INTEGER PROGRAMMING AND NONLINEAR INTEGER GOAL PROGRAMMING APPLIED TO SYSTEM RELIABILITY PROBLEMS

by

Hoon Byung Lee

B. S., Seoul National University, Seoul, Korea 1973

A MASTER'S THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1978

Approved by:

Ching-Lai Hwang

Major Professor

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Reliability is a measure of the capacity of a piece of equipment to operate without failure when put into service. Reliability has been defined in various ways. One of the best is that of the National Aeronautics and Space Administration (NASA) which defines reliability as the probability of a device performing adequately for the period of time intended under the operating conditions encountered [1].

There exist several methods to improve the system reliability, eg, using large safety factors, reducing the complexity of the system, increasing the reliability of components through a product improvement program, using structural redundancy, and practicing a planned maintenance and repair schedule [2]. In this thesis the method of allocating redundancy, the development of a general mathematical solution for the optimum number, and the type of redundant components in a mixed parallel-series system [2] are studied.

There have been a number of studies in the application of integer programming to the optimization of single objective system reliability problems. Tillman [3], [4] used Gomory's cutting plane algorithm for maximizing reliability or minimizing cost subject to several constraints and for components having different modes of failures. Hyun [5] solved the Tillman's same problem by Geoffrion's implicit enumeration method. Ghare and Tayler [6] used Branch-and-Bound algorithm to determine the exact solution to the problem of maximizing the total system reliability when it is subject to multiple resource constraints. Misra [7]

used the Lawler-Bell partial enumeration method to solve problems such as maximizing reliability or optimizing some other objective function subject to multiple separable constraints, which need not be linear. Hwang, Fan, Tillman, and Kumar [8] used zero-one integer programming to minimize the weight of the subsystems of a life support system subject to several separable constraints while maintaining an acceptable level of reliability of the system.

Researches on multiple objective problems became active recently, however, little study has been done on integer constrained multiple objective optimization. In the early 1960's Charnes and Cooper [9] presented an approach to the solution of linear decision models having more than a single objective. Later the work of Ijiri [10], and Lee [11] has resulted in a systematic methodology known as Goal Programming for solving linear, multiple objective problems. Ignizio [12] presented linear integer goal programming using Cutting Plane method, Branch-and-Bound method, and Zero-One algorithm to solve integer constrained multiple objective problems. He also presented nonlinear goal programming and nonlinear integer goal programming. His nonlinear integer goal programming method is to reformulate the nonlinear problem into zero-one linear problem and to solve it by linear integer goal programming using zero-one algorithm. However, the size of a problem increases dramatically during the reformulation, hence his method is not appropriate for big size problems. Hwang and Paidy [13] used nonlinear goal programming method to solve "Regional Water Quality Management" problem. But little study has been done on nonlinear integer goal programming so far.

The objective of this study is to present the techniques to solve integer constrained problems encountered in the system reliability optimization. In Chapter 2 integer programming techniques are used to solve single objective reliability problems with linear or nonlinear constraints. In Chapter 3 a nonlinear goal programming method is formulated using Hwang and Paidy's nonlinear goal programming technique [13] and Branch-and-Bound algorithm [12]. Four multiple objective nonlinear reliability optimization problem with integer constraint are solved to demonstrate the algorithm. Concluding remarks and proposals for further study are discussed in the final chapter.

Statement of Problem

Generally a system with redundant components at each stages has a structure shown in Figure 1-1. The reliability of this type of system can be increased by adding more redundant components at each stage. But by doing so, the weight, the cost, and the size of the system will also increase. There needs some trade-off among them. Hence, the problem of this thesis is how to allocate redundant components in order to optimize the system reliability.

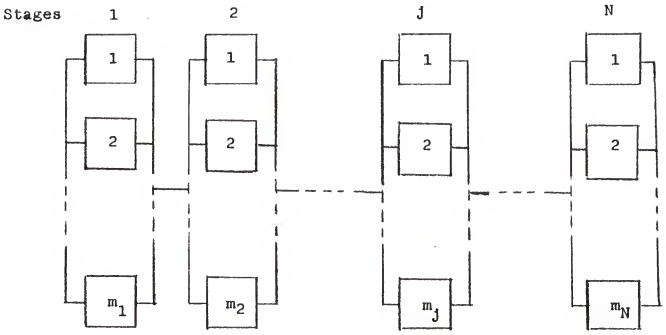

Figure 1-1. A system with N-stages in series where redundant components are in parallel at each stage.

REFERENCES

1. Smith, Charles O., Introduction to Reliability in Design, N.Y.: McGraw-Hill Book Company (1976).

2. Tillman, Frank A., Ching-Lai Hwang, and Way Kuo, "Optimization Techniques for system reliability with redundancy - A review," IEEE Transactions on Reliability, Vol. R-26, No. 3, pp. 148-155 (August 1977).

3. Tillman, F. A. and Liittschwager, "Integer programming formulation of constrained reliability problems," Management Science, Vol. 13, July 1967, pp. 887-899.

4. Tillman, F. A., "Optimization by integer programming of constrained reliability problems with several modes of failure," IEEE Transactions on Reliability, Vol. R-18, No. 2, pp. 47-53 (May 1969).

5. Hyun, K. N., "Reliability optimization by 0-1 programming for a system with several failure modes", IEEE Transactions on Reliability, Vol. R-24, No. 3, pp. 206-210 (August 1975).

6. Ghare, P. M. and R. E. Taylor, "Optimal redundancy for reliability in series system", Operations Research, Vol. 17, pp. 838-847 (Sept. 1969).

7. Misra, K. B., "A method of solving redundancy optimization problems", IEEE Transactions on Reliability, Vol. R-20, No. 3, pp. 117-120 (August 1971).

8. Hwang, C. L., L. T. Fan, F. A. Tillman, and S. Kumar, "Optimization of life support system reliability by an integer programming method", AIIE Transactions, Vol. 3, No. 3, pp. 229-238 (September 1971).

9. Charnes, A. and W. W. Cooper, Management Models and Industrial Applications of Linear Programming, Vols. I and II, New York: John Wiley & Sons, 1961.

10. Ijiri, Y., Management Goals and Accounting for Control, Chicago: Rand-McNally & Co., 1965.

11. Lee, Sang M., Goal Programming for Decision Analysis, Philadelphia: Auerbach Publishers, 1972.

12. Ignizio, James P., Goal Programming and Extensions, Lexington, Mass.: Lexington Books, 1976.

13. Hwang, C. L. and S. Paidy, "Regional water quality management by nonlinear goal programming ", submitted to Water Resources Bulletin for publication, 1978.

Chapter 2

INTEGER PROGRAMMING APPLIED TO OPTIMAL SYSTEM RELIABILITY

1. Introduction

Various papers have presented the application of integer programming to a variety of problems. Problems treated in these papers can be classified into the following examples [1]:

Nomenclature

c_j = cost per component at stage j.

h_i = number of class 0 failure modes in subsystem i.

m_j = number of redundant components at stage j.

$m_{j,max}$ = maximum number of redundant components at stage j.

N = number of stages in the system.

q_{iu} = probability of failure mode u for each element in subsystem i.

R_j = component reliability at stage j.

R_s = system reliability.

$R_{s,min}$ = minimum required system reliability.

s_i-h_i = number of class A failure modes in subsystem i.

W = weight of the system.

w_j = weight of a component at stage j.

Example 2-1 Linear Objective Function

The problem is to minimize a linear cost function

$$f = \sum_{j=1}^{N} c_j m_j$$

for an N-stage series system, where m_j+1 components are used in the jth stage, subject to the constraints:

$$R_s \geq R_{s,min}$$

$$\sum_{j=1}^{N} w_j(m_j+1) \leq W$$

where

$$R_s = \prod_{j=1}^{N} [1 - (1 - R_j)^{m_j+1}]$$

The constants associated with the problem are given as

$N = 2$, $R_{s,min} = .9903$, $W = 40$

$R_1 = .91$ $R_2 = .96$,

$c_1 = 5$, $c_2 = 8$

$w_1 = 9$, $w_2 = 6$

Example 2-2 Nonlinear Objective Function and Linear Constraint Functions

Consider the problem in which N stages are connected in series and redundant components, m_j, are added in parallel at each stage. The objective is to determine m_j at each stage, such that the system reliability is maximized and the weignt and cost constraints are not exceeded. The problem is stated as

Maximize

$$R_s = \prod_{j=1}^{N} [1 - (1 - R_j)^{m_j+1}]$$

subject to

$$g_1 = \sum_{j=1}^{N} c_j m_j \leq C$$

$$g_2 = \sum_{j=1}^{N} w_j m_j \geq W$$

Consider the set of data used in [2,3,4,5].

Stage	Cost	Weight	Probability of Survival
j	c_j	w_j	R_j
1	1.2	1.0	0.20
2	2.3	1.0	0.30
3	3.4	1.0	0.25
4	4.5	1.0	0.15

$C = 47.0, \quad W = 20.0$

Example 2-3 Nonlinear Objective Function and Nonlinear Constraint Functions

In this example [6,7,8], the system has N stages operating in series. We want to achieve a system reliability being at least $R_{s,min}$ while minimizing the cost. To attain this reliability, redundant components, m_j, are added in parallel up to a maximum of allowed number, $m_{j,max}$, at each stage. The problem is:

Minimize

$$Z = \sum_{j=1}^{N} c_j m_j \exp(-m_j/2)$$

subject to

$$g_1 = \sum_{j=1}^{N} [a_{j1} m_j + a_{j2} m_j^2 + a_{j3}] \leq A$$

$$g_2 = \sum_{j=1}^{N} b_j [m_j + \exp(-m_j) - \alpha_j] \geq B$$

$$g_3 = \sum_{j=1}^{N} d_j m_j \exp(-m_j/4) \geq D$$

$$R_s = \prod_{j=1}^{N} [1 - (1-R_j)^{m_j+1}] \geq R_{s,min}$$

$$0 \leq m_j \leq m_{j,max}, \qquad j = 1,2, \ldots, N$$

The constants assigned to this problem are:

j	c_j	a_{j1}	a_{j2}	a_{j3}	b_j	α_j	d_j	R_j
1	3	3	1	0	30	0	30	.90
2	2	3	1	1	30	4	30	.75

$N = 2$, $R_{s,min} = .85$, $m_{j,max} = 4$, $A = 37$, $B = 81$, $D = 38$

Example 2-4

The example is to maximize nonlinear system reliability subject to 3 nonlinear constraints with redundant components in each stage that are subject to type 1 failures [9].

Maximize

$$R(m) = \prod_{i=1}^{3} \left\{ 1 - \sum_{u=1}^{h_i} [1 - (1 - q_{iu})^{m_i+1}] - \sum_{u=h_i+1}^{s_j} (q_{iu})^{m_i+1} \right\}$$

subject to

$$G_1(m) = (m_1+3)^2 + (m_2)^2 + (m_3+2)^2 \leq 51,$$

$$G_2(m) = 20(m_1 \; p \exp(-m_1)) + 20(m_2 + \exp(-m_2)) + 20(m_3 + \exp(-m_3)) \geq 120,$$

$$G_3(m) = 20(m_1 \exp(-m_1/4)) + 20(m_2 \exp(-m_2/4))$$

$$+ 20(m_3 \exp(-m_3/4)) \geq 65,$$

$m = (m_1, m_2, m_3)$, m_i positive integer for $i = 1, 2, 3$.

The subsystems are subject to four failure modes ($s_i = 4$) with one 0 failure ($h_i = 1$) and three A failures, for $i = 1, 2, 3$. For each subsystem the failure probability of an element is shown in Table 2-1.

Classification of Examples and Approaches

Generally, to solve the previous examples, there are five distinct approaches capable of solving these examples by integer programming. These are: Partial enumeration - Lawler & Bell, Implicit enumeration - Lemke & Spielberg, Cutting plane method-Gomory, Branch-and-Bound, and Implicit enumeration-Geoffrion. They are classified in Table 2-2.

2. The Lawler-Bell Partial Enumeration Method

Lawler and Bell [15] describe a programmed method for solving discrete optimization problems with monotone objective functions and arbitrary (possible nonconvex) constraints.

A brief review of the Lawler-Bell method is provided in this section. The type of problems that can be solved by this method may be put in the following form. Minimize $g_0(x)$ subject to m constraints of the form

Table 2-1

The type of failure and its failure probability for each element.

subsystem i	type of failure u	failure probability q_{iu}
	0	.01
1	A	.05
	A	.10
	A	.18
	0	.08
2	A	.02
	A	.15
	A	.12
	0	.04
3	A	.05
	A	.20
	A	.10

Table 2-2 Classification of examples and approaches

Examples	Methods applied to the examples	References
Example 2-1	Partial enumeration-Lawler & Bell	6
	Implicit enumeration-Lemke & Spielberg	10
Example 2-2	Cutting plane method - Gomory	7,8
	Branch-and-Bound	2,3,4,11
	Partial enumeration-Lawler & Bell	6
	Partial enumeration	12
	Enumeration-Balas or Glover	13
Example 2-3	Cutting plane method - Gomory	7,8
	Partial enumeration-Lawler & Bell	6
Example 2-4	Cutting plane method - Gomory	9
	Implicit enumeration - Geoffrion	14

$$\left.\begin{aligned} & g_{i1}(x) - g_{i2}(x) \geq 0,\ i = 1, \ldots, m \\ \text{where} \\ & x = (x_1, x_2, \ldots, x_n), \\ \text{and} \\ & x_j = 0 \text{ or } 1,\ j = 1, \ldots, n \end{aligned}\right\} \qquad (1)$$

Each of the functions in (1) must be monotone nondecreasing in each of its arguments. With some ingenuity, many problems can be put in this form.

