Cost Effective Spares Provisioning for the Deep Space Network

I. Eisenberger and F. Maiocco
Communications Systems Research Section
G. Lorden
California Institute of Technology

This report discusses a cost effective procedure for spares provisioning of the various components of an assembly that is assumed to fail if any one of the components fails. The procedure not only provides a means for obtaining a given operational availability at minimum cost, but it is also applicable when a constraint is placed upon the total cost of the spares for the components making up the assembly.

I. Introduction

An efficient method for spares provisioning of repairable equipment used in the Deep Space Network (DSN) is given in Ref. 1. For simplification we call any such piece of equipment a module. That method can be used to determine the minimum number of spares necessary to obtain a required operational availability which, in general, is defined as the stationary probability that a system is operating or operational at any given time and is sometimes referred to as the uptime ratio (*UTR*). The method was applied to two types of situations. In the first case, the goal was to provide a common pool of spares for n identical, independently operated modules, $n = 1, 2, 3, \cdots$ situated within a single complex or DSS, and, in the second, to provide a spares complement for a so-called (m, n) system. The requirement for this system is that at least m out of n identical modules must be in operation at any given time in order for the system to adequately perform its required function. It should be noted that a $(1, n)$ system is equivalent to a simple parallel configuration with n components.

For these cases the problem of meeting a cost constraint is not difficult to solve. If it is determined that N spares are needed to achieve a required UTR and enough money is available to buy M spares then the number of spares that should be bought is the minimum of N and M. In practice, however, one often has a system configuration containing several types of modules, with the constraint placed on the *total* cost of spares for the entire system. The problem in this case is to choose the number of spares for each type of module in such a way that the system operational availability is maximized within the cost constraint. An equivalent problem is that of achieving a prescribed system operational availability for minimum cost. The method developed in the present report is easily applied to both problems.

II. Statement of the Problem

Let k be the number of types of modules in the system. For each type of module there is a failure rate λ_i, and a repair rate, μ_i, and it is assumed that repair times and

times between failures are exponentially distributed. (As explained in Ref. 1, constant repair times yield the same results to a reasonable degree of approximation.) For $i = 1, \cdots, k$, let n_i denote the number of modules of type i in the system and m_i the number that have to be unfailed in order for the system to perform its intended function satisfactorily. It is assumed that whenever fewer than the required minimum m_i are operable for one or more of the module types, the system is down. A further assumption is that failures and repairs of different modules are stochastically independent. Under these assumptions, the system *UTR* is the product of the k *UTR*s for the module types. The latter can be calculated by the algorithm in Ref. 1 that allows for so-called (m, n) systems. Of course, a single module constitutes a $(1, 1)$ system while n identical modules form an (n, n) system if in series and a $(1, n)$ system if in parallel.

We define a spares *package* to be a choice of $s_1, \cdots, s_k$, the number of spares for the respective types of modules. If $U_i(s)$ is the *UTR* for the type i module when s spares are provided, then we have

$$\text{System } UTR = U = \prod_{i=1}^{k} U_i(s_i) \tag{1}$$

The value of $U_i(s)$ is computed by the procedure given in Ref. 1. Let c_i denote the cost for each module of type i. Then

$$\text{Cost of Package} = C = \sum_{i=1}^{k} s_i c_i \tag{2}$$

A spares package with uptime ratio U_1 and Cost C_1 is preferable to one with U_2 and C_2 if $U_1 \geqq U_2$, $C_1 \leqq C_2$, and at least one of these inequalities is strict. A spares package will be called *efficient* if no other package is preferable to it in the sense just defined. Clearly it is only the *efficient* packages one would want to use, since any other package can be improved upon by reducing cost or improving *UTR* or both. The method presented in Section III constructs spares packages that are efficient under a mild condition on the $U_i(s)$'s. Note that if U and C are the *UTR* and cost of an efficient package, then U is the maximum possible *UTR* subject to the cost constraint C, *and* C is the minimum possible cost of achieving a $UTR \geqq U$.

