OPTIMIZATION OF SYSTEM RELIABILITY OF LIFE SUPPORT SYSTEMS USING AN INTEGER PROGRAMMING

by 264

SUDESH KUMAR

B.Sc. (Engg.) Mechanical
Ranchi University, India, 1966
M.E. (Production Engineering)
Roorkee Univ., India, 1968

A Master's Report

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

Kansas State University

Manhattan, Kansas

1970

Approved by

Ching-Lai Hwang

Major Professor

TABLE OF CONTENTS

1 INTRODUCTION

The purpose of this report is twofold: first, to summarize the introductory theory of reliability of a composite system consisting of multistages and multicomponents of various types, secondly, to present results of the optimization study on the reliability of life support systems by means of applying an integer programming technique, namely, zero-one integer programming.

A life support system is defined as a system that creates, maintains, and controls an environment adequate to permit the personnel or life operating in the system to function at a maximum efficiency for extended periods of time. The space environment is not compatible for life. The vehicle containing the occupant must provide a structurally adequate, hermetically sealed cabin encompassing an atmosphere adequate for the needs of the astronauts. The importance of the life support system demands a high reliability of the system for safe and flawless completion of the project. Each subsystem of the life support system must work properly.

Reliability is the probability of successful operation. Bazovsky [3] defines it in the following way, "Reliability is the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered." When equipment works well and works whenever it is called upon to do the job for which it was designed, it is said to be reliable. The measure of the reliability of equipment is determined by the frequency of failures occurring during the operating time of the equipment. If there are no failures, the equipment is one hundred percent reliable, if the failure frequency is low, the reliability of the equipment is usually still acceptable; however, if the failure frequency is high, the equipment is unreliable.

The reliability of the system can always be increased by using a large number of supporting units, but usually there are restrictions on the size of the reserve that may be provided. A large reserve involves increased cost, volume, weight etc.

A system can be made reliable by adding supporting components as spares. This procedure adds weight and volume to the system, and weight and volume of the space-craft are factors which must be kept as low as possible. Addition of a component increases the weight and volume but decreases the unreliability of the system; so here is an optimization problem. The reliability level must be achieved by adding a minimum to weight and/or volume to the system. Or the reliability must be maximized by utilizing all the weight lift-up capacity of the spacecraft. This is the problem considered in this report.

An introduction to the theory of reliability is presented in the second chapter. The first section describes the types of failure. It is followed by the presentation of the quantitative characteristics of reliability and the dependence of reliability on time. The non-replacement test for reliability estimation is introduced. The need of achieving high reliability and the means for improving the reliability are discussed. The expressions for reliability of systems with components or subsystems in series, parallel, stand-by, and systems with spares are derived. At the end of the chapter, the problem considered in this report is described.

The third chapter covers integer programming methods and indicates their application to optimization of system reliability. The methods of solution of the integer programming problems are described briefly, while an implicit enumeration technique is presented in detail. A computer program based upon this algorithm was used to obtain the optimal solutions for the problems in this report. Also included is the review of the literature on the application of the integer programming to the optimization of system reliability problems.

The fourth chapter presents optimum results of some circuits of life support systems. The results obtained are compared with the designs suggested by North American Rockwell Corp. [29]. A comparison indicates that present system formulation resulted in a better arrangement of spares, since the weight of the spares was less than that presented in ref. [29] to achieve the same level of reliability.

2 RELIABILITY

An introduction to the theory of reliability is presented in this chapter. The first section describes the types of failure. It is followed by the presentation of the quantitative characteristics of reliability and the dependence of reliability on time. The non-replacement test for reliability estimation is introduced. The need of achieving high reliability and the means for improving the reliability are discussed. The expressions for reliability of systems with components or subsystems in series, parallel, stand-by, and systems with spares are derived. At the end of this chapter the problem of applying integer programming to determine the optimum reliability of a life support system is introduced.

2.1 CONCEPTS OF RELIABILITY [3]

Reliability is the probability of the successful operation of a system. Bazovsky [3] defines it in the following way, "Reliability is the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered." When a piece of equipment functions whenever it is called upon to do the job for which it was designed, it is said to be reliable. The measure of the reliability is determined by the frequency of failures occurring during the operating time of the equipment. If there are no failures, the equipment is one hundred percent reliable, if the failure frequency is low, the reliability of the equipment is usually still acceptable; however, if the failure frequency is high, the equipment is unreliable.

Well designed, well manufactured, thoroughly tested, and properly maintained equipment should never fail in operation. However, practically, we can not completely eliminate the occurrence of failures even though the equipment is the well designed, and manufactured, and properly maintained. There exist the following three characteristic types of failures;

1. Early Failures - early failure of a component occurs early in the life of the component. In most cases this kind of failure is a result of poor manufacturing and poor quality control during the production processes.

2. Wearout Failures - these are the symptom of component aging. The age at which wearout occurs differs widely with components, ranging from a few minutes to years. These failures are caused by the wearing of the parts if the equipment is not properly maintained or not maintained at all.

3. Chance Failures - these are caused by sudden stress accumulations beyond the design strength of the component. Chance failures occur at random intervals, irregularly and unexpectedly. No one can predict when the chance failures will occur; however, they obey certain rules of collective behaviour so that the frequency of their occurrence during sufficiently long periods is approximately constant.

Reliability is the probability that no failure of any kind will occur in a given time interval of operation.

2.2 QUANTITATIVE CHARACTERISTICS OF RELIABILITY [36b]

Reliability as mentioned previously is the "probability of failure free operation of equipment"; thus it is the probability of the event that under definite operating conditions and within specified limits of operating duration, no failure will occur. A typical probability function, $P(t)$, of a failure free operation of a system as a function of time is shown in Fig. 1.

From the definition, the following characteristics of reliability can be listed:

1. It is a decreasing function of time.
2. $0 \leq P(t) \leq 1$
3. $P(0) = 1$ and $P(\propto) = 0$.

EXPONENTIAL CHARACTERISTICS [36b]

Consider a population of N items with the same failure - time distribution. The probability of success is $R(t)$. The items fails independently with probability of failure given by $F(t) = 1 - R(t)$. The number of units surviving at time t, $N(t)$, is a random number having $p = R(t)$. The expected valve of the $N(t)$, which is binomially distributed is

$$n(t) \equiv E[N(t)] = N\, R(t) \tag{2.1}$$

which gives the reliability

$$R(t) = \frac{n(t)}{N} \tag{2.2}$$

and

$$F(t) = 1 - R(t) = \frac{N - n(t)}{N} \tag{2.3}$$

The failure density function, $f(t)$, is defined as

$$f(t) = \frac{dF(t)}{dt} \tag{2.4}$$

Using equation (2.3), we have

$$f(t) = -\frac{1}{N}\frac{dn(t)}{dt}$$

$$\equiv \lim_{\Delta t \to 0} \frac{n(t) - n(t + \Delta t)}{N\, \Delta t} \tag{2.5}$$

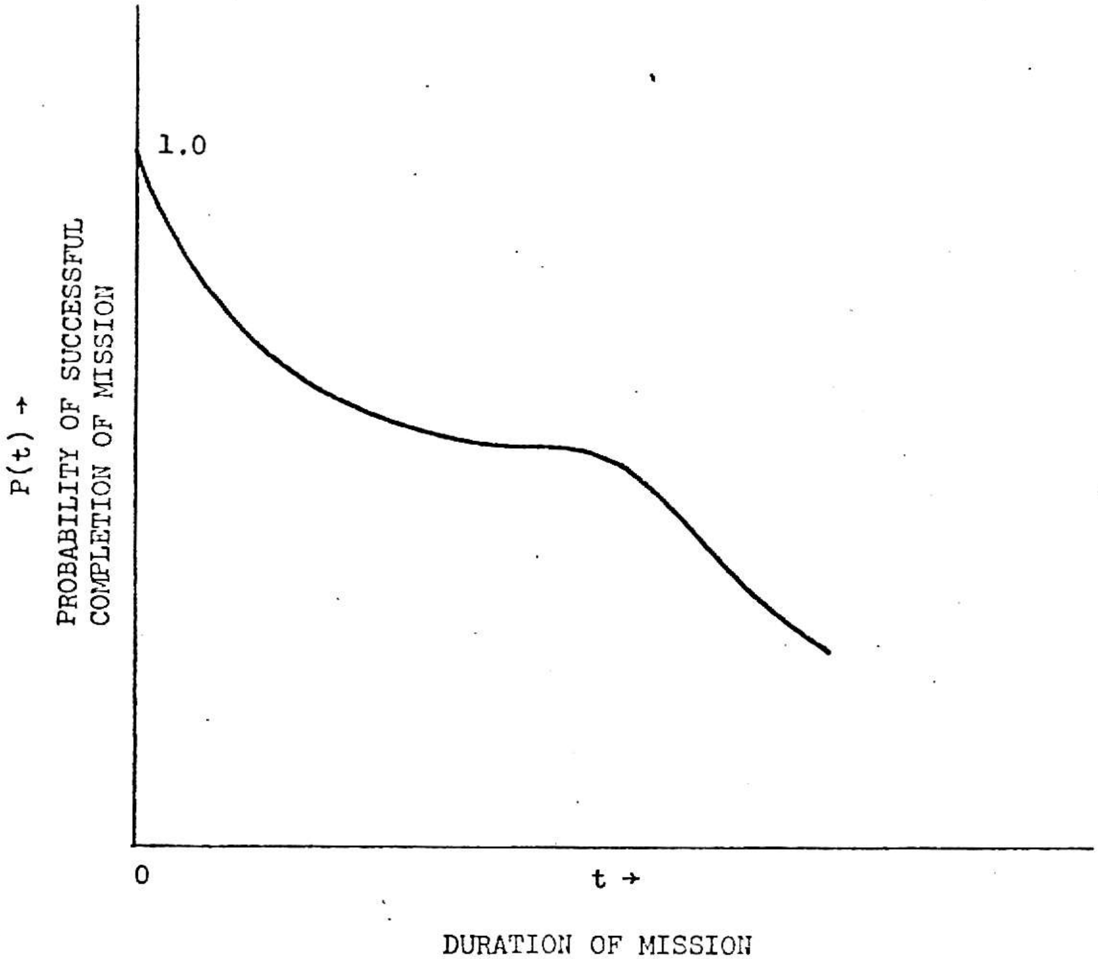

Fig. 1 Reliability - Time Relation

Defining hazard rate, $\lambda(t)$ as

$$\lambda(t) \equiv \lim_{\Delta t \to 0} \frac{n(t) - n(t + \Delta t)}{n(t) \quad \Delta(t)} \quad . \tag{2.6}$$

This is also called the failure intensity. The time dependent nature of failure intensity for some components is shown in Fig. 2. From equations (2.5) and (2.6) following relation can be obtained

$$\lambda(t) = \frac{N\ f(t)}{n(t)} = \frac{f(t)}{R(t)} \ . \tag{2.7}$$

From equations (2.4), (2.5) and (2.7) we obtain

$$\begin{aligned} \lambda(t) &= \frac{d\ F(t)}{dt} \cdot \frac{N}{n(t)} \\ &= - \frac{1}{N} \frac{d\ n(t)}{dt} \frac{N}{n(t)} \\ &= - \frac{d}{dt} \log n(t) \, . \end{aligned} \tag{2.8}$$

The solution to equation (2.8) becomes

$$n(t) = A_0 \ e^{-\int_0^t \lambda(t)\ dt}$$

where A_0 is constant of integration. Substituting the following initial condition in the above equation.

$$n(0) = A_0 = N$$

Then we have

$$n(t) = N \ e^{-\int_0^t \lambda(t)\ dt}$$

The expression for reliability, equation (2.2), then becomes

$$R(t) = e^{-\int_0^t \lambda(t)\ dt}$$

when failure intensity is constant, λ, the reliability is given by

$$R(t) = e^{-\lambda t}$$

Figure 3 shows this exponential function which represents the probability of failure free operation, i.e., the reliability of the equipment.

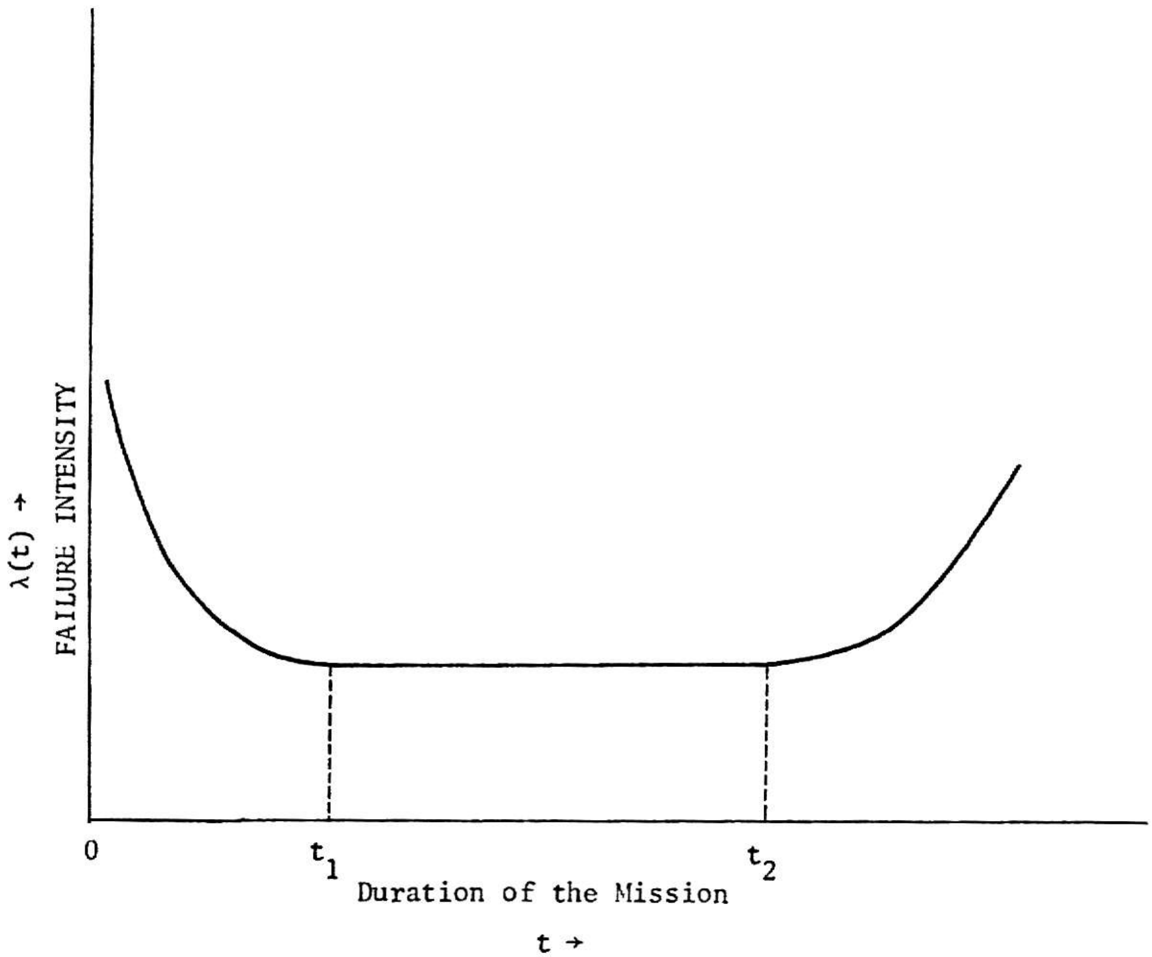

Fig. 2 Nature of Failure Intensity

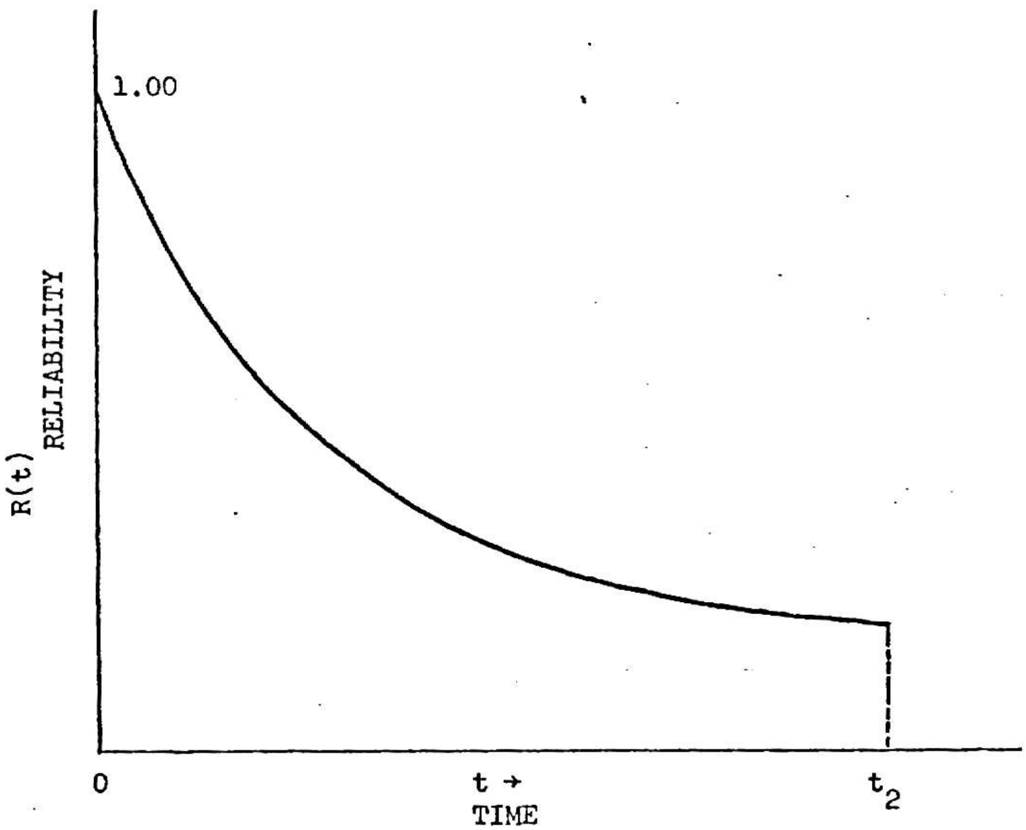

Fig. 3 Exponential - Reliability Function (When λ is Constant)

The exponential function expressed in equation (2.4) for probability of failure free operation is valid only for the interval between t_1 and t_2 in Fig. 2. This is the interval when the failure intensity is almost constant. The probability of failure is very small in this interval. One should not use this equation to predict the reliability of the equipment for any period beyond its useful life given by t_2.

2.3 ESTIMATION OF THE FAILURE CHARACTERISTICS OF COMPONENTS [3,7]

To perform reliability calculations and to determine optimal spare-parts provision, the distribution of chance failures and wearout failures must be known separately. One should know as closely as possible the true value of the mean time between failures, m, the true value of the mean wearout life, M, and the standard deviation of the wearout failures, σ. The parameters M and σ enable one to determine suitable replacement and overhaul schedules, and then the parameter, m, may be used to calculate the probability of no chance failures in the period between replacement and overhauls.

To determine the chance failure characteristics, one should, therefore, determine the parameter, m, the mean time between failures. It is estimated by measuring the times to failure of n specimens, t_i, $i = 1, 2, \ldots, n$, then obtaining the mean time of these n observations. But these failures may however include some wearout failures, which will contaminate the calculations. To exclude these wearout failures the duration of the test must be so limited as to be reasonably certain that no wearout failure will occur during the test period.

Epstein [7] has established that the maximum likelihood estimate of m can be made by the "Non-replacement Test" method. When n components are originally placed under test and r of them fail at times $t_1, t_2, \ldots, t_r$ which are counted from the beginning of the test, and the test is discontinued at the time, t_r, of the occurence of the r-th failure, so that the (n-r) components are still unfailed at the end of the test. The optimum estimate for the mean time to failure is given by the following formula.

$$\hat{m} = \frac{t_1 + t_2 + \ldots + t_r + (n-r)t_r}{r}$$

$$= \frac{\sum_{i=1}^{r} t_i + (n-r)t_r}{r}$$

To avoid component wearout failures during a test, the test truncation time, t_r, should be chosen to be as short as possible in comparison with the wearout time of the components.