Vector x is "binary" in the sense that each x_j is either 0 or 1; $x \leq y$ if and only if $x_j \leq y_j$ for j = 1, ..., n. eg., $x \leq y$, where x = (0,1,0) and y = (0,1,1). This is the vector partial ordering. There is also the lexicographic or numerical ordering of these vectors obtained by identifying with each x. Define the integer value $N(x) = x_1 2^{n-1} + x_2 2^{n-2} + \ldots + x_n 2^0$. Numerical ordering is a refinement of the vector partial ordering, i.e., $x \leq y$ implies $N(x) \leq N(y)$; however, $N(x) \leq N(y)$ does not imply $x \leq y$.

Suppose all binary n-vectors are listed in numerical order, i.e.,

(0,...,0,0,0),

(0,...,0,0,1),

(0,...,0,1,0),

(0,...,0,1,1),

(0,...,1,0,0),

etc.

Immediately following an arbitrary vector x, there may (or may not) be a number of vector x' with the property that $x \leq x'$. Roughly speaking, these are vectors that differ from x only in that they have 1's in place of one or more of the 'right-most' 0's of x. For example, immediately following x = (0,1,0,0) are (0,1,0,1), (0,1,1,0) and (0,1,1,1), each of which is greater than x in the vector partial ordering.

Let x* denote the first vector following x in the numerical ordering that has the property that $x \not\leq x^*$. For any given x, the vector x* is very easily calculated on a computer as follows:

Treat x as a binary number:

(1) Subtract 1 from x,

(2) Logically 'or' x and x-1 to obtain x*-1,

(3) Add 1 to obtain x*.

Some examples:

Let x = 0101100,

(1) x-1 = 0101011,

(2) x*-1 = 0101111,

(3) x* = 0110000,

Let x = 0101011,

(1) x-1 = 0101010,

(2) x*-1 = 0101011,

(3) x* = 0101100,

Let x = 0101000,

(1) x-1 = 0100111,

(2) x*-1= 0101111,

(3) x* = 0110000.

Note that x*-1 is greater than each of x, x+1,...,x*-2, in the vector partial ordering.

The method is basically a search method, which starts with $x = \{0,0,\ldots,0\}$ and examines the 2^n solution vectors in the numerical ordering described above. Further, the labor of examination is considerably cut down by the following rules. As the examination proceeds one can retain the least costly up-to-date solution. If $\hat{x}$ is the solution having "cost" $g_0(\hat{x})$ and x is the vector being examined, then the following steps indicates the conditions under which certain vectors may be skipped.

1) Test if $g_0(x) \geq g_0(\hat{x})$. If YES, skip to x* and repeat the operation; otherwise proceed to step 2).

2) Examine whether $g_{i1}(x^*-1) - g_{i2}(x) \geq 0$ for $i = 1,\ldots,m$. If YES, proceed to step 3); otherwise skip to x* and go to step 1).

3) Further, if $g_{i1}(x) - g_{i2}(x) \geq 0$, $(i = 1, \ldots, m)$, replace x by $\hat{x}$ and skip to x*; otherwise change x to x+1. In either case further execution is transferred to step 1). Lawler and Bell [15] call the above steps of the algorithm skipping rules 1,3,2, respectively. Following the above rules, all the vectors are examined and scanning continues until a vector having maximum numerical order, viz., $\{1,1,\ldots,1\}$, is found. In case one has skipped to a vector having numerical order higher than $\{1,\ldots,1\}$, designate this state by "overflow" and terminate

the procedure. The least "costly" vector recorded provides the optimum solution. One should not be over-whelmed by the number of trials. In practice the number of vectors to be examined may be quite small. For example, in all 11-variable problem with a total of 2^{11}-solution vectors, only 42 vectors were examined.

Example 2-1

This example should first of all be formulated as follows:

Minimize

$$f = 5m_1 + 8m_2$$

subject to

$$[1 - (1-.91)^{1+m_1}][1 - (1-.96)^{1+m_2}] \geq .9903$$

$$9(1+m_1) + 6(1+m_2) \leq 40 \qquad (2)$$

or

minimize

$$g_o(m) = 5m_1 + 8m_2$$

subject to

$$g_1(m) = \ln(1-.09^{1+m_1}) + \ln(1-.04^{1+m_2}) + .009747 \geq 0$$

$$g_2(m) = 25 - 9m_1 - 6m_2 \geq 0 \qquad (3)$$

Before m_1 and m_2, the nonnegative integer variables, can be transformed to the variables of zero-one type, it is necessary to estimate their maximum values. This is done by substituting zero for all variables in the constraints in (3) except the one for which the maximum range is desired. Denote these by m^*_{ij}, where subscripts i and j refer to the

constraint and stage, respectively. Then $\hat{m}_j = \min \{m^*_{ij}\}$ $(i = 1,\ldots,m)$ is an upper bound for m_j.

In problem (3) the upper limits of m_1 and m_2 can be found by letting $m_1 = 0$ or $m_2 = 0$ alternatively as follows.

stage 1:

$$\ln(1 - .09^{1+m_1}) + \ln(1 - .04) + .009747 \geq 0$$

or

$$\ln(1 - .09^{1+m_1}) \geq .031075$$

Hence

$$m_{11}^* = M_1 \quad (\text{where } M_1 \to \infty)$$

$$25 - 9m_1 - 6(0) \geq 0$$

or

$$m_1 \leq 2.78$$

Hence

$$m_{21}^* = 2$$

$$\hat{m}_1 = \min \{M_1, 2\}$$

$$= 2$$

stage 2;

$$\ln (1 - .09) + \ln (1 - .04^{1+m_2}) + .009747 \geq 0$$

or

$$\ln (1 - .04^{1+m_2}) \geq .08564$$

Hence

$$m_{12}^* = M_2 \quad (\text{where } M_2 \to \infty)$$

$$25 - 9(0) - 6m_2 \geq 0$$

or

$$m_2 \leq 4.17$$

Hence

$$m_{22}^{*} = 4$$

$$\hat{m}_2 = \min \{M_2, 4\}$$

$$= 4$$

Let

$$m_1 = x_{11} + x_{12}$$

$$m_2 = x_{21} + x_{22} + 2x_{23} \tag{4}$$

where x_{ij} is either 0 or 1.

Substituting these in problem (3), one can obtain the following problem as defined in problem (1):

$$g_o(x) = 5x_{11} + 5x_{12} + 8x_{21} + 8x_{22} + 16x_{23}$$

$$g_{11}(x) = \ln(1 - .09^{x_{11}+x_{12}+1}) + \ln(1 - .04^{x_{21}+x_{22}+2x_{23}+1}) + .009747$$

$$g_{12}(x) = g_{21}(x) = 0$$

$$g_{22}(x) = 9x_{11} + 9x_{12} + 6x_{21} + 6x_{22} + 12x_{23} - 25 \tag{5}$$

Now the problem conforms to the Lawler-Bell algorithm. The solution is arrived at after examining only twelve vectors out of the 32 generated by

the five binary variables of (4). The sequence of examination and the different rules applied are indicated in Table 2-3. The vector ordering used is also shown, viz., $x \equiv \{x_{23}, x_{12}, x_{22}, x_{11}, x_{21}\}$. There are no definite rules about the ordering of these variables. However, it has been observed for all the problems studied that the variables carrying least "numerical weights" are assigned the "rightmost" position in the ordering. This is done so that the numerical values of m_1 and m_2 increase as the examination of solution vectors x preceeds.

To begin with Table 2-3, we set $x = (0,0,\ldots,0)$ and $g_0(\hat{x}) = \infty$ and at the end of the table, the solution is $\hat{x}$ and the minimum cost is $g_0(\hat{x})$.

Iteration 1)

$x = (0,0,0,0,0)$

$x^*\text{-}1 = (0,0,0,0,0)$

$g_{11}(x^*\text{-}1) = g_{12}(x) = \ln(1 - .09) + \ln(1 - .04) + .009747$

$= .125386 < 0$

Skip to x* through step 2)

Iteration 2)

$x = (0,0,0,0,1)$

$g_0(x) = 8$

Hence

$g_0(x) < g_0(\hat{x})$

$x^*\text{-}1 = (0,0,0,0,1)$

Table 2-3

The examination sequence of Example 2-1 by Lawler-Bell algorithm.

	x_{23}	x_{12}	x_{22}	x_{11}	x_{21}		
	0	0	0	0	0	$g_{11}(x^*\text{-}1) - g_{12}(x) < 0$	skip to x* through step 2)
	0	0	0	0	1	$g_{11}(x^*\text{-}1) - g_{12}(x) < 0$	skip to x* through step 2)
	0	0	0	1	0	$g_{11}(x) - g_{12}(x) < 0$	change x → x+1 through step 3)
→	0	0	0	1	1	feasible, $g_o(x) = 13$	skip to x* through step 3)
	0	0	1	0	0	$g_{11}(x) - g_{12}(x) < 0$	change x → x+1 through step 3)
	0	0	1	0	1	$g_o(x) > g_o(\hat{x})$	skip to x* through step 1)
	0	0	1	1	0	$g_o(x) = g_o(\hat{x})$	skip to x* through step 1)
	0	1	0	0	0	$g_{11}(x) - g_{12}(x) < 0$	change x → x+1 through step 3)
	0	1	0	0	1	$g_o(x) = g_o(\hat{x})$	skip to x* through step 1)
	0	1	0	1	0	$g_{11}(x) - g_{12}(x) < 0$	change x → x+1 through step 3)
	0	1	0	1	1	$g_o(x) > g_o(\hat{x})$	skip to x* through step 1)
	0	1	1	0	0	$g_o(x) > g_o(\hat{x})$	skip to x* through step 1)
	1	0	0	0	0	$g_o(x) > g_o(\hat{x})$	skip to x* through step 1)

x* = 1 (0,0,0,0,0); therefore overflow takes place and we stop.

$$g_{11}(x^*\text{-}1) - g_{12}(x) = \ln(1 - .09) + \ln(1\text{-}.04^2) + .009747$$

$$= \text{-}.086165 < 0$$

skip to x* through step 2).

Iteration 3)

$$x = (0,0,0,1,0)$$

$$g_0(x) = 5$$

Hence

$$g_0(x) < g_0(\hat{x})$$

$$x^*\text{-}1 = (0,0,0,1,1)$$

$$g_{11}(x^*\text{-}1) - g_{12}(x) = \ln(1 - .09^2) + \ln(1 - .04^2) + .009747$$

$$= .000013 > 0$$

$$g_{21}(x^*\text{-}1) - g_{22}(x) = \text{-}(9+6-25)$$

$$= 10 > 0$$

change x to x+1 through step 3).

Iteration 4)

$$x = (0,0,0,1,1)$$

$$g_0(x) = 5+8 = 13$$

Hence

$$g_0(x) < g_0(\hat{x})$$

$$x^*\text{-}1 = (0,0,0,1,1)$$

$$g_{11}(x^*\text{-}1) - g_{12}(x) = \ln(1\text{-}.09^2) + \ln(1\text{-}.04^2) + .009747$$

$$= .000013 > 0$$

$$g_{21}(x^*\text{-}1) - g_{22}(x) = -(9+6-25)$$

$$= 10 > 0$$

$$g_{11}(x) - g_{12}(x) = .000013 > 0$$

$$g_{21}(x) - g_{22}(x) = 10 > 0$$

Therefore, $g_0(x) = 13$ is a feasible solution.

$$\hat{x} = (0,0,0,1,1)$$

skip to x* through step 3).

Iteration 5)

$$x = (0,0,1,0,0)$$

$$g_0(x) = 8$$

Hence

$$g_0(x) < g_0(\hat{x})$$

$$x^*\text{-}1 = (0,0,1,1,1)$$

$$g_{11}(x^*\text{-}1) - g_{12}(x) = \ln(1\text{-}.09^2) + \ln(1\text{-}.04^3) + .009747$$

$$= .001550 > 0$$

$$g_{21}(x^*\text{-}1) - g_{22}(x) = -(6-25)$$

$$= 19 > 0$$

$$g_{11}(x) - g_{12}(x) = \ln(1-.09) + \ln(1-.04^2) + .009747$$

$$= -.086165 < 0$$

change x to x+1 through step 3).

The process is repeated, and the true optimum is shown by the arrow in Table 1. Therefore, $m_1 = m_2 = 1$ from (4).

Actually in a large problem there is an appreciable reduction in the number of solution vectors being inspected. For example in a 5-stage problem of Bellman [16] requiring 11 binary variables, the solution was obtained by examining 42 of the 2^{11} solutions.

3. The Branch-and-Bound Method

The branch and bound method [2,17], is briefly introduced as follows:

Problem A: Maximize the total system reliability

$$R_s = \prod_{i=1}^{m} (1-p_i^{n_i})$$

subject to the constraints:

$$\sum_{i=1}^{m} a_{ij} n_i \leq d_j, \qquad j = 1, \ldots, s; \quad n_i \geq 1; \; n_i \text{ integer.}$$

If we make the following transformations:

$$c_{ik} = \ln (1 - p_i^{k+1}) - \ln (1 - p_i^k)$$

$$b_j = d_j - \sum_{i=1}^{m} a_{ij}$$

then Problem A can be identically formulated as:

Problem B: Maximize

$$Z = \sum_{i=1}^{m} \sum_{k=1}^{\infty} c_{ik} x_{ik},$$

subject to constraints:

$$\sum_{i=1}^{m} \sum_{k=1}^{\infty} a_{ij} x_{ik} \leq b_j, \qquad j = 1, \ldots, s;$$

where $x_{ik} = 0$ or 1; and $x_{ik} = 0$ implies $x_{i\ell} = 0$ if $\ell > k$, $x_{ik} = 1$ implies $x_{im} = 1$ if $m < k$.

