237

OPTIMIZATION TECHNIQUES FOR SYSTEMS RELIABILITY WITH REDUNDANCY

by

WAY KUO

B.S. (Nuclear Engineering), National Tsing-Hua University, Taiwan, 1972
M.S. (Nuclear Engineering), University of Cincinnati, Ohio, 1975

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

ACKNOWLEDGEMENTS

The author wishes to express his deep gratitude to Dr. C. L. Hwang, major professor, for his sincere guidance, constructive criticism, and comments through this work; to his supervisory committee members, Dr. F. A. Tillman, and Dr. D. Eckhoff for their encouragement and creative advice.

The author thanks Mrs. Marie Davis for typing this thesis.

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

TABLE OF CONTENTS

			page
ACKNOWLEDGEMENTS			
CHAPTER 1		INTRODUCTION	1
		REFERENCES	8
CHAPTER 2		OPTIMIZATION TECHNIQUES FOR SYSTEMS RELIABILITY WITH REDUNDANCY - A REVIEW	9
	2.1	Introduction	9
	2.2	Systems Models	10
	2.3	Statements of the Various Optimization Problems	15
	2.4	Optimization Techniques Used to Determine the Optimal Systems Reliability	20
	2.5	Concluding Remarks	21
		REFERENCES	23
CHAPTER 3		OPTIMIZATION OF SYSTEM RELIABILITY	34
	3.1	HEURISTIC METHODS IN OPTIMAL SYSTEM RELIABILITY	38
		1. Introduction	38
		2. Sharma-Venkateswaran's Approach	44
		3. Aggarwal's Approach	52
		4. Misra's Approach	59
		5. Ushakov's Approach	64
		6. Nakagawa - Nakashima's Approach	67
		REFERENCES	78
	3.2	DYNAMIC PROGRAMMING APPLIED TO OPTIMAL SYSTEM RELIABILITY	79

page

1. Introduction 79

2. Basic Dynamic Programming Approach 87

3. Dynamic Programming Approach Using Lagrange Multipliers 110

4. Dynamic Programming Approach Using the Concepts of Dominating Sequence 127

REFERENCES 138

3.3 THE DISCRETE MAXIMUM PRINCIPLE APPLIED TO OPTIMUM SYSTEM RELIABILITY 140

1. Introduction 140

2. Statement of the Problem and the Computational Procedure 140

3. Example 145

4. Numerical Results 151

5. Conclusion 153

REFERENCES 154

3.4 SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE (SUMT) APPLIED TO OPTIMAL SYSTEM RELIABILITY 155

1. Introduction 155

2. Formulation of the Problem 156

3. Computational Procedure of SUMT 160

4. Numerical Examples 163

REFERENCES 169

page

3.5 GENERALIZED REDUCED GRADIENT METHOD (GRG) APPLIED TO OPTIMUM RELIABILITY 181

1. Introduction 181

2. Numerical Examples 188

REFERENCES

3.6 METHOD OF LAGRANGE MULTIPLIERS AND THE KUHN-TUCKER CONDITIONS IN OPTIMAL SYSTEM RELIABILITY 190

1. Introduction 190

2. Lagrange Multiplier Method for Single Constraint Problem 192

3. Kuhn-Tucker Conditions 196

4. Method of Lagrange Multipliers and the Kuhn-Tucker Conditions for Two Linear Constraints Problems 198

5. Conclusion 201

REFERENCES 204

3.7 THE GENERALIZED LAGRANGIAN FUNCTION METHOD APPLIED TO OPTIMAL SYSTEMS RELIABILITY 204

1. Introduction 204

2. The Generalized Lagrangian Function and the Computational Procedures 205

3. Numerical Examples 208

REFERENCES 217

3.8 GEOMETRIC PROGRAMMING APPLIED TO OPTIMAL SYSTEM RELIABILITY 219

1. Introduction 219

2. Formulation of the Problem 219

			page
		3. A Numerical Example	225
		REFERENCES	230
	3.9	INTEGER PROGRAMMING APPLIED TO OPTIMAL SYSTEM RELIABILITY	231
		1. Introduction	231
		2. Partial Enumeration Method	238
		3. The Gomory's Cutting Plane Method	242
		4. The Branch and Bound Method	248
		5. The Gomory's Cutting Plane Method	254
		6. The Lawler-Bell Partial Enumeration Method	257
		7. The Geoffrion's Implicit Enumeration Method	259
		REFERENCES	265
	3.10	OTHER METHODS APPLIED TO THE SYSTEM RELIABILITY OPTIMIZATION PROBLEMS	269
		1. Introduction	269
		2. A Classical Approach	270
		3. Parametric Method	275
		4. Linear Programming	281
		5. Separable Programming	284
		REFERENCES	292
CHAPTER 4		DETERMINATION OF COMPONENT RELIABILITY AND REDUNDANCY FOR OPTIMUM SYSTEMS RELIABILITY	295
	4.1	Introduction	295
	4.2	Statement of the Problem	297
	4.3	An Optimization Procedure	300

			page
	4.4	Numerical Examples	303
	4.5	Concluding Remarks	303
		REFERENCES	308
APPENDIX		OUTLINE OF SEVERAL OPTIMIZATION TECHNIQUES	309
	A1.	DYNAMIC PROGRAMMING	310
	A2.	THE DISCRETE MAXIMUM PRINCIPLE	314
	A3.	THE GENERALIZED REDUCED GRADIENT METHOD (GRG)	316
	A4	GEOMETRIC PROGRAMMING	325

CHAPTER 1 INTRODUCTION

The system effectiveness is often used to describe the overall capability of a system to accomplish its mission. If the system is effective, it carries out its intended function well. If it is not, attention must be directed to those system attributes which are deficient. Of the major attributes determining system effectiveness, the one that has received the most thorough and systematic study is reliability.

Reliability is the probability of successful operation. One definition reads "Reliability is the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered" (Radio-Electronics-Television Manufacturers Association, 1955). Therefore, the probability that a system successfully performs as designed is called the "system reliability." Such probability is also referred to as the "probability of survival." In most cases the probability with which a device will perform its function is not known at first. Also the true reliability is never exactly known, which means the exact numerical value of the probability of adequate performance is not known. But numerical estimates quite close to this value can be obtained by the use of statistical methods and probability calculations.

Reliability has been a measure of the capacity of a system to operate without failure when it is put into service. For convenience, reliability has also been described as the ability of equipment to preserve its output characteristics (parameters) within established limits under given operating conditions. From this concept, it also follows that an unreliable system is a system made inoperative by mechanical or electrical damage or a system's output characteristics drift outside admissible limits [3]. Such characteristics can be accuracy, the nature of transient responses, the type of frequency characteristics, etc.

System reliability is a measure of how well a system performs or meets its design objective, and it is usually expressed in terms of the reliabilities of the subsystems or components. The following terminologies are defined. A "part" or "element" is the least subdivision of a system, or an item that cannot ordinarily be disassembled without being destroyed. A "circuit" is a collection of parts that has a specific function. A "component" then is a collection of parts and/or circuits, which represents a self-contained element of a complete operating system and performs a function necessary to the operation of that system. "Unit", "component", and "subsystem" are synonymous. A "system" can then be characterized as a group of subsystem especially integrated to perform a specific operational function or functions.

The "reliability" of a system is the probability of a successful operation of the system for a specified period of time. In describing the reliability of a given system it is necessary to specify (1) the equipment failure process, (2) the system configuration which describes how the equipment is connected and the rules of operation, and (3) the state in which the system is to be defined as failed. The equipment failure process describes the probability law governing those failures. The system configuration, on the other hand, defines the manner in which the system reliability function will behave. The third consideration in developing the reliability function for a non-maintained system is to define the conditions of system failures.

There exist several methods to improve the system reliability. Some of these methods approach the problem by using large safety factors, reducing the complexity of the system, increasing the reliability of constituent components through a product improvement program, using structural redundancy, and practicing a planned maintenance and repair schedule. A good deal of effort has been made in the field of optimal redundancy allocation.

A system in many cases is not confined to a single component. What we really want to evaluate is the reliability of those systems which are simple as well as those which are extremely complex. To develop functions expressing the system reliability, both conventional statistic (probabilistic) theory and Markovian process have been used.

Reliability engineering appeared on the scene in the late 1940's and early 1950's and first applied to the field of communication and transportation. Much of the early reliability work was confined to making trade-offs between certain performance and reliability aspects of systems. However, in the ever-increasing complex systems of today reliability has become increasingly important.

In this thesis a thorough discussion of reliability optimization problems is presented. The contents include a critical review of optimization techniques for system reliability with redundancy, and the determination of component reliability and redundancy for optimum system reliability.

As complexity increases, so must reliability.

The objectives of this thesis are:

(1.) to present a critical review and classification of all system reliability optimization problems which have been analyzed with various optimization techniques;

(2.) to study the generalized reduced gradient method (GRG) and a generalized Lagrangian function method applied to system reliability optimization problems. Both of these methods have not previously been applied to these problems;

(3.) to generalize the system reliability optimization problems to include reliability allocation and redundancy allocation simultaneously;

(4.) to propose new methods for determining integer solution, particularly, heuristic methods.

A state-of-the-art review of the literature related to optimal system reliability with redundancy is presented in Chapter 2. The literature is classified as follows.

Optimal system reliability models with redundancy

Series

Parallel

Series-parallel

Parallel-series

Standby

Complex (nonseries, nonparallel)

Optimization techniques for obtaining optimal system configurations are:

- Integer programming
- Dynamic programming
- Maximum principle
- Linear programming
- Geometric programming
- Sequential unconstrained minimization technique (SUMT)
- Modified sequential simplex pattern search
- Lagrange multipliers and Kuhn-Tucker conditions
- Generalized Lagrangian function
- Generalized reduced gradient (GRG)
- Heuristic approaches
- Parametric approaches
- Pseudo-Boolean programming
- Miscellaneous

One goal of the reliability engineer is to find the best way to increase the system reliability. Six important methods for doing this are [6]: (1) keep the system as simple as is compatible with the performance requirements. Nonessential components and unnecessarily complex configurations only increase the probability of system failure. One aspect of complexity which produces unreliability is subsystem interaction which may be environmental [2]. (2) increase the reliability of the components in the system. (3) use parallel redundancy to the less reliable components or stages. (4) use standby redundancy which is switched to the active components or stages when they fail. (5) use repair maintenance where failed components are replaced but not automatically switched in as in (4). (6) use preventive maintenance where components are replaced by new ones whenever they fail or at some specific time, whichever comes first. [1].

A good deal of effort has been made in the field of optimal redundancy allocations. However, to increase the system reliability, the system will spend more "cost" in weight, volume size, money expenditure, etc. to meet the requirement of higher system reliability. Hence, in Chapter 3 optimization techniques are introduced which include (a) to maximize the system reliability of various system configurations subject to the "cost" constraints, or (b) to minimize any specific "cost" while satisfying the minimum requirement of the system reliability.

In this chapter, the literature published on optimal system reliability is classified and critically reviewed. Various problems are also classified and solved by a heuristic approach, dynamic programming, and integer programming. These optimization techniques always give solution of integer numbers which meet the implied integer requirement of redundancy allocation problems.

Sequential Unconstrained Minimization Technique (SUMT) has been widely used in solving many optimization problems. The problem with a tangent form cost function is successfully solved by SUMT. Generalized Reduced Gradient method (GRG) and generalized Lagrangian functions method have been developed but have never been used in system reliability optimization problems. The GRG method is an elaborate extension of hill-climbing gradient techniques, and has been coded in FORTRAN in a program named GREG. A new type of the generalized or augmented Lagrangian function proposed by Sayama et al. for finding the solution of a non-linear programming problem with inequality constraints is also explored in this chapter. Since neither method gives integer solutions, rounding off procedures are applied whenever the redundancy allocation problem is solved.

The maximum principle, the method of Lagrange multipliers and the Kuhn-Tucker conditions, geometric programming, and several miscellaneous optimization techniques (eg. linear programming and separable programming) are also effectively introduced in this chapter which provides a comprehensive discussion of optimization techniques having been used in studying system reliability optimization problems.

It is noted that most of the optimization techniques employed in this thesis, are limited to solving small system reliability optimization problems.

In Chapter 4 a problem is presented which simultaneously includes the determination of the optimal level of component reliability and the number of redundancies in each of the stages. The problem is one in which the component failure rates are variables and the optimal trade-off between adding components in redundancy or the improvement of an individual component's reliability is considered. This becomes a mixed integer programming problem in which the system reliability is to be maximized as a function of component reliability level and the number of components used at each stage. The Hooke and Jeeves pattern search technique in combination with the heuristic approach by Aggarwal, et al. is utilized to solve this problem.

Chapter 4 extends the usual reliability optimization problem to determine both the optimization of component reliability and the number of redundancies in each of the stages.

REFERENCES

1. Goldman, A. S., and T. B. Slattery, Maintainability - A Major Element of System Effectiveness, N. Y. Wiley (1964).

2. Lloyd, D. K., and M. Lipow, Reliability: Management, Methods, and Mathematics, Englewood Cliffs, N.J.: Prentice-Hall (1962).

3. Polovko, A. M., Fundamentals of Reliability Theory, N.Y.: Academic (1968).

4. Sandler, G. H., System Reliability Engineering, Englewood Cliffs, N.J.: Prentice-Hall (1963).

5. Shooman, M. L., Probabilistic Reliability: An Engineering Approach, N.Y.: McGraw-Hill (1968).

6. Smith, C. O., Introduction to Reliability in Design, N. Y.: McGraw-Hill (1976).

CHAPTER 2 OPTIMIZATION TECHNIQUES FOR SYSTEMS RELIABILITY WITH REDUNDANCY - A REVIEW

I. INTRODUCTION

The reliable performance of a system for a mission under various conditions is of utmost importance in many industrial, military, and everyday life situations. Although the qualitative concepts of reliability are not new, its quantitative aspects have been developed over the past two decades. Such development has resulted from the increasing needs for highly reliable systems and components with more safety and less cost.

There exist several methods to improve the system reliability. Some of these methods approach the problem by using large safety factors, reducing the complexity of the system, increasing the reliability of constituent components through a product improvement program, using structural redundancy, and practicing a planned maintenance and repair schedule. A good deal of effort has been centered in the field of optimal redundancy allocation.

A state-of-art review of the literature related to optimal systems reliability with redundancy is presented in this paper. The first part of the reference list is concerned with basic reliability [1-17] and optimization techniques [18-66]. The reference for the various optimization techniques are: optimization techniques in general [18-23], integer programming [24-29], the maximum principle [30-33], the generalized reduced gradient method (GRG) [34-42], modified sequential simplex pattern search [43-46], the sequential unconstrained minimization technique (SUMT) [47-52], the method of Lagrange multipliers and the Kuhn-Tucker conditions [53-54], the generalized Lagrangian functions method [55-58], dynamic programming [59-63], and geometric programming [64-66].

The second part of the reference list concentrates mainly on articles relevant to the optimization of systems reliability with redundancy [67-143].

which are classified into two categories: the system configurations and the optimization techniques employed, See Tables 1 and 2. In Table 1, the literature for the different system configurations is separated into the following model sub-categories: series, parallel, series-parallel, parallel-series, standby, and non-series-parallel models. In Table 2, the same literature is reclassified to indicate the variety of optimization techniques utilized.

Although the authors have tried to give a reasonably complete survey, those papers, not included, were either inadvertently overlooked or considered not to bear directly on the topics of this survey. The authors apologize to both the readers and the researchers if we have omitted any relevant papers.

2. SYSTEMS MODELS

In this review, we assume that the reader is familiar with the material treated in these models. For a discussion of the definitions and formulations of the basic concepts, the authors suggest reviewing the books on reliability as stated in the references [1-17]. We will briefly review each of the models considered in this survey.

The first model considered is an N-stage series system and is shown in Fig. 1. In this system, the functional operation depends upon the proper operation of all system components. The second model is an M-stage parallel system which is shown in Fig. 2. There are M paths connecting the input to the output, and all components must fail for the system to fail.

Figure 3 shows a mixed series-parallel system in which N components are connected in a series arrangement where M such series connections are connected in parallel to form the system. Figure 4 shows a mixed parallel-series system. In this system, N stages are connected in series where components are connected in parallel at each stage.

Table 1. The reference classification for the optimization of systems reliability with redundancy with regard to various system configuration.

System Configuration	References
Series	70, 72, 76, 77, 78, 79, 83, 87, 88, 89, 108, 122, 125, 129, 130, 136, 139, 141, 142
Parallel	70, 71, 79, 88, 89, 95, 104, 122, 129, 136, 139, 141, 142
Series-parallel	79, 88, 89, 94, 95, 110, 122, 124, 129, 130, 141, 142
Parallel-series	62, 69, 72, 75, 77, 78, 79, 80, 81, 82, 83, 84, 87, 88, 89, 92, 93, 94, 96, 98, 100, 101, 104, 105, 106, 107, 108, 109, 110, 111, 112, 114, 115, 116, 117, 118, 119, 120, 122, 124, 126, 127, 128, 129, 130, 132, 134, 135, 136, 137, 138, 139, 141, 142, 143
Standby	77, 89, 90, 93, 108, 120, 121, 122, 127, 129, 130, 135, 141, 142
Non-series-parallel (including bridge network)	68, 70, 71, 73, 79, 80, 91, 97, 136, 140

Table 2. The reference classification for the optimization techniques employed for systems reliability with redundancy.

Optimization technique	References
Integer programming	86, 87, 90, 92, 98, 103, 105, 106, 107, 112, 116, 120, 135, 137, 139
Dynamic programming	75, 80, 84, 94, 96, 99, 101, 104, 108, 127, 128, 133, 143
The maximum principle	82, 110, 138
Linear programming	98, 131
Geometric programming	83, 114
Sequential unconstrained minimization technique	91, 134, 136, 140
Modified sequential simplex pattern search	71, 118
The method of Lagrange multipliers and the Kuhn-Tucker conditions	77, 78, 81, 108, 110, 111, 135
The generalized Lagrangian function method	100
Generalized reduced gradient method	100
Heuristic approach	68, 69, 100, 132
Parametric approach	71, 72, 73, 87
Pseudo-Boolean programming	93
Others (miscellaneous)	76, 77, 78, 88, 89, 95, 100, 109, 115, 117, 119, 121, 122, 125, 129, 130, 141, 142

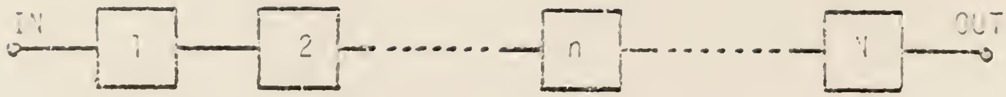


FIG. 1: AN N-STAGE SERIES SYSTEM

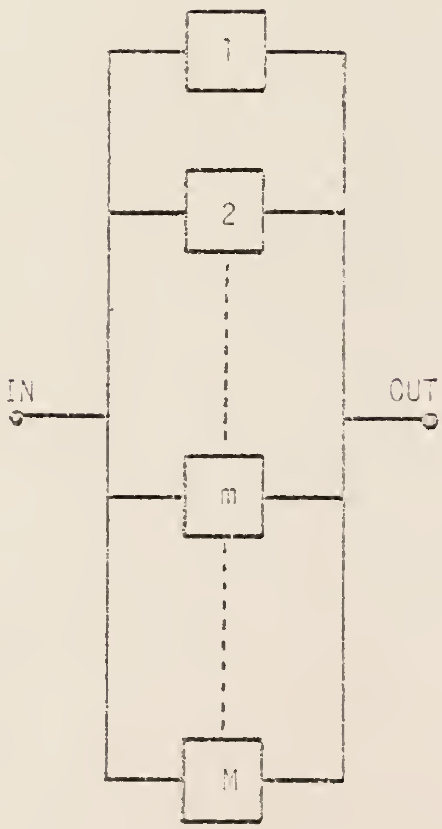


FIG. 2: AN M-STAGE PARALLEL SYSTEM

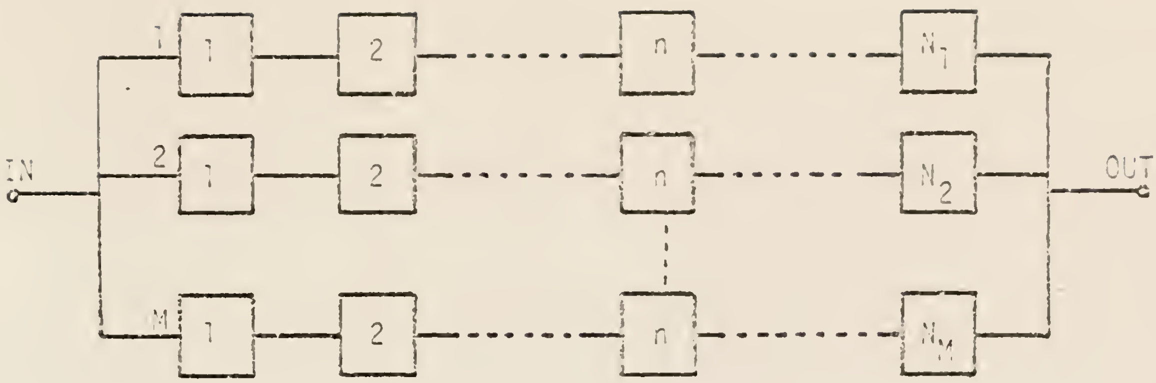


FIG. 3: A MIXED SERIES-PARALLEL SYSTEM

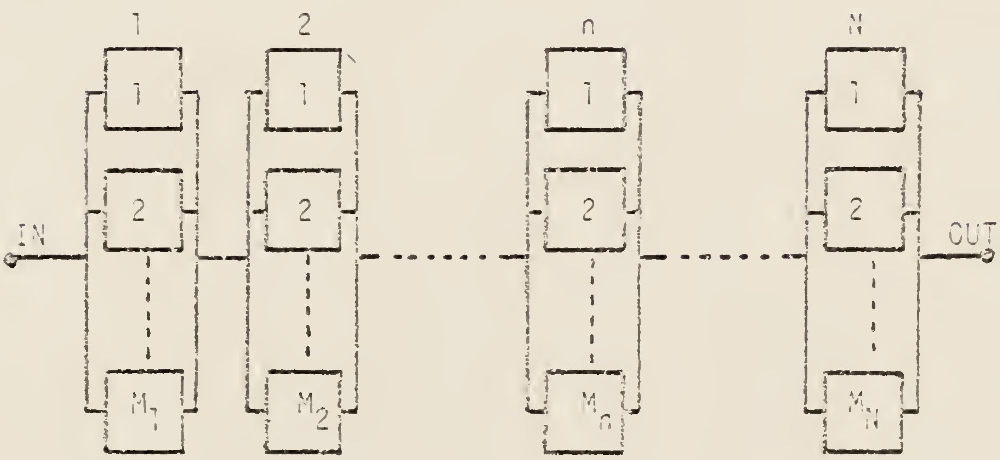


FIG. 4: A MIXED PARALLEL-SERIES SYSTEM

An element standby system is shown in Fig. 5, which has the same form as a mixed parallel-series system. However, in this system the parallel components are not all active at the same time. Figure 6 shows a system standby system which has the same form as a mixed series-parallel system. However, when a system standby system is used, the parallel M series subsystems are not all active at the same time.

Figure 7 shows a typical non-series-parallel reliability system. The reliability of this system can be evaluated by using conditional probabilities or other approaches. Figure 8 shows a complex bridge network system which is one of the complex reliability systems in the form of the bridge network.

Table 1 presents the literature on the optimization of systems reliability for the above systems models.

3. STATEMENT OF THE VARIOUS OPTIMIZATION PROBLEMS

The structure of the optimization problems, which are relevant to our survey, are stated below and the literature is identified in Table 3.

For an N-stage series model (see Fig. 1), the problem is one of allocating the reliability to each of the components so that the reliability is maximized and can be stated as

Problem 1

$$\text{Maximize } R_s = \prod_{j=1}^{N} R_j$$

Subject to

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

where R_s is the systems reliability, R_j is the component reliability of the jth stage, $g_{ij}(R_j)$ is the resource i consumed at stage j, and b_i is the total amount of resource i available. The function, $g_{ij}(R_j)$, can either be linear or nonlinear with respect to the component reliability, R_j.

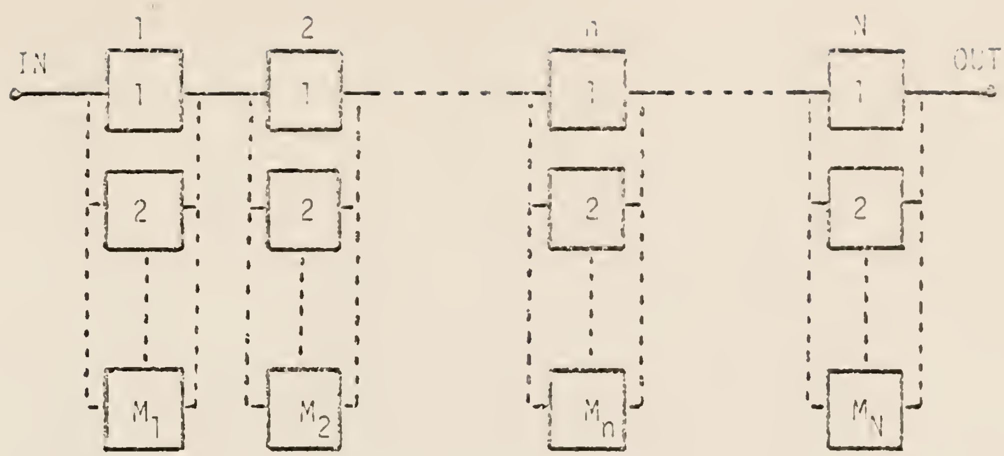


FIG. 5: AN ELEMENT STANDBY SYSTEM

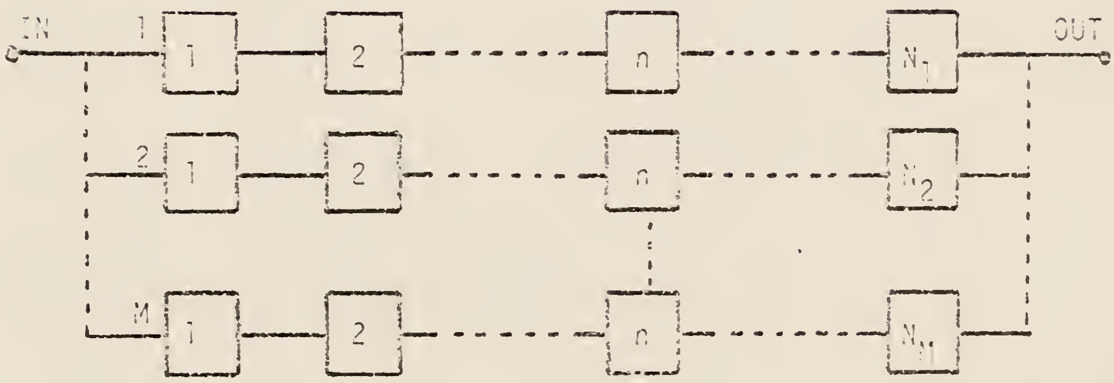


FIG. 6: A STANDBY SYSTEM

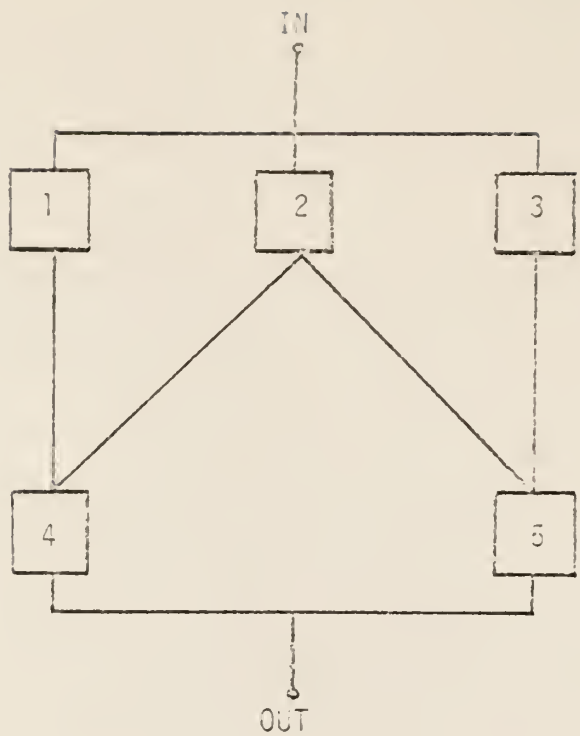


FIG. 7: A TYPICAL NON-SERIES-PARALLEL RELIABILITY SYSTEM

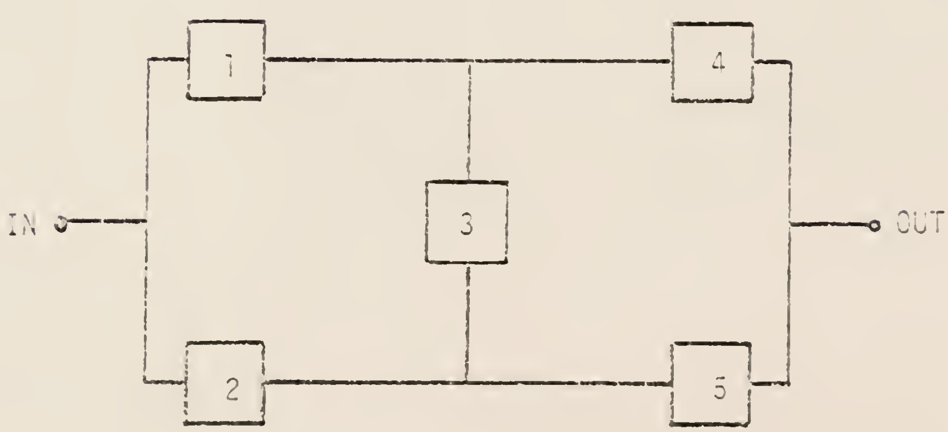


FIG. 8: A COMPLEX BRIDGE NETWORK SYSTEM

Table 3. The reference classifications with regard to the structure of the optimization problems

Formualtion of problems	References
Problem 1: Optimum reliability allocation for an N-stage series system	9, 76, 99, 100, 110, 113, 125
Problem 2: Optimum redundancy allocation, maximization of systems reliability subject to cost constraints	69, 72, 75, 77, 80, 81, 83, 84, 88, 89, 90, 92, 93, 94, 96, 98, 100, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 126, 129, 130, 132, 134, 135, 136, 137, 138, 139, 141, 142, 143
Problem 3: "Cost" minimization problems subject to the minimum requirement of the system reliability	72, 78, 90, 91, 93, 100, 112, 113, 122, 128, 131, 135, 137, 139
Problem 4: System reliability maximization for a non-series-parallel	
Reliability allocation	80, 91, 100, 140
Redundancy allocation	68, 71, 136
Others	
Maximization of the system profit	82, 128
Maximization of the ratio of the system reliability to the power demand of the system	104

A further delineation of this problem can be stated as finding the optimum number of redundancies (see Figs. 2-c) which maximize the system reliability subject to "cost" constraints, or the minimization of system costs subject to the condition that the system reliability is equal to or greater than a desired level. These problems are stated as:

Problem 2

$$\text{Maximize} \qquad R_s = \prod_{j=1}^{N} R_j(X_j)$$

subject to

$$\sum_{j=1}^{N} g_{ij}(X_j) \leq b_i, \; i = 1, 2, \ldots, m$$

where R_j is the reliability of the jth stage (subsystem), which is the function of the number of components, X_j.

Problem 3

$$\text{Minimize } C_s = \sum_{j=1}^{N} C_j(X_j)$$

subject to

$$R_s = \prod_{j=1}^{N} R_j(X_j) \geq R_r$$

where C_s is the total cost of the system and C_j is the cost of the jth stage which is a function of the number of components in each stage, X_j. The systems reliability, R_s, has to be greater than or equal to the required systems reliability, R_r.

The systems reliability for the complex systems (see Figs. 7 and 8) are obtained by using Bayes' theorem involving conditional probabilities or other network approaches. The optimization problem is stated as

Problem 4

Maximize $R_s = f(R_1, R_2, \ldots, R_N)$

subject to

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

where the systems reliability is a function of the component reliability, R_j.

4. OPTIMIZATION TECHNIQUES USED TO DETERMINE THE OPTIMAL SYSTEMS RELIABILITY

Most of the problems stated are nonlinear integer programming problems. These problems are more difficult to solve than the general nonlinear programming problems because the solutions are required to be integers. Many algorithms have been proposed but only a few have been demonstrated to be effective when applied to large-scale nonlinear programming problems. None have proven to be superior over the others so that it could be classified as the algorithm for solving general nonlinear programming problems [22].

The literature on the optimization techniques which are relevant to this survey is classified and presented in Table 2. All the optimization techniques employed in the 77 papers [67-143] have limited success in solving all of the problems.