The sample size, n, for the test can be calculated. When the available test time for a non-replacement experiment is t hours and the expected mean failure rate is λ, and the mean time between failures, m, has to be measured with a precision corresponding to r chance failures, the number of specimens is obtained by [3]

$$n = \frac{r}{1 - \exp(-\lambda t)} = \frac{r}{Q(t)}$$

where Q(t) is the expected unreliability of the component for an operation time t.

To find the mean and standard deviation of wearout failures, it is essential to start the test with new components or with components which have passed a burn-in procedure of a known number of hours to weed out early failures. The sample is then submitted to test its operation under simulated environmental conditions, and the test continues until either all or at least a substantial percentage of these fails. The lives of these failed components are measured separately. From this information the mean wearout life and its standard deviation can be computed.

2.4 THE NEED OF HIGHER RELIABILITY

An unreliable or less-reliable equipment has more break down time, higher cost of operations and lower productivity compared with that of a more reliable equipment. The importance of problems which are being solved by the modern automatic systems and the associated high costs of these systems require that their reliability be high. Of course the technology and methods of today enable one to design and fabricate automatic equipment and systems whose reliability is as high as desired. However, such a system may be very heavy, and its size and cost may be prohibitively large. Hence the systems must be designed for feasibility as well as for optimal characteristics. The systems must have the optimal reliability, established by proper criteria of requirements. The criteria may be any characteristic, for example cost, size, effectiveness, accuracy, and life etc. In a typical industrial situation, the common criteria are cost and productivity, in defence situations, combat effectiveness may be the most important criterion and in scientific experimentation such as space flights, etc., these may be dependability and accuracy.

Highly reliable equipment has a smaller number of failures than equipment which has a low reliability. This factor decreases the forced down-time of the equipment and the necessary number of spare parts and assemblies. When the reliability of any equipment increases, the cost of designing and manufacturing increases, but the cost of operation decreases; this situation justifies the need of higher reliability.

2.5 METHODS FOR IMPROVING SYSTEM RELIABILITY

It is, therefore, desirable to have the system reliability as high as possible within feasibility constraints. The following are ways to increase system reliability.

1. Redesigning either the system as a whole or the system components for better reliability.
2. Providing redundant systems in parallel.
3. Providing redundant units for weak components in the system.
4. Providing stand-by systems.
5. Providing stand-by components with failure sensing and switching devices.
6. Supporting the system with spare parts, which can be used to replace components that fail.

Any one of these methods can be used, or a combination of these can be utilized depending upon the requirements of the system and the failure characteristics of the components. The expressions for the reliability of special system configurations are as follows.

(a) Reliability of System with Parallel Redundancy

When the additional units to improve the system reliability are placed in such a way that all the units will be working at the same time and the failure of one or more units does not hinder the operations, the configuration is called as parallel redundant system.

For example, in a four engine aeroplane, if two engines are sufficient to pull it, the other two are simultaneously working so that each is working at 50% load. The failure of one or two causes the shift of the load to others, and the aeroplane can still fly.

The systems are working at underload but even then the characteristics are taken to be unchanged. The aging of the system is one of the factors which should be considered; the units which are provided to support the system are being used and thus the maximum duration of successful operation is equal to the life of the unit which fails last. Whereas in the stand by system, the supporting units are not working until the first unit fails and thus these are preserved, so they do not age, and the duration of operation is equal to the sum of the lives of the individual units.

For an N-stage series system, in which if one stage fails the system fails, the system reliability is determined by

$$R_s = \prod_{i=1}^{N} R_i$$

where R_i is the reliability of the ith stage in the series system. In some situations, several less reliable components are supported by some redundant units, making the total number of components of that type m_i. Thus if the system has $(m_i - 1)$ parallel redundancies at each stage, then the system reliability is

$$R_s = \prod_{i=1}^{N} (1 - (1 - R_i)^{m_i})$$

(b) Reliability of System with Stand-by Units [3]

Stand-by arrangements require failure sensing and switching devices to put the next unit into operation (Fig. 4).

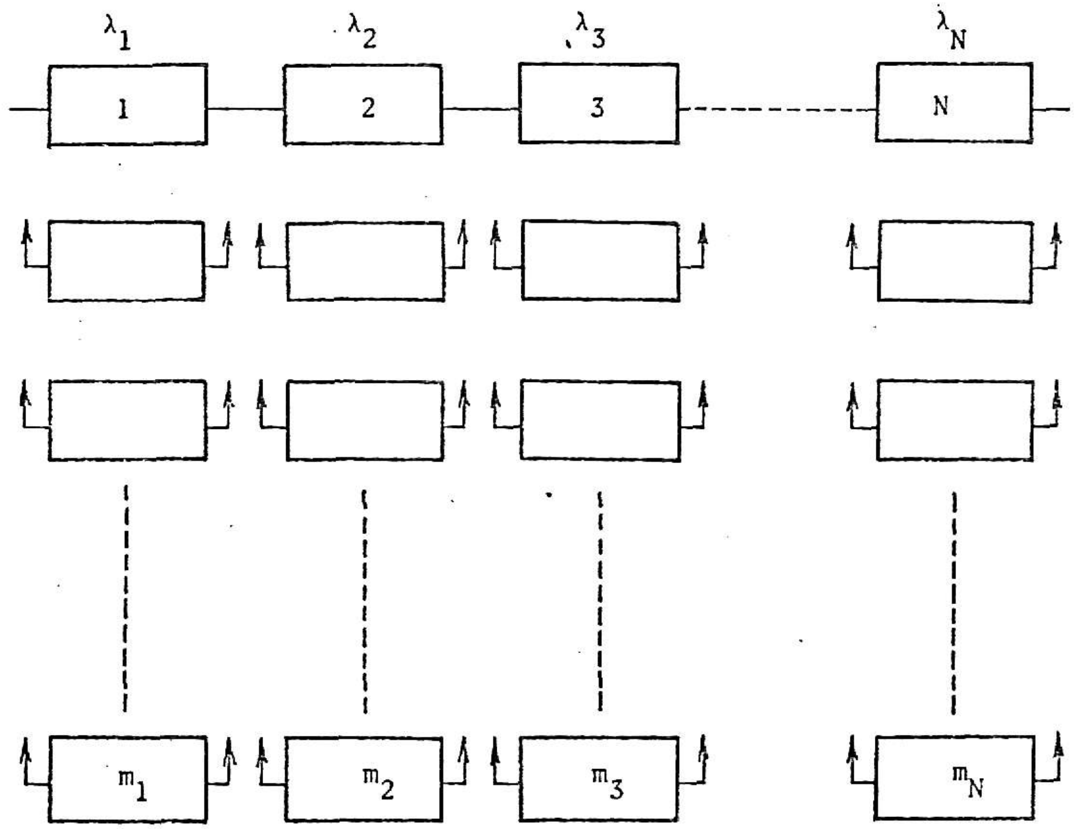

Fig. 4 A General Stand-By System

As a first case let us assume that the standing-by units have similar characteristics, i.e., the same constant failure rate, λ. If there is then one unit is required, and there are m units as stand-by's, the system can be considered as a simple system having λ as the failure rate and m failures are allowed before the system fails. Only the (m+1)th failure causes the system to fail; its reliability is

$$R_s = e^{-\lambda t} \left(1 + \lambda t + \frac{(\lambda t)^2}{2!} + \dots + \frac{(\lambda t)^m}{m!}\right) .$$

This equation is obtained using the Poisson's distribution of errors. If the probability of no failure is R, then it can be expressed as follows.

$$R = e^{-x}$$

where x is the mean fraction of expected failures for the project of given duration. Then using the Poisson's distribution one can write

$$\text{Prob. (exactly one failure)} = x \cdot e^{-x}$$

$$\text{Prob. (exactly two failures)} = \frac{x^2}{2!} \cdot e^{-x}$$

$$\cdots$$

$$\cdots$$

$$\text{Prob. (exactly n failures)} = \frac{x^n}{n!} \cdot e^{-x}$$

Hence the reliability of a component having n stand-by's is

$$\begin{aligned} R_s &= \text{Prob. (at the most n failures)} \\ &= \text{Prob. (no failure)} + \text{Prob. (one failure)} \\ &\quad + \text{Prob. (two failures)} + \dots + \text{Prob. (n failures)}. \\ &= e^{-x} \left[1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}\right] \end{aligned}$$

where x, mean fraction of expected failures which is the product, λt.

For a system shown in Fig. 4, the reliability of the first stage is

$$R_1 = e^{-\lambda_1 t}\left(1 + \lambda_1 t + \frac{(\lambda_1 t)^2}{2!}) \ldots + \frac{(\lambda_1 t)^{m_1}}{m_1!}\right)$$

of the second stage is

$$R_2 = e^{-\lambda_2 t}\left(1 + \lambda_2 t + \frac{(\lambda_2 t)^2}{2!}) \ldots + \frac{(\lambda_2 t)^{m_2}}{m_2!}\right)$$

and, in general, of the nth stage is

$$R_n = e^{-\lambda_n t}\left(1 + \lambda_n t + \frac{(\lambda_n t)^2}{2!} + \ldots + \frac{(\lambda_n t)^{m_n}}{m_n!}\right) .$$

The system reliability for a N stages system in series, shown in Fig. 4, can be written as

$$R_s = \prod_{i=1}^{N} R_i$$

$$= \prod_{i=1}^{N} e^{-\lambda_i t}\left(1 + \lambda_i t + \frac{(\lambda_i t)^2}{2!} + \ldots + \frac{(\lambda_i t)^{m_i}}{m_i!}\right)$$

(c) Reliability of System with Spares [3]

When a standing-by unit can be used at more than one place, the unit is called spare, (Fig. 5). For example a heat exchanger may have 5 similar control valves, each of which is necessary for successful operation, and there are 3 valves of this type in stores, so we can call these stand-by valves as spares which we can plug into use at any of the five locations.

Consider a system having s similar components. There are n spare components of this type in the reserve. The intensity of failure for a component of this type is λ; then the failure intensity for the

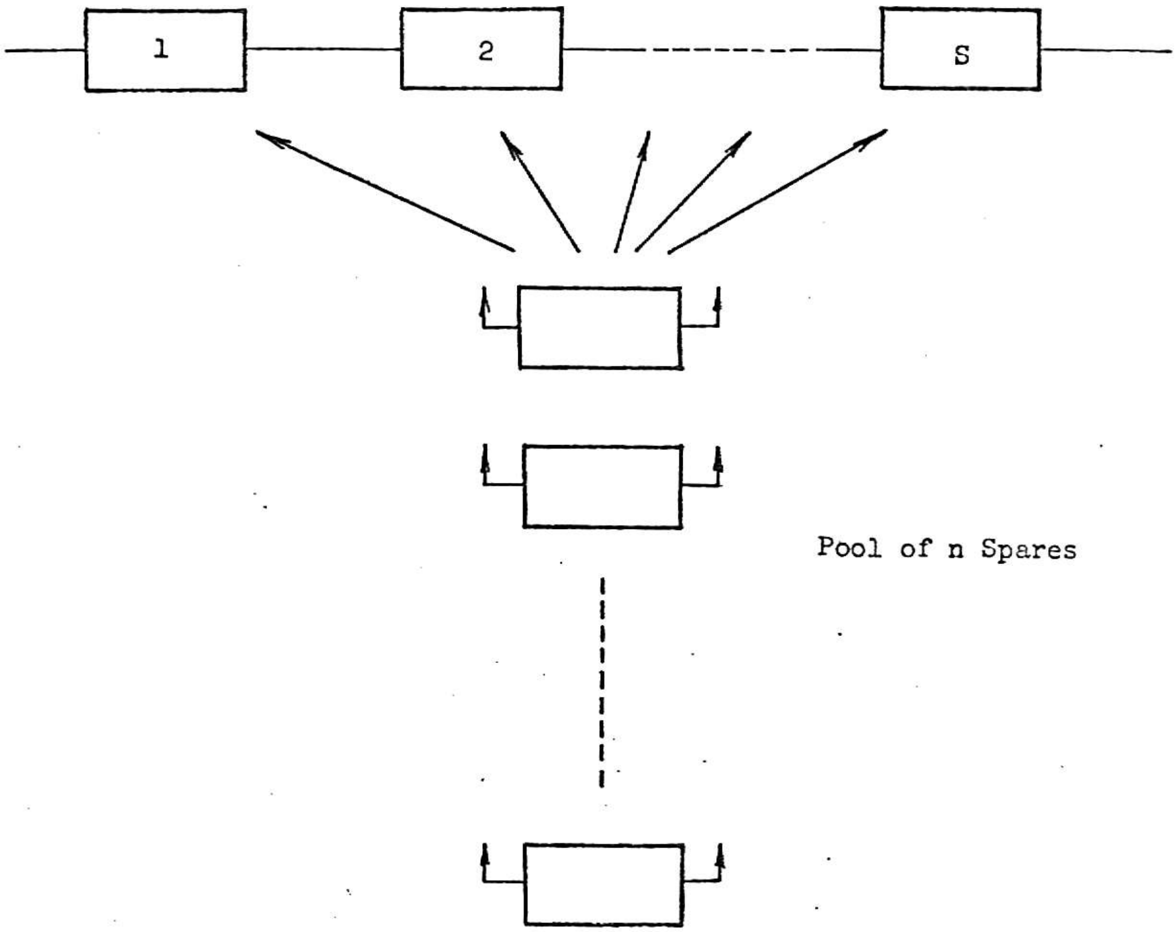

Fig. 5 A System with S Similar Components Supported by n Spares

whole group of components of this type in the system is $(s\lambda)$. The system can be operated until n components of this type fail; hence the reliability of the system can be written as

R_s = Probability of n failures of a component having failure intensity $(s\lambda)$

$$= e^{-(s\lambda t)}\left\{1 + (s\lambda t) + \frac{(s\lambda t)^2}{2!} + \ldots + \frac{(s\lambda t^n)}{n!}\right\} .$$

For additional information on the theory of reliability, one can refer to published books such as references [1,3,24a,30a,36a,36b].

2.6 THE PROBLEM CONSIDERED IN THIS REPORT

The subsystems of life support systems presented in details in the later part of this report have been designed at the space division of North American Rockwell Corporation [29] under the auspice of NASA. They have used a graphical method, a summary of which is given in Appendix A, to determine the number of spares required so that each component is equally reliable in a subsystem. The weights of the spares have been estimated to predict the total weight of the life support systems. They have not tried to optimize the system subject to some constraints of weight or the reliability, rather they have determined a feasible solution to the problem. In this report an attempt is made to optimize the weight of spare components while the system must have the reliability at least equal to the specified value.

The approach used in the report is to determine the contribution toward system reliability and the requirement of the resources by each of the spare components allowed in the system. From the feasible solutions, the optimum system is selected on the basis of the reliability contribution and the resources used. The units which consume minimum amounts of the resources and contribute maximum effectiveness toward the reliability of the system are included first in the system. The additional supporting units are selected and added on the same basis one by one until the specified value of reliability is achieved.

It is expected that one can maximize the reliability of the system by allocating the available resources in a selective manner. In like manner one can minimize the amount of the resources used while maintaining a minimum acceptable level of reliability. The technique of zero or one integer programming has been utilized to allocate the resources in the best manner. This approach guarantees that the solution will be obtained in terms of integers or whole units.

3 INTEGER PROGRAMMING AND ITS APPLICATION TO OPTIMIZATION OF SYSTEM RELIABILITY

This chapter discusses various integer programming approachs and how these are applied to the optimization of system reliability. The various integer programming methods are described briefly, while an implicit enumeration technique is given in detail. A computer program based upon this algorithm was used to obtain the optimal solutions for the problems in this report.

A great deal of the work of finding the optimal number of supporting units in reliability optimization has been carried out by assuming the decision variables to be continuous and that they can take any positive value. The solution then is obtained by rounding the variable values to the nearest integer. But as mentioned earlier, there is no guarantee that such a solution is optimal. These approximations may be quite acceptable when the variables assume large values, but in reliability optimization problems the number of supporting units usually falls below, say five units and in such cases making 2.4 as 2 or 2.6 as 3 may result in a non-optimum solution. Hence a method which would determine the integer values for the decision variables is very desirable. For this purpose, the reliability problem must be written in a form which is compatible with the available methods of obtaining all-integers solution. The reliability optimization problem is formulated in this form in this chapter. Also included is a review of the literature on the application of the integer programming to the optimization of system reliability problems. This discussion is followed by the formulation of the following problems as a integer programming and zero-one integer programming problems:

(i) maximizing system reliability subject to cost constraints,

(ii) minimizing the cost of the system subject to constraints on reliability and other resources.

3.1 INTEGER PROGRAMMING PROBLEMS

The integer programming problem possesses a characteristic which differentiates it from a linear or nonlinear programming problem. This is the requirement that all variables in the solution to be integers. The integer programming technique has been applied to a variety of the problems, namely, man or machine allocation, reliability optimization, machine scheduling and the travelling salesman problem.

One could simply solve a integer programming problem ignoring the integer requirement and then round the values of the resulting solution to the nearest integers satisfying the constraints. However, this procedure may result in a non optimal solution. Hence a method is needed which will find an optimal integral solution. The general problem can be stated as follows [17]:

Find the positive integer values for a set of n variables x_j satisfying m linear inequalities or equalities of the form

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$$

which minimize the objective function

$$Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad .$$

All a_{ij}, b_i, and c_j are assumed to be known constants. Mathematically we can write minimize $\quad c_1x_1 + c_2x_2 + \ldots + c_nx_n$

subject to restraints

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2,$$

$$\cdot \quad \cdot \quad \cdot$$

$$\cdot \quad \cdot \quad \cdot$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m,$$

where

$x_j \geq 0, \quad j = 1, 2, \ldots n$

x_j is an integer

x_j = the jth variable

c_j = unit cost of variable x_j

b_i = the constraint value on the ith resource

a_{ij} = the consumption of the ith resource by the jth variable.

3.2 TECHNIQUES FOR SOLVING INTEGER PROGRAMMING PROBLEMS

The available techniques for solving integer programming problems are presented in this section. The brief history of each group of techniques is also included. The basic approach of each of the techniques is also presented. The detail of implicit enumeration technique, used for solution of the problems in this report, is given in the next section of this chapter.

The available techniques for the solution of integer programming problems may be classified into three major groups depending upon the basic approach employed in the techniques. The groups can be identified as: Cutting Methods, Branch and Bound Methods and Miscellaneous methods.

The basic approach of the cutting methods is that of successively reducing the feasible solution space by deducing supplementary linear constraints from the constraints of the original problem until an optimal solution is obtained which satisfies the integer requirements. The basic idea of the supplementary constraints was proposed by Dantzig, Fulkerson, and Johnson [5] in 1954 for the solution of travelling salesman problem. In 1958 Gomory [14] developed the idea of new (or supplementary) constraints to obtain a systematic method for solution of integer programming problems of general nature. Gomory [15] also generalized the new constraints method to obtain a method which requires only addition and substraction in computations provided that the data of the problem is composed of only integer value. In 1965 Young [42] proposed primal integer programming algorithm. Eto [8] published a method in 1967 which develops an additional restraint based on the objective function to reduce the permissible combination of integer values of the variables so as to make use of either the branch and bound or enumeration search methods that are available.

Branch and bound methods are basically enumeration search. These methods involve two steps. The first step defines one or more subspaces of the feasible solution space to which branching can be carried on, while the second determines a bound on the value the objective function can attain in each of these subspaces. These algorithms consider the solution space as the space of all integer values. After branching it is determined whether a better solution than the one at the hand may be obtained. This procedure of alternatively dividng and examining the solution continues till the whole solution space is examined. The basic idea was proposed by Little et al. [26] for solving the traveling salesmen's problem, which involves zero-one integer programming problem. Land and Doig [24] published a method with a similar approach for solving pure and mixed integer programming problems. Most of the work on this approach has been done on zero-one integer programming. To utilize these techniques an integer programming problem must be reduced to zero-one integer programming problem. The methods to effect this conversion are discussed in the next section. In 1965 Balas [2] published the additive algorithm which was modified by Glover [13]. The most successful technique of this class was proposed by Lemke and Spielberg [25] in 1967. Goeffrion [10] in 1965 presented an algorithm using enumeration and imbedded linear programming. As far as computational efficiency is concerned the implicit enumeration search program based upon the algorithm of Lemke and Spielberg [25] appears to be very good, and it was used for the solution of problems in this report.