The one-to-one correspondence of Problem A and Problem B can be easily proved:

Let $\mathbf{X} = \{x_{ik}\}$ be a feasible solution to Problem B and let k_i be the largest index such that $x_{ik} = 1$.

Since X is a feasible solution for Problem B,

$$\sum_{i=1}^{m} \sum_{k=1}^{\infty} a_{ij} x_{ik} \leq b_j$$

$$\sum_{i=1}^{m} a_{ij} k_i \leq d_j - \sum_{i=1}^{m} a_{ij}$$

$$\sum_{i=1}^{m} a_{ij} (x_i + 1) \leq d_j$$

Hence $N = (n_i \mid n_i = k_i + 1)$ is a feasible solution for problem A. The other constraints are satisfied since k_i is a nonnegative integer.

The objective function for the feasible solution X in problem B is given by

$$Z = \sum_{i=1}^{m} \sum_{k=1}^{\infty} c_{ik} x_{ik}$$

$$= \sum_{i=1}^{m} \sum_{k=1}^{k_i} \{\ln(1 - P_i^{k+1}) - \ln(1 - P_i^{k})\}$$

$$= \sum_{i=1}^{m} \{\ln(1 - P_i^{k_i+1}) - \ln(1 - P_i)\}$$

$$= \ln R - \sum_{i=1}^{m} \ln(1 - P_i).$$

As a conclusion, R_s is maximized when Z is maximum, namely, the optimal solution to problem B corresponds to the optimal solution to problem A.

The proposed solution procedure first obtains an optimal solution to problem B and then converts it into an optimal solution for problem A by using the relation $n_i = k_i + 1$. Problem B is a multi-dimensional knapsack problem (MDK).

The branch-and-bound procedure consists of two phases: (1) partitioning the space of all feasible solutions into mutually exclusive and exhaustive subsets using a consistent decision rule, and (2) establishing an upper (or lower as the case may be) bound over the objective functions within these subsets. These phase of branching and bounding are repeated until a solution is found that is better than the bounds on

all the unexplored subsets. This solution is optimal. A complete description of the branch-and-bound process is given by Balas [17].

In order to develop a bounding procedure for an MDK, consider a single-dimensional knapsack problem:

Maximize

$$\sum_{i=1}^{m} \sum_{k=1}^{\infty} c_{ik} x_{ik},$$

subject to a single constraint

$$\sum_{i=1}^{m} \sum_{k=1}^{\infty} a_{ij} x_{ik} \leq b_j \quad \text{for a given } j.$$

Define the ratios $\gamma_{ik} = c_{ik}/a_{ij}$. Then, for a feasible solution,

$$Z = \sum_{i=1}^{m} \sum_{k=1}^{\infty} c_{ik} x_{ik} = \sum_{i=1}^{m} \sum_{k=1}^{\infty} \gamma_{ik} a_{ij} x_{ik}$$

$$\leq \max_{i,k} [\gamma_{ik}] \sum_{i=1}^{m} \sum_{k=1}^{\infty} a_{ij} x_{ik} \leq \max_{i,k} [\gamma_{ik}] \cdot b_j .$$

Also, since

$$\exp(c_{ik}) = (1 - p_i^{k+1})/(1 - p_i^k) = 1 + p_i^k/(1 + p_i + \ldots + p_i^{k-1})$$

and

$$\exp(c_{i,k+1}) = (1 - p_i^{k+2})/(1 - p_i^{k+1}) = 1 + p_i^{k+1}/$$

$$(1 + p_i + \ldots + p_i^{k-1} + p_i^k),$$

it can be seen that $c_{ik} > c_{i,k+1}$, which implies $\gamma_{ik} > \gamma_{i,k+1}$, or

$\max_{i,k} \{\gamma_{ik}\} = \max_{i} \{\gamma_{i1}\}$. Hence $Z \leq \max_{i} \{\gamma_{i1}\} \cdot b_j$.

In the MDK there are s constraints, one for each resource j. Therefore, for any feasible solution for the MDK,

$$Z \leq \max_{i} \{\gamma_{i1}\} \cdot b_j \text{ for any } j$$

or

$$Z \leq \min_{j} [\max_{i} \{\gamma_{i1}\} \cdot b_j].$$

Consequently, the optimal feasible solution Z* is bounded by the quantity $\min_{j}[\max_{i}\{\gamma_{i1}\} \cdot b_j]$. This quantity is the upper bound for the MDK.

Let $X = \{x_{ik}\}$ be an intermediate solution in which none of the resources is fully utilized. This intermediate solution can be augmented by including $x_{i\ell}$ if i and ℓ satisfy the conditions (1) $x_{i,\ell-1} = 1$, (2) $x_{i\ell} = 0$, and (3) no exclude decision has previously been made for $x_{i\ell}$. Any such qualified variable can form the basis of a decision either to include or to exclude. This decision would partition the set of all the feasible solutions based on the intermediate solution X into two mutually exclusive and exhaustive subsets, and it would be a basis for branching. The subset described by the decision to include $x_{i\ell}$ (i.e., $x_{i\ell} = 1$) would be termed an inclusive branch, and the subset described by the decision exclude (i.e., $x_{i\ell} = 0$) would be termed an exclusive branch.

Let k_i be defined, as before, as the largest index such that $X_{ik} = 1$ before the branching decision. It can be seen that $\ell = k_i + 1$. Also, let I be the set of all indices i for which an exclude decision is made before the branching decision. Then the bound for the inclusive and exclusive branches (subsets) can be computed as follows.

Inclusive branch:

$$k_{i*} = k_i + 1.$$

Unallocated resource $b_j' = b_j - \sum_{i=1}^{m} \sum_{k=1}^{k_i} a_{ij}x_{ik} = b_j - \sum_{i=1}^{m} k_i \cdot a_{ij}$.

Objective function (after branching) = $\sum_{i=1}^{m} \sum_{k=1}^{k_i} c_{ik}$. Hence the upper bound on the inclusive branch equals

$$\sum_{i=1}^{m} \sum_{k=1}^{k_i} c_{ik} + \min_{j}\left(\max_{i \notin I} (\gamma_{i,k_i+1}) \cdot b_j'\right). \tag{1}$$

Exclusive branch:

$$i' = I \cup i*.$$

Unallocated resource $b_j' = b_j - \sum_{i=1}^{m} k_i \cdot a_{ij}$.

Objective function (after branching) = $\sum_{i=1}^{m} \sum_{k=1}^{k_i} c_{ik}$. (2)

Hence the upper bound on the exclusive branch equals

$$\sum_{i=1}^{m} \sum_{k=1}^{k_i} c_{ik} + \min_{j}\left(\max_{i \notin I'} (\gamma_{i,k_i+1}) \cdot b_j'\right).$$

Figure 2-1 shows the overall computational flow chart using inclusive or exclusive decisions for branching, and equations (1) and (2) for computing the bounds on the objective functions. The first forward solution is obtained by selecting the component for a branching decision that yields the highest upperbound on the inclusive branch. During the forward procedure, the bounds for the exclusive branch are stored as temporary bounds. The bounds for the inclusive branches are not stored explicitly. After a complete solution is reached (i.e., at least one of the resources is depleted completely at a given solution X^0), all the temporary bounds on the exclusive branches are revised. For this revision the index k_i is changed to the largest such that $x_{ik}=1$ in the solution X^0, for $i \neq i^*$. These revised upper bounds are then compared with the objective function $Z(X^0)$. Only those branches need to be explored further for which the upper bound exceeds $Z(X^0)$. The method of branch-and-bound is used to solve Example 2-2.

Example 2-2

This example can be converted into the following problem:

Maximize

$$Z = \sum_{i=1}^{4} \sum_{k=1}^{\infty} c_{ik}x_{ik}$$

subject to

$$\sum_{k=1}^{\infty} (1.2x_{1k} + 2.3x_{2k} + 3.4x_{3k} + 4.5x_{4k}) \leq 47 - (1.2 + 2.3 + 3.4 + 4.5) = 35.6$$

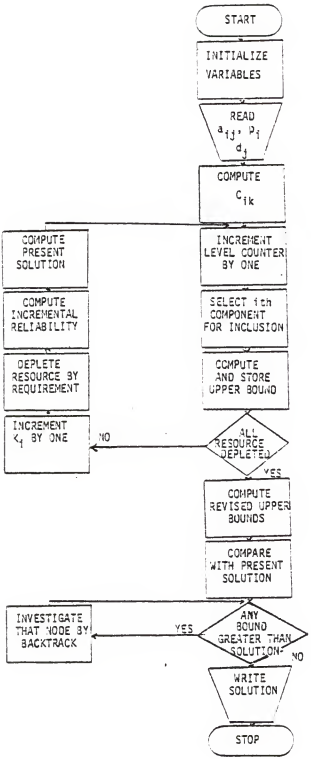

Figure 2-1. Macro flowchart of computational procedure.

$$\sum_{k=1}^{\infty} (x_{1k} + x_{2k} + x_{3k} + x_{4k}) \leq 20 - (1 + 1 + 1 + 1) = 16$$

where $c_{ik} = \ln (1 - p_i^{k+1}) - \ln (1 - p_i^k)$.

Hence $b_j = \{35.6, 16\}$

and $\{c_{ik}\} =$

i=1	i=2	i=3	i=4	
.182322	.262364	.223144	.139762	k=1
.032790	.066939	.048790	.061158	k=2
.006431	.019238	.011834	.002874	k=3
.001281	.005700	.002937	.000430	k=4
.000256	.001704	.000733	.000065	k=5
⋮	⋮	⋮	⋮	⋮

The computational procedure of Example 2-2 solved by branch-and-bound method is shown in Table 2-4. The optimal solution is reached at $(k_1,k_2,k_3,k_4) = (4,5,3,2)$ or $(n_1,n_2,n_3,n_4) = (5,6,4,3)$ which gives the system reliability, $R_s = 0.991691$. The same result has been obtained by Proschan and Bray [5].

4. The Gomory Cutting Plane Method

The reliability optimization problems solved by this method should have the objective and constraint functions in the separable types and need not satisfy any convexity and concavity conditions. A separable function of several variables is one that can be written as a sum of functions each with only one of the variables as argument.

Table 2-4

Illustration of the forward procedure

Level	k_i	Stage selected	$\min[\max_j \lfloor r_{ij} s_{ik_i+1} \rfloor, b'_j]$	$Z = \sum_{i=1}^{m} \sum_{k=1}^{k_i} c_{ik} x_{ik}$	Upper bound	b'_j revised	Comment
1	0,0,0,0	2	4.197824	0	4.197824	33.3, 15	Inclusive branch
2	0,1,0,0	3	3.347160	.262364	3.609524	29.9, 14	Inclusive branch
3	0,1,1,0	1	2.552508	.485508	3.038016	28.7, 13	Inclusive branch
4	1,1,1,0	4	0.891365	.667830	1.559195	24.2, 12	Inclusive branch
5	1,1,1,1	2	0.704317	.807592	1.511909	21.9, 11	Inclusive branch
6	1,2,1,1	1	0.590418	.874531	1.472949	20.7, 10	Inclusive branch
7	2,2,1,1	3	0.297045	.907321	1.204366	17.3, 9	Inclusive branch
8	2,2,2,1	4	0.235124	.956111	1.191235	12.8, 8	Inclusive branch
9	2,2,2,2	2	0.107059	1.017269	1.124328	10.5, 7	Inclusive branch
10	2,3,2,2	1	0.056270	1.036507	1.092777	9.3, 6	Inclusive branch
11	3,3,2,2	3	0.032373	1.042938	1.075311	5.9, 5	Inclusive branch
12	3,3,3,2	2	0.014620	1.054772	1.069392	3.6, 4	Inclusive branch
13	3,4,3,2	1	0.003845	1.060472	1.064317	2.4, 3	Inclusive branch
14	4,4,3,2	3	0.002074	1.061753	1.063827	-1, 2	Exclusive branch
15	4,4,3,2	2	0.001778	1.061753	1.063531	.1, 2	Inclusive branch
16	4,5,3,2	none	-	1.063457	-	.1, 2	Stop here

Initial solution $Z = 1.063457$

TABLE 2-5

The optimal configuration of Example 2-2

Stage i	No. of parallel components ni
1	5
2	6
3	4
4	3

Total system reliability = 0.991691

The reliability optimization problem can be formally stated as:

optimize

$$Z = \sum_{j=i}^{N} f_j(m_j)$$

subject to

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

$$\prod_{j=1}^{N} R_j' \geq R_{s,min}$$

and

$$m_j = 0, 1, \ldots, m_j' \qquad j = 1, 2, \ldots, N, \qquad (1)$$

where all the terms are known except Z and m_j and where,

Z = the objective function of the system to be optimized

N = the number of subsystems or stages

$f_i(m_j)$ = the objective function at stage j as a function of m_j, the number of redundant units

$g_{ij}(m_j)$ = the amount of the ith resource consumed at stage j as a function of m_j

b_i = the amount of the ith resource available

s = the number of constraints

$R_j' = 1 - (1-R_j)^{m_j+1}$, the reliability of the jth subsystem with m_j+1 units, where R_j is the reliability of each component

$R_{s,min}$ = the minimum acceptable reliability of the system

m_j = the number of redundant units used at stage j

m_j' = maximum number of redundant units allowed at stage j.

