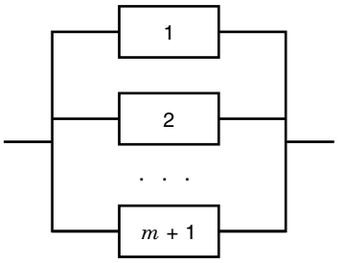

Figure 1. Redundant group of $m + 1$ units in parallel.

RELIABILITY OF REDUNDANT AND FAULT-TOLERANT SYSTEMS

Among the varied methods for improving the reliability of an engineering design, redundancy plays an important role. Redundancy is usually understood as the use of additional standby units to protect a system against the failure of its operating units. A redundant system does not necessarily have protection against a catastrophic failure of all (or most) of its units because some units are connected in series.

In this article we consider only structural redundancy of dependent units. This involves the use of standby redundant units in different ways or the specialized use of additional units as in a bridge network structure. There are other methods of achieving functional redundancy in a system (1). For instance, under the load-sharing regime, operating units work with a loading less than nominal. Systems with time redundancy have extra time to compensate for the consequences of current failures.

The effect of redundancy can be dramatically increased if one uses renewal (repair or replacement) of failed units. Redundancy with restoration is considered in more detail in the article entitled REPAIRABLE SYSTEMS in this encyclopedia. Some discussion on systems with dependent units can be found in Ref. 2.

MANY FLAVORS OF STRUCTURAL REDUNDANCY

Even structural redundancy comes in many flavors. A system is in *hot standby* if there are more units in operation than needed and all units are in the operational mode. Comparison with a *cold standby* scenario clarifies the previous statement. The use of cold standby originates from the belief that units on cold standby do not fail and degrade very slowly, compared with their counterparts on active duty. Spare units in storage come closest to this definition. A notion of *warm standby* is also found in the literature. In this case the units are supposed to be ready but not operational. They can be switched in when needed. Units on warm standby are assumed to degrade and fail, albeit more slowly than the units on hot standby. Mathematical models of the warm standby situation generally involve details unique to a system, which we will not pursue here.

Another dimension of redundancy is the notion of individual or group standby. Group standby is effective in a computer (possibly embedded) network. All components may be loaded with a specific software and may have unique roles in operation, whereas the hardware may be generic. Under this situation it may be enough to use a few redundant hardware units which may be configured to provide the functionality of a failed unit. By similar reasoning, one can see that same model also works for a multichannel communication system.

In this discussion we have implicitly assumed that the switches responsible for swapping failed operating units with a standby (hot or cold) unit work perfectly and instantly. This hardly represents the real situation. We will talk about the role and limitation of real switches later in the article.

Individual Hot Standby

Individual hot standby is identical to a parallel system under the assumption of perfect switching. Thus, a standby unit is in the same regime as an operational unit. In most situations all units in such a redundant group (RG) are considered stochastically identical. An RG of $(m + 1)$ units (an operational unit and m standby units) is assumed to be successfully operating if at least one unit of the group is in the up state. A reliability block diagram (RBD) of this redundant group is shown in Fig. 1. If ξ_k is the random time to failure for unit k, then the RG random time to failure $\xi_{RG} = \max_{1 \le k \le m+1}$ (see Fig. 2).

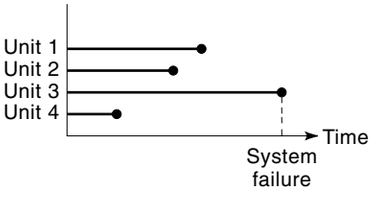

Figure 2. Time diagram for a group of four hot standby units.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

The probability of failure-free operation (PFFO) of the group can be written in a very simple form:

$$P_{\mathrm{RG}} = 1 - (1 - p)^{m+1}$$

The unit PFFO is denoted by p in this expression. The mean time to failure (MTTF) is easily written only for the exponential distribution of time to failure of individual units if the units are statistically identical:

$$T_{\mathrm{RG}} = T\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{m+1}\right)$$

The unit MTTF is denoted by T in this expression. Formally, the RG MTTF increases without limit with increasing number of units. The rise is very slow, however, and is merely logarithmic in m. We would like to mention without proof that for given T and m, T_{RG} is high if unit time to failure (TTF) exhibits large dispersion.