III. A Method for Constructing Efficient Spares Packages

Define the efficiency of the s^{th} spare for module type i as

$$R_i(s) = \frac{\log\left(U_i(s)/U_i(s-1)\right)}{c_i}$$

for $i = 1, \cdots, k$ and $s = 1, 2, \cdots$. Then we can rewrite Eq. (1) in the form

$$\log U = \sum_{i=1}^{k} \log U_i(0) + \sum_{i=1}^{k} \sum_{s=1}^{s_i} c_i R_i(s) \tag{3}$$

Note that the first summation is constant, i.e., does not depend on the choice of the spares package $s_1, \cdots, s_k$. The total cost of the spares package can be written as

$$C = \sum_{i=1}^{k} \sum_{s=1}^{s_i} c_i \tag{4}$$

Then Eq. (3) becomes

$$\frac{1}{C} \cdot \log U = \text{constant} + \sum_{i=1}^{k} \sum_{s=1}^{s_i} \frac{c_i}{C} R_i(s) \tag{5}$$

Since the sum of all c_i's is C, the sum of the (c_i/C)'s is 1, so that Eq. (5) expresses $1/C \log U$ as a constant plus a weighted average of the $R_i(s)$'s. Since we want to maximize $\log U$, it is plausible that we should proceed inductively as follows:

(1) Choose the first spare for the type i such that $R_i(1) > R_j(1)$ for all $j \neq i$.

(2) Continue by choosing the next spare to be the one yielding the largest efficiency among those spares immediately available. Stop at any time.

To clarify step (2), suppose that $r_1, \cdots, r_k$ spares of types 1 through k have already been chosen. Then if we choose type i for the next spare, the efficiency of that choice is $R_i(r_i + 1)$. Step (2) calls for choosing i to get the *largest* of $R_i(r_i + 1), \cdots, R_k(r_k + 1)$. Then r_i is increased by one (since we now have one more spare) and step (2) is repeated.

No matter when this process is stopped, the spares package is efficient, provided the condition in statement (6a) below is satisfied. In other words, the process produces a sequence of efficient spares packages, each adding one more spare to the previous package. Naturally, both cost and system *UTR* increase as more spares are added.

A sufficient condition to insure that the method just described produces efficient spares packages is the following "monotonicity condition."

For each $i = 1, \cdots, k$, $R_i(s)$ is decreasing in s (6a)

From the definition of $R_i(s)$ it is clear that statement (6a) reduces to

For each $i = 1, \cdots, k$, $U_i(s)/U_i(s-1)$ is decreasing in s (6b)

Using the algorithm developed in Ref. 1, condition (6b) was checked for several thousand cases with λ's and μ's varying over a broad range (it is sufficient to vary the ratio λ/μ) and s from 1 up to the value required to make $U_i(s) > 0.9999$. The cases tested included sparing for a single module as well as sparing for various (m, n) systems. Not a single exception to condition (6b) was found. Finding a mathematical proof of the condition seems to be very difficult because of the complexity of the recursive equation defining the $U_i(s)$'s.

A proof that the method produces efficient packages when the monotonicity conditions are met is given in Section IV.

To illustrate the use of this method, consider the following example. We have a system that consists of 24 modules, one module each of 12 types and 3 modules each of four types. For each of the latter four types, the modules are arranged in a (2, 3) configuration. We begin by determining $U_i(0)$ and $U_i(1)$ for each type, that is, we calculate by the method of Ref. 1 the *UTR* for each type when no spare is provided and when 1 spare is provided. From these calculations we determine $R_i(1)$ for $i = 1, \cdots, 16$. Now, if $R_j(1)$ is the greatest of these the first spares package consists of one spare of type j and no spares for each of the remaining types. To determine the next spares package (containing two spares) we need only calculate $U_j(2)$ and from this $R_j(2)$. We now look at the new set of $R_i(s_i)$'s where, in this step, $s_i = 0$ for $i \neq j$ and $s_j = 1$. If $R_k(s_k)$ is the greatest of these, the second spares package consists of one spare for type j, one spare for type k, and no spares for the remaining types. We continue in this fashion until a desired stopping point is reached. Each successive spares package generated in this way will contain one spare more than the previous one and will be efficient in the sense defined above. Since the cost of each spares package and the system *UTR* achieved are easily computed as the process proceeds, a cost vs *UTR* tradeoff can be made to determine when to stop, or one can continue until a specified *UTR* is achieved or a cost constraint would be exceeded by continuing.