There exists a wide variety of techniques for integer programming which are not included in the above two classes. One of the successful approaches is the heuristic programming, which employs both simple and sophisticated selection rules as well as partially corrective trial and error procedures to produce a feasible, and hopefully near optimal solution. The method seem to be efficient for specific problems. Kuehn and Hamburger [23] presented a method which is good for warehouse location or transportation problems. Kaplan [20] published a similar method which is applicable to a particular class of problems. Rao [32] presented a algorithm and computer program for solution of set covering problems which cover a very small class of integer programming problems. A dynamic programming approach was used by Glass [12] and improved by Rao [33].

3.3 REDUCTION OF INTEGER PROGRAMMING PROBLEM INTO ZERO - ONE INTEGER PROGRAMMING PROBLEM

The reduction of the general integer programming problem into a zero - one integer programming problem [39] will be discussed here. It is only necessary to know the upper bound $\bar{m}_j$ on the value each variable x_j can attain to effect this conversion. Most practical problems meet this criterion. There are two techniques for converting the integer programming problems into zero - one integer programming problems. They are the expansion technique and Balas Binary method. In the problem the objective is to find the positive integer values for a set of n variables x_j, which satisfy the m linear or nonlinear but separable constraints

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$$

and which minimize the objective function

$$Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

where the constants a_{ij}, b_i, and c_j are known.

(i) Expansion Technique

Replace x_j by $\bar{m}_j$ new variables y_{jk}, which can take a value either zero or one such that

$$x_j = y_{j1} + y_{j2} + \dots + y_{j\bar{m}_j}$$

Now the problem can be written in zero or one variables. A numerical example is given below.

Find the value of variables x_1 and x_2 which minimize the function

$$z = x_1 + 2x_2$$

subject to

$$-x_1 + 2x_2 \leq 5$$

$$4x_1 - x_2 < 10$$

x_j, $j = 1, 2$ an positive integer

$$x_1 \leq 5$$

$$x_2 \leq 4$$

Introducing zero or one variables

y_i, $i = 1, 2, \ldots, 9$, such that

$$x_1 = y_1 + y_2 + y_3 + y_4 + y_5$$

$$x_2 = y_6 + y_7 + y_8 + y_9$$

and the problem in zero - one integer variables becomes as follows:

Find the value of variables y_i which minimize the function

$$z = y_1 + y_2 + y_3 + y_4 + y_5 + 2(y_6 + y_7 + y_8 + y_9)$$

subject to

$$-(y_1 + y_2 + y_3 + y_4 + y_5) + 2(y_6 + y_7 + y_8 + y_9) \leq 5$$

$$4(y_1 + y_2 + y_3 + y_4 + y_5) - (y_6 + y_7 + y_8 + y_9) \leq 10$$

$$y_i = 0 \text{ or } 1, \qquad i = 1, 2, \ldots 9.$$

(ii) Balas Binary Method [2]

Each variable x_j is replaced by new variable y_{jk} which satisfies the following equation.

$$x_j = \sum_{k=1}^{P_j} 2^{P_j - k} y_{jk}$$

where y_{jk} are zero - one variables and P_j is given by

$$P_j = [\log_2 \bar{m}_j] + 1$$

$\bar{m}_j$ is the upper bound of the integer variable, x_j, and the brackets indicate the largest integer as not being greater than the quantity within the brackets.

This technique uses the fact that any integer number can be obtained by various combinations of the numbers which are generated by 2^n where n = 0, 1, 2, ..., i.e., 1, 2, 4, 8, 16, ... And the variable x_j can always be replaced by a proper combination of numbers from this series. The quantity p_j is the number of such numbers required to represent a variable x_j and $x_j \leq \bar{m}_j$.

The same integer programming example given in the preceding section is again considered

$$P_1 = [\log_2 5] + 1 = 2 + 1 = 3$$

$$P_2 = [\log_2 4] + 1 = 2 + 1 = 3.$$

The required substitutions are

$$x_1 = 2^{3-1}y_{11} + 2^{3-2}y_{12} + 2^{3-3}y_{13}$$

$$= 4y_{11} + 2y_{12} + y_{13}$$

$$x_2 = 2^{3-1}y_{21} + 2^{3-2}y_{22} + 2^{3-3}y_{23}$$

$$= 4y_{21} + 2y_{22} + y_{23} .$$

Here we need only 6 variables to replace x_1 and x_2 whereas by the expansion technique 9 are required. After the substitution the problem reduces to:

Find the value of variables y_{jk} which minimize the function

$$Z = 4y_{11} + 2y_{12} + y_{13} + 2(4y_{21} + 2y_{22} + y_{23})$$

subject to

$$-(4y_{11} + 2y_{12} + y_{13}) + 2(4y_{21} + 2y_{22} + y_{23}) \leq 5$$

$$4(ry_{11} + 2y_{12} + y_{13}) - (4y_{21} + 2y_{22} + y_{23}) \leq 10$$

$$y_{jk} = 0 \text{ or } 1, \qquad j = 1, 2$$

$$k = 1, 2, 3$$

Conversion by this method always result in fewer variables than does conversion by the simple expansion technique in the resulting zero - one integer programming problem.

3.4 IMPLICIT ENUMERATION TECHNIQUE

Implicit enumeration technique has been used for the solution of problems in this report. A summary of the algorithm of Adaptive Binary Programming based upon the implicit enumeration is given in this section, and a simple illustrative example is given in the next section.

Adaptive binary programming involving implicit enumeration search technique has been developed by Salkin and Spielberg [36]. It is basically an enumeration search scheme utilizing powerful tests to eliminate and select the branches of total enumeration search tree. This seems to make this algorithm efficient from a computational point of view. It uses tests so that nodes of the branching tree can be discarded without their explicit appearance, thus explaining why the scheme is called "Implicit Enumeration".

The brief statement of the algorithm and the general zero - one integer programming problem to be solved are given below.

The problem is one in which it is necessary to find the values of the variables y_j, $j = 1, 2, \ldots, n$, which can take value either zero or one to

minimize $\quad z = C^T y$

subject to $\quad Ay \leq b$

$$y_j = 0 \text{ or } 1 \qquad 1 \leq j \leq n$$

where C = cost vector (n x 1)

y = variable vector (n x 1)

A = constraint matrix (m x n)

b = constant vector (m x 1)

The total possible enumeration search paths are 2^n.

A computer program entitled "DZLP" is available from IBM's New York Scientific Center, 410 East 62nd Street, N.Y. [36]. The program is written in FORTRAN IV and can be used on IBM 360/50 systems. With minor modifications the program can be run on any sufficiently large computer with a FORTRAN compiler.

A brief statement of the algorithm is as follows:

Step 1 Determine the initial search origin either on a priori basis or by a linear programming round up procedure. The linear programming problem is solved assuming the variables y_j of the problem are continuous and unrestricted (any value between 0 and ∞). The values of the variables are then replaced by zero or one. Taking a constant TAU, if the variable y_j is less than TAU, make it zero otherwise make it one. And this gives an initial origin to start the search.

Step 2 Retain the node which generates a node having a lower value of objective function as the current origin for the further search. Record the rounded up original solution as the optimal solution at hand, and whenever a better solution is found during the search record it.

Step 3 Forward step - Involving the fixing of an additional free variable at 1. Pick up a variable from the preferred set and force it to one. That in effect moves the search from level k to (k+1). At the node which is being visited for the first time, the search applies the following tests.

(i) Cancellation test on free variables - Determine which of the variables not assigned any value can be cancelled, that means, can be made equal to zero. If there is no variable to be cancelled at this stage go to step 4.

If for any j, $j \varepsilon \Pi$, and any i $(1 \leq i \leq m)$

$$[\bar{b}_i - \sum_{j \varepsilon M_i} a_{ij} - \max_{a_{ij}>0} . \; a_{ij}] < 0$$

cancel j, and if j corresponds to a forced variable, take a backward step, go to step 4. After cancelling j go to (ii), where

$$\Pi = \{j | y_j \text{ is free, } i \leq j \leq n\}$$

$$M_i = \{j | j \varepsilon \Pi, \; a_{ij} < 0\}$$

$$\sigma = \{j | y_j = 1, \; 1 \leq j \leq n\}$$

$$\bar{b}_i = b_i - \sum_{j \varepsilon \sigma} a_{ij}$$

(ii) Infeasibility Test - it is a check on the feasibility of the constraints after some variables have been cancelled.

If for any constraint i $(1 \leq i \leq m)$

$$\bar{b}_i < 0 \quad \text{and} \quad [\bar{b}_i - \sum_{j \varepsilon M} a_{ij}] < 0$$

go to step 4 and perform the backward step. Otherwise go to step 5.

Step 4 Backward step - Make the forced variable free, by setting it to be zero again, thus moving from (k+1) to k. Find the next variable from the preferred set and take a forward step by going to step 3.

Step 5 Solve the associated linear program to find the values for free variables and the objective function, assuming the variables to be continuous and unrestricted. Find the rounded up solution by imposing the binary restriction on the variable and the corresponding objective function value, z.

If $z < z^*$ record the new rounded solution, update z^* and go to step 4 for backward step and then further branching.

If $z \geq z^*$ use LP solution as a lower bound on the current integer solution, and impose the constraint on all free variables $y_j = 0$ or 1 $(j \varepsilon \Pi)$. Perform one additional pivot step and go to step 6.

Step 6 Cancel some of the free variables and determine some of the variables, which must be forced to value one to insure improved integer solution.

If there is any variable which must have value one to insure improved integer solution, but has been cancelled, take a backward step by going to step 4.

If there is no such variable go to step 7.

If there is no forced variable go to step 8 to determine the next variable in the series of branching, and then go to step 3 for further branching.

Step 7 Take the set of forced variables determined in step 6 as the next series of branching and go to step 3 for further branching.

Step 8 Generate (0,1) Gomory's cuts via complete reduction, and take the smallest cut as the preferred set. Apply the Balas test to this set and create the next branch. Then go to step 3 to branch in the direction thus determined.

Gomory's cut:

Consider constraint τ, for which

$$\bar{b}_\tau = [b_\tau - \sum_{j\varepsilon\sigma} a_{\tau j}] < 0$$

and $$\sigma = \{j | y_j = 1, 1 \leq j \leq n\}$$

This constraint can be rewritten as

$$\sum_{j\varepsilon\Pi} a_{\tau j} \cdot y_j \leq \bar{b}_j < 0$$

This implies a Gomory's cut for y_j

$$\sum_{j\varepsilon P} y_j \geq 1$$

where P is some subset of indices of Π.

Balas Test - it selects a variable in a manner which drives towards the "natural" feasibility goal. That is, for each variable in preferred set compute the Balas value

$$v_j = \sum_{i \varepsilon Q} (\bar{b}_i - a_{ij})$$

where $Q = \{i | (\bar{b}_i - a_{ij}) \leq 0, \bar{b}_i < 0, 1 \leq i \leq m\}$.

The j, which maximizes v_j, is denoted by $\bar{j}$, and is the next branching path.

3.5 A SIMPLE EXAMPLE

An integer programming problem is given below and has been solved by the zero - one integer programming algorithm for the purpose of illustration.

The problem is to find the value of variables x_1 and x_2 which

minimize $\quad z = -3x_1 - 4x_2$

subject to $\quad 3x_1 + 2x_2 \leq 8$

$$x_1 + 4x_2 \leq 10$$

where x_1 and x_2 are positive integer variables. This problem is reduced to a zero - one programming problem by using the expansion technique which is described in section 3.3 and employing the following ceiling on the variables

$$x_1 \leq 2$$

$$x_2 \leq 2$$

Now substituting the variables y_j, j = 1, 2, 3 and 4 for x_1 and x_2 such that

$$x_1 = y_1 + y_2$$

$$x_2 = y_3 + y_4$$

where

$$y_i = 0 \text{ or } 1, \qquad i = 1, 2, 3, 4 .$$

The problem in zero or one variables is to find the values for the variables y_j which

minimize $\quad z = -3(y_1 + y_2) - 4(y_3 + y_4)$

subject to

$$3(y_1 + y_2) + 2(y_3 + y_4) \leq 8$$

$$y_1 + y_2 + 4(y_3 + y_4) \leq 10$$

where

$$y_i = 0 \text{ or } 1, \qquad i = 1, 2, 3, 4$$

The solution of this problem using the algorithm of implicit enumeration search is given below and the search paths are shown on the search tree in Fig. 6. This problem can be summarized in the following manner.

$$n = 4,$$

$$m = 2,$$

the cost vector $C = [-3, -3, -4, -4]$,

the variable vector $y^T = [y_1, y_2, y_3, y_4]$,

the constraint matrix $A = \begin{pmatrix} 3 & 3 & 2 & 2 \\ 1 & 1 & 4 & 4 \end{pmatrix}$,

and the constant vector $b = [{8 \atop 10}]$.

Total possible enumeration search paths $= 2^4 = 16$.

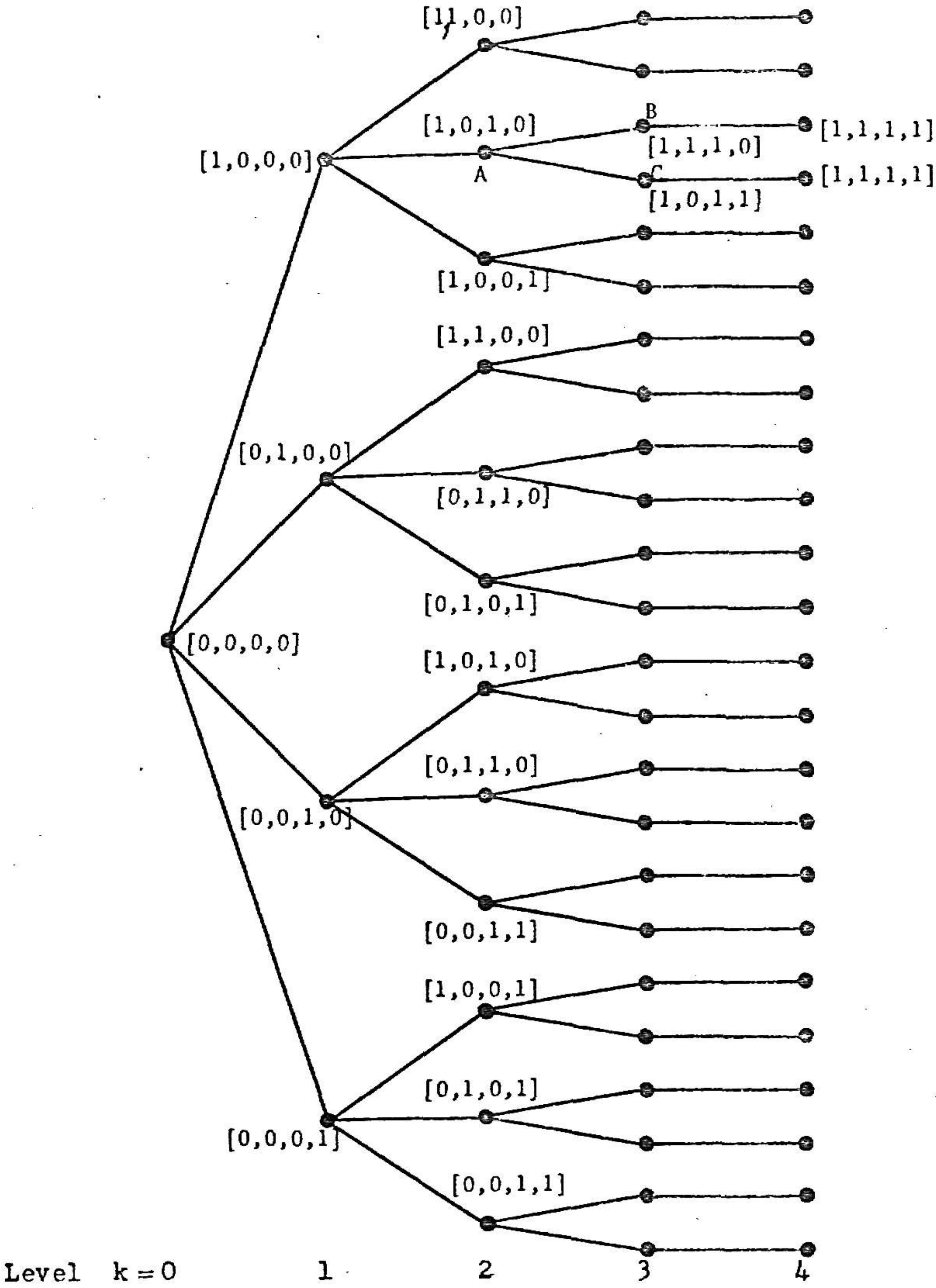

Fig. 6 - Complete Search Tree for 4-Variables Zero-Or-One Programming Problem.

Step 1. Finding the initial search origin:

The search origin can be determined on the priori basis or by linear programming round up method. This example was solved by linear programming round up method. For that the problem was solved by simplex method assuming the variables y_i, i = 1, 2, 3, 4 to be continuous positive and taking any value between 0 and ∞. This procedure yielded the following results after 3 iterations.

$$y_1 = 1.2, \qquad y_2 = 0$$
$$y_3 = 2.2, \qquad y_4 = 0$$
$$z = -12.4$$

After replacing by zero-one variables it gives the initial search origin as (taking TAU = 0.05)

$$y_1 = y_3 = 1$$
$$y_2 = y_4 = 0$$

and yields

$$z = -7$$

This is the point A in the search tree, Fig. 6.

Step 2. Recording the solution and value of the objective function. Retain it until a better solution is obtained.

$$z^* = -7$$

Step 3. Forward step (move from A to B):

At this stage, no variable has been assigned any value hence the set of free variables is = $\{y_2, y_4\}$ out of these variables, no variable has been cancelled, that is, the set of cancelled variables = Φ where Φ is a empty set.

(A variable which was made zero at any level of branching is called cancelled at that level, it must remain zero on all nodes successors to that node.) Preferred set is the set of variables which are permissible, here it is $= \{y_2, y_4\}$

So far k = 2, now one of the variables in preferred set is forced to value one, thus making k = 3. Now

Forced variable $= y_2$

i.e., $y_2 = 1$

Set of free variables $= \{y_4\}$

Sets $M_1 = \Phi$

$M_2 = \Phi$

Now the tests will be applied to this node.

(i) Cancellation test

$$\bar{b}_1 = 8 - 3 - 3 - 2 = 0$$

$$\bar{b}_2 = 10 - 1 - 1 - 4 = 4$$

cancel j, $j \varepsilon \Pi$ for which

$$[\bar{b}_i - \sum_{j \; M_i} a_{ij} - \max_{a_{ij} \; 0}. \; a_{ij}] < 0, \qquad 1 \leq i \leq m$$

In this example for j = 4, i = 1

$$0 - 0 - 2 = -2 < 0$$

that means cancel j = 4 or set $y_4 = 0$

for j = 4 and i = 2

$$4 - 0 - 4 = 0$$

On the search tree this point is B at level, k = 3. It is also the end of the branch of the search tree, since there is no other free variable. Go to (ii)

(ii) Infeasibility test

No $\bar{b}_i$ is negative. Go to step 5

Step 5. Solve the associated linear programming problem and determine the corresponding value of the objective function. There is no more variable to be determined, hence no need of solving any linear programming problem. Current solution is [1,1,1,0]

$$z = -10 < z^*$$

This is a better solution. Hence record the solution as current optimal solution.

$$z^* = -10$$

Go to step 4

Step 4. Backward step (move from B to A)

Go to level, $k = 2$, setting $y_2 = 0$. Find the next variable in the preferred set and go to step 3.

Step 3. Forward step (move from B to C)

Next variable in the preferred set = y_4

$k = 2$

Indices of the free variables = {2,4}

Forced variable = y_4, that is,

$y_4 = 1$

now $k = 3$

Free variable = y_2

set of indices of free variables $\Pi = \{2\}$

sets $M_1 = \Phi$

$M_2 = \Phi$

set $\sigma = \{1,3,4\}$

(i) Cancellation Test

$\bar{b}_1 = 8 - 3 - 2 - 2 = 1$

$\bar{b}_2 = 10 - 1 - 4 - 4 = 1$

For $j = 2$, $i = 1$

$1 - 0 - 4 = -3 < 0$

that means cancel y_2, or set $y_2 = 0$

In search tree it is the point C.

(ii) Infeasibility test

All $\bar{b}_i$ are positive, go to step 5.

Step 5. There is no other free variable at this stage, hence no more variables to be determined. Otherwise a linear programming problem has to be solved to get the values of free variables and objective function. Current solution [1,0,1,1]

The corresponding value of the objective function

$z = - 11$

Go to step 4

Step 4. Backward step (move from C to A)

Go back to $k = 2$ level, setting $y_4 = 0$.