Although integer programming [24-29] yields integer solutions, the transformation of nonlinear objective functions and constraints into a linear form so that integer programming can be applied is a difficult task. In addition, the various integer programming techniques do not guarantee that optimal solutions can be obtained in a reasonable time. Dynamic programming [59-63] has the dimensionality difficulties which increase with the increase of the number of state variables, and it is hard to solve problems with more than three constraints. Similarly, the maximum principle [30-33] has difficulty in solving problems with more than three constraints. Likewise geometric programming [64-66] is restricted to problems that can be formulated by posynomial functions.

The sequential unconstrained minimization technique (SUMT) [47-52], the generalized reduced gradient method (GRG) [34-42] modified sequential simplex pattern search [43-46], and the generalized Lagrangian function method [55-58] are probably the few techniques that have been demonstrated to be effective when applied to large-scale nonlinear programming problems. However, the solutions are nonintegers and hence the optimal solution which must be integer is not guaranteed.

5 CONCLUDING REMARKS

All the optimization techniques employed in the papers surveyed have limited success in solving some small-scale system reliability optimization problems. Few techniques have been demonstrated to be effective when applied to large-scale system reliability optimization problems.

There are some new directions which additional optimization work would be fruitful. For example, one extension to the usual reliability optimization problem is to include the determination of the optimal level of component reliability and the number of redundancies in each of the stages simultaneously. The problem is one in which the component failure rate is a variable and the optimal trade off between adding components in redundancy or improvement of the individual component reliability is to be determined. Another example is one where the optimization of multi-stage system reliability is achieved by choosing a more reliable component out of several possible candidates at stage 1, adding redundant components in parallel at stage 2, and using a k-out-of-n: G configuration at stage 3.

Cost data for improving systems reliability are critically needed. Very little of these cost data are available. In the formulation of objective functions or constraints, actual cost data are necessary to realistically model the problems.

Increasing complexity of modern-day equipment, both in military and commercial areas, has brought with it new engineering problems involving high performance,reliability, and maintainability. In this regard availability, which is a combined measure of maintainability and reliability, has received wide increased usage as a measure of system effectiveness. A survey of this literature is presented in reference (144). A logical classification of various aspects of this problem is presented.

REFERENCES

(1) ON RELIABILITY AND OPTIMIZATION TECHNIQUES

RELIABILITY

1. ARINC Res. Corp., Reliability Engineering, Englewood Cliffs, N.J.: Prentice-Hall (1964).

2. Amstadter, B. L., Reliability Mathematics, N.Y.: McGraw-Hill (1971).

3. Barlow, R. E., and F. Proschan, Mathematical Theory of Reliability, N.Y.: Wiley (1965).

4. Barlow, R. E., and F. Proschan, Statistical Theory of Reliability and Life Testing, N.Y.: Holt, Rinehart, and Winston (1975).

5. Bazovsky, I., Reliability Theory and Practice, Englewood Cliffs, N.J.: Prentice-Hall (1961).

6. Calabro, S. R., Reliability Principles and Practice, N.J.: McGraw-Hill (1962).

7. Gnedenko, B. V., Y. K. Belyayer, and A. D. Solovyev, Mathematical Methods of Reliability Theory, N.Y.: Academic Press (1969).

8. Grouchko, D. (ed.), Operations Research and Reliability, N.Y.: Gordon and Breach (1969).

9. Lloyd, D. K., and M. Lipow, Reliability: Management, Methods, and Mathematics, Englewood Cliffs, N.J.: Prentice-Hall (1962).

10. Mann, N. R., R. E. Shafer, and N. D. Singpurwalla, Methods for Statistical Analysis of Reliability and Life Data, N.Y.: Wiley (1974).

11. Meyer, P. L., Introductory Probability and Statistical Application, Mass.: Addison-Wesley Publishing Company (1966).

12. Myers, R. H., K. L. Wong, and H. M. Gordy, Reliability Engineering for Electronic System, N.Y.: Wiley (1964).

13. Pieruschaka, E., Principles of Reliability, Englewood Cliffs, N.J.: Prentice-Hall (1963).

13a. Polovko, A.M., Fundamentals of Reliability Theory, N.Y.: Academic (1968).

14. Rau, J. G., Optimization and Probability in Systems Engineering, N.Y.: Van Nostrand Reinhold Comp. (1970).

15. Sandler, G. H., System Reliability Engineering, Englewood Cliffs, N.J.: Prentice-Hall (1963).

16. Shooman, M. L., Probabilistic Reliability: An Engineering Approach, N.Y.: McGraw-Hill (1968).

17. Zelen, M. (ed.), Statistical Theory of Reliability, Madison, Wisconsin: University of Wisconsin Press. (1964).

GENERAL OPTIMIZATION TECHNIQUES

18. Abadie, J. (ed.), Integer and Nonlinear Programming, N.Y.: North-Holland/American Elsevier (1970).

19. Beveridge, G. S. G., and R. S. Schechter, Optimization: Theory and Practice, N.Y.: McGraw-Hill (1970).

20. Fletcher, R., Optimization, N.Y.: Academic Press (1969).

21. Leitmann, G., Optimization Techniques with Applications to Aerospace Systems, N.Y.: Academic Press (1962).

22. Himmelblau, D. M., Applied Nonlinear Programming, N.Y.: McGraw-Hill (1972).

23. Wilde, D. J., Optimum Seeking Methods, Englewood Cliffs, N.J.: Prentice-Hall (1964).

INTEGER PROGRAMMING

24. Garfinkel, R. S., and G. L. Nemhauser, Integer Programming, N.Y.: Wiley (1972).

25. 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).

26. Gomony, R. E., "Outline of an algorithm for integer solutions to linear problem," Bulletin of the American Mathematical Society, Vol. 64, No. 5, pp. 275-278 (1958).

27. Hu, T. C., Integer Programming and Network Flows, Mass.: Addison-Wesley Publishing Comp. (1969).

28. Salkin, H. M., Integer Programming, Mass.: Addison-Wesley (1975).

29. Taha, H. A., Integer Programming — Theory, Applications and Computations, N.Y.: Academic Press (1975).

THE MAXIMUM PRINCIPLE

30. Denn, M. M., Optimization by Variational Methods, N.Y.: McGraw-Hill (1969).

31. Fan, L. T., and C. S. Wang, The Discrete Maximum Principle - A Study of Multistage Systems Optimization, N.Y.: Wiley (1964).

32. Fan, L. T., The Continuous Maximum Principle — A Study of Complex Systems Optimization, N.Y.; Wiley (1966).

33. Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidge, and E. F. Mischenko, The Mathematical Theory of Optimal Processes, (English Translation by K. N. Trirogoff) Inerscience, N.Y. (1962).

GENERALIZED REDUCED GRADIENT METHOD

34. Abadie, J. and J. Guigou, Numerical experiments with the GRG method, in Integer and Nonlinear Programming, e.d. J. Abadie, Amsterdam: North Holland Publishing Co. (1970).

35. Abadie, J., Solution des questions de dégénérescence dans la methode GRG, EDF* note HI 143/100 due 25 (September 1969).

36. Abadie, J., Application of the GRG Algorithm to Optimal Control Problem, in Integer and Nonlinear Programming, e.d. J. Abadie, North Holland Publishing Co. Amsterdam (1970).

37. Abadie, J., and J. Carpenter, Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints, in Optimization, ed. R. Fletcher, N.Y.: Academic Press (1969).

38. Abadie, J. and J. Guigou, Gradient réduit généralisé, EDF* note HI 069/2 du Avril (1969).

39. Colvill, A. R., A comparative study of nonlinear programming codes, IBM New York Scientific Center, Report 320-2949 (1968).

40. Guigou, J., Presentation et utilisation du code GRG, EDF* note HI 102/02 du 9 Juin (1969).

41. Guigou, J., Presentation et utilisation du code GREG, EDF* note HI 582/2 due 25 Mai (1971).

42. Hwang, C. L., J. L. Williams, and L. T. Fan, Introduction to the Generalized Reduced Gradient Method, Institute for Systems Design and Optimization, Report No. 39, K.S.U., Manhattan, Kansas 66502.

MODIFIED SEQUENTIAL SIMPLEX PATTERN SEARCH

43. Box, M.J., "A new method of constrained optimization and a comparison with other methods", Computer Journal, Vol. 8, pp. 42-52 (1965).

44. Fan. L. T., C. L. Hwang, and F. A. Tillman, "A sequential simplex pattern search solution to production planning problems", AIIE Transactions, Vol. 1, pp. 267-273 (1969).

45. Nelder, J. A., and R. Mead. "A simplex mathod for function minimization", Computer Journal, Vol. 7, pp. 308-313 (1965).

46. Sheela, B. V., and P. Ramamoorthy, "SWIFT-A new constrained optimization technique", Computer Methods in Applied Mechanics and Engineering, Vol. 6, pp. 309-317 (1975).

SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE

47. Bracken, J., and G. P. McCormick, Selected Applications of Nonlinear Programming, N.Y.: Wiley (1968).

48. Fiacco, A. V., and G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, N.Y.: Wiley (1968).

49. Hwang, C. L., K. C. Lai, F. A. Tillman, ans L. T. Fan, "Optimization of system reliability by the sequential unconstrained minimization techniques", IEEE Transactions on Reliability, Vol. R-24, pp. 133-135 (1975)

50. Kowalik, J., and M.R. Osborne, Methods for Unconstrained Optimization Problems, Elsevier, N.Y. (1968).

51. Lai, K. C., Optimization of Industrial Management Systems by the Sequential Unconstrained Minimization Technique, A master's report, Kansas State University (1970).

52. McCormick, G. P., W. C. Mylander, III, and A. V. Fiacco, "RAC Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming", SHARE Number 3189.

METHOD OF LAGRANGE MULTIPLIERS AND THE KUHN-TUCKER CONDITIONS

53. Hwang, C. L., P. K. Gupta, and L. T. Fan, Method of Lagrange Multipliers and the Kuhn-Tucker-Conditions, Institute for Systems Design and Optimization, Report No. 60, Kansas State University (1974)

54. Kuhn, N. W. and Tucker, A. W., Non-Linear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statics and Probability, J. Neyman, Ed., University of California Press, Berkeley, California, pp. 481-492 (1951).

THE GENERALIZED LAGRANGIAN FUNCTIONS METHOD

55. Everett, H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimal Allocation of Resources", Operations Research, vol. 11, pp. 399-417 (1963).

56. F. A. Lootsma, "A Survey of Methods for Solving Constrained Minimization Problems via Unconstrained minimization", in Numerical Methods for Non-Linear Optimization, F. A. Lootsma (ed.) Academis Press (1973); also in Optimization and Design, M. Avriel, M. J. Rijckaert, and P. J. Wilde (eds.) Prentice-Hall, pp. 88-118 (1973).

57. Roode, J. D., Generalized Lagrange Function and Mathematical Programming, in Optimization, ed. R. Fletcher, N.Y.: Academic Press (1969).

58. Sayama, H., Y. Kameyama, H. Nahayama, and Y. Sawaiagi. The Generalized Lagrangian Function for Mathematical Programming Problems, Institute for Systems Design and Optimization, No. 55, Kansas State University, Manhattan, Kansas 66506 (1974).

DYNAMIC PROGRAMMING

59. Bellman, R., Dynamic Programming, Princeton, N.J.: Princeton University Press (1957).

60. Bellman, R., and S. E. Dreyfus, Applied Dynamic Programming, Princeton, N. J.: Princeton University Press (1962).

61. Bellman, R., Modern Analytic and Computational Methods in Science and Mathematics, N.Y.: Elsevier Publishing Co. (1968).

62. Hadley, G., Nonlinear and Dynamic Programming, Reading, Mass: Addison-Wesley (1964).

63. Nemhauser, G. L., Introduction to Dynamic Programming, N.Y.: Wiley (1967).

GEOMETRIC PROGRAMMING

64. Duffin, R. J., E. L. Peterson, and C. Zener, Geometric Programming, N.Y.: Wiley (1967).

65. Nijkamp, P., Planning of Industrial Complexes by Means of Geometric Programming, Rotterdam University Press, Rotterdam, Netherlands. (1972).

66. Zener, C., Engineering Design by Geometric Programming, N.Y.: Wiley-Interscience, (1971).

(2) ON OPTIMIZATION OF SYSTEMS RELIABILITY WITH REDUNDANCY

67. Aggarwal, K. K., and J. S. Gupta, "On minimizing the cost of reliable systems", IEEE Transactions on Reliability, Vol. R-24, No. 3, pp. 205 (August 1975).

68. Aggarwal, K. K., "Redundancy optimization in general systems", IEEE Transactions on Reliability, Vol. R-25, No. 5, pp. 330-332 (December 1976).

69. Aggarwal, K. K., J.S. Gupta, and K.B. Misra, "A new heuristic criterion for solving a redundancy optimization problem", IEEE Transactions on Reliability, Vol. R-24, No. 1, pp. 86-87 (April 1975).

70. Aggarwal, K. K., and K. B. Misra, and J. S. Gupta, "Reliability evaluation - a comparative study of different techniques", Microelectronics and Reliability, Vol. 14, No. 1, pp. 49-56 (Feb. 1975).

71. Banerjee, S. K., K. Rajamani, and S.S. Deshpande, "Optimal redundancy allocation for non series-parallel networks", IEEE Transactions on Reliability, Vol. R-25, No. 2, pp. 115-117 (June 1976).

72. Banerjee, S.K., and K. Rajamani, "Optimization of system reliability using a parametric approach", IEEE Transactions on Reliability, Vol. R-22, pp. 35-39 (April 1973).

73. Banerjee, S.K., and K. Rajamani, "Parametric representation fo probability in two dimensions - a new approach in system reliability evaluation", IEEE Transactions on Reliability, Vol. R-21, pp. 56-60 (February 1972).

74. Banerjee, S.K., and K. Rajamani, "Closed form solutions for delta-star and star-delta conversions of reliability networks", IEEE Transactions on Reliability, Vol. R-25, No. 2, pp. 118-119 (June 1976)

75. Bellamn, R.E., and S.E. Dreyfus, "Dynamic programming and reliability of multicomponent devices", Operations Research, Vol. 6, pp. 200-206 (March-April 1958).

76. Beraha, D. and K. B. Misra, "Reliability optimization through random search algorithm", Microelectronics and Reliability, Vol. 13, pp. 295-297 (August 1974).

77. Black, G., and F. Proschan, "On optimal redundancy", Operations Research, Vol. 7, pp. 581-588 (1959).

78. Bodin, L.D., "Optimization procedure for the analysis of coherent structures", IEEE Transactions on Reliability, Vol. R-18, No. 3, pp. 118-126 (August 1969).

79. Brown, D.B., "A computerized algorithm for determining the reliability of redundant configurations", IEEE Transactions on Reliability, Vol. R-20, No. 3, pp. 121-124 (August 1971).

80. Burton, R.M., and G.T. Howard, "Optimal system reliability for a mixed series and parallel structure", Journal of Mathematical Analysis and Applications, Vol. 28, pp. 370-382 (1969).

81. Everett, H. III, "Generalized Lagrange Multiplier Method for solving problems of optimum allocation of resources", Operations Research, Vol. 11, No. 3, pp. 399-417 (May-June 1963).

82. Fan, L.T., C.S. Wang, F.A. Tillman, and C.L. Hwang, "Optimization of systems reliability", IEEE Transactions on Reliability, Vol. R-16, pp. 81-86 (1967).

83. Federowicz, A.J., and M. Mazumdar, "Use of geometric programming to maximize reliability achieved by redundancy", Operations Research Vol. 16, No. 5, pp. 948-954 (Sept.-Oct. 1968)

84. Fyffe, D.E., W.W. Hines, and N.K. Lee, "System reliability allocation and a computational algorithm", IEEE Transactions on Reliability, Vol. R-17, No. 2, pp. 64-69 (June 1968)

85. Geisler, M.A., and H.W. Karr, "The design of military supply tables for spare parts", Operations Research, Vol. 4, pp. 431-444 (1956).

86. Gen, M., H. Okuno, and S. Shinofuji, "An optimizing method in system reliability with failure-modes by implicit enumeration algorithm", J. of the Operations Research of Japan, Vol. 19, pp. 99-116 (June 1976).

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

88. Gordon, R., "Optimum component redundancy for maximum system reliability", Operations Research, Vol. 5, pp. 229-243 (April 1957).

89. Hees, Ir.R.N.v., and Ir.H.W.v.d. Meerendonk, "Optimal reliability of parallel multi-component systems", Operations Research Quarterly, Vol. 12, No. 1, pp. 16-26 (May 1961).

90. 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).

91. Hwang, C.L., K.C. Lai, F.A. Tillman, and L.T. Fan, "Optimization of system reliability by the sequential unconstrained minimization techniques", IEEE Transactions on Reliability, Vol. R-24, pp. 133-135 (1975).

92. 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).

93. Inoue, K., S.L. Gandhi, and E.J. Henley, "Optimal reliability design of process systems", IEEE Transactions on Reliability, Vol. R-23, No, 1, pp. 29-33 (April 1974)

94. Jensen, P.A., "Optimization of series-parallel-series networks", Operations Research, Vol. 18, pp. 471-482 (May-June 1970).

95. Kneale, S.G., "Reliability of parallel systems with repair and switching", Proceedings of the Seventh National Symposium on Reliability and Qualtiy Control, pp. 129-133 (1961).

96. Kettelle, J.D., "Least-cost allocation of reliability investment", Operations Research, Vol. 10, pp. 249-265 (March-April 1967).

97. Kim, Y.H., K.E. Case, and P.M. Glare, "A method for computing complex system reliability", IEEE Transactions on Reliability, Vol. R-21, No. 4, pp. 215-219 (November 1972).

98. Kolesar, P.J., "Linear programming and the reliability of multicomponent systems", Naval Research Logistics Quarterly, Vol. 15, pp. 317-327 (Sept. 1967).

99. Kulshrestha, D.K.. and M.C. Gupta, "Use of dynamic programming for reliability engineers, IEEE Transactions on Reliability, Vol. R-22, pp. 240-241 (Oct. 1973).

100. Kuo, W., "Optimization Techniques for Systems Reliability with Redundancy", M. S. Thesis, Kansas State University, (1977).

101. Lambert, B.K., A.G. Walvekar, and J.P. Hirmas, "Optimal redundancy and availability allocation in multistage systems", IEEE Transactions on Reliability, Vol. R-20, No. 3, pp. 182-185 (August 1971).

102. Laut, S., "Subsystem optimization effectiveness improvement by the option tradeoff analysis process", IEEE Transactions on Systems Science and Cybernetics, Vol. SSC-4, No. 2, pp. 133-137 (July 1968)

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

104. Liittschwager, J.M., "Dynamic programming in the solution of a multistage reliability problem", Journal of Industrial Engineering, Vol. 15-16, pp. 168-175 (1964-1965).

105. 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).

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

107. 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).

108. Messinger, M., and Shooman, H., "Technique for optimum spares allocation: a tutorial review", IEEE Transactions on Reliability, Vol. R-19, pp. 156-166 (November 1970)

109. Misra, K.B., "A simple approach for constrained redundancy optimization problems", IEEE Transactions on Reliability, Vol. R-21, No. 1, pp. 30-34 (Feruary 1972).

110. Misra, K.B., "Reliability optimization of a series-parallel system", IEEE Transactions on Reliability, Vol. R-21, No. 4, pp. 230-238 (November 1972).

111. Misra, K.B., and M.D. Ljubojevic, "Optimal Reliability design of a system: a new look", IEEE Transactions on Reliability, Vol. R-22, pp. 255-258 (December 1973).

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

113. Misra, K. B., "On optimal reliability design: A review", IFAC for 6th World Conference, Boston, Mass. USA pp. 3.4,1-3.4,10 (August 1975).

114. Misra, K. B., and J. Sharma, "A new geometric programming formulation for a reliability problem", International Journal of ~~Quality~~ Control, Vol. 18, No. 3, pp. 497-503 (1973).

115. Misra, K. B., and C. E. Carter, "Redundancy allocation in a system with many stages", Microelectronics and Reliability, Vol. 12, pp. 223-228 (June 1973).

116. 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).

117. Misra, K. B., "A method for redundancy allocation", Microelectronics and Reliability, Vol. 12, pp. 389-393 (October 1973).

118. Misra, K. B., "Reliability optimization through sequential simplex search", Int. J. Control, Vol. 18, pp. 173-183 (1973).

119. Misra, K. B., and J. Sharma, "Reliability optimization with integer constraint coefficients", Microelectronics and Reliability, Vol. 12, pp. 431-433 (November 1973).

120. Mizukami, K., "Optimum redundancy for maximum system reliability by the method of convex and integer programming", Operations Research, Vol. 16, pp. 392-406 (March-April 1968).

121. Morrison, D.F., "The optimum allocation of spare components in system", Technometrics, Vol. 3, No. 3, pp. 399-406 (August 1961).

122. Moskowitz, F., and J.B. McLean, "Some reliability aspects of system design", IRE Transactions on Reliability and Quality Control, Vol. PGRQC-8, pp. 7-35 (September 1956).

123. Myers, B.L., and N.L. Enrich, "Algorithmic optimization of system reliability" Annual Technical Conference Transactions, American Society for Quality Control, pp. 455-460 (1968).

124. Nelson, A.C., Jr. I.R. Batts and R.L. Beadles, "A computer program for approximating system reliability", IEEE Transactions on Reliability, Vol. R-19, pp. 61-60 (May 1970)

125. Neuner, G.E., and R.N. Miller, "Resource allocation for maximum reliability", Proceedings of 1966 Annual Symposium on Reliability, pp. 332-346 (January 1966).

126. Proschan, F., "Redundancy for reliability improvement", in Statistical Theory of Reliability (ed. M. Zelen), Madison, The University of Wisconsin Press, pp. 55-70 (1964).

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

128. Rudd, D.F., "Reliability theory in chemical system design", I&EC Fundamental, Vol. 1, No. 2, pp. 138-143 (May 1962).

129. Sasaki, M., "A simplified mehtod of obtaining highest system reliability", Proceedings of the Eighth National Symposium on Reliability and Quality Control, pp. 489-502 (1962).

130. Sasaki, M., "An easy allotment method achieving maximum system reliability", Proceedings of the Ninth National Symposium on Reliability and Quality Control, pp. 109-124 (1963).

131. Selman, V., and N.T. Grisamore, "Optimum system analysis by linear programming", 1966 Processing Annual Symposium on Reliability, American Society for Quality Control, pp. 696-703 (1966)

132. Sharma, J., and K.V. Venkateswaran, "A direct method for maximizing the system reliability", IEEE Transactions on Reliability, Vol. R-20, No. 4, pp. 256-259 (November 1971).

133. Shershin, A.-C., "Mathematical optimization techniques for the simultaneous apportionments of reliability and maintainability", Operations Research, Vol. 18, No. 1, pp. 95-106 (January-February 1970).

134. Shetty, H.V.K., and D.P. Sengupta, "Reliability optimization using SLUMT IEEE Transactions on Reliability, Vol. R-24, No. 1, pp. 30-81 (April 1975).

135. 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).

136. Tillman, F.A., C.H. Lie, and C.L. Hwang, "Analysis of pseudo-reliability of a combat tank system and its optimal design", IEEE Transactions on Reliability, Vol. R-25, pp. 239-242 (1976).

137. Tillman, F.A., amd J.M. Liittschwager, "Integer programming formulation of constrained reliability problems", Management Science, Vol. 13, No. 11, pp. 887-899 (July 1967).

138. 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. R-17, No. 3, pp. 153-157 (September 1968)

139. 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).

140. Tillman, F.A., C.L. Hwang, L.T. Fan, and K.C. Lai, "Optimal reliability of a complex system", IEEE Transactions on Reliability. Vol. R-19, No. 3, pp. 95-100 (1970).

141. Webster, L.R., "Optimum system reliability and cost effectiveness", Proceedings of 1967 Annual Symposium on Reliability, pp. 489-500 (January 1967).

142. Webster, L.R., "Choosing optimum system configurations", Proceedings of the Tenth National Symposium on Reliability & Quality Control, Washington, D.C., pp. 345-359 (1964).

143. Woodhouse, C.F., "Optimal redundancy allocation by dynamic programming", IEEE Transactions on Reliability, Vol. R-21, No. 1, pp. 60-62 (February 1972).

144. Lie, C.H., C.L. Hwang, and F. A. Tillman, "Availability of Maintained systems: a state-of-the-art survey" AIIE Transactions, (in press 1977).

CHAPTER 3 OPTIMIZATION OF SYSTEM RELIABILITY

Optimization techniques have their inherent characteristics and specific superiorities to solve general linear or nonlinear programming problems. In this chapter, various optimization techniques are treated to

(1) maximize the system reliability by adding the redundant components in each specified subsystem,

(2) maximize the system reliability by choosing a suitable stage reliability in each specified subsystem, or

(3) minimize the "cost" of the system while satisfying the minimum requirement of the system reliability.

(4) minimize the "cost" of a multi-function system while satisfying the minimum requirement of each individual system reliability.

"Cost" constraints of cost, weight, volume, or some combination of these factors are imposed to a system with series, parallel, or complex configuration. Each of the constraint functions is an increasing function of the component reliability and/or the number of components used at each stage. Various "cost" functions are used.

In the previous chapter, references for optimization techniques for system reliability with redundancy have been reviewed. The computational procedures of the optimization techniques, which have or have not been applied in the optimization of system reliability, will be described in this chapter. These optimization techniques are:

1. heuristic approach
2. dyanmic programming
3. the discrete maximum principle
4. the sequential unconstrained minimization technique (SUMT)
5. the generalized reduced gradient method (GRG)
6. method of Lagrange multipliers and the Kuhn-Tucker conditions
7. the generalized Lagrangian functions method
8. geometric programming
9. integer programming
10. others (a classical approach, parametric method, linear programming, and separable programming).

Among these optimization techniques, both GRG and the generalized Lagrangian functions method are very promising ones and have never been applied; heuristic approach and dynamic programming, having been successfully applied for the redundancy allocation problems, will be critically reviewed, classified, and modified. To cover a comprehensive discussion, the other optimization techniques will also be used to solve various reliability optimization problems. Before dealing with each specific optimization technique to system reliability problems, the following assumptions are made:

(1). Each subsystem is considered to be essential for the overall operational success of the mission, if all the subsystems are operationally in series.

(2). All the subsystems in series, parallel, or complex configuration are s-independent, also there are statistically independent parallel redundant components in each subsystem. In parallel redundancy, all units have the same risk of failure (or success) regardless of whether or not they are spares or active.

(3). Before the requirement of linearization for some specific optimization techniques, if necessary, the constraints of "cost" are not necessarily in linear forms.

(4). Good/bad is a sufficient description for each component, subsystem, and the whole system. In parallel case, unless being specified, only one component needs be good for the subsystem to be good, namely, it is generally 1-out-of m: G configuration. No assumption is made about the hazard rates of the components, except that it is reflected in the reliability of the components.

(5). Without the specific optimization knowledge of the mission requirements, realistic decisions on redundancy, design change, and other aspects of reliability improvement can not be reached. Tradeoffs between optimal redundancy components and "cost" measures can be considered only.

(6). So far as constraints are considered, each one is additive among subsystems.

(7). The redundant models are also based on the assumption that individual component or path failure has no effect on the operation of the surviving paths. Consider a simple parallel unit composed of two components, A and B, each of which can fail in either of two ways-open failure or short-circuit failure (diodes are a good example of elements which can fail in either mode). A short in either or two elements will result in unit failure; however, it is generally assumed that individual path failure does not result in unit failure or the probability of a short-circuit failure is 0.

(8). The connection nodes may spend some "cost", but are assumed perfect with the reliability of one as long as the system functions good.

3.1 HEURISTIC METHODS IN OPTIMAL SYSTEM RELIABILITY

1. Introduction

It is well-known that by using redundancy we can increase system reliability. Many techniques have been applied to obtain the solution of optimization problem, however, several heuristic approaches are very attractive for solving the redundancy allocation problems.

Four kinds of heuristic approaches are presented in this section. Sharma and Venkateswaran [1971] developed an intuitive procedure for allocating redundancy among subsystems. To improve the system reliability at each step of the algorithm, the procedure is to add a redundancy in the stage which has the highest stage unreliability. The algorithm was applied for solving multistage system problems subject to multiple nonlinear constraints. In this approach, the constraints are never in active. Misra [1972] then introduced an approach for redundundancy optimization problem with multiple linear constraints. In the process of solving a problem, the problem with r-constraints is decoupled into r-problems, each has one constraint. "Desirability factor", i.e., ratio of the percentage increase in the system reliability to the percentage increase of the corresponding cost, is introduced to determine a stage to which a redundancy is to be added. Aggarwal et al. [1975] improved Sharma-Venkateswarn approach by introducing a relative increment in reliability versus decrement in slacks (the balance of the resources) as a criterion to select the stage to which a redundancy is to be added for solving series system problems with multiple nonlinear constraints. Aggarwal [1976] extended the approach to a problem of complex systems. Recently, Nakagawa and Nakashima [1977] presented the fourth approach to solve a different type of series system (described later). In this approach a thorough consideration of the balance between the objective function and the constraints is especially emphasized. A modified Nakagawa-Nakashima approach for solving complex system problems is also presented.

The approaches are illustrated by solving four examples. The first example is a five-stage series system with three non-linear constraints. The second one is a complex (non-series-parallel) system to which one linear constraint is imposed. The third one is a four-stage series system with two linear cost constraints. The fourth one is more complex than a simple parallel redundancy problems. These examples are presented here.

Example 1

This problem was presented oringinally by Tillman - Liittschiwager [1967], and used by Tillman et al. [1968], Sharma - Venkateswaran [1971], and others for demonstrating many optimization techniques.

The five-stage problem is stated as

Maximize

$$R_s = \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}] \tag{1}$$

subject to

$$\begin{aligned} g_1 &= \sum_{j=1}^{5} p_j (x_j)^2 \leq P \\ g_2 &= \sum_{j=1}^{5} c_j (x_j + \exp (x_j/4)) \leq C \\ g_3 &= \sum_{j=1}^{5} w_j x_j \exp (x_j/4) \leq W \end{aligned} \tag{2}$$

where $x_j \geq 1$, $j = 1, 2, \ldots, 5$, are integers.

The constants associated with the five-stage problem are

Constants Assigned for Five-Stage Problem

j	R_j	p_i	P	c_j	C	w_j	W
1	0.80	1		7		7	
2	0.85	2		7		8	
3	0.90	3	110	5	175	8	200
4	0.65	4		9		6	
5	0.75	2		4		9	

Example 2

Consider a non-series-parallel system shown in Fig. 1 [Aggarwal, 1975]. Let there be only one cost constraint and 20 units is the maximum permissible cost. The data for the various subsystems are

$$R_1 = 0.70,\ R_2 = 0.85,\ R_3 = 0.75,\ R_4 = 0.80,\ R_5 = 0.90$$

$$C_1 = 2\quad,\ C_2 = 3\quad,\ C_3 = 2\quad,\ C_4 = 3\quad,\ C_5 = 1$$

The problem is

Minimize

$$\begin{aligned} Q_s = {} & Q_1'Q_3' + Q_2'Q_4' + Q_1'Q_4'Q_5' + Q_2'Q_3'Q_5' - Q_1'Q_2'Q_3'Q_4' \\ & - Q_1'Q_3'Q_4'Q_5' - Q_1'Q_2'Q_3'Q_5' - Q_1'Q_2'Q_4'Q_5' - Q_2'Q_3'Q_4'Q_5' \\ & + 2Q_1'Q_2'Q_3'Q_4'Q_5' \end{aligned} \tag{3}$$

subject to

$$g = \sum_{j=1}^{5} c_j x_j \leq C \tag{4}$$

where $Q_j' = (1 - R_j)^{x_j}$, $x_j \geq 1$, $j = 1, 2, \ldots, 5$, are integers.

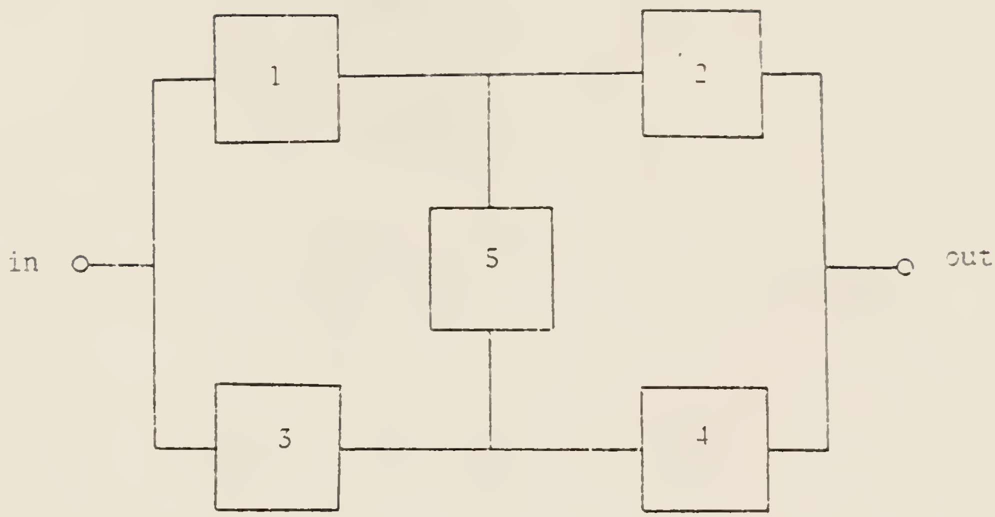


Fig. 1 A Bridge Structure

Example 3

Consider an example of a series system of four stages. The component reliability, cost, and weight data are:

Stage, j	1	2	3	4
Component reliability, R_j	0.80	0.70	0.75	0.85
Cost, c_j	1.2	2.3	3.4	4.5
Weight, w_j	5	4	8	7

The system cost and weight are 56 and 120, respectively.