The procedure was applied to the above described system for specific sets of λ's and unit costs. The repair time was assumed to be 336 hours for all of the types of modules. The upper graph in Fig. 1 labeled $\epsilon = 0$ is a plot from step 13 on of the *UTR*s achieved by the above procedure as a function of the costs incurred. Table 1 lists some of the numeral results. The entries in column 2 show the successive decisions on which type of spare should be added to the previous spares package in order to get the current one. In other words, if k is the entry in the j^{th} row of column 2, add a spare of type k module to the $(j-1)^{st}$ spares package to obtain the j^{th} spares package (containing j spares). The row denoted by "Final spares package" shows the contents of the last spares package (in this case the 37th). The k^{th} entry in this row denotes the number of spares for the k^{th} type of module (the total number of spares is 37). The entries in this row and those in column 2 enable one to determine the contents of any of the previous spares packages. For example, to obtain the 34th spares package, subtract from the entries in the final spares package one spare each for the fourth, second, and tenth type of module, obtaining the spares package 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3 with a cost of 1046 and a system *UTR* of 0.9936.

Each entry in column 4 of Table 1 is the inverse of the down time ratio (*DTR*), ($DTR = 1 - UTR$) at each step. These numbers are given because they have an interesting property. Figure 2 is a plot on log paper of $1/DTR$ versus cost starting with step 11, which shows a cost of 348 and $UTR = 0.8647$. In this range, it can be seen that the function is closely approximated by the straight line drawn on the graph by eye. This means that log $(1/DTR)$/cost is very nearly constant in the range of practical interest to a potential user of the procedure. If we denote this constant by b, we have

$$\log(1/DTR)/C = b$$

resulting in

$$e^{-bc} = DTR = 1 - UTR$$

so that a knowledge of b enables us to predict very closely what the *UTR* will be for a given cost using this method of sparing. Column 5 in the table gives the values of log $(1/DTR)/C$ for each step and confirm the results shown in Fig. 2. If we think of $1/DTR$ as the average number of random inspections (spread out over time) required to first find the system down, then we have the rule-of-thumb: this average number of inspections grows exponentially as a function of the cost of spares. The exponential growth rate parameter, b, is characteristic of the system configuration, the failure rates, and the repair rates.

IV. The Value/Cost Lemma

By taking logarithms in Eq. (1) it is easily seen that the problem of selecting spares efficiently fits the following general formulation, which is applicable in many contexts.

Let S be a set (e.g., all possible spares) whose members x each have a value $v(x)$ and a cost $c(x)$. It is desired to select subsets R of S whose total value

$$V(R) = \sum_{x \in R} v(x)$$

and cost

$$C(R) = \sum_{x \in R} c(x)$$

are *efficient* in the sense that neither can be improved without hurting the other. It is easy to see that a sufficient condition for R to be efficient is that there exists a $d > 0$ such that $V(R) - dC(R) \geqq V(R') - dC(R')$ for all $R' \subset S$. (If, for example, there were an R' such that $V(R') > V(R)$ while $C(R') \leqq C(R)$, then $V(R) - dC(R) < V(R') - dC(R')$, contrary to the condition.)

To find such an R for a given $d > 0$, note that

$$V(R) - dC(R) = \sum_{x \in R} [v(x) - dc(x)]$$

$$\leqq \sum_{x : v(x) - dc(x) \geqq 0} [v(x) - dc(x)]$$

and observe the upper bound on the right is attained if $R = R_d = \{x \mid v(x) \geqq dc(x)\}$. Thus, for all $d > 0$, R_d satisfies the sufficient condition and we have the *Value/Cost Lemma:* R_d is efficient.

