, 367–368, 383
  - Chance constraint, 28
  - Clustering problem, 23
  - CO<sub>2</sub>, 281–282, 402
  - Commonality, 137–138, 202, 204, 216–217
  - Compatibility, 138
  - Complexity, 157, 461
  - Composite method, 48
  - Compromise, 11, 19, 46, 59, 119, 138, 140–143, 153–156, 158, 203, 455, 458, 466, 482
  - Compromise programming, 46, 59
  - Computational time, 7, 13, 29, 47, 96, 481
  - Confidence, 86, 238, 410–412, 416, 418–420, 422–423, 487
  - Configuration, 41
  - Constrained-domination, 19
  - Constraint, 41, 43–45, 47–48, 61, 86, 89, 91, 97, 107, 124, 143, 145, 150, 201, 215, 252–253, 255, 349, 446, 448, 462
    - capacity constraint, 491
    - $\epsilon$ -constraint, 59
    - machine constraint, 349
    - precedence constraint, 313, 349
    - productivity constraint, 328
    - thermal constraint, 97, 100
    - time constraint, 349
  - CONstant WIP, CONWIP, 417
  - Constraint method, 45
  - Container, 385–386, 388, 390, 394–396, 399–400, 483
  - Continuous variable, 168–169, 176, 190, 197, 201–202, 215–216, 463
  - Contract net, 307
  - Control, 62, 66, 69–70, 110, 125, 127, 147, 189, 192–194, 253, 308, 325–326, 328, 345, 367, 387, 446, 455, 458, 461, 466–469, 483–486
    - inventory control, 458
    - production control, 70, 428
    - tolerance control, 253
  - Control parameter, 66, 455, 458, 461, 467–468, 475, 481–482
  - Convergence, 6, 11, 13, 17, 20, 24, 31, 33, 53, 86–87, 159, 160, 230–231, 234–235, 238, 284, 290, 302, 347, 351–352, 371
  - Coordination, 306
  - Cost, 36–37, 40–47, 49–53, 56, 59, 61, 65, 71–72, 84, 87, 90, 99–100, 117, 125, 138, 143, 145, 147–148, 153, 190, 196, 204, 216, 251–252, 254–255, 258–266, 268, 272, 274–275, 279, 282, 291, 309, 311, 328, 332, 328, 332–333, 336, 338, 345, 365–367, 369, 372, 374, 376, 377, 381, 396, 402, 431–434, 436, 438–441, 443, 446, 448–452, 455, 458–459, 461–462, 464–466, 469–472, 478–480, 486, 487
    - administration cost, 456
    - annual cost, 439, 442, 449–450
    - delta cost, 439
    - distribution cost, 36, 42
    - inventory cost, 61
    - inventory holding cost, 469
    - investment cost, 61
    - labour cost, 438
    - maintenance cost, 439
    - manufacturing cost, 252
    - material handling cost, 366, 368
    - operating cost, 142–143
    - ordering cost, 466, 469
    - over-capacity cost, 469, 471
    - penalty cost, 328, 469, 485
    - procurement cost, 466
    - production cost, 36, 481
    - purchasing cost, 61
    - reconfiguration cost, 366, 383
    - relocation cost, 368, 372
    - running cost, 449, 452
    - shipment cost, 470–471
    - shortage cost, 52, 55–56, 466
    - supply chain cost, 56, 61
    - transportation cost, 42, 55, 61, 255, 456, 461–462, 464–465, 470, 472, 478, 481
    - warehousing cost, 456
    - wholesaler cost, 50, 56
  - Cost deployment, 433, 436–437, 453

- Cost estimation, 431–432, 436, 453
- Cost management, 433, 453
- Crossover, 47, 53, 87, 98, 117, 202, 293, 296, 316, 346, 371–373, 393, 464–466, 469–472, 478–480, 486–487, 476
  - blend crossover, 202
  - uniform crossover, 475
- Crossover operator, 5, 371
- Crowding distance, 14
- Customised algorithm, 5
- Customer assignment, 46, 56
- Customer demand, 41, 445
- Customer service, 40, 42–43, 55–56, 59, 62, 351, 455, 458–459, 465–467, 487
- Cycle time, 367
  