The problem is

Maximize

$$R_s = \prod_{j=1}^{4} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=1}^{4} c_j x_j \leq 56$$

$$g_2 = \sum_{j=1}^{4} w_j x_j \leq 120$$

where $x_j \geq 1$, $j = 1, 2, 3, 4$, are integers.

Example 4

This example presented by Nakagawa-Nakashima [1977] is more complex than a simple parallel-series redundancy problem.

Consider the system composed of 3 stages operating in series. The system reliability is increased by choosing a more reliable component out of 4 candidates at stage 1, adding redundant components in parallel at stages 2 and using 2-out-of- $(x_3 + 1)$: G configuration at stage 3. The problem is

Maximize

$$R_s = \prod_{j=1}^{3} R_j'(x_j) \tag{5}$$

subject to

$$g_1 = 4 \cdot \exp\left\{\frac{0.02}{1 - R_1(x_1)}\right\} + 5x_2 + 2(x_3 + 1) \leq 45$$

$$g_2 = 5 + e^{x_1/8} + 3(x_2 + e^{x_2/4}) + 5(x_3 + 1 - e^{x_3/4}) \leq 75$$

$$g_3 = 10 + 8x_2 \cdot e^{x_2/4} + 6x_3 \cdot e^{x_3/4} \leq 240 \tag{6}$$

where

$R_1'(x_1) = 0.88,\ 0.92,\ 0.98,\ 0.99$ for $x_1 = 1, 2, 3, 4$ respectively

$$R_2'(x_2) = 1 - (1 - 0.81)^{x_2}$$

$$R_3'(x_3) = \sum_{k=2}^{x_3+1} \binom{x_3+1}{k} 0.77^k (1 - 0.77)^{x_3+1-k}$$

<u>Example 5</u>

A multi-function system is considered, which contains N distinct components and to increase the reliability of each, parallel or stand-by redundancy can be used. The problem is stated by Ushakov [9] as:

Minimize

$$C_s = \sum_{j=1}^{N} c_j x_j$$

subject to

$$R_{si} = \prod_{j \in J_i} R_j'(x_j) \geq R_{si,min} \qquad i = 1,2, \ldots, k$$

$$x_j \geq 0,\ j = 1,2, \ldots, N, \text{ are integers.}$$

where J_i denotes the subset of components that ensure the execution of function i with the minimum requirement for the ith function reliability, R_i, i = 1,2, ..., k.

The data associated with this example are:

Stage, j	1	2	3	
Component reliability, R_j	0.8	0.75	0.85	$R_{s1,min}=0.94$
Cost, c_j	3	2	4	$R_{s2,min}=0.96$

2 Heuristic Method - Sharma - Venkateswaran's Approach

Formulation of the Problem

In addition to the general assumptions made for the system reliability optimization problems of an N-stage in series with x_j redundant components at stage j, the unreliability of one component at the jth stage, Q_j, j=1,2,...,N, should be small enough ($\leq$ 0.5) so that

$$Q_s = 1 - \prod_{j=1}^{N} (1 - Q_j^{x_j})$$

can be approximated to

$$Q_s \approx \sum_{j=1}^{N} Q_j^{x_j}$$

where Q_s is the system unreliability. Therefore, the system reliability problem subject to nonlinear cost constraints can be formulated as

Minimize

$$Q_s = \sum_{j=1}^{N} Q_j^{x_j} \tag{7}$$

subject to

$$\sum_{j=1}^{N} g_{ij} (x_j) \leq b_i \qquad i=1,2,...,r \tag{8}$$

where g_{ij} (x_j) is the resource i consumed in stage j, and b_i is the available resource for constraint i.

The objective is to reduce Q_s in successive steps. The procedure at each step is to add one redundant component to the stage with the highest $Q_j^{x_j}$ in eq. (.7), if constraints in e.g. (.8) are not violated.

Therefore, the constraints become active only in the neighborhood of the boundary of the feasible region. The sequential steps involved in solving the problem are as follows:

Step 1. Assign $x_j = 1$ for $j=1,2,\ldots,N$. Because this is a cascade system, there must be at least one component in each stage and should not violate any constraints at all.

Step 2. Find the stage which has the highest unreliability. Add a redundant component to that stage.

Step 3. Check the constraints:

a) if any constraint is violated, go to Step 4.

b) if no constraint has been violated, go to Step 2.

c) if any constraint is exactly satisfied, stop.

The current x_j's are the optimum numbers of allocation.

Step 4. Remove the redundant component added in Step 2. The resulting number is the optimum allocation for that stage. Remove this stage from further consideration.

Step 5. If all stages have been removed from consideration, the current x_j's are the optimum solution. Otherwise, go to Step 2.

Numerical Examples

Example 1

The five-stage with three nonlinear constraints problem shall be solved here. The objective of the problem (e.g. (1)) can be approximated by

Minimize

$$Q_s = (1 - R_1)^{x_1} + (1 - R_2)^{x_2} + (1 - R_3)^{x_3} + (1 - R_4)^{x_4} + (1 - R_5)^{x_5}$$

where $(1 - R_1)^{x_1}$, $(1 - R_2)^{x_2}$, $(1 - R_3)^{x_3}$, $(1 - R_4)^{x_4}$, $(1 - R_5)^{x_5}$ are stage unreliabilities and are represented by Q_1', Q_2', Q_3', Q_4', Q_5', respectively.

The basic allocation (1, 1, 1, 1, 1) is assigned to the system. The stage unreliabilities under this configuration are (0.20, 0.15, 0.10, 0.35, 0.25). The resources consumed are (12, 73.1, 48.8) which have not violated the constraints. Since stage 4 has the highest unreliability, i.e., $Q_4' = 0.35$, we add one redundancy to this stage to form the system configuration (1, 1, 1, 2, 1) and consume the resources (24, 85.4, 60.8). Following the steps of the algorithm, we obtain the results presented in Table 1. The optimum result is $(x_1, x_2, x_3, x_4, x_5) = (3, 2, 2, 3, 3)$ with the system reliability of $R_s = 1 - (0.008 + 0.0225 + 0.01 + 0.04288 + 0.01562) = 0.90$.

It is worth noting that at the optimal configuration (3, 2, 2, 3, 3), no constraints are violated, Q_4' is already removed from further consideration, and $Q_2' = 0.0225$ is the highest unreliability, so a redundant component may be added to stage 2 to form a new system configuration (3, 3, 2, 3, 3) [Step 2]. However, constraint 3 is violated [Step 3a]; therefore, $x_2 = 2$ is the optimal one, and stage 2 is removed from further consideration [Step 4]. Go back to (3, 2, 2, 3, 3) configuration. Similarly, following the steps of the algorithm, a redundancy may be added to stage 5 ($Q_5' = 0.015625$, the largest unreliability among Q_1', Q_3', and Q_5') [Step 2]; however, constraint 3 is again violated [Step 3a], therefore, Step 5 is removed from further consideration [Step 4]. Similar procedures are applied to stage 3 and then to stage 1, but constraint 3 is violated in both cases as shown in Table 1. Therefore, the optimum allocation is (3, 2, 2, 3, 3).

Table 1 Results of Example 1 by Sharma-Venkateswarn's Method

Number of Components in Stage					Stage Unreliability					Constraints		
x_1	x_2	x_3	x_4	x_5	Q'_1	Q'_2	Q'_3	Q'_4	Q'_5	$g_1(\bar{x})$	$g_2(\bar{x})$	$g_3(\bar{x})$
1	1	1	1	1	0.2	0.15	0.1	0.35[a]	0.25	12	73.1	48.8
1	1	1	2	1	0.2	0.15	0.1	0.1225	0.25[a]	24	85.4	60.8
1	1	1	2	2	0.2[a]	0.15	0.1	0.1225	0.0625	30	90.8	79.0
2	1	1	2	2	0.04	0.15[a]	0.1	0.1225	0.0625	33	100.4	93.1
2	2	1	2	2	0.04	0.0225	0.1	0.1225[a]	0.0625	39	109.9	109.2
2	2	1	3	2	0.04	0.0225	0.1[a]	0.042875	0.0625	59	123.1	127.5
2	2	2	3	2	0.04	0.0225	0.01	0.042875	0.0625[a]	68	130.0	143.6
2	2	2	3	3	0.04[a]	0.0225	0.01	0.042875[b]	0.015625	78	136.0	171.1
3	2	2	3	3	0.008	0.0225	0.01	0.042875[b]	0.015625	83	146.1	192.5
3	3	2	3	3						93	157.4	216.9[c]
3	2	2	3	4						83	142.0	220.1[c]
3	2	3	3	3						91	153.0	208.6[c]
4	2	2	3	3						88	156.2	213.9[c]
3	2	2	4	3						93	159.3	210.2[c]

[a] This is the stage to which a redundant component is to be added.

[b] This indicates that the stage has been removed from further consideration.

[c] The constraint is violated.

Example 2

A solution for the complex (non-series-parallel) system is presented.

The basic allocation (1, 1, 1, 1, 1) is again assigned to the system. The stage unreliabilities under this configuration are (0.30, 0.15, 0.25, 0.20, 0.10) which consumes 11 cost units and obviously does not exceed the available resource. Since stage 1 has the highest unreliability, i.e., $Q_1' = 0.30$, we add one redundancy to this stage to form the new configuration (2, 1, 1, 1, 1) and check the consumed resource. Following the steps of the algorithm, we obtain the results which are summarized in Table 2. The last row of Table 2 shows the cost 20 is consumed under the allocation of (2, 1, 2, 2, 3). The system unreliability after substituting (2, 1, 2, 2, 3) into eg. (.3) is 0.0116, hence the system reliability is 0.9884.

Example 3

The procedures to reach the solution for this example are presented in Table 3.

Example 4

The results of this example are listed in Table 4.

Table 2 Results of Example 2 by Sharma-Venkateswaran's method

Number of Components in Stage					Stage Unreliabilities					Constraint
x_1	x_2	x_3	x_4	x_5	Q'_1	Q'_2	Q'_3	Q'_4	Q'_5	$g(\bar{x})$
1	1	1	1	1	0.30[a]	0.15	0.25	0.20	0.10	11
2	1	1	1	1	0.09	0.15	0.25[a]	0.20	0.10	13
2	1	2	1	1	0.09	0.15	0.0625	0.20[a]	0.10	15
2	1	2	2	1	0.09	0.15[b]	0.0625	0.04	0.10	18
2	2	2	2	1						21[c]
2	1	2	2	2	0.09[b]	0.15[b]	0.0625[b]	0.04[b]	0.01[a]	19
2	1	2	2	3	0.09	0.15	0.0625	0.04	0.001	20

[a] This is the stage to which a redundant component is to be added.

[b] This indicates that the stage has been removed from further consideration.

[c] The constraint is violated.

Table 3 Results of Example 3 by Sharma-Venkateswaran's method

Number of Components in stage				State Unreliability				Constraints	
x_1	x_2	x_3	x_4	Q'_1	Q'_2	Q'_3	Q'_4	$g_1(\bar{x})$	$g_2(\bar{x})$
1	1	1	1	0.20	0.30[a]	0.25	0.15	11.4	24
1	2	1	1	0.20	0.09	0.25[a]	0.15	13.7	28
1	2	2	1	0.20[a]	0.09	0.0625	0.15	17.1	36
2	2	2	1	0.04	0.09	0.0625	0.15[a]	18.3	41
2	2	2	2	0.04	0.09[a]	0.0625	0.0225	22.8	48
				.	.	.	.	.	.
				.	.	.	.	.	.
				.	.	.	.	.	.
4	5	5	4	0.0016	0.00243[a]	0.0009765	0.00050625	51.3	108
4	6	5	4	0.0016[a]	0.000729	0.0009765	0.00050625	53.6	112
5	6	5	4	0.00032	0.000729	0.0009765	0.00050625	54.8	117

[a] This is the stage to which a redundant component is to be added.

Table 4 Results of Example 4 by Sharma-Venkateswaran's Method

Number of Components in Stage			Stage Unreliability			Constraints		
x_1	x_2	x_3	Q'_1	Q'_2	Q'_3	$g_1(\bar{x})$	$g_2(\bar{x})$	$g_3(\bar{x})$
1	1	1	0.12	0.19	0.4071[a]	13.73	26.92	27.98
1	1	2	0.12	0.19[a]	0.13437	15.73	33.74	40.06
1	2	2	0.12	0.0361	0.13437[a]	20.73	40.93	56.16
1	2	3	0.12[a]	0.0361	0.04027	22.73	48.27	74.48
2	2	3	0.08[a]	0.0361	0.04027	23.13	48.42	74.48
3	2	3	0.02	0.0361	0.04027[a]	28.87	48.59	74.48
3	2	4	0.02	0.0361[a]	0.0114	30.87	56.59	101.61
3	3	4	0.02[b]	0.0069	0.0114	35.87	60.63	126.04
4	3	4				54.55[c]	60.82	126.04
3	3	5	0.02[b]	0.0069	0.0114	37.88	70.32	165.51
3	4	5				42.88	76.39[c]	201.68[c]
3	3	6				39.88	78.91[c]	222.14[c]

[a] This is the stage to which a redundant component is to be added.

[b] This indicates that the stage has been removed from further consideration.

[c] The constraint is violated.

3 Heuristic Method - Aggarwal's Approach

Formulation of the Problem

Sharma-Venkateswaran's heuristic approach consists of adding redundancy to the stage where the stage unreliability is so far the highest. The method is applicable to problems with any number of general constraints. The method may not yield an optimum solution if the stages have components of similar reliability but quite different cost. Aggarwal et al. then proposed an alternate algorithm using a new criterion for selecting the stage where redundancy is to be added. In certain cases of Sharma-Venkataswaran's approach, slacks (balance of resources) prevent the addition of only one component to the particular stage having the lowest reliability but these slacks permit the addition of more than one component to other stage having higher reliability. The net increase in reliability for the latter might be more than that for the former.

This heuristic approach is based on the concept that a component is added to the stage where its addition produces the greatest ratio "increment increases in reliability" to the "product of decrements in slacks". This ratio is defined by

$$F_j(x_j) = \frac{\Delta(1-R_j)^{x_j}}{\prod_{i=1}^{5} \Delta g_{ij}(x_j)} \tag{9}$$

where

$$\Delta(1-R_j)^{x_j} = (1-R_j)^{x_j} - (1-R_j)^{x_j+1} = R_j(1-R_j)^{x_j}$$

and

$$\Delta g_{ij}(x_j) = g_{ij}(x_j+1) - g_{ij}(x_j)$$

$F_j(x_j)$ is a function of j as well as x_j; hence in the computation, it keeps changing even for fixed j. In the case of linear constraints, however, all

$F_j(x_j)$ can be evaluated by using recursive relation

$$F_j(x_j+1) = Q_j F_j(x_j) \tag{10}$$

The computational procedure is

Step 1. Let $\bar{x} = (x_1, x_2, \ldots, x_N) = (1, 1, \ldots, 1)$.

Step 2. a) Calculate $F_j(x_j)$ for all j using (.9).

b) Select the stage having the highest $F_j(x_j)$. A redundant component is proposed to add to that stage.

Step 3. Check to see if the constraints are violated.

a) If the solution is still feasible, add one redundant component to the stage having the highest $F_j(x_j)$. Modify the value of x_j and hence $F_j(x_j)$ and go to Step 2.

b) If at least one constraint is exactly satisfied; the current value of $\bar{x}$ is an optimal solution.

c) If at least one constraint is violated, cancel the proposed addition of the redundant component; remove that stage from further consideration and repeat step 2. When all the stages are excluded from further consideration, the current values of $\bar{x}$ are the optimal solution.

Step 4. Calculate the system reliability, R_s, for the optimum $\bar{x}^*$.

<u>Numerical Examples</u>

<u>Example 1</u> To obtain the selection factors, $\Delta g_{ij}(x_j)$, i=1,2,3,..., are:

$$\Delta g_{1j}(x_j) = p_j(x_j+1)^2 - p_j(x_j)^2$$

$$\Delta g_{2j}(x_j) = c_j\{(x_j+1) + \exp[(x_j+1)/4]\} - c_j[x_j - \exp x_j/4)]$$

$$\Delta g_{3j}(x_j) = w_j(x_j+1)\exp[(x_j+1)/4] - w_j x_j \exp(x_j/4)$$

and the selection factors are:

$$F_j(X_j) = \frac{R_j Q_j^{x_j}}{\prod_{i=1}^{3} \Delta g_{ij}(x_j)}, \; j=1, 2, \ldots, 5$$

Starting with $\bar{x}$ = (1, 1, 1, 1, 1) and add one component to a stage at a time as shown in Table 5. The stage selection factors of (1, 1, 1, 1, 1) are (0.000396, 0.000138, 0.000091, 0.000128, 0.000316). The resources consumed are (12, 73.09, 48.79) which have not violated the constraints. Since stage 1 has the highest stage selection factor, i.e., $F_1(x_1)$ = 0.000396, we add one redundancy to this stage to form the system configuration (2, 1 , 1, 1, 1), and check the consumed resources (15, 82.64, 62.88). Following steps of the computational procedures, the final optimum result is obtained as $(x_1, x_2, x_3, x_4, x_5)$ = (3, 2, 2, 3, 3). The optimum system reliability is 0.9045.

Example 2

This example, solved by Aggarwal's approach, shows that the Sharma-Venkateswaran's approach will not always give an optimum solution.

The allocation of redundancy at each subsystem is started at $\bar{x}$ = (1, 1, 1, 1, 1). The stage selection factors of this configuration are (0.02649, 0.01783, 0.02759, 0.01068, 0.00533) and the resource consumed is 11. Since subsystem 3 has the highest stage selection factor, i.e., $F_3(x_3)$ = 0.02759, we add one redundancy to this stage to form the system configuration (1, 1, 2, 1, 1) and to consume 13 units cost. Following the computational procedures, the final optimum result is with the system configuration of $x_1, x_2, x_3, x_4, x_5)$ = (3, 1, 2, 2, 1) as shown in Table 6 . The optimum system reliability is 0.9914. It is noted that this problem solved by Sharma - Venkateswaran's

Table 5 Results of Example I by Aggarwal's Method

Number of Components in Stage					Stage Selection Factor					constraints		
x_1	x_2	x_3	x_4	x_5	$F_1(x_1)$	$F_2(x_2)$	$F_3(x_3)$	$F_4(x_4)$	$F_5(x_5)$	$g_1(\bar{x})$	$g_2(x)$	$g_3(\bar{x})$
1	1	1	1	1	0.000396[a]	0.000138	0.000091	0.000128	0.000316	12	73.09	18.79
2	1	1	1	1	0.0000291	0.000138	0.000091	0.000128	0.000316[a]	15	82.64	62.88
2	1	1	1	2	0.0000291	0.000138[a]	0.000091	0.000128	0.0000290	21	88.10	81.00
2	2	1	1	2	0.0000291	0.0000076	0.000091	0.000128[a]	0.0000290	27	97.65	97.11
2	2	1	2	2	0.0000291	0.0000076	0.000091[a]	0.000016	0.0000290	39	109.93	109.19
2	2	2	2	2	0.0000291[a]	0.0000076	0.0000033	0.000016	0.0000290	48	116.75	125.30
3	2	2	2	2	0.00000258	0.0000076	0.0000033	0.000016	0.0000290[a]	53	127.63	146.67
3	2	2	2	3	0.00000258	0.0000076	0.0000033	0.000016[a]	0.0000032	63	132.90	174.15
3	2	2	3	3	0.00000258	0.0000076[a]	0.0000033	0.0000025	0.0000032	83	145.11	197.47
3	3	2	3	3						93	157.39	216.90[c]
3	2	3	3	3						91	153.00	208.60[c]
3	2	2	3	4						80	142.00	220.10[c]
4	2	2	3	3						88	156.20	213.90[c]
3	2	2	4	3						93	159.30	210.20[c]

[a] This is the stage to which a redundant component is to be added.

[b] This indicated that the stage has been removed from further consideration.

[c] The constraint is violated.

Table 6 Results of Example 2 by Aggarwal's Method

Number of Components in Stage					Stage Selection Factor					constraint
x_1	x_2	x_3	x_4	x_5	$F_1(x_1)$	$F_2(x_2)$	$F_3(x_3)$	$F_4(x_4)$	$F_5(x_5)$	$g(\bar{x})$
1	1	1	1	1	0.02649	0.01783	0.02759[a]	0.01068	0.00533	11
1	1	2	1	1	0.00796	0.00825	0.00690	0.01149[a]	0.00478	13
1	1	2	2	1	0.01053[a]	0.00180	0.00720	0.00230	0.00048	16
2	1	2	2	1	0.00316[a]	0.00191	0.00160	0.00180	0.00033	18
3	1	2	2	1	0.000024	0.00030	0.00039	0.00092	0.00009	20

[a] This is the stage to which a redundant component is to be added.

[b] This indicates that the stage has been removed from further consideration.

approach gives the solution of the system configuration of (2, 1, 2, 2, 3) and system reliability of 0.9884 which is not an optimal solution.

Example 3

Since this example contains linear constraints, all $F_j(x_j)$ can be evaluated by using recursive relation of e.g. (10). This equation makes it more convenient to find the stage selection factors from which we decid the stage to add a redundancy. The summarized results is shown in Table 7.

Table 7 Results of Example 3 by Aggarwal's Method

Number of Components in Stage				Stage Selection Factor				constraints	
x_1	x_2	x_3	x_4	$F_1(x_1)$	$F_2(x_2)$	$F_3(x_3)$	$F_4(x_4)$	$g_1(\bar{x})$	$g_2(\bar{x})$
1	1	1	1	1.3333[a]	0.0913	0.0662	0.0283	11.4	24
2	1	1	1	0.2666[a]	0.0913	0.0662	0.0283	12.6	29
3	1	1	1	0.0533	0.0913[a]	0.0662	0.0283	13.8	34
3	2	1	1	0.0533	0.0274	0.0662[a]	0.0283	16.1	38
3	2	2	1	0.0533[a]	0.0274	0.0165	0.0283	19.5	46
4	2	2	1	0.0011	0.0274	0.0165	0.0283[a]	20.7	53
.				.	.	.	.	.	.
.				.	.	.	.	.	.
.				.	.	.	.	.	.
4	5	5	4	0.0011[a]	0.0007	0.0010	0.0006	51.3	104
5	5	5	4	0.0002	0.0007[a]	0.0013[b]	0.0006	52.5	109
5	6	5	4	0.0002	0.0002	0.0013	0.0006	54.8	117

[a] This is the stage to which a redundant component is to be added.

[b] This indicated that the stage has been removed from further consideration.

4 Heuristic Method - Misra's Approach

Formulation of the Problem

The approach is for the solution of the redundancy problem with multiple linear constraints. Basically, the solution to an r-constraint problem is obtained successively from the solution of r-unconstrained problems. At each step, an active constraint is picked out, and then the maximum gradient concept (explained later) is used to find a closer point. The actual computational procedure is presented in the following 5 steps.

Step 1. Within the feasible solution domain, the attainable reliability should be roughly estimated by the allocation of redundancy, stage by stage, until any constrainted resource is met.

Step 2. Using this estimate of system reliability, R_s, find the individual optimum allocations with respect to each "cost" constraint by

$$x_j = \frac{\log (1 - R_s^{a_j})}{\log Q_j} \tag{11}$$

where

$$a_j = \frac{c_j/\ln Q_j}{\sum_{j=1}^{N} c_j/\ln Q_j} \quad , \; j=1,2,\ldots,N$$

Usually a different system configuration is obtained for each constraint.

Step 3. From these allocations choose the highest system reliability as the reference reliability index for comparison. Other allocations will be lying on a lower reliability plane. To each allocation having lower system reliability, add one component to the stage which has the highest desirability factor defined by

$$F_j = \frac{\Delta R_s/R_s}{c_j/b_j} \quad , \; j=1,2,\ldots,N. \qquad (12)$$

Step 4. Now, each allocation is moved to a higher system reliability point, except the reference reliability index. Then

(a). go to Step 3, if none of these allocations give the same system reliability within the domain of feasible solutions (the constraints are not violated if "moved" to a higher reliability point).

(b). go to Step 3, if all allocations give the same system reliability and no constraint is violated by changing to a higher reliability point. Otherwise, go to Step 5 (a).

Step 5.

(a). Stop, the common allocation is the optimum one.

(b). If a common reliability point is not available, the allocation with the highest reliability will provide a near optimum point.

<u>A Numerical Example</u>

<u>Example 3</u>

To find a suboptimal system reliability which does not violate any of the cost or weight constraint, an enumeration method is used (see Table 8). The system reliability, R_s, is 0.99577 [Step 1]. Using this system reliability in eq. (11), the optimum allocations with respect to cost constraint and weight constraint are obtained as (5, 5, 4, 3) and (4, 6, 4, 3), respectively [Step 2].

Table 8 Suboptimum Results of Example 3 by an Enumeration Method

Stage 1	Stage 2	Stage 3	Stage 4	Cost	Weigh
1	1	1	1	11.4	24
2	1	1	1	12.6	29
2	2	1	1	14.9	33
2	2	2	1	18.3	41
2	2	2	2	22.8	48
3	2	2	2	24.0	53
3	3	2	2	26.3	57
3	3	3	2	29.7	65
3	3	3	3	34.2	72
4	3	3	3	35.4	77
4	4	3	3	38.7	81
4	4	4	3	41.1	89
4	4	4	4	45.6	96
5	4	4	4	46.8	101
5	5	4	4	49.1	105
5	5	5	4	52.5	113
5	5	5	5	57.0	120

R_s = 0.99577

Now we can construct Table 9 in the following. The reliability of system configuration of (5, 5, 4, 3) and (4, 6, 4, 3) are 0.9900 and 0.9904, respectively. Since 0.9900 is smaller than 0.9904, we then add one more redundancy to a stage of the system (5, 5, 4, 3). Since the desirability factors, F_j, for each stage are (0.0139, 0.0356, 0.0386, 0.0310), and 0.0386 is the largest, one redundancy is added to stage 3. The system reliability for (5, 5, 5, 3) is 0.9929. [Step 3]

This reliability is better than the system reliability (0.9904) of (4, 6, 4, 3). (5, 5, 5, 3) and (4, 6, 4, 3) are obviously not giving same system reliability and if we move to a higher reliability point the constraints are not violated. [Step 4a]. We, therefore, add one component to a stage of (4, 6, 4, 3). Since the desirability factors, F_j, for each stage are now (0.0287, 0.0270, 0.0382, 0.0399), and 0.0399 is the largest, we add one redundancy to stage 4. The resulting system reliability for (4, 6, 4, 4) is 0.9933 [Step 3]. This reliability then is compared with 0.9929. (5, 5, 5, 3) and (4, 6, 4, 4) are still not giving same system reliability and if move to a higher reliability point the constraints are not violated [Step 4a]. We, therefore, add one component to a stage of (5, 5, 5, 3). Following the iterative procedures, finally the common allocation (5, 6, 5, 4) is obtained. The consumed cost resource and weight resource are 54.8 and 117.0 respectively. Since a redundancy add to any stage will exceed the available resources, (5, 6, 5, 4) is the optimum allocation of the system which gives the system reliability 0.9975 [Step 5a].

Table 9 Results of Example 3 by Misra's Method

	Based on Cost Constraint											Based on Weight Constraint									
Iteration	x_1	x_2	x_3	x_4	Cost	R_s	F_1	F_2	F_3	F_4	x_1	x_2	x_3	x_4	Weight	R_s	1	F_2	F_3	F_4
1	5	5	4	3	44.6	0.9900	0.0139	0.0356	0.0386*	0.0310	4	6	4	3	97.0	0.9904	———	———	———	———
2	5	5	5	3	48.0	0.9929	———	———	———	———	4	6	4	3	97.0	0.9904	0.0287	0.0270	0.0382	0.0399*
3	5	5	5	3	48.0	0.9929	0.0139	0.0356*	0.0290	0.0322	4	6	4	4	104.0	0.9933	———	———	———	———
4	5	6	5	3	50.3	0.9946	———	———	———	———	4	6	4	4	104.0	0.9933	0.0287	0.0270	0.0382*	0.0223
5	5	6	5	3	50.3	0.9946	0.0139	0.0310	0.0290	0.0322*	4	6	5	4	112.0	0.9962	———	———	———	———
6	5	6	5	4	54.8	0.9975	———	———	———	———	4	6	5	4	112.0	0.9962	0.0287*	0.0270	0.0243	0.0223
7	5	6	5	4	54.8	0.9975	———	———	———	———	5	6	5	4	117.0	0.9975	———	———	———	———

*This indicates the maximum F_j, j = 1, 2, 3, 4.

5. Heuristic Method - Ushakov's Approach [8]

Formulation of the problem

Suppose that the system consists of n distinct elements and that, to increase the reliability of each of them, we can use an arbitrary type of stand-by. Let $R_j'(X_j)$ denote the probability of failure-free operation of element j when $X_j - 1$ stand-by elements are used to ensure its operability, let c_j denote the cost of a single element of type j and let J_i denote the subset of system elements that ensure execution of function i with given probability $P_{si,min}$ $(i = 1, \ldots, k)$. With this notation, we can formulate the following problem:

Find min $\sum_{j=1}^{n} c_j X_j$ under the condition $\prod_{j \in J_i} R_j'(X_j) \geq R_{si,min}$ $(i = 1, \ldots, k)$,

where X_j is a natural number $(j = 1, \ldots, n)$.

The computational procedure can be stated as.

1. We solve k problems of finding min $\sum_{j \in J_i} c_j X_j$ under the condition $\prod_{j \in J_i} R_j'(X_j) \geq P_{si,min}$ $(i = 1, \ldots, k)$. For problem i, we find the corresponding optimum values $X_1^i, \ldots, X_n^i$.

2. For element i, we find the greatest value, that is, $X_j^* = \max X_j^i$.

3. For each subset of elements J_i, we find a smaller subset (which we denote by J_i^{**}) that is necessary only for executing function i. We denote by J_i^* the remaining portion of the subset J_i.

4. For each subset J_i^*, we find the value of $\prod_{j \in J_i^*} R_j'(X_j^*) = R_i^*$. If J_i^* is empty, that is, if function i is executed with an independent group of elements, we take $R_i^* = 1$.

5. For each set J_i, we calculate $R_i^{**} = R_{si,min} / R_i^*$, which is a requirement on the probability of failure-free operation of the elements belonging to subset J_i^{**}.

6. In addition we solve k problems of finding min $\sum_{j=J_i^*} c_j X_j$ under the condition $\prod_{j=J_i^{**}} R_j'(X_j) \geq R_i^{**}$ $(i = 1, \ldots, k)$. We find the values of X_j^{**} .

7. For the solution we take the values of m_j^* for $j \in J_i^*$ found in item 2, and the values of X_j^{**} for $j \in \bar{J}_i^{**}$ found in item 6. Items 3-6 are introduced in order to lower, if possible, the superfluously high reliability in conservation of the given requirements on the probability of execution of the individual functions.

A Numerical Example

Example 5

Following the computational procedure to obtain

(1) $X_1' = 2$ $\qquad$ $X_2' = 3$

$X_1^2 = 3$ $\qquad$ $X_3^2 = 2$

(2) $X_1^* = 3$

$X_2^* = 3$

$X_3^2 = 2$

(3) $J_1^{**} = \{1, 2\}$ $J_2^{**} = \{1, 3\}$

$J_1^* = \{3\}$ $J_2^* = \{2\}$

(4) $R_1^* = 1 - (1 - 0.85)^2 = 0.9775$

$R_2^* = 1 - (1 - 0.75)^2 = 0.9844$

(5) $R_1^{**} = 0.94/0.9775 = 0.9616$

$R_2^{**} = 0.96/0.9844 = 0.9752$

(6) $X_1^{**} = 3$

$X_2^{**} = 3$

$X_3^{**} = 2$

(7) Since both the configuration $(X_1^{**}, X_2^{**}, X_3^{**}) = (3,3,2)$ and $(X_1, X_2, X_3) = (3, 3, 2)$ satisfy the reliability constraints and the former one costs only 23 which is less than 26 spent by the second one. Therefore the optimal solution is (3, 3, 2)

6. Nakagawa-Nakashima's Approach

6.1 Formulation of the Problem

In the previous three approaches, it is assumed that the component unreliability at each stage, Q_j, $j = 1,2, \ldots, n$ is small so that the objective function can be approximated. However, no approximation on the objective function shall be made in Nakagawa-Nakashima's approach.

We state again the general nonlinear optimization problem (Problem A) for an N-stage series system as

Problem A

Maximize

$$R_s = \prod_{j=1}^{N} R_j'(x_j)$$

subject to

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

$$1 \leq x_j \leq \bar{x}_j \qquad , j = 1,2, \ldots, N$$

where, $\bar{x}_j$ is the maximum number of components used at stage j, and all x_j's are integers.