Individual Cold Standby

Redundant units on cold standby are assumed not to fail while not operating. If the cold standby implies spares, the time to switch them in may not be neglected. Thus the system with only cold standby units is exposed to system failure. In this case it may be more appropriate to talk about sufficiency of a spare stock rather than system reliability. All units (operating and redundant) are assumed identical with the same MTTF, T. A time diagram for such a system is presented in Fig. 3.

If a unit has m standby redundant ones, then the MTTF of such a redundant group $T_{\mathrm{RG}} = (m + 1)T$. In this case the PFFO of the RG, $P^{(m+1)}(t)$, can be expressed in more complex way than for *hot* redundancy:

$$P^{(m+1)}(t) = \int_0^t P^{(m)}(t - x) f(x)\, dx$$

where $P^{(k)}(t)$ is the convolution of order k of the unit's PFFO and $f(t)$ is the density function of the distribution of the unit's time to failure. The latter expression has a simple form if $f(t)$ is an exponential function $f(t) = \lambda \exp(-\lambda t)$:

$$P^{(m+1)}(t) = \sum_{k=0}^{m} \frac{(\lambda t)^j}{j!} \cdot \exp(-\lambda t) = 1 - \sum_{j=m+1}^{\infty} \frac{(\lambda t)^j}{j!} \cdot \exp(-\lambda t)$$

Thus $P^{(m+1)}(t)$ involves a Poisson distribution. The defining parameter of the Poisson distribution λt is the mean number of failures in a time interval of length t. For highly reliable units, that is, $\lambda t \ll 1$, one can use an approximation:

$$P^{(m+1)}(t) \approx 1 - \frac{(\lambda t)^{m+1}}{(m+1)!} \cdot \exp(-\lambda t)$$

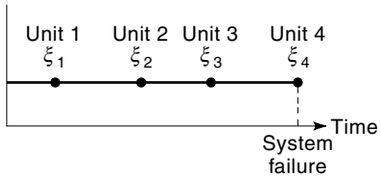

Figure 3. Time diagram for a group of four cold standby units.

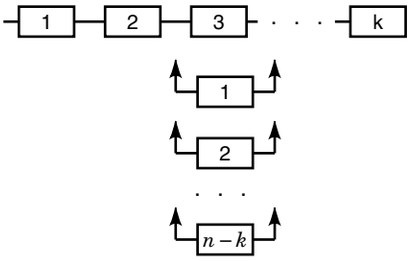

Figure 4. Redundant group of k main and $n - k$ standby units.

Group Hot Standby

A routine engineering practice is to use a group of common standby units for a group of operating units (see Fig. 4). Such a structure is sometimes called *k-out-of-n,* implying that the system operates successfully if k units among the total of n survive. Usually, such cases appear in communication and computer systems. This is an obvious generalization of a parallel system, where $k = 1$. The case with $k = n - 1$ is sometimes called the *fail safe* configuration. In the following discussion we will denote the number of operating units by k and the total number of the system's units by n (the number of hot standby units equals $n - k$).

Each unit in the RG fails at a random moment. We denote the random time to failure of unit j by ξ_j and order them in increasing order. Now we introduce another notation: $\xi_{(k)}$ is the time when the kth unit fails. By construction,

$$\min_{1 \le k \le n} \xi_k = \xi_{(1)} < \xi_{(2)} < \cdots < \max_{1 \le k \le n} \xi_k$$

In this case the RG's random time to failure $\xi_{RG} = \xi_{(n-k+1)}$ (see Fig. 5).