A similar result describing efficient performance of statistical hypothesis tests is called the Neyman-Pearson Lemma. The same mathematics has been used in problems close to the present one (Ref. 2) and is related to the more general theory of nonlinear programming (Ref. 3).

To apply this result to the sparing problem, identify the members of S as x_{ij}'s where

$x_{ij} = j^{\text{th}}$ spare of type i; $\quad i = 1, \cdots, k; \quad j = 1, 2, \cdots$

In the notation of the previous section, if $s_i = 3$, say, then we understand that x_{i1}, x_{i2}, and x_{i3} have been selected, but not x_{i4}, x_{i5}, etc. We define

$$c(x_{ij}) = c_i = \text{unit cost for spares of type } i$$

$$v(i, j) = \log U_i(j) - \log U_i(j-1)$$

Thus if $s_1, \cdots, s_k$ spares of the various types are selected, the total value is

$$V = \sum_{i=1}^{k} \sum_{j=1}^{s_i} [\log U_i(j) - \log U_i(j-1)]$$

The sum over j telescopes and we get

$$V = \sum_{i=1}^{k} [\log U_i(s_i) - \log U_i(0)]$$

$$= \log \prod_{i=1}^{k} U_i(s_i) - \log \prod_{i=1}^{k} U_i(0)$$

Thus V is the log of system *UTR* minus a constant not depending on the s_i's, so that the comparison between system *UTR*s for different packages is equivalent to a comparison of their values, $V = \sum v(x_{ij})$. Therefore a spares package that selects those spares x_{ij} in R_d (for a $d > 0$) is efficient in the sense of Section II.

To see that the method of Section III produces R_d's under the monotonicity condition (6a), note that wherever we stop, the efficiencies $R_i(s)$ of the spares selected are all greater than or equal to the efficiencies of the spares not selected. Hence, there is some $d > 0$ that is a lower bound on the efficiency of selected spares and a strict upper bound on the efficiency of spares not selected. Since

$$R_i(s) = \frac{v(x_{is})}{c_i}$$

$R_i(s)$ is $\geqq$ or $< d$ according as $v(x_{is}) - dc_i$ is $\geqq$ or < 0; hence the members of R_d are precisely the spares selected by the method. Actually, for a spares package of this kind costing C, say, to be efficient, it is sufficient that for every i the monotonicity condition holds for values of $s \leqq C/c_i$, since this is as many spares of type i as can be bought for C or less.

V. The Cost of Misestimating Failure Rates

Since the results obtained from our sparing procedure depend upon the estimated failure rates of the various types of modules, a decrease in efficiency will result when the procedure is used with a set of estimated $\bar{\lambda}$'s that differ substantially from the true λ's.

In order to see the effect of this kind of error, we applied the procedure to the system described in Section III under

the assumption that the true λ's were as given but that the estimated $\bar{\lambda}_j$'s were given by

$$\lambda_j = \begin{cases} (1+\epsilon)\lambda_j, & j = 1,3,5,\cdots,15 \\ \lambda_j/(1+\epsilon), & j = 2,4,6,\cdots,16 \end{cases}$$

for $\epsilon = 0.5$ and $\epsilon = 1.0$.

Thus the contents of each spares package was determined by using the $\bar{\lambda}_j$'s while the *UTR* was computed using the true λ's. The graphs in Fig. 1 labeled $\epsilon = 0.5$ and $\epsilon = 1.0$ are plots of *UTR* vs cost for the two examples of misestimated failure rates.

It is readily seen that for a given cost one does not achieve as high a *UTR* when the estimated failure rates used to generate the spares package differ from the true ones.

Looking at it the other way around, the cost of achieving a given *UTR* is increased when the failure rates are in error. In the example depicted in Fig. 1, the percentage increase in the cost for fixed *UTR*'s is about 10% when $\epsilon = 0.5$ and about 20 to 25% when $\epsilon = 1$. Note that in the latter case *all* the failure rates are in error by a factor of two, half of them too high and half too low. If failure rates are in error by random factors less than or equal to two, the degradation in efficiency will be less.