Basing on the definitions of

$$\Delta f_j(x_j) = \begin{cases} \ell n R_j'(x_j) & , \text{for } x_j = 1 \\ \ell n R_j'(x_j) - \ell_n R_j'(x_j-1) & , \text{for } x_j > 1 \end{cases} \qquad (15)$$

and

$$\Delta g_{ij}(x_j) = \begin{cases} g_{ij}(x_j) & , \quad \text{for } x_j = 1 \\ g_{ij}(x_j) - g_{ij}(x_j - 1) & , \quad \text{for } x_j > 1 \end{cases} , \tag{14}$$

We transform Problem A to Problem B as follows:

$$\ell n R_s = \ell n \prod_{j=1}^{N} R_j'(x_j) \quad .$$

$$= \sum_{j=1}^{N} \ell n R_j'(x_j)$$

Basing on the definition in eq. (13)

$$\text{if } x_j = 1 , \qquad \sum_{j=1}^{N} \ell n R_j'(1) = \sum_{j=1}^{N} \Delta f_j(x),$$

$$\sum_{j=1}^{N} g_{ij}(1) = \sum_{j=1}^{N} \Delta g_{ij}(x_j)$$

$$\text{if } 1 < x_j \leq \bar{x}_j \quad , \qquad \sum_{j=1}^{N} \ell n R_j'(x_j) = \sum_{j=1}^{N} \Big\{ [\ell n R_j'(\bar{x}_j) - \ell n R_j'(\bar{x}_j - 1)] + [\ell n R_j'(\bar{x}_j - 1)$$

$$- \ell n R_j'(\bar{x}_j - 2)] + \ldots + [\ell n R_j'(2) - \ell n R_j'(1)] + \ell n R_j'(1) \Big\}$$

$$= \sum_{j=1}^{N} \sum_{x_j = 1}^{\bar{x}_j} \Delta f_j(x_j)$$

$$\sum_{j=1}^{N} g_{ij}(x_j) = \sum_{j=1}^{N} \{ [g_{ij}(\bar{x}_j) - g_{ij}(\bar{x}_j - 1)] + [g_{ij}(\bar{x}_j - 1) -$$

$$g_{ij}(\bar{x}_j - 2)] + \ldots + [g_{ij}(2) - g_{ij}(1)] +$$

$$[g_{ij}(1)] \}$$

$$= \sum_{j=1}^{N} \sum_{x_j = 1}^{\bar{x}_j} \Delta g_{ij}(x_j)$$

Problem B

Maximize

$$\ell n R_s = \sum_{j=1}^{N} \sum_{x_j=1}^{\bar{x}_j} \Delta f_j(x_j)$$

subject to

$$\sum_{j=1}^{N} \sum_{x_j=1}^{\bar{x}_j} \Delta g_{ij}(x_j) \leq b_i \qquad , \ i = 1,2, \ldots, r$$

$$1 \leq x_j \leq \bar{x}_j \qquad , \ j = 1,2, \ldots, N$$

where, $\bar{x}_j$ is the maximum number of components for stage j, and all x_j's are integers.

Since $R_j(x_j)$ and $g_{ij}(x_j)$ are monotonic increasing functions in x_j for j = 1,2, ..., N, and i = 1,2, ..., r,

then

$$\Delta f_j(x_j) \geq 0, \qquad \text{for all } j \text{ and } x_j,$$

and

$$\Delta g_{ij}(x_j) \geq 0, \qquad \text{for all } i,j, \text{ and } x_j.$$

The computational procedure for solving Problem B is stated as follows.

Step 1. Set the first current solution as

$$x^c = (x_1^c, x_2^c, \ldots, x_N^c) = (1,1, \ldots, 1)$$

Step 2. Calculate b_i^c for all i, where

$$b_i^c = b_i - \sum_{j=1}^{N} \sum_{x_j=1}^{x_j^c} \Delta g_{ij}(x_j). \quad (15)$$

Step 3. Calculate Δx_j for all j, where

$$\Delta x_j \equiv \min_i \{b_i^c / \Delta g_{ij}(x_j^c + 1)\}$$

Step 4. Let $L_{+1} = \{j \mid \Delta x_j \geq 1\}$

If L_{+1} is empty, stop. An optimal solution is obtained. Otherwise go to step 5.

Step 5. Search over m such that $S_m = \max_{j \in L_{+1}} \{S_j\}$,

where

$$S_j \equiv \Delta f_j(x_j^c + 1) \cdot [(1-\alpha) \cdot \min_{\ell \in L_{+1}} \{\Delta x_\ell\} + \alpha \, \Delta x_j]. \quad (16)$$

Step 6. If $x_m^c = \bar{x}_m$, then x_m^c is the optimal number used in stage m, exclude this stage and go to step 3. If $x_m^c < \bar{x}_m$, then set $x_m^c = x_m^c + 1$ and go to step 3.

The above procedure is for a given balancing coefficient "α". Optimal solutions for a set of "α" (probably α = 0, 0.1, ..., 0.9, 1.0, $1/\alpha$ = 0.9, 0.6, 0.3) should be obtained. The best solution among the solutions for the given set of "α's" is the optimal solution.

6.2 Numerical Examples

Example 1

The problem, after transformation, is formed as follows.

Maximize

$$\ln R_s = \sum_{j=1}^{5} \sum_{x_j=1}^{\bar{x}_j} \Delta f_j(x_j)$$

subject to

$$\sum_{j=1}^{5} \sum_{x_j=1}^{\bar{x}_j} \Delta g_{1j} (x_j) \leq P$$

$$\sum_{j=1}^{5} \sum_{x_j=1}^{\bar{x}_j} \Delta g_{2j} (x_j) \leq C$$

$$\sum_{j=1}^{5} \sum_{x_j=1}^{\bar{x}_j} \Delta g_{3j} (x_j) \leq W$$

$$1 \leq x_j \leq \bar{x}_j \qquad , j = 1,2, \ldots, 5$$

where, x_j's are integers, $\Delta f_j\ (x_j)$ and $\Delta g_{ij}\ (x_j)$ are given in (13) and (14)

Set the first current solution as $x^c = (x_1^c, x_2^c, x_3^c, x_4^c, x_5^c) = (1,1,1,1,1)$ [Step 1]. The current amounts of the resource available (b_1^c, b_2^c, b_3^c) are calculated by eq (15) and are (98, 101.91, 151.21) [Step 2]. Then the quantities obtained from the constraints at the current solution, $(\Delta x_1, \Delta x_2, \Delta x_3, \Delta x_4, \Delta x_5)$, are calculated to be (10.67, 9.39, 9.39, 8.17, 8.39) [Step 3]. Since Δx_j, j = 1,2,3,4,5, are greater than 1, then $L_{+1} = \{1,2,3,4,5\}$ which means the redundancy to any stage will not violate the constraints [Step 4]. We can find the stage sensititivies, $(S_1, S_2, S_3, S_4, S_5)$, to be (1.717, 1.227, 0.837, 2.451, 1.847), if $\alpha = 0.50$. Since stage 4 shows the most sensitive effect, i.e., $S_4 = 2.451$, we should add one redundancy to this stage [Step 5]. Following the steps of the algorithm, we obtain the results as presented in Table 12. The last system configuration in Table 12 is (3,2,2,3,3) and all the Δx_j, j = 1,2,3,4,5, are less than 1. This means that it is impossible to add any more redundancy to any stage, and (3,2,2,3,3) is the optimum solution. This is the same solution obtained by Sharma - Venkateswaran's approach and by Aggarwal et al.'s approach.

Table 10 Results of Example 1 by Nakagawa-Nakahima's Approach

Number of components in stage													Stage Sensitivity				
x_1	x_2	x_3	x_4	x_5	b_1^c	b_2^c	b_3^c	Δx_1	Δx_2	Δx_3	Δx_4	Δx_5	S_1	S_2	S_3	S_4	S_5
1	1	1	1	1	98	101.91	151.21	10.67	9.39	9.39	8.17	8.39	1.717	1.227	0.837	2.451[a]	1.84
1	1	1	2	1	86	89.63	139.13	9.38	8.64	8.64	4.30	7.68	1.247	0.904	0.617	0.520	1.336[a]
1	1	1	2	2	80	84.17	121.01	8.59	7.51	7.51	4.00	4.40	1.147[a]	0.805	0.549	0.347	0.205
2	1	1	2	2	77	74.62	106.92	5.00	6.64	6.64	3.85	3.89	0.145	0.733[a]	0.500	0.334	0.189
2	2	1	2	2	71	65.06	90.80	4.25	3.72	5.64	3.55	3.30	0.124	0.068	0.426[a]	0.298	0.161
2	2	2	2	2	62	58.24	74.70	3.49	3.06	3.06	3.10	2.72	0.102	0.056	0.026	0.253[a]	0.133
2	2	2	3	2	42	45.03	56.38	2.64	2.31	2.31	1.50	2.05	0.068	0.037	0.017	0.043	0.087[a]
2	2	2	3	3	32	39.15	28.89	1.35	1.18	1.18	1.06	0.71	0.040[a]	0.022	0.010	0.031	0.011
3	2	2	3	3	27	28.88	7.52	0.24	0.31	0.31	0.20	0.23[b]					

[a]This is the stage to which a redundant component is to be added.

[b]Since $L_{+1} = \phi$, stop the procedure.

Example 4

This example is presented in Nakagawa-Nakashima's paper. The results are summarized in Table 11. The optimal solution is $(x_1, x_2, x_3) = (3, 3, 5)$. The resources consumed are 37.88, 70.32, and 211.8 for g_1, g_2, and g_3 respectively.

Table 11 Results of Example 4 by Nakagawa-Nakashima's Approach

Number of components in stage									Stage sensitivity		
x_1	x_2	x_3	b_1^c	b_2^c	b_3^c	Δx_1	Δx_2	Δx_3	S_1	S_2	S_3
1	1	1	31.27	48.08	212.02	10.71	10.72	11.33	1.71	2.01	2.36[a]
1	1	2	29.27	41.26	199.94	8.07	9.35	9.22	1.70[a]	1.42	1.31
2	1	2	29.86	41.11	199.94	7.21	8.17	9.01	0.27	0.88[a]	0.31
2	2	2	23.86	33.91	183.83	5.10	4.73	5.03	0.09	0.21	0.42[a]
2	2	3	21.87	26.58	159.42	3.15	3.90	4.10	0.21[a]	0.11	0.14
3	2	3	16.13	26.41	159.42	2.33	3.10	2.97	0.04	0.09[a]	0.03
3	3	3	9.13	22.01	134.99	1.70	1.90	2.01	0.02	0.03	0.07[a]
3	3	4	9.13	14.37	96.11	0.43	1.21	1.71	0.002	0.007	0:01[a]
3	3	5	7.12	4.68	28.2	0.31	0.27	0.27[b]			

[a] This is the stage to which a redundant component is to be added

[b] Since $L_{+1} = \phi$, stop the procedure

6.3 Modification of Nakagawa - Nakashima's approach for complex system optimization

Nakagawa - Nakashima's approach can only solve series-system problems. For a complex system, e.g., Example 2, the problem does not have the objective function in the form of Problem A. Therefore, the problem cannot be transformed into Problem B.

To solve the problem of a complex configuration, $\Delta f_j(x_j)$ will be redefined as follows:

$$\Delta f_j(x_j) = Q_s(Q_1', \ldots, (Q_j' = Q_j^{x_j}), \ldots, Q_k') - Q_s(Q_1', \ldots, (Q_j' = Q_j^{x_j+1}), \ldots, Q_k')$$

$$\approx \frac{\partial Q_s}{\partial Q_j'} (Q_j^{x_j} - Q_j^{x_j+1})$$

$$= (1-Q_j)\, Q_j^{x_j} \frac{\partial Q_s}{\partial Q_j'}$$

$$= R_j Q_j^{x_j} \frac{\partial Q_s}{\partial Q_j'} \qquad \text{(A)}$$

As $\Delta f_j(x_j)$ is defined, we can follow the same computational procedures presented in Section 5.2 to obtain the optimal solution.

<u>Example 2</u>

To solve this example, we have to use eq. (A) to determine $\Delta f_j(x_j)$. In eq. (A), $\frac{\partial Q_s}{\partial Q_j'}$, $j = 1,2, \ldots, 5$, for this example are given by

$$\frac{\partial Q_s}{\partial Q_1'} = Q_3' + Q_4'Q_5' - Q_2'Q_3'Q_4' - Q_3'Q_4'Q_5' - Q_2'Q_3'Q_5' - Q_2'Q_4'Q_5' + 2\, Q_2'Q_3'Q_4'Q_5'$$

$$\frac{\partial Q_s}{\partial Q_2'} = Q_4' + Q_3'Q_5' - Q_1'Q_3'Q_5' - Q_1'Q_3'Q_4' - Q_1'Q_4'Q_5' - Q_3'Q_4'Q_5' + 2\, Q_1'Q_3'Q_4'Q_5'$$

$$\frac{\partial Q_s}{\partial Q_3'} = Q_1' + Q_2'Q_5' - Q_1'Q_2'Q_4' - Q_1'Q_4'Q_5' - Q_1'Q_2'Q_5' - Q_2'Q_4'Q_5' + 2\, Q_1'Q_2'Q_4'Q_5'$$

$$\frac{\partial Q_s}{\partial Q_4} = Q_2' + Q_1'Q_5' - Q_1'Q_2'Q_3' - Q_1'Q_3'Q_5' - Q_1'Q_2'Q_3' - Q_2'Q_3'Q_5' + 2\, Q_1'Q_2'Q_3'Q_5'$$

$$\frac{\partial Q_s}{\partial Q_5'} = Q_1'Q_4' + Q_2'Q_3' - Q_1'Q_3'Q_4' - Q_1'Q_2'Q_3' - Q_1'Q_2'Q_4' - Q_2'Q_3'Q_4' + 2\, Q_1'Q_2'Q_3'Q_4'$$

The proceeding to obtain the optimal result is presented in Table 12. The optimal result is $(x_1, x_2, x_3, x_4, x_5) = (3, 1, 2, 2, 1)$. The system reliability is 0.9914. The resource, 20, is totally consumed.

Table 12 Results of Example 2 by the Modified Nakagawa-Nakashima's Approach

Number of components in subsystem															
x_1	x_2	x_3	x_4	x_5	b[c]	Λx_1	Λx_2	Λx_3	Λx_4	Λx_5	S_1	S_2	S_3	S_4	S_5
1	1	1	1	1	9	4.5	3.0	4.5	3.0	9.0	0.199	0.161	0.207[a]	0.096	0.032
1	1	2	1	1	7	3.5	2.33	3.5	2.33	7.0	0.046	0.058	0.040	0.080[a]	0.022
1	1	2	2	1	4	2.0	1.33	2.0	1.33	4.0	0.035[a]	0.007	0.024	0.009	0.001
2	1	2	2	1	2	1.0	0.67	1.0	0.67	2.0	0.006[a]		0.003		0.001
3	1	2	2	1	0[b]										

[a]This is the stage to which a redundant component is to be added.

[b]Since no resource is available, stop the procedure.

REFERENCES

1. Aggarwal, K. K., "Redundancy optimization in general systems", IEEE Transactions on Reliability, Vol. R-25, No. 5, pp. 330-332 (December 1976).

2. Aggarwal, K. K., J. S. Gupta, and K. B. Misra, "A new heuristic criterion for solving a redundancy optimization problem", IEEE Transactions on Reliability, Vol. R-24, No. 1, pp. 86-87 (April 1975).

3. Misra, K. B., "A simple approach for constrained redundancy optimization problems", IEEE Transactions on Reliability, Vol. R-21, No. 1, pp. 30-34 (Feb. 1972).

4. Nakagawa, Y., and K. Nakashima, "A heuristic method for determining optimal reliability allocation", IEEE Transactions on Reliability, (August 1977).

5. Sharma, J., and K. V. Venkateswaran, "A direct method for maximizing the system reliability", IEEE Transactions on Reliability, Vol. R-20, No. 4, pp. 256-259 (November 1971).

6. Tillman, F. A., and J. M. Liittschwager, "Integer programming formulation of constrained reliability problems", Management Science, Vol. 13, No. 11, pp. 887-899 (July 1967).

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. Ushakov, I. A., "A heuristic method of optimization of the redundancy of multi-function systems". IZV. AN SSSR, Tekhnicheskaya Kibernetika (Engineering Cybernetics) Vol. 10, No. 4, pp. 612-613 (1972).

3.2 DYNAMIC PROGRAMMING APPLIED TO OPTIMAL SYSTEMS RELIABILITY

1. Introduction

Dynamic programming is based on the so-called "principle of optimality" [1-3] and employs the techniques of invariant imbedding. The problem is usually structured as an N-stage sequential decision problem so as to apply a dynamic programming approach for its solution. Bellman has shown that applying a dynamic programming approach will readily yield an exact solution to a system allocation problem. The basic characteristic of the approach is that in the computational procedure an N-variable decision problem is solved by a sequential solutions of N single-variable problems.

Various papers have presented the application of dynamic programming to a variety of problems. Problems treated in these papers can be classified into the following examples.

Example 1

The basic approach is illustrated by an example [Rudd, 1962] which is to locate the redundancy at each stage of a series system so that the net system profit will be maximized. [17]

Consider an N stage mixed system shown in Fig. 1 having $(x_j - 1)$ parallel redundancies at each stage; the system reliability is

$$R_s = \prod_{j=1}^{N} (1 - (1 - R_j)^{x_j}) \tag{1}$$

Let us assume that P is the profit obtained when the system operates successfully. The system reliability, R_s, is the fraction of the trials that are successful and hence the expected profit for the system is PR_s. Suppose that the costs C_j of the redundant components of the jth stage are the construction cost (suitably distributed over the life of the process) and the operating cost. The total cost of a system with redundancies is then

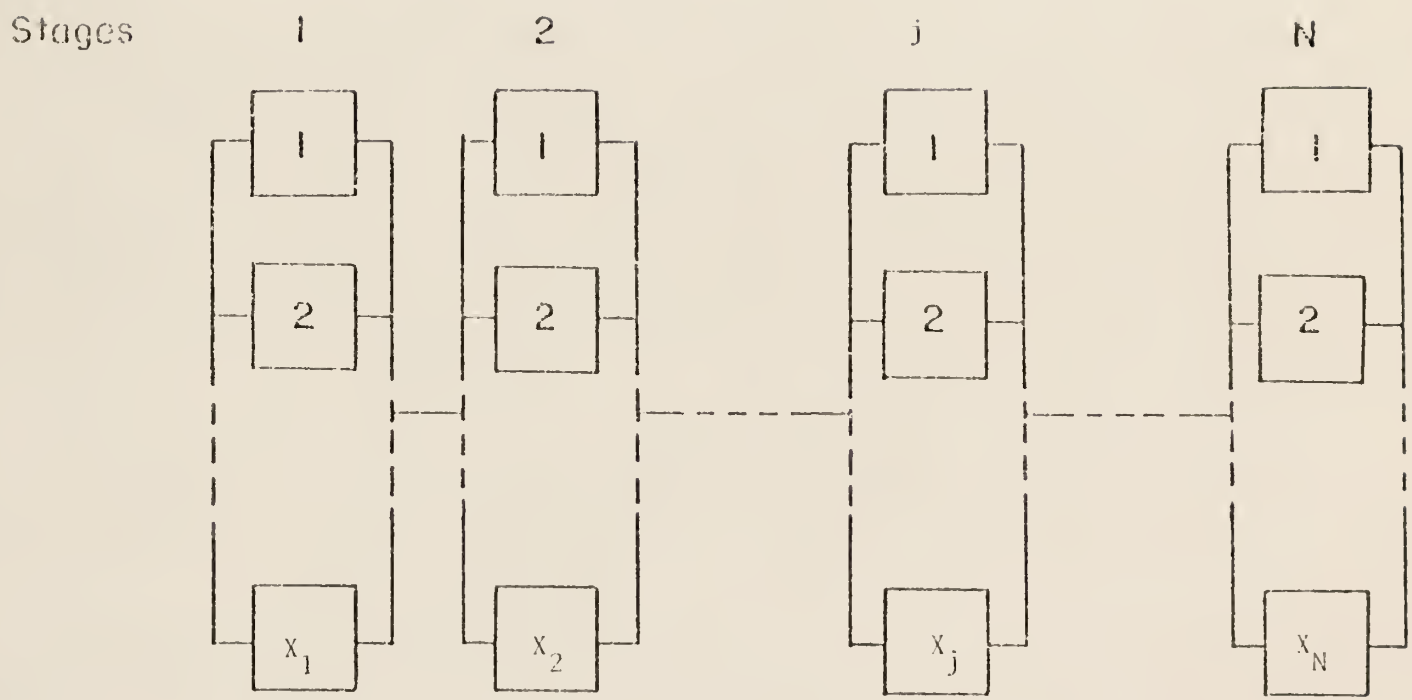


Fig. 1 A mixed system with N-stages in series where components are in parallel at each stage .

$$\sum_{j=1}^{N} C_j x_j$$

The net profit N_p for the entire system is the profit less the total cost, that is

$$N_p = PR_s - \sum_{j=1}^{N} C_j x_j \tag{2}$$

The optimal parallel redundancy configuration then is the one to find x_j, $j = 1, \ldots, N$, which maximizes the net system profit.

In this example, no constraints are imposed to the problem.

Consider a three stage process. The profit associated with the final product is P = 10 unit. The cost of each of the redundant components, C_j, the reliability of each of the components, R_j, are given as

	R_j	C_j
process 3	0.333	0.20
process 2	0.500	1.0
process 1	0.750	1.0

Example 2

Given a system reliability requirement $R_{s,\ min}$, the problem is to determine a least-cost allocation of an N-stage series system that yields $R_s \geq R_{s,min}$. The example is from Kettelle [1962]. As an example, consider the following four-stage system with a system reliability requirement of $R_{s,min} = 0.99$ and total cost less than $b_1 = 61$ [10]:

Stage	4	3	2	1
C_j	1.2	2.3	3.4	4.5
R_j	0.8	0.7	0.75	0.85

The problem is

Maximize

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

subject to

$$g_1 = \sum_{j=1}^{N} C_j x_j \leq b_1$$

and

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

Example 3

Consider the problem originally presented by Tillman-Liittschwager [1967], but the second constraint in the original problem is excluded. The five-stage problem is stated as

Maximize

$$R_s = \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=1}^{N} p_j (x_j)^2 \leq P$$

$$g_2 = \sum_{j=1}^{N} w_j x_j \exp (x_j/4) \leq W$$

where $x_j \geq 1$, $j = 1, 2, \ldots, N$, are integers.

The constants associated with this problem are:

j	R_j	p_j	P	w_j	W
1	0.80	1		7	
2	0.85	2		8	
3	0.90	3	110	8	200
4	0.65	4		6	
5	0.75	5		9	

Example 4

Let a_j represent the design alternatives available for the jth stage with a specified inherent component reliability, and $R_j'(x_j, a_j)$ denote the known reliability function of the jth stage when x_j identical components of design alternative a_j are used. For an N-stage series system, the problem [Fyffe, 1968] is [7]

Maximize

$$R_x = \prod_{j=1}^{N} R'_j \, (x_j, a_j)$$

subject to

$$g_1 = \sum_{j=1}^{N} g_{1j} \, (x_j, a_j) \leq C$$

$$g_2 = \sum_{j=1}^{N} g_{2j} \, (x_j, a_j) \leq W$$

where

$x_j \geq 1$, $a_j \geq 1$, $j=1,2,\ldots,N$ are all integers.

Example 5

Consider a five stage problem with three non-linear constraints [Tillman - Liittschwager, 1967]:

Maximize

$$R_s = \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=1}^{N} p_j (x_j)^2 \leq P$$

$$g_2 = \sum_{j=1}^{N} c_j \, (x_j + \exp\,(x_j/_4)) \leq C$$

$$g_3 = \sum_{j=1}^{N} w_j x_j \exp\,(x_j/_4) \leq W$$

where x_j, $j=1,2,\ldots,N$, are integers.

The objective function R_s can be approximated by

$$R_s \approx 1 - [(1 - R_1)^{x_1} + (1 - R_2)^{x_2} + (1 - R_3)^{x_3} + (1 - R_4)^{x_4} + (1 - R_5)^{x_5}],$$

where $(1 - R_1)^{x_1}$, $(1 - R_2)^{x_2}$, $(1 - R_3)^{x_3}$, $(1 - R_4)^{x_4}$, and $(1 - R_5)^{x_5}$ are stage unreliabilities and are represented by Q_1', Q_2', Q_3', Q_4', and Q_5', respectively.

The constants associated with the five-stage problem are given as

CONSTANTS ASSIGNED FOR FIVE-STAGE PROGRAM

j	R_j	p_j	P	c_j	C	w_j	W
1	0.80	1		7		7	
2	0.85	2		7		8	
3	0.90	3	110	5	175	8	200
4	0.65	4		9		6	
5	0.75	2		4		9	

Since by dynamic programming approach the number of constraints will result in the so-called dimensionality difficulty, three different approaches have been used for solving the problems. They are classified in Table 1.

A basic plain dynamic programming approach is used for a problem without constraints or with a single constraint. Whenever there are constraints in a problem, the computation required for solving the problem increases exponentially. The second method in Table 1 was originally introduced by

Table 1 Classification of Approaches

Methods	Application to Examples	References
Basic dynamic programming approach	Examples 1 and 2	1-4, 6, 7 11, 12, 13, 14, 15, 17
Dynamic programming approach using Lagrange Multipliers	Examples 3, 4, and 5	4, 7, 8, 14
Dynamic programming approach using the concept of dominating sequence	Examples 2, 3, 4, and 5	9, 10, 16 19

Bellman [1958] where Lagrange multiplier was used when two or more constraints were considered in a problem. By introducing the Lagrange multiplier; the dimensionality of the problem coming from the constraint is reduced.

If three constraints are considered in a problem, then two Lagrange multipliers have to be introduced, which gives us another problem for finding out the two optimal Lagrange multipliers. Therefore, the third approach is suggested which utilizes the concept of dominating sequence (see Table 1). Kettelle [1962] may be the first one to introduce the concept of dominating sequence to solve a single linear constraint problem. The approach is applicable to a 3-nonlinear-constraints problem. To use this approach we have to find both the upper bound and the lower bound of the number of components used at each stage to reduce the length of the dominating sequence. The detailed discussions with examples are shown in the following sections.

Example 1

The basic dynamic programming is used in solving Example 1 which is described in the Introduction section.

For the one-stage process, the optimal design is determined for the single decision variable, x_1, by the solution of

$$f_1(v_2) = \max_{x_1} \{Pv_1 - C_1x_1\} \tag{3}$$

for a spectrum of v_2 values, where $v_2 = \prod_{j=N+1}^{2} R_{sj}$ is the probability that all upstream stages work, $v_1 = v_2R_{s1} = v_2[1 - (1 - R_1)^{x_1}]$, and

$R_{s,N+1} = v_{N+1} = 1$. R_{sj} is the reliability of the jth stage with x_j parallel components, and R_j is the reliability of each component.

For the two-stage process, the optimal design is obtained by

$$f_2(v_3) = \max_{x_2} \{f_1(v_2) - C_2x_2\}$$

and for the j-stage process, the recursive functional equation is

$$f_j(v_{j+1}) = \max_{x_j} \{f_{j-1}(v_j) - C_jx_j\} \tag{4}$$

Now, if the optimal design for the subsystem including the stages, N-1, N-2, ..., and 1, is known, then stage N can be designed optimally solving the maximum problem for the single decision variable x_N, i.e.,

$$f_N(v_{N+1}) = \max_{x_N} \{f_{N-1}(v_N) - C_Nx_N\} \tag{5}$$

Substituting the constants into the equations, the recursive dynamic programming algorithms are

$$f_1(v_2) = \max_{x_1} \{10v_1 - 1.0x_1\} \tag{6}$$

$$f_2(v_3) = \max_{x_2} \{f_1(v_2) - 1.0x_2\} \tag{7}$$

$$f_3(v_4) = \max_{x_3} \{f_2(v_3) - 0.20x_3\} \qquad (8)$$

where

$$v_1 = v_2\{1 - (1 - R_1)^{x_1}\} \qquad (9)$$

$$v_2 = v_3\{1 - (1 - R_2)^{x_2}\} \qquad (10)$$

$$v_3 = v_4\{1 - (1 - R_3)^{x_3}\} \qquad (11)$$

$$v_4 = 1.0$$

The first maximum problem (Stage 1) is solved for the optimal x_1 for a spectrum of v_2 values. Since v_2 is the probability that all upstream stages work, v_2 takes a value between 0 and 1. Equations (6) and (9) are employed in a systematic search for x_1 which maximizes $\{10v_1 - 1.0x_1\}$ for an assigned v_2 value. Any one-dimensional search technique can be used; however, since x_1 usually takes a small integer value, a simple exhaust search is carried out and the results are presented in Table 1a. The optimal returns, $f_1(v_2)$, and the optimal parallel components, x_1, for v_2 = 1.0, 0.9, ..., 0.1 are presented in Table 2 and Fig. 2. Usually only Table 2 is presented as a dynamic programming table and detailed calculation presented in Table 1 is omitted.

Similarly equations (7) and (10) are employed in the systematic search for x_2 which maximizes $\{f_1(v_2) - C_2x_2\}$ for each value of v_3. The results are presented in Table 1b, and the optimal results in Table 2 and Fig. 2. In the process of calculation, the value of $f_1(v_2)$ is obtained by interpolation. For example, v_2 is given as 0.88 from equation (10) for v_3 = 1.0 and M_2 = 3. The value of $f_1(v_2)$ for v_2 = 0.88, which is used in equation (7), is determined by interpolation of $f_1(0.9)$ and $f_1(0.8)$ obtained in stage 1 optimization. Equations (8) and (11) are used in search of x_3 to maximize $\{f_2(v_3) - C_3x_3\}$ for S_4 = 1.0, since v_{N+1} is always 1. The results are presented in Table 1c, Table 2 and Fig. 2.

Starting with the three-stage process, its optimal system profit is $f_3 = 1.32$ units with the corresponding optimal values of $x_3 = 7$ and $v_3 = 0.94$. Entering stage 2 at $v_3 = 0.94$ gives $x_2 = 3$ and $v_2 = 0.82$, and entering stage 1 and $v_2 = 0.82$ gives $x_1 = 2$ and $v_1 = 0.77$. Thus the optimal parallel design consists of seven parallel components for stage 3, three parallel components for stage 2, and two parallel components for stage 1. This gives rise to the system reliability of 0.77 at a profit of 1.32 units. Without parallel redundancy the system has $P \prod_{j=3}^{1} R_j - \sum_{j=3}^{1} C_j x_j = 10(0.333 \times 0.80 \times 0.75) - (0.20 + 1.0 + 1.0) = -0.95$ profit.

