OPTIMUM ALLOCATION OF REDUNDANCY AMONG SUBSYSTEMS CONNECTED IN SERIES

by Darl D. Bien

Lewis Research Center
Cleveland, Ohio 44135
### Abstract

This analysis considers the optimum allocation of redundancy in a system of serially connected subsystems in which each subsystem is of the k-out-of-n type. Redundancy is optimally allocated when (1) reliability is maximized for given costs or (2) costs are minimized for given reliability. Several techniques are presented for achieving optimum allocation and their relative merits are discussed. Approximate solutions in closed form were attainable only for the special case of series-parallel systems and the efficacy of these approximations is discussed.

### Key Words (Suggested by Author(s))

- Redundancy
- Redundancy allocation
- Reliability
- Optimum redundancy
- Series-parallel system
OPTIMUM ALLOCATION OF REDUNDANCY AMONG SUBSYSTEMS
CONNECTED IN SERIES*

by Darl D. Bien

Lewis Research Center**

SUMMARY

This analysis considers the optimum allocation of redundancy in a system of serially connected k-out-of-n subsystems. The two problems treated are (1) maximization of system reliability subject to multiple cost constraints and (2) minimization of some function of multiple costs subject to maintenance of a minimum acceptable level of system reliability. These are problems for which there are no general solutions. It is shown herein that several techniques applied previously to analysis of the special case of series-parallel system optimization can, with certain adjustments and under certain conditions, be used to optimize redundancy in the series-k-out-of-n system. The techniques used to solve these problems are (1) dynamic programming, (2) generalized Lagrange multipliers, (3) sensitivity analysis, (4) multidimensional knapsack formulation with a branch-and-bound procedure, and (5) a synthesis of concave and integer programming. None of the methods is new but the applications to the series-k-out-of-n problems are unique. Some of the methods produce only approximate solutions, while others produce exact solutions with increased effort. Some are particularly suited to the determination of optimum redundancy for a single specification; others, because of their sequential natures, are more suitable for determining a range or family of optimal solutions. The procedures are discussed and compared; an example problem is solved by using three of the techniques. The special case of redundancy optimization under a single constraint for the series-parallel system is treated with the derivation by Lagrange multipliers of approximate equations. These equations are shown to be useful in rapidly determining the approximate optimal allocation of redundancy in the series-parallel system. An example problem is solved by using these equations, and the results are compared with optimal solutions produced by a method of exact optimization.

---

*The material in this report was submitted as a thesis in partial fulfillment of the requirements for the degree Doctor of Philosophy in Statistics at Case Western Reserve University, Cleveland, Ohio, in September 1970.

**The author is now at the University of Denver.
INTRODUCTION

There is a basic conflict in utilizing redundancy in a system. The addition of redundant components increases the costs - measured in weight, volume, area, etc. - at the same time that it increases reliability. This conflict cannot be circumvented, but it can be minimized through optimum design.

The conflict between quality and the outlay of resources is everywhere present. It is prominent, for example, in the design of complex electronic equipment for military or space use. Often there are constraints on some of the resources. In the case of space systems, the payload weight is limited by the capability of the launch vehicle. There are also often minimum acceptable reliability requirements, as in the case of manned space flight.

The simplest redundancy arrangement is one where \( n \) components are connected in parallel and where the required function is performed as long as at least one of the components functions properly. If all components are identical and equally susceptible to failure during the entire mission, the arrangement is called parallel redundancy. It is for this case that most of the redundancy optimization procedures have been developed.

A more general redundancy arrangement is one which requires at least \( k \) of \( n \) components to function properly. This is called \( k \)-out-of-\( n \) redundancy (ref. 1). As before, all components are identical and equally susceptible to failure during the entire mission. The \( n \) identical components have a single function to perform and, taken together, they are called a subsystem. A system generally requires the performance of several functions. This is accomplished by serially connecting a number of subsystems. Hence, the system under consideration is called a series-\( k \)-out-of-\( n \) system.

In practice many subsystems are in modular form and the work load of the subsystem is distributed uniformly over \( k \) components. This is the case, for example, in a waste-heat rejection subsystem constructed of \( n \) identical modules, any \( k \) of which can reject the required amount of heat. Subsystem success is achieved by \( k \) or more successfully operating modules; failure occurs as soon as \( n - k + 1 \) modules have failed. It is noted that parallel redundancy is the special case where \( k = 1 \); a purely series subsystem is the special case where \( k = n \).

The problem of optimally allocating redundancy in the series-\( k \)-out-of-\( n \) system is treated herein. Study of this problem was motivated because (1) \( k \)-out-of-\( n \) subsystems are important by themselves and (2) \( k \)-out-of-\( n \) subsystems include as special cases the parallel and series subsystems.

The object of this analysis is to formulate methods for optimally allocating redundancy in series-\( k \)-out-of-\( n \) systems. The techniques are not new; they have been applied previously to a special case, namely, the case of series-parallel systems - systems composed of serially connected subsystems where only one component in each subsystem is required to survive.
The conditions under which these techniques can be used for the more general series-k-out-of-n system are developed. The methods are compared for effectiveness and utility.

Finally, the special series-parallel system is treated. Closed-form equations are available in the literature, and they are compared with equations derived herein by the method of Lagrange multipliers.

PROBLEMS TO BE TREATED

General Redundancy Allocation Problems

The technique of using redundancy in an optimal way to increase the reliability of a system is the subject of this analysis. Redundancy involves the inclusion of more components than necessary to perform a given function. These additional components allow for some component failures without causing the system to fail.

Reliability is the probability that a component, subsystem, or system will operate satisfactorily for a specified period of time. For reliability analysis, any complex system may be represented as m serially connected subsystems.

The reliability of the i\textsuperscript{th} subsystem is \( r_i \). For m subsystems connected in series, the product rule gives the system reliability as

\[
R = \prod_{i=1}^{m} r_i \tag{1}
\]

and thus the system reliability is less than the reliability of the least reliable subsystem. Associated with each component of subsystem i, there are s different costs, such as weight, area, volume, money, etc. Since there are \( n_i \) components in the i\textsuperscript{th} subsystem, the total system cost of the j\textsuperscript{th} type is

\[
C \cdot j = \sum_{i=1}^{m} c_{ij} n_i \quad j = 1, \ldots, s; \ n_i \ \text{integer} \tag{2}
\]

where \( c_{ij} \) is the j\textsuperscript{th} type cost of a single component of the i\textsuperscript{th} subsystem and is assumed to be positive and where the subscript denotes summation over \( i = 1, \ldots, m \).

A problem of interest is to maximize the system reliability \( R \) without exceeding any of s cost constraints. If \( C \cdot j, \text{max} \) is the allowable use of resource \( j \), this prob-
lem can be stated symbolically as follows:

Problem I: Maximize the total system reliability

$$R = \prod_{i=1}^{m} r_i$$

subject to

$$\sum_{i=1}^{m} c_{ij} n_i \leq C_{j, \text{max}} \quad j = 1, \ldots, s; \quad n_i \text{ integer}$$

A related problem is to minimize some function of the cost factors $C_{j}$ ($j = 1, \ldots, s$) without allowing the system reliability to drop below an acceptable level. If $R_{\text{min}}$ is the minimum allowable system reliability, the problem can be stated symbolically as follows:

Problem II: Minimize some function of the $s$ cost factors

$$C_{j} = \sum_{i=1}^{m} c_{ij} n_i \quad j = 1, \ldots, s; \quad n_i \text{ integer}$$

subject to

$$\prod_{i=1}^{m} r_i \geq R_{\text{min}}$$

Often the designer is not simply interested in the one solution which satisfies the constraints of problems I and II. Rather he may be interested in relaxing the constraints and generating a family of optimum solutions for a range of constraint values. In problem I terminology, several values of $C_{j, \text{max}}$ ($j = 1, \ldots, s$) are to be selected and, for each, a solution having maximum $R$ is of interest. In problem II terminology, several values of $R_{\text{min}}$ are to be selected and, for each, some minimum cost function is to be generated. Problems I and II describe the same points in $RC_{j}$ space. That is, the redundancy allocation which yields a maximum $R$ in problem I equal to $R_{\text{min}}$ of prob-
lem II is the same allocation which produces a minimum $C_{j}$ in problem II equal to $C_{j, \text{max}}$ of problem I. Therefore, problems I and II can be treated simultaneously by generating a family of undominated allocations. Undominated allocations are defined in the next main section.

**Subsystems of k-out-of-n Type**

The series-k-out-of-n system is of particular interest because of its general form and also because it has several direct applications. A vapor chamber radiator for space application is one of the subsystems of the spacecraft which must operate successfully in order to have mission success. It consists of several hundred (n) independent chambers of which some number (k) must function in order to reject the required waste heat from a powerplant (ref. 2).

Another example is a panel of solar cells used to supply electricity. Such a panel generally consists of hundreds or thousands (n) of independent cells, each capable of supplying some small quantity of electrical power. As long as some number (k) of these cells operate successfully, the solar cell subsystem supplies the required power. But, when $n - k + 1$ of the cells have failed, the electrical subsystem has failed and the system fails.

This analysis deals only with active redundancy; that is, all n of the components are in the "on" position and vulnerable to failure during the entire mission. Such redundancy is quite common in complex remote systems, such as space applications, where all components are subjected to a hazardous environment. The standby redundancy case, where redundant components are switched on only when the active components fail, is not treated herein. It is more complex in that it requires additional sensing and switching devices and these devices affect the system reliability.

A further justification for active redundancy is found in much of the present design philosophy for complex systems. Redundancy is usually costly; because of this, many systems are designed to begin a mission by utilizing all available system capability. As failures occur, the performance is degraded, but vital functions of the system can still be performed. System failure occurs when the performance of the system falls below acceptable levels. In the case of space systems, this acceptable system performance might logically be survival of the crew in manned missions or performance of key experiments in unmanned missions. In such cases, system reliability is the probability of performing some basic vital functions for the duration of the mission.

It is assumed that the reliability of any component is independent of the state of the other components. All components within a given subsystem are identical. Since the concern here is with end-of-mission success and it is presumed that each designer specifies his required mission time, the individual component success probability is
treated as an assigned constant \( p_i \). That is, time dependency would likely play a part in the determination of \( p_i \), but for the purposes of this analysis \( p_i \) is just another specified constant. If the component failure probability is \( q_i = 1 - p_i \), the subsystem reliability is given by the binomial summation as

\[
 r_i = \sum_{x=k_i}^{n_i} \binom{n_i}{x} p_i^x q_i^{n_i-x} \quad i = 1, \ldots, m; \; k_i, n_i \text{ integer} \tag{7}
\]

Another form of this equation, which is sometimes less cumbersome for computational purposes, is the negative binomial

\[
 r_i = p_i \sum_{x=0}^{n_i-k_i} \binom{k_i - 1 + x}{x} q_i^x \quad i = 1, \ldots, m; \; k_i, n_i \text{ integer} \tag{8}
\]

A typical series-k-out-of-n system is shown in figure 1.

In principle, the optimum allocation of redundancy can be achieved by examining all possible combinations of components and choosing the ones which satisfy the require-

![Figure 1. - Typical series-k-out-of-n system. Shading indicates redundant component.](image-url)
ments of reliability or costs. This, however, involves an inordinate expenditure of
effort. Several methods have been used to obtain solutions for the series-parallel sys-
tem without searching all combinations. Five of these methods have been adapted to the
problem of series-k-out-of-n systems, and these solutions are presented in the next
main section.

The first method is that devised by Kettelle (ref. 3) and was modified by Proschans
and Bray (ref. 4) to include multiple cost factors. Their work, of course, is for the
series-parallel system. Kettelle's modified method is used to generate the complete
family of solutions over a range of reliability and costs; but since it is a dynamic pro-
gramming technique, it is plagued by the problem of dimensionality. The second method
uses Lagrange multipliers as applied to nondifferentiable functions (ref. 5) to generate
the "best" solutions with much less effort than required by Kettelle's method. The
third method involves selecting consecutively for redundancy the subsystem which con-
tributes the greatest reliability per unit of weighted costs (ref. 1). By generating solu-
tions for a grid of weighting factors, this method gives the same solutions as the
Lagrange multiplier method. These last two methods are subject to some restrictions
on the subsystem reliability equation, which are discussed herein. An example prob-
lem is worked using all three techniques and comparisons are made.

Two other methods for treating problem I, the maximization of reliability subject
to several cost constraints, are presented. Ghare and Taylor (ref. 6) use a multi-
dimensional knapsack formulation of the series-parallel system optimization problem.
Using a branch-and-bound procedure, they obtain the exact solution to problem I. It is
shown in the following main section of this report that their procedure is also valid for
the series-k-out-of-n system. The final method of solving this problem is that of
Mizukami (ref. 7). His method for maximizing concave functions is referred to as the
method of concave and integer programming. The reliability function which is to be
maximized is made approximately piecewise linear in reference 7. Methods of linear
programming can thus be used to obtain approximate solutions to this problem. Some
discussion is included herein concerning the advantages and disadvantages of all five
techniques.

Subsystems of 1-out-of-n Type

The series-parallel system is of interest because many systems require the per-
formance of m functions where each function can be performed by only one component.
A subsystem then consists of n components, only one of which needs to survive for sub-
system success.

The section SERIES-PARALLEL SYSTEM presents the derivation of two
closed-form equations for the approximate optimum distribution of redundancy under the models of problems I and II when there is only a single type of cost. The derivation of both equations is by the method of Lagrange multipliers.

Only one of the equations is new, however. The equation derived herein for solving problem I is the same as that derived by Mine (ref. 8) using calculus of variations and by Federowicz and Mazumdar (ref. 9) using geometric programming. The equation derived herein for solving problem II differs slightly from those derived by Moskowitz and McLean (ref. 10) using a variational technique and by Kulakov and Zagoruyko (ref. 11) using sensitivity balancing. This new equation is to be compared with those cited from the literature.

The reason for inclusion of the section on the series-parallel system is, for the first time, to present solutions for both problems I and II in closed form and derived by the same method. A comparison is made of the three solution equations to problem II.

It is assumed, as with the k-out-of-n subsystem, that the reliability of any component is independent of the state of the other components. Each component within a given subsystem has success probability \( p_i \) and a single type of cost \( c_i \). If the component failure probability is \( q_i = 1 - p_i \), the subsystem reliability is given by

\[
r_i = 1 - q_i^n_i \quad i = 1, \ldots, m
\]

Since only a single type of cost is used in the analysis of series-parallel systems, the total cost in equation (2) is just

\[
C = \sum_{i=1}^{n} c_i n_i
\]

Likewise, the models of problems I and II must be modified in that the constraint of equation (4) becomes

\[
\sum_{i=1}^{m} c_i n_i \leq C_{\text{max}}\quad n_i \text{ integer}
\]

and the objective function of equation (5) becomes equation (10).

A typical series-parallel system is shown in figure 2.

The problem, as before, is to generate the optimum solutions to the redundancy allocation problem. Explicit equations are derived for the values of \( n_i \) (\( i = 1, \ldots, m \)
resulting in (1) the greatest reliability for a given cost constraint and (2) the least cost for a given reliability goal. An example problem is solved by use of these equations and comparisons are made with the exact integer solutions.

**REVIEW OF THE LITERATURE**

In an analysis of redundancy optimization, one is aided by varying disciplines, from reliability to optimization procedures. Although not all directly related to the problem of redundancy optimization, the references cited herein were of value in offering insights into the problem.