VI. Applications

The principal applications envisaged for the method developed in this report are to the problem of initial spares provisioning and to improving the operational availability of present subsystems that are relatively unreliable. In the former case, only crude estimates of failure rates are available, but the analysis in Section V indicates that the method still yields useful results.

For improving operational availability of existing systems or subsystems, more accurate judgments are possible by utilizing reliability histories to estimate failure rates. Once these rates are determined, the efficiencies of additional spares of all types can be calculated and any improved level of *UTR* can be reached efficiently by purchasing additional spares with the greatest efficiencies.

References

1. Eisenberger, I., Lorden, G., and Maiocco, F., "A Preliminary Study of Spares Provisioning for the Deep Space Network," in *The Deep Space Network Progress Report*, TR 32-1526, Vol. XVIII, pp. 102–110. Jet Propulsion Laboratory, Pasadena, Calif., Dec. 15, 1973.
2. Proschan, F., "Optimal System Supply," *Naval Reserve Logistics Quarterly*, pp. 609–646. Office of Naval Research, Washington, D.C., 1960.
3. Kuhn, H. W., and Tucker, A. W., "Non-linear Programming," Proceedings of 2nd Berkeley Symposium, pp. 481–492. University of California Press, Berkeley, Calif., 1951.

Table 1. Results obtained by efficient sparing

Step	Type of spare	Cost C	$1/DTR$	$\log(1/DTR)/C$	UTR
1	14	18	1.64	0.02757	0.3911
2	10	40	1.76	0.01406	0.4301
3	15	86	1.97	0.00789	0.4925
4	16	102	2.07	0.00711	0.5158
5	13	146	2.39	0.00596	0.5814
6	11	188	2.81	0.00550	0.6447
7	6	214	3.17	0.00538	0.6840
8	9	254	3.93	0.00539	0.7456
9	7	292	4.96	0.00548	0.7984
10	12	312	5.64	0.00554	0.8226
11	5	348	7.39	0.00575	0.8647
12	3	382	9.22	0.00581	0.8915
13	2	412	11.11	0.00584	0.9100
14	8	436	13.18	0.00591	0.9241
15	14	454	14.79	0.00593	0.9324
16	16	470	16.53	0.00597	0.9395
17	1	502	19.73	0.00594	0.9493
18	15	548	24.36	0.00583	0.9590
19	10	570	27.45	0.00581	0.9636
20	12	590	30.47	0.00579	0.9672
21	13	634	39.60	0.00580	0.9748
22	4	662	47.94	0.00585	0.9791
23	11	704	65.66	0.00594	0.9848
24	9	744	88.18	0.00602	0.9887
25	6	770	104.90	0.00604	0.9905
26	7	808	140.80	0.00612	0.9929
27	16	824	163.04	0.00618	0.9939
28	8	848	206.45	0.00629	0.9952
29	5	884	280.63	0.00638	0.9964
30	14	902	320.04	0.00640	0.9969
31	12	922	361.50	0.00639	0.9972
32	3	956	436.37	0.00636	0.9977
33	15	1002	568.77	0.00633	0.9982
34	13	1046	713.52	0.00628	0.9936
35	10	1068	811.32	0.00627	0.9988
36	2	1098	981.96	0.00627	0.9990
37	4	1126	1206.67	0.00630	0.9992

Parameters		
Type	$\lambda/10^{-6}$	Cost/unit
1	31.25	32
2	62.50	30
3	93.75	34
4	125.00	28
5	156.25	36
6	187.50	26
7	218.75	38
8	250.00	24
9	281.25	40
10	312.50	22
11	343.75	42
12	375.00	20
13	406.25	44
14	437.50	18
15	460.75	46
16	500.00	16

Final spares package:
1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3, 3, 3, 3, 3