Table 1a. Results of Stage 1

v_2	x_1	v_1	$Pv_1-C_1x_1$	$f_1(v_2)$ *
1.0	0	0.00	0.00	
1.0	1	0.75	6.50	
1.0	2	0.94	7.38	*
1.0	3	0.98	6.84	
1.0	4	1.00	5.96	
0.9	0	0.00	0.00	
0.9	1	0.68	5.75	
0.9	2	0.84	6.44	*
0.9	3	0.89	5.86	
0.9	4	0.90	4.96	
0.8	0	0.00	0.00	
0.8	1	0.60	5.00	
0.8	2	0.75	5.50	*
0.8	3	0.79	4.88	
0.8	4	0.80	3.97	
0.7	0	0.00	0.00	
0.7	1	0.53	4.25	
0.7	2	0.66	4.56	*
0.7	3	0.69	3.89	
0.7	4	0.70	2.97	
0.6	0	0.00	0.00	
0.6	1	0.45	3.50	
0.6	2	0.56	3.63	*
0.6	3	0.59	2.91	
0.6	4	0.60	1.98	
0.5	0	0.00	0.00	
0.5	1	0.38	2.75	*
0.5	2	0.47	2.69	
0.5	3	0.49	1.92	
0.4	0	0.00	0.00	
0.4	1	0.30	2.00	*
0.4	2	0.38	1.75	
0.4	3	0.39	0.94	
0.3	0	0.00	0.00	
0.3	1	0.23	1.25	*
0.3	2	0.28	0.81	
0.3	3	0.30	-0.05	
0.2	0	0.00	0.00	
0.2	1	0.15	0.50	*
0.2	2	0.19	-0.13	
0.2	3	0.20	-1.03	
0.1	0	0.00	0.00	*
0.1	1	0.07	-0.25	
0.1	2	0.09	-1.06	

Table 1b. Results of stage 2 (and stage 1)

v_3	x_2	v_2	$f_1(v_2)$	C_2x_2	$f_1(v_2)-C_2x_2$	$f_2(v_3)$
1.0	0	0.00	-0.00	0.00	0.00	
1.0	1	0.50	2.75	1.00	1.75	
1.0	2	0.75	5.03	2.00	3.03	
1.0	3	0.88	6.20	3.00	3.20	*
1.0	4	0.94	6.79	4.00	2.79	
1.0	5	0.97	7.08	5.00	2.08	
0.9	0	0.00	-0.00	0.00	0.00	
0.9	1	0.45	2.38	1.00	1.38	
0.9	2	0.68	4.33	2.00	2.33	
0.9	3	0.79	5.38	3.00	2.38	*
0.9	4	0.84	5.91	4.00	1.91	
0.9	5	0.87	6.17	5.00	1.17	
0.8	0	0.00	-0.00	0.00	0.00	
0.8	1	0.40	2.00	1.00	1.00	
0.8	2	0.60	3.63	2.00	1.63	*
0.8	3	0.70	4.56	3.00	1.56	
0.8	4	0.75	5.03	4.00	1.03	
0.7	0	0.00	-0.00	0.00	0.00	
0.7	1	0.35	1.63	1.00	0.63	
0.7	2	0.53	2.97	2.00	0.97	*
0.7	3	0.61	3.74	3.00	0.74	
0.7	4	0.66	4.15	4.00	0.15	
0.6	0	0.00	-0.00	0.00	0.00	
0.6	1	0.30	1.25	1.00	0.25	
0.6	2	0.45	2.38	2.00	0.38	*
0.6	3	0.53	2.97	3.00	-0.03	
0.6	4	0.56	3.30	4.00	-0.70	
0.5	0	0.00	-0.00	0.00	0.00	*
0.5	1	0.25	0.88	1.00	-0.13	
0.5	2	0.38	1.81	2.00	-0.19	
0.4	0	0.00	-0.00	0.00	0.00	*
0.4	1	0.20	0.50	1.00	-0.50	
0.4	2	0.30	1.25	2.00	-0.75	
0.3	0	0.00	-0.00	0.00	0.00	*
0.3	1	0.15	0.25	1.00	-0.75	
0.3	2	0.23	0.69	2.00	-1.31	
0.2	0	0.00	-0.00	0.00	0.00	*
0.2	1	0.10	0.00	1.00	-1.00	
0.2	2	0.15	0.25	2.00	-1.75	
0.1	0	0.00	-0.00	0.00	0.00	*
0.1	1	0.05	-0.00	1.00	-1.00	
0.1	2	0.07	-0.00	2.00	-2.00	

Table 1c. Results of stage 3 (and stage 2 and stage 1)

v_4	x_3	v_3	$f_2(v_3)$	C_3x_3	$f_2(v_3)-C_3x_3$	$f_3(v_4)$ *
1.0	0	0.00	-0.00	0.00	0.00	
1.0	1	0.33	0.00	0.20	-0.20	
1.0	2	0.56	0.21	0.40	-0.19	
1.0	3	0.70	0.99	0.60	0.39	
1.0	4	0.80	1.64	0.80	0.84	
1.0	5	0.87	2.14	1.00	1.14	
1.0	6	0.91	2.48	1.20	1.28	
1.0	7	0.94	2.72	1.40	1.32	*
1.0	8	0.96	2.88	1.60	1.28	
1.0	9	0.97	2.99	1.80	1.19	

Table 2. The Dynamic Programming Table

stage 3 (and stage 2 and stage 1)

v_4	$f_3(v_4)$	x_3	v_3	$f_2(v_3)$
1.0	1.32	7	0.94	2.72

stage 2 (and stage 1)

v_3	$f_2(v_3)$	x_2	v_2	$f_1(v_2)$
1.0	3.20	3	0.88	6.20
0.9	2.38	3	0.79	5.38
0.8	1.63	2	0.60	3.63
0.7	0.97	2	0.53	2.97
0.6	0.38	2	0.45	2.38
0.5	0.	0	0.	0.
0.4	0.	0	0.	0.
0.3	0.	0	0.	0.
0.2	0.	0	0.	0.
0.1	0.	0	0.	0.

stage 1

v_2	$f_1(v_2)$	x_1
1.0	7.38	2
0.9	6.44	2
0.8	5.50	2
0.7	4.56	2
0.6	3.63	2
0.5	2.75	1
0.4	2.00	1
0.3	1.25	1
0.2	0.50	1
0.1	0.	0

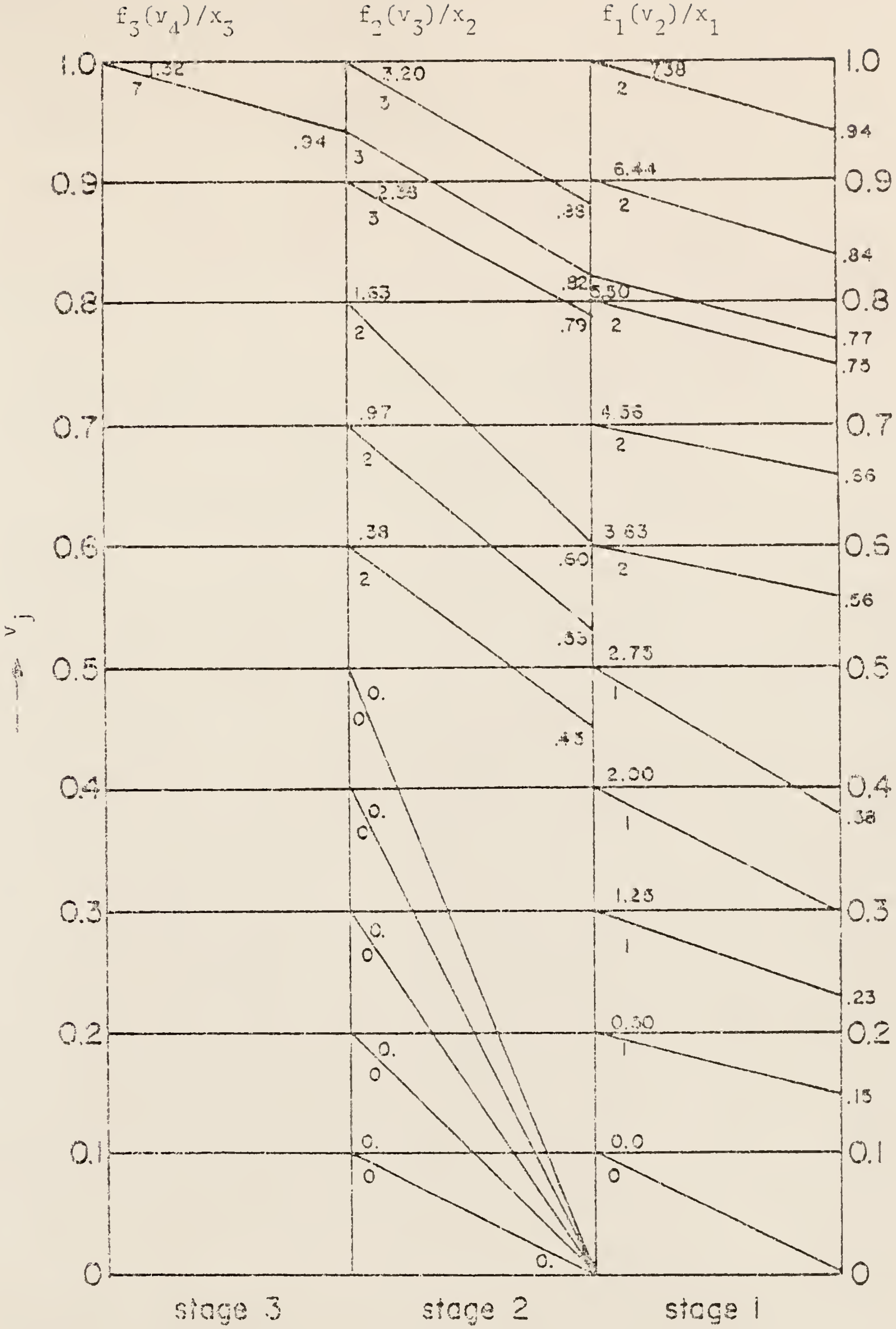


Fig. 2 Results for Example 1.

Example 2

The problem with single constraint of Example 2 in the Introduction to this chapter is considered.

The recursive formula of the problem in the basic dynamic programming algorithm then is formulated:

$$f_1(b) = \max_{x_1^l \le x_1 \le x_1^u} \left[R_1'(x_1)\right]$$

$$f_2(b) = \max_{x_2^l \le x_2 \le x_2^u} [R_2'(x_2)\; f_1(b - g_{12}(x_2))]$$

$$\vdots$$

$$f_N(b) = \max_{x_N^l \le x_N \le x_N^u} [R_N'(x_N)\; f_{N-1}(b - g_{1N}(x_N))] \tag{12}$$

where

$$R_j'(x_j) = 1 - (1 - R_j)^{x_j},\; j = 1, 2, \ldots, N.$$

x_j^l, $j = 1, 2, \ldots, N$, is the minimum integer number used at each stage. It is usually that $x_j^l = 1$, $j = 1, 2, \ldots, N$ if no restriction on the minimum system reliability is imposed to the problem. x_j^u, $j = 1, 2, \ldots, N$, is the maximum integer number used at each stage such that $\sum_{\substack{p=1 \\ p \neq j}}^{N} g_{1p}(x_j^l) + g_{1j}(x_j) \le b_1$.

This example has been solved by Kettelle [1962] by dynamic programming algorithm using the concept of dominating sequence. Now, it is solved by the basic dynamic programming approach.

Since the goal of system reliability is 0.99, the minimum reliability at each stage at least is also 0.99. It is required to determine the minimum number of components, x_j^l, used at each stage to attain the stage reliability goal of 0.99. Since the component reliability is 0.85 at stage 1, two components in parallel (one redundancy) give the stage reliability of 0.9775, and three components in parallel (two redundancies) give the stage reliability of 0.9966 which is greater than 0.99. Therefore, three components are the minimum to be used at stage 1. Similarly, the minimum requirement for stages 2, 3 and 4 are determined to be $(x_2^{\ell}, x_3^{\ell}, x_4^{\ell}) = (4, 4, 3)$, respectively.

Since the maximum of R_s over the feasible region will depend upon the number of stages, N, and the available resource, b_1, we denote by $f_N(b_1)$ the maximum of R_N. That is,

$$f_N(b_1) = \max_{x_N, x_{N-1}, \ldots, x_1} \left[\prod_{j=N}^{1} R_j'(x_j)\right] \tag{13}$$

where x_j, j = 1, 2, ..., N are positive integers satisfying the constraint:

$$\sum_{j=1}^{N} g_{1j}(x_j) \leq b_1 \tag{14}$$

For the one-stage process, the optimal design is determined for the single decision variable, x_1, by the solution of

$$f_1(b) = \max_{x_1^{\ell} \leq x_1 \leq x_1^{u}} R_1'(x_1) \tag{15}$$

where, $x_1^{\ell} = 3$, and the upper bound used in stage 1, x_1^u, is restricted by the cost constraint. The first maximum problem (Stage 1) is solved for the optimal x_1 for a spectrum of b values. The spectrum of b is determined from the consumed resource 39.9, of basic allocation $(x_4, x_3, x_2, x_1) = (3, 4, 4, 3)$, to the total available resource, 61.0. Thus, for each value of b between 39.9 and 61.0, we will find an optimal allocation for x_1 when all the upstream stage allocation are fixed by $(x_4, x_3, x_2) = (3, 4, 4)$. The optimal allocation for x_1 is shown in Table 3a. All possible b values should be searched exhaustively to find the optimal x_1. Since (x_4, x_3, x_2) are fixed, optimal x_1 is 3 for b in the region, $39.90 \leq b < 44.40$, which gives $f_1(b) = 0.9768$. For b in the region of

$44.40 \leq b < 48.90$, the optimal x_1 becomes 4 and $f_1(b)$ is 0.9796. Similarly, $x_1 = 5$ for $48.90 \leq b < 53.40$, and $f_1(b)$ is 0.9800; $x_1 = 6$ for $53.40 \leq b < 57.90$ and $f_1(b)$ is 0.9801; and $x_1 = 7$ for $57.90 \leq b \leq 61.00$ and $f_1(b)$ is 0.9802.

For all possible b we have searched the optimal x_1's when stages 4, 3, and 2 are fixed at x_4^l, x_3^l, x_2^l. The next step is to search for the optimal combinations of Stage 2 and Stage 1 when Stage 4 and Stage 3 are fixed at the minimum required components, x_4^l, and x_3^l. It is still necessary to consider all possible b between 39.90 and 61.0. For convenience, the maximazation will be carried out for b from 40.0 to 61.0 with a discrete spectrum. The difference between every two near searching point is one. Table 3a then is used to construct Table 3b.

In Table 3b, $(x_4, x_3) = (3, 4)$ is always fixed. An optimal (x_2, x_1) is searched for the maximum system reliability for the corresponding value of b given. For example, if $b = 40$, from Table 3a, x_1 could be 3; with $(x_4, x_3, x_1) = (3, 4, 3)$, the optimal x_2 is 4; then the system reliability of $(x_4, x_3, x_2, x_1) = (3, 4, 4, 3)$ is 0.9768. Similarly, when b = 41, 42, and 43, the optimal allocation for (x_2, x_1) are (4, 3). When b is increased to 44, from Table 3a, x_1 is still 3, but x_2 can be 4 or 5, although $x_2 = 5$ gives the greater system reliability of 0.9797. Therefore, $f_2(44) = 0.9797$. When b is increased to 45, then from Table 3a, x_1 can be either 3 or 4. When $x_1 = 3$, we search for the optimal x_2 to be 5 and $R_s = 0.9797$; when $x_1 = 4$, we search for the optimal x_2 to be 4 and $R_s = 0.9796$, the optimal allocation for b = 45 are $(x_4, x_3, x_2, x_1) = (3, 4, 5, 3)$. The computational results presented in Table 3b are carried out similarly. For another example, for b = 54, from Table 3a, x_1 can be 3, 4, 5, or 6 as (x_4, x_3) is fixed as (3, 4). For $x_1 = 3$, the maximum system reliability is obtained when $x_2 = 8$. Similarly for $x_1 = 4$, the maximum system reliability is given at $x_2 = 6$; for $x_1 = 5$ at $x_2 = 5$; for $x_1 = 6$ at $x_2 = 4$. The optimum result for b = 54, $f_2(54)$, is the maximum system reliability among (x_2, x_1) = (8, 3), (6, 4), (5, 5), and (4, 6), which is $R_s = 0.9832$ and (x_2^*, x_1^*) = (6, 4). Usually the computational results for stage 2 (and stage 1) presented in Table 3b are not presented and only the dynamic programming table of Table 4 is presented.

Similarly, we can construct Table 3c for all possible b values and for fixed $x_4 = 3$. For each b value, a systematic search procedure is carried out by looking back the optimal allocation of (x_2, x_1) shown in Table 4 for stage 2. For example, when b = 52 is of interest, from Table 4, the optimal allocation of (x_2, x_1), for $b \leq 52$ are:

b	optimum $(x_2$	$x_1)$
40	4	3
41	4	3
42	4	3
43	4	3
44	5	3
45	5	3
46	5	3
47	6	3
48	5	4
49	5	4
50	5	4
51	5	4
52	6	4

Therefore, the optimal allocation for (x_2, x_1) can only be one of the following: (4, 3), (5, 3), (6, 3), (5, 4), (6, 4). Since x_4 is fixed, for (x_2, x_1)

= (4, 3) we find the optimal is $x_3 = 9$ with system reliability, $R_s = 0.9848$; for $(x_2, x_1) = (5, 3)$, the optimal is $x_3 = 7$ and $R_s = 0.9875$; for $(x_2, x_1) = (6, 3)$ the optimal is $x_3 = 6$, and $R_s = 0.9877$; for $(x_2, x_1) = (5, 4)$, the optimal is $x_3 = 5$ and $R_s = 0.9881$; and for $(x_2, x_1) = (6, 4)$ the optimal is $x_3 = 4$ and $R_s = 0.9832$. Among these system reliabilities, 0.9881 is the largest one, hence the allocation of $(x_4, x_3, x_2, x_1) = (3, 5, 5, 4)$ is the optimal one for b = 52. The optimum results for Stage 3 (and Stage 2 and Stage 1) are presented in Table 4.

Finally we can construct Table 3d for b = 61, which is the total allowable resource. For b = 61, from Table 4 for Stage 3, all the optimal allocation of (x_3, x_2, x_1) for $b \leq 61$ are: (4, 4, 3), (5, 4, 3), (6, 4, 3), (5, 5, 5), (6, 5, 3), (5, 5, 4), (6, 5, 4), (7, 5, 4), (6, 6, 4), (7, 6, 4), and (8, 6, 4). For each allocation, the optimum x_4 (the maximum allowable x_4 to give the maximum system reliability) is calculated. Among all these system reliabilities the optimal system reliability for this problem as shown in Table 3d is $(x_4, x_3, x_2, x_1) = (5, 7, 6, 4)$ which gives the largest system reliability, $R_s = 0.99871$. The dynamic programming table for this problem is given in Table 4.

Table 3a The dynamic programming table of Example 2 for Stage 1

b	x_4^ℓ	x_3^ℓ	x_2^ℓ	x_1	R_s	$f_1(b)$
39.90 - 44.39	3	4	4	3	0.9768	*
44.40 - 48.89	3	4	4	4	0.9796	*
48.90 - 53.39	3	4	4	5	0.9800	*
53.40 - 57.89	3	4	4	6	0.9807	*
57.90 - 61.00	3	4	4	7	0.9802	*

Table 3b Computational results of Example 2 for Stage 2 (and Stage 1)

b	x_4^l	x_3^l	x_2	x_1	R_s	$f_2(b)$
40	3	4	4	3	0.9768	*
41	3	4	4	3	0.9768	*
42	3	4	4	3	0.9768	*
43	3	4	4	3	0.9768	*
44	3	4	4	3	0.9768	
	3	4	5	3	0.9797	*
45	3	4	5	3	0.9797	*
	3	4	4	4	0.9796	
46	3	4	5	3	0.9797	*
	3	4	4	4	0.9796	
47	3	4	6	3	0.9804	*
	3	4	4	4	0.9796	
48	3	4	6	3	0.9804	
	3	4	5	4	0.9825	*
49	3	4	6	3	0.9804	
	3	4	5	4	0.9825	*
	3	4	4	5	0.9800	
50	3	4	6	3	0.9804	
	3	4	5	4	0.9825	*
	3	4	4	5	0.9800	
51	3	4	7	3	0.9806	
	3	4	5	4	0.9825	*
	3	4	4	5	0.9800	
52	3	4	7	3	0.9806	
	3	4	6	4	0.9832	*
	3	4	4	5	0.9800	
53	3	4	7	3	0.9806	
	3	4	6	4	0.9832	*
	3	4	5	5	0.9829	
54	3	4	8	3	0.9806	
	3	4	6	4	0.9832	*
	3	4	5	5	0.9829	
	3	4	4	6	0.9801	

b	x_4^ℓ	x_3^ℓ	x_2	x_1	R_s	$f_2(b)$
55	3	4	8	3	0.9806	
	3	4	7	4	0.9834	*
	3	4	5	5	0.9829	
	3	4	4	6	0.9801	
56	3	4	8	3	0.9806	
	3	4	7	4	0.9834	
	3	4	6	5	0.9836	*
	3	4	4	6	0.9801	
57	3	4	9	3	0.9806	
	3	4	7	4	0.9834	
	3	4	6	5	0.9836	*
	3	4	5	6	0.9830	
58	3	4	9	3	0.9806	
	3	4	8	4	0.9835	
	3	4	6	5	0.9836	*
	3	4	5	6	0.9830	
	3	4	4	7	0.9802	
59	3	4	9	3	0.9806	
	3	4	8	4	0.9835	
	3	4	6	5	0.9836	*
	3	4	5	6	0.9830	
	3	4	4	7	0.9802	
60	3	4	9	3	0.9806	
	3	4	8	4	0.9835	
	3	4	7	5	0.9838	*
	3	4	5	6	0.9830	
	3	4	4	7	0.9802	
61	3	4	10	3	0.9806	
	3	4	8	4	0.9835	
	3	4	7	5	0.9838	*
	3	4	6	6	0.9837	
	3	4	4	7	0.9802	

Table 3c Computational results of Example 2 for Stage 3 (and Stage 2 and Stage 1)

b	x_4^2	x_3	x_2	x_1	R_s	$f_3(b)$
40	3	4	4	3	0.9768	*
41	3	4	4	3	0.9768	*
42	3	4	4	3	0.9768	*
43	3	5	4	3	0.9824	*
44	3	5	4	3	0.9824	*
	3	4	5	3	0.9797	
45	3	6	4	3	0.9841	*
	3	4	5	3	0.9797	
46	3	6	4	3	0.9841	
	3	5	5	3	0.9853	*
47	3	7	4	3	0.9846	
	3	5	5	3	0.9853	*
	3	4	6	3	0.9804	
48	3	7	4	3	0.9846	
	3	6	5	3	0.9870	*
	3	4	6	3	0.9804	
	3	4	5	4	0.9825	
49	3	7	4	3	0.9846	
	3	6	5	3	0.9870	*
	3	5	6	3	0.9860	
	3	4	5	4	0.9825	
50	3	8	4	3	0.9847	
	3	6	5	3	0.9870	*
	3	5	6	3	0.9860	
	3	4	5	4	0.9825	
51	3	8	4	3	0.9847	
	3	7	5	3	0.9875	
	3	5	6	3	0.9860	
	3	5	5	4	0.9881	*
52	3	9	4	3	0.9848	
	3	7	5	3	0.9875	
	3	6	6	3	0.9877	
	3	5	5	4	0.9881	*
	3	4	6	4	0.9852	

b	x_4^{ℓ}	x_3	x_2	x_1	R_s	$f_3(b)$
53	3	9	4	3	0.9848	
	3	8	5	3	0.9876	
	3	6	6	3	0.9877	
	3	6	5	4	0.9898	*
	3	4	6	4	0.9832	
54	3	10	4	3	0.9848	
	3	8	5	3	0.9876	
	3	7	6	3	0.9882	
	3	6	5	4	0.9898	*
	3	5	6	4	0.9888	
55	3	10	4	3	0.9848	
	3	9	5	3	0.9877	
	3	7	6	3	0.9882	
	3	7	5	4	0.9903	*
	3	5	6	4	0.9888	
	3	4	7	4	0.9834	
56	3	11	4	3		
	3	9	5	3	0.9877	
	3	8	6	3	0.9883	
	3	7	5	4	0.9903	
	3	6	6	4	0.9905	*
	3	4	6	5	0.9836	
57	3	11	4	3		
	3	9	5	3	0.9877	
	3	8	6	3	0.9883	
	3	8	5	4	0.99046	
	3	6	6	4	0.99053	*
	3	4	6	5	0.9836	
58	3	12	4	3		
	3	10	5	3	0.9877	
	3	8	6	3	0.9883	
	3	8	5	4	0.99046	
	3	6	6	4	0.99053	*
	3	5	6	5	0.9893	
59	3	12	4	3		
	3	10	5	3	0.9877	
	3	9	6	3	0.9884	
	3	8	5	4	0.99046	
	3	7	6	4	0.9910	*
	3	5	6	5	0.9893	
60	3	13	4	3	0.9849	
	3	11	5	3	0.9877	

b	x_4^0	x_3	x_2	x_1	R_s	$f_3(b)$
60	3	9	6	3	0.9884	
	3	9	5	4	0.9905	
	3	7	6	4	0.9910	*
	3	5	6	5	0.9893	
	3	4	7	5	0.9838	
61	3	13	4	3	0.9849	
	3	11	5	3	0.9877	
	3	10	6	3	0.9884	
	3	9	5	4	0.9905	
	3	8	6	4	0.9912	*
	3	6	6	5	0.9910	
	3	4	7	5	0.9838	

Table 3d Computational results of Example 2 for Stage 4 (and Stage 3, Stage 2 and Stage 1)

b	x_4	x_3	x_2	x_1	R_s	$f_4(b)$
61	20	4	4	3	0.9848	
	18	5	4	3	0.9904	
	16	6	4	3	0.9921	
	15	5	5	3	0.9933	
	14	7	4	3	0.9926	
	13	6	5	3	0.9951	
	11	5	5	4	0.99609	
	10	6	5	4	0.99779	
	8	7	5	4	0.99830	
	7	6	6	4	0.99851	
	5	7	6	4	0.99871	*
	3	8	6	4	0.99119	

Table 4 The dynamic programming table of Example 2

Stage 4 (and Stage 3, Stage 2 and Stage 1)

b	x_4	x_3	x_2	x_1	$f_4(b)$
61	5	7	6	4	0.99871

Stage 3 (and stage 2 and stage 1)

b	x_4^{2}	x_3	x_2	x_1	$f_3(b)$
40	3	4	4	3	0.9768
41	3	4	4	3	0.9768
42	3	4	4	3	0.9768
43	3	5	4	3	0.9824
44	3	5	4	3	0.9824
45	3	6	4	3	0.9841
46	3	5	5	3	0.9853
47	3	5	5	3	0.9855
48	3	6	5	3	0.9870
49	3	6	5	3	0.9870
50	3	6	5	3	0.9870
51	3	5	5	4	0.9881
52	3	5	5	4	0.9881
53	3	6	5	4	0.9898
54	3	6	5	4	0.9898
55	3	7	5	4	0.9903
56	3	6	6	4	0.9905
57	3	6	6	4	0.9905
58	3	6	6	4	0.9905
59	3	7	6	4	0.9910
60	3	7	6	4	0.9910
61	3	8	6	4	0.9912

Stage 2 (and stage 1)

b	x_4^{ℓ}	x_3^{ℓ}	x_2	x_1	$f_2(b)$
40	3	4	4	3	0.9768
41	3	4	4	3	0.9768
42	3	4	4	3	0.9768
43	3	4	4	3	0.9768
44	3	4	5	3	0.9797
45	3	4	5	3	0.9797
46	3	4	5	3	0.9797
47	3	4	6	3	0.9804
48	3	4	5	4	0.9825
49	3	4	5	4	0.9825
50	3	4	5	4	0.9825
51	3	4	5	4	0.9825
52	3	4	6	4	0.9832
53	3	4	6	4	0.9831
54	3	4	6	4	0.9831
55	3	4	7	4	0.9834
56	3	4	6	5	0.9836
57	3	4	6	5	0.9836
58	3	4	6	5	0.9836
59	3	4	6	5	0.9836
60	3	4	7	5	0.9838
61	3	4	7	5	0.9838

Table 4 (continued)

Stage 1

b	x_4^{ℓ}	x_3^{ℓ}	x_2^{ℓ}	x_1	$f_1(b)$
39.90 - 44.39	3	4	4	3	0.9768
44.40 - 48.89	3	4	4	4	0.9796
48.90 - 53.39	3	4	4	5	0.9800
53.40 - 57.89	3	4	4	6	0.9807
57.90 - 61.00	3	4	4	7	0.9802

3. Dynamic Programming Approach Using Lagrange Multipliers

Formulation of the Problem

If multiple constraint functions are imposed to restrict the objective function, then the Lagrange multipliers may be introduced to eliminate some constraints and hence reduce the dimension of the problem.

In section 2, we have formulated the single constraint problem solved by the basic dynamic programming approach. Now, if the second constraint,

$$\sum_{j=1}^{N} g_{2j}(x_j) \leq b_2 \tag{16}$$

is also imposed to the problem, we have to consider the sequence of functions defined by the relation

$$f_1(b_1, b_2) = \max_{1 \leq x_1 \leq x_1^u} R_1'(x_1) \tag{17}$$

$$f_2(b_1, b_2) = \max_{1 \leq x_2 \leq x_2^u} [R_2'(x_2) \cdot f_1(b_1 - g_{12}(x_2), b_2 - g_{22}(x_2))]$$

$$\vdots$$

$$f_N(b_1, b_2) = \max_{1 \leq x_N \leq x_N^u} [R_N'(x_N) \cdot f_{N-1}(b_1 - g_{1N}(x_N), b_2 - g_{2N}(x_N))]$$

where x_j^u , $j=1, 2, \ldots, N$ is the minimum integer between $(x_j^u)^1$ and $(x_j^u)^2$; and $(x_j^u)^1$ is the maximum integer satisfying

$$\sum_{\substack{p=1 \\ p \neq j}}^{N} g_{1p}(1) + g_{ij}(x_j) \leq b_1$$

and $(x_j^u)^2$ is the maximum integer satisfying

$$\sum_{\substack{p=1 \\ p \neq j}}^{N} g_{2p}(1) + g_{2j}(x_j) \leq b_2$$

The recursive formula for a two-constraints problem is basically following the same approach as those for a one-constraint problem. Although the formula is simple and very straightforward, it involves sequences of functions of two variables which will require a large memory capacity and are quite time-consuming. Therefore, it is not very desirable from the computational standpoint.

An alternative method to solve this problem is by introducing a Lagrange multiplier, λ, as a penalty term. The problem now is stated as

Maximize

$$\prod_{j=1}^{N} R_j' (x_j) \left[e^{-\lambda \sum_{j=1}^{N} g_{2j} (x_j)} \right] \tag{18}$$

subject to

$$\sum_{j=1}^{N} g_{1j} (x_j) \leq b_1$$

The Lagrange multiplier, λ, is to be chosen so that the constraint in eq. (16) is as nearly as possible an equality. Now the problem becomes the sequence of functions of one variable which has the following recursive formula.

$$f_1(b) = \max_{x_1^{\ell} \leq x_1 \leq x_1^u} [R_1' (x_1) \exp (-\lambda g_{21}(x_1))]$$

$$f_2(b) = \max_{x_2^{\ell} \leq x_2 \leq x_2^u} [R_2' (x_2) f_1(b_1 - g_{12}(x_2)) \exp(-\lambda g_{22}(x_2))]$$

$$\vdots$$

$$f_N(b) = \max_{x_N^{\ell} \leq x_N \leq x_N^u} [R_N' (x_N) f_{N-1} (b_1 - g_{1N}(x_N)) \exp(-\lambda g_{2N}(x_N))]$$

where

$$R'_j(x_j) = 1 - (1 - R_j)^{x_j}, \; j=1, 2, \ldots, N$$

x_j^{ℓ}, j=1, 2, ..., N, is the minimum integer used at each stage and x_j^u, j=1, 2, ..., N, is the maximum integer used at each stage such that

$$\sum_{\substack{p=1 \\ p \neq j}}^{N} g_{1p}(x_j^{\ell}) + g_{1j}(x_j) \leq b_1$$

λ is to be chosen so that $\sum_{j=1}^{N} g_{2j}(x_j)$ is as close to b_2 as possible. For a fixed value of λ, the maximum system reliability is obtained; that is,

$$R_s = f_N(b_1) \exp\left[-\lambda \sum_{j=1}^{N} g_{2j}(x_j)\right]$$

a one-dimensional search for λ should be carried out to find the optimal solution for R_s.

A Numerical Example

Example 3

To solve this example, we first find the lower bound of components to be used at each stage.

By an enumeration method as shown in Table 5, redundancies are allocated stage by stage until one of the constraints is exceeded. The basic system configuration for calculating the lower bound, is assumed to be the one before exceeding one of the constraints, i.e., (3, 3, 3, 2, 2) for this numerical example. The system reliability corresponding to this configuration, $R_s(\bar{x})$, is 0.8125. This is not, however, an optimal solution. The optimal system reliability should be equal to or greater than this value. Therefore, we assume that the lower bound of stage reliability is 0.8125 and calculate the corresponding lower bound of stage components.

Table 5 The allocation of elements stage by stage until any one constraint is violated.

Element Allocations: stages 1	2	3	4	5	Resources used $\sum_{j=1}^{5} g_{1j}$	$\sum_{j=1}^{5} g_{2j}$
1	1	1	1	1	12	48.79
2	1	1	1	1	15	62.83
2	2	1	1	1	21	78.99
2	2	2	1	1	30	95.10
2	2	2	2	1	42	107.19
2	2	2	2	2	48	125.30
3	2	2	2	2	53	146.67
3	3	2	2	2	63	171.11
3	3	3	2	2	78	195.53
3	3	3	3	2	98	213.86

That is, for j=1,

$$1 - (1 - 0.80)^{x_1} \geq 0.8125$$

which gives $x_1 \geq 2.31$, say, $x_1^{\ell} = 2$.

Similarly we obtain $x_2^{\ell} = 2$, $x_3^{\ell} = 1$, $x_4^{\ell} = 3$, and $x_5^{\ell} = 2$.

The recursive equations modified by using the Lagrange multiplier are

$$f_1(b) = \max_{x_1^{\ell} \leq x_1 \leq x_1^{u}} \{(1 - Q_1^{x_1}) \exp [-\lambda(w_1 x_1 e^{x_{1/4}})]\} \tag{19}$$

$$f_2(b) = \max_{x_2^{\ell} \leq x_2 \leq x_2^{u}} (1 - Q_2^{x_2}) \exp [-\lambda(w_2 x_2 e^{x_{2/4}})] \; f_1(b - p_2 x_2^{2}) \tag{20}$$

$$f_3(b) = \max_{x_3^{\ell} \leq x_3 \leq x_3^{u}} (1 - Q_3^{x_3}) \exp [-\lambda(w_3 x_3 e^{x_{3/4}})] \; f_2(b - p_3 x_3^{2}) \tag{21}$$

$$f_4(b) = \max_{x_4^{\ell} \leq x_4 \leq x_4^{u}} (1 - Q_4^{x_4}) \exp [-\lambda(w_4 x_4 e^{x_{4/4}})] \; f_3(b - p_4 x_4^{2}) \tag{22}$$

$$f_5(b) = \max_{x_5^{\ell} \leq x_5 \leq x_5^{u}} (1 - Q_5^{x_5}) \exp [-\lambda(w_5 x_5 e^{x_{5/4}}(] \; f_4(b - p_5 x_5^{2}) \tag{23}$$

where $Q_j = (1 - R_j)$, j=1,2,...,5.