A comprehensive discussion of redundancy optimization is found in Barlow and Proschan (ref. 1). Chapter 6 of that reference is devoted to both active and standby 1-out-of-n redundancy subject to single and multiple constraints. Kettelle (ref. 3) provides a dynamic programming algorithm for obtaining the complete family of optimum integer solutions for the parallel redundancy case. He also derives an equation for maximum reliability as a function of cost, but this equation gives no indication of the magnitude of \( n_i \) (\( i = 1, \ldots, m \)). Proschan and Bray (ref. 4) extend Kettelle's algorithm to include multiple constraints. Dynamic programming is the subject of a book by Bellman (ref. 12).

The parallel redundancy allocation problem is solved graphically by DiToro (ref. 13), Herron (ref. 14), van Hees and Meerendonk (ref. 15), Thakkar and Hughes (ref. 16), and Korman (ref. 17). Korman also discusses the standby redundancy problem.

Everett (ref. 5) discusses the method of Lagrange multipliers as applied to optimization of nondifferentiable functions. He uses this to get some solutions for the single-constraint parallel redundancy case. Fox and Landi (ref. 18) and Zahl (ref. 19) present
a general study of constrained optimization problems.

Kolesar (ref. 20) uses integer linear programming to maximize reliability subject to constraints for three parallel redundancy problems: (1) single type of failure, (2) two types of failure, and (3) failure to operate or premature operation. A good book on integer programming is that by Saaty (ref. 21).

Alekseyev and Iakushev (ref. 22) and Alekseyev (ref. 23) use dynamic programming for the multiple-constraint parallel redundancy problem. Several restrictions on the problem result in a much smaller search area than by previous methods. Under these restrictions this method is efficient; otherwise, it is plagued by dimensionality. Bellman and Dreyfus (ref. 24) use dynamic programming to solve the two-constraint case for parallel redundancy.

Sasaki (refs. 25 and 26) and Webster (ref. 27) develop a computational procedure for the case of multiple linear constraints. Tillman (ref. 28) and Tillman and Liittschwager (ref. 29) consider the optimal multiple linear and nonlinear separable constraint solutions to both parallel and standby problems by the method of integer programming. Mizukami (ref. 7) maximizes reliability subject to multiple constraints by the methods of concave and integer programming for the parallel redundancy case. Ghare and Taylor (ref. 6) solve the same type of problem by a multidimensional knapsack formulation using a branch-and-bound procedure.

Standby redundancy is discussed by Geisler and Karr (ref. 30), Black and Proschan (ref. 31), Morrison (ref. 32), Morrison and David (ref. 33), and Subba Rao and Natarajan (ref. 34). The concepts of sensitivity are discussed by Thakkar and Hughes (ref. 16), Breipohl (ref. 35), Kulakov and Zagoruyko (ref. 11), and Isayev and Mamed-Zade (ref. 36).

Several of the references present some approximate equations for the single-constraint parallel redundancy optimization problems. Federowicz and Mazumdar (ref. 9) use geometric programming and Mine (ref. 8) uses calculus of variations to obtain the optimum number of components as a function of cost. Moskowitz and McLean (ref. 10) use a variational technique and Kulakov and Zagoruyko (ref. 11) use sensitivity balancing to arrive at two different expressions for the optimum number of components as a function of reliability.

Optimum arrangement of components based on several types of failures and without constraints is discussed by von Neumann (ref. 37); Moore and Shannon (refs. 38 and 39); Birnbaum (ref. 40); Hanne (ref. 41); Barlow, Hunter, and Proschan (ref. 42); and Gordon (ref. 43). Morrison (ref. 44) considers not only costs associated with the components but also cost of system failure and cost of component failure.

Kondo (ref. 45) discusses a method of successive approximations for solving the nonlinear, parallel redundancy equations. Fan, Wang, Tillman, and Hwang (ref. 46) maximize by numerical methods the expected net profit of a system. Expected net profit
is the product of success probability and profit realized with successful operation, decreased by the cost of the system. Neuner and Miller (ref. 47) develop an empirical curve fit for the reliability-resource tradeoff of complex systems whose reliability equations may be difficult to derive. Beznosov, Zelentsov, and Romanov (ref. 48) realize the difficulty in trying to compare different schemes of redundancy, so they choose as a basis for comparison the mean time between failures and then compare parallel, stand-by, majority vote, and logic redundancy.

**SERIES-K-OUT-OF-N SYSTEM**

In this section, solutions to the redundancy allocation problem for serially connected k-out-of-n subsystems are considered. The special case \( k = 1 \) of the series-parallel system problem has been solved in the literature by several methods. Five of these existing techniques are adapted in this section to give solutions to the more general series-k-out-of-n problem.

Kettelle's method (ref. 3) is an application of dynamic programming. It gives the complete family of optimal solutions but has the disadvantage of being rather lengthy in application. A second method uses Lagrange multipliers as applied to optimization of discrete and, hence, nondifferentiable functions. A third method constructs successively larger allocations by adding to the system the redundant component which provides greatest improvement in reliability per unit of weighted costs. These last two methods are used to generate incomplete families of optimal solutions; but generally, adjacent solutions are close enough to each other so that a good approximation to the true solution can be obtained.

The problem of maximizing reliability subject to several cost constraints is treated by Ghare and Taylor (ref. 6) by the multidimensional knapsack formulation with a branch-and-bound procedure. Their procedure gives the exact solution to the series-parallel redundancy problem, and it is shown herein that the method can also be used for series-k-out-of-n redundancy optimization. Finally, the concave and integer programming method of Mizukami (ref. 7) is shown herein to give an approximate solution to the problem under consideration.

No new methods are developed in solving this problem. Rather it is shown that these existing techniques, proven applicable to the special case of series-parallel redundancy optimization problems, are applicable to the more general series-k-out-of-n problems under certain conditions which are specified. The relative merits of the five procedures are discussed.
Kettelle's Method

Statement of the problem. - Kettelle (ref. 3) presents an algorithm for allocating redundancy in the series-parallel system so as to maximize system reliability without exceeding a specified single-cost constraint (or equivalently, to minimize a single type of cost while maintaining at least a minimum reliability level). Proschan and Bray (ref. 4) extend Kettelle's algorithm to the problem of several cost constraints.

A system of \( m \) serially connected subsystems is specified. The system is operational if and only if each subsystem is operational; the \( i^{th} \) subsystem is operational if and only if at least \( k_i \) of the \( n_i \) components in that subsystem are operational (\( i = 1, \ldots, m \)). It is assumed that the system, subsystem, or component can exist in only two possible states: success or failure. The failure of one component has no effect on the failure of any other component. Suppose component \( i \) has a cost \( c_{ij} \) of the \( j^{th} \) type, where \( i = 1, \ldots, m \) and \( j = 1, \ldots, s \). A linear constraint exists on each cost as given by equation (4):

\[
\sum_{i=1}^{m} c_{ij} n_i \leq C_{j, \text{max}} \quad j = 1, \ldots, s; \quad n_i \text{ integer} \tag{4}
\]

A component of subsystem \( i \) has probability \( p_i \) of successfully functioning, independent of the states of the other components in the system. For the \( k_i \)-out-of-\( n_i \) subsystem, the subsystem reliability is given by equation (7) as

\[
r_i = \sum_{x=k_i}^{n_i} \binom{n_i}{x} p_i^x q_i^{n_i-x} \tag{7}
\]

System reliability is given by equation (1) as

\[
R = \prod_{i=1}^{m} r_i \tag{1}
\]

By combining equations (1) and (7), the system reliability \( R(\bar{n}) \), where \( \bar{n} = (n_1, \ldots, n_m) \), is
Problem I is to choose \( \overline{n} \), a vector of positive integers, in such a way that \( R(\overline{n}) \) is maximized subject to the constraints of equation (4). Conversely, one may be interested in minimizing the costs \( C_j(\overline{n}) \), where \( j = 1, \ldots, s \), in such a way that the reliability is at least as great as \( R_{\text{min}} \), as shown in equation (6) of problem II.

**Dynamic programming approach.** - The method developed by Kettelle (ref. 3) is a dynamic programming technique. Dynamic programming is a decomposition technique for solving multiple decision problems. The optimum redundancy problem fits this description because there are \( m \) decisions to be made concerning the optimum values of \( n_i (i = 1, \ldots, m) \). The approach is to decompose the \( m \)-decision variable problem into \( m \) one-decision variable problems. These \( m \) subproblems are, in the optimum redundancy application, easier to solve than the original problem. This decomposition is attained in such a way that the optimal solutions to the subproblems yield the optimal solution to the original problem. This is assured by the principle of optimality of dynamic programming of Bellman (ref. 12, p. 83) which says:

An optimal policy (rule for decisions)\(^1\) has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

This principle yields a useful recurrence relation, which follows after some preliminaries.

Consider the sequence of \( m \) functions defined by

\[
f_m(\overline{C}_{\text{max}}) = \max_{T_m} \prod_{i=1}^{m} r_i(n_i) \tag{13}
\]

where \( \overline{C}_{\text{max}} = (C_{j, \text{max}}, \ldots, C_{s, \text{max}} ; C_{j, \text{max}} \geq 0 (j = 1, \ldots, s; m = 1, 2, \ldots) ; \) and \( T_m \) is the set of \( n_i \) values defined by the constraints of equation (4).

\(^1\)Contents of parentheses are not included in Bellman's principle.
\begin{equation}
\sum_{i=1}^{m} c_{ij} n_i \leq C_{j, \text{max}} \quad j = 1, \ldots, s; \quad n_i \text{ integer}
\end{equation}

Since the constraints of equation (4) limit \( n_i \) in such a way that none of the costs is exceeded, it is obvious that

\begin{equation}
n_i \leq \min_{j=1}^{s} \left[ \frac{C_{j, \text{max}}}{c_{ij}} \right] \quad i = 1, \ldots, m
\end{equation}

where \([y]\) denotes the greatest integer less than or equal to \( y \). From equation (13) with \( m = 1 \),

\begin{equation}
f_1(\overline{C}_{\text{max}}) = \max f_1(n_1)
\end{equation}

But to maximize \( r_1(n_1) \), \( n_1 \) should be chosen as large as possible because reliability increases monotonically as \( n_1 \) increases. Putting the largest \( n_1 \) satisfying equation (14) into equation (15) yields

\begin{equation}
f_1(\overline{C}_{\text{max}}) = r_1 \left( \min_{j=1}^{s} \left[ \frac{C_{j, \text{max}}}{c_{1, j}} \right] \right)
\end{equation}

where again \([y]\) denotes the greatest integer less than or equal to \( y \).

The principle of optimality of Bellman (ref. 12, p. 83) yields the recurrence relation

\begin{equation}
f_m(\overline{C}_{\text{max}}) = \max_{n_m} \left\{ f_m(n_m) f_{m-1}(\overline{C}_{\text{max}} - c_{m n_m}) \right\}
\end{equation}

where \( \overline{C}_{\text{max}} - c_{m n_m} = \left( C_{1, \text{max}} - c_{1 n_m}, \ldots, C_{s, \text{max}} - c_{s n_m} \right) \) and where \( n_m \) must be chosen subject to the constraints

\begin{equation}
k_m \leq n_m \leq \min_{j=1}^{s} \left[ \frac{C_{j, \text{max}}}{c_{mj}} \right] \quad n_m \text{ integer}
\end{equation}
where \([y]\) denotes the greatest integer less than or equal to \(y\). The lower limit on \(n_m\) follows from the fact that there must be at least \(k_m\) components in the \(m^{th}\) subsystem in order for it to perform its intended function; that is, it is a \(k_m\)-out-of-\(n_m\) subsystem.

The recurrence relation of equation (17) allows uncoupling of the \(m\)-dimensional problem into subproblems. This is the principle of dynamic programming.

The dynamic programming algorithm of Kettelle (ref. 3), as modified by Proschan and Bray (ref. 4), is used here to generate successive allocations until either the design level of system reliability is reached or one of the cost constraints is exceeded. Thus, this procedure readily lends itself to either of the two redundancy allocation problems of interest, that is, maximizing reliability without exceeding any of several cost constraints or minimizing some function of the costs while achieving at least a minimum prescribed level of system reliability. Because of this, the procedure readily lends itself to applications where tradeoffs between reliability and costs are of interest.

**Undominated allocations.** - Proschan and Bray (ref. 4) define dominating allocations. Let \(C_{j}(\tilde{n})\) represent the amount of resource \(j\) utilized in redundancy allocation \(\tilde{n}\). Then \(\tilde{n}^1\) is said to dominate \(\tilde{n}^2\) if \(C_{j}(\tilde{n}^1) \leq C_{j}(\tilde{n}^2)\) for all \(j = 1, \ldots, s\) while \(R(\tilde{n}^1) \geq R(\tilde{n}^2)\). If, in addition, at least one of these inequalities is strict, \(\tilde{n}^1\) dominates \(\tilde{n}^2\) strictly. No strictly dominated allocation is a part of the set of optimum allocations.

The purpose of the dynamic programming procedure is to generate the complete family of undominated allocations, meaning those allocations which are not strictly dominated. The allocation \(\tilde{n}^2\) is undominated by \(\tilde{n}^1\) if any one of the following conditions is true:

1. \(R(\tilde{n}^1) \geq R(\tilde{n}^2)\), but at least one \(C_{j}(\tilde{n}^1) > C_{j}(\tilde{n}^2)\), where \(j = 1, \ldots, s\).

2. All \(C_{j}(\tilde{n}^1) \leq C_{j}(\tilde{n}^2)\), where \(j = 1, \ldots, s\), but \(R(\tilde{n}^1) < R(\tilde{n}^2)\).

3. All \(C_{j}(\tilde{n}^1) = C_{j}(\tilde{n}^2)\), where \(j = 1, \ldots, s\), and \(R(\tilde{n}^1) = R(\tilde{n}^2)\).

Kettelle's procedure (ref. 3) constructs undominated allocations for successively larger groups of subsystems until the entire system is included. For a single subsystem, each possible allocation is an undominated allocation because, with the addition of each redundant component, \(r_i\) increases and \(C_{ij}\) \((j = 1, \ldots, s)\) increases. The procedure consists of systematically eliminating all strictly dominated allocations, after which only undominated allocations are left. The final set of undominated allocations, satisfying the cost and/or reliability constraints, forms the complete family of solutions to the redundancy allocation problems.

**Kettelle's procedure used on an example problem.** - The details of the dynamic programming algorithm are discussed by Kettelle (ref. 3) and Proschan and Bray (ref. 4) and are not repeated here. The algorithm consists of the pairwise combination of the
m subsystems in all possible ways. For a system of m subsystems, there will, thus, be m - 1 pairings. Since each pairing is facilitated by a table, m - 1 tables will be generated in this procedure.

For the example problem, a system of four subsystems is specified. The i^{th} component has associated with it two cost characteristics, monetary cost $c_{i1}$ and weight $c_{i2}$, as well as success probability $p_i$. The i^{th} subsystem requires at least $k_i$ components; additional components are redundant.