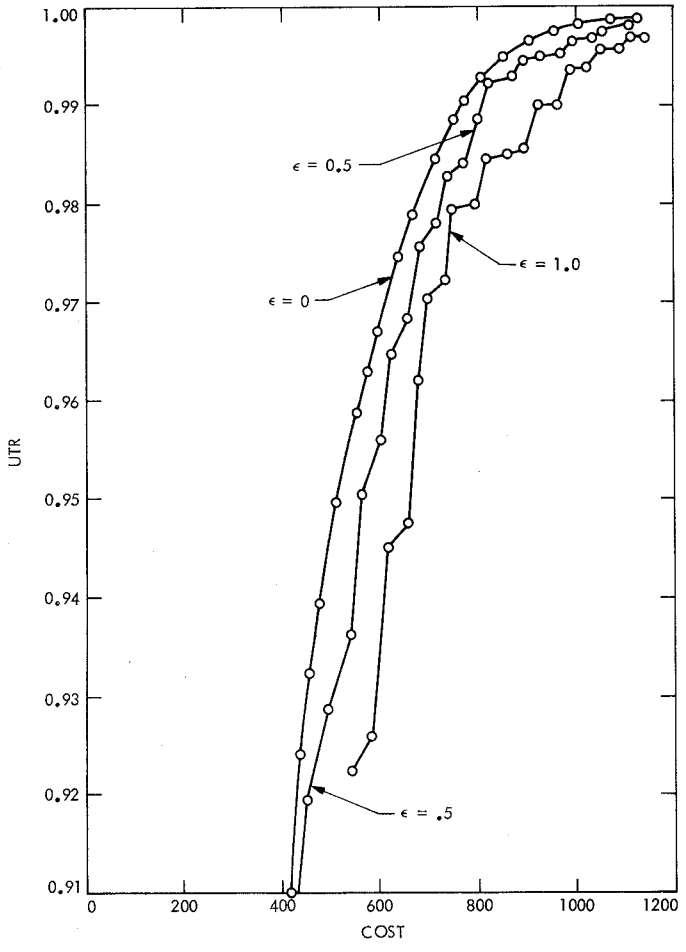

Fig. 1. *UTR* vs cost for efficient sparing with $\epsilon = 0, 0.5, 1.0$

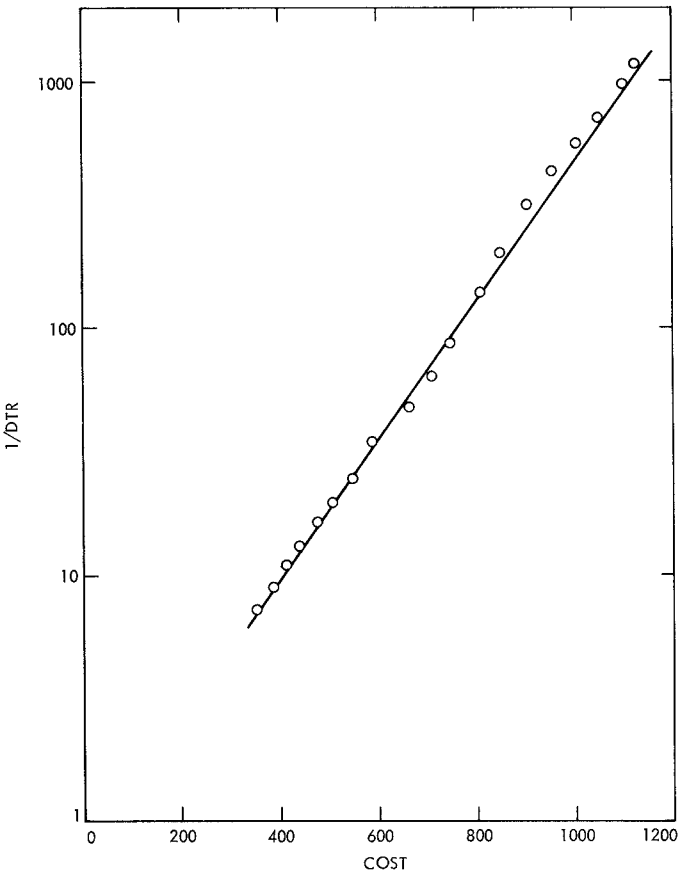

Fig. 2. 1/*DTR* vs cost for efficient sparing