The quantity λ is to be determined so that

$$g_2 = \sum_{j=1}^{N} w_j x_j \exp (x_{j/4}) \approx W$$

To solve this example, λ should be assigned, say $\lambda = 0.001$. Since the objective over the feasible region depends upon the number of stages, N, the available resource, b_1, and the Lagrange multiplier, λ, we denote by $f_N(b_1)$ the maximization of the objective. That is

$$f_N(b_1) = \max_{x_N, x_{N-1}, \ldots, x_1} \left[\prod_{j=N}^{1} R_j'(x_j) \exp(-\lambda \sum_{j=1}^{n} g_{2j}(x_j)) \right]$$

where x_j, $j=1,2,\ldots,N$ are positive integers satisfying the constraint

$$\sum_{j=1}^{n} g_{1j}(x_j) \leq b_1$$

For the one-stage process, the optimal design is determined for the single decision variable, x_1, by the solution of

$$f_1(b) = \max_{x_1^{\ell} \leq x_1 \leq x_1^{u}} R_1'(x_1) \exp(-\lambda g_{2j}(x_1))$$

where, $x_1^{\ell} = 2$, and the upper bound used in stage 1, x_1^{u}, is restricted by the constraint. The spectrum of b is determined from the consumed resource, 59.0, for the basic allocation $(x_5^{\ell}, x_4^{\ell}, x_3^{\ell}, x_2^{\ell}, x_1^{\ell}) = (2,3,1,2,2)$ to the total available resource, 110.0. When b increases, stage redundancy, x_1-1, stage reliability, $R_1'(x_1)$, and stage cost, $g_{21}(x_1)$, will increase, but the penalty term, $\exp(-\lambda g_{21}(x_1))$ will decrease. Since $f_1(b)$ is a maximization of the product of $R_1'(x_1)$ and $\exp(-\lambda g_{21}(x_1))$, $f_1(b)$ is not a monotonic increasing function of b. In other words, the increasing in b allows us to add more components in stage 1 but the configuration from these more redundancy may not give us a optimal return value. When the upstream stages are fixed by $(x_5^{\ell}, x_4^{\ell}, x_3^{\ell}, x_2^{\ell}) = (2,3,1,2)$, the optimal allocation for x_1 is obtained as shown in Table 6a. All possible b values should be searched exhaustively to find the optimal x_1. Since (x_5, x_4, x_3, x_2) are fixed, x_1 is 2 for b in the region of $59.0 \leq b < 64.0$ which gives the functional value of 0.66710. For b in the region of $64.0 \leq b < 71.0$, the optimal x_1 becomes 3 and $f_1(b)$ is 0.67476. For b in the region of $71.0 \leq b < 80.0$, we may allocate 4 components for x_1, but this gives the functional value of 0.65860 which is smaller than the functional value of 0.67476 as $x_1 = 3$. Therefore, $x_1 = 3$ is the optimal one for $71.0 \leq b < 80.0$. Similarly, for $b \geq 80.0$, we may allocate $x_1 = 3, 4, 5, 6,$ or 7; however, $f_1(b)$, the optimum is at $x_1 = 3$. Therefore, the optimal is $x_1 = 3$ for $64.0 \leq b \leq 110$.

The results for stage 1 are presented in the dynamic programming table (Table 7, stage 1).

The next step is to search for the optimal combinations of stage 2 and stage 1 since $(x_5^{\ell}, x_4^{\ell}, x_3^{\ell}) = (2,3,1)$ is fixed. It is necessary to consider all possible b between 59.0 and 110.0. For convenience, the maximization will be carried out for b with a discrete spectrum. Since the cost of adding one more component to the minimum required components of any stage will consume at least 5 cost units, the difference between two near searching point can be chosen as 5. Table 7 (stage 1) then is used to construct Table 6b.

In Table 6b, $(x_5^{\ell}, x_4^{\ell}, x_3^{\ell}) = (2,3,1)$ is fixed. An optimal (x_2, x_1) is searched for the maximum function value $f_2(b)$, for the corresponding value of b given. This procedeure is similar to one given in the basic dynamic programming algorithm. For example, if b = 84, from Table 7 (stage 1) x_1 can be 2 or 3. When $x_1 = 2$, we search for the optimal x_2 to be 2, and the functional value is 0.66710; when $x_1 = 3$, we search for the optimal x_2 to be 2 and the functional value is 0.67476. Since 0.67476 is greater than 0.66710, the optimal allocation for b = 84 is $(x_5^{\ell}, x_4^{\ell}, x_3^{\ell}, x_2, x_1) = (2,3,1,2,3)$. In Table 6b, when $(x_5^{\ell}, x_4^{\ell}, x_3^{\ell}) = (2,3,1)$ is fixed, only two possible allocations exist for x_2 and x_1, namely, $(x_2, x_1) = (2,2)$ or $(2,3)$, which are presented in the dynamic programming table, Table 7 (stage 2). Similarly, we can construct

Table 6c for stage 3 (and stage 2 and stage 1) for all possible b values and for fixed $(x_5^{\ell}, x_4^{\ell}) = (2,3)$. For each b value, a systematic search procedure is carried out by using the previously determined optimal allocation of (x_2, x_1) shown in Table 7 (stage 2). We can also construct Table 6d for stage 4 (and stage3, stage 2, and stage 1) for all possible b values and for fixed $x_5^{\ell} = 2$.

Finally we can construct Table 5e for b = 110 which is the total allowable resource. For b = 110, from Table 7 (stage 4), all the possible optimal allocations of (x_4, x_3, x_2, x_1) for $b \leq 110$ are: (3,1,2,2), (3,1,2,3), (3,2,2,2), and (3,2,2,3). For each allocation, the optimum x_5 (the maximum allowable x_5 to give the maximum functional value) is calculated, and from all these possible function values we choose the largest one as the optimal one. As shown in Table 6e, $(x_5, x_4, x_3, x_2, x_1) = (3,3,2,2,3)$ gives

$f_5(b=110) = 0.74610$. The dynamic programming table for $\lambda = 0.001$ is given in Table 7.

Table 8 shows that when $\lambda = 0.001$, we have $f_5(b=110) = 0.74610$. Then the total consumed $g_2 = \sum_{j=1}^{N} g_{2j}(x_j) = 192.5$. The system reliability, R_s, is 0.9045, which is given by

$$f_5(b)/\exp(-\lambda \sum_{j=1}^{n} g_{2j}(x_j)).$$

For searching the proper value of the Lagrange multiplier, λ , which shall spend a cost as close to 200 (but always less than 200) as possible (since $g_2 \leq W$, where W = 200), several values of λ have been tried. For each value of λ, the procedures presented above are carried out and an optimum configuration is obtained. The results are summarized in Table 8 .

As $\lambda = 0.0001$, the optimal allocations are $(x_5, x_4, x_3, x_2, x_1) =$ (4,3,2,3,3), which give the system reliability, $R_s = 0.9331$, and consume

$g_1 = 107$, and $g_2 = 257.6$, i.e., the second constraint is violated. For $\lambda = 0.01$, the optimal allocations are $(x_5, x_4, x_3, x_2, x_1) = (2,3,1,2,2)$ which give the system reliability, $R_s = 0.7578$, and consume $g_1 = 59$, and $g_2 = 127.5$. Now 127.5 is smaller than 200 and the solution is a feasible one. Howvever, we can increase the stage redundancies, and consume more resource to increase the system reliability. The one-dimensional search for λ can be applied between 0.0001 and 0.01. Table 8 gives the optimal solution: $0.0008 \leq \lambda \leq 0.0015$, $(x_5, x_4, x_3, x_2, x_1) = (3,3,2,2,3)$, $g_1 = 83$, $g_2 = 192.5$, and $R_s = 0.9045$.

Table 6a Calculated Results of Example 3 for Stage 1 as λ = 0.0010.

b	x_5^{ℓ}	x_4^{ℓ}	x_3^{ℓ}	x_2^{ℓ}	x_1	functional value	$f_1(b)$
59-63.9	2	3	1	2	2	0.66710	*
64-70.9	2	3	1	2	3	0.67476	*
71-79.9	2	3	1	2	3	0.67476	*
	2	3	1	2	4	0.65860	
80-90.9	2	3	1	2	3	0.67476	*
	2	3	1	2	4	0.65860	
	2	3	1	2	5	0.62915	
91-103.9	2	3	1	2	3	0.67476	*
	2	3	1	2	4	0.65860	
	2	3	1	2	5	0.62915	
	2	3	1	2	6	0.60322	
104-110	2	3	1	2	3	0.67476	*
	2	3	1	2	4	0.65860	
	2	3	1	2	5	0.62915	
	2	3	1	2	6	0.60322	
	2	3	1	2	7	0.58209	

Table 6b Calculated Results of Example 3 for Stage 2 (and Stage 1) as $\lambda = 0.0010$

b	x_5^ℓ	x_4^ℓ	x_3^ℓ	x_2	x_1	functional value	f (b)
59	2	3	1	2	2	0.66710	*
64	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
69	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
74	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
79	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
84	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
89	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
94	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
99-110	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*

Table 6c Calculated Results of Example 3 for Stage 3 (and Stage 2 and Stage 1) as $\lambda = 0.0010$

b	x_5^{λ}	x_4^{λ}	x_3	x_2	x_1	functional value	$f_3(b)$
59	2	3	1	2	2	0.66710	*
64	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
69	2	3	2	2	2	0.72208	*
	2	3	1	2	3	0.67476	
74	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*
79	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*
84	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*
89	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*
94	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*
99-110	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*

Table 6d Calculated Results of Example 3 for Stage 4 (and Stage 3, Stage 2 and Stage 1) as $\lambda = 0.0010$

b	x_5^{λ}	x_4	x_3	x_2	x_1	functional value	$f_4(b)$
59	2	3	1	2	2	0.66710	*
64	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	*
69	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	
	2	3	2	2	2	0.72208	*
74	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	
	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*
79-110	2	3	1	2	2	0.66710	
	2	3	1	2	3	0.67476	
	2	3	2	2	2	0.72208	
	2	3	2	2	3	0.73037	*

Table 6e Calculated Results of Example 3 for Stage 5 (and Stage 4, Stage 3, Stage 2 and Stage 1) as $\lambda = 0.001$

b	x_5	x_4	x_3	x_2	x_1	functional value	$f_5(b)$
110	3	3	1	2	2	0.68147	
	3	3	1	2	3	0.68929	
	3	3	2	2	2	0.73764	
	3	3	2	2	3	0.74610	*

Table 7 The dynamic programming table for Example 3 as $\lambda = 0.0010$.

Stage 5 (and Stage 4, Stage 3, Stage 2 and Stage 1)

b	x_5	x_4	x_3	x_2	x_1	$f_5(b)$
110	3	3	2	2	3	0.74610

Stage 4 (and Stage 3, Stage 2 and Stage 1)

b	x_5^ℓ	x_4	x_3	x_2	x_1	$f_4(b)$
59	2	3	1	2	2	0.66710
64	2	3	1	2	3	0.67476
69	2	3	2	2	2	0.72208
74-110	2	3	2	2	3	0.73037

Stage 3 (and Stage 2 and Stage 1)

b	x_5^ℓ	x_4^ℓ	x_3	x_2	x_1	$f_3(b)$
59	2	3	1	2	2	0.66710
64	2	3	1	2	3	0.67476
69	2	3	2	2	2	0.72208
74-110	2	3	2	2	3	0.73037

Stage 2 (and Stage 1)

b	x_5^ℓ	x_4^ℓ	x_3^ℓ	x_2	x_1	$f_2(b)$
59	2	3	1	2	2	0.66710
64-110	2	3	1	2	3	0.67476

Stage 1

b	x_5^ℓ	x_4^ℓ	x_3^ℓ	x_2^ℓ	x_1	$f_1(b)$
56-63.9	2	3	1	2	2	0.66710
64-110	2	3	1	2	3	0.67476

Table 8 Optimum System Reliabilities for Various Values of Lagrange Multiplier

Lagrange Multiplier λ	Optimum System Configuration x_5	x_4	x_3	x_2	x_1	Optimum System Reliability R_s	g_1	g_2
0.0001	4	3	2	3	3	0.9331	107	257.6
0.0002	4	3	2	3	3	0.9331	107	257.6
0.0004	4	3	2	3	3	0.9331	107	257.6
0.0006	3	3	2	3	3	0.9222	93	216.9
0.0008	3	3	2	2	3	0.9045	83	192.5
0.0015	3	3	2	2	3	0.9045	83	192.5
0.0016	3	3	2	2	2	0.8753	78	171.1
0.0040	2	3	2	2	2	0.8336	68	143.6
0.0060	2	3	1	2	2	0.7578	59	127.5
0.0080	2	3	1	2	2	0.7578	59	127.5
0.0100	2	3	1	2	2	0.7578	59	127.5

4 Dynamic Programming Approach Using the Concept of Dominating Sequence

Formulation of the Problem

The number of computations required for maximizing the system reliability

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

subject to

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

can be reduced by defining a condition of dominance for alternative system configurations.

A system configuration $\bar{x}'$ is said to dominate another system configuration $\bar{x}$, if

$$R_s(\bar{x}') \geq R_s(\bar{x}),$$

and the inequality sign (<) holds in at least one of the following conditions

$$\sum_{j=1}^{N} g_{ij}(\bar{x}'_j) \leq \sum_{j=1}^{N} g_{ij}(\bar{x}_j) \quad , i=1,2,\ldots,r$$

This implies that the dominating system configuration has better system reliability and using less cost (resources). A sequence S of redundancy allocations, satisfying the constraints in (5) and none of them being dominated by the others, is said to form a dominating sequence.

In the dynamic programming formulation, combinations of two stages are searched for a dominating sequence of configurations which is then combined with a third stage to yield another dominating sequence. A sequence ends whenever a constraint is violated. The final dominant configuration yielding the optimal system configuration is the last entry in the dominating sequence generated by the combination of the dominating sequence from stage 1, stage 2, ..., stage N-1, and stage N.

To reduce the length of the dominating sequence, the heuristic techniques used to determine the upper and lower bounds of x_j, $j=1,2,\ldots,N$ may be suggested.

i) Upper bound of x_j, x_j^u:

Each stage should have at least one component. If the upper bound of the jth stage, x_j^u, is to be determined, we let

$$x_k = 1, \qquad k=1,2,\ldots,N \quad, k \neq j$$

x_j^u is the smallest integer number in the set $\{c_1, c_2, \ldots, c_r\}$, where $c_\ell = \max \{x_j | x_j \text{ is integer, and } g_{\ell k} (1,\ldots,1,x_j,1\ldots,1) \leq b_\ell\}$ for $\ell=1,2,\ldots,r$.

ii) Lower bound of x_j, x_j^ℓ:

Redundancies are allocated stage by stage until a constraint is met. If the reliability of the configuration $\bar{x}$, which is the last step of allocation, while not violating any constraint, is $R_s(\bar{x})$, then N equations of the form $R_s(\bar{x}) \leq 1 - (1 - R_j)^{x_j}$ are solved for x_j, where x_j^ℓ is the minimum integer numbers satisfying the above equations for $j=1,2,\ldots,N$. x_j^ℓ is the lower bound of components used at stage j.

Example 5

To use the concept of dominating sequence to solve this example, we first find the upper and lower bounds of components used at each stage.

i) Upper bound, x_j^u:

To find the upper bounds of components for the jth stage, all the other stages are assumed to have one component. The upper bound of the first stage, x_1^u, will be the largest integer number satisfying the following three constraints:

$$g_1 = 1\cdot(x_1^u)^2 + 2\cdot(1)^2 + 3\cdot(1)^2 + 4\cdot(1)^2 + 2\cdot(1)^2 \leq 110$$

$$g_2 = 7\cdot(x_1^u + \exp(x_1^u/4)) + 7\cdot(1 + \exp(1/4)) + 5\cdot(1 + \exp(1/4))$$
$$+ 9\cdot(1 + \exp(1/4)) + 4\cdot(1 + \exp(1/4)) \leq 175$$

$$g_3 = 7\cdot x_1^u \exp(x_1^u/4) + 8\cdot 1\cdot\exp(1/4) + 3\cdot 1\cdot\exp(1/4)$$
$$+ 6\cdot 1\cdot\exp(1/4) + 9\cdot 1\cdot\exp(1/4) \leq 200$$

By plugging the integer number, $x_1^u = 1, 2, \ldots$ into g_1, g_2, g_3, when $x_1^u = 6$, we have $g_1 = 47$, $g_2 = 137.34$, and $g_3 = 227.36$, that is, $g_3\ (x_1^u = 6)$ is greater than 200. When $x_1^u = 5$, however, $g_1 = 36$, $g_2 = 91.44$, and $g_3 = 161.59$. None of the constraints is violated. Therefore, x_1^u is 5. By similar procedures, the upper bounds of components for the other stages are found to be all 5.

ii) Lower bound, x_j^{ℓ}:

By an enumeration method as shown in Table 9, redundancies are allocated stage by stage until one of the constraints is exceeded. The basic system configuration for calculating the lower bound, then, is assumed to be the one before the exceeding one of the constraints, i.e.,(3, 3, 3, 2, 2) for this numerical example.

The system reliability corresponding to this configuration of (3, 3, 3, 2, 2), $R_s(\bar{x})$, is 0.8124.

Table 9 The allocation of elements stage by stage until any one constraint is violated.

Element Allocation					Resources used		
stages							
1	2	3	4	5	$\sum_{j=1}^{5} g_{1j}$	$\sum_{j=1}^{5} g_{2j}$	$\sum_{j=1}^{5} g_{3j}$
1	1	1	1	1	12	73.09	48.79
2	1	1	1	1	15	82.64	62.88
2	2	1	1	1	21	92.19	78.99
2	2	2	1	1	30	99.01	95.10
2	2	2	2	1	42	111.29	107.18
2	2	2	2	2	48	116.75	125.30
3	2	2	2	2	53	127.03	146.67
3	3	2	2	2	63	138.31	171.11
3	3	3	2	2	78	144.65	195.53
3	3	3	3	2	98	157.87	213.86

This is, however, not an optimal solution. The optimal system reliability should be equal to or greater than this value. Therefore, we assume that the lower bound of stage reliability is 0.8124, and calculate the corresponding lower bound of stage components.

That is, for j=1,

$$1 - (1 - 0.80)^{x_1} \geq 0.8124$$

gives $x_1 \geq 2.31$, say $x_1^{\ell} = 2$. Similarly we obtain $x_2^{\ell} = 2$, $x_3^{\ell} = 1$, $x_4^{\ell} = 3$, and $x_5^{\ell} = 2$.

The optimum components at each stage will then lie between the lower and upper bounds of that stage.

To solve this example, the first step in the computational procedures is to set up a matrix for the combination of stage 1 and stage 2 (see Table 9a). In Table 9a, the number of components, stage unreliability, g_1, g_2, and g_3 for stage 1 and stage 2 are presented as the rows above the matrix and column left of the matrix, respectively. The starting number of components used for each stage is the lower bound of the stage, and the ending point, the upper bound. It is easier to consider unreliabilities than reliabilities, although it involves an approximation.

Each entry of the matrix in Table 9a is a vector, which shows the system unreliabilities, g_1, g_2, and g_3 which are results of the combination of stage 1 and stage 2. The system unreliability is approximated by the addition of the unreliabilites of stage 1 and stage 2, if both R_1 and R_2 are near unity, namely, R_1 and $R_2 \geq 0.5$,

$$Q' = 1 - (1 - (1 - R_1)^{x_1})\,(1 - (1 - R_2)^{x_2})$$
$$\approx (1 - R_1)^{x_1} + (1 - R_2)^{x_2}$$

where $(1 - R_1)^{x_1}$ and $(1 - R_2)^{x_2}$ are the unreliabilities of stage 1 and stage 2, respectively.

The dominating sequence for the system combining stage 1 with stage 2 is obtained by eliminating entries of the matrix which are dominated by others. The eliminating procedures are:

(1) Any cost of the entries in the matrix exceeds the constrained available resource, then the entry is eliminated. For example, the entries of (x_1, x_2) = (5, 4), (4, 5), and (5, 5) are eliminated, because all g_3's of the entries exceed 200.

(2) The dominating sequence will then be determined as follows:

a. Consider the entry having the highest reliability (i.e., the lowest unreliability), which is always one term of the dominating sequence no matter what costs the term has. In Table 10a, this entry is (x_1, x_2) = (4, 4), which has the highest reliability, 1 - 0.0021 = 0.9979. Compare costs of all the other entries with costs of this entry. Eliminate all entries which have lower reliability and higher cost. In table 9a, the highest entry is (x_1, x_2) = (4, 4), which has reliability 0.9979, g_1 = 48, g_2 = 94.04, and g_3 = 163.08. Comparing with (x_1, x_2) = (4, 4), the entry (x_1, x_2) = (3, 5), which has reliability 1 - 0.0081 = 0.9919, g_1 = 59, g_2 = 95.17, and g_3 = 183.66, is eliminated, because the latter one is less reliable and requires higher costs for g_1, g_2, and g_3. That is, entry (4, 4) dominates entry (3, 5).

b. Choose the next higher reliability (lower unreliability), i. e., entry (5, 3). Compare the costs of all other entries which have lower reliability than entry (5, 3). However, no entry is dominated by (5, 3).

c. Next, entries (3, 4) and (2,5) are eliminated by comparing with entry (4, 3); entry (2, 4) and (5, 2) are eliminated by comparing with entry (3, 3); and entry (2, 3) is eliminated by comparing with (3, 2). Finally the dominating sequence of (1), (2), (3), (4), (5), (6), and

(7) is obtained which is the system composing of stage 1 and stage 2.

The dominating sequence for the combination of stage 1 and stage 2 from Table 9a will be the new row entry above the matrix in Table 10a. The number of components, stage unreliabilities, g_1, g_2, and g_3 of stage 3 will be the column left to the matrix of Table 9b. Similar procedures are now carried out to eliminate the entries of this matrix whose costs exceed the constraint, i.e., (4 - 4, 4); (4 - 4, 3); (3 - 3, 4); (3 -3, 3); (4 - 3, 4). The dominating sequence is then determined. (3 -3, 2) and (4 - 2, 4) are eliminated by comparing (3 - 3, 3); (4 - 4, 1) and (3 - 3, 1) by (4 - 3, 2); (4 - 2, 3), (3 - 2, 4) and (3 - 2, 3) by (3 - 3, 2); (4 - 3, 1) by (4 -2, 2); (3 - 3, 1) and (2 - 2, 4) by (3 -2, 2); and (4 - 2, 1) by (2 - 2, 3). The dominating sequence is (2 - 2, 1), (3 - 2, 1), (2 - 2, 2), (2 - 2, 3), (3 - 2, 2), (4 - 2, 2), (3 - 3, 2), (4 - 3, 2), (3 -3, 3) (4 - 4, 2), (3 - 3, 4), and (4 - 3, 3).

The dominating sequence obtained for system composed by stages 1, 2, and 3 then forms the row entries above the matrix of Table 10c. Stage 4 is combined with stages 1 - 2 - 3 to form a system, and its dominating sequence is obtained from Table 9c. This dominating sequence for the system composing stages 1, 2, 3, and 4 is used to combine with stage 5 to get the last dominating sequence as shown in Table 10d. In Table 10d, a dominating sequence is obtained, and the optimal one has the system configuration of (3, 2, 2, 3, 3) which has the highest reliability, 1 - 0.0990 = 0.9010.

Table 10a Computational results of Example 5 for Stage 1 and Stage 2

	Stage 1			
Number of components used	$x_1^{\ell} = 2$	3	4	$x_1^u = 5$
Stage unreliability	0.04	0.008	0.0016	0.0003
g_1 used	4	9	16	25
g_2 used	25.54	35.81	47.02	59.36
g_3 used	23.08	44.45	76.10	121.81
Stage 2				
$x_2^{\ell} = 2$	(1)	(2)	(3)	
0.0225	0.0625	0.0305	0.0241	0.0228
8	12	17	24	33
25.54	51.08	61.35	72.56	84.90
26.38	49.46	70.83	102.48	148.19
3		(4)	(5)	(6)
0.0034	0.0434	0.0114	0.0050	0.0037
18	22	27	34	43
35.81	61.35	71.62	82.83	95.17
50.81	73.89	95.26	126.91	172.61
4			(7)	
0.0005	0.0405	0.0085	0.0021	0.0008
32	36	41	48	57
47.02	72.56	82.83	94.04	106.38
86.98	110.06	131.43	163.08	208.79
$x_2^u = 5$				
0.0001	0.0401	0.0081	0.0017	0.0004
50	54	59	66	75
59.36	84.90	95.17	106.38	118.72
139.21	162.29	183.66	215.31	261.02

Table 10b Computational results of Example 5 for Stage 1-2 and Stage 3

	Stage 1-2						
Number of components used	2-2	3-2	4-2	3-3	4-3	5-3	4-4
Stage unreliability	0.0625	0.0305	0.0241	0.0114	0.0050	0.0037	0.0021
g_1 used	12	17	24	27	34	43	48
g_2 used	51.08	61.35	72.56	71.62	82.83	95.17	94.04
g_3 used	49.46	70.83	102.48	95.26	126.91	172.62	163.08
$x_3^{\ell} = 1$	(1)	(2)					
0.10	0.1625	0.1305	0.1241	0.1114	0.1050	0.1037	0.1021
3	15	20	27	30	37	46	51
11.42	62.50	72.76	83.98	83.04	94.25	106.59	105.46
10.27	59.73	81.10	112.75	105.53	137.28	182.89	173.35
2	(3)	(5)	(6)	(7)	(8)		(10)
0.01	0.0725	0.0405	0.0341	0.0214	0.0150	0.0137	0.0121
12	24	29	36	39	46	55	60
18.24	69.52	79.59	90.80	89.86	101.07	113.41	112.28
26.38	75.84	97.21	128.86	121.64	153.29	199.00	189.46
3	(4)			(9)	(12)		
0.001	0.0655	0.0315	0.0251	0.0124	0.0060	0.0047	0.0031
27	39	44	51	54	61	70	75
25.58	76.66	89.93	98.14	97.20	108.41	120.75	119.02
50.81	100.27	121.64	153.29	146.07	177.72	223.43	213.89
$x_3^{u} = 4$				(11)			
0.0001	0.0626	0.0306	0.0242	0.0115	0.0052	0.0038	0.0022
48	60	65	72	75	82	91	96
33.58	84.66	94.93	106.14	105.20	116.41	128.75	127.61
86.98	136.44	157.81	189.46	182.24	213.89	259.60	250.06

Table 10c Computational results of Example 5 for Stage 1-2-3 and Stage 4

		Stage 1-2-3											
	Number of components used	2-2-1	3-2-1	2-2-2	2-2-3	3-2-2	4-2-2	3-3-2	4-3-2	3-3-3	4-4-2	3-3-4	4-3-3
	Stage unreliability	0.1625	0.1305	0.0725	0.0635	0.0405	0.0341	0.0214	0.0150	0.0124	0.0121	0.0115	0.0060
	g_1 used	15	20	24	39	29	36	39	46	54	60	75	61
	g_2 used	62.50	72.76	69.32	76.66	79.59	90.80	89.86	101.07	97.20	112.28	105.20	103.41
	g_3 used	59.73	81.10	75.84	100.27	97.21	128.86	121.64	153.29	146.07	189.46	182.24	177.72
Stage 4	$x_4^{\ell} = 3$	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(10)			
	0.0429	0.2054	0.1734	0.1154	0.1064	0.0834	0.0770	0.0643	0.0579	0.0553	0.0550	0.0544	0.0489
	36	51	56	60	75	65	72	75	82	90	96	111	97
	46.05	108.55	118.81	115.37	122.71	126.64	136.85	135.91	147.12	108.55	158.33	118.81	154.45
	38.10	97.83	119.20	113.94	138.37	135.31	166.96	159.74	191.39	184.17	227.56	220.34	215.82
	$x_4^{u} = 4$					(9)	(11)	(12)					
	0.0150	0.1775	0.1455	0.0875	0.0785	0.0555	0.0491	0.0364	0.0300	0.0274	0.0271	0.0265	0.0210
	64	79	84	88	103	93	100	103	110	118	124	139	125
	60.47	122.97	133.23	129.79	137.13	140.06	151.27	150.33	161.54	157.67	172.75	165.67	168.88
	65.23	124.96	146.33	141.07	165.50	162.44	194.09	186.87	218.52	211.30	254.69	247.47	242.95

Table 10d Computational results of Example 5 for Stage 1-2-3-4 and Stage 5

		Stage 1-2-3-4											
	Number of components used	2-2-1-3	3-2-1-3	2-2-2-3	2-2-3-3	3-2-2-3	4-2-2-3	3-3-2-3	4-3-2-3	3-2-2-4	3-3-3-3	4-2-2-4	3-3-2-4
	Stage unreliability	0.2054	0.1734	0.1154	0.1064	0.0834	0.0770	0.0643	0.0579	0.0555	0.0553	0.0491	0.0364
	g_1 used	51	56	60	75	65	72	75	82	93	90	100	103
	g_2 used	108.55	118.81	115.37	122.71	126.64	136.85	135.91	147.12	140.06	108.55	151.27	150.33
	g_3 used	97.83	119.20	113.94	138.37	135.31	166.96	159.74	191.39	162.44	184.17	194.09	186.87
Stage 5	$x_5^{\ell} = 2$	(1)	(2)	(4)	(5)	(6)		(8)		(10)			
	0.0625	0.2679	0.2359	0.1779	0.1689	0.1459	0.1395	0.1268	0.1204	0.1180	0.1178	0.1116	0.0989
	8	59	64	68	83	73	80	83	90	101	98	108	111
	14.59	123.14	133.40	129.96	137.30	141.23	151.44	150.50	161.71	154.65	123.14	165.86	164.92
	29.68	127.51	148.88	143.62	168.05	164.99	196.64	189.42	221.07	192.12	213.85	223.77	216.55
	$x_5^{u} = 3$	(3)		(7)	(9)	(11)*							
	0.0156	0.2210	0.1890	0.1310	0.1220	0.0990	0.0926	0.0799	0.0735	0.0711	0.0709	0.0647	0.0520
	18	69	74	78	93	83	90	93	100	111	108	118	121
	20.46	129.01	139.27	135.83	143.17	147.10	157.31	156.47	167.58	160.52	129.09	171.73	170.79
	57.16	154.99	176.36	171.19	195.53	192.47	224.12	216.90	248.55	219.60	241.33	251.25	244.03

REFERENCES

1. Bellman, R., Dynamic Programming, Princeton, N.J.: Princeton University Press (1957).

2. Bellman, R., and S. E. Dreyfus, Applied Dynamic Programming, Princeton, N.J.: Princeton University Press (1962).

3. Bellman, R., Modern Analytic and Computational Methods in Science and Mathematics, N.Y.: Elsevier Publishing Co. (1968).

4. Bellman, R. E., and S. E. Dreyfus, "Dynamic programming and reliability of multi-component devices", Operations Research, Vol. 6, pp. 200-206 (March-April 1958).

5. Black, G., and F. Proschan, "On optimal redundancy", Operations Research, Vol. 7, pp. 581-588 (1959).

6. Burton, R. M., and G. T. Howard, "Optimal system reliability for a mixed series and parallel structure", Journal of Mathematical Analysis and Applications, Vol. 28, pp. 370-382 (1969).

7. Fyffe, D. E., W. W. Hines, and N. K. Lee, "System reliability allocation and a computational algorithm", IEEE Transactions on Reliability, Vol. R-17, No. 2, pp. 64-69 (June 1968).

9. Hadley, G., Nonlinear and Dynamic Programming, Reading, Mass: Addison-Wesley (1964).

9. Jensen, P. A., "Optimization of series-parallel-series networks", Operations Research, Vol. 18, pp. 471-482 (May-June 1970).

10. Kettelle, J. D., "Least-cost allocation of reliability investment", Operations Research, Vol. 10, pp. 249-265 (March-April 1967).

11. Kulshrestha, D. K., and M. C. Gupta, "Use of dynamic programming for reliability engineers, IEEE Transactions on Reliability, Vol. R-22, pp. 240-241 (Oct. 1973).

12. Lambert, B. K., A. G. Walvekar, and J. P. Hirmas, "Optimal redundancy and availability allocation in multistage systems", IEEE Transactions on Reliability, Vol. R-20, No. 3, pp. 182-185 (August 1971).

13. Liittschwager, J. M., "Dynamic programming in the solution of a multistage reliability problem", Journal of Industrial Engineering, Vol. 15-16, pp. 168-175 (1964-1965).

14. Messinger, M., and Shooman, H., "Technique for optimum spares allocation: a tutorial review", IEEE Transactions on Reliability, Vol. R-19, pp. 156-166 (November 1970).

15. Nemhauser, G. L., Introduction to Dynamic Programming, N.Y.: Wiley (1967).

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

17. Rudd, D. F., "Reliability theory in chemical system design", I&EC Fundamental, Vol. 1, No. 2, pp. 138-143 (May 1962).