- D**
- Darwinian selection, 139
- Data analysis, 252, 407
- Data cleaning, 413
- Data field, 405
- Data integration, 407
- Data mining, 415, 452
- Data quality, 437
- Data reduction, 407
- Data selection, 406
- Datum, 258, 264
- Deformation, 73, 75–76, 81, 92, 123
- Decision making, 432
- Decision rule, 64, 141, 329
- Decision support, 158, 432
- Decision tree, 423
- Decision variable, 5, 8–10, 17, 23, 28–29, 38, 43, 139, 165, 200, 223, 345, 348–349, 387–388, 393, 401, 404, 412–413, 415, 418–420, 425–426, 458, 466–468, 481–482, 486–487
- Delaunay triangulation, 168–169
- Delivery performance, 45, 50, 57–58
- Delivery time, 47
- Demand, 328, 344, 357, 366
  - customer demand, 41, 445
  - energy demand, 48, 58
  - forecasted demand, 63–64
- Demand satisfaction, 62
- Demand uncertainty, 41
- Dendrogram, 175–176, 186
- Design, 325, 351, 360, 365–367
  - concurrent design, 138, 157
  - engineering design, 129
  - fixture design, 252, 277
  - layout design, 365–367, 369, 371, 373, 375, 377, 379, 381, 383–389, 393, 398
  - product design, 366
  - product family design, 137–143, 148, 154, 157–158, 161, 163–166, 168, 181, 185–187, 189–190, 192, 195, 198, 216–217
  - stiffness design, 193, 194
  - thermal design, 193
- Design of experiments, 405
- Design parameter, 392
- Design principle, 32, 86, 129, 162, 186, 221, 401, 403–404, 427, 432, 436, 453
- Design space, 388–389
- Design variable, 72, 75, 84, 87–89, 97–98, 107, 116–117, 121, 126, 139, 142–146, 162–166, 168–170, 172, 175–181, 183–185, 196, 211, 213
- Design vector, 163, 168, 170–172, 176, 185
- Deviation, 53, 90, 118, 143–144, 231, 256, 310, 411, 444, 449, 478, 485, 488
- Differential evolution, 31, 82
- Dimensionality, 33, 132, 139, 149, 155, 161, 164, 166–169, 181–182, 184–185, 187, 346
- Dispatching rule, 286, 308, 327–328
- Disruption, 326, 329, 343–346, 348, 359
- Distinctiveness, 138, 155
- Distortion, 92–93, 104
- Distributed GA, 44, 55
- Distribution, 36, 40–43, 46–47, 49, 56, 60, 67, 82, 86, 87, 90, 103, 105, 115, 117, 123–124, 130, 147, 287, 292–293, 297, 299, 408, 413, 458–466, 468–469, 476, 478, 481, 485
  - distribution centre, 462, 469
  - distribution network, 460
  - distribution planning, 43, 47, 56, 60
  - distribution strategy, 42
- Disturbance, 307, 328, 344–346, 366, 383, 411, 417
- Diversified solutions, 22
- Diversity, 473
- Driving variable, 163, 169–170
- Dominance, 121, 411
- Dual simplex, 45, 56
- Dynamic optimisation, 27
- Dynamics, 194, 458, 468
- $\epsilon$ -dominance, 146–147, 149, 155

**E**

- Effectiveness, 455
- Efficiency, 36, 39, 42, 47, 53, 68, 72–74, 81–82, 96, 106, 147, 279, 282–283, 307, 318, 332, 335–336, 346, 372, 386, 432–433, 461, 466, 477, 481
  - production efficiency, 282
- Elite set, 389, 394
- Elitism, 87
- Elitist strategy, 372
- EMO!Application, 17
- EMO!constraint handling, 19
- EMO!performance metrics, 20
- EMO!decision-making, 21
- EMO!hybrid approaches, 24
- EMO!dynamic, 27
- EMO!uncertainty handling, 28
- EMO!reliability based, 29
- EMO!meta-model based, 29
- Energy conservation, 282, 301
- Energy consumption, 281–283, 439
- Energy demand, 48, 58
- Energy efficiency, 280
- Epsilon-indicator, 21
- Equilibrium equation, 78
- Error, 103, 125, 251–254
  - classification error, 415
  - locating error, 252–253
  - machining error, 251–254
  - mean squared error, 414
  - stack-up error, 252, 258
- Euclidean distance, 86–87, 146, 171, 234
- Evaluation, 406
- Evolution strategy, 99, 128, 284, 302
- Evolutionary optimisation, 4
- Extended enterprise, 456