The PFFO of the RG can be written as

$$P_{\mathrm{RG}} = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j} = \sum_{j=0}^{n-k} \binom{n}{j} (1-p)^j p^{n-j}$$
$$= 1 - \sum_{j=n-k+1}^{n} \binom{n}{j} (1-p)^j p^{n-j}$$

It follows from the very right part of the preceding formula that, for highly reliable units ($1 - p \ll 1/n$), the approximate

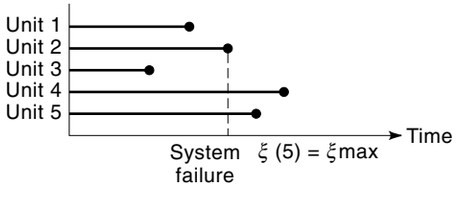

Figure 5. Time diagram for a group of three operating and two hot standby units ($\xi_{(3)}$ is the moment of system failure).

value for the RG PFFO is given by

$$P_{\mathrm{RG}} \approx 1 - \binom{n}{n-k+1}(1-p)^{n-k+1}p^{k-1}$$

The MTTF can be easily written only for the exponential distribution:

$$T_{\mathrm{RG}} = T\left(\frac{1}{k} + \frac{1}{k+1} + \cdots + \frac{1}{n}\right) = T\sum_{j=k}^{n}\frac{1}{j}$$

It is important to notice that group hot standby is effective for increasing the system PFFO. The MTTF, however increases very slowly.

Group Cold Standby

An economical way of achieving the benefits of redundancy is to use a group of $m = n - k$ cold standby units for a group of k operating units if short interruptions to system operation can be tolerated. In the general case the description of the process is very complicated. Simple results are known only for the exponential TTF distribution. The PFFO of the system can be written as

$$P_{\mathrm{RG}} = \sum_{j=0}^{n-k}\frac{(kt\lambda)^j}{j!}\exp(-kt\lambda) = 1 - \sum_{j=n-k+1}^{\infty}\frac{(kt\lambda)^j}{j!}\exp(-kt\lambda)$$

The MTTF can be easily written for the exponential distribution as

$$T_{\mathrm{RG}} = (n-k+1)\cdot\frac{T}{k}$$

For arbitrary distribution of unit time to failure in the case of group cold standby, numerical results can be obtained with the help of Monte Carlo simulation.

Voting Systems

A common use of a voting system is in mission-critical software decisions. In such systems, n independent outputs (signals) are compared against each other. If k signals coincide, the system is assumed to be operating successfully. If the number of possible outputs are many in number (like correction to a spacecraft trajectory), two identical but independently computed outputs often offer a reasonable level of confidence unless there is a tie.

At the other extreme of binary decisions, the situation is somewhat different. If there are only two possible answers, a correct (output) signal or a mistaken signal, the system output signal corresponds to the majority signal. Of course, there is a possibility of adopting the wrong signal for the system if the majority reports a mistaken signal. The probability of a mistaken output for the system as a whole is small if each unit generates the correct signal with relatively high probability. It is easy to see that this system is a modified version of the group hot standby model.

Redundancy Depth

The effectiveness of redundancy depends on the application depth. In most cases, the smaller a part (deeper application) which uses standby, the better. We illustrate this (Fig. 6) by considering the limiting cases of system-level and unit-level redundancy for a series system under hot standby.

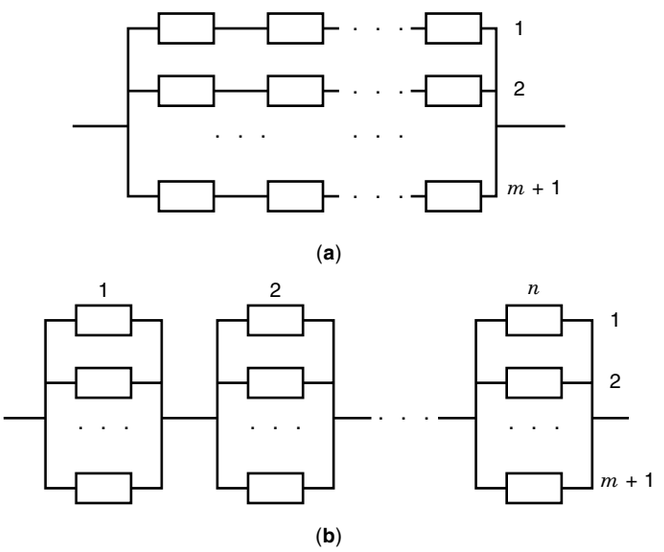

Figure 6. A series system with system-level and unit-level redundancy: (a) system-level; (b) unit-level.