Table I lists the component monetary cost, the component weight, the component success probability, and the minimum number of components required for subsystem success for each of the four subsystems. The system is illustrated in figure 3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Subsystem 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Minimum number of components required, $k_i$</td>
<td>3</td>
</tr>
<tr>
<td>Component monetary cost, $c_{i1}$</td>
<td>4</td>
</tr>
<tr>
<td>Component weight, $c_{i2}$</td>
<td>1</td>
</tr>
<tr>
<td>Component success probability, $p_i$</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Figure 3. - System of four k-out-of-n subsystems connected in series - example problem. Shading indicates redundant component.
In addition, the formulation of problem II is specified for this example. It is desired to generate the complete family of undominated allocations having reliabilities from 0.90 to 0.95. It is important to specify precisely the lower limit on system reliability - in this case $R_{\min} = 0.90$ - because the procedure begins there and produces solutions having increasingly higher reliabilities. That is, the tables generated by this method of solution can be extended to higher reliabilities, but the whole problem may have to be reworked in order to get lower reliabilities.

There is a multitude of different subsystem combinations which, when hooked together in a system, would yield system reliability somewhere in the specified range - in this case, 0.90 to 0.95. Each of these unique systems could be characterized by the quantities $R(n)$ and $C_j(n) (j = 1, \ldots, s)$. The steps of the procedure are outlined as applied to this problem.

Step 1: Plan the successive combinations of subsystems which use subsystems 1 and 2 together. Following this, combine subsystems 3 and 4 to generate the undominated allocations. For larger systems, continue this procedure until all subsystems have been combined.

The next level of combinations uses the results of the first set of combinations. That is, the undominated allocations for subsystems 1 to 4 are generated by using the undominated allocations of subsystems 1 and 2 in conjunction with the undominated allocations of subsystems 3 and 4. The combining stops when all subsystems have been included.

The combinations used for the example problem of table I are illustrated in figure 4. The combinations selected are of equal size; that is, (1, 2) paired with (3, 4). This is not necessary, however, in that the same optimal allocations are obtained by using an alternate sequence shown in figure 5. But, according to Barlow and Proschan (ref. 1), the procedure is more "regular" if the combinations, as nearly as possible, contain an equal number of subsystems. Thus, the combinations of figure 4 are used here in preference to those of figure 5.

Step 2: For each subsystem, determine the minimum number of components required to achieve the least allowable system reliability (in this case, $R_{\min} = 0.90$) for that subsystem alone. The reasoning here is that any subsystem configuration which cannot, by itself, support the reliability $R_{\min}$ will certainly not be a part of the system.

![Figure 4 - Combinations used for four-subsystem example problem](image1)

![Figure 5 - Alternate combinations for four-subsystem problems.](image2)
whose reliability is the product of the subsystem reliabilities. Because of the product rule, the system reliability can never be greater than the reliability of the least reliable subsystem.

These minimum requirements are obtained by determining the values of \( n_i \) \((i = 1, \ldots, m)\) which satisfy

\[
 r_i(n_i) \geq R_{\text{min}} \quad i = 1, \ldots, m; \ n_i \text{ integer}
\]  

(19)

The resultant values of \( n_i \) \((i = 1, \ldots, m)\) are called the base requirements. Once the base requirements are established, the rest of the computation concentrates on additional costs and additional reliability improvements above these base values. In other words, only configurations with reliability equal to or greater than \( R_{\text{min}} \) will be considered.

Base requirements, subsystem monetary costs, subsystem weights, and subsystem reliabilities for the example problem, determined from equation (19) with \( R_{\text{min}} = 0.90 \), are shown in table II. The total monetary cost of this basic system is 91 units, obtained from equation (2) with \( j = 1 \). When the same equation is used with \( j = 2 \), the total weight of the basic system is 24 units. The basic system reliability, obtained from equation (1), is 0.690886.

Step 3: For the first pair of subsystems, generate table III. The row headings pertain to subsystem 1, beginning with the basic subsystem and adding one component at a time to get the successive blocks of row headings. The four row headings in each block are \( n_1, C_{11} = c_{11}n_1, C_{12} = c_{12}n_1, \) and \( r_1(n_1) \). The column headings are the analogous quantities for subsystem 2, beginning with the basic subsystem and increas-
TABLE III - COMBINATION OF SUBSYSTEMS 1 AND 2 - EXAMPLE PROBLEM

(Combinations are formed as follows: number of components required, n1, n2; monetary cost, C11 + C21; weight, C12 + C22; reliability, r1 x r2. Shading outlines eliminated areas, line connects undominated allocations (completely shaded blocks).)

<table>
<thead>
<tr>
<th>Subsystem 1</th>
<th>k1 = 3, p1 = 0.6, c11 = 4; c12 = 1</th>
<th>Subsystem 2</th>
<th>k2 = 2, p2 = 0.7, c21 = 5, c22 = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n1</td>
<td>C11</td>
<td>C21</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>n1</td>
<td>7</td>
<td>7.4</td>
<td>48</td>
</tr>
<tr>
<td>C11</td>
<td>28</td>
<td>53</td>
<td>12</td>
</tr>
<tr>
<td>C12</td>
<td>7</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>r1</td>
<td>0.903744</td>
<td>0</td>
<td>0.870662</td>
</tr>
<tr>
<td>n1</td>
<td>8</td>
<td>8.4</td>
<td>52</td>
</tr>
<tr>
<td>C11</td>
<td>32</td>
<td>57</td>
<td>13</td>
</tr>
<tr>
<td>C12</td>
<td>8</td>
<td>66</td>
<td>16</td>
</tr>
<tr>
<td>r1</td>
<td>0.950193</td>
<td>0</td>
<td>0.920946</td>
</tr>
<tr>
<td>n1</td>
<td>9</td>
<td>9.4</td>
<td>56</td>
</tr>
<tr>
<td>C11</td>
<td>36</td>
<td>61</td>
<td>14</td>
</tr>
<tr>
<td>C12</td>
<td>9</td>
<td>66</td>
<td>16</td>
</tr>
<tr>
<td>r1</td>
<td>0.974965</td>
<td>0</td>
<td>0.920946</td>
</tr>
<tr>
<td>n1</td>
<td>10</td>
<td>10.5</td>
<td>60</td>
</tr>
<tr>
<td>C11</td>
<td>40</td>
<td>65</td>
<td>15</td>
</tr>
<tr>
<td>C12</td>
<td>10</td>
<td>70</td>
<td>16</td>
</tr>
<tr>
<td>r1</td>
<td>0.987705</td>
<td>0</td>
<td>0.957303</td>
</tr>
<tr>
<td>n1</td>
<td>11</td>
<td>11.5</td>
<td>69</td>
</tr>
<tr>
<td>C11</td>
<td>44</td>
<td>74</td>
<td>17</td>
</tr>
<tr>
<td>C12</td>
<td>11</td>
<td>79</td>
<td>18</td>
</tr>
<tr>
<td>r1</td>
<td>0.994076</td>
<td>0</td>
<td>0.983206</td>
</tr>
<tr>
<td>n1</td>
<td>12</td>
<td>12.6</td>
<td>73</td>
</tr>
<tr>
<td>C11</td>
<td>48</td>
<td>78</td>
<td>18</td>
</tr>
<tr>
<td>C12</td>
<td>12</td>
<td>83</td>
<td>19</td>
</tr>
<tr>
<td>r1</td>
<td>0.997190</td>
<td>0</td>
<td>0.986286</td>
</tr>
</tbody>
</table>
ing $n_2$ by one unit to obtain the next block of column headings; that is, the column headings in each block are $n_2$, $C_{21}$, $C_{22}$, and $r_2(n_2)$.

The question arises concerning the number of rows and columns needed. Provision should be left for extending the table if necessary, but successive headings should be posted until the subsystem reliability is large enough to support the largest system reliability of interest - in this case, 0.95. Because $R$ is the product of the $r_i$ ($i = 1, \ldots, m$), a good guess on how far to carry the row and column headings for table III is until $r_i \geq R^{1/m}$; in this case, until $r_i \geq (0.95)^{1/4} = 0.987259$. In addition, two additional rows and two additional columns are included in case later expansion of the table is needed.

Step 4: To construct the set of undominated combinations for the first two subsystems, start by entering in table III the monetary cost, the weight, and the reliability of the combination of subsystems 1 and 2. For example, the monetary cost to be entered in the upper left block is the summation of the monetary cost of subsystem 1 with $n_1 = 7$ and the monetary cost of subsystem 2 with $n_2 = 4$. That is, the monetary cost to be entered is $28 + 20 = 48$. Likewise, the weight of that block is $7 + 4 = 11$. The reliability of the combination is the product $r_1r_2 = 0.828101$.

Since this combination does not meet the system reliability requirement of at least 0.90, proceed to the next allocation by combining $n_1 = 8$ with $n_2 = 4$. This results in a monetary cost of 52, a weight of 12, and a reliability of 0.870662. This process continues in the direction of increasing reliability until a reliability of at least 0.90 is reached. This occurs at the intersection of the second row and second column giving a monetary cost of 57, a weight of 13, and a reliability of 0.920946. Thus, the allocation $(n_1, n_2) = (8, 5)$ is the first undominated allocation which satisfies the reliability constraint.

Often the entire remainder of a row or column is dominated and can be rejected from further consideration. If the reliability of a particular combination is at least as great as the reliability in the heading of a previous row, all entries in that previous row having monetary cost and weight at least as great as the present combination can be rejected. The reason is that such dominated entries always have monetary costs and weights at least as great as the dominating allocation and their reliabilities are lower.

For example, in table III, the reliability of combination $(8, 5)$ is greater than the reliability in the block heading the first row and, hence, all entries in that row costing 57 or more and weighing 13 or more can be rejected.

An analogous argument holds for eliminating from consideration the remaining portion of a column. For example, in table III, the combination $(8, 5)$ dominates anything in the first column costing 57 or more and weighing 13 or more. Shading is used in table III to outline the remainders of rows and columns which are eliminated from consideration by this means.

From this first undominated allocation $(8, 5)$ which satisfies the reliability con-
straint, other entries are posted only as needed, that is, until the reliability of the combination is able to support the system upper reliability goal of 0.95. Since two subsystems are combined in table III, the undominated set should be generated until the combination reliability $r_1r_2$ is equal to or greater than $R^{2/m}$; in this case, until the combination reliability is equal to or greater than $(0.95)^{1/2} = 0.974679$. In addition, two extra entries are generated to facilitate the next level of combinations. The undominated entries are joined together sequentially in order of increasing reliability, and they form the basis for the next level of combinations.

Table IV lists the complete family of undominated combinations of subsystems 1 and 2 which satisfy the reliability requirements, corresponding to the completely shaded blocks in table III.

<table>
<thead>
<tr>
<th>$(n_1,n_2)$</th>
<th>Monetary cost, $C_{11} + C_{21}$</th>
<th>Weight, $C_{12} + C_{22}$</th>
<th>Reliability, $r_1r_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8,5)</td>
<td>57</td>
<td>13</td>
<td>0.920946</td>
</tr>
<tr>
<td>(9,5)</td>
<td>61</td>
<td>14</td>
<td>0.944956</td>
</tr>
<tr>
<td>(10,5)</td>
<td>65</td>
<td>15</td>
<td>0.957303</td>
</tr>
<tr>
<td>(9,6)</td>
<td>66</td>
<td>15</td>
<td>0.964304</td>
</tr>
<tr>
<td>(10,6)</td>
<td>70</td>
<td>16</td>
<td>0.976804</td>
</tr>
<tr>
<td>(11,6)</td>
<td>74</td>
<td>17</td>
<td>0.983206</td>
</tr>
<tr>
<td>(10,7)</td>
<td>75</td>
<td>17</td>
<td>0.983961</td>
</tr>
</tbody>
</table>

Step 5: Similarly, for subsystems 3 and 4, the procedure as outlined in steps 2 to 4 can be used to generate the family of undominated combinations of those subsystems. For larger systems, pairing would continue for subsystems 5 and 6, 7 and 8, etc.

Table V shows the development of the undominated combinations of subsystems 3 and 4. Note that entry (2, 17) - which, of course, is in the sixth column - dominates everything in the fourth column costing 65 or more and weighing 19 or more. This illustrates the fact that, in eliminating partial rows or columns, attention should be given to previous rows or columns rather than just the immediately preceding row or column. Also the undominated allocations (2,16), (3,15), and (2,17) are joined sequentially in order of increasing reliability even though the last entry (2,17) costs less than the next to last entry (2,16). If there had been just a single constraint - monetary cost, allocation (3, 15) would have been eliminated because it would, under that condition, have been a dominated allocation. Table VI lists the complete family of undominated combinations of subsystems 3 and 4 satisfying the reliability requirements.
### TABLE V - COMBINATION OF SUBSYSTEMS 3 AND 4 - EXAMPLE PROBLEM

[Combinations are formed as follows: number of components required, $n_3$, $n_4$, C_{31} + C_{41}$, weight, C_{32} + C_{42}, reliability, $r_3 \times r_4$. Shading outlines eliminated areas, line connects undominated allocations (completely shaded blocks).]

<table>
<thead>
<tr>
<th>Subsystem 3</th>
<th>Subsystem 4</th>
<th>$n_3$</th>
<th>$n_4$</th>
<th>$C_{31} + C_{41}$</th>
<th>$C_{32} + C_{42}$</th>
<th>$r_3 \times r_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_3 = 1$, $r_3 = 0.9$, $c_{31} = 7$, $c_{32} = 1$</td>
<td>$k_3 = 4$, $p_3 = 0.5$, $c_{41} = 3$, $c_{42} = 1$</td>
<td>12</td>
<td>36</td>
<td>0.927002</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>$C_{31}$</td>
<td>$C_{32}$</td>
<td>1</td>
<td>1</td>
<td>0.953557</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.900000</td>
<td>0.834302</td>
<td>0.858471</td>
<td>0.874182</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>$n_4 = 2$</td>
<td>$n_4 = 2$</td>
<td>2</td>
<td>12</td>
<td>0.917732</td>
<td>21</td>
<td>3.16</td>
</tr>
<tr>
<td>$C_{41}$</td>
<td>$C_{42}$</td>
<td>14</td>
<td>50</td>
<td>0.944318</td>
<td>60</td>
<td>1.18</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.990000</td>
<td>0.961600</td>
<td>0.972509</td>
<td>0.979471</td>
<td>0.983701</td>
<td>0.988269</td>
</tr>
<tr>
<td>$n_3 = 3$</td>
<td>$n_4 = 4$</td>
<td>3</td>
<td>57</td>
<td>0.981440</td>
<td>60</td>
<td>0.988376</td>
</tr>
<tr>
<td>$C_{31}$</td>
<td>$C_{32}$</td>
<td>15</td>
<td>16</td>
<td>0.992643</td>
<td>72</td>
<td>0.995235</td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.999000</td>
<td>0.992643</td>
<td>0.995235</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n_3 = 28$</td>
<td>$n_4 = 4$</td>
<td>4</td>
<td>38</td>
<td>19</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>$C_{31}$</td>
<td>$C_{32}$</td>
<td>19</td>
<td>19</td>
<td></td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>$r_3$</td>
<td>0.999000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE VI. - UNDOMINATED COMBINATIONS OF SUBSYSTEMS 3 AND 4 SATISFYING RELIABILITY REQUIREMENTS - EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>(n₃, n₄)</th>
<th>Monetary cost, C₃₁ + C₄₁</th>
<th>Weight, C₃₂ + C₄₂</th>
<th>Reliability, r₃ᵣ₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2, 12)</td>
<td>50</td>
<td>14</td>
<td>0.917732</td>
</tr>
<tr>
<td>(2, 13)</td>
<td>53</td>
<td>15</td>
<td>0.944318</td>
</tr>
<tr>
<td>(2, 14)</td>
<td>56</td>
<td>16</td>
<td>0.961600</td>
</tr>
<tr>
<td>(2, 15)</td>
<td>59</td>
<td>17</td>
<td>0.972598</td>
</tr>
<tr>
<td>(2, 16)</td>
<td>62</td>
<td>18</td>
<td>0.979471</td>
</tr>
<tr>
<td>(3, 15)</td>
<td>66</td>
<td>18</td>
<td>0.981440</td>
</tr>
<tr>
<td>(2, 17)</td>
<td>65</td>
<td>19</td>
<td>0.983701</td>
</tr>
</tbody>
</table>

Step 6: The next step is to combine the undominated combinations obtained from subsystems 1 and 2 with the undominated combinations obtained from subsystems 3 and 4. This is shown in table VII. The entries along the left of the rows are the characteristics of the combination of subsystems 1 and 2: first, (n₁, n₂); second, the monetary cost of the combination, C₁₁ + C₂₁; third, the weight of the combination C₁₂ + C₂₂, and, fourth, the reliability of the combination r₁ᵣ₂. Likewise, the column heads are (n₃, n₄), C₃₁ + C₄₁, C₃₂ + C₄₂, and r₃ᵣ₄.