18. Shershin, A. C., "Mathematical optimization techniques for the simultaneous apportionments of reliability and maintainability", Operations Research, Vol. 18, No. 1, pp. 95-106 (January-February 1970).

19. Woodhouse, C. F., "Optimal redundancy allocation by dynamic programming", IEEE Transactions on Reliability, Vol. R-21, No. 1, pp. 60-62 (February 1972).

3.3 THE DISCRETE MAXIMUM PRINCIPLE APPLIED TO OPTIMUM SYSTEM RELIABILITY

1. Introduction

A simple computational procedure based on the discrete maximum principle has been developed for maximizing reliability of multistage parallel systems subject to multiple nonlinear constraints [1]. It appears that the procedure can be applied to a variety of optimization problems with separable and multiple constaints functions.

2. Statement of the Problem and the Computational Procedure

The problem of maximizing the reliability of an N-stage series system with redundant units in parallel (see Fig. 1) subject to multiple linear and nonlinear separable constraints can be stated as follows:

Maximize

$$R_s = \prod_{n-1}^{N} (1-(1-R^n)^{\theta^n}) \tag{1}^*$$

subject to

$$\sum_{n=1}^{N} g_i^n(\theta^n) \leq b_i, \qquad i = 1, 2, \ldots, s, \tag{2}$$

where

* The superscript n indicates the stage number. The exponents are written with parentheses or brackets such as $(x^n)^2$ or $\{T^n(x^{m-1};\theta^n)\}^2$.

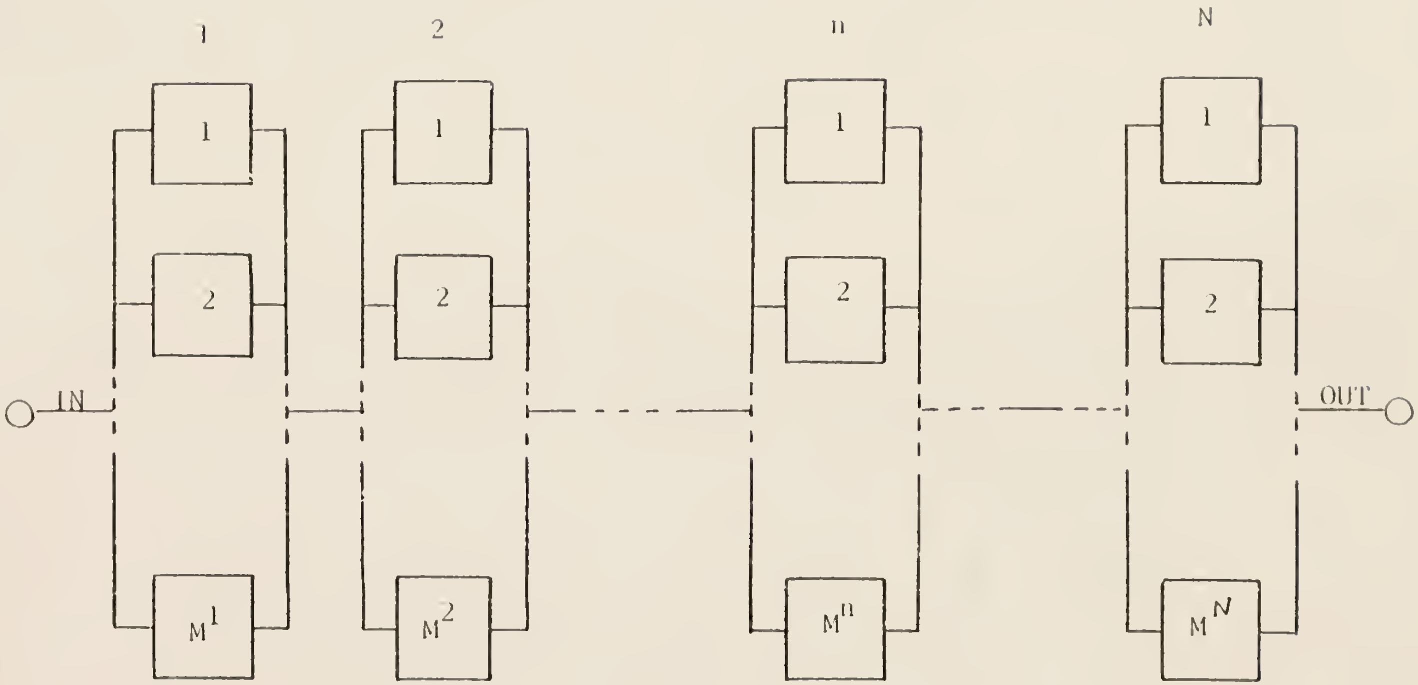


Fig. 1. A mixed system with N-stages in series where components are in parallel at each stage.

R_s = the system reliability,

N = the total number of stages,

R^n = the reliability of one element at the nth stage,

θ^n = the number of elements at the nth stage, where (θ^n-1) is the number of redundant units,

$g_i^n(\theta^n)$ = the function representing the amount of the ith resource consumed at the nth stage as a function of θ^n,

s = the number of constraints,

b_i = the total amount of the ith resource available.

Let

x_i^n = the ith resource corresponding to the ith constraint, which is consumed in the first n stages, $i = 1, 2, \ldots, s$.

Then, the performance equations for this N-stage system may be written as

$$x_i^n = x_i^{n-1} + g_i^n(\theta^n), \qquad n = 1, 2, \ldots, N, \quad i = 1, 2, \ldots, s, \tag{3}$$

$$x_i^0 = 0, \tag{3a}$$

$$x_i^N \leq b_i. \tag{3b}$$

By defining

$$x_{s+1}^n = x_{s+1}^{n-1} + \ln\left(1-(1-R^n)^{\theta^n}\right), \qquad n = 1, 2, \ldots, N, \tag{4}$$

$$x_{s+1}^0 = 0,$$

the objective function to be optimized can be written as

$$S = \ln R_s$$

$$= x_{s+1}^N$$

$$= \sum_{i=1}^{s+1} c_i x_i^N, \tag{5}$$

where

$$c_i = 0, \qquad i = 1, 2, \ldots, s, \tag{6a}$$

$$c_{s+1} = 1.$$

The Hamiltonian and the adjoint variables of the system can be defined as

$$H^n = \sum_{i=1}^{s+1} z_i^n x_i^n$$

$$= \sum_{i=1}^{s} z_i^n \left(x_i^{n-1} + g_i^n (\theta^n)\right) + z_{s+1}^n \left(x_{s+1}^{n-1} + \ln\left(1-(1-R^n)^{\theta^n}\right)\right), \tag{7}$$

$$n = 1, 2, \ldots, N,$$

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}} = z_i^n, \qquad n = 1, 2, \ldots, N, \tag{8}$$

$$i = 1, 2, \ldots, s, s+1,$$

$$z_{s+1}^N = c_{s+1} = 1. \tag{9}$$

Equations (8) and (9) yield

$$z_{s+1}^n = 1, \qquad n = 1, 2, \ldots, N. \tag{10}$$

Assuming that the non-trivial and unique Hamiltonian and adjoint variables of the system exist, the stational necessary condition for local optimality can be obtained as (see Appendix 4 of Ref. 2)

$$\frac{\partial H^n}{\partial \theta^n} = 0 = \sum_{i=1}^{s} z_i^n \frac{\partial g_i^n(\theta^n)}{\partial \theta^n} + \frac{-(1-R^n)^{\theta^n} \ln (1-R^n)}{1-(1-R^n)^{\theta^n}} . \tag{11}$$

In employing this condition in determining the optimal condition of the system, we assume its existence. In reality, θ^n, $n = 1, 2, \ldots, N$, are positive integers. We, however, assume that θ^n are continuous variables.

Now we assume that one of the constraints, say the jth constraint given by equation (2) or equivalently by equation (3), is active and the rest are free. This means the end condition corresponding to the jth constraint is fixed and the rest of them are free. Then, we have

$$z_i^N = c_i = 0, \qquad i = 1, 2, \ldots, s, \quad i \neq j. \tag{12}$$

From equations (8) and (12), we obtain

$$z_i^n = 0, \qquad i = 1, 2, \ldots, s, \quad i \neq j, \quad n = 1, 2, \ldots, N.$$

Therefore, equation (11) reduces to

$$z_j^n \frac{\partial g_j^n(\theta^n)}{\partial \theta^n} - \frac{(1-R^n)^{\theta^n} \ln (1-R^n)}{1-(1-R^n)^{\theta^n}} = 0. \tag{13}$$

The procedure for solving the problem may be described in the following steps.

Step 1. Assuming a value for θ^1 in equation (13), we obtain z_j^1.

Furthermore, equation (8) gives

$$z_j^1 = z_j^n, \qquad n = 2, \ldots, N.$$

Step 2. We find θ^n, $n = 2, 3, . . . , N$, from equation (13) by using the values of z_j^n obtained.

Step 3. We compute x_i^N, $i = 1, 2, . . . , s$, from equation (3).

Step 4. One of the following conditions will occur.

a) If $x_i^N < b_i$ for all $i = 1, 2, . . . , s$, then we assume a higher value for θ^1 and reutrn to step 1. b) If $x_j^N > b_j$ and $x_i^N < b_i$ for $i \neq j$, $i = 1, 2, . . . , s$, we assume a smaller value for θ^1 and return to step 1. c) If $x_k^N > b_k$, $k \neq j$ and $x_i^N < b_i$, $i = 1, 2, , j , , s$, $i \neq k$, where j is the active constraint, then we go to step 5. d) If $x_j^N = b_j$, and $x_i^N < b_i$, $i = 1, 2, . . . , s$, $i \neq j$, that is, the jth constraint reaches its limit while none of the other constraints are violated, we have a candidate for the optimal solution.

Step 5. We replace constraint j by constraint k. Accordingly we replace j by k in equation (13) and steps 1 and 2 and repeat the procedure given by step 1 through step 4.

3. Example

Constraints of a system can be the total weight, the total cost, the total volume and so on. In general, such constraints are in nonlinear forms. As the number of units at each stage is increased, it requires the increased number of connecting equipment, and thus, the cost and weight may increase exponentially.

Let

c^n = cost per element at the nth stage,

w^n = weight per element at the nth stage,

v^n = volume per element at the nth stage,

θ^n = number of elements in parallel at the nth stage.

Therefore the following nonlinear constraints on the combination of weight and volume, cost, and weight are considered.

(1) The constraint which is imposed on the combination of weight and volume is

$$\sum_{n=1}^{N} g_1^n(\theta^n) = \sum_{n=1}^{N} p^n (\theta^n)^2 \leq P,$$

where $p^n = w^n v^n$ is the product of weight per unit and volume per unit at the nth stage.

(2) The cost constraint is

$$\sum_{n=1}^{N} g_2^n(\theta^n) = \sum_{n=1}^{N} c^n \left(\theta^n + \exp(\theta^n/4)\right) \leq C,$$

where $c^n\theta^n$ is the cost of units at the nth stage and $c^n(e)^{\theta^n/4}$ is the additional cost for interconnecting parallel units (4).

(3) The weight constraint is

$$\sum_{n=1}^{N} g_3^n(\theta^n) = \sum_{n=1}^{N} w^n\theta^n \exp(\theta^n/4) \leq W,$$

where $w^n\theta^n$ is the weight of the total units at the nth stage. This is increased by a factor $\exp(\theta^n/4)$ due to the weight of the interconnecting links (4).

The problem is to maximize the system reliability subject to the above constraints. State variables of the system are defined as follows:

$$x_1^n = x_1^{n-1} + p^n(\theta^n)^2, \qquad n = 1, 2, \ldots, N, \tag{14}$$

$$x_1^0 = 0,$$

$$x_1^N \leq P,$$

$$x_2^n = x_2^{n-1} + c^n\left(\theta^n + (e)^{\theta^n/4}\right), \qquad n = 1, 2, \ldots, N, \tag{15}$$

$$x_2^0 = 0,$$

$$x_2^N \leq C,$$

$$x_3^n = x_3^{n-1} + w^n\theta^n(e)^{\theta^n/4}, \qquad n = 1, 2, \ldots, N, \tag{16}$$

$$x_3^0 = 0,$$

$$x_3^N \leq W,$$

$$x_4^n = x_4^{n-1} + \ln\{1 - (1 - R^n)^{\theta^n}\}, \qquad n = 1, 2, \ldots, N, \tag{17}$$

$$x_4^0 = 0.$$

The objective function to be maximized is

$$S = \sum_{n=1}^{N} \ln\left(1 - (1 - R^n)^{\theta^n}\right)$$

$$= \sum_{i=1}^{4} c_i x_i^N \tag{18}$$

$$= x_4^N$$

where

$$c_i = 0, \qquad i = 1, 2, 3, \tag{19}$$

$$c_4 = 1.$$

The Hamiltonian and adjoint variables of the system are

$$H^n = \sum_{i=1}^{4} z_i^n x_i^n$$

$$= z_1^n\left(x_1^{n-1} + p^n(\theta^n)^2\right) + z_2^n\{x_2^{n-1} + c^n\left(\theta^n + (e)^{\theta^n/4}\right)\} + z_3^n\left(x_3^{n-1} + w^n\theta^n(e)^{\theta^n/4}\right)$$

$$+ z_4^n\left(x_4^{n-1} + \ln\{1 - (1 - R^n)^{\theta^n}\}\right), \tag{20}$$

$$n = 1, 2, 3, \ldots, N.$$

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}} = z_i^n, \qquad i = 1, 2, 3, 4, \tag{21}$$

$$z_4^N = c_4 = 1 . \tag{22}$$

From equations (21) and (22), we obtain

$$z_4^n = 1, \qquad n = 1, 2, \ldots, N.$$

Differentiating equation (20) with respect to θ^n and equating to zero, we obtain

$$\frac{\partial H^n}{\partial \theta^n} = 0$$

$$= 2z_1^n p^n \theta^n + z_2^n c^n \left(1 + \frac{1}{4}(e)^{\theta^n/4}\right)$$

$$+ z_3^n w^n\left((e)^{\theta^n/4} + \frac{1}{4}\theta^n (e)^{\theta^n/4}\right) + \frac{-(1-R^n)^{\theta^n} \ell n(1-R^n)}{1-(1-R^n)^{\theta^n}} . \tag{23}$$

Whenever the j-th constraint, represented by x_j^N, is active, this has the effect of fixing its boundary value. Thus

$$z_i^N = c_i, \qquad i \neq j . \tag{24}$$

Now, if the first constraint is the only one in active, we obtain the following relations from equations (24), (21), and (19).

$$z_i^N = c_i = 0, \qquad i = 2, 3,$$

and

$$z_i^n = 0, \qquad n = 1, 2, \ldots, N;$$
$$i = 2, 3.$$

Consequently equation (23) can be written as

$$2z_1^n p^n \theta^n = \frac{(U^n)^{\theta^n} \ln U^n}{1-(U^n)^{\theta^n}} , \tag{25}$$

where

$$U^n = 1-R^n .$$

Rearranging the terms in equation (25), we have

$$z_1^n = \frac{1}{2p^n\theta^n} \frac{(U^n)^{\theta^n} \ln U^n}{1-(U^n)^{\theta^n}} , \tag{26}$$

and

$$\theta^n = (U^n)^{\theta^n}\left(\theta^n + \frac{\ln U^n}{2z_1^n p^n}\right) . \tag{27}$$

Letting

$$A^n = \frac{\ln U^n}{2z_1^n p^n} ,$$

equation (27) becomes

$$\theta^n = (U^n)^{\theta^n}\left(\theta^n + A^n\right),$$

or

$$f(\theta^n) = \theta^n - (U^n)^{\theta^n}\left(\theta^n + A^n\right) = 0. \tag{28}$$

This equation can be solved by Newton's method for θ^n.

Similarly, if the second constraint is active and the rest of them are free, we obtain the following relations.

$$z_i^N = c_i = 0, \qquad i = 1, 3,$$

$$z_i^n = 0, \qquad n = 1, 2, \ldots, N,$$

$$i = 1, 3,$$

$$z_2^n c^n \left(1 + \tfrac{1}{4}(e)^{\theta^n/4}\right) = \frac{(U^n)^{\theta^n} \ln U^n}{1-(U^n)^{\theta^n}}, \tag{29}$$

$$z_2^n = \frac{1}{c^n\left(1 + \tfrac{1}{4}(e)^{\theta^n/4}\right)} \left(\frac{(U^n)^{\theta^n} \ln U^n}{1 - (U^n)^{\theta^n}}\right), \tag{30}$$

and

$$f(\theta^n) = \left(1 + \tfrac{1}{4}(e)^{\theta^n/4}\right) - (U^n)^{\theta^n} \left(\left(1 + \tfrac{1}{4}(e)^{\theta^n/4}\right) + \frac{\ln U^n}{z_2^n c^n}\right) = 0. \tag{31}$$

This equation is well behaved and Newton's method can be employed to obtain θ^n.

Similarly, if the third constraint is active and the rest of them are free we obtain the following relations.

$$z_i^N = c_i = 0, \qquad i = 1, 2,$$

$$z_i^n = 0, \qquad n = 1, 2, \ldots, N,$$

$$i = 1, 2,$$

$$z_3^n w^n \left((e)^{\theta^n}(e)^{\theta^n/4}\right) = \frac{(U^n)^{\theta^n} \ln U^n}{1 - (U^n)^{\theta^n}}, \tag{32}$$

$$z_3^n = \frac{1}{w^n\left((e)^{\theta^n/4} + \frac{1}{4}\theta^n(e)^{\theta^n/4}\right)} \left(\frac{(U^n)^{\theta^n} \ln U^n}{1 - (U^n)^{\theta^n}}\right) , \tag{33}$$

$$f(\theta^n) = (e)^{\theta^n/4}\left(1 + \frac{1}{4}\theta^n\right) - (U^n)^{\theta^n}\left((e)^{\theta^n/4}\left(1 + \frac{1}{4}\theta^n\right) + \frac{\ln U^n}{z_3^n w^n}\right) = 0. \tag{34}$$

This is again a well behaved equation and can be solved by Newton's method.

4. Numerical Results

A five stage problem was solved with the constants given in Table 1. The optimum redundancy obtained is as follows.

$\theta^1 = 2.6000,$

$\theta^2 = 2.2816,$

$\theta^3 = 2.0075,$

$\theta^4 = 2.6882,$

$\theta^5 = 3.3981,$

Since θ^n, $n = 1, 2, \ldots, 5$, in reality, should be positive integers, we approximately obtain

$\theta^1 = 3,$

$\theta^2 = 2,$

$\theta^3 = 2,$

$\theta^4 = 3,$

$\theta^5 = 3.$

The number of redundant elements at each stage can be obtained by subtracting one from each of the above figures.

Table 1. Constants assigned for 5 stage problem.

n	R^n	p^n	P	c^n	C	w^n	W
1	.80	1		7		7	
2	.85	2		7		8	
3	.90	3	110	5	175	8	200
4	.65	4		9		6	
5	.75	2		4		9	

From the result we find that the total of the product of weight and volume is 83 with a slack of 27 units, the cost of the system is 146.12 with a slack of 28.88 units and the weight of the system is 192.48 with a slack of 7.52 units. This policy results in a system reliability of 0.9045. A numerical simulation indicated that the above result is not significantly different from the true optimum.

5. Conclusion

A simple and practical computational procedure is presented for maximizing the reliability of a system under multiple nonlinear constraints. An example with three nonlinear constraints is solved in detail to illustrate the method. Problems with multiple linear constraints are special cases of the problems presented here.

The objective function given by equation (4), that is, the logarithm of the system reliability given by equation (1), and the functions representing the constraints given by equation (2) are separable functions. Therefore, the present method may be applied, in general, to optimization of problems with separable objective and constraint functions.

In applying the above technique it is assumed that the optimal sequence of θ^n is obtained by using the recurrence relation given by equation (13) when only the jth constraint is active. In the computational procedure only the necessary condition for local optimality is used to obtain the candidate for the optimal solution. Therefore, simulation is involved to assure numerically the sufficiency of the optimal solution. In spite of this shortcoming the present method appears to overcome some of the practical limitations of other methods used to solve this class of problems.

REFERENCES

1. L.T. Fan, C.S. Wang, F.A. Tillman, and C.L. Hwang, "Optimization of Systems Reliability", IEEE Transactions on Reliability, vol. R-16, pp. 81-86 (1967).

2. L.T. Fan, and C.S. Wang, The Discrete Maximum Principle - A Study of Multistage Systems Optimization, Wiley, New York, 1964.

3. L.T. Fan, The Continuous Maximum Principle - A Study of Complex Systems Optimization, New York, Wiley, 1966.

4. F.A. Tillman, and J. Liittschwager, "Integer Programming Formulation of Constrained Reliability Problems," Management Science, vol. 13, pp. 887-899 (1967).

3.4 SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE (SUMT) APPLIED TO OPTIMAL SYSTEM RELIABILITY

1. Introduction

The problems considered in this section are optimization of system reliability of a complex system. The optimization method employed is the sequential unconstrained minimization technique (SUMT). This method is considered as one of the simplest and the most efficient methods for solving the constrained nonlinear programming problems.

The principle of the sequential unconstrained minimization technique (SUMT) is a transformation of a constrained minimization problem into a sequence of unconstrained minimization problems. This transformation enables us to use well developed unconstrained optimization techniques to solve the constrained problem without inventing a new technique for such a constrained optimization problem. The method was first proposed by Carroll in 1959 [1,2] and further developed by Fiacco and McCormick [3,4,5,6,12]. In 1964, Fiacco and McCormick developed a general algorithm based on SUMT, and in 1965, they proposed a method which is called SUMT without parameters. By using this method, the difficulty of choosing the penalty parameters can be avoided, although some difficulties still exist. There is a general computer program provided by McCormick, Mylander and Fiacco called "RAC Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming," (IBM SHARE number 3189) [12]. In this computer program, the unconstrained minimization technique used is the second order gradient method.

Difficulties which arise from use of the second order gradient method as a unconstrained minimization technique in SUMT become predominant in a large size and/or very complex nonlinear problem. The difficulties arise particularly in taking correctly the first order and second order partial derivatives of very complex nonlinear functions which most practical problems have. Therefore, a new algorithm which uses a much simpler direct search technique is very desirable.

For the above reason, a new technique of implementing SUMT with Hooke and Jeeves pattern search technique as its unconstrained minimization process is developed [9,11]. The procedures are presented in [9] in details. Hooke and Jeeves pattern search technique [7,8] is different from the gradient method in the decision making process used to decide the direction of search. The direction of search in the gradient method is in the steepest decent direction while that of the Hooke and Jeeves pattern search technique is determined by direct comparison of the values of the objective function at two points separated from each other for a finite step. For this reason, when the pattern search is getting close to the boundary of some inequality constraints, it will frequently go out of the feasible region bounded by inequality constraints, and the search might be terminated at some point near the boundary which might not be the real constrained optimum. A heuristic programming technique developed by Paviani and Himmelblau [13] is used, which enables one to make turns at the pattern search near the boundary of constraints. The details of the method are described and a general FORTRAN-IV program together with detailed computer diagrams is presented in [9].

The optimization of the complex system reliability by using RAC SUMT computer program has been carried out in [11,4], and by using LAI SUMT computer program in [9,11]. In this section, the same complex system problems but an improved cost function [15] for each component have been solved by using LAI SUMT program.

2. Formulation of the Problem

A system whose redundant units are not in a purely series configuration is considerably more difficult to optimize. One such example is shown in Fig. 1. In the system, unit 1 is backed up by a parallel unit 4. There

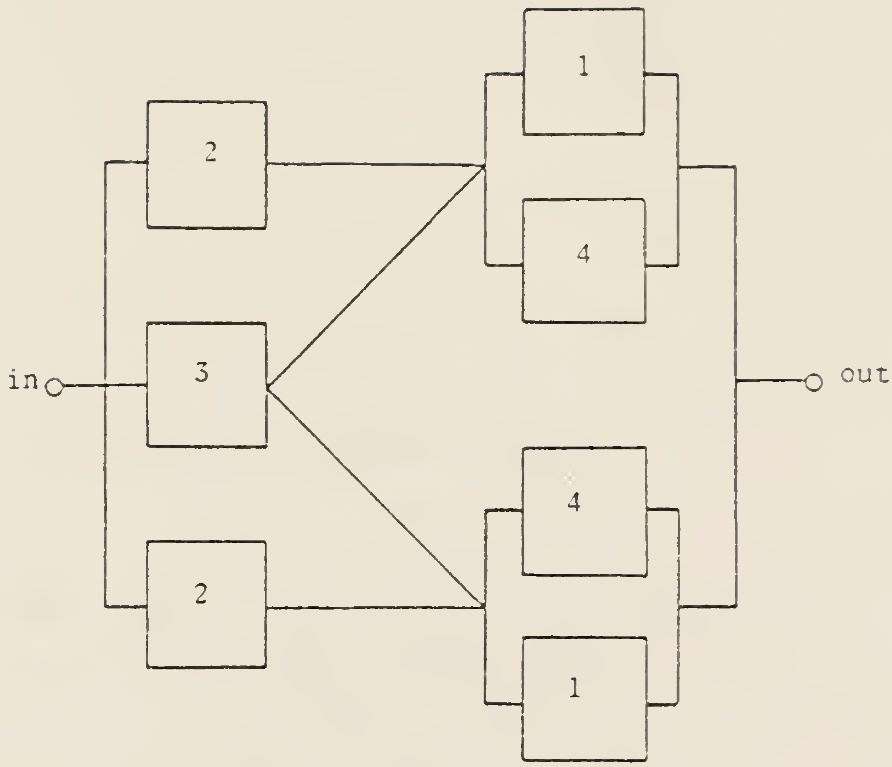


Fig. 1 A schamatic diagram of a complex system

are two equal paths, each of which has unit 2 in series with the stage formed by units 1 and 4. These two equal paths operate in parallel so that if at least one of them is good the output is assured. However, because unit 2 does not have a high degree of reliability, a third unit, unit 3, is inserted into the circuit. Therefore, the following operations are possible: 2-1, 2-4, 3-1, and 3-4, and each operation has two equal paths.

In attempting to optimize the reliability of a system with such a configuration, a major difficulty is encountered in that the reliability expression is not a separable function and thus cannot be analyzed as a multistage process. Hence a different approach is used to solve this type of the problem where the reliability is obtained by Bayes' theorem, which utilizes conditional probabilities [16]. With this in mind a formula for the nonlinear system reliability, subject to some constraints is formulated. A nonlinear programming problem of optimizing the system reliability based on the model is then solved by the SUMT. This method appears to be one of the more efficient methods of solving constrained nonlinear optimization problems.

SYSTEM RELIABILITY USING CONDITONAL PROBABILITIES

In a complex system where the redundant units are not in a purely parallel or series configuration the reliability can be evaluated by using Bayes' theorem of conditional probabilities.

In solving this problem, a simplified form of Bayes' probability thereom is used. The theorem states that if A is an event that depends on one or two mutually exclusive events B_i and B_j of which one must necessarily occur, then the probability of the occurrence of A is given by

$$P(A) = P(A, \text{ given } B_i) \cdot P(B_i) + P(A, \text{ given } B_j) \cdot P(B_j) \qquad (1)$$

Let Q_s represent the probability of system failure, R_k the probability that component K is good, and Q_k the probability that component K is bad. Then we obtain the following expression for system unreliability,

$$Q_s = Q_s \text{ (given K is good)} \cdot R_k + Q_s\text{(given K is bad)} \cdot Q_k. \qquad (2)$$

The corresponding system reliability R_s is

$$R_s = 1 - Q_s, \qquad (3)$$

To obtain the reliability of the system presented in Fig. 1 we select component 3 as the key component in (2), denoted by K. Thus we have the expression for system unreliability

$$Q_s = Q_s \text{ (if 3 is good)} \cdot R_3 + Q_s\text{(if 3 is bad)} \cdot Q_3 \qquad (4)$$

If component 3 is good, the system can fail if the stage formed by units 1 and 4 fails. Thus, the system's unreliability, given that unit 3 is good, is

$$Q_s\text{(if 3 is good)} = [(1 - R_1)\ (1 - R_4)]^2 \qquad (5)$$

If, on the other hand, unit 3 is bad the system's unreliability is

$$Q_s \text{ (if 3 is bad)} = \{1 - R_2[1 - (1 - R_1)\ (1 - R_4)]\}^2 \qquad (6)$$

From (4) the unreliability of the system is

$$Q_s = [(1 - R_1)\ (1 - R_4)]^2 \cdot R_3$$
$$+ \{1 - R_2[1 - (1 - R_1)\ (1 - R_4)]\}^2 \cdot (1 - R_3) \qquad (7)$$

The assumption is made that the reliability of the components are independent of each other. That is, for example, the reliability of component 4 would not be affected by the failure of component 1. The system reliability is then given by (3).

3. Computational Procedures of SUMT

The general nonlinear programming problem with nonlinear inequality and/or equality constraints is to choose x to

minimize $f(x)$ (8)

subject to

$$g_i(x) \geq 0, \; i = 1, \ldots, m;$$

$$h_j(x) = 0, \; j = 1, \ldots, \ell.$$

The SUMT technique for solving (8) is based on minimizing the function

$$P(x, r_u) = f(x) + r_u \sum_{i=1}^{m} [g_i(x)]^{-1} + r_u^{-\frac{1}{2}} \sum_{j=1}^{\ell} h_j^2 (x) \qquad (9)$$

over a strictly monotnic decreasing sequence $\{r_u\}$. $P(x, r_u)$ is minimized with respect to $x(r_u)$ for a given value of r_u. The sequence of values of $\{P(x,r_u)\}$ converges to the constrained optimum value of the original objective function, $f(x)$, as $\{r_u\} \to 0$. The essential requirement is the convexity of the P-function. Mathematical proof of the convergence of the method is given in [6].

The new algorithm for implementing SUMT by the Hooke and Jeeves pattern search and heuristic programming is summarized below [9,11].

Step 1 Select a starting point x^0, the initial value of the penalty coefficient r^0, the initial tolerance limit of the violation to constraints B^0, and the initial step-sizes d^0 needed in the searches.

Step 2 Go to step (3) if x^0 is feasible (viz., inside the region bounded by the inequality constraints). Otherwise select a feasible starting point by minimizing the total weight of violation. The total weight of violation, TGH, is defined by

$$(TGH)^2 \equiv \sum_{\tau\varepsilon T} g_\tau^2(x^0) + \sum_{s\varepsilon S} h_s^2(x^0)$$

where $T \equiv \{t | g_t(x^0) < 0\}$, $S \equiv \{s | h_s(x^0) = 0\}$. TGH includes only the violated constraints.

Step 3 Minimize P in (9) by the Hooke and Jeeves pattern search technique. Check after every move: if the move goes outside the feasible region, go to step 4; otherwise, after x^* is reached for the current r_u, go the step 5.

Step 4 Move back to the near-feasible region and then return to step 3. The near-feasible region is defined as the region where all points satisfy the condition: TGH < B, where B is the tolerance limit of violation; B is sequentially decreased after every violation to the inequality constraints.

Step 5 Check if the x* obtained in step 3 is feasible. If x* is feasible, go the step 7; otherwise go to step 6.

Step 6 Move x* (in the infeasible region) back into the feasible region along the direction toward the last optimum point; then go to step 7.

Step 7 Check if a stopping criterion such as

$$\left| \left| \frac{f(x^*)}{G(x^*, r_u)} \right| - 1 \right| < \varepsilon \tag{10}$$

is satisfied. The solution is the optimal one if the criterion is satisfied; otherwise, go to step 8. $G(x, r_u)$ in (9) is defined as [1].

$$G(x, r_u) \equiv f(x) - r_u \sum_{i=1}^{m} [g_i(x)]^{-1} + r_u^{-1/2} \sum_{j=1}^{\ell} h_j^2(x) \tag{11}$$

Step (8) Set $u \leftarrow u + 1$; $r_{u+1} \leftarrow r_u/C$, where C is a constant greater than 1; and $d^{u+1} \leftarrow d^0/(u + 1)$. Return to step 3.