There exists the fact that after some transformations the above problem given by eq. (1) can be solved as the following integer programming problem expressed by eq. (2):

optimize

$$Z = \sum_{j=1}^{N} \sum_{k=0}^{m'_j} \Delta f_{jk} m_{jk}$$

subject to

$$\sum_{j=1}^{N} \sum_{k=0}^{m'_j} \Delta g_{jk} m_{jk} \leq b_i \qquad i = 1, 2, \ldots, s$$

$$\sum_{j=1}^{N} \sum_{k=0}^{m'_j} \Delta \ln R'_{jk} m_{jk} \geq \ln R_{s,\min}$$

and

$$m_{jk} = 1 \qquad \text{for } k = 0$$

$$m_{jk} - m_{j,k-1} \leq 0 \qquad \text{for } k = 1, \ldots, m'_j$$

$$j = 1, \ldots, N$$

$$m_{jk} \geq 0 \qquad \text{for all } j \text{ and } k \qquad (2)$$

where in addition to the same notations used in problem (1).

k = index used to denote a particular redundant unit at stage j

Now the new variable m_{jk} is introduced which represents the kth redundancy at stage j and is defined as follows

$$m_{jk} = 1 \qquad \text{for } 0 \leq k \leq m_j$$

$$= 0 \qquad m_j < k \leq m'_j \qquad (3)$$

with the obvious result that

$$m_j = \sum_{k=1}^{m_j} m_{jk} \tag{4}$$

In the above it is understood that the Δ in R'_{jk} are numerically evaluated coefficients. To complete the integer programming formulation it is necessary to formulate the relationships of (3) as restrictions. Equation (3) includes the requirement that each subsystem shall contain at least one component. This is accomplished by including the following restrictions

$$m_{jk} = 1 \qquad \text{for } k = 0, \quad j = 1, \ldots, N \tag{5}$$

The remaining part of (4) insures that at each stage j, the kth redundant unit m_{jk} equals one if it is in the solution and that it is in the solution only if the (k-1)th redundant unit is included. This is incorporated into the problem by including the restraints

$$m_{jk} \leq m_{j,k-1} \qquad k = 1, \ldots, m'_j; \quad j = 1, \ldots, N \tag{6}$$

Thus including (5) and (6) completes the formulation of the problem (1) as an integer programming problem as stated by (2).

m_{jk} = the variable representing the kth redundancy at stage j, where $m_{jk} = 1$ for $k \leq m_j$ and $m_{jk} = 0$ for $m_j < k \leq m'_j$

$\Delta f_{jk} = f_{jk}$ for $k = 0$

$= f_{jk} - f_{j,k-1}$ for $k = 1, \ldots, m_j$

is the change in $f_j(m_j)$ by adding the kth redundancy at stage j, where f_{jk} is the objective function of stage j when exactly k redundant units are used.

$$\Delta g_{ijk} = g_{ijk} \qquad \text{for } k = 0$$

$$= g_{ijk} - g_{ij,k-1} \qquad \text{for } k = 1, \ldots, m_j$$

$$k = 1, \ldots, s$$

is the change in $g_{ij}(m_j)$ by adding the kth redundancy at stage j and where g_{ijk} is the function of the ith resource consumed when k redundant units are used at stage j.

$$\Delta \ln R'_{jk} = \ln R'_{jk} \qquad \text{for } k = 0$$

$$= \ln R'_{jk} - \ln R'_{j,k-1} \qquad \text{for } k = 1, \ldots, m_j$$

is the change in $\ln R'_j$ by adding the kth redundancy at stage j, and where R'_{jk} is the reliability at stage j when k redundant units are used.

Example 2-3

To solve this example, the minimum system reliability requirement,

$$R_s \geq R_{s,min}$$

should be transformed into

$$\ln R_s \geq \ln R_{s,min} = -0.1625$$

This can be written as

$$-0.1625 \leq \sum_{j=1}^{2} \sum_{k=0}^{4} \Delta \ln R'_{jk} m_{jk}$$

or

$$0,1625 \geq - \sum_{j=1}^{2} \sum_{k=0}^{4} \Delta \ln R'_{jk} m_{jk}.$$

The objective is to determine m_j, the number of redundant units as stage j, that minimizes the following cost function.

$$Z = [3m_1 e^{-m_1/2}] + [2m_2 e^{-m_2/2}]$$

while not violating the system restraints:

$$g_1 = [3m_1 + m_1^2] + [3m_2 + m_2^2 + 1] \leq 37$$

$$g_2 = [30(m_1 + e^{-m_1})] + [30(m_2 + e^{-m_2}) - 4] \geq 81$$

$$g_3 = [20m_1 e^{-m_1/4}] + [20m_2 e^{-m_2/4}] \geq 38$$

$$g_4 = \prod_{j=1}^{2} [1 - (1-R_j)^{m_j+1}] \geq 0.85$$

Problems with fewer linear constraints such as Example 2-1 can be readily solved by methods presented in [5,16], but it seems that these methods are inadequate for solving Example 2-3 which includes multiple nonlinear restraints. The integer programming formulation of this problem is illustrated in Fig. 2-2.

The Group I equations represent the greater than restriction, and the Group IV equations represent the less than restrictions on the system. The Group II equations insure that one basic unit is in each stage. The Group III equations allow the kth redundant unit to be in the solution only if the (k-1)th redundant unit is included and requires

			Stage I					Stage II				
			m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{20}	m_{21}	m_{22}	m_{23}	m_{24}
Group I	81	$\leq$	30	11	23	27	29	26	11	23	27	29
	38	$\leq$	0	23	13	6	2	0	23	13	6	2
Group II	1	=	1									
	1	=						1				
Group III	0	$\geq$	-1	1								
	0	$\geq$		-1	1							
	0	$\geq$			-1	1						
	0	$\geq$				-1	1					
	0	$\geq$						-1	1			
	0	$\geq$							-1	1		
	0	$\geq$								-1	1	
	0	$\geq$									-1	1
Group IV	37	$\geq$	0	4	6	8	10	1	4	6	8	10
	0.1625	$\geq$	0.1054	-.0953	-.0090	-.0009	-.0001	0.2877	-.2231	-.0488	-.0118	-.0029
	C_j		0.0	-1.81959	-0.38768	0.19911	0.38415	0.0	-1.21506	-0.25846	0.13274	0.25610

Fig. 2-2

The integer programming formulation of Example 2-3.

the m_{jk} variables to be either zero or one. This minimization problem is converted to a maximization problem by multiplying the objective function by (-1). The c_j equation is the objective function to be maximized. The integer programming solution is as follows

$$m_{10} = 1 \qquad m_{20} = 1 \qquad m_{22} = 1 \qquad m_{24} = 1$$

$$m_{21} = 1 \qquad m_{23} = 1$$

and all other $m_{jk} = 0$. To summarize, there are

$m_1 = 0$ redundant units at stage 1.

$m_2 = 4$ redundant units at stage 2.

The minimum cost $Z = 1.0827$ and the system reliability $R_s = 0.899$ where $\ln R_s = -0.1065$.

5. The Geoffrion's Implicit Enumeration Method

Geoffrion's implicit enumeration method was to solve the reliability optimization problem with two classes of failure modes [9]. A formulation by 0-1 linear programming [18,19] is introduced.

Notation and nomenclature:

k-out-of-n:F — the system is failed if and only if at least k out of its n elements are failed.

k-out-of-n:G — the system is good if and only if at least k out of its n elements are good.

m_i — number of failure modes in subsystem i.

s_i — total number of failure modes in subsystem i.

Class O failure modes those of which subsystem i is 1-out-of-m_j:F

h_i number of class O failure modes in subsystem i(u=1, 2, ..., h_i)

Class A failure modes those of which subsystem i is 1-out-of-m_i:G

s_i-h_i number of class A failure modes in subsystem i ($u=h_i+1, h_i+2, \ldots, s_i$)

q_{iu} probability of failure mode u for each element in subsystem i.

b_t the amount of system resource t available, size of the constraint

T number of kinds of resources (t=1, 2, ..., T)

$g_{ti}(m_i)$ subsystem i requires this much of resource t when there are m_i elements in it.

$Q_u^O(m_i), Q_u^A(m_i)$ failure probability in subsystem i (m_i elements) for failure mode u of the class O or A failure modes.

$Q^O(m_i), Q^A(m_i)$ failure probability in subsystem i (m_i elements) subject to the class O or A failure modes.

$Q_i(m_i)$ failure probability in subsystem i (m_i elements)

R(m) reliability of the system when the element allocation is m.

$G_t(m)$ the system requires this much of resource t when the element allocation is m.

X matrix with element x_{ij}

N number of subsystems

v_i upper bound of m_i elements in subsystem i

γ_i lower bound of m_i elements in subsystem i (γ_i=k for k-out-of-m_i:G subsystems)

$Q^S_{iu}(m_i), Q^P_{iu}(m_i)$ probability of failure for subsystem i (m_i elements) by mode u; the modes are separated into classes S & P

Assumptions:

1. All elements in subsystem i are s-independent with respect to failure mode u; for each i, u combination taken by itself.
2. The system is 1-out-of-N:F, with respect to its subsystems.
3. The failure modes are a partitioning (mutually exclusive and exhaustive) of the failure event. Thus failure probabilities for failure modes add to get total failure probability.
4. Within a subsystem, the elements are 1-out-of-m_i:G for some failure modes; and at the same time are 1-out-of-m_i:F for the other failure modes.

We state the original system reliability problem as

Problem A

Maximize

$$R_S = \prod_{i=1}^{N} [1 - Q_i(m_i)] \tag{1}$$

subject to

$$G_t(m) = \prod_{i=1}^{N} g_{ti}(m_i) \leq b_t, \quad t = 1, 2, \ldots, T. \tag{2}$$

where

$$Q_i(m_i) = Q^O(m_i) + Q^A(m_i)$$

$Q^O(m_i)$ and $Q^A(m_i)$ are the unreliability of subsystem i obtained for class O failure modes and for class A failure modes, respectively. [9]

To formulate Problem A into a 0-1 linear programming problem, we define the following 0-1 variable:

$$x_{ij} = \begin{cases} 1 \text{ ; allocate } j \text{ elements to subsystem } i, \\ 0 \text{ ; otherwise.} \end{cases} \tag{3}$$

When we introduce this 0-1 variable the nonlinear system reliability (Problem A) of the NIP-m problem can be expressed by the following linearized objective function:

$$f(X) = \sum_{i=1}^{N} \sum_{j=\gamma_i}^{v_i} c_{ij} x_{ij}, \tag{4}$$

where, for all i and j,

$$c_{ij} \equiv \ln\left[1 - \left\{ \sum_{u=1}^{h_i} Q_{iu}^P(j) + \sum_{u=h_i+1}^{s_i} Q_{iu}^S(j) \right\}\right] .$$

$$Q_{iu}^P(j) \equiv 1 - (1-q_{iu})^{j+1}, \; Q_{iu}^S(j) \equiv (q_{iu})^{j+1} \tag{5}$$

When we introduce the 0-1 variable into the T nonlinear constraints (2), we get

$$g_t(X) = \sum_{i=1}^{N} \sum_{j=\gamma_i}^{v_i} a_{tij} x_{ij} \leq b_t, \quad t = 1, 2, \ldots, T, \tag{6}$$

$$a_{tij} \equiv g_{ti}(j), \quad \text{for all } t, i, \text{ and } j. \tag{7}$$

By definition of the 0-1 variable (3), we add the following N linear constraints to the constraints (6):

$$g_{T+i}(X) \equiv 1 - \sum_{j=\gamma_i}^{v_i} x_{ij} \geq 0, \quad i = 1, 2, \ldots, N. \tag{8}$$

By introducing the 0-1 variable, we have therefore reformulated problem A into a ZOLP problem which maximizes the linear objective function (4) - (5) subject to the T+N linear constraints (6) - (8); this is the ZOLP-m problem. It is proved, as follows, that there is a one-to-one correspondence between the NIP-m and the ZOLP-m proposed here.