**F**

- Facility layout, 367
- Facility location, 39, 457
- Feasibility, 476
- Feasible decision variable space, 9
- Fill-rate, 461
- Fitness function, 369
- Flexibility, 36, 48, 311
- Flow-shop, 52, 291, 366
- Friction stir welding, 73, 75, 79–80, 126–132
- Function block, 308, 365, 368–369, 374–377, 380, 382–384
- Functional diversity, 162
- Fuzzy, 40–41, 43, 53, 56, 58–59, 61, 159, 347, 352, 458
  - fuzzy logic, 61

- fuzzy optimisation, 55–56
- fuzzy programming, 43
- fuzzy set, 43, 50, 58, 253, 276, 360

**G**

- Game theory, 308, 324
- Genetic algorithm, 43, 49, 67, 98, 108, 121, 128, 159, 190, 217, 254, 280, 284, 308, 346–347, 367, 369, 432, 467
  - chromosome, 475, 482
  - convergence, 6, 11, 13, 17, 20, 24, 31, 33, 53, 86, 87, 159, 160, 230–231, 234–235, 238, 284, 290, 302, 347, 351–352, 371
  - crossover, 47, 43, 88, 98, 117, 202, 293, 296, 316, 346, 371–372, 393, 478, 485
  - elitism, 87
  - fitness, 174
  - gene, 5–9, 12–17, 21–25, 28–32, 34–35, 37–38, 40, 43–44, 47–53, 67–70, 72, 75–78, 80–87, 89–90, 93–95, 97–98, 103, 105, 108, 113, 117, 123, 126–129, 132, 139, 142, 145–149, 155–156, 158–159, 161, 163–165, 171–172, 185–186, 190–192, 194–195, 197, 202, 210, 212–217, 219–220, 222–223, 227, 229–232, 234–236, 238–240, 242, 244, 246–247, 251–254, 256, 260–261, 263–264, 266–267, 273, 277, 280, 283–288, 293, 295, 298–299, 302–308, 311, 315, 318–325, 327–328, 343–344, 346–347, 351–354, 356–360, 365–367, 369–373, 379, 382–385, 387–389, 391–395, 397–399, 401–409, 411, 413–415, 417–418, 420, 422–424, 426–428, 432–433, 436–437, 441, 453, 455, 458–459, 463–478, 481–482, 485, 487–488, 490–491
    - mutation, 316, 476
    - offspring, 14, 230, 284, 371, 473–476
    - population, 268, 316, 473
- Genetic programming, 23, 32
- Geometric reasoning, 255, 270
- Global warming, 280
- Globalisation, 457
- Goal programming, 472
- Gradient-based approach, 167

**H**

- Heat flux, 77, 83, 86
- Heuristics, 141, 285, 367
- High-dimensionality, 346
- Hybrid approach, 50, 253, 276, 285, 367, 382–383, 458
- Hypervolume indicator, 21

**I**

- Identification, 26, 122, 129, 140, 149, 161, 185, 306, 353, 409
- Importance score, 412, 415, 420, 423–425
- Information flow, 402
- Initial condition, 105
- Initial solution, 411–413, 442
- Initialisation, 5, 394
- Innovation, 23–24, 32, 89, 122, 129, 162, 165, 186, 401, 403–405, 407, 409, 411, 413, 415, 417, 419, 421, 423, 425–427, 429, 432, 436, 453
- Input space, 413
- Integration, 441
- Interestingness, 409–410, 416, 428
- Interpretation, 406, 408–409, 412, 415
- Interruption, 286, 326, 328, 383
- Inventory, 36, 37, 40, 43, 47, 49, 50, 55, 60–63, 66, 138, 328, 387, 455, 458–460, 463–467, 469, 471, 478, 480–482, 485, 486
  - inventory control, 458
  - inventory cost, 55
  - inventory model, 50, 463, 490
  - inventory optimisation, 52, 69
  - inventory planning, 55–56, 58
  - inventory policy, 481–482
- Investment, 442