Hot standby system-level redundancy means that $(m + 1)$ series circuits of n different independent units are connected in parallel. The PFFO of such a redundant system is given by

$$P_{\mathrm{System}} = 1 - \left(1 - \prod_{k=1}^{n} p_k\right)^{m+1}$$

In the case of unit-level redundancy, each unit j of the circuit has m redundant units for itself. In this case,

$$P^*_{\mathrm{System}} = \prod_{k=1}^{n}[1 - (1-p_k)^{m+1}]$$

It can be shown that $P^*_{\mathrm{System}} > P_{\mathrm{System}}$ (1,3).

FAILURE MONITORING AND ROLE OF SWITCHES

It is usually assumed that standby units replace failed units instantaneously and certainly. Engineers know, however, that the problem of replacing a failed unit is not that simple. In most cases, the reliability of the switching device is a restricting factor. A redundant group might be very reliable, but the switch itself becomes the *troublemaker*.

To perform switching effectively and quickly, one must have built-in monitoring equipment that monitors all units thoroughly and frequently. We emphasize that the monitoring device must monitor all units (at least for group hot redundancy) and not just the operating ones. Otherwise some redundant units may fail quietly and switching will have no real effect on reliability. The monitoring device itself is subject to hardware and software failures. Some discussion of the problem can be found in Refs. 1 and 4.

So far all units are implicitly assumed to be bistate, operating or failed. A switch is a tristate unit because it has

two distinct failure modes, failure to switch when needed and failure to idle (premature and unnecessary switching). Reliability analysis of a system involving a switch connected to many units (Fig. 7) is complicated because the switch may get stuck with one unit and it may not be able to connect to another with positive probability.

To simplify the situation, the reliability analysis of a switch is generally performed by modeling the switch as an abstract two-state object considered in series with the redundant components. An RBD with such an abstract switch is called a *relay* configuration (5).

We remind the readers that cold redundancy is usually considered a model for spare units. Any replacement of a failed unit by a spare takes some time. In this case the role of the noninstant switch is played by the repairperson. If a failed unit is not redundant, then the system interrupts its operation for the replacement time.

REDUNDANCY WITH REPAIRABLE UNITS

So far we have implicitly assumed that failed units are never repaired but are replaced with new ones when needed. However, most failed units are not thrown out. They are repaired where appropriate and retained for the future use. So, there is some kind of unit circulation:

- An operational unit fails;
- The failed unit is replaced by a spare unit; and
- The failed unit is repaired and becomes a spare unit.

If a nonredundant unit fails and it has to be replaced or repaired before the system can be returned to the operating state, the system downtime can be unacceptably high. A judicious use of spares (cold redundancy) with (hot) redundancy reduces the system downtime to an acceptable level. If a failed unit is redundant, then an almost instant switch for the standby unit might not influence the system operation. If replacement (or even repair) of a failed unit is fast enough,

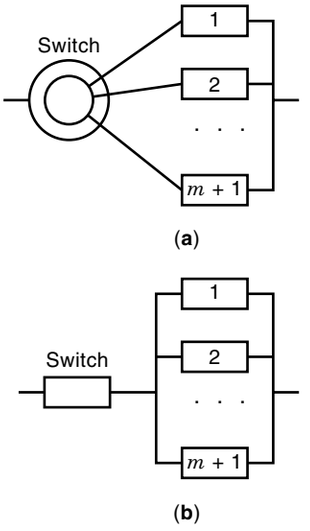

Figure 7. Redundant group with a switch; (a) real switch; (b) abstract switch.