The undominated family satisfying the reliability range of interest becomes quite irregular, as shown by the lines joining the entries in table VII. The complete set of undominated allocations satisfying the reliability requirements is shown in table VIII. Weight and cost increased about 10 percent in raising the reliability from the minimum R satisfying R ≥ 0.90 to the minimum R satisfying R ≥ 0.95.

Figure 6 shows the same set of optimum allocations. The solid stepped line is the complete optimum function of reliability against investable monetary cost and weight. The number by each of the optimum points is the weight C₂.

The problem here was to generate all undominated solutions yielding reliabilities from 0.90 to 0.95. This has been accomplished. The procedure, however, need not be as lengthy if the optimum configuration for achieving a single specified reliability (say 0.90) is desired. For this case, table III would have required only three rows and two columns; table V, only two rows and three columns; and table VII, only two rows and three columns. But this method is really best suited for generating the complete family of undominated allocations over a range of R.
Combinations are formed as follows:

- number of components required, \( n_1, n_2, n_3, n_4 \)
- reliability, \( r_1, r_2, r_3, r_4 \)

Shading outlines eliminated areas.

### Table VII: Combinations of Subsystems

<table>
<thead>
<tr>
<th>Subsystems 1 and 2</th>
<th>( P_3 \cdot P_4 )</th>
<th>( C_{31} \cdot C_{41} )</th>
<th>( C_{32} \cdot C_{42} )</th>
<th>( r_3 \cdot r_4 )</th>
<th>( P_5 \cdot P_6 )</th>
<th>( C_{51} \cdot C_{61} )</th>
<th>( C_{52} \cdot C_{62} )</th>
<th>( r_5 \cdot r_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_1, n_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6, 5</td>
<td>9, 5, 2, 12</td>
<td>107</td>
<td>110</td>
<td>28</td>
<td>8, 5, 2, 14</td>
<td>113</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>( C_{11} \cdot C_{21} )</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{12} \cdot C_{22} )</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1 \cdot r_2 )</td>
<td>0.0500516</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( n_1, n_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9, 5, 2, 12</td>
<td>9, 5, 2, 12</td>
<td>111</td>
<td>114</td>
<td>29</td>
<td>9, 5, 2, 14</td>
<td>117</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>( C_{12} \cdot C_{21} )</td>
<td>61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{13} \cdot C_{22} )</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1 \cdot r_2 )</td>
<td>0.046956</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( n_1, n_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10, 5, 2, 12</td>
<td>10, 5, 2, 12</td>
<td>115</td>
<td>118</td>
<td>30</td>
<td>10, 5, 2, 14</td>
<td>121</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>( C_{11} \cdot C_{21} )</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{12} \cdot C_{22} )</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1 \cdot r_2 )</td>
<td>0.0013234</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( n_1, n_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9, 6, 2, 12</td>
<td>9, 6, 2, 12</td>
<td>116</td>
<td>118</td>
<td>30</td>
<td>9, 6, 2, 14</td>
<td>121</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>( C_{12} \cdot C_{21} )</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{13} \cdot C_{22} )</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1 \cdot r_2 )</td>
<td>0.004394</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( n_1, n_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10, 6, 2, 12</td>
<td>10, 6, 2, 12</td>
<td>120</td>
<td>123</td>
<td>31</td>
<td>10, 6, 2, 14</td>
<td>126</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>( C_{11} \cdot C_{21} )</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{12} \cdot C_{22} )</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1 \cdot r_2 )</td>
<td>0.006900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( n_1, n_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11, 6, 2, 14</td>
<td>11, 6, 2, 14</td>
<td>124</td>
<td>127</td>
<td>32</td>
<td>11, 6, 2, 14</td>
<td>130</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>( C_{11} \cdot C_{21} )</td>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{12} \cdot C_{22} )</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1 \cdot r_2 )</td>
<td>0.003369</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( n_1, n_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10, 7, 3, 13</td>
<td>10, 7, 3, 13</td>
<td>128</td>
<td>130</td>
<td>33</td>
<td>10, 7, 2, 14</td>
<td>131</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>( C_{12} \cdot C_{21} )</td>
<td>79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{13} \cdot C_{22} )</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_1 \cdot r_2 )</td>
<td>0.002901</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1, 2, 3, AND 4 - EXAMPLE PROBLEM

Monetary cost, $C_{11} + C_{21} + C_{31} + C_{41}$, weight, $C_{12} + C_{22} + C_{32} + C_{42}$, line connects undominated allocations (completely shaded blocks).

<table>
<thead>
<tr>
<th>$n_2$</th>
<th>$n_3$</th>
<th>$C_{31} + C_{41}$</th>
<th>$C_{23} + C_{43}$</th>
<th>$r_3\cdot r_4$</th>
<th>$n_2$</th>
<th>$n_3$</th>
<th>$C_{31} + C_{41}$</th>
<th>$C_{23} + C_{43}$</th>
<th>$r_3\cdot r_4$</th>
<th>$n_2$</th>
<th>$n_3$</th>
<th>$C_{31} + C_{41}$</th>
<th>$C_{23} + C_{43}$</th>
<th>$r_3\cdot r_4$</th>
<th>$n_2$</th>
<th>$n_3$</th>
<th>$C_{31} + C_{41}$</th>
<th>$C_{23} + C_{43}$</th>
<th>$r_3\cdot r_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,15</td>
<td>50</td>
<td>17</td>
<td>0</td>
<td>$972598$</td>
<td>2,15</td>
<td>62</td>
<td>18</td>
<td>0</td>
<td>$972471$</td>
<td>3,15</td>
<td>66</td>
<td>18</td>
<td>0</td>
<td>$881469$</td>
<td>2,17</td>
<td>85</td>
<td>19</td>
<td>0</td>
<td>$937051$</td>
</tr>
<tr>
<td>2,3,2,15</td>
<td>116</td>
<td>30</td>
<td>0</td>
<td>$895110$</td>
<td>9,5,2,16</td>
<td>119</td>
<td>31</td>
<td>0</td>
<td>$892043$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,3,2,15</td>
<td>311</td>
<td>32</td>
<td>0</td>
<td>$925657$</td>
<td>9,5,2,16</td>
<td>123</td>
<td>32</td>
<td>0</td>
<td>$927418$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,3,2,15</td>
<td>311</td>
<td>32</td>
<td>0</td>
<td>$925657$</td>
<td>9,5,2,16</td>
<td>127</td>
<td>33</td>
<td>0</td>
<td>$927418$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,3,2,15</td>
<td>311</td>
<td>32</td>
<td>0</td>
<td>$925657$</td>
<td>9,5,2,16</td>
<td>129</td>
<td>33</td>
<td>0</td>
<td>$927418$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,5,2,15</td>
<td>127</td>
<td>33</td>
<td>0</td>
<td>$937651$</td>
<td>10,3,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$938599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,5,2,15</td>
<td>127</td>
<td>33</td>
<td>0</td>
<td>$937651$</td>
<td>10,3,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$938599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,5,2,15</td>
<td>127</td>
<td>33</td>
<td>0</td>
<td>$937651$</td>
<td>10,3,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$938599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,5,2,15</td>
<td>127</td>
<td>33</td>
<td>0</td>
<td>$937651$</td>
<td>10,3,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$938599$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10,5,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$956949$</td>
<td>11,6,2,15</td>
<td>133</td>
<td>35</td>
<td>0</td>
<td>$950713$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10,5,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$956949$</td>
<td>11,6,2,15</td>
<td>133</td>
<td>35</td>
<td>0</td>
<td>$950713$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10,5,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$956949$</td>
<td>11,6,2,15</td>
<td>133</td>
<td>35</td>
<td>0</td>
<td>$950713$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10,5,2,15</td>
<td>133</td>
<td>34</td>
<td>0</td>
<td>$956949$</td>
<td>11,6,2,15</td>
<td>133</td>
<td>35</td>
<td>0</td>
<td>$950713$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>14</td>
<td>35</td>
<td>0</td>
<td>$9363022$</td>
<td>134</td>
<td>14</td>
<td>35</td>
<td>0</td>
<td>$934056$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>14</td>
<td>35</td>
<td>0</td>
<td>$9363022$</td>
<td>134</td>
<td>14</td>
<td>35</td>
<td>0</td>
<td>$934056$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>14</td>
<td>35</td>
<td>0</td>
<td>$9363022$</td>
<td>134</td>
<td>14</td>
<td>35</td>
<td>0</td>
<td>$934056$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

25
TABLE VIII. - UNDOMINATED COMBINATIONS OF
SUBSYSTEMS 1, 2, 3, AND 4 SATISFYING
RELIABILITY REQUIREMENTS -
EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>(n_1, n_2, n_3, n_4)</th>
<th>Monetary cost, (C_1)</th>
<th>Weight, (C_2)</th>
<th>Reliability, (R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(9, 5, 2, 14)</td>
<td>117</td>
<td>30</td>
<td>0.908669</td>
</tr>
<tr>
<td>(9, 6, 2, 13)</td>
<td>119</td>
<td>30</td>
<td>0.910610</td>
</tr>
<tr>
<td>(9, 5, 2, 15)</td>
<td>120</td>
<td>31</td>
<td>0.919062</td>
</tr>
<tr>
<td>(10, 5, 2, 14)</td>
<td>121</td>
<td>31</td>
<td>0.920543</td>
</tr>
<tr>
<td>(9, 6, 2, 14)</td>
<td>122</td>
<td>31</td>
<td>0.927274</td>
</tr>
<tr>
<td>(10, 5, 2, 15)</td>
<td>124</td>
<td>32</td>
<td>0.931071</td>
</tr>
<tr>
<td>(9, 6, 2, 15)</td>
<td>125</td>
<td>32</td>
<td>0.937880</td>
</tr>
<tr>
<td>(10, 6, 2, 14)</td>
<td>126</td>
<td>32</td>
<td>0.939391</td>
</tr>
<tr>
<td>(9, 6, 2, 16)</td>
<td>128</td>
<td>33</td>
<td>0.944508</td>
</tr>
<tr>
<td>(10, 6, 2, 15)</td>
<td>129</td>
<td>33</td>
<td>0.950135</td>
</tr>
</tbody>
</table>

Figure 6. - Optimum solutions to example problem showing system reliability as function of system monetary cost with weight as the parameter.
Generalized Lagrange Multiplier Method

Usefulness of the technique. - The generalized Lagrange multiplier method is discussed by Everett (ref. 5). This method offers a means of obtaining solutions to the optimization of nondifferentiable functions.

The technique does not, in general, produce all the undominated solutions as does the previous dynamic programming technique. The allocations generated by this method are, however, shown by Everett (ref. 5) to be optimum.

Consider the \((s + 1)\)-dimensional space of reliability \(R\) against resource expenditures \(C_j, j = 1, \ldots, s\). The problem of finding the maximum of \(R\) subject to the constraints \(C_j, \max (j = 1, \ldots, s)\) is simply the problem of selecting that point in \(RC_j\) space of maximum \(R\) contained in the subspace bounded by the constraints. As the constraints are varied, a set of optimum solutions is generated.

Figure 7 illustrates a typical set of possible allocations, along with the complete family of undominated allocations and the incomplete set generated by Lagrange multipliers, for the simple case of a single type of cost. Only a part of the complete family of solutions is generated by this method, namely, those points lying on the convex hull of the reliability-cost space. The points generated by this procedure are, thus, in a sense the "best" of the optimum solutions.

The solutions produced by the method of generalized Lagrange multipliers are usually close enough together so that the fact that some solutions are missed is not often a problem. However, on occasion, when these so-called gaps occur in regions of critical interest, there are several useful techniques that can be used to fill the gaps. These are discussed by Everett (ref. 5) and are not repeated herein.

![Possible allocations, Undominated allocations, Lagrange solutions](image)

Figure 7 - Typical reliability-cost space for single type of cost
**Formulation of the method.** - The basis of the integer Lagrange method is that the Lagrangian can be maximized by any means whatsoever. In differentiable functions this is done by setting the derivative to zero. The Lagrangian can also be maximized by trial and error, computer scanning of all possibilities, or analytical maximization (by zeroing the derivative) of a continuous function which approximates the discrete function. Lagrange multipliers is a technique for converting constrained optimization problems into unconstrained maximization problems.

The problem to be solved is stated in terms of problem I by equations (3) and (4) which are repeated here.

Problem I: Maximize

\[
R = \prod_{i=1}^{m} r_i
\]  

subject to

\[
\sum_{i=1}^{m} c_{ij} n_i \leq C_{j, \text{max}} \quad j = 1, \ldots, s; \quad n_i \text{ integer}
\]

The logarithm of \( R \) is a monotonic function of \( R \), so equation (3) could be restated as follows:

Problem I: Maximize

\[
\ln R = \sum_{i=1}^{m} \ln r_i
\]

The objective function of equation (7) is now in the form of a general cell or separable problem. There are \( m \) subsystems (cells) into which the resources may be committed and for which the overall objective function is simply the sum of the objective functions of each cell.

Everett's technique (ref. 5) applies to the solution of cell problems for which the objective function is a concave function. It is shown in appendix A that \( \ln R \) is concave when
for the series-$k$-out-of-$n$ system.

Under these conditions, the generalized Lagrange multiplier technique can be used. The technique requires the use of Lagrange multipliers $\lambda_j$ ($j = 1, \ldots, s$) associated with each of the $s$ types of cost. The unconstrained objective function for solving this problem by Everett's method (ref. 5) is to maximize

$$H(n_i) = \ln r_i(n_i) - \sum_{j=1}^{s} \lambda_j c_{ij} n_i$$

for each subsystem independently of the strategy choices in other subsystems. The resulting allocation is designated $\overline{n} = (n_1, \ldots, n_m)$. Summing $\ln r_i(n_i)$ over $i = 1, \ldots, m$ and exponentiating yields total system reliability $R(\overline{n})$. Summing $c_{ij} n_i$ over $i = 1, \ldots, m$ yields the total system cost of the $j$th type, $C_j(\overline{n})$. This allocation is a solution to the overall constrained problem with constraints equal to the total resources $C_j(\overline{n})$ ($j = 1, \ldots, s$) consumed by the strategy $\overline{n}$.