First we prove that (4) and (5) are correct. It is obvious that (5) is necessary. In order to prove that it is sufficient, substitute (5) into (4):

$$f(X) = \sum_{i=1}^{N} \sum_{j=\gamma_i}^{v_i} \&n[1 - \{Q_i^O(j) + Q_i^A(j)\}] x_{ij}$$

$$= \sum_{i=1}^{N} \left\{ \sum_{j \in Z_i^+} \&n[1 - \{Q_i^O(j) + Q_i^Q(j)\}] x_{ij} + \right.$$

$$\left. \sum_{j \in \bar{Z}_i^+} \&n[1 - \{Q_i^O(j) + Q_i^A(j)\}] x_{ij} \right\} \tag{9}$$

where

$$Q_i^O(j) \equiv \sum_{u=1}^{h_i} Q_{iu}^P(j), \quad Q_i^A(j) \equiv \sum_{u=h_i+1}^{s_i} Q_{iu}^S(j),$$

and

$$Z_i = [j : \gamma_i, \ \gamma_i + 1, \ \ldots, \ v_i]$$

is a set of subsystem i and a direct sum of the Z_i^+ and $\bar{Z}_i^+$ (which are partitionings of Z_i). Let X* be a feasible solution to the ZOLP-m problem. Then, (9) is as follows:

$$f(X^*) = \sum_{i=1}^{N} \sum_{j \in Z_i^+} \ln[1 - \{Q_i^O(j) + Q_i^A(j)\}] x_{ij}$$

$$= \sum_{i=1}^{N} \ln[1 - \{Q_i^O(j_i^*) + Q_i^A(j_i^*)\}] = \ln R(j^*), \qquad (10)$$

where

$$j^* = (j_1^*, j_2^*, \ldots, j_n^*) .$$

Now we prove that (6) and (7) are correct. It is obvious that (7) is necessary. In order to prove that it is sufficient, substitute (7) into (6):

$$g_t(X) = \sum_{i=1}^{N} \sum_{j=\gamma_i}^{v_i} g_{ti}(j) x_{ij} = \sum_{i=1}^{N} \left\{ \sum_{j \in Z_i^+} t_{ti}(j) x_{ij} + \sum_{j \in Z_i^+} g_{ti}(j) x_{ij} \right\}, \quad t = 1, 2, \ldots, T. \qquad (11)$$

Let X^* be a feasible solution to the ZOLP-m problem, then (11) is

$$g_t(X^*) = \sum_{j=1}^{N} \sum_{j \in Z_j^+} g_{ti}(j) x_{ij} = \sum_{i=1}^{N} g_{ti}(j_i^*)$$

$$= G_t(j^*) \leq b_t, \quad t = 1, 2, \ldots, T. \qquad (12)$$

Example 2-4

The ZOLP-m problem is to maximize the linear objective function (4) - (5) subject to the linear constraints (6) - (8) for N=3 and T=3,

where the coefficients a_{tij} of (6) for j = 1, 2, 3, 4 are:

$$a_{11j} = (3 + j)^2, \quad a_{12j} = (j)^2, \quad a_{13j} = (2 + j)^2,$$

$$a_{2ij} = -20(j + \exp(-j)), \quad a_{3ij} = 20j \exp(-j/4)$$

for i = 1, 2, 3.

$$b_1 = 51, \quad b_2 = -120, \quad b_3 = -65.$$

The ZOLP-m problem is illustrated in Table 2-6 in the required ZOLP-m formulations which 1000 times the coefficient c_{ij} for all i and j of the linear objective function (4). The variables in Table 2-6 can be converted into single subscript variables as follows:

$$x_{1j} = x_j, \quad x_{2j} = x_{4+j}, \quad x_{3j} = x_{8+j}.$$

The feasible and optimal solution of the ZOLP-m example are shown in Table 2-7; the optimal solutions are $x_2 = 1$, $x_5 = 1$, and $x_{11} = 1$.

Table 2-6

The integer programming formulation of Example 2-4

Objective Function, x_{ij}

i \ j:	1	2	3	4
1	67.0	37.3	41.4	50.5
2	211.8	256.7	334.5	417.1
3	140.3	132.8	165.3	204.5

Constraints

i \ j:	1	2	3	4	1	2	3	4
1	-16	-25	-36	-49	27.4	42.7	61.0	80.4
2	-1	-4	-9	-16	----------same----------			
3	-9	-16	-25	-36	----------same----------			
	#1, $G_1 \leq 51.0$				#2, $G_2 \leq -120.0$			
1	15.6	24.3	28.3	29.4	-1	-1	-1	-1
2	----------same----------				0	0	0	0
3	----------same----------				----------same----------			
	#3, $G_3 \leq -65.0$				#4, $G_4 \leq 1$			
1	0	0	0	0	0	0	0	0
2	-1	-1	-1	-1	----------same----------			
3	0	0	0	0	-1	-1	-1	-1
	#5, $G_5 \leq 1$				#6, $G_6 \leq 1$			

Table 2-7

The feasible and optimal solutions to Example 2-4

		Feasible Solutions							
i	j:	1	2	3	4	1	2	3	4
1		1	0	0	0	0	1	0	0
2		0	0	0	1	0	0	0	1
3		0	1	0	0	1	0	0	0
		step 37				step 41			
1		0	1	0	0	0	1	0	0
2		0	0	1	0	1	0	0	0
3		1	0	0	0	0	0	1	0
		step 61				step 91			
1		0	1	0	0				
2		1	0	0	0	Optimal Solution			
3		0	0	1	0				

$m_1^* = 2, \quad m_2^* = 1, \quad m_3^* = 3.$

REFERENCES

1. Kuo, Way, Optimization Techniques for Systems Reliability with Redundancy, M.S. Thesis, Kansas State University, 1978.

2. Ghare, P. M. and R. E. Taylor, "Optimal redundancy for reliability in series system", Operations Research, vol. 17, pp. 838-847 (Sept. 1969).

3. McLeavey, D. W., Numerical investigation of parallel redundancy in series systems", Operations Research, vol. 22, pp. 1110-1117 (Sept. - Oct. 1974).

4. McLeavey, D. W. and J. A. McLeavey, "Optimization of system reliability by branch-and-bound", IEEE Transactions on Reliability, Vol. R-25, No. 5, pp. 327-329 (December 1976).

5. Proschan, F. and T. A. Bray, "Optimal redundancy under multiple constraints", Operations Research, Vol. 13, pp. 800-814 (1965).

6. Misra, K. B., "A method of solving redundancy optimization problems", IEEE Transactions on Reliability, Vol. R-20, No. 3, pp. 117-120 (August 1971).

7. Tillman, F. A., "Integer programming solutions to constrained reliability optimization problems", Transactions of Twentieth Annual Technical Conference American Society for Quality Control, paper number 66-174, pp. 676-693 (1966).

8. Tillman, F. A. and J. M. Liittschwager, "Integer programming formulation of constrained reliability problems", Management Science, Vol. 13, No. 11, pp. 887-899 (July 1969).

9. Tillman, F. A., "Optimization by integer programming of constrained reliability problems with several modes of failure", IEEE Transactions on Reliability, Vol. R-18, No. 2, pp. 47-53 (May 1969).

10. Hwang, C. L., L. T. Fan, F. A. Tillman, and S. Kumar, "Optimization of life support system reliability by an integer programming method", AIIIE Transactions, Vol. 3, No. 3, pp. 229-238 (September 1971).

11. Misra, K. B. and J. Sharma, "Reliability optimization of a system by zero-one programming", Microelectronics and Reliability, Vol. 12, pp. 229-233 (June 1973).

12. Luus, R., "Optimization of system reliability by a new nonlinear integer programming procedure", IEEE Transactions on Reliability, Vol. R-24, pp. 14-16 (April 1975).

13. Kolesar, P. J., "Linear programming and the Reliability of multi-component systems", Naval Research Logistics Quarterly, Vol. 15, pp. 317-327 (Sept. 1967).

14. Hyun, K. N., "Reliability optimization by 0-1 programming for a system with several failure modes", IEEE Transactions on Reliability, Vol. R-24, No. 3, pp. 206-210 (August 1975).

15. Lawler, E. L., and M. D. Bell, "A method for solving discrete optimization problems", Operations Research, Vol. 14, pp. 1098-1112 (Nov. - Dec. 1966).

16. Bellman, R. E., and S. E. Dreyfus, "Dynamics programming and the reliability of multicomponent devices", Operations Research, Vol. 6, pp. 200-206 (Mar.-Apr. 1958).

17. Balas, E., "A note on the branch-and-bound principle," Operations Research, Vol. 16, pp. 442-445 (1968).

18. Geoffrion, A. M., "An improved implicit enumeration approach for integer programming", RAND Corp. Rept. RM-5644-PR June 1968, or Oper. Res. Vol. 17, pp. 437-454 May-June (1969).

19. Hyun, K. N., "Reliability optimization by 0-1 programming for a system with several failure modes", IEEE Transactions on Reliability, Vol. R-24, No. 3, pp. 206-210 (August 1975).

Chapter 3

NONLINEAR INTEGER GOAL PROGRAMMING APPLIED TO OPTIMAL SYSTEM RELIABILITY

1. Introduction

In the previous chapter several integer programming methods to solve single objective reliability problems are introduced. In practice, however, there arise a lot of multiple objective decision problems in the system reliability optimization. For example a decision maker may want to minimize the total cost and maximize the system reliability at the same time, or he may want to make a system which has high reliability, small volume, light weight, and low cost. Also there always exists conflict among the decision maker's objectives or the attributes of a system.

The purpose of this chapter is to present an algorithm to solve the multiple objective optimization problems which need all integer solution. The algorithm, which can be called nonlinear integer goal programming, is formulated using Hwang and Paidy's nonlinear goal programming technique [1] and the branch-and-bound algorithm applied in Ignizio's linear integer goal programming [2].

2. Algorithm

The general form of the multiple objective model considered within this chapter is given as:

$$
\left.
\begin{aligned}
&\text{Find} \quad \bar{x} = (x_1, x_2, \ldots, x_J) \text{ so as to minimize} \\
&\bar{a} = \{[g_1(\bar{n}, \bar{p})], [g_2(\bar{n}, \bar{p})], \ldots, [g_k(\bar{n}, \bar{p})]\} \\
&\text{such that} \\
&f_i(x_1, x_2, \ldots, x_J) + n_i - p_i = b_i, \quad \forall i \\
&\bar{x}, \bar{n}, \bar{p} \geq 0 \\
&x_t = 0, 1, 2, \ldots \text{ for all } t
\end{aligned}
\right\} \qquad (3\text{-}1)
$$

As may be seen, the only difference between this formulation and that of the general nonlinear goal programming is the integer constraints:

$$x_t = 0, 1, 2, \ldots \text{ for all } t$$

The technique of Branch-and-Bound employed in linear integer goal programming by Ignizio [2] was Dakin algorithm [3,4]. Branch-and-Bound, in essence, is a way to solve integer programming problems by implicitly enumerating all possible solution combinations. The Dakin algorithm, when applied to nonlinear integer goal programming, proceeds by first solving the problem by general nonlinear goal programming (i.e., ignoring the requirements of the integer variables). If the solution achieved by nonlinear goal programming happens to satisfy all the integer conditions, the optimal solution is found. If not, one variable, that is not integer valued and that provides the most preferrable achievement function value in the nonlinear goal programming solution, is

selected. Using this variable, two new absolute objectives are developed as follows:

Let x_s be an integer constrained variable that is presently fractional valued where the fractional value of x_s is given by $\hat{x}_s$. Then let $[\hat{x}_s]$ represent the largest integer that is less than the value $\hat{x}_s$. Thus, the range given by:

$$[\hat{x}_s] < x_s < [\hat{x}_s] + 1$$

is infeasible for x_s. To avoid a solution in this range the following two objectives may be utilized:

$$x_s \leq [\hat{x}_s] \tag{3-2}$$

$$x_s \geq [\hat{x}_s] + 1 \tag{3-3}$$

Dakin lets each of the above objectives represent a new "branch" from the previous problem. The problem associated with each new branch is the goal programming problem as previously specified plus the new objective associated with the branch under consideration. These new, augmented problems are solved and the process is repeated according to the certain rules until convergence to an optimal integer solution is obtained. In this process priority level one must be reserved solely for absolute objectives. Figure 3-1 serves to illustrates the basic procedure.

Some simplifying notation and concepts used in describing the nonlinear integer goal programming algorithm are specified as follows:

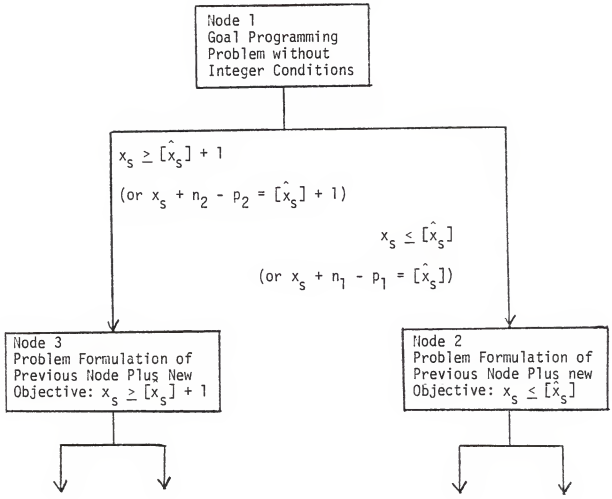

Figure 3-1. Branch-and-Bound Schematic

$\bar{x}_q$ represents the solution, as achieved via nonlinear goal programming, at node q.

$\bar{\gamma}_q$ represents the value of $\bar{a}$, as computed by nonlinear goal programming, at node q (and thus is a lower bound for mode q).

$\bar{\gamma}^*$ represents the best feasible solution (i.e., all variables are integer) thus far obtained.

GP_q represents the goal programming formulation at node q.

NG_q represents the new objective (from either (3-2) or (3-3)) at node q.

For any two solutions, say $\bar{\gamma}_a$ and $\bar{\gamma}_b$, $\bar{\gamma}_a$ is preferred to $\bar{\gamma}_b$ if a priority level of $\bar{\gamma}_a$ is lower in value than the corresponding priority levels in $\bar{\gamma}_b$ and all preceding priority levels are equal in both $\bar{\gamma}_a$ and $\bar{\gamma}_b$. For example, if:

$$\bar{\gamma}_3 = (0, 170, 18, 200)$$

and

$$\bar{\gamma}_5 = (0, 170, 14, 303)$$

then $\bar{\gamma}_5$ is preferred to $\bar{\gamma}_3$.

The details of the nonlinear integer goal programming algorithm is stated as follows:

Step 1: Establish the problem in terms of the goal programming formulation as specified in equation (3-1). Reserve priority level one solely for absolute objectives.