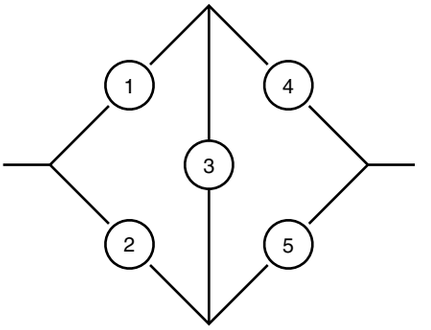

Figure 8. Bridge structure.

the redundant unit is back in operation and again the system is protected against a failure. System failure occurs only if all units of the redundant group fail during the renewal procedure. For highly reliable systems the probability of a second failure during the replacement (repair) time is insignificant. For more details concerning the reliability analysis of repairable systems, the reader may consult the article entitled REPAIRABLE SYSTEMS in this encyclopedia.

NETWORK REDUNDANCY

One of the most sophisticated methods of redundancy is represented by network structures. In this case all units (links) can be considered operating units. At the same time, failure of some of them might not affect performance of the network as a whole. The impact of failure depends on the failure location, the current system loading (for instance, the level of traffic in a telecommunication network), the algorithm of network operation, and other factors. We will only consider two-pole networks to confine the discussion to standard reliability block diagrams. The main feature of a general network structure is that it cannot be reduced to series–parallel and parallel–series connections.

Irreducible Bridge

Obviously there are systems which cannot be reduced to a combination of series and parallel structures. The simplest planar structure of this kind is called a *bridge structure* (Fig. 8). There are other structures which may not be depicted on a plane. A general structure of this kind is analyzed by studying its *paths* and *cuts*.

A system with a two-pole network structure is assumed to be operating successfully if there is at least one path from the input node to the output node. Thus, a path is a minimal set which connects the input with the output. The failure of this structure means that there is at least one cut, that is, a cut is a minimal set of units such that their simultaneous failure leads to disconnecting the input and output nodes of the network. The system with a bridge structure has four different paths, {1, 4}, {1, 3, 5}, {2, 5}, and {2, 3, 4}, and four different cuts, {1, 2}, {1, 3, 5}, {2, 3, 4}, and {4, 5}. One can find that the subsets of units (links) forming the paths intersect, that is, they have some common units. The same is true for the cuts.

General Irreducible Network

Reliability analysis of networks in the most general case is very complicated. It is not possible to calculate exact reliability indexes of a general network analytically. One has to resort to approximate, numerical, or simulation (Monte Carlo) techniques. However, it is possible to find simple upper and lower limits for the PFFO of a two-pole network under the condition that any connection of the input and output nodes is admissible and constitutes an operating state of the network. Of course, this assumption is very restrictive for real systems because the existence of a path does not mean that the network is operating successfully. (A telecommunication network with a traffic load greater than the available capacity of the path is a simple counterexample.) We consider two main methods of boundary evaluation of the network PFFO (or availability coefficient).

BOUNDS ON TWO-POLE NETWORKS

In this section we consider some approximate analytical bounds on two-pole networks. Unfortunately, these bounds are generally not very tight, and it is not possible to infer which of these methods is better for a given network before actual computation. In spite of all these shortcomings, these techniques form a starting point for further analyses of general networks.

Esary–Proschan Bounds

The Esary–Proschan method compares a general system with a suitably constructed series system of cuts and parallel system of paths. The Esary–Proschan bounds can be computed only after finding all paths and cuts for a system. We illustrate this method for the simplest irreducible network with a bridge structure. All simple cuts and paths of the structure are enumerated in Fig. 9.

It is not possible to derive a precise formula for the system PFFO by using formulas for series–parallel and parallel–series connections because of dependent units (both in paths and cuts). From reliability theory (4,5), it is known that, for a series system of n dependent units, the PFFO satisfies the inequality

$$P_{\text{Series}} \geq \prod_{k=1}^{n} p_k$$

For a parallel system of m dependent units, the PFFO satisfies the inequality

$$P_{\text{Parallel}} \leq 1 - \prod_{k=1}^{m} q_k$$

where $q_k = 1 - p_k$. This immediately allows us to write the following for a system with a bridge structure:

$$\begin{aligned}(1 - q_1q_2)(1 - q_1q_3q_5)(1 - q_4q_5)(1 - q_2q_3q_4) &< P_{\text{Bridge}} \\ &< 1 - (1 - p_1p_4)(1 - p_1p_3p_5)(1 - p_2p_5)(1 - p_2p_3p_4)\end{aligned}$$