But there is no more variable in the preferred set. So this is also the end of the search, the recorded solution and the value of the objective function are optimal.

3.6 REVIEW OF LITERATURE ON OPTIMIZATION OF SYSTEM RELIABILITY

In this section a review of the literature on optimization of system reliability will be presented. In 1956 Ditoro [6] formulated the problem of maximizing the reliability of a system subject to single equality constraint. He used Lagrange multipliers to arrive at a equation which must be satisfied for the system to be optimal. He presented a two component relay system in which the reliability was to be maximized and spare weight was allocated using the graphical method. In the same issue of IRE Transactions, Moskowitz and McLean [28] solved the problem of determining the optimum distribution of elements for maximum reliability at minimum cost using the classical differential method. They solved a five-components problem taking decision variables to be continuous and then rounding these to nearest integers.

In 1958 Bellman and Dreyfus [4] formulated the problem of maximizing the multi-component system reliability subject to cost, size and weight constraints. A five-stage problem has been solved where the two constraints are weight and cost. The supporting units were identical to original units and were placed as parallel redundancies. The Lagrange-multiplier method was used. In the same paper they also considered the problem of chosing the type and number of component in a system.

Kettelle [21], in 1962, used dynamic programming, which was suggested by Bellman and Dreyfus [4], to determine the number of stand-by units for a system where the reliability requirements were specified and the total cost was a minimum. The allocation of components to stages of the system with (continuous) exponential availabilities were considered. Availability, here, means that the component will operate when it is needed. It was noted that if the unavailability decreases exponentially then so does the unreliability of the system. Proschan and Bray [31], in 1965, modified this formulation to consider more than one constraint. They also programmed the algorithm and indicated they have solved problems with 20 stages. There is a problem with this formulation as is common to all dynamic programming problems which is a requirement of a large storage as well as time consuming calculations. An example has been presented in which the reliability of a 4 stages system was maximized subject to 3 constraints, weight, volume and money.

Kolesar [22] in 1967 used a linear programming formulation for the reliability optimization problem and solved the problem by zero or one programming methods of Balas [2] and Glover [13]. The problem considered linear constraints of weight, cost and volume. In the first part of the paper, the author formulated the maximization of the reliability of a system dealing with a single type of failure. In the second part of the problem of systems with two types of failure have been considered. He solved the system of relays which have independent probability of short circuiting and open circuiting.

In 1967, Tillman and Littschwager [38,39] presented an integer programming formulation for optimization of system reliability subject to linear and nonlinear separable constraints. Gomory's algorithm [14,15] was used to obtain the solution of the problems. The problems dealt with were (i) maximizing reliability for a parallel redundancy system subject to multiple linear constraints, (ii) minimizing cost of a parallel redundancy system subject to multiple nonlinear and separable restraint functions while maintaining an acceptable level of reliability, and (iii) optimal choice of design for a parallel redundancy system. The optimal solution appears in integer form as required.

In 1967, Fan, Wang, Tillman, and Hwang [9] used the discrete maximum principle to obtain the optimum number of parallel redundancies for a stagewise system. The authors assumed the variables to be continuous which were rounded to the nearest integers. They maximized the net profit, which is the gross profit from the system, if it operates successfully, minus the construction costs.

In 1968 Tillman, Hwang, Fan, and Balbale [40] solved the optimization of system reliability problem which has single and multiple, linear and nonlinear constraints. The authors used a discrete version of the maximum principle. The computer program was written and a logic flow diagram was presented to solve the problems on an IBM 1620. The computational experience has been reported to be satisfactory. The results obtained were non-integers, and the approximations were required to obtain the optimum number of parallel redundancies.

Tillman, in 1969 [41], formulated the problem of optimizing a constrained reliability problem where the components can fail in s modes. The reliability expressions were presented. Integer programming was used to solve the problems of (i) maximizing system reliability subject to two nonlinear constraints and s modes of failures (ii) minimizing the cost subject to nonlinear constraints while maintaining an acceptable level of reliability subject to s modes of failure. Again the optimal solution appears in integer form as required.

Ghare and Taylor [11a] formulated a problem similar to one considered in this report, and optimized it using zero-one programming based upon branch and bound technique. They considered the problem of parallel redundancy, rather their formulation can not handle any other case, because the formulation includes an equation which is valid only for a system having parallel redundancies.

Most of the work done so far involves either parallel or stand-by redundancies as the means of increasing the system reliability. In this report, the formulation imposes no restrictions on the configurations used for redundancies.

3.7 FORMULATION OF THE PROBLEM

The system considered is a general system with N components, each of which is essential for successful operation of the system and this has a series logic diagram as shown in Fig. 7. There are m_j supporting units for each component, where j denotes the jth type of component. The maximum number of supporting units of jth type is restricted to be M_j. The failure characteristics and the cost of the units in the system and the supporting units are assumed to be known. The supporting units may have different reliability characteristics. The system may have either parallel redundant units, stand-by units or spare units.

There are r restrictions associated with each unit in the system, which may represent money, weight and volume. The coefficients of the ith restriction at the jth stage, c_{ij}, is a function of number of supporting units (or spares) at jth stage m_j.

The reliability of jth stage is a function of unit reliability, r_j, and the number of supporting units (or spares) at jth stage. The system reliability for a series system is the product of the reliabilities at each stage

$$R_s = R_1 \cdot R_2 \dots R_N$$

where

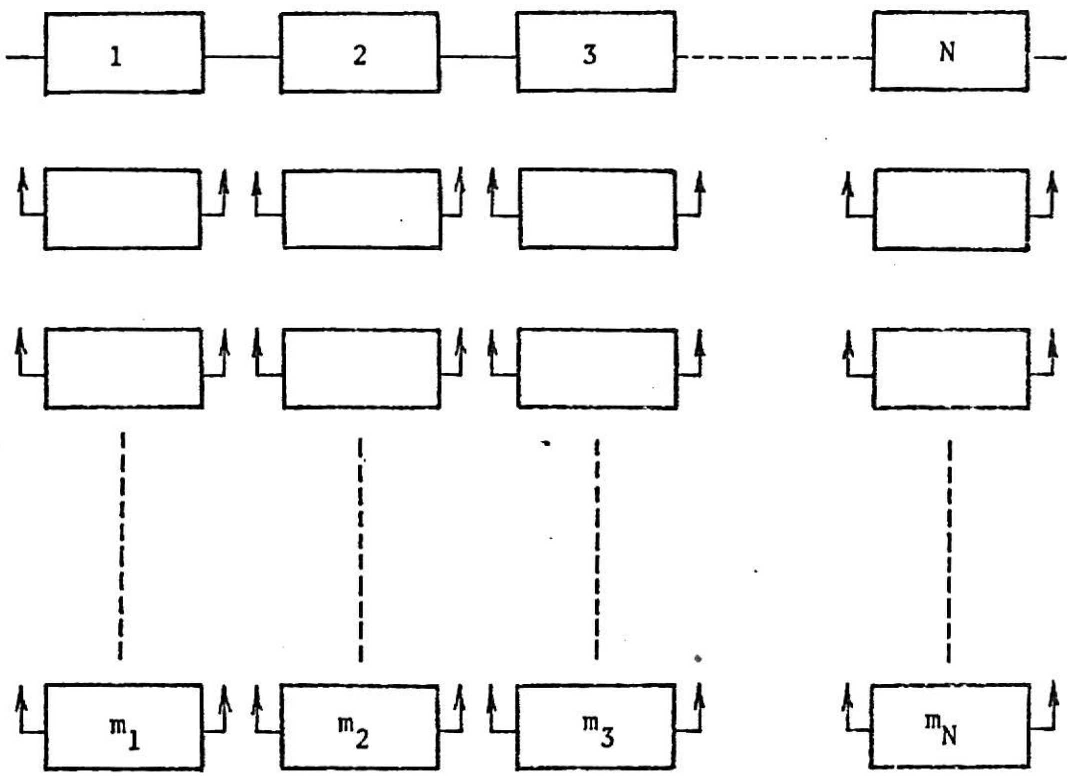

Fig. 7 A System with Components of N Stages and Supporting Units.

$R_j = f(r_j, m_j), \qquad j = 1, 2, \ldots, N$

Defining two quantities, M_j, as the maximum number of supporting units allowed at jth stage, and $\bar{R}_s$, minimum acceptable reliability of the system, the problem can be written as follows:

1. Maximize the reliability of a system with supporting units subject to several separable nonlinear constraints.
2. Minimize the cost of a complex system subject to several separable nonlinear constraints while maintaining an acceptable level of reliability of the system.

The formulations of the problems can be written as

1. Find the value of the variables m_j which maximize

$$z = \prod_{j=1}^{N} R_j(m_j)$$

subject to

$$\sum_{j=1}^{N} c_{ij}(m_j) \leq b_i \qquad i = 1, 2, \ldots, r$$

2. Find the value of the variables m_j which

minimize $z = \sum_{j=1}^{N} C_{pj}(m_j)$

and subject to

$$\sum_{j=1}^{N} c_{ij}(m_j) \leq b_i, \qquad i = 1, 2, \ldots, p-1, p+1, \ldots r$$
$$\neq p$$

and $\quad R_s = \prod_{j=1}^{N} R_j \geq \bar{R}_s$

where z is the objective function, and $\bar{R}_s$ is the minimum acceptable value of the reliability of the system.

These formulations can not be used directly in a zero-one integer programming algorithm because of product terms in reliability inequality. This situation is handled by using the sum of logarithms of various terms instead of the product of those terms. This procedure is valid since the logarithm is a monotonic function of the argument. This substitution makes the following inequality separable.* The modified formulations are

3. Find the values of the variables m_j which maximize

$$z = \sum_{j=1}^{N} \log R_j(r_j, m_j)$$

subject to

$$\sum_{j=1}^{N} c_{ij}(c_i, m_j) \leq b_i, \qquad i = 1, 2, \ldots, r$$

4. Find the values of the variables m_j which

$$\text{minimize } z = \sum_{j=1}^{N} c_{pj}$$

subject to

$$\sum_{j=1}^{N} c_{ij}(c_i, m_j) \leq b_i, \quad i = 1, 2, \ldots, p-1, p+1, \ldots, r$$
$$\neq p$$

and

$$\log R_s = \sum_{j=1}^{N} \log R_j \geq \log \bar{R}_s$$

These formulations must be further modified to put them into the zero - one format. To do this, one has to introduce

*Function $f(x_1, x_2, \ldots x_n)$ is said to be separable if it is possible to express it as

$$f_1(x_1) + f_2(x_2) + f_3(x_3) + \ldots + f_n(x_n),$$

where $f_1(x_1)$, $f_2(x_2)$, ..., $f_n(x_n)$ are the functions of x_1, x_2, ... x_n respectively.

new variables which take values of zero - one. Let this variable be y_{jk}, which represents kth supporting unit of jth type in the system. If this variable assumes a value of 1, it means that the unit must be included in the optimal system; on the other hand, if it has a zero value this indicates that it is not included. This new variable is defined as

$$m_j = \sum_{k=0}^{M_j} y_{jk} \qquad \text{for all } j\text{'s}$$

and

$$y_{jk} = 0 \text{ or } 1$$

where

$$y_{jk} = 1 \text{ when } k \leq m_j$$

$$\quad\;\; = 0 \qquad k > m_j$$

With the proper substitutions the problems can be stated as

5. Find the values of the variables y_{jk} which

maximize $$z = \sum_{j=1}^{N} \sum_{k=0}^{M_j} (\Delta \log R_j) \cdot y_{jk}$$

and subject to

$$\sum_{j=1}^{N} \sum_{k=0}^{M_j} (c_{ijk} \cdot y_{jk}) \leq b_i,$$

$$i = 1, 2, \ldots, r$$

6. Find the values of the variables y_{jk} which

minimize $$z = \sum_{j=1}^{N} \sum_{k=0}^{M_j} c_{pjk} \cdot y_{jk}$$

subject to

$$\sum_{j=1}^{N} \sum_{k=0}^{M_j} c_{ijk} \cdot y_{jk} \leq b_i \; ,$$

$$i = 1, 2, \ldots, p-1, p+1, \ldots r$$
$$\neq p$$

and

$$\log R_s = \sum_{j=1}^{N} \sum_{k=0}^{M_j} (\Delta \log R_j) \cdot y_{jk} \geq \log \bar{R}_s$$

where the k index denotes a particular supporting unit of the jth type and

$\Delta \log R_{jk}$ = the increase in $\log R_{jk}$ due to addition of the kth unit of jth type

$$\Delta \log R_{jk} = \begin{cases} \log R_{jk} & \text{for } k = 0 \\ \log R_{jk} - \log R_{j(k-1)} & \text{for } k = 1, \ldots, M_j \end{cases}$$

where R_{jk} is the reliability of jth stage with k supporting units

c_{ijk} = is the coefficient of ith restriction associated with kth supporting unit of jth type.

The life support systems which are presented in the next chapter are solved using this last formulation 6.

3.8 DETERMINATION OF COEFFICIENTS

The cost coefficients c_{ijk} are assumed to be known for all i's, j's and k's. In most cases the supporting units are similar to the original units and c_{ijk} for all k's are equal.

To determine the coefficients in the reliability expression ($\Delta \log R_{jk}$), it is necessary to calculate actually the value of reliability of the jth stage with all the supporting units. First we must find the values of R_{jk} for $k = 0, 1, 2, \ldots M_j$, where R_{jk} is the reliability of jth stage with k supporting units. To find R_{jk}, one must use one of the methods for calculating the reliability as discussed in section 2.5, depending upon the type of system used. In this report the reliability is improved by providing spare units. Hence equation of section 2.5 (c) was used to calculate R_{jk} as follows:

$$R_{j0} = e^{-(m\lambda t)} \qquad = \Pr \text{ (No failure)}$$

$$R_{j1} = e^{-m\lambda t} \cdot \left(1 + \frac{m\lambda t}{1!}\right) \qquad = \Pr \text{ (One failure or less)}$$

$$R_{j2} = R_{j1} + e^{-m\lambda t} \cdot \frac{(m\lambda t)^2}{2!} \qquad = \Pr \text{ (2 or less failures)}$$

$$\vdots$$

$$R_{jk} = R_{j(k-1)} + e^{-m\lambda t} \cdot \frac{(m\lambda t)^k}{k!} \qquad = \Pr \text{ (k or less failures)}$$

where t is the duration of the mission, λ is the failure intensity, and assumed to be a constant, and m is the number of units at jth stage.

Now the coefficients ($\Delta \log R_{jk}$) can be determined as follows.

$$\Delta \log R_{j0} = \log R_{j0}$$

$$\Delta \log R_{j1} = \log R_{j1} - \log R_{j0}$$

$$\Delta \log R_{j2} = \log R_{j2} - \log R_{j1}$$

$$\vdots$$

$$\Delta \log R_{jk} = \log R_{jk} - \log R_{j(k-1)}$$

$$\vdots$$

$$\Delta \log R_{jM_j} = \log R_{jM_j} - \log R_{j(M_j-1)}$$

With the above information, the problem can then be stated as a zero-one integer programming problem. This approach for determining the optimal configuration for a life support system is presented in the next section.

4 APPLICATION OF ZERO - ONE INTEGER PROGRAMMING TO LIFE SUPPORT SYSTEMS

The formulation presented in Chapter 3 was used to obtain the spare allocations to achieve the specified system reliability and minimize the weight of the spares. Some circuits of life support systems were optimized. The results obtained are compared with the designs suggested by North American Rockwell Corp. [29]. A comparison indicates that present system formulation resulted in a better arrangement of spares, the weight of the spares being less than that presented in ref. [29] and the system has the same level of reliability.

4.1 PROBLEM OF LIFE SUPPORT SYSTEMS

Basically the life support system must support both men and equipment for the mission by providing the required environmental characteristics. It must provide the crew with an atmosphere controlled in terms of both content, temperature and pressure for both the mission module and the earth re-entry module. It must provide temperature control for all temperature sensitive systems in both modules. The system is called an integrated life support system, because it is complete in itself and does not require any other supporting system. A partially closed life support system is presented in Fig. 8 The principal subsystems have been marked. The aspects of constrained reliability of integrated life support systems have been studied at North American Rockwell Corporation under the auspices of NASA [29]. The systems are required to perform with a reliability of at least 0.999 for a project duration of 16,800 hours, that is, not more than one failure in 1000 experiments, each has approximately 2 years

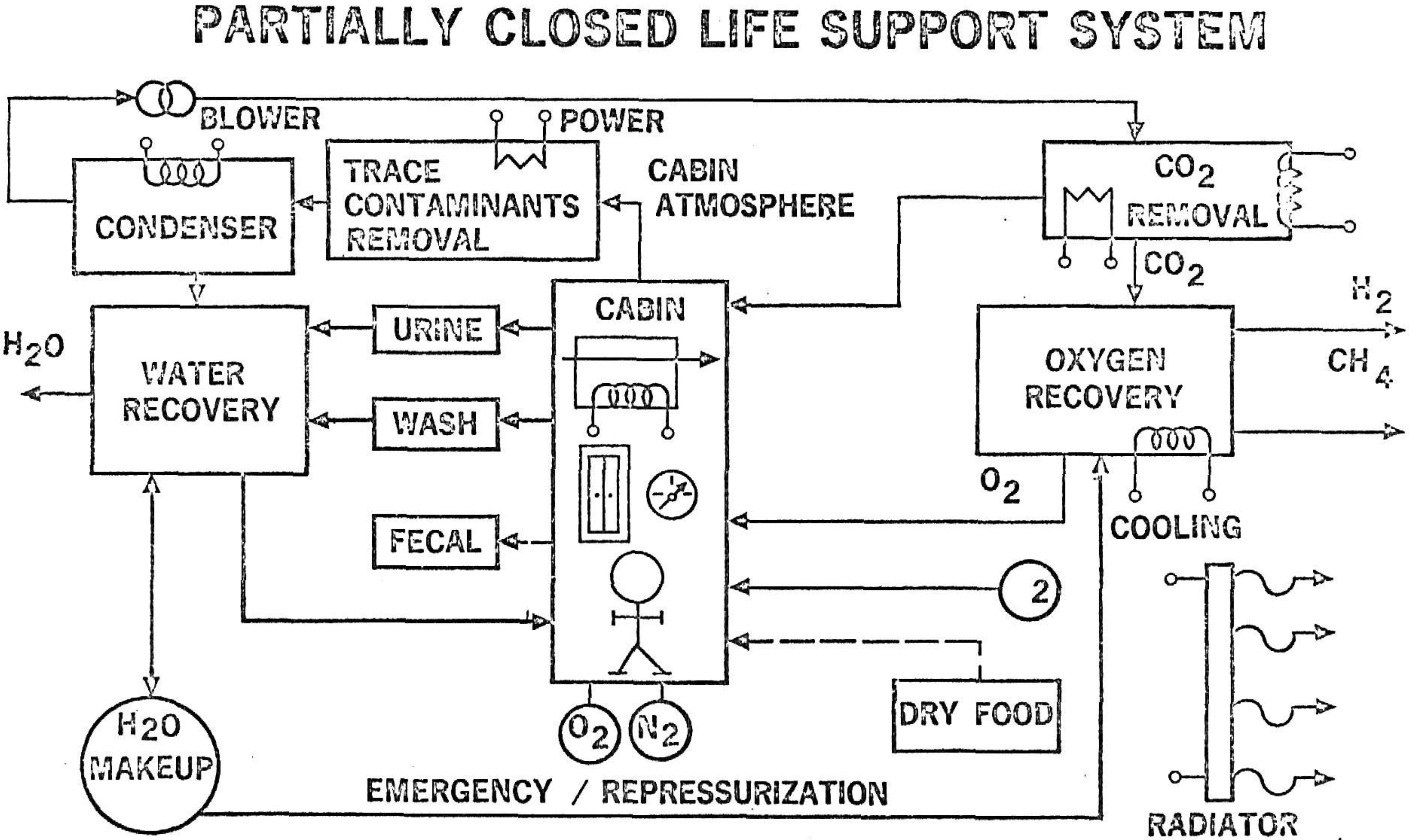

Fig. 8 - An example of Life Support Systems [27a]

duration. This required a very carefull study of the failure characteristics of every component of the system. The study was conducted to estimate the values of mean failure rates and the reliability of each part, these estimated values were listed in a report [29]. In that report an attempt was made to design the allocation scheme for spares to achieve a reliability level such that all components would be equally reliable.