The proof that equation (21) leads to the same optimal allocation as equation (20) follows that of Everett (ref. 5) and is presented in appendix B: That proof shows that any solutions produced by Lagrange multipliers are optimum, undominated solutions.

Experience has shown that a good practice is to set all except one value of $\lambda_j$ ($j = 1, \ldots, s$) to zero and vary the nonzero $\lambda_j$ in such a way that at most one $n_i$ changes by at most one unit until all allocations are produced which yield system reliabilities of interest. This is then repeated for each value of $j$. If more solutions are desired, combinations of the $\lambda_j$'s used previously are tried.

Figure 8 shows the steps in Everett's generalized Lagrange multiplier procedure.

Lagrange multipliers used on an example problem. - The example problem given in the preceding main section is used here to illustrate the usefulness of this method in
generating a part of the family of undominated allocations. The system is illustrated in figure 3. The goal is to generate the incomplete set of undominated allocations having reliabilities from 0.90 to 0.95.

Table IX shows the solutions produced by a series of \( \lambda_1, \lambda_2 \) values. Inspection of the results for varying only one of the \( \lambda \)'s while keeping the other fixed shows that the changes in allocation from one solution to the next consist of at most one additional component in at most one subsystem. Values of the Lagrange multiplier between these solutions will thus produce no new solutions.

Combinations of nonzero \( \lambda_1 \) and \( \lambda_2 \) were tried and produced only repeats of those optimum allocations presented in table IX.

Figure 9 shows the solutions produced by this method superimposed on the complete family of undominated allocations. Several points cannot be obtained from the use of Lagrange multipliers. The importance of these missed points is less pronounced for
### Table IX. - Lagrange Multiplier Solutions to Example Problem for System Reliability from 0.90 to 0.95

<table>
<thead>
<tr>
<th>$\lambda_1$</th>
<th>$\lambda_2$</th>
<th>$n_1$</th>
<th>$n_2$</th>
<th>$n_3$</th>
<th>$n_4$</th>
<th>Monetary cost, $C_1$</th>
<th>Weight, $C_2$</th>
<th>Reliability, $R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005</td>
<td>0</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>117</td>
<td>30</td>
<td>0.908669</td>
</tr>
<tr>
<td>0.004</td>
<td>9</td>
<td>6</td>
<td>14</td>
<td>15</td>
<td>122</td>
<td>31</td>
<td>.927274</td>
<td></td>
</tr>
<tr>
<td>0.0035</td>
<td>9</td>
<td>10</td>
<td>15</td>
<td>14</td>
<td>125</td>
<td>32</td>
<td>.937880</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.02</td>
<td>9</td>
<td>13</td>
<td>14</td>
<td>129</td>
<td>33</td>
<td>.950135</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>.015</td>
<td>9</td>
<td>14</td>
<td>122</td>
<td>31</td>
<td>a.927274</td>
<td>.950135</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>.012</td>
<td>10</td>
<td>14</td>
<td>128</td>
<td>32</td>
<td>a.939391</td>
<td>.950135</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>.01</td>
<td>10</td>
<td>15</td>
<td>129</td>
<td>33</td>
<td>a.950135</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Repeat solution.*

---

**Figure 9.** - Optimum solutions by Lagrange multipliers to example problem showing system reliability as a function of system monetary cost with weight as the parameter.
systems consisting of many subsystems because, in general, more solutions are generated. Everett (ref. 5) describes several schemes for generating the missed solutions.

An advantage for the Lagrange multiplier technique is that, rather than producing the entire set of undominated allocations, it allows one to approach more readily a reliability or cost constraint.

Figure 10 shows the complete set of optimum solutions plotted as reliability as a function of a single type of cost. Each pair of convex hull points produced by the method of Lagrange is connected by a straight dashed line to show why these points are referred to as the "best" of the undominated solutions.

![Figure 10. Optimum solutions by Lagrange multipliers to example problem showing system reliability as function of system monetary cost and weight.](image-url)
Method of Balancing Sensitivities

Usefulness of the technique. - Another procedure which readily produces some of
the optimum solutions is discussed by Barlow and Proschan (ref. 1), Breipohl (ref. 35),
Isayev and Mamed-Zade (ref. 36), Kulakov and Zagoruyko (ref. 11), and Thakkar and
Hughes (ref. 16). The procedure consists of selecting consecutively for redundancy the
subsystem which has the most to offer in increasing reliability per unit of weighted
costs.

The technique produces only a part of the complete family of undominated alloca-
tions. It is useful in generating a set of solutions over a range of reliabilities and costs
rather than in approaching readily a reliability or cost constraint. As with the two pre-
vious techniques, the solutions obtained are optimum and, hence, satisfy the formula-
tions of both problems I and II. This method is shorter to use than Lagrange multi-
pliers and involves less trial and error.

Formulation of the method. - The optimality of this procedure is dependent upon the
concavity of the subsystem reliability \( r_i(n_i) \) (\( i = 1, \ldots, m \)). That condition is satisfied
by equation (A6)

\[
n_i > \frac{k_i - 1}{p_i} - 1 \quad i = 1, \ldots, m; \quad n_i \text{ integer}
\]  

(A6)

for the case where the subsystems are of the k-out-of-n type.

The cheapest allocation, but one which may not satisfy reliability requirements, is
one with no redundancy. In such a system, \( n_i = k_i \) (\( i = 1, \ldots, m \)). It is unreasonable to
start at this allocation, however, because the resultant system would likely be very un-
reliable or else redundancy would not be considered. Instead a more reasonable start-
ing point is with the system consisting of so-called basic subsystems. A basic subsys-
tem has been defined in equation (19) as one whose subsystem reliability is at least as
great at the minimum system reliability goal \( R_{min} \). That is, choose \( n_i \) such that

\[
r_i(n_i) \geq R_{min} \quad i = 1, \ldots, m; \quad n_i \text{ integer}
\]  

(19)

For a single type of cost \( (s = 1) \), the sensitivity of the \( i^{th} \) subsystem is the ratio
of the change in system reliability to the change in cost resulting from the addition of
only one component in only the \( i^{th} \) subsystem. Suppose the allocation of basic sub-
systems satisfying equation (19) is designated \( (n_1, \ldots, n_m) \). The next solution is obtained
by adding one component in the subsystem where it will provide the greatest improve-
ment in system reliability per unit cost, that is, by selecting for redundancy the sub-
system with greatest sensitivity. The sensitivity of the \( i^{th} \) subsystem is just
\[
S_i = \frac{R(n_1, \ldots, n_{i-1}, n_i + 1, n_{i+1}, \ldots, n_m) - R(n_1, \ldots, n_m)}{c_i} \quad i = 1, \ldots, m
\] (22)

The subsystem having maximum \( S_i \) \((i = 1, \ldots, m)\) is selected for redundancy. The new allocation is used to compute a new set of \( S_i \) \((i = 1, \ldots, m)\), and successive solutions are generated until either the cost or reliability constraint is reached.

Sensitivity must be redefined for the case of multiple cost factors. It is the ratio of change in system reliability to change in some weighted cost resulting from the addition of only one component in only the \( i \)th subsystem. The sensitivity of the \( i \)th subsystem is

\[
S_{i, j} = \frac{R(n_1, \ldots, n_{i-1}, n_i + 1, n_{i+1}, \ldots, n_m) - R(n_1, \ldots, n_m)}{\sum_{j=1}^{s} a_j c_{ij}} \quad i = 1, \ldots, m
\] (23)

where the \( a_j \) \((j = 1, \ldots, s)\) are nonnegative weights satisfying

\[
\sum_{j=1}^{s} a_j = 1
\] (24)

The vectors \((a_1, \ldots, a_s)\) are chosen by varying the \( a_j \) by some fixed increment until all choices from \((1,0,\ldots,0)\) to \((0,\ldots,0,1)\) have been exhausted (ref. 1).

Equation (8) can be used with equation (1) to give

\[
R(n_1, \ldots, n_m) = \prod_{i=1}^{m} \left[ \sum_{x=0}^{k_i-1} \left( \frac{k_i - 1 + x}{q_i} \right)^{k_i} \right]
\] (25)

and

\[
R(n_1, \ldots, n_{i-1}, n_i + 1, n_{i+1}, \ldots, n_m) = R(n_1, \ldots, n_m) \left[ 1 + p_i q_i^{n_i-k_i+1} \binom{n_i}{k_i-1} \right]
\] (26)
Using equation (26) in (28) yields

$$S_i = \frac{\sum_{j=1}^{s} a_j c_{ij}}{k_i \binom{n_i}{k_i-1} R(n_1, \ldots, n_m)} \quad i = 1, \ldots, m \quad (27)$$

A set \((a_1, \ldots, a_s)\) is selected and \(S_i\) is computed for each value of \(i = 1, \ldots, m\); the subsystem having the maximum sensitivity is selected to receive one additional component.

It is noted that \(R(n_1, \ldots, n_m)\) is a constant for all \(i, \ldots, m\) so a new quantity is defined by dividing both sides of equation (27) by \(R(n_1, \ldots, n_m)\),

$$B_i = \frac{k_i \binom{n_i}{k_i-1}}{\sum_{j=1}^{s} a_j c_{ij}} \quad i = 1, \ldots, m; \quad n_i \text{ integer} \quad (28)$$

The subsystem having the maximum value of \(B_i\) is selected for the next level of redundancy. The process is repeated until a cost or reliability constraint is reached. A new set of \(a_j\) \((j = 1, \ldots, s)\) is chosen and the entire procedure repeated. The steps are shown in figure 11.

Balancing sensitivities for an example problem. - Table X shows the operation of this procedure on the example problem previously discussed in connection with Kettelle's procedure and the Lagrange multiplier method. The reliability range of interest here again is 0.90 to 0.95. The basic requirements, which exceed the minimal concavity conditions and which are used to initialize the procedure, are \(n_1 = 7, n_2 = 4, n_3 = 1,\) and \(n_4 = 12\). The solutions having reliability from 0.90 to 0.95 are the same ones determined by the integer Lagrange method. This is because both methods determine points on the convex hull of the possible allocations.

Values of \(a_1\) from 0.0 to 1.0 in increments of 0.05 (and \(a_2 = 1 - a_1\)) failed to produce any new solutions.
Choose \( a_j \) \((j = 1, \cdots, s)\)

Choose \( n_j \) such that \( r_j(n_j) \geq R_{\text{min}} \)

\[ n_j = \max (n_j, \text{min integer} > \frac{k_l - 1}{\rho_l - 1}) \]

Select \( i \) having \( \max B_i \)

Are constraints reached?

Have all \( a_1, \cdots, a_s \) been investigated?

Figure 11. - Steps in procedure of balancing sensitivities.
TABLE X - SENSITIVITY PROCEDURE ON EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>n₁</th>
<th>B₁</th>
<th>n₂</th>
<th>B₂</th>
<th>n₃</th>
<th>B₃</th>
<th>n₄</th>
<th>B₄</th>
<th>Reliability, R</th>
<th>Monetary cost, C₁</th>
<th>Weight, C₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.011612</td>
<td>4</td>
<td>0.010584</td>
<td>1</td>
<td>0.012857</td>
<td>12</td>
<td>0.008952</td>
<td>0.690886</td>
<td>91</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>0.011612</td>
<td>4</td>
<td>0.010584</td>
<td>2</td>
<td>0.001286</td>
<td>13</td>
<td>0.008952</td>
<td>0.759974</td>
<td>98</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>0.008193</td>
<td>4</td>
<td>0.010584</td>
<td>2</td>
<td>0.001286</td>
<td>14</td>
<td>0.008952</td>
<td>0.799034</td>
<td>102</td>
<td>26</td>
</tr>
<tr>
<td>8</td>
<td>0.008193</td>
<td>5</td>
<td>0.003969</td>
<td>1</td>
<td>0.001286</td>
<td>13</td>
<td>0.008952</td>
<td>0.845182</td>
<td>107</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>0.008193</td>
<td>5</td>
<td>0.003969</td>
<td>2</td>
<td>0.001286</td>
<td>14</td>
<td>0.008952</td>
<td>0.869666</td>
<td>110</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>0.003185</td>
<td>6</td>
<td>0.003969</td>
<td>2</td>
<td>0.001286</td>
<td>14</td>
<td>0.005819</td>
<td>0.892339</td>
<td>114</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>0.003185</td>
<td>6</td>
<td>0.001429</td>
<td>1</td>
<td>0.001286</td>
<td>13</td>
<td>0.005819</td>
<td>0.908669</td>
<td>117</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>0.003185</td>
<td>6</td>
<td>0.001429</td>
<td>2</td>
<td>0.001286</td>
<td>14</td>
<td>0.003703</td>
<td>0.927274</td>
<td>122</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>0.001593</td>
<td>6</td>
<td>0.001429</td>
<td>2</td>
<td>0.001286</td>
<td>15</td>
<td>0.002314</td>
<td>0.937880</td>
<td>125</td>
<td>32</td>
</tr>
</tbody>
</table>

\( a₁ = 1, a₂ = C \)

<table>
<thead>
<tr>
<th>n₁</th>
<th>B₁</th>
<th>n₂</th>
<th>B₂</th>
<th>n₃</th>
<th>B₃</th>
<th>n₄</th>
<th>B₄</th>
<th>Reliability, R</th>
<th>Monetary cost, C₁</th>
<th>Weight, C₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.046449</td>
<td>4</td>
<td>0.052920</td>
<td>1</td>
<td>0.090000</td>
<td>12</td>
<td>0.026855</td>
<td>0.690886</td>
<td>91</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>0.046449</td>
<td>4</td>
<td>0.052920</td>
<td>2</td>
<td>0.009000</td>
<td>13</td>
<td>0.026855</td>
<td>0.759974</td>
<td>98</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>0.046449</td>
<td>5</td>
<td>0.019845</td>
<td>1</td>
<td>0.009000</td>
<td>13</td>
<td>0.026855</td>
<td>0.829881</td>
<td>103</td>
<td>26</td>
</tr>
<tr>
<td>8</td>
<td>0.024773</td>
<td>5</td>
<td>0.019845</td>
<td>2</td>
<td>0.009000</td>
<td>13</td>
<td>0.026855</td>
<td>0.845182</td>
<td>107</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>0.024773</td>
<td>6</td>
<td>0.019845</td>
<td>2</td>
<td>0.009000</td>
<td>13</td>
<td>0.017456</td>
<td>0.869666</td>
<td>110</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>0.012740</td>
<td>6</td>
<td>0.007144</td>
<td>1</td>
<td>0.009000</td>
<td>13</td>
<td>0.017456</td>
<td>0.892339</td>
<td>114</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>0.012740</td>
<td>6</td>
<td>0.007144</td>
<td>2</td>
<td>0.009000</td>
<td>14</td>
<td>0.011108</td>
<td>0.910610</td>
<td>119</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>0.006370</td>
<td>6</td>
<td>0.007144</td>
<td>2</td>
<td>0.009000</td>
<td>14</td>
<td>0.011108</td>
<td>0.927274</td>
<td>122</td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td>0.006370</td>
<td>6</td>
<td>0.007144</td>
<td>3</td>
<td>0.009000</td>
<td>15</td>
<td>0.006943</td>
<td>0.939391</td>
<td>126</td>
<td>32</td>
</tr>
</tbody>
</table>

\( a₁ = 0, a₂ = 1 \)

\( ^a \)Repeat solution.
Conversion of Problem I to Multidimensional Knapsack Formulation

Ghare and Taylor (ref. 6) develop a method for determining the optimum number of redundant components in a series-parallel system subject to several cost constraints. This is a nonlinear integer programming problem which they show can be transformed to an associated zero-one programming problem solvable by a branch-and-bound procedure. Their work shows that the optimal solution to the associated problem is equivalent to the optimal solution for the redundancy problem.