For the general case (details of which can be found in (4,5)), the Esary–Proschan bounds can be written in the form

$$\prod_{\forall k} B_k \leq P_{\text{Bridge}} \leq 1 - \prod_{\forall j} A_j$$

where

$$B_k = 1 - \prod_{i \in b_k} q_i$$

B_k is the set of units belonging to the kth minimum cut

$$A_j = 1 - \prod_{i \in a_j} p_i$$

and A_k is the set of units belonging to the jth minimum path in the network.

The inconvenience of this method is in the necessity of enumerating all cuts and paths of the network which is not a simple problem for networks of large dimension. Besides, the larger the network, the weaker the bounds generally turn out to be.

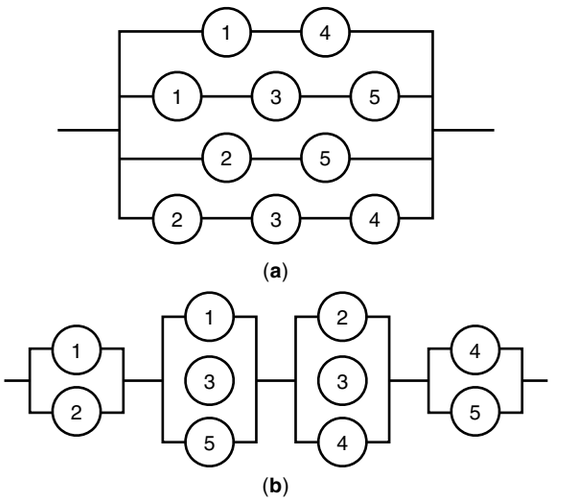

Figure 9. Decomposition of a bridge structure into paths and cuts (a) parallel configuration of paths; (b) series configuration of cuts.

Litvak–Ushakov Bounds

The Litvak–Ushakov method also compares the general network with a suitably constructed set of series and parallel structures. The main advantage of this method over the Esary–Proshan method is that it is possible to compute a weak bound with relative ease. In addition, this bound can be improved upon by finding more bounding structures.

We illustrate this method on a bridge structure. All simple cuts and paths of the structure were enumerated above (Fig. 9). The idea of the Litvak–Ushakov method is in presentating a network as a parallel connection of *nonintersected* (i.e., independent) paths or as a parallel connection of *nonintersected* cuts. The Litvak–Ushakov presentation of a network is not

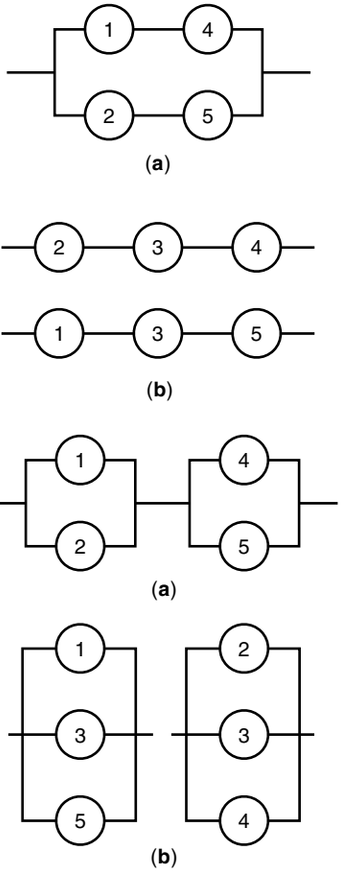

Figure 10. Decomposition of a bridge structure into nonintersected paths and cuts (a) paths; (b) cuts.

unique. For the bridge structure (Fig. 8), there are three possible presentations each for paths and cuts (Fig. 10).