**J**

- Jacobian, 194, 199–200, 215
- Job-shop, 285, 301, 303–304, 323, 342, 359, 365, 367–368, 379, 382–383

**K**

- Karush-Kuhn-Tucker conditions, 6
- Kinematics, 192–194, 198–199, 203, 208, 215
- Knee point, 233
- Knowledge, 42, 74–75, 77, 83, 88, 89, 141, 144, 150, 156, 254, 287, 291, 308, 311, 317, 327, 402–403, 405, 409, 411, 434, 436, 452, 458
  - knowledge base, 216, 327
  - knowledge discovery, 403
  - knowledge extraction, 404

- KPI, 76, 80–87, 91–92, 96–97, 100–101, 122–123, 125, 252–254, 258–261, 264–266, 329, 467, 486
- KUR, 185

**L**

- Lagrangian relaxation, 463–465, 490
- Lateness, 279
- Latin Hypercube Design, LHD
- Lead time, 37, 252, 417, 441, 485
- Lean production, 426
- Learning, 74, 158, 308, 325–328, 334–335, 339, 342, 343–346, 348–350, 354, 356, 408, 409
  - supervised learning, 167–169, 178–179, 181, 187, 408–409
  - unsupervised learning, 167–169, 178–179, 181, 187, 408–409
- Linear programming, 36, 158, 466
- Logistics, 466
- Logistics network, 67–68, 463, 489–490

**M**

- Machining feature, 265–266, 270, 275
- Maintenance, 195, 285, 303, 328, 342, 439, 453
- Make-span, 279, 282, 309, 344
- Make-to-stock, 460, 463
- Malfunction, 326
- Manipulator, 199
- Manufacturing, 36, 40, 43, 45, 49, 50, 55, 71–73, 78, 102–103, 114, 117–118, 122, 125–126, 137–138, 190, 251–252, 254–256, 258, 260–261, 265–266, 276, 279–280, 282–285, 301, 306–309, 311, 313, 315, 319, 322, 325, 327–328, 330, 334–335, 338, 341–346, 348–350, 365–369, 354, 356, 365–366, 403, 417, 433–434, 437, 440, 460, 470
  - adaptive manufacturing, 383
  - flexible manufacturing, 348
  - green manufacturing, 280–281, 301
  - responsive manufacturing, 323
  - virtual manufacturing, 308, 323
- Manufacturability, 253, 276
- Many-objective, 122, 132, 137, 139–141, 143, 145, 147, 149, 151, 153, 155–159
- Mass customisation, 139
- Material flow, 86