The purpose of this section is to show that the series-k-out-of-n redundancy allocation problem can also be transformed to the multidimensional knapsack formulation and, hence, is amenable to solution by the same techniques used by Ghare and Taylor in reference 6. The knapsack problem is discussed in a general way by Saaty (ref. 21).

The nonlinear programming formulation of the series-k-out-of-n redundancy allocation problem is stated as problem I (eq. (3))

Problem I: Maximize

\[
R = \sum_{i=1}^{m} r_i
\]  

subject to the several cost constraints of equation (4).

\[
\sum_{i=1}^{m} c_{ij}n_i \leq C_{j,\text{max}} \quad j = 1, \ldots, s; \quad n_i \text{ integer}
\]  

The subsystem reliability \( r_i \) \((i = 1, \ldots, m)\) is given by equation (8) for k-out-of-n subsystems as

\[
r_i = \sum_{x=0}^{n_i-k_i} \binom{k_i}{x} \left( \frac{k_i-1+x}{q_i} \right)^{p_i} x \quad i = 1, \ldots, m; \quad k_i, n_i \text{ integer}
\]

When equation (8) is used in equation (3), problem I becomes
Problem I: Maximize

\[ R = \begin{bmatrix} n_1 & \cdots & n_m \end{bmatrix} \begin{bmatrix} \sum_{x=0}^{n_i-k_i} \left( \begin{array}{c} k_i - 1 + x \\ x \end{array} \right) q_i^x \end{bmatrix} \begin{bmatrix} k_i, n_i \text{ integer} \end{bmatrix} \]

subject to the constraints of equation (4).

A feasible solution to this problem is the set of numbers \( \mathbf{n} = (n_1, \ldots, n_m) \) satisfying the constraints of equation (4). In order to solve this problem, consider an alternate problem, designated problem IA. It must be shown that a one-to-one correspondence exists between the feasible solutions for problems I and IA as applied to the series-k-out-of-n system.

The associated zero-one programming problem is

Problem IA: Maximize the quantity

\[ Z = \sum_{i=1}^{m} \sum_{l=1}^{\infty} a_{il} x_{il} \]

subject to the constraints

\[ \sum_{i=1}^{m} \sum_{l=1}^{\infty} c_{ij} x_{il} \leq b_j \quad j = 1, \ldots, s \]

where \( x_{il} \) equals 0 or 1 and where \( x_{il} = 0 \) implies \( x_{i l'} = 0 \) if \( l' > l \).

Further, let the following relations hold:

\[ a_{il} = \ln \left( \sum_{x=0}^{k_i - 1 + x} \left( \begin{array}{c} k_i - 1 + x \\ x \end{array} \right) q_i^x \right) - \ln \left( \sum_{x=0}^{k_i - 1 + x} \left( \begin{array}{c} k_i - 1 + x \\ x \end{array} \right) q_i^x \right) \quad i = 1, \ldots, m; \quad k_i \text{ integer} \]

\[ b_j = c_{j, \max} - \sum_{i=1}^{m} c_{ij} k_i \quad j = 1, \ldots, s \]
In order to show a one-to-one correspondence between the feasible solutions to problems I and IA, it is necessary (1) to express the set of numbers \( \vec{n} \) satisfying problem I in terms of problem IA variables and (2) to show that \( R \) is maximized when \( Z \) is maximized.

Let \( X = \{x_{1l}\} \) be a feasible solution to problem IA and let \( l_i \) be the largest index such that \( x_{1l} = 1 \). Using this and equation (33) in equation (31) yields

\[
\sum_{i=1}^{m} c_{ij} l_i \leq C \cdot j, \max - \sum_{i=1}^{m} c_{ij} k_i \quad j = 1, \ldots, s
\]  

(34)

Rearranging equation (34) results in

\[
\sum_{i=1}^{m} c_{ij} (l_i + k_i) \leq C \cdot j, \max \quad j = 1, \ldots, s; \quad l_i, k_i \text{ integer}
\]  

(35)

Now equation (35) is equivalent to the constraint of problem I, given by equation (4), when

\[
n_i = l_i + k_i \quad i = 1, \ldots, m; \quad l_i, k_i, n_i \text{ integer}
\]  

(36)

Thus, the set of numbers \( \vec{n} \) forming the solution to problem I is expressible in terms of problem IA variables.

To show that \( R \) is maximized when \( Z \) is maximized, equation (32) is used in equation (30) along with the definition of \( l_i \) being the largest index such that \( x_{1l} = 1 \),

\[
Z = \sum_{i=1}^{m} \sum_{l=1}^{l_i} \left \{ \ln \left [ \sum_{x=0}^{l} \left ( \frac{k_i - 1 + x}{x} q_i^x \right ) \right ] - \ln \left [ \sum_{x=0}^{l-1} \left ( \frac{k_i - 1 + x}{x} q_i^x \right ) \right ] \right \} \quad l_i, k_i \text{ integer}
\]  

(37)

When the second summation, that is, the summation from \( l = 1 \) to \( l = l_i \), is expanded, successive terms of the summation cancel and \( Z \) reduces to
When the relation of equation (36) is used, equation (38) becomes

\[
Z = \sum_{i=1}^{m} \left\{ \ln \left( \sum_{x=0}^{n_i-k_i} \left( \frac{k_i - 1 + x}{x} \right) q_i^x \right) \right\} \quad k_i, n_i \text{ integer} \tag{39}
\]

From equation (29), the natural logarithm of \( R \) is

\[
\ln R = \sum_{i=1}^{m} k_i \ln p_i + \sum_{i=1}^{m} \left\{ \ln \left( \sum_{x=0}^{n_i-k_i} \left( \frac{k_i - 1 + x}{x} \right) q_i^x \right) \right\} \quad k_i, n_i \text{ integer} \tag{40}
\]

Using equation (40) in equation (39) results in

\[
Z = \ln R - \sum_{i=1}^{m} k_i \ln p_i \quad k_i \text{ integer} \tag{41}
\]

The summation term in equation (41) is a constant. Since \( R \) and \( \ln R \) maximize simultaneously, equation (41) shows that when \( Z \) is a maximum, \( R \) is also. This completes the proof of a one-to-one correspondence between problems I and IA.

Problem IA is a multidimensional knapsack problem for which Ghare and Taylor (ref. 6) develop a solution by a branch-and-bound procedure. The optimum solution to problem IA is obtained and converted to the optimal solution for problem I by the relation
\[ n_i = l_i + k_i \quad (i = 1, \ldots, m). \]

The method of this section is relatively free of the problem of dimensionality that plagues the dynamic programming techniques, according to Ghare and Taylor (ref. 6). The solutions produced are exact solutions.

It has been shown that the method of Ghare and Taylor (ref. 6) is applicable to the problem of optimally allocating redundancy in the series-k-out-of-n system in such a way that reliability is maximized without exceeding any of s cost constraints.

### Solution of Problem I by Method of Integer Concave Programming

Mizukami (ref. 7) presents a method for determining the optimal redundancy for the series-parallel system so as to achieve maximum system reliability under multiple cost constraints. He approximates the reliability function with piecewise linearization and, thus, obtains a solution which is approximate.

The method used is a synthesis of the methods of concave and integer programming. He refers to his method as the method of convex and integer programming because his general problem involves minimization of a convex objective function. For reliability optimization, however, the problem is one of maximizing a concave reliability function. Hence, the method which he uses is, in reality, integer concave programming, the maximization analog of integer convex programming.

The system reliability is the product of the subsystem reliabilities and, hence, is nonlinear; thus, exact solutions are not attainable by linear programming. In the method of Mizukami (ref. 7) the objective function to be maximized is first converted to a sum of separable concave functions, which are approximated by piecewise linearization. Thus, the methods of linear programming can be used for optimizing the approximate objective function. A method of integer linear programming is finally used to obtain the optimal redundancy allocation in integral numbers of components.

The purpose of this section is to show that the method used by Mizukami (ref. 7) for the series-parallel system can be used as well for the series-k-out-of-n system with some limitations on the objective function.

The reliability of the \( i \)th subsystem when it contains \( n_i \) components is given by equation (8) as

\[
 r_i = p_i \sum_{x=0}^{n_i-k_i} \left( \frac{k_i - 1 + x}{q_i} \right) x_i \quad i = 1, \ldots, m; \quad k_i, n_i \text{ integer} \quad (8)
\]
The \( r_i \) as determined from equation (8) is a step function of \( n_i \), as shown for example in figure 12, because \( n_i \) is an integer. To use the method of concave programming, a continuous function is needed. A continuous approximation to the step function, drawn through the integer points, is shown in figure 12.

Replacing \( n_i \) by \( x_i \) \((i = 1, \ldots, m)\) to denote continuity (for the time being) in the number of components, the reliability of the \( i \)th subsystem with \( x_i \) components is \( r_i(x_i) \) \((i = 1, \ldots, m)\). Since \( \ln R \) is a monotone-increasing function of \( R \), maximizing \( \ln R \) is equivalent to maximizing \( R \). Thus, problem I can be rewritten as

Problem I: Maximize

\[
\ln R = \sum_{i=1}^{m} \ln r_i(x_i) \quad (42)
\]

subject to the cost constraints

\[
\sum_{i=1}^{m} c_{ij} x_i \leq C_{j, \text{max}} \quad j = 1, \ldots, s \quad (43)
\]

where

\[
x_i \geq k_i \quad i = 1, \ldots, m \quad (44)
\]
The objective function of equation (42) must be concave in order to use the method of concave programming. This results in a further restriction on $x_i$, as developed in appendix A,

$$x_i > \frac{k_i - 1}{p_i} - 1 \quad i = 1, \ldots, m$$  \hspace{1cm} (45)

In figure 13 the concave portion of $\ln r_i(x_i)$ is approximated by a broken-line function. Since $\ln r_i(x_i)$ is continuous, $h + 1$ points on the curve can be selected such that

the broken-line fit through these points is a sufficiently close approximation, according to Dantzig (ref. 49). The rectilinear equation is of the form

$$y_i = \lambda_{i,t}(x_i - k_i) + \mu_i, t \quad i = 1, \ldots, m; \ t = 1, \ldots, h$$  \hspace{1cm} (46)

where $y_i$ is the broken-line approximation for $\ln r_i(x_i)$, $\lambda_{i,t}$ is the slope of the broken line, $\mu_i, t$ is the value of $y_i$ at $x_i = k_i$, and $t$ simply designates which value of $x_i$ is under consideration. Since $\ln r_i(x_i)$ is concave, the region below $\ln r_i(x_i)$ may be represented approximately by

$$y_i \leq \lambda_{i,t}(x_i - k_i) + \mu_i, t \quad i = 1, \ldots, m; \ t = 1, \ldots, h$$  \hspace{1cm} (47)
Finally, the problem may be written in linear programming notation as

Problem: Maximize

\[ Z = \sum_{i=1}^{m} y_i \]  

subject to

\[ y_i \leq \lambda_{i,t} (x_i - k_i) + \mu_{i,t} \quad i = 1, \ldots, m; \quad t = 1, \ldots, h \]  

\[ \sum_{i=1}^{m} c_{ij} x_i \leq C_{j,\text{max}} \quad j = 1, \ldots, s \]  

\[ x_i \geq k_i \quad i = 1, \ldots, m; \quad x_i, k_i \text{ integer} \]  

\[ x_i > \frac{k_i - 1}{p_i} \quad i = 1, \ldots, m; \quad x_i, k_i \text{ integer} \]

where the slope \( \lambda_{i,t} \) and the intercept \( \mu_{i,t} \) are given by (see fig. 13)

\[ \lambda_{i,t} = \frac{\ln r_i(x_i, t) - \ln r_i(x_i, t-1)}{x_i, t - x_i, t-1} \quad i = 1, \ldots, m; \quad t = 1, \ldots, h \]  

and

\[ \mu_{i,t} = \frac{(x_i, t - k_i) \ln r_i(x_i, t) - (x_i, t-1 - k_i) \ln r_i(x_i, t)}{x_i, t - x_i, t-1} \quad i = 1, \ldots, m; \quad t = 1, \ldots, h \]  

The derivation of equation (54) is shown in appendix C as equation (C3).

The problem is now in linear form, amenable to solution by linear programming methods, if the integer restrictions on \( x_i \) \((i = 1, \ldots, m)\) are ignored temporarily. If the optimum solution happens to satisfy the integer conditions, the problem is solved. If not, the integer programming approach of Gomory (ref. 50) can be used to produce the integer solutions.
In summary, the nonlinear reliability function for the series-k-out-of-n system can be approximated by piecewise linearization. The reliability function must be concave and must be approximated by a continuous function. Under these conditions, the approximate reliability function can be maximized subject to constraints by the method of linear programming. Integer programming can then be used to satisfy the integer restrictions.

Although this procedure yields solutions which are approximate, the solutions can be made arbitrarily accurate by the appropriate choice of the number of linear functions.

Comparison of Methods

The method of Kettelle (ref. 3) provides the complete family of undominated allocations to the redundancy optimization problems. It is a dynamic programming procedure and, hence, becomes quite unwieldy for large systems subject to many constraints. It is readily adapted to computer analysis, however; Proschan and Bray (ref. 4) discuss a computer program capable of handling a maximum of three constraints, a maximum of 64 subsystems, a maximum of 10 components in each subsystem, and a maximum of 1024 entries in the dominating set at any combination of subsystems. They give guidelines in estimating the practicality of attempting to solve problems by their procedure. According to them, the only method for determining whether it is practical to solve a given problem is to attempt to find the solution. Kettelle (ref. 3) and Proschan and Bray (ref. 4) both introduced an assumption in producing their dominating sets which resulted in an error in the resulting reliability of no more than \((1 - R)^2\). That unnecessary assumption was eliminated from the development presented herein. The procedure produces exact solutions.

The partial list of undominated solutions comprising the convex hull, or "best," optimum allocations is determined by the method of Lagrange multipliers developed by Everett (ref. 5). A trial-and-error procedure is required in the selection of the Lagrange multipliers; the multipliers yielding the optimal solutions are not known beforehand but are produced in the course of the solution. This technique is most useful in determining the single best allocation satisfying the constraints rather than in generating the optimum solutions over a range of the constraints. The complexity increases substantially with the number of constraints - an obvious disadvantage for the many-constraint problem.

The same convex hull points are produced more readily by the method of balancing sensitivities. Because the procedure begins with so-called basic subsystems, the procedure is particularly well-suited to generating a set of solutions satisfying a range of constraints. The convex hull solutions are produced with much less effort than by the
method of Kettelle (ref. 3). The trial and error involved in selecting the appropriate
weighting factors is minimal compared to the trial and error involved in the method of
Lagrange multipliers. Since all combinations of weighting factors from 1, 0, ..., 0 to
0, ..., 0, 1 must be investigated to assure that none of the possible solutions is missed,
there is an obvious problem of dimensionality here for problems involving many con-
straints.

These last two procedures produce only the convex hull points and, as such, they
may miss the optimal allocations of interest. It is suggested that one of these last two
methods be used to produce an optimal allocation whose reliability or cost vector is at
the lower extreme of the range of interest. This allocation could then be used as a start-
ing point for the dynamic programming procedure, and the successively larger redun-
dancy allocations could be produced with much less effort than if the dynamic program-
ming procedure were begun with basic subsystems.