Step 2: Set q=1, and $\bar{\gamma}^*$ = (M, M, ..., M) where M is some arbitrary high value. Designate the goal programming formulation, excluding integer condition, as GP_q. Solve GP_q via nonlinear goal programming at node q so as to establish $\bar{x}_q$ and $\bar{\gamma}_q$.

Step 3: Check the first term in $\bar{\gamma}_q$. If this term is positive then one or more of the absolute objectives have been violated and thus no implementable solution exists at this node. Terminate this node and go to Step 5 (termination of a node infers that the node cannot be branched from in any future steps). If, however, the first term in $\bar{\gamma}_q$ is not positive go to Step 4 (without terminating node q).

Step 4: Compare $\bar{\gamma}_q$ with $\bar{\gamma}^*$. If $\bar{\gamma}_q = \bar{\gamma}^*$ and $\bar{x}_q$ is all integer solution, terminate node q and go to Step 5. If $\bar{\gamma}_q$ is preferred to $\bar{\gamma}^*$, go to Step 6 (without terminating node q). However, if $\bar{\gamma}^*$ is preferred to $\bar{\gamma}_q$, terminate node q and go to Step 5.

Step 5: If all nodes have been terminated (not including those which have been branched from), then stop and specify $\bar{\gamma}^*$ as the optimal solution. If any nodes are unterminated, proceed to Step 6.

Step 6: If q is even valued, go to Step 9. Otherwise go to Step 7.

Step 7: Check all $\bar{\gamma}_q$ values associated with any node not yet terminated or branched from. Select the node with the most preferable $\bar{\gamma}_q$ as the node to be branched on next. Go to Step 8.

Step 8: Try branching with each variable which has a fractional value at the node selected to be branched on next as follows:

(1) Select a fractional valued variable (designated as x_s) for branching.

(2) Specify a new objective $x_s \leq [\hat{x}_s]$, where $\hat{x}_s$ represent the fractional value of x_s.

(3) Add this objective to the goal programming formulation of the previous node[a] to obtain GP_{q+1}. Solve GP_{q+1} to establish $\overline{x}_{q+1}$ and $\overline{\gamma}_{q+1}$.

(4) If $\overline{x}_{q+1}$ contains all integers and $\overline{\gamma}_{q+1}$ is better than $\overline{\gamma}^*$, set $\overline{\gamma}^* = \overline{\gamma}_{q+1}$ and terminate those nodes in which $\overline{\gamma}^*$ is preferred. Otherwise $\overline{\gamma}^*$ remains at the same value.

(5) Specify a new objective $x_s \geq [\hat{x}_s] + 1$.

(6) Add this objective to the goal programming formulation of the previous node[b] to obtain GP_{q+2}. Solve GP_{q+2} to establish $\overline{x}_{q+2}$ and $\overline{\gamma}_{q+2}$.

(7) If $\overline{x}_{q+2}$ contains all integers and $\overline{\gamma}_{q+2}$ is better than $\overline{\gamma}^*$, set $\overline{\gamma}^* = \overline{\gamma}_{q+2}$ and terminate those nodes in which $\overline{\gamma}^*$ is preferred. Otherwise $\overline{\gamma}^*$ remains at the same value.

After going through above procedures for every fractional valued variable, select a branching which gives the most preferable $\overline{\gamma}_{q+1}$ or $\overline{\gamma}_{q+2}$ for the new two nodes. Go to Step 9.

---

[a] That is, the node branched from to reach this node.

[b] Again, the node branched from to reach this node.

Step 9: Set q = q+1. Go to Step 3.

In the next section numerical examples in the system reliability optimization are provided to illustrate the algorithm.

3. Numerical examples

Example 3-1:

$$\min \bar{a} = [(p_1 + p_2), (n_3)]$$

$$G_1: \quad 1.2x_1 + 2.3x_2 + 3.4x_3 + 4.5x_4 + n_1 - p_1 = 35.6$$

$$G_2: \quad x_1 + x_2 + x_3 + x_4 + n_2 - p_2 = 16$$

$$G_3: \quad (1 - .20^{x_1+1})(1 - .30^{x_2+1})(1 - .25^{x_3+1})(1 - .15^{x_4+1}) + n_3 - p_3 = 1$$

where x_1, x_2, x_3, and x_4 are nonnegative integers

and $\bar{n}, \bar{p} \geq \bar{0}$

The solution to the problem is summarized via the Branch-and-Bound schematic of Figure 3-2. The objectives shown at each branch are those that must be included along with the goal programming formulation of the preceding node to form the problem to be solved at the corresponding node.

The progress of the algorithm may be described with reference to Figure 3-2 as follows:

Node 1: The original goal programming problem is solved. No variable is integer. Each variable is tried for branching and lower bounds are found as shown in Table 3-1. x_1 has the lowest bound and is picked for

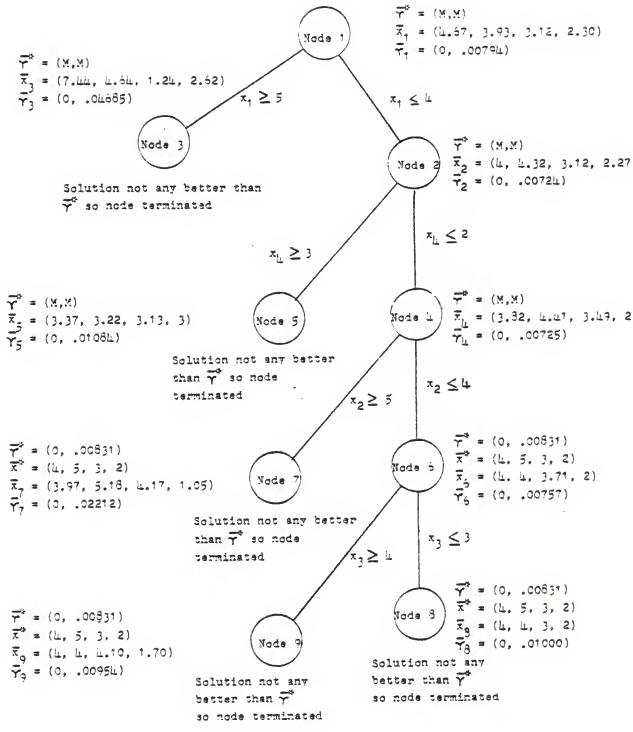

Figure 3-2 Solution to Example 3-1

Table 3-1.

Trial branchings from node 1 of Example 3-1

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_1 \leq 4$	4.00, 4.32, 3.12, 2.27	0, .00724	lowest bound
$x_1 \geq 5$	7.44, 4.64, 1.24, 2.62	0, .04685	
$x_2 \leq 3$	3.83, 3.00, 3.75, 2.52	0, .01113	
$x_2 \geq 4$	3.18, 3.15, 3.05, 3.15	.85, .01195	
$x_3 \leq 3$	3.18, 3.18, 3.00, 3.17	0, .01198	
$x_3 \geq 4$	3.52, 3.47, 4.10, 2.10	0, .00889	
$x_4 \leq 2$	4.02, 4.00, 3.70, 2.00	0, .00758	
$x_4 \geq 3$	3.37, 3.22, 3.13, 3.00	0, .01084	

next branching. $x_1 \leq 4$ and $x_1 \leq 5$ are used as the new branching objectives.

Node 2: Node 2 is the same goal programming problem as at node one with one additional objective of $x_1 + n_4 - p_4 = 4$ (where p_4 is to be minimized at priority level 1). The solution of this problem can be found from Table 3-1, i.e., $x_2 = (4, 4.32, 3.12, 2.27)$ and $\overline{\gamma}_2 = (0, .00724)$. Move to node 3.

Node 3: Node 3 is the same goal programming problem as at node 1 with the additional objective of $x_1 + n_4 - p_4 = 5$ (where n_4 is to be minimized at priority level 1). The solution of this problem can be found from Table 3-1, i.e., $\overline{x}_3 = (7.44, 4.64, 1.24, 2.62)$ and $\overline{\gamma}_3 = (0, .04685)$. The best $\overline{\gamma}_q$ at an unterminated node is that associated with node 2. Thus each fractional valued variable at node 2 is tried for branching as shown in Table 3-2. x_4 has the lowest bound and is picked for next branching. $x_4 \leq 2$ and $x_4 \geq 3$ are used as the new objectives.

Node 4: Node 4 has the same goal programming formulation as its immediate predecessor, node 2, plus the additional absolute objective of $x_4 + n_5 - p_5 = 2$ (where p_5 is to be minimized at priority level one). The solution of this problem can be found from Table 3-2, i.e., $\overline{x}_4 = (3.82, 4.41, 3.49, 2)$ and $\overline{\gamma}_4 = (0, .00725)$. Moved to node 5.

Node 5: Node 5 represents the same goal programming problem as at node 2 plus the additional absolute objective of $x_4 + n_5 - p_5 = 3$ (where n_5 is to be minimized at priority level 1). The solution can be found from Table 3-2, i.e., $\overline{x}_5 = (3.37, 3.22, 3.13, 3)$ and $\overline{\gamma}_5 = (0, .01084)$.

Table 3-2.

Trial branchings from node 2 of Example 3-1.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_2 \leq 4$	3.93, 3.90, 3.24, 2.42	0, .00740	
$x_2 \geq 5$	3.18, 3.15, 3.05, 3.15	1.85, .01195	
$x_3 \leq 3$	3.18, 3.18, 3.00, 3.17	0, .01198	
$x_3 \geq 4$	3.52, 3.47, 4.10, 2.10	0, .00889	
$x_4 \leq 2$	3.82, 4.41, 3.49, 2.00	0, .00725	lowest bound
$x_4 \geq 3$	3.37, 3.22, 3.13, 3.00	0, .01084	

The best $\overline{\gamma}_q$ at an unterminated node is that associated with node 4. Thus each fractional valued variable at node 4 is tried for branching as shown in Table 3-3. x_2 has the lowest bound and is picked for next branching. $x_2 \leq 4$ and $x_2 \geq 5$ are used as the new objectives. Note from Table 3-3 that $\overline{x}$ has all integer solution when $x_3 \leq 3$, i.e., $\overline{x}$ = (4, 5.04, 3, 2) and $\overline{\gamma}$ = (0, .00827), which can be approximated as $\overline{x}$ = (4, 5, 3, 2) and $\overline{\gamma}$ = (0, .00831). Obviously $\overline{\gamma}$ = (0, .00831) is better than $\overline{\gamma}^*$ = (M, M). Set $\overline{\gamma}^*$ = (0, .00831) and $\overline{x}^*$ = (4, 5, 3, 2). Node 3 and node 5 are terminated because their solution is not any better than $\overline{\gamma}^*$. Move to node 6.

Node 6: Node 6 has the same goal programming formulation as its immediate predecessor, node 4, plus the additional absolute objective of $x_2 + n_6 - p_6 = 4$ (where p_6 is to be minimized at priority level one). The solution is, from Table 3-3, $\overline{x}_6$ = (4, 4, 3.71, 2) and $\overline{\gamma}$ = (0, .00757). Move to node 7.

Node 7: Node 7 represents the same goal programming problem as at node 4 plus the additional absolute objective of $x_2 + n_6 - p_6 = 5$ (where n_6 is to be minimized at priority level one). The solution is, from Table 3-3, $\overline{x}_7$ = (3.98, 5.18, 4.17, 1.05) and $\overline{\gamma}_7$ = (0, .02212). Node 7 is terminated because $\overline{\gamma}_7$ is not any better than $\overline{\gamma}^*$.

The only node that has not been yet terminated is Node 6. All the variables except x_3 have integer value. Thus, x_3 is tried for branching for the next new two nodes as shown in Table 3-4. $x_3 \leq 3$ and $x_3 \geq 4$ are used as the new objectives. Note from Table 3-4 that $\overline{x}$ has all integer solution when $x_3 \leq 3$ but $\overline{\gamma}$=(0, .01000) is not any better than

Table 3-3.

Trial branchings from node 4 of Example 3-1.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_2 \leq 4$	4.00, 4.00, 3.71, 2.00	0, .00757	lowest bound
$x_2 \geq 5$	3.98, 5.18, 4.17, 1.05	0, .02212	
$x_3 \leq 3$	4.00, 5.04, 3.00, 2.00	0, .00827	feasible
$x_3 \geq 4$	3.79, 4.22, 4.10, 1.65	0, .00973	

Table 3-4.

Trial branching from node 6 of Example 3-1.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_3 \leq 3$	4.00, 4.00, 3.00, 2.00	0, .01000	Feasible
$x_3 \geq 4$	4.00, 4.00, 4.10, 1.70	0, .00954	

$\bar{\gamma}^*$ = (0, .00831). Hence $\bar{\gamma}^*$ remains at the same value.

Node 8: Node 8 has the same goal programming formulation as its immediate predecessor, node 6, plus the additional absolute objective of $x_3 + n_7 - p_7 = 3$ (where p_7 is to be minimized at priority level one). The solution is, from Table 3-4, $\bar{x}_8$ = (4, 4, 3, 2) and $\bar{\gamma}_8$ = (0, .01000). $\bar{\gamma}_8$ is not any better than $\bar{\gamma}^*$, so node 8 is terminated. Move to Node 9.