**M** (*cont.*)

- Material handling, 365–368, 372, 374–377, 383, 417
- Mathematical programming, 36, 39, 46, 49, 56, 58, 67, 457, 463, 467, 490
- Matrix-based GA, 42
- Maximisation, 42, 47, 52, 96–97, 99, 486
- Measure, 37, 141, 143–144, 153, 193, 199, 201, 207, 204, 217, 291, 295, 299, 326, 388, 397, 410, 415, 461, 469, 481
  - interestingness measure, 409–410, 428
  - manipulability measure, 199, 204, 206–207, 215–216
  - objective measure, 409–410
  - semantics-based measure, 410
  - subjective measure, 409–410
- Mechanics, 75, 78, 81, 195
  - solid mechanics, 72, 75
  - thermo-mechanics, 78
- Mechatronic, 189, 192
- Memetic algorithm, 42, 58, 61, 66–67
- Meta-heuristics, 68, 367
- Meta-modal based EMO, 29
- Metal casting, 73–74, 102–103
- Micro-GA, 53
- Microstructure, 75, 81–82, 95, 104, 130–131
- Min-max criterion, 144
- Minimisation, 42, 47, 50, 52, 87, 96–97, 99–101, 117, 124, 126, 181, 220, 222, 230–231, 437, 443, 461, 482, 486
- Mixed-integer programming, 36, 457–458, 461, 463
- Model, 35–40, 43, 45, 47, 52, 58, 60–63, 65, 67, 69–70, 75, 77, 82–84, 123–124, 127, 159, 210–211, 213, 216, 253–255, 260–261, 264, 266, 275, 277, 279, 283, 287, 301, 307, 309, 326, 348, 350, 354, 385–389, 391, 393, 394, 404, 406, 408, 411, 414, 417–418, 427, 433, 435, 437, 438, 440–444, 446, 448, 458, 460–467, 475, 482, 484, 486
  - CAD model, 266
  - classification model, 413–414
  - cost model, 251, 255
  - data model, 407
  - descriptive model, 414
  - deterministic model, 461–462
  - distribution model, 67, 413, 463
  - dynamic model, 194, 209–210
  - economic model, 433, 436, 453
  - finite element model, 86, 130
  - heat source model, 82, 84–85
  - integration model, 307–308
  - location model, 254, 277, 457, 461, 464, 489–490
  - logistics model, 68, 457
  - loss model, 433, 437
  - material model, 77
  - mathematical model, 32, 43, 45–48, 62, 74–75, 85, 105, 131, 354, 489
  - mechanical model, 92
  - meta-model, 29, 34, 122, 127
  - metallurgical model, 92
  - numerical model, 72, 75, 84, 103, 121, 128, 130
  - pheromone model, 287
  - predictive model, 414, 423
  - process model, 76, 82, 104–106, 131, 405–406, 410–411, 469–470
  - scheduling model, 282, 301
  - simulation model, 49, 52, 54, 63, 66, 74, 194, 208, 216, 327, 342, 385–389, 391, 393–394, 399–400, 404, 411–412, 414, 417, 427, 429, 432, 435, 440–444, 458, 466–468, 472, 481–482, 484, 486
  - stochastic model, 43, 47, 463, 466
  - stress model, 93–94
  - thermal model, 83, 84
  - thermo-mechanical model, 78–79, 83, 90, 92–93, 123, 130
- Modelling, 73, 75, 191, 193–194, 209, 251, 283, 404, 408, 414, 432–433, 437, 441
  - analytically modelling, 84
  - cost modelling, 251, 433, 437
  - inventory modelling, 463
  - inverse modelling, 82
  - numerical modelling, 72, 75, 128, 130
  - process modelling, 104
  - simulation modelling, 342, 404
- Modularisation, 138
- Modularity, 209
- MOEA, 285
- MOGA, 473
- Monte Carlo, 28, 254, 337
- Morphology, 211
- Multi-objectivisation, 22
- Multi-agent, 69, 305, 307–308, 310–311, 313, 317–318, 323–326, 328, 334–335, 341–342, 457
- Multi-criteria, 51, 166–167, 219, 221, 223, 225, 227, 229, 231, 233, 235, 237, 239, 241, 243, 245, 247
- Multi-echelon, 40, 460, 477

Multi-physics, 71, 77, 86, 104, 130  
 Multi-modal, 5, 313  
 Mutation, 316, 476  
 Mutation operator, 6

## N

Negotiation, 308, 470  
 Net present value, 40, 57, 432  
 Network structure, 463, 468  
 Networked enterprise, 49, 68, 456  
 Neural network, 29, 31, 34, 162, 186, 308, 324, 360, 404, 418, 428  
 Niyama criterion, 112, 120, 131  
 Non-dominated solution, 4, 11, 14, 17, 22, 25, 37, 42, 44–45, 50, 52–53, 86–88, 117, 147, 149, 170–171, 174, 178, 180, 182–184, 229, 231, 234, 244, 292–295, 297, 299–300, 347, 447, 473  
 Non-dominated sorting, 87  
 Non-domination, 10, 14, 20, 146–147, 149, 166, 168, 170–172, 174–175, 179, 473  
 Nonlinearity, 76  
 NP-complete, 252, 262–264, 276  
 NPGA, 284, 347, 473  
 NP-hard, 464  
 NSGA, 17  
 NSGA-II, 14, 65, 87, 96