Without any supporting parts, either as redundant components or as spares, the system of life supporting equipments has the reliability of 0.1041, that is, slightly more than 10% chance of safe project completion. This is a very low value of reliability and thus the system's reliability must be improved. As mentioned in the second chapter, there are many methods by which to increase the system reliability. In this report the weak components of the systems have been assumed to be supported with the spare parts which can be placed in operation as soon as the failure is sensed by a sensing device. It has been assumed that the reliability of the failure sensing unit is 1. It is also assumed that the time required to put the new part into operating condition is very brief and that it is available for the necessary switching and replacement.

The reliability estimates for the subsystems are listed in Table 1. The reliability standards, which are the reliability levels the subsystems are required to achieve, for each circuit have been taken as those obtained in the report [29]. The problem is to minimize the weight of the spares which will satisfy the minimum requirement of system reliability. It is expected that the allocation of spares

Table 1. Integrated Life Support Systems

Subsystem Reliability Estimates [29]

No.	Subsystem	Duty Cycle in Hrs.	Estimated Reliability
1	Radiator Circuit	16800	0.9316
2	Refrigerant Circuit	16800	0.7756
3	Atmospheric Circuit	16800	0.6149
4	Coolant Circuit	16800	0.8658
5	Humidity Circuit	16800	0.9243
6	Water Reclamation	16800	0.8929
7	Potable Water	16800	0.9838
8	Waste water	16800	0.9773
9	Carbonation Cell	16800	0.8612
10	Cryogenic Supply	16800	0.4583
11	High Pressure O_2	16800	0.8560
12	Bosch Reactor	1680	0.9174
13	Electrolysis	1680	0.9809
	System Reliability		0.1041

determined in the present report will satisfy the minimum reliability requirements while the spare weight is minimized.

4.2 OPTIMUM SOLUTION OF THE REFRIGERANT CIRCUIT

A. STATEMENT OF THE PROBLEM

The refrigerant circuit of the life support systems given in [29] has 14 components, which were studied and tested minutely. The reliabilities of these components were estimated [29] and are listed in Table 2. The reliability of the components varies from 0.946 to 0.999998. The system reliability without supporting components is 0.776. To achieve the required reliability of integrated life support systems, this circuit must have the reliability of 0.99972 or better. The reliability of the circuit can be increased by any of the methods listed in Chapter 2. In this report the method of spares is used to increase the reliability. The problem is how to minimize the weight of spares while achieving the specified level of reliability.

The refrigerant circuit of the life support systems considered here consists of 14 components, and all the 14 types of components must operate properly to keep the refrigerant circuit in working condition. This situation suggests that the components are in series from a logic diagram and a reliability point of view. The spare units are provided for each component separately, at the most five spares are allowed for each component. This system is shown in Fig. 9. The system reliability, R_s, is given by

$$R_s = \prod_{j=1}^{14} R_j$$

Table 2. Refrigerant Circuit Analysis [29]

No.	Component Type	Duty Cycle	Weight	Reliability
1	Pump	16,800	1.5	.987
2	Pump Control	"	1.0	.9994
3	Ref. Evaporator	"	10.0	.99998
4	Evap. back Pr. Control	"	4.8	.9994
5	Evap. back Pr. valve	"	10.0	.999998
6	Coolant Heat Exch.	"	10.0	.99998
7	Cabin Heat Exch.	"	10.0	.99998
8	Cabin Temp. Control	"	0.9	.987
9	Evap. water control	"	1.0	.952
10	Heat exch. diverter	"	1.0	.946
11	Ref. temp. con. valve	"	0.75	.987
12	Ref. shutoff valve	"	0.8	.963
13	Ref. check valve	"	0.1	.963
14	Selector valve	"	0.6	.963
	Total	16,800		.776

Note - Weight for component type 3,5,6, and 7 have been assumed.

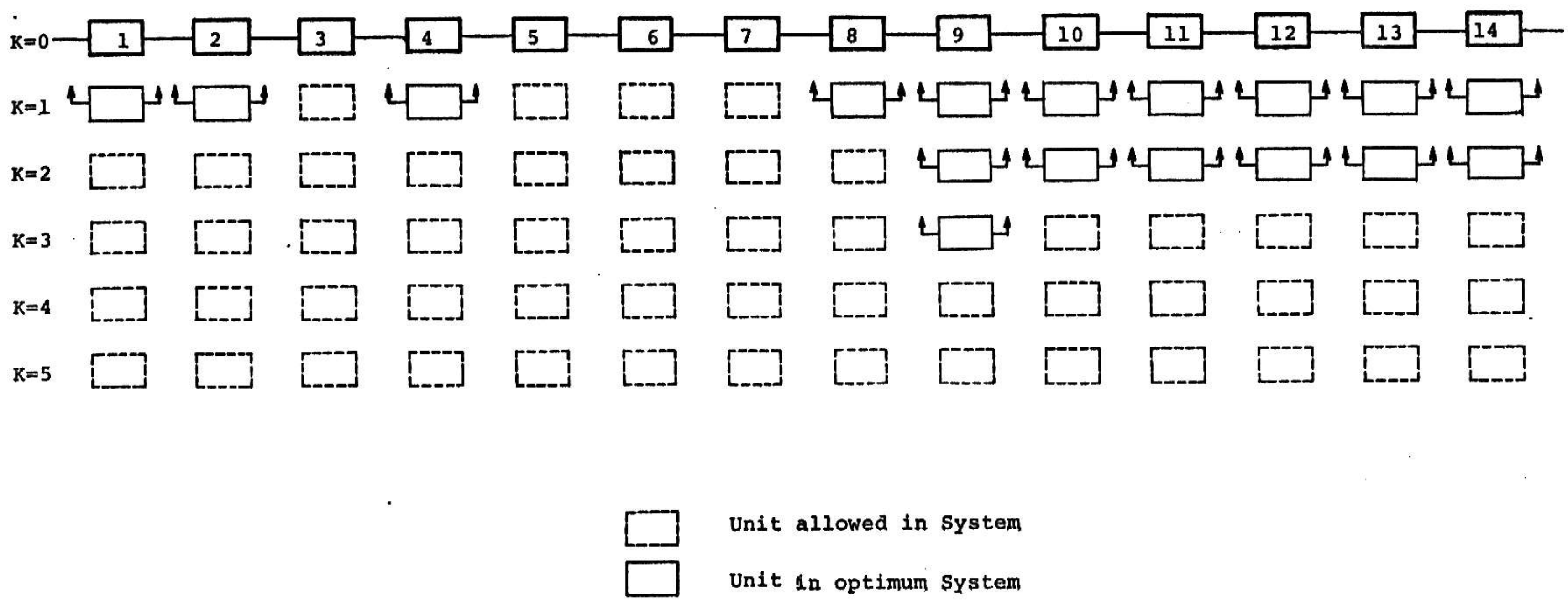

Fig. 9 - Representation of Refrigerant Circuit Problem

where R_j is the reliability of the jth component including the supporting units if there are any.

With no spare units the system reliability is 0.776. This reliability is to be improved to at least 0.99972 by providing spare units while minimizing the weight of the spares. It is assumed that the spares are identical and have the characteristics identical to those of the original units and that their weights are known.

If there are k spares ($k \leq 5$) of the jth component in the system, the reliability of the jth component, R_{jk}, is

$$
\begin{aligned}
R_{jk} &= \text{Prob. (at most k failures)} \\
&= \text{Prob. (no failure)} + \text{Prob. (one failure} + \ldots \\
&\quad \ldots + \text{Prob. (exactly k failures)} \\
&= e^{-\lambda_j t}\left(1 + \lambda_j t + \frac{(\lambda_j t)^2}{2!} + \ldots + \frac{(\lambda_j t)^k}{k!}\right)
\end{aligned}
$$

Now the following terms are defined

$$
\begin{aligned}
\Delta\log R_{j0} &= \log R_{j0} \\
&= \log (e^{-\lambda_j t}) \\
&= -\lambda_j t \\
\Delta\log R_{j1} &= \log R_{j1} - \log R_{j0} \\
&= \log (e^{-\lambda_j t}(1 + \lambda_j t)) - \log (e^{-\lambda_j t}) \\
&= \log (1 + \lambda_j t) \\
\ldots \quad & \ldots \quad \ldots \\
\Delta\log R_{jk} &= \log R_{jk} - \log R_{j(k-1)}
\end{aligned}
$$

$$= \log \left(e^{-\lambda_j t}\left(1 + \lambda_j t + \ldots + \frac{(\lambda_j t)^k}{k!}\right)\right)$$

$$- \log \left(e^{-\lambda_j t}\left(1 + \lambda_j t + \ldots + \frac{(\lambda_j t)^{k-1}}{(k-1)!}\right)\right)$$

$$= \log \left(\frac{1 + \lambda_j t + \ldots + \frac{(\lambda_j t)^k}{k!}}{1 + \lambda_j t + \ldots + \frac{(\lambda_j t)^{k-1}}{(k-1)!}}\right) .$$

We also define the binary variables u_{jk}, such that

$u_{jk} = 1$ if the unit represented by u_{jk}, that is, the kth unit of the jth type, is in the system

$= 0$ if that unit is not in the system for $1 \leq j \leq 14$ and $0 \leq k \leq 5$,

we can rewrite the reliability constraint as follows;

$$\log R = \sum_{j=1}^{14} \log R_j$$

and

$$\log R_j = \sum_{k=0}^{5} \Delta\log R_{jk} \cdot u_{jk}$$

$$\sum_{j=1}^{14} \sum_{k=0}^{5} u_{jk} \cdot \Delta\log R_{jk} \geq \log (0.99972).$$

The objective function to be minimized can be written as

z = sum of weights of all the spare units

$$= \sum_{j=1}^{14} \sum_{k=0}^{5} u_{jk} \cdot (w_j)$$

where w_j is weight of the jth type of unit.

To represent all the possible 14 x 6 units in the system 84 binary variables are required. If the variables y_i ($1 \leq i \leq 84$) are introduced, the indices of the u_{jk} and y_i will have the following relationship.

$$
\begin{aligned}
i &= 1 + k + 6(j-1) \\
&= 6j + k - 5.
\end{aligned}
$$

The problem in terms of the variables y_i becomes:

Determine the values of variables y_i so that the objective function

$$z = \sum_{i=1}^{84} y_i \cdot w_i$$

is minimized subject to

$$\sum_{i=1}^{84} y_i \cdot a_{1i} \geq \log(0.99972)$$

where

$$
\begin{aligned}
a_{1i} = \Delta\log R_{jk}, \qquad & i = 6j + k - 5. \\
& j = 1, 2, \ldots, 14 \\
& k = 0, 1, \ldots, 5
\end{aligned}
$$

To ensure that one unit of each type (14 in all) appears in the optimal system, following additional constraint is added

$$\sum_{i=1}^{84} a_{2i} \cdot y_i \geq 14$$

where $a_{2i} = 1$ for $k = 0$ and all j

that is, $i = 1,7,13, \ldots, 79.$

$a_{2i} = 0$ otherwise.

The values of indices j, k and i are listed in Table 3, along with the values of coefficients w_i, a_{1i}, and a_{2i}. The key for Table 3 is given below.

TABLE 3. ZERO-ONE INTEGER PROGRAMMING FORMULATION OF REFRIGERANT CIRCUIT

J	K	I	COST COEFF	LOG R(I)	
1	2	3	4	5	6
1	0	1	0.00	0.1309E-01	1
1	1	2	1.50	-0.1300E-01	0
1	2	3	1.50	-0.8450E-04	0
1	3	4	1.50	-0.3686E-06	0
1	4	5	1.50	-0.1206E-08	0
1	5	6	1.50	-0.3155E-11	0
2	0	7	0.00	0.6002E-03	1
2	1	8	1.00	-0.6000E-03	0
2	2	9	1.00	-0.1800E-06	0
2	3	10	1.00	-0.3601E-10	0
2	4	11	1.00	-0.5398E-14	0
2	5	12	1.00	0.0000	0
3	0	13	0.00	0.2000E-04	1
3	1	14	10.00	-0.2000E-04	0
3	2	15	10.00	-0.2000E-09	0
3	3	16	10.00	-0.1332E-14	0
3	4	17	10.00	0.0000	0
3	5	18	10.00	0.0000	0
4	0	19	0.00	0.6002E-03	1
4	1	20	4.80	-0.6000E-03	0
4	2	21	4.80	-0.1800E-06	0
4	3	22	4.80	-0.3601E-10	0
4	4	23	4.80	-0.5398E-14	0
4	5	24	4.80	0.0000	0
5	0	25	0.00	0.2000E-05	1
5	1	26	10.00	-0.2000E-05	0
5	2	27	10.00	-0.2000E-11	0
5	3	28	10.00	0.0000	0
5	4	29	10.00	0.0000	0
5	5	30	10.00	0.0000	0
6	0	31	0.00	0.2000E-04	1
6	1	32	10.00	-0.2000E-04	0
6	2	33	10.00	-0.2000E-09	0
6	3	34	10.00	-0.1332E-14	0
6	4	35	10.00	0.0000	0
6	5	36	10.00	0.0000	0
7	0	37	0.00	0.2000E-04	1
7	1	38	10.00	-0.2000E-04	0
7	2	39	10.00	-0.2000E-09	0
7	3	40	10.00	-0.1332E-14	0
7	4	41	10.00	0.0000	0
7	5	42	10.00	0.0000	0

TABLE 3. ZERO-ONE INTEGER PROGRAMMING FORMULATION OF REFRIGERANT CIRCUIT (CONTINUED)

J	K	I	COST COEF	LOG R(I)	
1	2	3	4	5	6
8	0	43	0.00	0.1309E-01	1
8	1	44	.90	-0.1300E-01	0
8	2	45	.90	-0.8450E-04	0
8	3	46	.90	-0.3686E-06	0
8	4	47	.90	-0.1206E-08	0
8	5	48	.90	-0.3155E-11	0
9	0	49	0.00	0.4919E-01	1
9	1	50	1.00	-0.4802E-01	0
9	2	51	1.00	-0.1152E-02	0
9	3	52	1.00	-0.1889E-04	0
9	4	53	1.00	-0.2322E-06	0
9	5	54	1.00	-0.2285E-08	0
10	0	55	0.00	0.5551E-01	1
10	1	56	1.00	-0.5403E-01	0
10	2	57	1.00	-0.1459E-02	0
10	3	58	1.00	-0.2697E-04	0
10	4	59	1.00	-0.3743E-06	0
10	5	60	1.00	-0.4156E-08	0
11	0	61	0.00	0.1309E-01	1
11	1	62	.75	-0.1300E-01	0
11	2	63	.75	-0.8450E-04	0
11	3	64	.75	-0.3686E-06	0
11	4	65	.75	-0.1206E-08	0
11	5	66	.75	-0.3155E-11	0
12	0	67	0.00	0.3356E-01	1
12	1	68	.80	-0.3301E-01	0
12	2	69	.80	-0.5446E-03	0
12	3	70	.80	-0.6090E-05	0
12	4	71	.80	-0.5109E-07	0
12	5	72	.80	-0.3429E-09	0
13	0	73	0.00	0.3770E-01	1
13	1	74	.10	-0.3701E-01	0
13	2	75	.10	-0.6847E-03	0
13	3	76	.10	-0.8601E-05	0
13	4	77	.10	-0.8107E-07	0
13	5	78	.10	-0.6113E-09	0
14	0	79	0.00	0.3770E-01	1
14	1	80	.60	-0.3701E-01	0
14	2	81	.60	-0.6847E-03	0
14	3	82	.60	-0.8601E-05	0
14	4	83	.60	-0.8107E-07	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Column 1 index j

2 index k

3 index i

4 cost coefficients w_i

5 coefficients for reliability constraint a_{1i}

6 coefficient for constraint ensuring one unit of each type in the solution a_{2i}.

B. OPTIMUM SOLUTION AND DISCUSSION

The refrigerant circuit is considered in detail for which the data are given in Table 2. The problem formulation is summarized in Table 3, computer output is given in Appendix B. The optimal systems are presented in Table 4 along with the system obtained in the report of NARC [29] for comparison. In the NARC report it is indicated that spares allocated to the system increase the systems reliability from 0.776 to 0.99972, and that the spares add a weight of 20.3 lb. . In the system suggested in the present work the arrangement of spares weighs 17.7 lb. and achieves the same reliability level of 0.99972. The reduction in weight is achieved by extracting out those units which cost more in terms of resources and contribute less to the system's reliability. The reduction in the subsystem weight of 2.7 lb. appears to be very attractive. Similar reduction in all other subsystems of the life support systems may reduce the total weight by a considerable amount. It is worth mentioning here that the approximate cost of lifting up of one pound in a space effort is in the order of $10,000 at present.

Table 4. Comparison of Optimal Number of Spares for Refrigerant Circuit of Life Support System

No.	Components	Spares Allocation by NARC [29]	Stage Reliability	Solution Obtained	Stage Reliability
1	Refrigerant Pump	2	.9999996	1	.99991513
2	Pump Control	1	.99999981	1	.99999981
3	Refrigerant Evaporator	0	.99998	0	.99998
4	Evap. back press. cont.	1	.99999981	1	.99999981
5	Evap. back press. valve	0	.999998	0	.999998
6	Coolant heat exch.	0	.99998	0	.99998
7	Cabin heat exch.	0	.99998	0	.99998
8	Cabin Temp. cont.	2	.99999966	2	.99991513
9	Evap. water control	3	.99999976	2	.99999976
10	Heat exch. diverter	3	.99999962	2	.99997264
11	Ref. temp. cont. valve	2	.99999962	2	.99999962
12	Ref. shutoff valve	1	.9994494	2	.9999938
13	Ref. check valve	2	.9999913	2	.9999913
14	Selector valve	2	.9999913	2	.9999913
System	Weight	20.3		17.6	
	Reliability	.999367927		.99971633	

NARC - North American Rockwell Corporation, Space Division.

4.3 OPTIMUM SOLUTION OF CRYOGENIC OXYGEN SUPPLY CIRCUIT

It is one of the two circuits, namely Cryogenic oxygen supply and High-pressure oxygen supply, capable of supplying the oxygen to the atmosphere of the cabin. The basic source of supply is the cryogenically stored oxygen. The high pressure oxygen supply is provided for the emergency repressurization capability which the relatively slow gas flow characteristics of cryogenic supply does not provide. The circuit included sensors, regulators and valves etc.

A. STATEMENT OF THE PROBLEM

The cryogenic oxygen supply circuit has 11 components, which were studied and tested minutely. The reliabilities of the components presented in Table 5 were previously estimated in [29]. The reliabilities of the components vary from 0.7 to 0.9998. The system without any supporting unit has a reliability of 0.458.

The problem formulated in integer programming has 11 decision variables and one constraint which is the minimum system reliability to be satisfied. A maximum number of supporting units allowed at each stage is assumed to be 5. The problem is reduced to zero or one programming problem. The size of the zero or one programming problem becomes 11x6 = 66 variables and two constraints where the constraint is added to ensure that there is one unit of each type (see section 4.2). The formulation of cryogenic oxygen supply circuit in the zero - one integer programming is presented in Table 6.