The method of Ghare and Taylor (ref. 6) is the preferred method of producing exact
solutions to problem I, the problem of maximizing reliability subject to several cost
constraints. They have written a computer program solving problems with as many as
100 subsystems, 15 constraints, and 500 redundant components. The program requires
only 5500 words of memory space on the IBM 360/50 system.

The method of integer concave programming (Mizukami, ref. 7) produces approxi-
mate solutions to problem I. The advantage of this technique is that standard linear pro-
gramming methods can be used. For integer solutions, Mizukami suggests Gomory's
(ref. 50) integer linear programming technique. The approximations introduced by this
method can be made arbitrarily good but at a cost in complexity.

**SERIES-PARALLEL SYSTEM**

Much of the literature on redundancy deals with the case where only one component
in a given subsystem is required to operate for subsystem success. Such a subsystem
is a special case of the k-out-of-n subsystem, that is, a 1-out-of-n subsystem. More
commonly, this is referred to as a parallel subsystem. When subsystems which utilize
parallel redundancy are serially connected in a system, the resulting system is called a
series-parallel system. Such a system is shown in figure 2.

The problem of redundancy optimization for series-parallel systems is not unsolved,
as evident from the literature review where most of the references cited refer to solu-
tions for just such a problem. All five of the methods used in the preceding section for
the series-k-out-of-n system were originally applied, with success, to the series-
parallel redundancy problem.

In this section, two closed-form equations are derived for the approximate optimum
distribution of redundancy under the models of problems I and II when there is only a single type of cost. The method used in the derivations is the method of Lagrange multipliers.

Closed-form equations have been derived previously by several different methods. Federowicz and Mazumdar (ref. 9) used geometric programming and Mine (ref. 8) used calculus of variations to determine the allocation of redundant components maximizing reliability for a fixed cost constraint. Their equations are identical to each other and to the one derived herein by Lagrange multipliers. Moskowitz and McLean (ref. 10) used a variational technique and Kulakov and Zagoruyko (ref. 11) used sensitivity balancing to produce two slightly different equations for minimizing a single type of cost while achieving a fixed reliability. An equation which is slightly different from those two equations is derived herein by the use of Lagrange multipliers. A comparison is made of the two equations cited from the literature and the equation derived herein.

The equations cited herein and derived herein are approximate in that they result in nonintegral numbers of components. The optimum reliability-cost curve generated by these equations is what could be achieved by using fractional components. Integer programming (ref. 21) could be used to adjust these solutions to the optimal integer solutions.

Solutions by use of the equations derived herein are compared with the exact integer solutions produced by Kettelle's dynamic programming procedure (ref. 3) on an example problem. The agreement is quite good.

**Maximizing Reliability for Fixed Cost**

The problem is to optimize the number of redundant components in each of m subsystems of a series system. Each subsystem has \( n_i \) components in parallel, at least one of which must be operative for subsystem success. The optimization is done in such a way that the reliability is maximized for a given cost.

It is assumed that the system, subsystem, or component can exist in only two possible states: success or failure. The failure of one component has no effect on the failure of any other component. System success requires operation of all subsystems; that is, the subsystems are all in series.

The appropriate equations are shown in the section Subsystems of the 1-out-of-\( n \) Type. If equation (9) is used in equation (3) and \( n_i \) is replaced by the continuous variable \( x_i \) (\( i = 1, \ldots, m \)), the problem can be stated as
Problem I: Maximize

\[ R = \prod_{i=1}^{m} \left( 1 - q_i^{x_i} \right) \]  \hspace{1cm} (55)

for a fixed cost given by equation (10) with \( n_i \) replaced by \( x_i \),

\[ C = \sum_{i=1}^{m} c_i x_i \]  \hspace{1cm} (56)

Since \( \ln R \) is a monotone increasing function of \( R \), the logarithm of \( R \) is chosen for maximization. If equation (2) is expressed as

\[ g = C - \sum_{i=1}^{m} c_i x_i = 0 \]  \hspace{1cm} (57)

problem I can be expressed as the unconstrained maximization problem,

Problem I: Maximize

\[ h = \ln R + \lambda g \]  \hspace{1cm} (58)

where \( \lambda \) is the Lagrange multiplier to be determined. Using \( R \) from equation (55) and \( g \) from equation (57) in equation (58) results in

\[ h = \sum_{i=1}^{m} \ln \left( 1 - q_i^{x_i} \right) + \lambda c - \lambda \sum_{i=1}^{m} c_i x_i \]  \hspace{1cm} (59)

The reliability is maximized subject to the constraint of equation (57) when the partial derivative of \( h \) with respect to \( x_i \) is set equal to zero,
\[ \frac{\partial h}{\partial x_i} = \frac{x_i (\ln q_i)}{1 - q_i} - \lambda c_i = 0 \quad i = 1, \ldots, m \] (60)

because the condition for a maximum, namely,

\[ \frac{\partial^2 h}{\partial x_i^2} = \frac{-(\ln q_i)^2 x_i}{(1 - q_i)^2} < 0 \] (61)

is met.

Equation (60) can be rewritten

\[ \lambda = \frac{(-\ln q_i) x_i}{c_i (1 - q_i)} \quad i = 1, \ldots, m \] (62)

The quantity \( \lambda \) in equation (62) is a Lagrange multiplier which is a constant for all sub-

systems under optimality conditions.

Equation (62) can be rewritten as

\[ \lambda' = \frac{1}{\lambda \left(1 - q_i\right)} = \frac{c_i}{(-\ln q_i) x_i} \quad i = 1, \ldots, m \] (63)

An approximation is now introduced in assuming \( \lambda' \) to be nearly constant for all sub-

systems. This is a reasonable assumption for reasonably reliable systems consisting

of a large number of subsystems because, in those cases, \( 1 - q_i \approx 1 \) for \( i = 1, \ldots, m \).

It is shown later how the approximation introduced by this assumption can be circum-

vented.

Working with the logarithm of equation (63) yields, after rearrangement,

\[ x_i \approx \frac{1}{-\ln q_i} \left[ -\ln \left(\frac{c_i}{-\ln q_i}\right) + \ln \lambda' \right] \quad i = 1, \ldots, m \] (64)
The quantity $\lambda'$ is still undetermined, so an expression for it in terms of known quantities is needed.

Putting equation (64) into the system cost equation (eq. (56)) yields

$$ C \approx \sum_{i=1}^{m} \frac{c_i}{-\ln q_i} \left[ -\ln \left( \frac{c_i}{-\ln q_i} \right) + \ln \lambda' \right] $$

(65)

This can be solved for $\ln \lambda'$ in terms of known quantities. After rearrangement,

$$ \ln \lambda' \approx \frac{C + \sum_{i=1}^{m} \left[ \left( \frac{c_i}{-\ln q_i} \right) \ln \left( \frac{c_i}{-\ln q_i} \right) \right]}{\sum_{i=1}^{m} \left( \frac{c_i}{-\ln q_i} \right)} $$

(66)

This expression can be put into equation (64) yielding

$$ x_i \approx \frac{1}{-\ln q_i} \left\{ C + \sum_{i=1}^{m} \left[ \left( \frac{c_i}{-\ln q_i} \right) \ln \left( \frac{c_i}{-\ln q_i} \right) \right] \right\} + \frac{\sum_{i=1}^{m} \left( \frac{c_i}{-\ln q_i} \right)}{m} i = 1, \ldots, m $$

(67)

The usefulness of equation (67) is in being able to determine the required number of components in each subsystem in such a way that the reliability is maximized for a given cost. The $x_i$ ($i = 1, \ldots, m$) are seen to be functions of known quantities, namely, the system cost, the component costs, the component failure probabilities, and the number of serially connected subsystems.
The effect of the approximation in equation (67) is usually negligible for large, reasonably reliable systems. The error can be eliminated, however, by the following adjustments: An \( x_1 \), say \( x_1 \), is computed from equation (67). The quantity \( \lambda \) is computed from equation (62) with \( i = 1 \) as

\[
\lambda = \frac{(-\ln q_1)q_1^{x_1}}{c_1 (1 - q_1^{x_1})} \tag{68}
\]

But rearrangement of equation (62) results in

\[
x_1 = \frac{1}{-\ln q_i} \ln \left[ 1 + \frac{1}{\lambda} \left( \frac{-\ln q_1}{c_1} \right) \right] \quad i = 1, \ldots, m \tag{69}
\]

The \( x_1 \) \((i = 1, \ldots, m)\) of equation (69) are optimal, but the reliability and cost which they yield may differ slightly from that specified. Therefore, the reliability should be computed by using \( x_1 \) from equation (69) in equation (55), and the cost should be computed by using \( x_1 \) from equation (69) in equation (56).

The \( x_1 \) \((i = 1, \ldots, m)\) determined from equation (69) are not exact, in the sense that they may not be integers. Integer programming (ref. 21) could be used to determine the optimal integer solution \( n_1 \) \((i = 1, \ldots, m)\) starting with the nonintegers \( x_1 \) \((i = 1, \ldots, m)\).

### Minimizing Cost for Fixed Reliability

The purpose here is to generate the approximate optimum arrangement of components for achieving a fixed reliability goal at minimum expenditure of resources. This one cost factor which is being minimized can be any single factor such as weight, volume, or monetary cost.

If equation (10) is used and \( n_1 \) is replaced by the continuous variable \( x_1 \) \((i = 1, \ldots, m)\), the problem can be stated as

**Problem II: Minimize**

\[
C = \sum_{i=1}^{m} c_i x_1 \tag{70}
\]
for a fixed reliability given by combining equations (9) and (1), replacing $n_i$ by $x_i$

$$R = \prod_{i=1}^{m} \left(1 - q_i^{x_i}\right) \quad (71)$$

When the logarithm of both sides of equation (71) is taken, it can be written that

$$g = \ln R - \sum_{i=1}^{m} \ln \left(1 - q_i^{x_i}\right) = 0 \quad (72)$$

Problem II can be expressed, by introducing a Lagrange multiplier $\lambda$, as the unconstrained minimization problem,

Problem II: Minimize

$$h = C + \lambda g \quad (73)$$

Using $C$ from equation (70) and $g$ from equation (72) in equation (73) results in

$$h = \sum_{i=1}^{m} c_ix_i + \lambda \ln R - \lambda \sum_{i=1}^{m} \ln \left(1 - q_i^{x_i}\right) \quad (74)$$

The cost is minimized subject to the constraint of equation (72) when the partial derivative of $h$ with respect to $x_i$ is set equal to zero

$$\frac{\partial h}{\partial x_i} = c_i + \frac{\lambda q_i^{x_i} (\ln q_i)}{1 - q_i^{x_i}} = 0 \quad i = 1, \ldots, m \quad (75)$$
because the condition for a minimum is met, namely,

\[
\frac{\partial^2 \mathcal{H}}{\partial x_i^2} = \lambda \left( \frac{\ln q_i}{q_i} \right)^2 \frac{x_i}{(1 - q_i)^2} > 0
\]  

(76)

Equation (75) can be rewritten

\[
X = x_i
\]

in \( q_i \) for \( i = 1, \ldots, m \)  

(77)

The quantity \( X \) in equation (77) is a Lagrange multiplier which is a constant for all subsystems under optimality conditions. Equation (77) can be rewritten as

\[
\lambda = \frac{c_i (1 - q_i)}{(-\ln q_i) q_i x_i} \quad i = 1, \ldots, m
\]  

(78)

An approximation is now introduced in assuming \( \lambda' \) to be nearly constant for all subsystems. This is a reasonable assumption for reasonably reliable systems because, in those cases, \( 1 - q_i \approx 1 \) for \( i = 1, \ldots, m \). The quantity \( \lambda \) is constant under optimality conditions. It is shown later how the approximation introduced by this assumption can be circumvented.

Working with the logarithm of equation (78) yields the equation for \( x_i \) as

\[
x_i \approx \frac{1}{-\ln q_i} \left[ -\ln \left( \frac{c_i}{-\ln q_i} \right) + \ln \lambda' \right] \quad i = 1, \ldots, m
\]  

(79)

The quantity \( \lambda' \) is still unknown so an expression for it in terms of known quantities is needed.
The system reliability of equation (71) can be approximated by

\[ R \approx 1 - \sum_{i=1}^{m} \frac{x_i}{q_i} \]  

(80)

because the crossproduct terms are small for reasonably reliable systems. In fact, the error in \( R \) as computed from equation (80) is at most \((1 - R)^2\). If \( q_i \) from equation (78) is substituted into equation (80),

\[ R \approx 1 - \frac{1}{\lambda'} \sum_{i=1}^{m} \left( \frac{c_i}{-\ln q_i} \right) \]  

(81)

Rearranging equation (81) and taking logarithms results in

\[ \ln \lambda' \approx -\ln (1 - R) + \ln \sum_{i=1}^{m} \left( \frac{c_i}{-\ln q_i} \right) \]  

(82)

This expression for \( \ln \lambda' \) can now be used in equation (79), giving

\[ x_i \approx \frac{1}{-\ln q_i} \left\{ -\ln (1 - R) - \ln \left[ \frac{c_i}{-\ln q_i} \right] \right\} \]  

\[ i = 1, \ldots, m \]  

(83)

The effect of the approximations introduced by assuming \( \lambda' \) constant and by equation (80) upon the results of equation (83) is usually negligible for reasonably reliable systems. The error can be eliminated, however, by the following adjustments: An \( x_i \), say \( x_1 \), is computed from equation (83). The quantity \( \lambda \) is computed from equation (77) with \( i = 1 \) as
\[ \lambda = \frac{c_i (1 - q_i^x)}{(-\ln q_i) q_i^x} \]  

But rearrangement of equation (77) results in

\[ x_i = \frac{1}{-\ln q_i} \ln \left[ 1 + \lambda \left( \frac{-\ln q_i}{c_i} \right) \right] \quad i = 1, \ldots, m \]  

The \( x_i \) \((i = 1, \ldots, m)\) of equation (85) are optimal, but the reliability and cost which they yield may differ slightly from that specified. Therefore, the reliability should be computed by using \( x_i \) from equation (85) in equation (71), and the cost should be computed by using \( x_i \) from equation (85) in equation (70).

The \( x_i \) \((i = 1, \ldots, m)\) determined from equation (85) are not exact, in the sense that they may be nonintegers. Integer programming (ref. 21) could be used to determine the optimal integer solution \( n_i \) \((i = 1, \ldots, m)\), starting with the nonintegers \( x_i \) \((i = 1, \ldots, m)\).

Moskowitz and McLean (ref. 10) used a variational technique to arrive at an equation analogous to equation (83). Their equation is

\[ x_i \approx \frac{1}{\ln q_i} \ln \left[ 1 - R \left( \frac{c_i}{-\ln q_i} \right) \sum_{i=1}^{m} \left( \frac{c_i}{-\ln q_i} \right) \right] \quad i = 1, \ldots, m \]  

Kulakov and Zagoruyko (ref. 11) used balancing of sensitivities to get
\[ x_i \approx \frac{1}{\ln q_i} \ln \left( \frac{1}{R + (1 - R) \frac{\sum_{i=1}^{m} \frac{c_i}{\ln q_i}}{\sum_{i=1}^{m} \frac{c_i}{\ln q_i}}} \right) \]

\[ i = 1, \ldots, m \]  

A comparison of equations (83), (86), and (87) is shown in the next section, where an example problem is solved.