Omitting all details we only mention that series connections of nonintersected cuts can be obtained by assuming that some units (links) are absolutely reliable, which means that the system as a whole has increased reliability. Analogously, parallel connection of nonintersected paths can be obtained by replacing some units by permanently failed ones, which decreases the system reliability. Following their argument, the bounds for the bridge structure have the following form:

$$\max\{1-(1-p_1p_4)(1-p_2p_5), p_2p_3p_4, p_1p_3p_5)\} < P_{\text{Bridge}} < 1-\min\{(1-q_1q_2)(1-q_4q_5), (1-q_2q_3q_4), (1-q_1q_3q_5)\}$$

We reiterate that not all terms (inside min or max) need to be evaluated to obtain a bound. In the general case (details for which can be found in (1)), Litvak–Ushakov bounds can be written in the following form:

$$\max\left\{\left(1-\prod_{j\in\alpha_1}A_j\right),\ldots,\left(1-\prod_{j\in\alpha_N}A_j\right)\right\} \le P_{\text{Bridge}} \le \min\left\{\prod_{k\in\beta_1}B_k,\ldots,\pi_{k\in\beta_M}B_k\right\}$$

where B_k and A_j have the same sense as above; $\alpha_1, \ldots, \alpha_N$ and $\beta_1, \ldots, \beta_M$ are different sets formed from nonintersected paths and cuts, respectively. One can find more details in (1,3).

DYNAMIC REDUNDANCY

A class of realizations of unit redundancy can be presented by an interesting scheme called *dynamic redundancy*. In this scheme, a redundant group consists of three subgroups, operating units, hot standby units, and cold standby units. Dynamic redundancy is applicable in situations where a failed unit cannot be repaired and failing stock (cold standby units) cannot be replenished. Thus the dynamic redundancy problem is closely related to the problem of inventory control, though it is different from the classical problem of inventory control. Because of this stringent definition, dynamic redundancy is not widely applicable. Ideas of dynamic redundancy may be applied to ensure the reliability of power supply equipment on an orbiting man-made satellite.

Assuming perfect switching, hot standby units are modeled as operating in parallel with main operating units. It is further assumed that cold standby units can be switched in only at some predefined moment. If there is a deficit of hot standby units before a cold switch is scheduled, the system fails. One may consider assigning all redundant units to the hot standby pool. In this case redundant units may be spent too soon, and they can not be replaced. On the other hand, if the number of hot standby units is smaller than some threshold, the probability of system failure before the switching of cold standby units increases significantly. Depending on the actual situation, it may even be beneficial not to replace all failed units at the time of maintenance after most of the spare units have been used up. Thus the problems of finding the optimal number of hot standby units and the number cold standby units to be switched in at prescribed moments arise.

The situation is further complicated by the existence of different goals that dynamic redundancy tries to maximize. A military satellite operation may not be ever interrupted even at the price of shortened life span. On the other hand, short interruptions can be tolerated by a planetary explorer if this leads to a significant increase in its total life. This is still an area of research (1).

BIBLIOGRAPHY

1. B. Gnedenko and I. Ushakov, in J. Falk, ed., *Probabilistic Reliability Engineering,* New York: Wiley, 1995.
2. B. Gnedenko, Yu. Belyayev, and A. Solovyev, *Mathematical Methods of Reliability Theory,* New York: Academic Press, 1969.
3. E. Elsayed, *Reliability Engineering,* Reading, MA: Addison-Wesley Longman, 1996.
4. I. Ushakov, ed., *Handbook of Reliability Engineering,* New York: Wiley, 1994.
5. R. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing: Probability Models,* New York: Holt, Rinehart and Winston, 1975.

IGOR USHAKOV
SUMANTRA CHAKRAVARTY
QUALCOMM Inc.

RELIABILITY, POWER DEVICES. See POWER DEVICE RELIABILITY.

RELIABILITY OF SOFTWARE. See AUTOMATIC TEST SOFTWARE.

RELIABILITY, SEMICONDUCTOR PACKAGING. See PACKAGING RELIABILITY, CHIP-SCALE SEMICONDUCTOR.

RELIABILITY, SOFTWARE. See SOFTWARE RELIABILITY; SOFTWARE VALIDATION FOR RELIABILITY.

RELIABILITY SPECIFICATIONS AND STANDARDS. See HANDBOOKS AND STANDARDS.