B. OPTIMUM SOLUTION AND DISCUSSION

The solution obtained is presented in Table 7 along with that suggested in ref. [29]. The method used to obtain allocation scheme

Table 5. Reliability Analysis of Cryogenic Oxygen Supply Circuit of Life Support Systems [29]

No.	Component	Duty Cycle in Hrs.	Estimated Reliability	Spare Weight
1	O_2 Partial Pressure Control	16,800	.9994	.8
2	O_2 Partial Pressure Sensor	16,800	.9994	.2
3	Display	16,800	.9998	2.0
4	Regulator	16,800	.9994	1.0
5	Check Valve	16,800	.987	.1
6	Shutoff Valve	16,800	.990	.4
7	Selector Valve	16,800	.928	1.4
8	Heater Control	16,800	.773	.9
9	Relief Valve	16,800	.936	2.4
10	Pressure Transducer	16,800	.700	.5
11	Cryogenic O_2 Tank	16,800	1.0	-
	System Reliability		.458	

TABLE 6. ZERO-ONE INTEGER PROGRAMMING FORMULATION OF CRYOGENIC OXYGEN SUPPLY CIRCUIT (CONTINUED)

J 1	K 2	I 3	COST COEF 4	LOG R(I) 5	 6
7	0	37	0.00	0.7472E-01	1
7	1	38	1.40	-0.7206E-01	0
7	2	39	1.40	-0.2594E-02	0
7	3	40	1.40	-0.6453E-04	0
7	4	41	1.40	-0.1205E-05	0
7	5	42	1.40	-0.1802E-07	0
8	0	43	0.00	0.2575	1
8	1	44	.90	-0.2291	0
8	2	45	.90	-0.2602E-01	0
8	3	46	.90	-0.2202E-02	0
8	4	47	.90	-0.1416E-03	0
8	5	48	.90	-0.7289E-05	0
9	0	49	0.00	0.6614E-01	1
9	1	50	2.40	-0.6404E-01	0
9	2	51	2.40	-0.2049E-02	0
9	3	52	2.40	-0.4514E-04	0
9	4	53	2.40	-0.7463E-06	0
9	5	54	2.40	-0.9872E-08	0
10	0	55	0.00	0.3567	1
10	1	56	.50	-0.3050	0
10	2	57	.50	-0.4582E-01	0
10	3	58	.50	-0.5311E-02	0
10	4	59	.50	-0.4722E-03	0
10	5	60	.50	-0.3367E-04	0
11	0	61	0.00	0.0000	1
11	1	62	100.00	0.0000	0
11	2	63	100.00	0.0000	0
11	3	64	100.00	0.0000	0
11	4	65	100.00	0.0000	0
11	5	66	100.00	0.0000	0

Table 7. Comparison of Optimal Number of Spares for Cryogenic Oxygen Supply Circuit of Life Support System

No.	Components	Spares Allocation by NARC [29]	Stage Reliability	Solution Obtained	Stage Reliability
1	O_2 Partial Pressure Control	1	.99999981	1	.99999981
2	O_2 Partial Pressure Control	1	.99999981	1	.99999981
3	Display	1	.99999997	1	.99999997
4	Regulator	1	.99999981	1	.99999981
5	Check Valve	2	.99999963	2	.99999963
6	Shutoff Valve	2	.99999982	2	.99999982
7	Selector Valve	3	.99999987	3	.99999987
8	Heater Control	4	.9999924	4	.9999924
9	Relief Valve	3	.99999928	3	.99999928
10	Pressure Transducer	5	.9999978	4	.99997422
11	Cryogenic O_2 Tank	-	1.00	-	1.00
System	Weight	22.5		22.0	
	Reliability	.9999882		.9999646	

NARC - North American Rockwell Corporation, Space Division.

of spares is presented and illustrated in Appendix A. In the spare allocation obtained by the method discussed in this report, the spares weigh 22 lb. The reference [29] suggests an allocation of spares weighing 22.5 lb. to achieve the same level of reliability, that is, 0.99996. There is no improvement in the weight of system spares because the original design seems to quite near optimal.

4.4 OPTIMUM SOLUTION OF CARBONIZATION CELL CIRCUIT

It is the circuit which handles the CO_2 of the atmosphere of the cabin. It is CO_2 concentrator, and an electrochemical device removes the CO_2 from cabin atmosphere. From the atmospheric gases O_2 is removed and recycled to the atmosphere and about 90% concentrated CO_2 is dumped overboard into the space. In the event of loss of access to the O_2 source or an inadequate remaining supply, the cabin CO_2 will then become an input to the oxygen regeneration function.

A. STATEMENT OF THE PROBLEM

The circuit of carbonization cell has 17 types of components. The estimated reliability characteristics are listed in Table 8. The system has reliability of 0.861 without any supporting units. When formulated in integer programming formulation the problem involved 17 decision variables and are reliability constraint. A maximum number of supporting units allowed at each stage is assumed to be 5. The problem was reduced to a zero or one programming problem. The size if this problem was 17 x 6 = 102 variables and two constraints where the constraint added is to ensure are unit of each type in the optimal system (see section 4.2).

Table 8 . Carbonization Cell Analysis [29]

No.	Components	Duty Cycle Hrs.	Weight	Reliability
1	Stage	1680	10.0	.999999
2	CO_2 Condenser	"	10.0	.99998
3	Condenser	"	10.0	.99998
4	Cationic Exchanger	"	10.0	.99996
5	Anionic Exchanger	"	10.0	.99996
6	Charcoal bed	"	0.3	.999995
7	Filter	"	10.0	.99986
8	Water Tank	"	10.0	.999995
9	CO_2 Tank	"	10.0	.999999
10	Fan	"	1.3	.987
11	Water Pump	"	2.0	.987
12	CO_2 Pump	"	5.0	.963
13	Relief Valve	"	2.4	.936
14	Selector Valve	"	0.6	.981
15	Water Meter	"	10.0	.999996
16	Shutoff Valve	"	10.0	.999999
17	Check Valve	"	10.0	.999999
	System Reliability			.861

B. OPTIMUM SOLUTION AND DISCUSSION

Table 9 presents the solutions obtained in report [29] along with that by present work. In present work the weight of spares required is 25.3 lb. whereas the report [29] suggests an allocation requiring 27.9 lb. of spares to achieve the system reliability of 0.99980. The present system allocation achieves a higher level of reliability.

4.5 OPTIMUM SOLUTION OF WATER RECLAMATION LOOP

It is the circuit which reclaims water from the waste and used water. The involved process is very simple, the water is evaporated at a definite pressure and is condensed. In case of emergency the reserve stock of water is used.

A. STATEMENT OF THE PROBLEM

The water reclamation loop consists of 8 different type of components. The weight and failure characteristics of the components are listed in Table 10. These values were taken from ref. [29]. The reliabilities of the components vary from 0.973 to 0.99999. The water reclamation loop has a reliability of 0.838 without any supporting item. The target is to increase the reliability of this loop to at least 0.9999.

The integer programming formulation of this problem consists of 8 decision variables and one constraint which is the minimum subsystem reliability to be satisfied. A maximum number of supporting units allowed at each stage is assumed to be 5. The problem is reduced to zero or one programming problem. The size of the zero or one programming problem becomes 8 x 6 = 48 variables and two constraints, where the constraint added is to ensure one unit of each type in the optimal system (see section 4.2).

Table 9 . Comparison of Optimal Number of Spares for Carbonization Cell of Life Support Systems

No.	Components	Spares Allocation by NARC [29]	Stage Reliability	Solution Obtained	Stage Reliability
1	Stage	0	.999999	0	.99999
2	CO_2 Condenser	0	.99998	0	.99998
3	Condenser	0	.99998	0	.99998
4	Cationic Exchanger	0	.99996	0	.99996
5	Anionic Exchanger	0	.99996	0	.99996
6	Charcoal bed	0	.999995	0	.999995
7	Filter	1	.9999999	1	.9999999
8	Water Tank	0	.999995	0	.999995
9	CO_2 Tank	0	.999999	0	.999999
10	Fan	2	.9999995	2	.9999995
11	Water Pump	2	.9999995	2	.9999995
12	CO_2 Pump	3	.9999999	2	.999991316
13	Relief Valve	2	.9999541	3	.99999924
14	Selector Valve	2	.9999988	2	.9999988
15	Water Meter	0	.999996	0	.9999996
16	Shutoff Valve	0	.999999	0	.999999
17	Check Valve	0	.999999	0	.999999
System	Weight	27.9		25.3	
	Reliability	.9998138		.99985036	

NARC - North American Rockwell Corporation, Space Division.

Table 10. Water Reclamation Loop Analysis [29]

No.	Component Type	Duty Cycle Hrs.	Weight	Reliability
1	Condenser	16800	5.2	.994
2	Recuperator	"	5.2	.994
3	Pyro Reactor	"	1.73	.974
4	Evaporator	"	5.2	.994
5	Pressure Regulator	"	1.0	.9994
6	Heater Control	"	0.9	.973
7	O_2 Flow Control	"	10.0	.99999
8	Check Valve	"	0.1	.946
	System Reliability			0.838

Note - Weight for item 7 has been assumed.

B. OPTIMUM SOLUTION AND DISCUSSION

The solution obtained is presented in Table 11 along with that suggested in ref. [29]. The method used to obtain allocation scheme of spares in ref. [29] is presented and illustrated in Appendix A.

In present system the weight of the spares required is 22.57 lb. whereas report [29] suggests an allocation system requiring 24.8 lb. of spares to achieve the system reliability of 0.99992. The present method of optimization of system reliability is a better one, it resulted in a better arrangement of spares, the weight of spares being less than that presented in reference [29] to achieve the same level of reliability.

Table 11. Comparison of Optimal Number of Spares for Water Reclamation Circuit

No.	Components	Spares Allocation by NARC [29]	Stage Reliability	Solution Obtained	Stage Reliability
1	Condenser	1	.99998	1	.99998
2	Recuperator	1	.99998	1	.99998
3	Pyro Reactor	3	.9999999	2	.999987
4	Evaporator	1	.99998	1	.99998
5	Pressure Regulator	1	.9999998	1	.9999998
6	Heater Control	3	.9999999	2	.9999966
7	O_2 Flow Control	0	.99999	0	.99999
8	Check Valve	3	.999999	3	.999999
System	Weight	24.8		22.57	
	Reliability	.9999286		.9999224	

NARC - North American Rockwell Corporation, Space Division.

5 PROPOSALS FOR FUTURE STUDIES

Problem 1.

The complete life support system should be optimized as a whole. So far an attempt has been made to design optimum subsystems. To optimize the life support systems as a whole, the reliability levels for each subsystems must be assigned in a systematic way. In reference [29] the reliability levels are assigned such as to make each subsystem equally reliable. It is proposed here to use some search technique to assign the reliability levels to subsystems so that the overall system is optimized.

Consider an example of a system with two subsystems. The system is required to have a reliability of at least 0.64. If an equal reliability concept is used, as in ref. [29], the specifications would be 0.8 for each of the subsystems. The objective function to be minimized is the total weight of the system. Now the specifications for reliability of the subsystems can be altered to the following combination which yields the reliability of whole system equal to 0.64:

(i) 0.90 and .711

(ii) 0.85 and .753

(iii) 0.75 and .853

(iv) 0.70 and .914

An overall minimum value of objective function can be achieved by designing the each of the subsystems for these specifications. Same search technique, for example, golden section for this example can be used to determine the optimum specifications.

Problem 2.

The formulation presented in this report will obtain a global optimum only for convex objective functions. To incorporate the capability of arriving at an optimum in a multimodal objective function the following restrictions must be added. For a given j

$$y_{jk} \geq y_{j(k+1)}, \qquad \text{for } k = 0, \ldots, M_j.$$

This insures the inclusion in optimal solution of the kth unit at jth stage before the (k+1)th unit.

The addition of these constrains, $[N \cdot (M_j - 1)]$ in number, increases the size of the problem. Hence these should be added whenever the nature of the objective function is not definitely convex. When the system reliability is the objective function to be maximized the function is certainly convex and these restrictions are redundant. But after addition of these constraints the optimization method can attain an optimum even for highly nonlinear functions, and the method becomes much more general in nature.

Problem 3.

In this report it has been mentioned that the integer programming solution will be better as compared to one obtained by rounding off the continuous variables. It will be quite reasonable to find the difference between the two solutions. The same problems may be solved by conventional methods taking the decision variables continuous and then comparing the solution with those obtained by the formulation and program suggested in this report. A comparison of the two will bring out the validity of the statement. It is a fact that integer programming will come out with a better solution, but one can determine in how many cases both obtain the similar solution.

Thus arriving at a efficiency of the method.

Problem 4.

If the system can fail in more than one ways the problem becomes more difficult to solve. Such problems have been solved by a number of people for exam see Tillman [41]. They have used methods other than zero - one integer programming. It is proposed here to solve this problem using zero - one integer programming. It is expected that it would be an efficient tool and can be used for larger and more complicated problems than those solved so far.

Problem 5.

In this report spares have been considered as a method of improving system reliability. A failure sensing device is used to detect the component failure, the reliability of which has been assumed to be one, which is not a very good assumption. If the actual reliability of the failure sensing device can be estimated, it should be considered in the formulation. Such a system can be compared to a system having parallel redundancies without failure sensing devices. A comparison will indicate which systems are more economical. It is expected that in certain situations it would be economical to put the supporting units as spares whereas in the others parallel redundancy may prove to be a better means of improving system reliability.

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. C. L. Hwang, major professor, for his sincere guidance, valuable comments, constructive criticism, deep interest, and careful review of this report.

The author would also like to express his appreciation to Dr. L. T. Fan for valuable discussions and to Dr. F. A. Tillman for his assistance in all the possible ways at every step.

The author also thanks Dr. Ashour who helped the author to obtain the computer program for Implicit Enumeration technique.

This study was supported in part by NASA Grant No. NGR-17-001-034, and by the Air Force Office of Scientific Research, Office of Aerospace Research, United States Air Force, Under Contract F44620-68-0020 (Themis Project).

BIBLIOGRAPHY

1. ARNIC, Reliability Engineering, Prentice-Hall, Englewood Cliffs, New Jersey, 1958.

2. Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables", Operations Research, Vol. 13, No. 4, pp. 517-549 (1965).

2a. Balinski, M. L., "Integer Programming: Methods, Uses, Computation," Management Science, Vol. 12, No. 3, pp. 253-313 (1965).

2b. Balinski, M. L., "On Recent Developments in Integer Programming," Proceedings of the International Symposium on Mathematical Programming, Princeton University Press, Princeton, New Jersey (1967).

3. Bazovsky, I., Reliability Theory and Practice, Prentice-Hall, Englewood Cliffs, New Jersey, 1961.

3a. Beale, E. M. L., "Survey of Integer Programming", Operational Research Quarterly (U.K.), Vol. 16, No. 2, pp. 219-228 (1965).

4. Bellman, R., and S. Dreyfus, "Dynamic Programming and the Reliability of Multicomponent Devices", Operations Research, Vol. 6, No. 2, pp. 200-206 (1958).

4a. Dantzig, G. B., "On the Significance of Solving Linear Programming Problems with Some Integer Variables", Econometrica, Vol. 28, No. 1, pp. 30-44 (1960).

5. Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "Solution of A Large-Scale Travelling Salesman Problem", Operations Research, Vol. 2, No. 2, pp. 393-410 (1954).

6. Ditoro, M. J., "Reliability Criterion for Constrained Systems", IRE Transactions on Reliability and Quality Control, Vol. PGRQC-8, pp. 1-6 (1956).

7. Epstein, B., "Estimation from Life Test Data", IRE Transactions on Reliability and Quality Control, Vol. RQC-9, pp. 104-108 (1960).

8. Eto, H., "An Algorithm for Integer Linear Programming by Parameteric Modification of an Added Constraint", Report ORC 67-32, Operations Research Center, University of California, Berkeley, California, 1967.

9. Fan, L. T., C. S. Wang, F. A. Tillman, and C. L. Hwang, "Optimization of Systems Reliability", IEEE Transactions on Reliability and Quality Control, Vol. R-16, No. 2, pp. 81-86 (1967).

9a. Federal Electric Corp., Bureau of Ships Reliability Design Handbook, NAVSHIPS 94501, Prepared for Fleet Electronics Effectiveness Branch, Bureau of Ships, Department of the Navy, by Equipment and Systems Evaluation Department, Engineering Division, Federal Electric Corp. Paramus Industrial Park, Paramus, New Jersey, 1963. (available from U.S. Government Printing Office Washington, D.C. 20302).

10. Geoffrion, A. M., "Implicit Enumeration using an Imbedded Linear Program", Working Paper No. 120, Western Management Science Institute, University of California, Los Angeles, California, 1965.

11. Geoffrion, A. M., "An Improved Implicit Enumeration Approach for Integer Programming", Operations Research, Vol. 17, No. 3, pp. 437-454 (1969).

11a. Glare, P. M., and R. E. Taylor, "Optimal Redundancy for Reliability in Series Systems", Operations Research, Vol. 17, No. 7, pp. 838-847 (1969).

12. Glass, H., "The Zero-One Problem", Paper Presented at the 28th National Meeting of the Operations Research Society of America, Houston (1965).

13. Glover, F., "A Multiphase Dual Algorithm for the Zero-One Integer Programming Problem", Operations Research, Vol. 13, No. 6, pp. 879-919 (1965).

14. Gomory, R. E., "Outlines of An Algorithm for Integer Solutions to Linear Programs", Research Announcements, Bulletin of the American Mathematical Society, Vol. 64, No. 5 (1958).

15. Gomory, R. E., "An All Integer Programming Algorithm", Princeton-IBM Mathematics Research Report, Nov. 1958. Included in Industrial Scheduling, J. F. Muth and G. L. Thomson Eds, Prentice Hall, Englewood Cliffs., New Jersey, 1963.

16. Hadley, G., Linear Programming, Addison Wesley, Reading, Mass., 1963.

17. Hadley, G., "Non-Linear and Dynamic Programming, Addison Wesley, Reading, Mass., 1964.

18. Hees, R. N., and H. W. Meerendonk, "Optimal Reliability of Parallel Multicomponent Systems", Operations Research Quarterly (U.K.), Vol. 12, No. 1, pp. 16-26 (1961).

19. Hillier, F. S., and G. L. Lieberman, Introduction to Operations Research, Holden-Day, San Francisco, Cal., 1961.

20. Kaplan, S., "Solution of Lorie-Savage and Similar Integer Programming Problems by the Generalized Lagrange Multipliers Method", Operations Research, Vol. 14, No. 6, pp. 1130-1136, (1966).

21. Kettelle, J. D., "Least Cost Allocations of Reliability Investment" Operations Research, Vol. 10, No. 2, pp. 249-265 (1962).

22. Kolesar, P. G., "Linear Programming and the Reliability of Multi-component Systems", Naval Research Logistics Quarterly, Vol. 14, No. 3, pp. 317-327 (1967).

23. Kuehn, A. A., and M. J. Hamburger, "A Heuristic Program for Locating Warehouses", Management Science, Vol. 9, No. , pp. 643-666 (1963).

24. Land, A. H., and A. C. Doig, "An Automatic Method of Solving Discrete Programming Problems", Econometrica, Vol. 28, pp. 497-520 (1960).

24a. Landers, R. R., Reliability and Product Assurance, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

25. Lemke, C. E., and K. Spielberg, "Direct Search Algorithms for Zero-One and mixed Integer Programming", Operations Research, Vol. 15, No. 5, pp. 892-914 (1967).

26. Little, J. D. C., K. G. Murthy, D. W. Sweeney, and C. Karel, "An Algorithm for Travelling Salesman Problem", Operations Research, Vol. 11, No. 6, pp. 972-989 (1963).

27. Lloyd, D. K., and M. Liplow, Reliability: Management, Methods and Mathematics, Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

27a. McDonnel Douglas Astronautics Company, "Parameteric Study of Manned Life Support Systems Final Briefing Brochure," Prepared for National Aeronautics and Space Administration, Mission Analysis Division, by McDonnell Douglas Astronautics Company - Western Division, 1968.

28. Moskowitz, F., and J. B. McLean, "Some Reliability aspects of System Design", IRE Transactions on Reliability and Quality Control, Vol. PGRQC-8, pp. 7-36 (1956).

29. North American Rockwell Corporation, "A Study of Space Mission Duration Extension Problems", Vol. III, Prepared for NASA by Space Division of North American Rockwell Corporation, Report No. SD 67-478-3, 1967.

30. Polovko, A. M., Fundamentals of Reliability Theory, Academic Press, New York, 1968.

30a. Pieruschka, E., Principles of Reliability, Prentice-Hall, Englewood, New Jersey, 1963.

31. Proschan, F., and T. A. Bray, "Optimum Redundancy Under Multiple Constraints", Operations Research, Vol. 13, No. 5, pp. 800-814 (1965).

32. Rao, A., "Zero-One Integer Linear Programming", Ph.D. Thesis, University of Iowa, 1968.

33. Rao, V. B., "Some Approaches to Integer Programming", Ph.D. Thesis, Washington University, St. Louis, Missouri (1967).

34. Roth, R., "Computer Solutions to Minimum- Cover Problems", Operations Research, Vol. 17, No. 3, pp 455-465 (1969).

35. Saaty, T. L., "On Nonlinear Optimization in Integers", Naval Research Logistics Quarterly, Vol. 15, No. 1, pp. 1-22 (1958).

36. Salklin, H., and K. Spielberg, "Addaptive Binary Programming", IBM Scientific Research Centre, 410 East 62nd St. New York, N.Y., Report No. 320-2951, March 1968.

36a. Sandler, G. H., System Reliability Engineering, Prentice Hall, Englewood Cliffs, New Jersey, 1963.