Node 9: Node 9 has the same goal programming formulation as its immediate predecessor, node 6, plus the additional absolute objective of $x_3 + n_7 - p_7 = 4$ (where n_7 is to be minimized at priority level one). The solution is, from Table 3-4, $\bar{x}_9$ = (4, 4, 4.10, 1.70) and $\bar{\gamma}_9$ = (0, .00954). $\bar{\gamma}_9$ is not any better than $\bar{\gamma}^*$, so node 9 is terminated.

At this point, all nodes are terminated and the optimal solution is $\bar{x}^*$ = (4, 5, 3, 2) and $\bar{a}^* = \bar{\gamma}^*$ = (0, .00831).

Note that this example problem is the reformulation of Example 2-2 by taking the constraints as the priority level one and the objective as the priority level two. Both solutions are the same, which is a rational result.

Example 3-2:

$$\min \bar{a} = \{(p_1 + n_2 + n_3), (n_4)\}$$

$$G_1: \quad (x_1+3)^2 + x_2^2 + (x_3+2)^2 + n_1 - p_1 = 51$$

$$G_2: \quad 20(x_1+\exp(-x_1)) + 20(x_2+\exp(-x_2)) + 20(x_3+\exp(-x_3)) + n_2-p_2 = 120$$

$$G_3: \quad 20(x_1\exp(-x_1/4)) + 20(x_2\exp(-x_2/4)) + 20(x_3\exp(-x_3/4)) + n_3 - p_3 = 65$$

$$G_4: \quad \{1-[1-(1-.01)^{x_1+1}] - .05^{x_1+1} - .10^{x_1+1} - .18^{x_1+1}\}$$

$$\{1 - [1 - (1 - .08)^{x_2+1}] - .02^{x_2+1} - .15^{x_2+1} - .12^{x_2+1}\}$$

$$\{1 - [1 - (1 - .04)^{x_3+1}] - .05^{x_3+1} - .20^{x_3+1} - .10^{x_3+1}\} + n_4 - p_4 = 1$$

where x_1, x_2, and x_3 are nonnegative integers and $\overline{n}, \overline{p} \geq \overline{0}$.

This problem is the reformulation of Example 2-4 by taking the constraints as priority level one and the objective as priority level two. The solution to Example 3-2 is given in Figure 3-3. Trial branchings from each node are presented in Tables 3-5 through 3-9. Note that in solving this problem the actual optimal solution is found at node 8 but must proceed to investigate three additional nodes (one of which is also an optimal) before the search is terminated. The solution to the problem is $\overline{x}^* = (2, 1, 3)$ and $\overline{\gamma}^* = (0, .33924)$, which is the same as that of Example 2-4 as expected.

Example 3-3:

$$\min \overline{a} = \{(p_1 + p_2 + p_3), (n_4)\}$$

$$G_1: \quad x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 2x_5^2 + n_1 - p_1 = 110$$

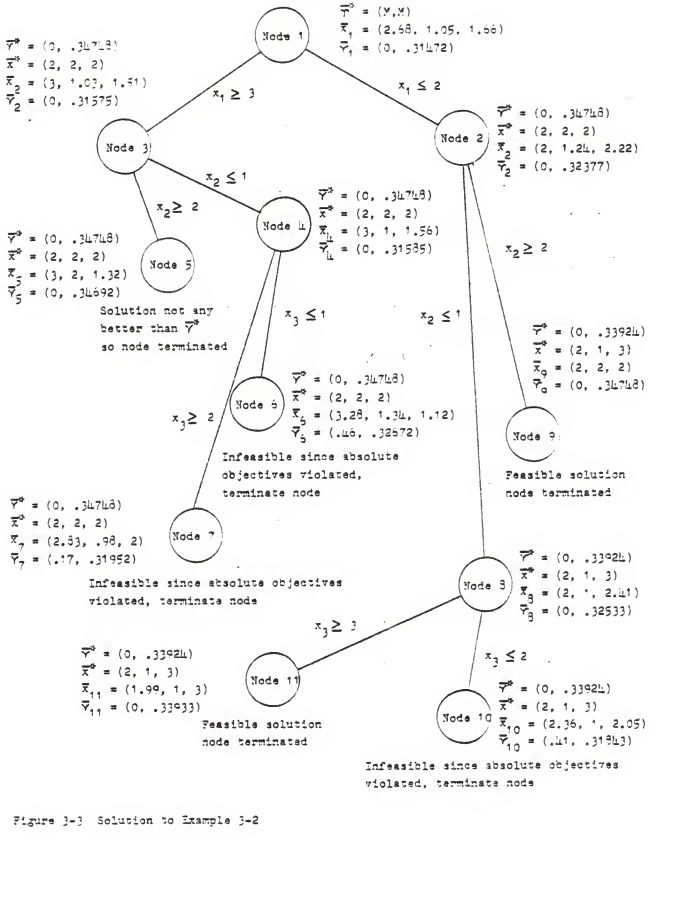

Figure 3-3 Solution to Example 3-2

Table 3-5.

Trial branchings from node 1 of Example 3-2.

Branch	$\bar{X}$	$\bar{Y}$	Remark
$x_1 \leq 2$	2.00, 1.24, 2.22	0, .32377	
$x_1 \geq 3$	3.00, 1.03, 1.52	0, .31575	lowest bound
$x_2 \leq 1$	2.22, 1.00, 2.20	0, .32090	
$x_2 \geq 2$	2.05, 2.00, 2.00	0, .34740	feasible
$x_3 \leq 1$	3.23, 1.50, 1.00	0, .33577	
$x_3 \geq 2$	2.34, 1.09, 2.00	0, .31774	

Table 3-6.

Trial branchings from node 3 of Example 3-2.

Branch	$\bar{X}$	$\bar{Y}$	Remark
$x_2 \leq 1$	3.00, 1.00, 1.56	0, .31585	lowest bound
$x_2 \geq 2$	3.00, 2.00, 1.32	0, .34692	
$x_3 \leq 1$	3.02, 1.55, 1.00	0, .33635	
$x_3 \geq 2$	2.58, 1.96, 2.00	.42, .34637	infeasible

Table 3-7.

Trial branchings from node 4 of Example 3-2.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_3 \leq 1$	3.28, 1.34, 1.12	.46, .32672	infeasible
$x_3 \geq 2$	2.83, .98, 2.00	.17, .31952	infeasible

Table 3-8

Trial branchings from node 2 of Example 3-2.

Branch	$\bar{X}$	$\bar{Y}$	Remark
$x_2 \leq 1$	2.00, 1.00, 2.41	0, .32533	lowest bound
$x_2 \geq 2$	2.00, 2.00, 2.00	0, .34748	feasible
$x_3 \leq 2$	2.00, 1.51, 2.00	0, .32712	
$x_3 \geq 3$	1.99, 1.03, 3.00	0, .33928	feasible

Table 3-9

Trial brancnings from node 8 of Example 3-2.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_3 \leq 2$	2.36, 1.00, 2.05	.41, .31843	infeasible
$x_3 \geq 3$	1.99, 1.00, 3.00	0 , .33933	feasible

$$G_2: \quad 7[x_1 + \exp(x_1/4)] + 7[x_2 + \exp(x_2/4)] + 5[x_3 + \exp(x_3/4)]$$

$$+ 9[x_4 + \exp(x_4/4)] + 4[x_5 + \exp(x_5/4)] + n_2 - p_2 = 175$$

$$G_3: \quad 7x_1\exp(x_1/4) + 8x_2\exp(x_2/4) + 8x_3\exp(x_3/4)$$

$$+ 6x_4\exp(x_4/4) + 9x_5\exp(x_5/4) + n_3 - p_3 = 200$$

$$G_4: \quad [1 - (1 - .80)^{x_1}][1 - (1 - .85)^{x_2}][1 - (1 - .09)^{x_3}]$$

$$[1 - (1 - .65)^{x_4}][1 - (1 - .75)^{x_5}] + n_4 - p_4 = 1$$

where x_1, x_2, x_3, x_4, and x_5 are positive integers and $\bar{n}$, $\bar{p} \geq \bar{0}$.

In this problem G_1, G_2, and G_3 are nonlinear resource constraints and G_4 is system reliability, which is the same problem as single objective decision problem when G_4 is taken as the objective and G_1, G_2, and G_3 as the constraints. The solution to Example 3-3 is given in Figure 3-4. Trial branchings from each node are presented in Tables 3-10 through 3-14. The solution is $\bar{x}^* = (3, 2, 2, 3, 3)$ and $\bar{a}^* = \bar{\gamma}^* = (0, .09553)$. Hence, the optimal system reliability is:

$$R_s = 1 - .09553 = .90447$$

Consumption of each resource is:

$$G_1 = 83, \; G_2 = 146.12, \; G_3 = 192.48$$

Example 3-4:

$$\min \bar{a} = \{(n_4), (p_2), (p_3), (p_1)\}$$

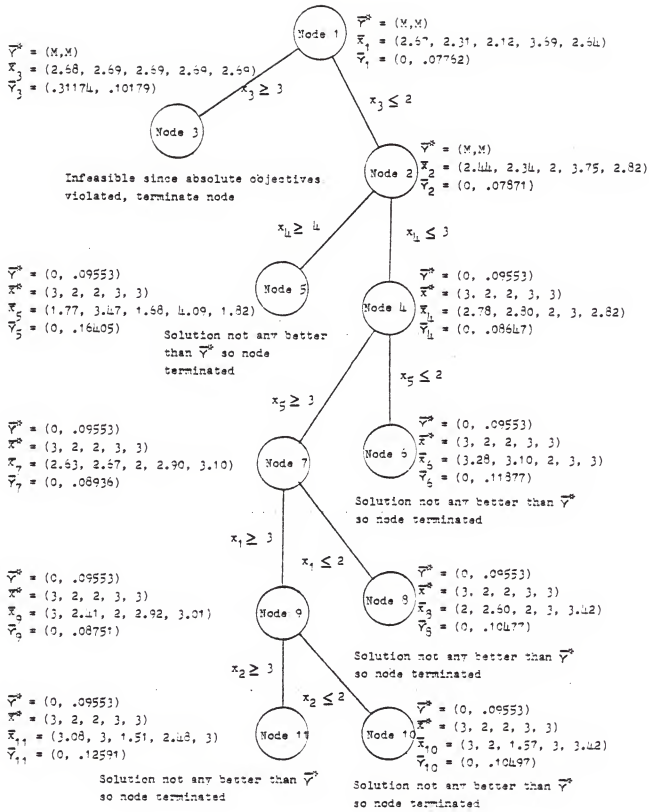

Figure 3-4 Solution to Example 3-3

Table 3-10

Trial branchings from node 1 of Example 3-3.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_1 \leq 2$	2.00, 2.69, 2.72, 3.28, 2.67	0, .10067	
$x_1 \geq 3$	3.00, 2.09, 3.47, 2.43, 2.20	0, .14544	
$x_2 \leq 2$	2.57, 2.00, 2.36, 3.65, 2.80	0, .08230	
$x_2 \geq 3$	2.72, 3.00, 2.35, 3.02, 2.39	0, .09550	
$x_3 \leq 2$	2.44, 2.34, 2.00, 3.75, 2.82	0, .07871	lowest bound
$x_3 \geq 3$	2.68, 2.69, 2.69, 2.69, 2.69	.31, .10179	infeasible
$x_4 \leq 3$	2.84, 2.66, 2.35, 3.00, 2.65	0, .08693	
$x_4 \geq 4$	2.85, 2.54, 2.78, 3.15, 2.22	.85, .09923	infeasible
$x_5 \leq 2$	2.71, 2.36, 2.34, 4.00, 2.00	0, .10276	
$x_5 \geq 3$	2.26, 2.41, 2.80, 2.85, 3.00	0, .10072	

Table 3-11

Trial branchings from node 2 of Example 3-3.

Branch	$\bar{x}$	$\bar{y}$	Remark
$x_1 \leq 2$	2.00, 2.80, 2.00, 3.36, 3.04	0, .09559	
$x_1 \geq 3$	3.00, 2.96, 1.55, 2.97, 2.77	0, .10117	
$x_2 \leq 2$	3.10, 2.00, 2.00, 3.09, 3.09	0, .08927	feasible
$x_2 \geq 3$	2.40, 3.00, 1.84, 3.00, 3.00	0, .09403	
$x_4 \leq 3$	2.78, 2.80, 2.00, 3.00, 2.82	0, .08647	lowest bound
$x_4 \geq 4$	1.77, 3.47, 1.68, 4.09, 1.82	0, .16405	
$x_5 \leq 2$	2.50, 2.50, 2.00, 2.50, 2.00	0, .16191	
$x_5 \geq 3$	2.63, 2.67, 2.00, 2.90, 3.10	0, .08936	

Table 3-12

Trial branchings from node 4 of Example 3-3.

Branch	$\bar{X}$	$\bar{Y}$	Remark
$x_1 \leq 2$	2.00, 3.09, 2.00, 3.00, 3.05	0, .10615	feasible
$x_1 \geq 3$	3.00, 2.96, 1.55, 2.97, 2.77	0, .10117	
$x_2 \leq 2$	3.12, 2.00, 2.00, 3.00, 3.12	0, .09195	feasible
$x_2 \geq 3$	2.40, 3.00, 1.84, 3.00, 3.00	0, .09403	
$x_5 \leq 2$	3.28, 3.10, 2.00, 3.00, 2.00	0, .11877	
$x_5 \geq 3$	2.63, 2.67, 2.00, 2.90, 3.10	0, .08936	lowest bound

Table 3-13

Trial branchings from node 7 of Example 3-3.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_1 \leq 2$	2.00, 2.60, 2.00, 2.00, 3.42	0, .10477	
$x_1 \geq 3$	3.00, 2.41, 2.00, 2.92, 3.01	0, .08751	lowest bound
$x_2 \leq 2$	2.99, 1.99, 1.51, 2.99, 3.46	0, .10892	
$x_2 \geq 3$	2.28, 3.00, 2.00, 2.94, 3.02	0, .09624	

Table 3-14

Trial branchings from node 9 of Example 3-3.