## O

Objective function, 3, 9, 12–13, 17, 22, 28–29, 31, 36–38, 41–46, 48, 50, 63, 84, 121, 126, 168, 170, 178–179, 191, 201–202, 212–213, 221–222, 231, 234, 261–262, 268, 287, 315, 326, 329, 332–341, 345, 347–350, 354, 357, 359, 367, 463–464, 467  
 Objective space, 9, 144  
 Operation, 36, 313, 316, 343, 476  
 Operation rule, 468, 471–472, 475, 482  
 Operator, 144, 395  
 Optimality, 346  
 Optimisation, 35–39, 43–57, 61–63, 65–66, 71–76, 82, 86–88, 92–94, 96, 99, 107, 114, 116, 117, 121–126, 137–141, 144, 146, 148–149, 155–156, 189–191, 195, 202, 211–212, 215–216, 252–255, 260, 263, 272, 279, 281–282, 284, 287, 305–308, 311, 313, 316, 318–320, 322, 343–344, 346–347, 349–352,

369, 385, 387–389, 393, 395, 401, 403–404, 407, 413, 418, 426, 431–439, 441–446, 448–450, 452, 455–456, 458, 463, 465–468, 473, 475, 480, 483–484  
 ant colony optimisation, 251, 263, 285, 287, 347, 367  
 combinatorial optimisation, 190  
 discrete optimisation, 190  
 engineering optimisation, 76  
 evolutionary optimisation, 282  
 multi-objective optimisation, 9, 401, 461  
 particle swarm optimisation, 219, 220, 228, 244  
 simulation-based optimisation, 49, 69, 432, 452, 458–459, 467–468  
 single-objective optimisation, 4, 9, 12–13, 22–23, 36, 38–39, 231  
 stochastic optimisation, 34, 467  
 Optimum, 7, 10, 13, 26–29, 36, 38, 75, 86, 99–100, 107, 117, 124, 126, 167, 170, 203, 242, 246–247, 291, 304, 316, 319, 358, 393, 476  
 Order splitting, 455, 466  
 Out-of-order, 375, 377, 382  
 Over-capacity, 469, 471, 485  
 Over-sampling, 414

## P

PAES, 284  
 Parallelisation, 75, 96, 123  
 Parallelism, 7, 31, 72, 346  
 Pareto bee colony, 48, 55, 61  
 Pareto filtering, 474  
 Pareto front, 38, 50, 65, 139, 171, 174, 186, 191, 202, 204, 209, 213–216, 241–245, 279, 282, 288, 290, 299, 397, 411, 418, 420, 424, 434, 447, 450, 473–474, 487  
 Pareto frontier, 139, 186, 213–214, 216, 279, 282, 288, 299  
 Pareto-optimal, 3–4, 10–15, 19–29, 33, 37, 40–41, 43, 45, 47–50, 52, 54, 65, 86–90, 99–100, 132–133, 147, 149, 159, 162, 165–167, 169–170, 177, 180, 189, 191, 198, 201, 219, 223, 231, 233–234, 238, 244, 261, 282, 288, 295, 298–299, 301, 347, 403, 407, 411–415, 417–418, 423–424, 433, 472–473, 482, 487  
 Pareto set, 100, 117–118, 139, 295, 299, 387, 393–395, 400, 487–488

**P** (*cont.*)

Part family, 161–164, 166–167, 170, 185  
 Particle swarm optimisation, PSO, 219–220, 228, 244  
 Pattern detection, 406–408, 411–412  
 Pattern search, 467  
 Penalty, 22, 32, 46, 139, 144, 151–152, 154, 157, 213, 328, 469, 485  
 Performance measure, 388  
 Performance metrics, 20  
 Permutation, 292  
 Perturbation, 126, 229, 231, 456  
 Petri net, 312  
 Pitfall, 433, 437, 441  
 Planning, 41–43, 45, 47, 55–56, 58  
   capacity planning, 45  
   distribution planning, 43, 47, 56, 60, 489  
   manufacturing planning, 307  
   motion planning, 189, 192, 194  
   operational planning, 67  
   process planning, 307  
   production planning, 68, 277, 323, 361, 384, 457  
   setup planning, 251–252, 257, 261, 266, 269  
   tolerance planning, 251, 265  
   trajectory planning, 210  
 Polynomial mutation, 6  
 Porosity, 107, 111, 116  
 Portfolio, 470  
 Post-optimality, 86, 88, 165, 401, 431–432, 434–436, 448  
 Posteriori approach, 21–22  
 Precedence relationship, 253, 306, 349, 354  
 Preference vector, 297  
 Prioritisation, 437  
 Probability, 298  
 Procurement, 40, 44–45, 58, 138, 456, 466  
 Product family, 196, 211, 216  
 Product mix, 366, 427  
 Product platform, 137–139, 157, 190, 195, 198, 202, 211  
 Product variety, 138, 143, 156–157, 159, 166, 186, 217  
 Production, 36, 40–43, 48–53, 60, 63–66, 83, 86, 90, 96, 118–119, 121, 137–138, 191, 252, 255, 257–258, 279, 281–283, 285, 306–308, 326, 328, 346, 352, 366–367, 401–405, 417, 423, 426–427, 431–435, 437–446, 448, 452, 457–459, 461–463, 466, 468, 469–471, 477–478, 480–483, 485  
   production capacity, 462