36b. Shooman, M. L., Probabilistic Reliability Engineering, Prentice Hall, Englewood Cliffs, New Jersey, 1963.

37. Simonnard, M., Linear Programming , Translation by W. S. Jewell, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

38. Tillman, F. A., "Integer Programming Solutions to Reliability Optimization Problems Subject to Linear and Nonlinear Separable Constraints", Ph.D. Thesis, State University of Iowa, 1965.

39. Tillman, F. A., and Littschwager, "Integer Programming Formulations of Constrained Reliability Problems", Management Science, Vol. 13, No. 11, pp. 887-899 (1967).

40. Tillman, F. A., C. L. Hwang, L. T. Fan, and S. A. Balbale, "Systems Reliability Subject to Multiple Nonlinear Constraints", IEEE Transactions on Reliability, Vol. 17, No. 3, pp. 153-157 (1968).

41. Tillman, F. A., "Optimization by Integer Programming of Constrained Reliability Problems with Several Failure Modes", IEEE Transactions on Reliability, Vol. R-18, No. 2, pp. 47-53 (1969).

42. Young, R. D., "A Primal (All Integer) Integer Programming Algorithm", Journal of Research of the National Bureau of Standards, No. 698, pp. 213-250 (1965).

APPENDIX A

SPARES ALLOCATION PROCEDURE OF THE REPORT [29]

The work in the report [29] employs a principle of "Equal risk within a function", which is stated as, 'Ideally, each item in a system with its provided maintenance capability, should present the same risk or probability of causing the subsystem to become non-functional by virtue of its lack of supporting spares'. To illustrate the application of this principle a example is given, which is summarized below.

Consider subsystem of four different kinds of items A,B,C and D, shown in Fig. A-1. This will have the reliability

$$P_s = R_A\ R_B(1-(1-R_A\ R_C\ R_D) \cdot (1-(1-R_B)^2))$$

$$= R_A\ R_B - 2R_A\ R_B^2 + 2R_A^2\ R_C\ R_D R_B^2 - R_A\ R_B^3 + R_A^2\ R_B^3\ R_C\ R_D$$

If the concept of equalized risk is applied

$$R_i = R_A = R_B = R_C = R_D$$

and now

$$P_s = R_i^2 - 2R_i^3 - R_i^4 + 2R_i^6 + R_i^7$$

For a given $P_s = 0.8636$, the item contribution must be $R_i = 0.9870$ or better by solving the above equation.

A maintenance logic diagram was presented for determining the number of supporting units which give the item reliability of 0.987. The system can be represented by equivalent items as shown in Fig. A-2. It is possible to simplify this diagram to Fig. A-3, since item A1 and A2 are physically identical, their failure rates are also equal

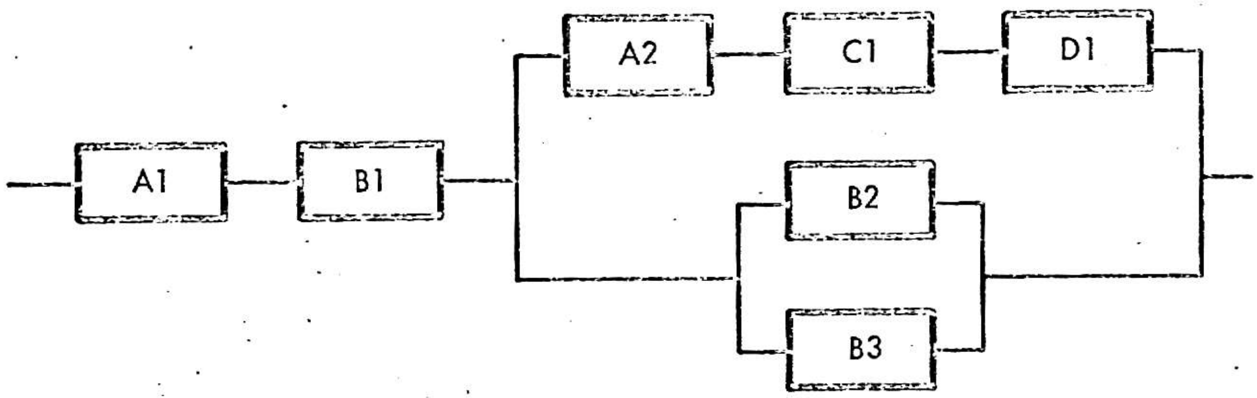

Figure A-1 Example Subsystem of Four Different Items, A, B, C, D

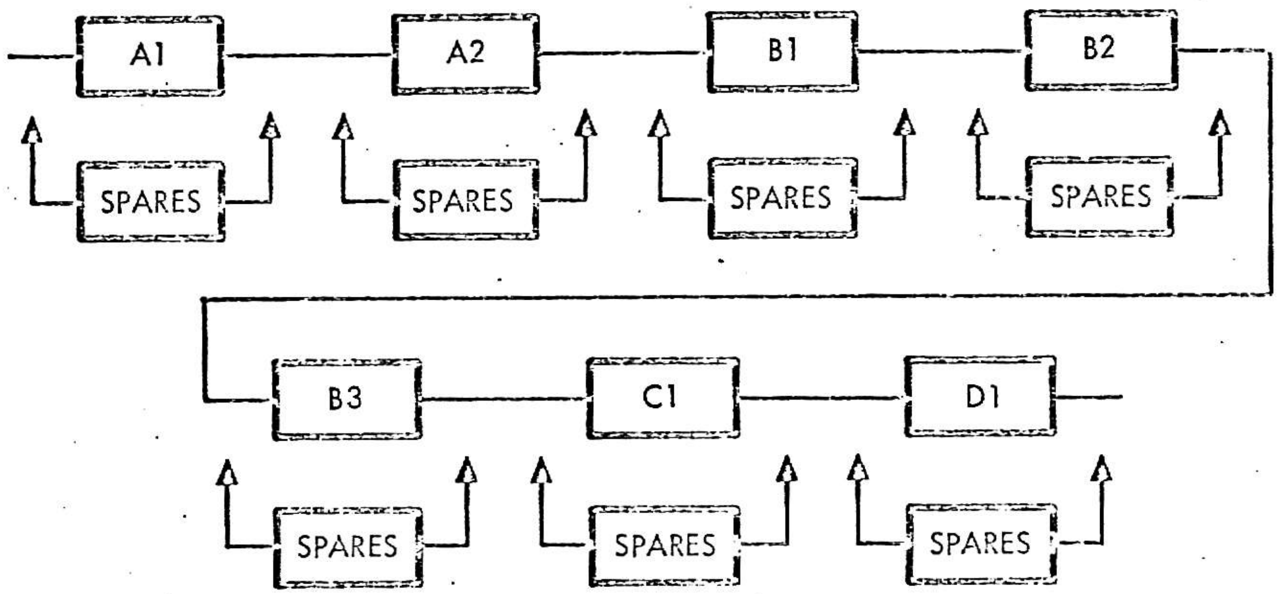

Figure A-2 Maintenance Logic Diagram [29]

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

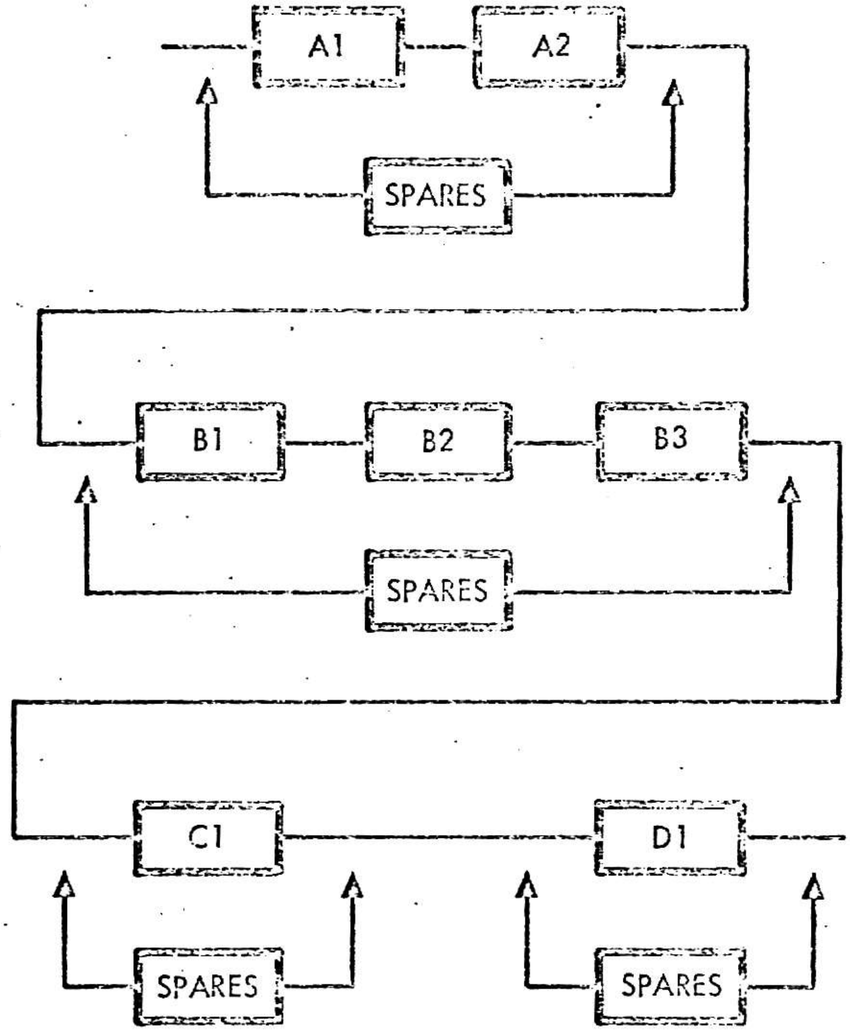

Figure A-3 **Use of an A Type Spare**

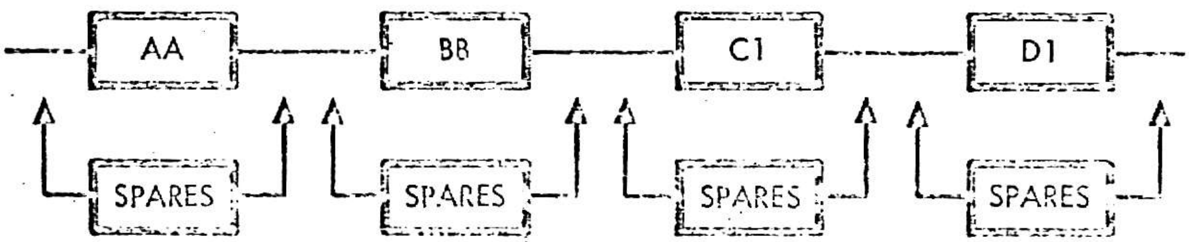

Figure A-4 **Series Maintenance Logic Diagram [29]**

to λ_A. Their duty cycle may be different but to put a equivalent item for items A1 and A2, the expected failure rate of equivalent item must be determined. Any spare of A type can be used at the place of A1 or A2.

To utilize the spares of type A the items A1 and A2 are replaced by item AA, as shown in Fig. A.4. To determine the spare requirements for equivalent item AA, the following equation must be satisfied

$$\lambda_{AA}\, t_{AA} = \lambda_{A1}\, t_{A1} + \lambda_{A2}\, t_{A2}$$

since

$$\lambda_{A1} = \lambda_{A2}$$

$$\lambda_{AA}\, t_{AA} = \lambda_{A}\, (t_{A1} + t_{A2})$$

similarly

$$\lambda_{BB}\, t_{BB} = \lambda_{B}\, (t_{B1} + t_{B2} + t_{B3})$$

Thus a purely series diagram (Figs. A-3, A-4) may be developed to represent the series parallel logic diagram of Fig. A-1. To illustrate the procedure, consider the example with the characteristics given below

Item	Failure Rate	Duty Cycle
A1	19.6×10^{-6}	10^4
A2	19.6×10^{-6}	10^2
B1	30.0×10^{-6}	10^4
B2	30.0×10^{-6}	10
B3	30.0×10^{-6}	10
C	90.0×10^{-6}	10^2
D	80.0×10^{-6}	10^2

This data gives the values for equivalent items as follows

Item	λ	t	λt
AA	19.6×10^{-6}	10100	0.200
BB	30.0×10^{-6}	10020	0.300
C	90.0×10^{-6}	100	0.009
D	80.0×10^{-6}	100	0.008

Now referring to Fig. A-5, one finds the broken horizontal line represents the reliability level of 0.987 required for each item. The product of failure itensity and duty cycle duration for each equivalent item has been plotted as a vertical line. Lines for the reliability of spares have been plotted. An inspection of this diagram gives the following observational conclusions:

Item	Spare required	reliability
A1-A2	1 or 2	0.9823 or 0.9989
B1-B2-B3	1 or 2	0.965 or 0.9967
C	0 or 1	0.914 or 0.9963
D	0 or 1	0.924 or 0.9969

The selection of a proper number of spare items involves a judgement, but the list of value given above helps the designer to make a reasonably correct decision. This simplification and the determination of spares procedure cannot be termed as being exact, but can be used to determine a near optimal allocation of available resources.

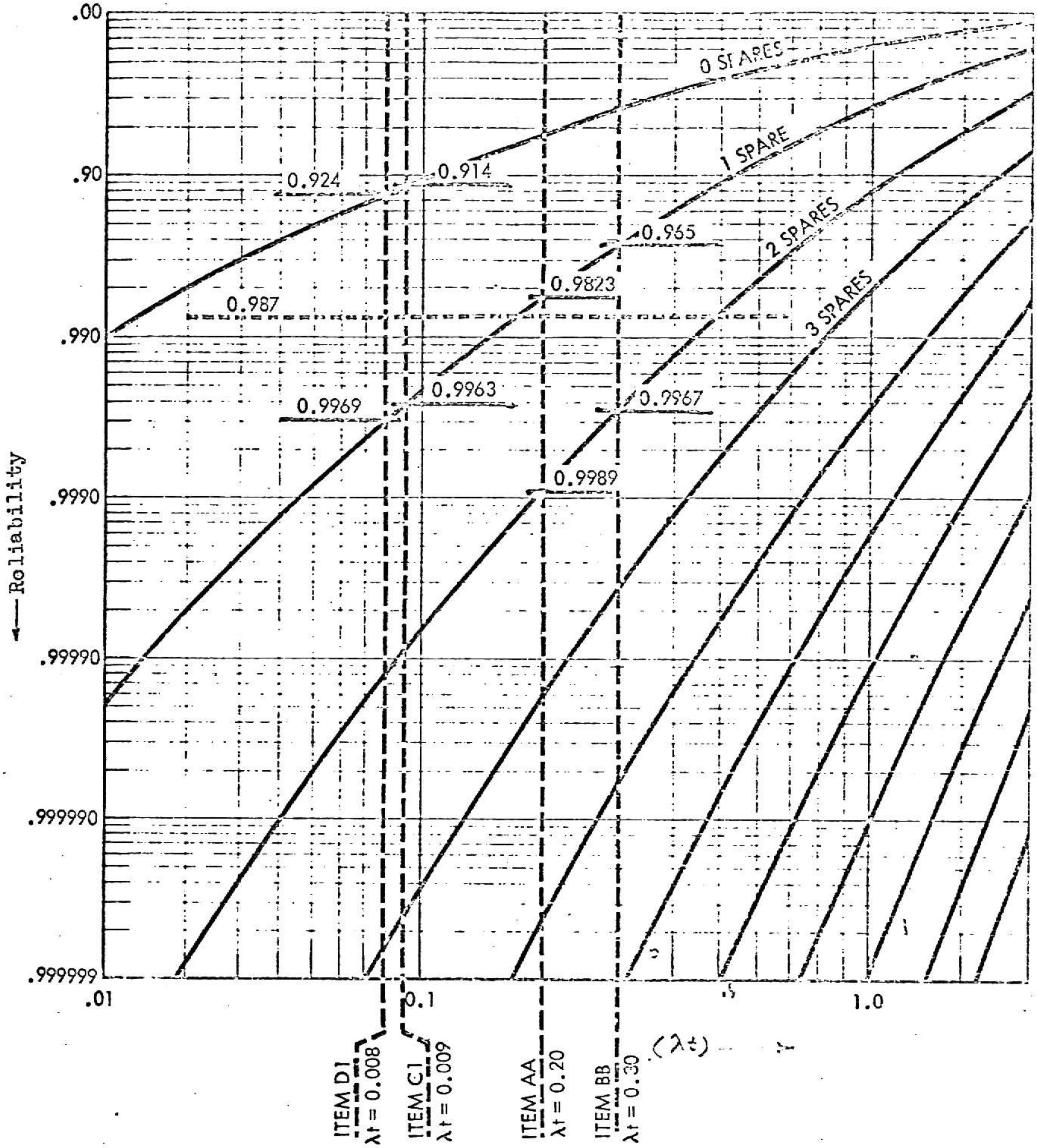

Figure A-5. A Series to Determine Spares Requirements [29]

APPENDIX B

The appendix containts:

1. Listing of the input to the computer program for the optimization of the system reliability of the refrigerant circuit of life support systems.
2. Output of the program for the optimization of system reliability of the refrigerant circuit of the life support systems.

The enclosed information is just sufficient for the problems considered in the report. To become familiar with the input procedure of zero - one integer programming computer program one should consult the IBM publication "Adaptive Binary Programming" [36].