**Example Problem**

An illustrative example of the use of the closed-form approximate solutions for \( x_i \ (i = 1, \ldots, m) \) is the problem solved by Kettelle (ref. 3). The system consists of four subsystems in series, each subsystem requiring operation of one of the possibly more than one parallel components. Component costs and success probabilities are shown in table XI. The system is also illustrated in figure 14.

**TABLE XI. - COMPONENT PARAMETERS FOR EXAMPLE PROBLEM**

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component cost, ( c_i )</td>
<td>1.2</td>
<td>2.3</td>
<td>3.4</td>
<td>4.5</td>
</tr>
<tr>
<td>Component success probability, ( p_i )</td>
<td>0.8</td>
<td>0.7</td>
<td>0.75</td>
<td>0.85</td>
</tr>
</tbody>
</table>
The problem is to determine the optimum values of $x_i$ (i = 1, 2, 3, 4) such that $C = 30, 45, and 60$. The algorithm of Kettelle (ref. 3) was used to get the complete set of integer solutions for $0.90 \leq R \leq 0.999$, which is shown in table XII. Those solutions generated by Everett's generalized Lagrange multiplier method (ref. 5) are noted in the table.

The solutions shown in table XIII are the ones giving $C$ closest to the specified values. The results of using equation (67) are shown in the same table for comparison. The approximation is very good. The complete set of solutions obtained by Kettelle's procedure (ref. 3) is shown in figure 15. Equation (67) is plotted to show the good agreement with the exact solutions.

Similar agreement is evidenced when comparing the results of equation (83) with the exact solutions. The problem is to determine the optimum values of $x_i$ (i = 1, 2, 3, 4) such that $R = 0.90, 0.99, and 0.999$. The members of the complete family of integer solutions having reliabilities nearest these values are chosen for comparison. The results are shown in table XIV.

Table XV shows the comparisons of equations (83), (86), and (87) for the example problem. There is seen to be very little difference in the results obtained with the three equations. This agreement is better for the higher reliabilities, where the effect of the approximations used in deriving the equations is less pronounced. In general, equation (86), derived in reference 10, is preferred because it produces an allocation having reliability closest to that specified in the first column of table XV.
TABLE XII. - COMPLETE FAMILY OF SOLUTIONS TO EXAMPLE PROBLEM GENERATED BY KETTELLE'S DYNAMIC PROGRAMMING PROCEDURE$^a$ FOR $0.90 \leq R \leq 0.999$

<table>
<thead>
<tr>
<th>$n_1$</th>
<th>$n_2$</th>
<th>$n_3$</th>
<th>$n_4$</th>
<th>Reliability, $R$</th>
<th>Cost, $C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0.9107</td>
<td>28.60</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td>b.9288</td>
<td>29.70</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td>.9347</td>
<td>30.90</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>b.9468</td>
<td>32.00</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td>b.9529</td>
<td>33.20</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>9541</td>
<td>34.40</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td>.9581</td>
<td>35.40</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td>.9653</td>
<td>36.50</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td>b.9715</td>
<td>37.70</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td>9728</td>
<td>38.90</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td>.9768</td>
<td>39.90</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>3</td>
<td></td>
<td>.9771</td>
<td>40.00</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td>b.9831</td>
<td>41.10</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td>.9844</td>
<td>42.30</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td>b.9887</td>
<td>43.40</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td>b.9900</td>
<td>44.60</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
<td>9904</td>
<td>45.70</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td></td>
<td>.9916</td>
<td>46.80</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>4</td>
<td></td>
<td>.9917</td>
<td>46.90</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>b.9929</td>
<td>48.00</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
<td>.9933</td>
<td>49.10</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td>b.9946</td>
<td>50.30</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
<td>.9949</td>
<td>51.50</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td>.9958</td>
<td>52.50</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
<td>.9962</td>
<td>53.60</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td>b.9975</td>
<td>54.80</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
<td></td>
<td>.9977</td>
<td>56.00</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td>b.9980</td>
<td>57.10</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>6</td>
<td></td>
<td>9982</td>
<td>58.20</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>5</td>
<td></td>
<td>.9982</td>
<td>58.30</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
<td>.9985</td>
<td>59.40</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>6</td>
<td></td>
<td>b.9987</td>
<td>60.50</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>6</td>
<td></td>
<td>b.9990</td>
<td>61.70</td>
</tr>
</tbody>
</table>

$a$Ref. 3
$b$Lagrange solutions.
TABLE XIII - COMPARISON OF EQUATION (67) WITH OPTIMAL INTEGER SOLUTIONS - EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>Cost constraint</th>
<th>Method</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>Reliability, R</th>
<th>Cost, C</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Optimum integer</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0.9288</td>
<td>29.70</td>
</tr>
<tr>
<td></td>
<td>Equation (67)</td>
<td>3.112</td>
<td>3.378</td>
<td>2.754</td>
<td>2.030</td>
<td>0.9346</td>
<td>30.00</td>
</tr>
<tr>
<td>45</td>
<td>Optimum integer</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>0.9900</td>
<td>44.60</td>
</tr>
<tr>
<td></td>
<td>Equation (67)</td>
<td>4.358</td>
<td>5.044</td>
<td>4.200</td>
<td>3.087</td>
<td>0.9910</td>
<td>45.00</td>
</tr>
<tr>
<td>60</td>
<td>Optimum integer</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>0.9985</td>
<td>59.40</td>
</tr>
<tr>
<td></td>
<td>Equation (67)</td>
<td>5.603</td>
<td>6.709</td>
<td>5.647</td>
<td>4.144</td>
<td>0.9988</td>
<td>60.00</td>
</tr>
</tbody>
</table>

Figure 15 - Comparison of exact and approximate solutions - example problem
It has been shown in this section that closed-form equations for approximating optimum redundancy in the series-parallel system can be derived by the use of Lagrange multipliers. These equations are used on the example problem of Kettelle (ref. 3) and result in good agreement. Only one of the equations is new, however; and it is shown to be no better than the one developed by Moskowitz and McLean (ref. 10) using a variational technique. The reason for presenting the two derivations is that for the first time equations for both maximum reliability for fixed cost and minimum cost for fixed reliability have been derived by the same technique, namely, the technique of Lagrange multipliers. In addition, adjustment schemes are given to circumvent the approximations inherent in the development of the equations. The results are still approximate in that
they, in general, involve noninteger numbers of components. They do, however, form a convenient basis for exact solutions by the method of integer programming.

In the search for optimal solutions, a reasonable question concerns the maximum reliability achievable for a fixed cost or the minimum cost achievable for a fixed reliability. The equations presented in this section allow rapid determination of those optimum conditions theoretically achievable with noninteger numbers of components. In this way a systems designer can determine whether a given design (arrangement of components) is near optimum and, hence, whether much could be gained by searching for the unique optimum design.

CONCLUSIONS

This analysis illustrates that the optimization of redundancy in the series-k-out-of-n system can be accomplished by appropriate use of some of the techniques used to solve similar problems for series-parallel systems. The two types of optimization problems considered herein are

1. Maximizing system reliability for the series-k-out-of-n system subject to multiple cost constraints
2. Minimizing some function of the multiple cost factors for the series-k-out-of-n system subject to maintaining a minimum acceptable level of system reliability

Five methods are presented for solving one or both of these problems without resorting to the inordinate expenditure of effort required in examining all possible combinations of components. These five methods are

1. Kettelle's dynamic programming procedure as modified by Proschan and Bray to include multiple constraints
2. Everett's generalized Lagrange multiplier technique
3. The method of selecting consecutively for redundancy those subsystems which increase the reliability per unit of weighted costs by the greatest amount, which was used, for example, by Barlow and Proschan
4. Ghare and Taylor's formulation of the problem as a multidimensional knapsack problem using a branch-and-bound procedure
5. Mizukami's combined concave and integer programming procedure

These five methods are compared and an example problem is solved by the first three methods.

Closed-form equations are derived by the use of Lagrange multipliers for the series-parallel system optimization problems of

1. Maximizing system reliability while maintaining a fixed single cost constraint
2. Minimizing a single cost while maintaining a fixed system reliability
The equations are no better than those already available, but they are derived by a method (Lagrange multipliers) that differs from the methods in the literature. The equations are used to solve an example problem, and on this basis a comparison is made with the equations cited from the literature.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, October 17, 1972,
502-05.
APPENDIX A

CONCAVITY CONDITIONS

The conditions for concavity of the logarithm of reliability for a series-k-out-of-n system are developed herein. By equation (20)

\[ \ln R = \sum_{i=1}^{m} \ln r_i \]  

(20)

Since the logarithm of concave functions is concave and since the sum of concave functions is concave, the conditions need only be established for which \( r_i \) is a concave function of \( n_i \) \((i = 1, \ldots, m)\). For k-out-of-n subsystems, \( r_i \) is given by equation (8) as

\[ r_i(n_i) = p_i \sum_{x=0}^{n_i-k_i} \binom{k_i-1+x}{x} q_i^x \]  

\( i = 1, \ldots, m \)  

(8)

Concavity is satisfied by a negative value of

\[ \Delta^2 r_i(n_i) = \Delta \left[ \Delta r_i(n_i) \right] \]  

\( i = 1, \ldots, m \)  

(A1)

where, in general,

\[ \Delta f(n) = f(n+1) - f(n) \]  

(A2)

Expanding equation (A1) results in

\[ \Delta^2 r_i(n_i) = \Delta \left[ r_i(n_i + 1) - r_i(n_i) \right] \]  

\( i = 1, \ldots, m \)  

(A3)

or

\[ \Delta^2 r_i(n_i) = r_i(n_i + 2) - 2r_i(n_i + 1) + r_i(n_i) \]  

\( i = 1, \ldots, m \)  

(A4)

Now using equation (8) in equation (A4) and simplifying yields...
\[
\Delta^2 r_i(n_i) = p_i^{k_i} q_i^{n_i-k_i+2} \binom{n_i + 1}{k_i} \left[ 1 - \frac{n_i - k_i + 2}{q_i(n_i + 1)} \right] \quad i = 1, \ldots, m
\] (A5)

The concavity condition is satisfied when \( \Delta^2 r(n_i) \) is negative, that is, when

\[
n_i > \frac{k_i - 1}{p_i} - 1 \quad i = 1, \ldots, m
\] (A6)

In many cases this restriction results in smaller \( n_i \) than the necessary condition

\[
n_i \geq k_i \quad i = 1, \ldots, m
\] (A7)
APPENDIX B

CORRESPONDENCE BETWEEN PROBLEM I AND THE UNCONSTRAINED MAXIMIZATION PROBLEM

The proof that the solution to Everett's unconstrained maximization problem (ref. 5) is the solution to the constrained optimization problem is given herein. Everett's unconstrained problem is given by equation (21) as maximizing

\[ H(n_i) = \ln r_i(n_i) - \sum_{j=1}^{s} \lambda_j c_{ij} n_i \quad i = 1, \ldots, m \]  \hspace{1cm} (21)

The constrained optimization problem is given by equation (20) as problem I: Maximize

\[ \ln R = \sum_{i=1}^{m} \ln r_i \] \hspace{1cm} (20)

subject to

\[ \sum_{i=1}^{m} c_{ij} n_i \leq C_{j, \text{max}} \quad j = 1, \ldots, s \] \hspace{1cm} (4)

Let \( \tilde{\lambda} = (\lambda_1, \ldots, \lambda_s) \) be the set of nonnegative real numbers (not all zero) which yields \( n_i \) \( (i = 1, \ldots, m) \). Let \( n_i(\tilde{\lambda}) \) \( (i = 1, \ldots, m) \) designate the value of \( n_i \) which maximizes \( H(n_i) \) for the chosen vector \( \tilde{\lambda} \). Since any other value of \( n_i \) results in no greater value of \( H \),

\[ H[n_i(\tilde{\lambda})] \geq H(n_i) \quad i = 1, \ldots, m \]  \hspace{1cm} (B1)

or

\[ \ln r_i[n_i(\tilde{\lambda})] - \sum_{j=1}^{s} \lambda_j c_{ij} n_i(\tilde{\lambda}) \geq \ln r_i(n_i) - \sum_{j=1}^{s} \lambda_j c_{ij} n_i \quad i = 1, \ldots, m \]  \hspace{1cm} (B2)
Rearranging equation (B2) yields

\[ \ln r_i \left[ n_i(\lambda) \right] \geq \ln r_i(n_i) + \sum_{j=1}^{s} \lambda_j \left[ c_{ij} n_i(\lambda) - c_{ij} n_i \right] \quad i = 1, \ldots, m \]  

(B3)

Summing equation (B3) over all subsystems results in

\[ \sum_{i=1}^{m} \ln r_i \left[ n_i(\lambda) \right] \geq \sum_{i=1}^{m} \ln r_i(n_i) + \sum_{j=1}^{s} \lambda_j \left\{ \sum_{i=1}^{m} \left[ c_{ij} n_i(\lambda) - c_{ij} n_i \right] \right\} \]  

(B4)

But the costs of any allocation cannot exceed \( C_{j,\text{max}} = \sum_{i=1}^{m} c_{ij} n_i(\lambda) \), where \( j = 1, \ldots, s \); so

\[ \sum_{i=1}^{m} \left[ c_{ij} n_i(\lambda) - c_{ij} n_i \right] \geq 0 \quad j = 1, \ldots, s \]  

(B5)

Since all \( \lambda_j \) (\( j = 1, \ldots, s \)) are nonnegative, equation (B4) can be written

\[ \sum_{i=1}^{m} \ln r_i \left[ n_i(\lambda) \right] \geq \sum_{i=1}^{m} \ln r_i(n_i) \quad i = 1, \ldots, m \]  

(B6)

Therefore, the \( n_i(\lambda) \) (\( i = 1, \ldots, m \)) which maximizes equation (21) also is the value of \( n_i \) which maximizes equation (20) subject to the cost constraint

\[ C_{j,\text{max}} = C_{j,\left[ n_i(\lambda) \right]} \]  

(B7)

And there is a one-to-one correspondence between problem I and the unconstrained maximization problem solvable by Everett's technique (ref. 5).
APPENDIX C

DEVELOPMENT OF INTERCEPT FOR CONCAVE PROGRAMMING

The intercept for the concave programming procedure is developed herein. As shown in figure 13, the slope of the line \( y_i \) can be written

\[
\lambda_{i,t} = \frac{\ln r_i(x_{i,t}) - \mu_{i,t}}{x_{i,t} - k_i} \quad i = 1, \ldots, m; \ t = 1, \ldots, h
\]  

But \( \lambda_{i,t} \) is also given by

\[
\lambda_{i,t} = \frac{\ln r_i(x_{i,t}) - \ln r_i(x_{i,t-1})}{x_{i,t} - x_{i,t-1}} \quad i = 1, \ldots, m; \ t = 1, \ldots, h
\]  

Setting equation (C1) equal to equation (C2) and solving for \( \mu_{i,t} \) results in

\[
\mu_{i,t} = \frac{(x_{i,t} - k_i) \ln r_i(x_{i,t-1}) - (x_{i,t-1} - k_i) \ln r_i(x_{i,t})}{x_{i,t} - x_{i,t-1}} \quad i = 1, \ldots, m; \ t = 1, \ldots, h
\]  

68
REFERENCES


"The aeronautical and space activities of the United States shall be conducted so as to contribute ... to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."
—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546