Branch	$\bar{X}$	$\bar{Y}$	Remark
$x_2 \leq 2$	3.00, 2.00, 1.57, 3.00, 3.42	0, .10497	lowest bound
$x_2 \geq 3$	3.08, 3.00, 1.51, 2.48, 3.00	0, .12591	

$$G_1: \quad x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 2x_5^2 + n_1 - p_1 = 110$$

$$G_2: \quad 7[x_1 + \exp(x_1/4)] + 7[x_2 + \exp(x_2/4)]$$
$$+ 5[x_3 + \exp(x_3/4)] + 9[x_4 + \exp(x_4/4)] + 4[x_5 + \exp(x_5/4)] + n_2 - p_2 = 175$$

$$G_3: \quad 7x_1\exp(x_1/4) + 8x_2\exp(x_2/4) + 8x_3\exp(x_3/4)$$
$$+ 6x_4\exp(x_4/4) + 9x_5\exp(x_5/4) + n_3 - p_3 = 200$$

$$G_4: \quad [1 - (1 - .80)^{x_1}][1 - (1 - .85)^{x_2}][1 - (1 - .90)^{x_3}]$$
$$[1 - (1 - .65)^{x_4}][1 - (1 - .75)^{x_5}] + n_4 - p_4 = .99$$

where x_1, x_2, x_3, x_4, and x_5 are positive integers and $\bar{n}$, $\bar{p} \geq \bar{0}$.

This problem is a slight modification of Example 3-3, i.e., the achievement of the minimum required system reliability .99 at priority level one, resource consumption of G_2 at priority level two, resource consumption of G_3 at priority level three, and resource consumption of G_1 at priority level four. The solution to Example 3-4 is given in Figure 3-5. Trial branchings from each node are presented in Tables 3-15 through 3-20. Note that at node 10 every variable has been branched, however, the solution still contains fractional valued variables. In this case, it is recommended that the fractional valued variable be tried for another branchings as shown in Table 3-20.

Note that, in Table 3-20, $3.95 \leq x_2 \leq 4.05$ and $2.95 \leq x_3 \leq 3.05$ are used for branching instead of $x_2 = 4$ and $x_3 = 3$ in order to give more flexibility in finding the optimal solution using Hwang and Paidy's

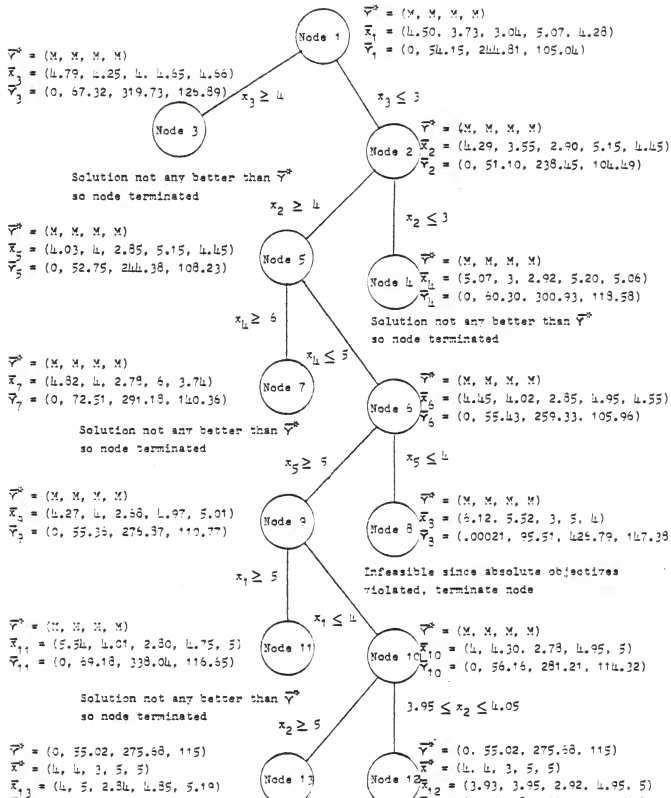

Figure 3-5 Solution to Example 3-4

Table 3-15

Trial branchings from node 1 of Example 3-4.

Branch	$\bar{X}$	$\bar{Y}$	Remark
$x_1 \leq 4$	4.00, 4.68, 4.01, 4.77, 4.69	0, 65.29, 314.68, 133.05	
$x_1 \geq 5$	5.00, 4.01, 3.76, 4.95, 4.08	0, 66.17, 290.83, 120.88	
$x_2 \leq 3$	5.22, 3.00, 3.69, 5.00, 5.00	0, 64.62, 321.68, 126.13	
$x_2 \geq 4$	4.82, 4.00, 3.30, 4.98, 4.14	0, 60.88, 268.19, 111.28	
$x_3 \leq 3$	4.29, 3.55, 2.90, 5.15, 4.45	0, 51.10, 238.45, 104.49	lowest bound
$x_3 \geq 4$	4.79, 4.25, 4.00, 4.65, 4.66	0, 67.32, 319.73, 126.89	
$x_4 \leq 5$	4.39, 3.93, 3.31, 4.95, 4.30	0, 55.42, 252.57, 107.99	
$x_4 \geq 6$	4.49, 3.58, 2.92, 6.00, 3.80	0, 65.03, 265.09, 134.40	
$x_5 \leq 4$	4.75, 3.58, 3.06, 5.35, 4.00	0, 58.66, 251.52, 112.62	
$x_5 \geq 5$	4.62, 3.67, 3.05, 4.78, 5.00	0, 55.41, 280.15, 107.55	

Table 3-16

Trial branchings from node 2 of Example 3-4.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_1 \leq 4$	4.00, 4.55, 3.00, 4.88, 4.81	0, 58.15, 283.44, 115.74	
$x_1 \geq 5$	5.17, 4.54, 2.60, 4.89, 4.88	0, 70.44, 332.97, 121.36	
$x_2 \leq 3$	5.07, 3.00, 2.92, 5.20, 5.06	0, 60.30, 300.93, 118.58	
$x_2 \geq 4$	4.03, 4.00, 2.85, 5.15, 4.45	0, 52.75, 244.38, 108.23	lowest bound
$x_4 \leq 5$	4.49, 3.77, 2.95, 4.95, 4.54	0, 53.61, 252.58, 103.82	
$x_4 \geq 6$	4.80, 3.80, 2.90, 6.00, 3.72	0, 70.68, 284.11, 138.83	
$x_5 \leq 4$	4.20, 3.79, 2.90, 5.55, 4.00	0, 56.63, 241.15, 116.73	
$x_5 \geq 5$	5.02, 4.40, 2.58, 4.89, 5.00	0, 67.55, 325.22, 119.66	

Table 3-17

Trial branchings from node 5 of Example 3-4.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_1 \leq 4$	4.00, 4.55, 3.00, 4.88, 4.81	0, 58.15, 283.44, 115.74	
$x_1 \geq 5$	5.17, 4.54, 2.60, 4.89, 4.88	0, 70.44, 332.97, 121.36	
$x_4 \leq 5$	4.45, 4.02, 2.85, 4.95, 4.55	0, 55.43, 259.33, 105.96	lowest bound
$x_4 \geq 6$	4.82, 4.00, 2.78, 6.00, 3.74	0, 72.51, 291.18 140.36	
$x_5 \leq 4$	4.23, 4.00, 2.97, 5.40, 4.00	0, 57.38, 245.15, 115.00	
$x_5 \geq 5$	5.02, 4.40, 2.58, 4.89, 5.00	0, 67.55, 325.22, 119.66	

Table 3-18

Trial branchings from node 6 of Example 3-4.

Branch	$\overline{x}$	$\overline{Y}$	Remark
$x_1 \leq 4$	4.00, 4.55, 3.00, 4.88, 4.81	0, 58.15, 283.44, 115.74	
$x_1 \geq 5$	5.10, 4.50, 2.99, 4.75, 4.60	0, 67.63, 313.38, 115.93	
$x_5 \leq 4$	6.12, 5.52, 3.00, 5.00, 4.00	.00021, 95.51, 426.79, 147.38	
$x_5 \geq 5$	4.27, 4.00, 2.68, 4.97, 5.01	0, 55.36, 276.87, 110.77	lowest bound

Table 3-19

Trial branchings from node 9 of Example 3-4.

Branch	$\bar{x}$	$\bar{Y}$	Remark
$x_1 \leq 4$	4.00, 4.30, 2.78, 4.95, 5.00	0, 56.16, 281.21, 114.32	lowest bound
$x_1 \geq 5$	5.54, 4.01, 2.80, 4.75, 5.00	0, 69.18, 338.04, 116.65	

Table 3-20.

Trial branchings from node 10 of Example 3-4.

Branch	$\bar{X}$	$\bar{Y}$	Remark
$3.95 \leq x_2 \leq 4.05$	3.93, 3.95, 2.92, 4.95, 5.00	0 , 52.28 266.93, 110.47	lowest bound
$x_2 \geq 5$	4.00, 5.00, 2.84, 4.85, 5.19	0 , 65.10, 330.59, 128.09	
$2.95 \leq x_3 \leq 3.05$	3.93, 4.00, 2.95, 4.93, 5.00	0 , 52.72, 268.60, 110.89	feasible
$x_3 \leq 2$	4.00, 6.33, 2.00, 5.00, 7.97	.00680, 106.24, 779.80, 225.15	

nonlinear goal programming technique. The solution is $\bar{x}^* = (4, 4, 3, 5, 5)$ and $\bar{a}^* = \bar{\gamma}^* = (0, 55.02, 275.68, 115)$, which gives:

$$R_s = .99069,\ G_1 = 225,\ G_2 = 230.02,\ G_3 = 475.68$$

REFERENCES

1. Hwang, C. L. and S. Paidy, "Regional water quality management by non-linear goal programming", submitted to Water Resource Bulletin for publication, 1978.

2. Ignizio, James P., Goal Programming and Extensions, Lexington, Mass.: Lexington Books, 1976.

3. Dakin, R. J., "A tree search algorithm for mixed integer programming problems", The Computer Journal, 1966, pp. 250-255.

4. Taha, Hamdy A., Operations Research, New York: The Macmillan Co., 1971.

CHAPTER 4

CONCLUSION

As is discussed in Chapter 2, the techniques to solve a single objective nonlinear integer problem is first to reformulate the problem into a zero-one linear integer problem and to solve it by zero-one linear integer programming algorithm. In Chapter 3 the single objective nonlinear integer problem is solved by nonlinear integer goal programming taking the constraints at priority level one and the objective at priority level two, i.e., as a nonlinear problem itself. Of course, the nonlinear integer goal programming method can solve a multiple objective problem like Example 3-4. But the difficulty in handling the nonlinear problem is that there is, in general, no way to guarantee finding the global optimum for a given problem. This mean that the investigater must usually be satisfied with a local optimum or set of local optima. In fact, even if the method employed happened to give a global optimum, it is usually impossible to distinguish this circumstance. Even in applying the Hwang and Paidy's nonlinear goal programming method to solve the goal programming formulation at each node has some difficulty, i.e., the solution is quite dependable to the initial starting point and stepsizes. Hence, several different starting points and stepsizes should be tried in order to get the real optimum solution. .05's are recommended as stepsizes. However, the algorithm presented in Chapter 3 seems to give pretty assuring solution to the multiple objective problems as is demonstrated through examples in Chapter 3. One thing that should

be mentioned is that it needs $n(n+1)$ times computer runs of nonlinear goal programming to solve a problem with n variables. For example, to solve a problem with five variables takes $5(5+1) = 30$ runs and a problem with ten variables takes $10(10+1) = 110$ runs, which takes up a lot of computing time. But this problem may be eliminated by taking the whole algorithm into one computer programming, which is left for future study.

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to his major professor, Dr. C. L. Hwang, for his helpful advice and personal interest taken in the preparation of this thesis.

The author also wishes to thank Dr. F. A. Tillman and Dr. D. S. Chung for serving his supervisory committee.

This study was partly supported by Office of Naval Research, Contract No. N00014-76C-0842.

INTEGER PROGRAMMING AND NONLINEAR INTEGER GOAL PROGRAMMING APPLIED TO SYSTEM RELIABILITY PROBLEMS

by

HOON BYUNG LEE

B.S., Seoul National University, Seoul, Korea, 1973

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1978

ABSTRACT

The purpose of this thesis is to present the techniques to solve integer constrained nonlinear problems encountered in the system reliability optimization.

There have been a number of applications of integer programming techniques to single objective reliability problems, which are first to reformulate the nonlinear integer problems into zero-one linear integer problems and to solve them by zero-one linear integer algorithms. Those techniques are classified into four types. Each integer programming technique is used to solve a numerical example in detail.

Few studies have been done on the techniques to solve multiple objective nonlinear integer problems. In this thesis, an algorithm for nonlinear integer goal programming is formulated utilizing a Branch-and-bound method and a nonlinear goal programming method. The application of this algorithm is demonstrated by solving reliability problems with single objective and multiple objectives.

One interesting feature in using the present algorithm is that the problem is solved by traditional nonlinear search techniques, such as Hooke and Jeeves pattern search, that are originally intended for solving the so called "unconstrained" problem.