LISTING OF THE INPUT TO THE DZLP PROGRAM FOR THE OPTIMIZATION OF THE
SYSTEM RELIABILITY OF REFRIGERANT CIRCUIT OF LIFE SUPPORT SYSTEMS (CONTINUED)

```
            2         84                                                                  DATA  10
                       1           1           1           1           1                  DATA  20
                                                                                          DATA  30
            1          1                                                                  DATA  40
                                                                                          DATA  50
            1  10000000                                                                   DATA  60
                 0.1500E 01  0.1500E 01  0.1500E 01  0.1500E 01  0.1500E 01DATA  70
                 0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01DATA  80
                 0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02DATA  90
                 0.4800E 01  0.4800E 01  0.4800E 01  0.4800E 01  0.4800E 01DATA 100
                 0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02DATA 110
                 0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02DATA 120
                 0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02DATA 130
                 0.9000E 00  0.9000E 00  0.9000E 00  0.9000E 00  0.9000E 00DATA 140
                 0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01DATA 150
                 0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01DATA 160
                 0.7500E 00  0.7500E 00  0.7500E 00  0.7500E 00  0.7500E 00DATA 170
                 0.8000E 00  0.8000E 00  0.8000E 00  0.8000E 00  0.8000E 00DATA 180
                 0.1000E 00  0.1000E 00  0.1000E 00  0.1000E 00  0.1000E 00DATA 190
                 0.6000E 00  0.6000E 00  0.6000E 00  0.6000E 00  0.6000E 00DATA 200
 0.2800E- 3 -0.1400E  2                                                            DATA 210
 0.1309E-01 -0.1300E-01 -0.8450E-04 -0.3686E-06 -0.1206E-08 -0.3155E-11DATA 220
 0.6002E-03 -0.6000E-03 -0.1800E-06 -0.3601E-10 -0.5398E-14 -0.0000E 00DATA 230
 0.2000E-04 -0.2000E-04 -0.2000E-09 -0.1332E-14 -0.0000E 00 -0.0000E 00DATA 240
 0.6002E-03 -0.6000E-03 -0.1800E-06 -0.3601E-10 -0.5398E-14 -0.0000E 00DATA 250
 0.2000E-05 -0.2000E-05 -0.2000E-11 -0.0000E 00 -0.0000E 00 -0.0000E 00DATA 260
 0.2000E-04 -0.2000E-04 -0.2000E-09 -0.1332E-14 -0.0000E 00 -0.0000E 00DATA 270
 0.2000E-04 -0.2000E-04 -0.2000E-09 -0.1332E-14 -0.0000E 00 -0.0000E 00DATA 280
 0.1309E-01 -0.1300E-01 -0.8450E-04 -0.3686E-06 -0.1206E-08 -0.3155E-11DATA 290
 0.4919E-01 -0.4802E-01 -0.1152E-02 -0.1889E-04 -0.2322E-06 -0.2285E-08DATA 300
```

```
0.5551E-01 -0.5403E-01 -0.1459E-02 -0.2697E-04 -0.3743E-06 -0.4156E-08DATA 310
0.1309E-01 -0.1300E-01 -0.8450E-04 -0.3686E-06 -0.1206E-08 -0.3155E-11DATA 320
0.3356E-01 -0.3301E-01 -0.5446E-03 -0.6090E-05 -0.5109E-07 -0.3429E-09DATA 330
0.3770E-01 -0.3701E-01 -0.6847E-03 -0.8601E-05 -0.8107E-07 -0.6113E-09DATA 340
0.3770E-01 -0.3701E-01 -0.6847E-03 -0.8601E-05 -0.8107E-07 -0.6113E-09DATA 350
-.1000E  1                                                            DATA 360
-.1000E  1                                                            DATA 370
-.1000E  1                                                            DATA 380
-.1000E  1                                                            DATA 390
-.1000E  1                                                            DATA 400
-.1000E  1                                                            DATA 410
-.1000E  1                                                            DATA 420
-.1000E  1                                                            DATA 430
-.1000E  1                                                            DATA 440
-.1000E  1                                                            DATA 450
-.1000E  1                                                            DATA 460
-.1000E  1                                                            DATA 470
-.1000E  1                                                            DATA 480
-.1000E  1                                                            DATA 490
                                                                      DATA 500
                                                                      DATA 510
                                                                      DATA 520
     1        -1         1        -1         1        -1         1    DATA 530
    -1         1        -1         1        -1        -1         1    DATA 540
    -1        -1         1        -1        -1         1        -1    DATA 550
    -1        -1        -1         1        -1        -1        -1    DATA 560
    -1        -1        -1        -1        -1        -1        -1    DATA 570
    -1        -1        -1        -1        -1        -1        -1    DATA 580
    -1        -1        -1        -1        -1        -1        -1    DATA 590
     1        -1         1        -1         1        -1         1    DATA 600
    -1         1        -1         1        -1        -1         1    DATA 610
    -1        -1         1        -1        -1         1        -1    DATA 620
    -1        -1        -1         1        -1        -1        -1    DATA 630
    -1        -1        -1        -1        -1        -1        -1    DATA 640
/*                                                                    DATA 650
FOR DETAIL OF THE PROGRAM SEE REFERENCE  29
```

```
************************************************

CONSTRAINT MATRIX HAS        2 ROWS AND       84 COLUMNS

************************************************

COST VECTOR
         0.0        0.1500E 01  0.1500E 01  0.1500E 01  0.1500E 01  0.1500E 01
         0.0        0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01
         0.0        0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02
         0.0        0.4800E 01  0.4800E 01  0.4800E 01  0.4800E 01  0.4800E 01
         0.0        0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02
         0.0        0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02
         0.0        0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02
         0.0        0.9000E 00  0.9000E 00  0.9000E 00  0.9000E 00  0.9000E 00
         0.0        0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01
         0.0        0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01
         0.0        0.7500E 00  0.7500E 00  0.7500E 00  0.7500E 00  0.7500E 00
         0.0        0.8000E 00  0.8000E 00  0.8000E 00  0.8000E 00  0.8000E 00
         0.0        0.1000E 00  0.1000E 00  0.1000E 00  0.1000E 00  0.1000E 00
         0.0        0.6000E 00  0.6000E 00  0.6000E 00  0.6000E 00  0.6000E 00

RIGHT HAND SIDE
         0.2800E-03 -0.1400E 02

CONSTRAINT MATRIX
         0.1309E-01 -0.1300E-01 -0.8450E-04 -0.3686E-06 -0.1206E-08 -0.3155E-11
         0.6002E-03 -0.6000E-03 -0.1800E-06 -0.3601E-10 -0.5398E-14  0.0
         0.2000E-04 -0.2000E-04 -0.2000E-09 -0.1332E-14  0.0         0.0
         0.6002E-03 -0.6000E-03 -0.1800E-06 -0.3601E-10 -0.5398E-14  0.0
         0.2000E-05 -0.2000E-05 -0.2000E-11  0.0         0.0         0.0
         0.2000E-04 -0.2000E-04 -0.2000E-09 -0.1332E-14  0.0         0.0
         0.2000E-04 -0.2000E-04 -0.2000E-09 -0.1332E-14  0.0         0.0
         0.1309E-01 -0.1300E-01 -0.8450E-04 -0.3686E-06 -0.1206E-08 -0.3155E-11
         0.4919E-01 -0.4802E-01 -0.1152E-02 -0.1889E-04 -0.2322E-06 -0.2285E-08
         0.5551E-01 -0.5403E-01 -0.1459E-02 -0.2697E-04 -0.3743E-06 -0.4156E-08
         0.1309E-01 -0.1300E-01 -0.8450E-04 -0.3686E-06 -0.1206E-08 -0.3155E-11
         0.3356E-01 -0.3301E-01 -0.5446E-03 -0.6090E-05 -0.5109E-07 -0.3429E-09
         0.3770E-01 -0.3701E-01 -0.6847E-03 -0.8601E-05 -0.8107E-07 -0.6113E-09
         0.3770E-01 -0.3701E-01 -0.6847E-03 -0.8601E-05 -0.8107E-07 -0.6113E-09
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0
        -0.1000E 01  0.0         0.0         0.0         0.0         0.0

************************************************

ADAPTIVE OPTION USED, SEARCH ORIGIN NOT FIXED AT 0
```

LP GENERATES FIRST ORIGIN VIA ROUNDUP SOLUTION

****

```
MAIN PROGRAM DATA

**************************************************

DATA AFTER COMPLEMENTATION AND ORDERING

TRANSLATION VECTOR
       20         2         8        50        51        56        57        44        45        68
       69        62        63        80        81        74        75         1         7        13
       19        25        31        37        43        49        55        61        67        73
       79        76        77        78        82        83        84        64        65        66
       70        71        72        46        47        48         9        10        11        12
       52        53        54        58        59        60         3         4         5         6
       21        22        23        24        14        15        16        17        18        26
       27        28        29        30        32        33        34        35        36        38
       39        40        41        42

CONSTANT TERM   -0.17600E 02

**************************************************

COST VECTOR
          -0.4800E 01 -0.1500E 01 -0.1000E 01 -0.1000E 01 -0.1000E 01 -0.1000E 01
          -0.1000E 01 -0.9000E 00 -0.9000E 00 -0.8000E 00 -0.8000E 00 -0.7500E 00
          -0.7500E 00 -0.6000E 00 -0.6000E 00 -0.1000E 00 -0.1000E 00  0.0
           0.0         0.0         0.0         0.0         0.0         0.0
           0.0         0.0         0.0         0.0         0.0         0.0
           0.0         0.1000E 00  0.1000E 00  0.1000E 00  0.6000E 00  0.6000E 00
           0.6000E 00  0.7500E 00  0.7500E 00  0.7500E 00  0.8000E 00  0.8000E 00
           0.8000E 00  0.9000E 00  0.9000E 00  0.9000E 00  0.1000E 01  0.1000E 01
           0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01
           0.1000E 01  0.1000E 01  0.1500E 01  0.1500E 01  0.1500E 01  0.1500E 01
           0.4800E 01  0.4800E 01  0.4800E 01  0.4800E 01  0.1000E 02  0.1000E 02
           0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02
           0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02
           0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02  0.1000E 02

RIGHT HAND SIDE
           0.6160E-04  0.0

CONSTRAINT MATRIX
           0.6000E-03  0.1300E-01  0.6000E-03  0.4802E-01  0.1152E-02  0.5403E-01
           0.1459E-02  0.1300E-01  0.8450E-04  0.3301E-01  0.5446E-03  0.1300E-01
           0.8450E-04  0.3701E-01  0.6847E-03  0.3701E-01  0.6847E-03 -0.1309E-01
          -0.6002E-03 -0.2000E-04 -0.6002E-03 -0.2000E-05 -0.2000E-04 -0.2000E-04
          -0.1309E-01 -0.4919E-01 -0.5551E-01 -0.1309E-01 -0.3356E-01 -0.3770E-01
          -0.3770E-01 -0.8601E-05 -0.8107E-07 -0.6113E-09 -0.8601E-05 -0.8107E-07
          -0.6113E-09 -0.3686E-06 -0.1206E-08 -0.3155E-11 -0.6090E-05 -0.5109E-07
          -0.3429E-09 -0.3686E-06 -0.1206E-08 -0.3155E-11 -0.1800E-06 -0.3601E-10
          -0.5398E-14  0.0        -0.1889E-04 -0.2322E-06 -0.2285E-08 -0.2697E-04
          -0.3743E-06 -0.4156E-08 -0.8450E-04 -0.3686E-06 -0.1206E-08 -0.3155E-11
          -0.1800E-06 -0.3601E-10 -0.5398E-14  0.0        -0.2000E-04 -0.2000E-09
          -0.1332E-14  0.0         0.0        -0.2000E-05 -0.2000E-11  0.0
           0.0         0.0        -0.2000E-04 -0.2000E-09 -0.1332E-14  0.0
           0.0        -0.2000E-04 -0.2000E-09 -0.1332E-14  0.0         0.0
           0.0         0.0         0.0         0.0         0.0         0.0
           0.0         0.0         0.0         0.0         0.0         0.0
           0.0         0.0         0.0         0.0         0.0         0.1000E 01
           0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01
           0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01  0.1000E 01
```

```
         0.1000E 01  0.0         0.0         0.0         0.0         0.0
         C.0         0.0         0.0         0.0         0.0         0.0
         C.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0
         0.0         0.0         0.0         0.0         0.0         0.0

***********************************************

MAXIMUM POSITIVES IN ROW (HMAXP)
   C.054      1.000

MAXIMUM NEGATIVES IN ROW (HMAXN)
  -C.C00**********

***********************************************

OUTPUT OF PROBLEM VARIABLES

NEW SUBOPTIMAL FEASIBLE SOLUTION AFTER ITERATIVE STEP NO.        0

OBJECTIVE FUNCTION =      0.175999603E 02

NON-ZERO COMPONENTS CF  X
       1         2         7         8        13        19        20        25        31        37
      43        44        45        49        50        51        55        56        57        61
      62        63        67        68        69        73        74        75        79        80
      81

***********************************************

***********************************************

     CURRENT PARAMETER AND COUNTER VALUES
     (SEE 2002 AREA CF MAIN FOR LABELS)
     2C00        1         0         0         0         0         0         0         0
        0       31        31         0         1         0         1         0
        0        0         0         0         0
        0        0         0         0         0
        1        0         1         1         1         1
        1        1        84        42         1         0         0         0
   C.0       0.0      -0.6560

     CURRENT SEQUENCE VECTOR (THOSE J S.T. YJ=1) (KSEQU)
        0

     CURRENT ORIGIN (THOSE J S.T. YJ=1) (KORIG)
        1        2         7         8        13        19        20        25        31        37
       43       44        45        49        5C        51        55        56        57        61
       62       63        67        68        69        73        74        75        79        80
       81

     LATEST POSSIBLE ORIGIN (KORIG2)
        1        2         7         8        13        19        20        25        31        37
       43       44        45        49        50        51        55        56        57        61
       62       63        67        68        69        73        74        75        79        80
       81
```

```
**************************************************

     CURRENT PARAMETER AND COUNTER VALUES
     (SEE 2002 AREA CF MAIN FOR LABELS)
     2CCO         2**********        1         0        29         0         0         0
        0        31        31        0         1         0         0         0
        0        24         0        0         0
        0         0         0        0         0
        1         0         1        1         1         1
        2         1        84        9         1         0         0         0
  -C.5CCO    0.0    **********

     CURRENT SEQUENCE VECTOR (THOSE J S.T. YJ=1) (KSEQU)
        0         9

     CURRENT ORIGIN (THOSE J S.T. YJ=1) (KORIG)
        1         2         7        8        13        19        20        25        31        37
       43        44        45       49        50        51        55        56        57        61
       62        63        67       68        69        73        74        75        79        80
       81

     LATEST POSSIBLE ORIGIN (KORIG2)
        1         2         7        8        13        19        20        25        31        37
       43        44        45       49        50        51        55        56        57        61
       62        63        67       68        69        73        74        75        79        80
       81

**************************************************

     CURRENT PARAMETER AND COUNTER VALUES
     (SEE 2C02 AREA CF MAIN FOR LABELS)
     2C00        31206960924         2         1        29         1         0         0
        0        31        31        0         1         0         0         0
        0        24         0        0         0
        0         0         0        0         0
        1         0         1        1         1         1
        3        13        60       54         1         0         0         0
   C.1C00    0.0      -0.0000

     CURRENT SEQUENCE VECTOR (THOSE J S.T. YJ=1) (KSEQU)
        0         9        54

     CURRENT ORIGIN (THOSE J S.T. YJ=1) (KORIG)
        1         2         7        8        13        19        20        25        31        37
       43        44        45       49        50        51        55        56        57        61
       62        63        67       68        69        73        74        75        79        80
       81

     LATEST POSSIBLE ORIGIN (KORIG2)
        1         2         7        8        13        19        20        25        31        37
       43        44        45       49        50        51        55        56        57        61
       62        63        67       68        69        73        74        75        79        80
       81

**************************************************

     CURRENT PARAMETER AND COUNTER VALUES
     (SEE 2002 AREA CF MAIN FOR LABELS)
     2C00        41206960924         2         1        29         1         0         0
        0        31        31        0         2         1         1         0
```

```
     0        24         0         0         0
     0         0         0         0         0
     1         0         1         1         1         1
     3        13        60        54         1         0         0         0
  0.1000     0.0      -0.0000

   CURRENT SEQUENCE VECTOR (THOSE J S.T. YJ=1) (KSEQU)
     0         9        54

   CURRENT ORIGIN (THOSE J S.T. YJ=1) (KORIG)
     1         2         7         8        13        19        20        25        31        37
    43        44        45        49        50        51        55        56        57        61
    62        63        67        68        69        73        74        75        79        80
    81

   LATEST POSSIBLE ORIGIN (KORIG2)
     1         2         7         8        13        19        20        25        31        37
    43        44        45        49        50        51        55        56        57        61
    62        63        67        68        69        73        74        75        79        80
    81

************************************************

   CURRENT PARAMETER AND COUNTER VALUES
   (SEE 2002 AREA OF MAIN FOR LABELS)
  2000         5         1         2         1        30         1         0         0
     0        31        31         0         2         1         0         0
     0        24         1         0         0
     0         0         0         0         0
     1         0         1         1         1         1
     2        13        60        54         0         0         0         0
 -0.5000     0.0      -0.6560

   CURRENT SEQUENCE VECTOR (THOSE J S.T. YJ=1) (KSEQU)
     0         9

   CURRENT ORIGIN (THOSE J S.T. YJ=1) (KORIG)
     1         2         7         8        13        19        20        25        31        37
    43        44        45        49        50        51        55        56        57        61
    62        63        67        68        69        73        74        75        79        80
    81

   LATEST POSSIBLE ORIGIN (KORIG2)
     1         2         7         8        13        19        20        25        31        37
    43        44        45        49        50        51        55        56        57        61
    62        63        67        68        69        73        74        75        79        80
    81

************************************************

   CURRENT PARAMETER AND COUNTER VALUES
   (SEE 2002 AREA OF MAIN FOR LABELS)
  2000         6        -1         2         1        30         1         0         0
     0        31        31         0         2         1         0         0
     0        24         1         0         0
     0         0         0         0         0
     1         0         1         1         1         1
     1         1        84         9         0         0         0         0
  0.0        0.0      -0.6560
```

```
     CURRENT SEQUENCE VECTOR (THOSE J S.T. YJ=1) (KSEQU)
        0

     CURRENT ORIGIN (THOSE J S.T. YJ=1) (KORIG)
        1          2          7          8         13         19         20         25         31         37
       43         44         45         49         50         51         55         56         57         61
       62         63         67         68         69         73         74         75         79         80
       81

     LATEST POSSIBLE ORIGIN (KORIG2)
        1          2          7          8         13         19         20         25         31         37
       43         44         45         49         50         51         55         56         57         61
       62         63         67         68         69         73         74         75         79         80
       81

**************************************************

     CURRENT PARAMETER AND COUNTER VALUES
     (SEE 2002 AREA CF MAIN FOR LABELS)
     2CC0          7         -1          2          1         30          1          0          0
        0         31         31          0          2          1          1          0
        0         24          1          0          0
        0          0          0          0          0
        1          0          1          1          1          1
        1          1         84          9          1          0          0          0
   C.0        0.0       -0.6560

     CURRENT SEQUENCE VECTOR (THOSE J S.T. YJ=1) (KSEQU)
        0

     CURRENT ORIGIN (THOSE J S.T. YJ=1) (KORIG)
        1          2          7          8         13         19         20         25         31         37
       43         44         45         49         50         51         55         56         57         61
       62         63         67         68         69         73         74         75         79         80
       81

     LATEST POSSIBLE ORIGIN (KORIG2)
        1          2          7          8         13         19         20         25         31         37
       43         44         45         49         50         51         55         56         57         61
       62         63         67         68         69         73         74         75         79         80
       81

OPTIMAL SOLUTION ASCERTAINED AFTER ITERATIVE STEP NO          7

SOLUTION WAS PRINTED AFTER ITERATIVE STEP NO.          0

**************************************************

FINAL OUTPUT...COUNTER RESULTS

NUMBER OF . . .

     PCINTS INVESTIGATED (KFOR)          2

     TIMES PREFERRED ROW SELECTED (NTSI)          1

     INFEASIBILITY BACKUPS (KCTBI)          0

     CANCELLATIONS (KCTIF)          31
```

```
     TIMES LP USED (NLP)        1

     TIMES LP EXCEEDED CEILING (NCEL)        1

     INFEASIBLE LP'S (NOLPJ)        0

     CANCELLATIONS DURING LP (KCLP)        0

     VAR. CANCELLED TO 1 DETERMINED BY LP (MFORCE)        0

     INFEASIBLE LP'S DUE TOO POST OPTIMALITY CONSTRAINTS (KABOOM)        0

     FORCED VARIABLES CANCELLED (JOY)        0

     LP'S TERMINATED BECAUSE OF NO. OF PIVOT STEPS (NITE)        0

     NONCURRENTLY FEASIBLE SOLUTIONS REACHED (MIST)        1

     TIMES CEILING TESTS EFFECTIVE (KLOP+JA+KEG)        26

**********************************************

ELAPSED TIME FOR JOB    1 IS     25.590     SEC

ALL JOBS COMPLETED
```

OPTIMIZATION OF SYSTEM RELIABILITY

OF LIFE SUPPORT SYSTEMS USING

AN INTEGER PROGRAMMING

by

SUDESH KUMAR

B.Sc. (Engg.) Mechanical

Ranchi University, India, 1966

M.E. (Production Engineering)

Roorkee Univ., India, 1968

AN ABSTRACT OF A MASTER'S REPORT

Submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

Kansas State University

Manhattan, Kansas

1970

ABSTRACT

The purpose of this report is twofold: first to summarize the introductory theory of reliability of a multistages system, secondly, to present the optimization of the reliability of life support systems obtained by applying zero-one integer programming.

Integer programming methods and their application to optimization of system reliability are described briefly; an implicit enumeration technique is given in detail. The problems of maximizing the reliability of a system with supporting units subject to several separable nonlinear constraints and minimizing the cost of a certain type for a complex system subject to several separable nonlinear constraints while maintaining an acceptable level of reliability of the system are considered. The zero - one integer programming is used to optimize the above mentioned problems.

An example of the subsystems of a life support system which has been designed by the Space Division of North American Rockwell Corporation is presented. Four of these subsystems are considered to investigate the feasibility of the integer programming approach.

The results obtained are compared with the design suggested by North American Rockwell Corporation. The comparison indicates that the use of integer programming for the optimization of reliability resulted in a better arrangement of spares. The reason for this is that the present system has less weight than that suggested and at the same time achieves the same level of reliability.

Specifically the four subsystems of the life support systems optimized are a refrigerant circuit, a cryogenic oxygen circuit, a carbrnization cell, and a water reclamation unit. The refrigerant

circuit is required to have a reliability of 0.99970, the weight of the spares is reduced from 20.3 lb. to 17.7 lb. The weight of the spares for cryogenic oxygen supply circuit is reduced from 22.5 lb. to 22.0, while maintaining a level of reliability of 0.99996. The carbonization cell is designed with a reliability of 0.9998 and the weight of the spares is reduced from 27.9 lb. to 25.3 lb. The weight of the spares for water reclamation loop is reduced from 24.5 lb. to 22.57 lb. while maintaining a reliability of 0.99992.