  production control, 70, 428  
   production efficiency, 282  
   production order, 49, 62, 65, 459, 469–471, 477, 480–482  
   production planning, 68, 277, 323, 361, 384, 457  
   production rate, 66, 86  
   production strategy, 52, 469  
   production systems analysis, 402  
   production variance, 65  
   virtual production, 72  
 Productivity, 387  
 Profit, 36  
 Profit distribution, 43  
 Profitability, 39, 55, 138, 141, 162, 402, 461, 477  
 Proliferation, 138

**R**

Recursive partitioning, 414  
 Redundant objective, 26  
 Reference direction, 21, 32  
 Reference frame, 78, 81, 84–85, 91, 93  
 Reference point, 21, 25, 32–33, 132, 230–234, 236, 238–240  
 Reinforcement learning, 328, 342  
 Reliability, 409  
 Reliability based EMO, 29  
 Replication, 385, 387, 389, 392–394, 417, 444, 446, 482, 487  
 Rescheduling, 308, 313, 323, 343–348, 350, 356–357, 359–360  
 Residual stress, 95, 97  
 Response time, 45, 49, 56  
 Responsiveness, 368  
 Reverse logistics, 48, 56, 58–59, 67–68  
 Robot, 190–192, 194–195, 198–202, 203–204, 210–211, 213–217, 369–370, 372, 374–379, 382–383  
   robot manipulator, 192, 194  
   robot weight, 212  
 Robotics, 192, 217, 246, 277, 342, 361, 383–384  
 Robustness, 33, 41, 51, 55, 69, 106–107, 126, 133, 296, 320–321, 344, 354, 490  
 Roulette wheel, 202, 371–373  
 Routing, 378

**S**  
 SBX, 6, 15, 87, 230  
 Scatter search, 45, 56



- Scheduling, 36, 48, 52, 252, 263, 279, 282, 285–288, 301, 305–309, 312–313, 318, 325–329, 331, 335, 341–345, 352, 427, 471  
 process scheduling, 48  
 procurement scheduling, 44  
 production scheduling, 44, 282, 302, 326, 328, 342, 352, 359  
 supply chain scheduling, 52, 67–68
- Semantics, 161, 409–410
- Sensitivity, 49, 99, 101, 124, 157, 176, 186, 295, 488
- Setup datum, 251, 257, 272
- Shifting bottleneck, 423, 429, 442, 444–445, 453
- Shipping, 51
- Shrinkage, 91
- Simulated binary crossover, 30, 87
- Simulation, 35, 38–39, 48–52, 71–77, 87, 93–94, 103–105, 107, 114, 118, 121–122, 125  
 agent based simulation, 39  
 casting simulation, 103–104, 131  
 cost simulation, 437  
 discrete-event simulation, 469  
 dynamic simulation, 194, 212–213, 216  
 high fidelity simulation, 72  
 mechanical simulation, 71, 73, 79, 93, 97, 106, 127  
 Monte-Carlo simulation, 148–149  
 motion simulation, 194  
 numerical simulation, 71, 73, 75–76, 108, 130  
 process simulation, 72–77, 107, 121–122, 126, 131  
 solidification simulation, 105  
 thermo-mechanical simulation, 79
- Skeleton layout, 390–391
- Solidification, 110, 113, 117
- Solution space, 198
- SPEA, 25, 27, 32, 53, 58, 77, 221, 234, 280, 284–285, 293, 297–302, 413, 417, 429
- Specification, 142, 193, 252–255, 264, 270, 276, 306, 391, 395–396, 484
- Spring back, 73
- Stability, 48, 66, 73, 127, 138, 221, 252, 359, 458
- Stack layout, 386
- Stack-up, 252
- Standard deviation, 90, 231, 310, 444, 449, 478, 485, 488
- Standardisation, 138, 162
- State transition, 327
- STEP, 11–12, 49, 52, 58, 64, 77, 84, 87, 92, 103–104, 107, 114, 117, 120, 140–141, 165–166, 168–169, 176, 193–195, 208, 210, 220–221, 227, 260, 263, 283, 287, 296, 308, 315, 327, 329, 332, 334–335, 354, 358–359, 369, 372, 393–394, 403, 405–407, 409–411, 413–415, 418, 434, 438, 442, 446, 474
- Stochastic programming, 57
- Stochastic search, 6, 287, 467
- Stock-out, 63, 460, 477, 480, 483
- Supplier reliability, 465
- Supplier selection, 47, 51, 66, 466
- Supply chain, 36–40, 42, 54, 65, 427, 484
- Supply chain design, 39, 457, 466, 471
- Supply chain network, 40
- Supply network, 44–45, 456
- System, 35, 37–39, 45, 48, 52, 62–63, 75–76, 84, 106, 108–110, 113–115, 157, 159, 189, 191, 193, 208, 251, 253–254, 258, 264, 269, 277, 286–287, 305–307, 310–311, 318, 325–328, 334, 343, 345–347, 351–352, 367–369, 374, 386, 388, 391, 393, 401–404, 417–418, 420, 422, 427, 431–435, 437, 438, 441–443, 445, 449, 451, 457–458, 460, 463–464, 466, 468, 477  
 production system, 66, 166, 283, 302–303, 328, 401–405, 426–427, 431–443, 445, 447, 449, 451–453
- System dynamics, 35, 62–63, 69, 427
- T**
- Tabu search, 48–49, 55, 308, 324, 353
- Tardiness, 45, 56, 286, 309–310, 328
- Tardy job, 69, 279, 282
- Termination criteria, 6
- Test problem, 15, 23, 26, 28, 30, 32, 146, 293
- Theory of Constraints, TOC, 423
- Time-to-market, 437
- Tolerance, 252, 260, 265
- Tool access direction, TAD, 251
- Tool approach direction, TAD, 255
- Tool orientation space, TOS, 257
- Topology, 124, 132, 168, 172, 211, 247
- Tournament selection, 5, 19, 293
- Trade-off, 3–4, 10–13, 17–20, 22–23, 26, 30–31, 37–38, 40–41, 43, 45–46, 48, 50, 63, 65–66, 72, 82, 86–88, 97, 99–101, 117, 122, 137–141,

**T** (*cont.*)

143–144, 146–150, 153, 155–156,  
 162, 164–165, 168, 189–191, 195,  
 201–202, 204, 209, 214–216, 220,  
 223, 232, 234–235, 238, 241–245,  
 260, 272, 274, 343, 345, 348–349,  
 352, 403–404, 417, 423, 426,  
 433–435, 441, 448, 467, 484

Trajectory, 17–18, 30, 194, 210, 213

Transportation, 37, 43, 427, 457, 466, 471

Triangulation, 168–169

Throughput, 426, 431–432, 439, 444

**U**

Uncertainty, 158, 365, 467

Under-sampling, 413

Up-time, 434, 440

Utility function, 21–22, 31, 44–45, 55, 57,  
 142, 232–233, 236, 325–326,  
 329–336, 339, 341, 410, 464

Utility indicator, 21

**V**

Value creation, 432

Variability, 41, 62, 147, 366, 483

Variation operator, 5

VEGA, 284

Vehicle routing, 53, 58, 346, 357, 360–361

Virtual cell, 367–368, 383

Visual analytics, 137–141, 143–145, 147–149,  
 151, 153–157, 159

Visualisation, 141, 184, 195–196, 204,  
 207–209, 412

**W**

Warm-up time, 444

Waste, 280

Weighted objective, 46

Work-in-process, WIP, 37, 417, 431, 432

**Z**

ZDT2, 1