The flow diagram of SUMT with Hooke and Jeeves pattern search technique is shown in Fig. 2. The detailed discussions about "procedure for selecting a feasible starting point from the infeasible initial point", "Computational procedure for minimizing $P(\bar{x}, r_k)$ function by the Hooke and Jeeves pattern search", "procedure for moving an infeasible point into the feasible or near-feasible region bounded by inequality constraints", and "Procedure for moving

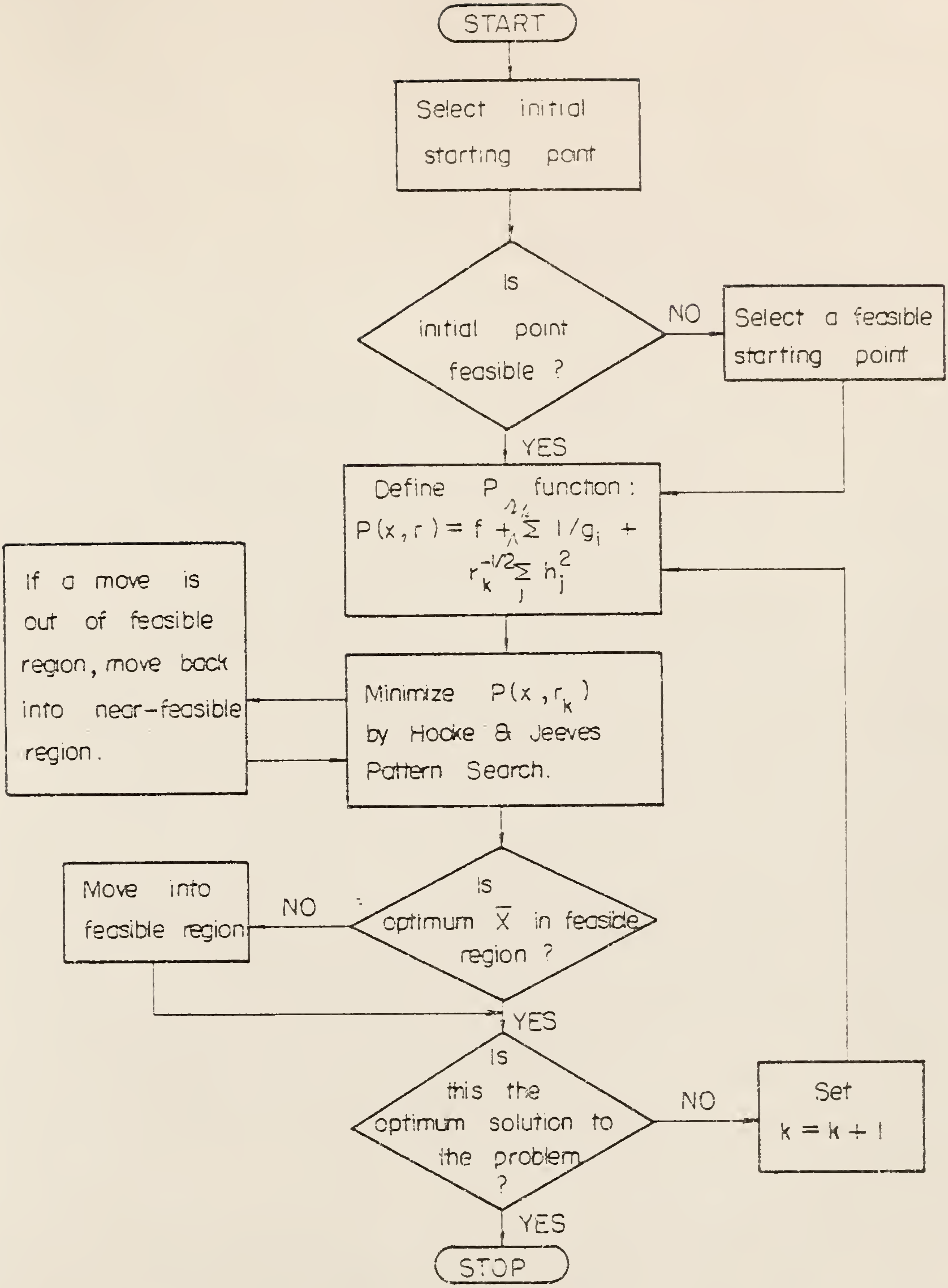


Fig. 2. Descriptive flow diagram for SUMT with Hooke and Jeeves Pattern Search.

the near-feasible k-th sub-optimum into the feasible region" are all referenced in Lai [11].

4. Numerical Examples

Example 1

The problem of maximizing the reliability of the complex system given in Fig. 1, and which is subject to a single constraint can be stated as follows by using (3) and (7).

Maximize the system reliability

$$R_s = 1 - Q_s$$
$$= 1 - R_3[(1 - R_1)(1 - R_4)]^2$$
$$-(1 - R_3)\{1 - R_2[1 - (1 - R_1)(1 - R_4)]\}^2 \tag{12}$$

subject to

$$C_s = \sum_i C_i \leq C, \tag{13}$$
$$R_i \geq R_{i,min}.$$

where

$$C_i \equiv K_i R_i^{\alpha_i}. \tag{14}$$

The constraint given by (13) can be interpreted as follows. C_i can represent the weight, cost, or volume of each unit or component of the system, and the total weight, cost, or volume of the system must be less than C. The weight, cost, or volume of each unit or component of the system is a function of reliability that can be expressed by (14) where K_i is a proportionality constant and α_i the exponential factor that related C_i and the reliability. That is, K_i is the weight, cost, or volume of the component when R ≡ 1 and $K_i R_i^{\alpha_i}$ is the reduced cost, weight, or volume when $R_i < 1$. Usually α_i is less than one. The following values are assigned to the constants K_1, K_2, K_3, and K_4, the constraint C and the exponential constant α_i, $1 = 1,2,3,4$.

$$K_1 \equiv 100, \qquad K_2 \equiv 100, \qquad K_3 \equiv 200, \qquad K_4 \equiv 150,$$

$$C \equiv 800, \qquad \alpha_i \equiv 0.6, \qquad i = 1,2,3,4.$$

The problem is formulated in SUMT format as follows.

Minimize

$$\begin{aligned} f(x) &\equiv -R_s \\ &\equiv -1 + R_3[(1 - R_1)(1 - R_4)]^2 + (1 - R_3) \\ &\quad + \{1 - R_2[1 - (1 - (1 - R_1)(1 - R_4)]\}^2 \end{aligned}$$

subject to the constraints

$$g_1(x) \equiv C - (2K_1R_1^{\alpha_1} + 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4}) \geq 0$$

$$g_{i+1}(x) \equiv (1 - R_i) \geq 0, \qquad i = 1,2,3,4$$

$$g_{i+5}(x) \equiv R_i - R_{i,min} \geq 0 \qquad i = 1,2,3,4.$$

The P function of (9) is

$$P(x,r_k) \equiv -1 + R_3[(1 - R_1)(1 - R_4)]^2 + (1 - R_3)\{1 - R_2[1 - (1 - R_1)(1 - R_4)]\}^2$$

$$+ r_k \left\{ \frac{1}{C - (2K_1^{\alpha_1} - 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4})} + \sum_{i=1}^{4} \left(\frac{1}{1 - R_i} + \frac{1}{R_i - R_{i,mi}} \right. \right.$$

The optimal solutions obtained from two sets of different starting components reliabilities, namely, $[R_1, R_2, R_3, R_4,] = [0.7, 0.7, 0.7, 0.7]$ and $[R_1, R_2, R_3, R_4] = [0.6, 0.6, 0.6, 0.6]$, are presented in Table 1 together with the corresponding results obtained by RAC program [12]. The solutions are almost identical, that is, the optimal system reliability, R_s, of 0.999998 with the cost of 799.733 for the first set of starting components reliabilities, and the optimal system reliability, R_s, of 0.999997 with the cost of 799.908 for the second set of starting components reliabilities are obtained. Recall

Table 1. Comparison of the Optimal Solutions of the System Reliability Maximization Problem (Example 1)

Program	Number of k iterated	Component reliability R_1	R_2	R_3	R_4	System Reliability	Cost	Stopping criteria for each	Stopping criteria for final	Computing time
RAC	0	0.7	0.7	0.7	0.7					exceeds 20 min.
								$\varepsilon'=10^{-5}$	$\varepsilon=10^{-4}$	
	10	0.9876	0.9936	0.6972	0.6941	0.99996	799.78			
Program	0	0.6	0.6	0.6	0.6					exceeds 20 min.
	11	0.9889	0.9921	0.7019	0.6886	0.99995	799.28	$\varepsilon'=10^{-5}$	$\varepsilon=10^{-4}$	
New	0	0.7	0.7	0.7	0.7					(both problems
	12	0.997626	0.998399	0.682652	0.694958	0.999998	799.733	INCUT=3	$\varepsilon=10^{-5}$	together)
Program	0	0.6	0.6	0.6	0.6					
	12	0.997409	0.998117	0.702590	0.682817	0.999997	799.908	INCUT=3	$\varepsilon=10^{-5}$	90.4 sec.

that the constraint on the cost is 800. The optimal components reliabilities are almost the same for the both starting sets of the starting points. The stopping criterion for terminating the minimization of the P function at each k iteration is that terminating when the number of cut-down step-size operations in the Hooke and Jeeves pattern search is 3, and the final stopping criterion for terminating the problem is $\varepsilon = 10^{-4}$. For the first set of starting points, it takes 12 iterations for P functions, k = 12, with totally 1192 f-functional values evaluated. And for the second set, 12 iterations for P functions, K = 12, with totally 1194 f-functional values evaluated.

Tables 2a and 2b present the iteration results converging to the optimal solution. Results given in these tables show that the system reliability, R_s, is monotonically increasing as iteration k increases. The value of P function approaches to that of f function ($= -R_s$) as the iteration proceeds. Thus the minimization of P function will eventually lead us to the minimization of f function.

The values of r_0 used in Tables 2a and 2b are determined by

$$f(x_0) = r_0 \sum_i \frac{1}{g_i(x_0)} \tag{15}$$

where x_0 is the initial point. The basis for this selection procedure is to render the value of the penalty of the constraints to be approximately the same order of magnitude as the value of the f-function at the starting point in the P-function formulation

$$P(x_0,r_0) = f(x_0) + r_0 \sum_i \frac{1}{g_i(x_0)}$$

Example 2

This example is to find the optimal component re minimize the cost of the system, i.e., [15].

Minimize

$$C_s = \sum_{i=1}^{4} K_i [\tan(\frac{\pi}{2} R_i)]^{\alpha_i}$$

Table 2a. Computer Results of the System Reliability Maximization Problem

[Start at $R_i = 0.6$, for all i]

Iteration k	Times of f-value calculated at each iteration	Value of r_k	R_1	R_2	R_3	R_4	$-P$	$-f$ $(= R_s)$	Cost
0		2.214×10^{-2}	0.6	0.6	0.6	0.6	0.6647	0.8862	662.4
1	70	2.214×10^{-2}	0.6200	0.7150	0.5850	0.6175	0.677501	0.924867	683.298
2	68	5.535×10^{-3}	0.7900	0.7900	0.6600	0.6750	0.88815	0.970493	753.431
3	59	1.384×10^{-3}	0.8700	0.8700	0.7100	0.70833	0.991246	0.991240	776.258
4	89	3.459×10^{-4}	0.872493	0.911250	0.791250	0.736458	0.986439	0.996132	796.919
5	72	8.648×10^{-5}	0.904999	0.942749	0.767749	0.722707	0.994712	0.997952	798.981
6	174	2.162×10^{-5}	0.944567	0.968158	0.730748	0.707252	0.997960	0.999207	798.889
7	202	5.405×10^{-6}	0.973031	0.980765	0.715331	0.693293	0.999216	0.999740	798.831
8	129	1.351×10^{-6}	0.983419	0.988173	0.710188	0.688561	0.999691	0.999898	799.273
9	110	3.378×10^{-7}	0.989665	0.992536	0.704067	0.687510	0.999878	0.999960	799.504
10	85	8.446×10^{-8}	0.993554	0.995303	0.703067	0.685176	0.999952	0.999984	799.680
11	76	2.111×10^{-8}	0.996045	0.997200	0.702590	0.683271	0.999981	0.999994	799.730
12	60	5.279×10^{-9}	0.997469	0.998117	0.702590	0.682817	0.999992	0.999997	799.908

Table 2b. Computer Results of the System Reliability Maximization Problem

[Start at $R_i = 0.7$, for all i]

Iteration k	Times of f-value calculated at each iteration	Value of r_k	R_1	R_2	R_3	R_4	-P	-f (= R_s)	Cost
			0.7	0.7	0.7	0.7	0.7161	0.9548	726.6
1	68	1.788×10^{-2}	0.640000	0.730000	0.610000	0.632500	0.726307	0.936015	695.180
2	100	4.471×10^{-3}	0.726250	0.816250	0.711250	0.783750	0.906109	0.992678	753.053
3	64	1.118×10^{-3}	0.816250	0.876250	0.791250	0.783750	0.967770	0.992678	790.870
4	149	2.794×10^{-4}	0.873124	0.922249	0.766874	0.744843	0.988329	0.996569	797.342
5	38	6.986×10^{-5}	0.881124	0.927249	0.763874	0.750843	0.994753	0.996927	798.672
6	126	1.747×10^{-5}	0.911037	0.949323	0.740436	0.736226	0.997489	0.998211	799.632
7	232	4.366×10^{-6}	0.969679	0.980016	0.692256	0.715007	0.999295	0.999699	799.637
8	115	1.092×10^{-6}	0.983835	0.989126	0.657223	0.702855	0.999731	0.999911	799.305
9	94	2.729×10^{-7}	0.990263	0.993277	0.685080	0.698569	0.999894	0.999965	799.322
10	68	6.822×10^{-8}	0.993763	0.995777	0.684080	0.697069	0.999958	0.999986	799.593
11	69	1.706×10^{-8}	0.996263	0.997206	0.682652	0.695640	0.999983	0.999994	799.568
12	69	4.264×10^{-9}	0.997626	0.998399	0.662652	0.694958	0.999993	0.999998	799.733

subject to the constraints

$$R_{s,min.} \leq 1 - R_3[(1 - R_1)(1 - R_4)]^2 - (1 - R_3)\{1 - R_2[1 -(1 - R_1)(1 - R_4)]\}$$

$$R_{i,min.} \leq R_i \leq 1.0$$

The numerical values of parameters are

$$K_1 = 25. \quad ,K_2 = 25. \quad ,K_3 = 50. \quad ,K_4 = 37.5$$

$$R_{i,min.} = 0.50, \qquad \alpha_i = 1.0, \qquad \text{for } i = 1,2,3,4.$$

$$R_{s,min.} = 0.99$$

The cost function suggested in (16) satisfies the following basic requirements, especially when the reliability of each component, R_i, is greater than 0.50.

1. Cost of a low reliability component is very low.
2. Cost of a high reliability component is very high
3. Cost is a monotone increasing function of reliability.
4. Derivative of cost (with respect to reliability) is a monotone increasing function of reliability.

The problem is formulated in SUMT format as follows:

Minimize

$$f(x) = C = \sum_{i=1}^{4} K_i[\tan(\frac{\pi}{2} R_i)]^{\alpha_i}$$

subject to

$$g_1(x) = 1 - R_3[(1 - R_1)(1 - R_4)]^2 - (1 - R_3)\{1 - R_2[1 -(1 - R_1)(1 - R_4)]\}^2$$

$$-R_{x,min.} \geq 0$$

$$g_{i+1} = R_i - R_i, \text{ min.} \geq 0 \qquad ,i=1,2,3,4$$

$$g_{i+5} = 1.0 - R_i \geq 0 \qquad ,i=1,2,3,4$$

The P function for this problem is

$$P(x,r_k) = f(x) + r_k \sum_i 1/g_i(x)$$

$$= \sum_{i=1}^{4} K_i[\tan(\frac{\pi}{2} R_i)]^{\alpha_i} + r_k \left\{ \frac{1}{1 - R_3[(1 - R_1)(1 - R_4)]^2 - (1 - R_3)} \right.$$

$$\frac{}{\{1 - R_2[1 -(1 - R_1)(1 - R_4)]\}^2 - R_{s,min.}} + \sum_{i=1}^{4} \frac{1}{R_1 - R_{i,min.}}$$

$$\left. + \sum_{i=1}^{4} \frac{1}{1.0 - R_i} \right\}$$

where x is the row vector of (R_1, R_2, R_3, R_4).

For this problem, the RAC program fails to satisfy the special requirement that the violable non-negativity constraints should never be violated during the search. The results obtained by applying the new developed program are presented in Tables 3, 4a and 4b.

The optimal solutions obtained from two sets of different starting components reliabilities, namely, $[R_1, R_2, R_3, R_4] = [0.6, 0.6, 0.6, 0.6]$ and $[R_1, R_2, R_3, R_4] = [0.5, 0.5, 0.5, 0.5]$ are presented in Table 3. The solutions are almost identical, that is, the optimal minimum cost, C, of 394.806 with the system reliability, R_s, of 0.990311 for the first set of starting components reliabilities, and the optimal minimum cost, C, of 397.879 with the system reliability, R_s, of 0.990406 for the second set of starting components reliabilities are obtained. Recall that the constraint on the system reliability is 0.99. The optimal components reliabilities are almost the same for both starting sets. The stopping criterion for terminating minimization of the P function at each iteration is that terminating when the number of cut-down step-size operations is 4. And the final stopping criterion for terminating the problem is $\varepsilon = 10^{-2}$. For the first set of starting points, it takes 10 iterations for P functions, k = 10, with totally

Table 3. Optimal Solution of the Cost Minimization Problem (Example 2)

Iteration of γ_k k	Iteration of f n	Values of Component Reliabilities R_1	R_2	R_3	R_4	System Reliability R_s	Cost	Stopping Criteria for each k 'INCUT'	for final ε 'THETA'
0		0.6	0.6	0.6	0.6	0.8862	189.3		
	1135							4	10^{-2}
10		0.825758	0.886685	0.645749	0.739332	0 990311	394.806		
0		0.5	0.5	0.5	0.5	0.7997	137.5		
	2124							4	10^{-2}
11		0.834409	0.879062	0.667413	0.741163	0.990406	397.879		

Table 4a Computer results of the cost function minimization problem (Example 2)

[Start at $R_i = 0.6$, for all i]

Iteration k	Times of f-value calculated at each iteration	Value of r_k	R_1	R_2	R_3	R_4	P	f (= cost)	R_s
			0.6	0.6	0.6	0.6	596600	189.3	0.8862
0			0.88	0.88	0.88	0.88	1737000	720.8	0.993028
1	128	10000	0.879375	0.932500	0.890000	0.880000	1607460	848.322	0.999092
2	44	2000	0.879375	0.932500	0.890000	0.878750	322169	846.247	0.999085
3	60	400	0.880000	0.932500	0.887500	0.876250	65107.1	836.419	0.999060
4	68	80	0.878750	0.932500	0.873750	0.871250	13673.5	796.267	0.998937
5	100	16	0.876250	0.930000	0.835000	0.848750	3325.04	696.863	0.998447
6	110	3.2	0.864375	0.920000	0.771875	0.809375	1156.76	568.441	0.9970269
7	143	0.64	0.849062	0.907500	0.710625	0.773125	637.137	477.356	0.994747
8	136	0.128	0.837187	0.898750	0.670625	0.750625	483.416	430.118	0.992646
9	128	0.0256	0.827500	0.893750	0.651875	0.743125	428.820	408.365	0.991314
10	218	0.00512	0.825758	0.886685	0.645749	0.739332	411.440	394.806	0.990311

Table 4b Computer results of the cost function minimization problem (Example 2)

[Start at $R_i = 0.5$, for all i]

Iteration k	Times of f-value calculated at each iteration	Value of r_k	R_1	R_2	R_3	R_4	P	f (=cost)	R_s
			0.5	0.5	0.5	0.5	4×10^{53}	137.5	0.7997
0			0.82	0.82	0.80	0.80	5.483×10^{7}	441.3	0.990738
1	159	5000	0.878750	0.932500	0.890000	0.880000	8.042×10^{5}	847.634	0.999089
2	68	1000	0.879375	0.932500	0.888750	0.878750	1.615×10^{5}	842.963	0.999078
3	76	200	0.879375	0.932500	0.883750	0.876250	3.297×10^{4}	826.498	0.999032
4	107	40	0.878750	0.931250	0.862500	0.863750	7225.49	760.683	0.998791
5	115	8	0.871875	0.926250	0.810000	0.833750	1994.90	640.217	0.997972
6	166	1.6	0.857500	0.914375	0.743750	0.792500	849.602	522.950	0.996097
7	211	0.32	0.843125	0.903750	0.688125	0.761875	550.085	452.783	0.993768
8	303	0.064	0.828108	0.885363	0.673714	0.747464	467.363	406.040	0.991086
9	305	0.0128	0.834409	0.879062	0.667413	0.741163	429.843	397.879	0.990406
10	307	0.00256	0.834409	0.879062	0.667413	0.741163	404.272	397.879	0.990406
11	307	0.000512	0.834409	0.879062	0.667413	0.741163	399.158	397.879	0.990406

1135 f-functional values calculated, and for the second set, 11 iterations for P functions, k = 11, with totally 2124 f-functional values calculated.

Results given in Tables 4a and 4b show that the cost of the system, C, is monotonically decreasing as iteration k increases. The value of the p function approaches to that of the f function (=C) as the iteration proceeds. Thus the minimization of the P function will eventually lead us to the minimization of f function.

Again, the values of r_0 are determined from eg. (15) as explained in Example 1.

It is worth noting that the starting points $R^0 = [R_1, R_2, R_3, R_4] = [0.6, 0.6, 0.6, 0.6]$ and $R^0 = [R_1, R_2, R_3, R_4] = [0.5, 0.5, 0.5, 0.5]$ in Table 4a and Table 4b are in infeasible region. The system reliability given by $R^0 = [0.6, 0.6, 0.6, 0.6]$ is 0.8862 and by $R^0 = [0.5, 0.5, 0.5, 0.5]$ is 0.7997, both of which are less than $R_{s,min.}$, of 0.99. Therefore, before the P-function minimization routine is started, a new feasible point is searched first. The point [0.88, 0.88, 0.88, 0.88] in the second row of Table 4a is thus selected and is used as the feasible starting point to start the minimization procedure. Also, the point [0.82, 0.82, 0.80, 0.80] is selected and used as feasible starting point for Table 4b.

Example 3

To demonstrate the technique, the five-stage reliability problem is solved. The problem is

Maximize

$$R_s = \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=1}^{5} p_j (x_j)^2 \leq P$$

$$g_2 = \prod_{j=1}^{5} c_j (x_j + \exp (x_j/4)) \leq C$$

$$g_3 = \sum_{j=1}^{5} w_j x_j \exp (x_j/4) \leq W$$

where $x_j \geq 1$, j=1, 2, ..., 5, are integers.

The constants associated with the five-stage problem are

j	R_j	p_j	P	c_j	C	w_j	W
1	0.80	1		7		7	
2	0.85	2		7		8	
3	0.90	3	110	5	175	8	200
4	0.65	4		9		6	
5	0.75	2		4		9	

The problem is formulated in SUMT format as follows:

Minimize

$$Z = f(x) = -R_s$$

$$= - \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1' = P - \sum_{j=1}^{5} p_j (x_j)^2 \geq 0$$

$$g_2' = C - \sum_{j=1}^{5} c_j (x_j + \exp (x_j/4)) \geq 0$$

$$g_3' = W - \sum_{j=1}^{5} w_j x_j \exp (x_j/4) \geq 0$$

$$g_{j+3}' = x_j - 1 \geq 0 \quad , \qquad j=1, 2, \ldots, 5$$

$$g_9' = Z + 1 \geq 0$$

The P function for this problem is

$$P(x, \gamma_k) = f(x) + \gamma_k \sum_{i=1}^{9} \frac{1}{g_i} (x)$$

$$= - \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}] + \gamma_k [\frac{1}{P - \sum_{j=1}^{5} p_j (x_j)^2}$$

$$+ \frac{1}{C - \sum_{j=1}^{5} c_j (x_j + \exp (x_j/4))} + \frac{1}{W - \sum_{j=1}^{5} w_j x_j \exp (x_j/4)}$$

$$+ \sum_{j=1}^{5} \frac{1}{x_j - 1} + \frac{1}{Z + 1}]$$

where x is the row vector of $(x_1, x_2, x_3, x_4, x_5)$, each of the components are assumed continuous variables.

The optimal solutions obtained from the starting components used at each stage, namely, $(x_1, x_2, x_3, x_4, x_5) = (2, 2, 2, 2, 3)$, are presented in Table 5. The stopping criterion for terminating the minimization of the P function at each k iteration is when the number of cut-

Table 5 Computer Results of the System Reliability Optimization Problem (Example 3)

Iteration k	Times of f-value calculated at each iteration	Value of T_k	x_1	x_2	x_3	x_4	x_5	P	f $(= R_s)$	Slacks g_1	g_2	g_3
0		5000	2.000	2.000	2.000	2.000	2.000	4.811×10^4	0.8025	52.00	35.21	37.22
1	267	2500	2.480	3.080	3.280	1.640	2.000	2.084×10^4	0.7531	33.84	37.28	10.43
2	185	312.5	2.563	3.070	3.230	1.615	1.855	2.602×10^3	0.7393	35.97	38.05	14.05
3	84	39.06	2.557	3.07	3.23	1.615	1.860	3.245×10^2	0.7396	35.96	38.08	14.06
4	83	4.88	2.565	3.065	3.23	1.615	1.865	3.992×10^1	0.7401	35.94	38.02	13.94
5	184	0.610	2.610	3.045	3.200	1.635	1.895	4.339	0.7468	36.04	37.59	13.63
6	195	0.0763	2.748	2.910	3.025	1.795	2.150	0.1314	0.7914	35.93	35.52	12.04
7	302	0.00238	3.016	2.590	2.463	3.011	2.436	0.8728	0.9078	21.15	22.63	0.39
8	300	0.000298	2.691	2.323	2.047	3.521	2.809	0.9175	0.9229	14.03	22.40	0.60

down step-size operations in the Hooke and Jeeves pattern search is 4, and the final stopping criterion for terminating the problem is $\varepsilon = 10^{-3}$. As shown in Table 5 it takes 8 iterations for P functions, k = 8, with totally 1600 f-functional values evaluated. The table also shows that the system reliability, R_s, monotonically increases after iteration 2 as iteration k increases. The value of P function approaches to that of f function ($-R_s$) as the iteration proceeds. Thus the minimization of P function will eventually lead us to the minimization of f function.

Again, the values of r_0 are determined from eq. (15) as explained in Example 1.

The five-stage reliability problem solved by Lai's SUMT gives the optimal system configuration, $(x_1, x_2, x_3, x_4, x_5) = (2.691, 2.323, 2.047, 3.521, 2.809)$. The system reliability with this configuration is 0.9229. However, all x_j, j=1, 2, ..., 5, should be positive integers, therefore the rounding off procedure to the nearest integers is required. Two possible rounding off results may exist, namely,

(A) $(x_1, x_2, x_3, x_4, x_5) = (3, 2, 2, 4, 3)$, and

(B) $(x_1, x_2, x_3, x_4, x_5) = (3, 2, 2, 3, 3)$.

Configuration (A) gives higher system reliability than (B) (because of one more redundancy used at stage 4); but configuration (A) consumes 111 of g_1, which is greater than the available resource, 110. Therefore (A) is not desirable. Under the configuration (B), we calculate $R_s = 0.9045$, $g_1 = 83$, $g_2 = 146.1$ and $g_3 = 194.5$. Configuration (B) is the optimal components used at each stage.

REFERENCES

1. Carroll, C. W., "An Operations Research Approach to the Economic Optimization of a Kraft Pulping Process," Ph.D. Dissertation, Institute of Paper Chemistry, Appletown, Wisc., 1959.

2. Carroll, C. W., "The Created Response Surface Technique for Optimizing Nonlinear Restrained Systems," Operations Research, 9, 169-184 (1961).

3. Fiacco, A. V., and G. P. McCormick, "The Sequential Unconstrained Minimization Technique for Nonlinear Programming: A Primal-Dual Method," Management Sci., 10, 601-617 (1964).

4. Fiacco, A. V., and G. P. McCormick, "Computational Algorithm for the Sequential Unconstrained Minimization Technique for Nonlinear Programming," Management Sci., 10, 601-617 (1964).

5. Fiacco, A. V., and G. P. McCormick, "Extension of SUMT for Nonlinear Programming: Equality Constraints and Extrapolation," Management Sci., 12 (11): 816-829 (1966).

6. Fiacco, A. V., and G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York, 1968.

7. Hooke, R., and T. A. Jeeves, Direct Search Solution of Numerical and Statistical Problems, J. Assoc. Compt. Mach., 8, 212 (1961).

8. Hwang, C. L., L. T. Fan, and S. Kumar, "Hooke and Jeeves Pattern Search Solution to Optimal Production Planning Problems" Report No. 18, Institute for Systems Design and Optimization, Kansas State University, 1969.

9. Hwang, C. L., K. C. Lai, F. A. Tillman, L. T. Fan, "Optimization of System Reliability by the Sequential Unconstrained Minimization Technique", IEEE Trans. on Reliability, Vol. R-24, pp. 133-135 (1975).

10. Hsu, F. T., L. T. Fan and C. L. Hwang, "Sequential Unconstrained Minimization Technique (SUMT) for Optimal Production Planning", paper summitted for publication, 1969.

11. Lai, K. C., "Optimization of Industrial Management Systems by the Sequential Unconstrained Minimization Technique", M.S. Report, Dept. of Industrial Engineering, Kansas State University, 1970.

12. McCormick, G. P., W. C. Mylander, III., and A. V. Fiacco, "RAC Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming," SHARE Number 3189.

13. Paviani, D. A., and D. M. Himmelblau, "Constrained Nonlinear Optimization by Heuristic Programming", Operations Research, Vol. 17, pp. 872-882 (1969).

14. Tillman, F. A., C. L. Hwang, L. T. Fan and K. C. Lai, "Optimal Reliability of a Complex System," IEEE Trans. on Reliability, Vol. R-19, pp. 95-99 (1970).

15. Aggarwal, K. K., and J. S. Gupta, "On Minimizing the Cost of Reliability Systems", IEEE Trans. on Reliability, Vol. R-24, p. 205 (1975).

16. Bazovsky, I., Reliability Theory and Practice, Englewood Cliffs, N.J.: Prentice-Hall, (1961)

3.3 GENERALIZED REDUCED GRADIENT METHOD (GRG) APPLIED TO OPTIMUM SYSTEM RELIABILITY

1. Introduction

The Generalized Reduced Gradient Method (GRG) was proposed by Abadie and Carpentier [1,2]. The method is a generalization of the Wolfe reduced gradient method [8,9], which solves problems having a nonlinear objective function and linear equality constraints. It classifies the variables as dependent and independent ones, and substitutes into the objective function the expressions obtained from the linear equality constraints for the dependent variables, in terms of the independent variables. Thus the original problem reduces to an unconstrained one with reduced dimension. A variety of optimization techniques may now be used. Applying the same concept to a problem with a set of non-linear constraints, complications may be added, but it is possible by using numerical methods to obtain the solution.

The GRG has been studied extensively and coded in FORTRAN by Abadie [3], Abadie and Guigou [4], and Guigou [5,6]. Three generations of programs, namely, GRG 66, GRG 69, and GREG, have been developed. The improved code, GREG, is the outgrowth of the first two codes and promises to remain among the highly regarded nonlinear programming procedures.

The algorithm of the generalized reduced gradient method is presented in Appendix.

2. Numerical Examples

Example 1

The problem of maximizing the reliability of the complex system, given in Fig. 1 and which is subject to a single constraint, can be stated as follows (see Example 1 in the SUMT section): [7]

Maximize

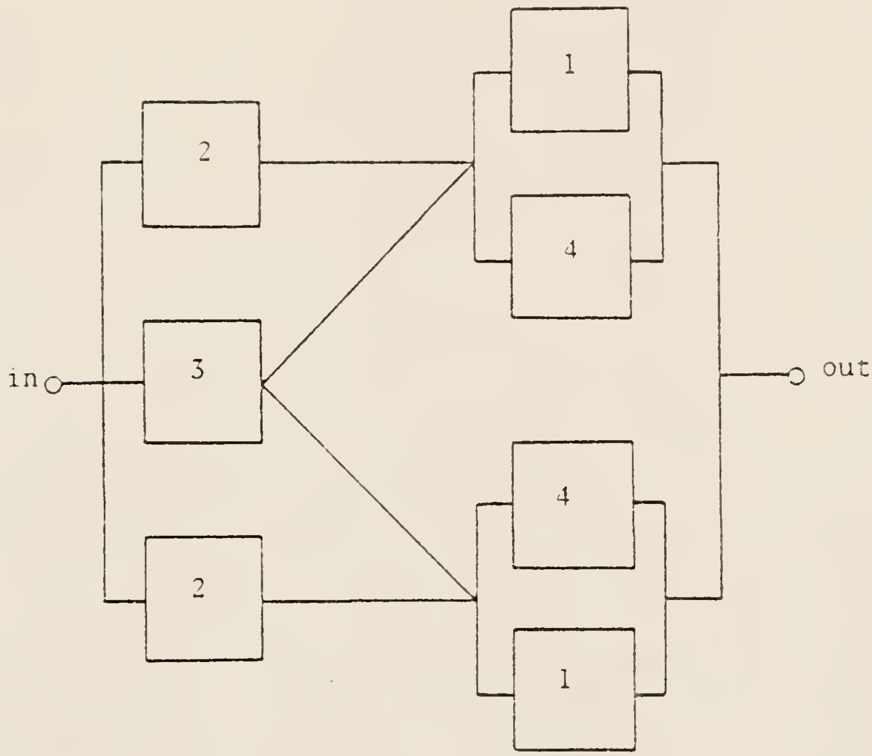


Fig. 1 A schamatic diagram of a complex system

$$R_s = 1 - Q_s$$
$$= 1 - R_3[(1 - R_1)(1 - R_4)]^2 - (1 - R_3)\{1 - R_2[1 - (1 - R_1)(1 - R_4)]\}^2$$

subject to

$$c_s = \sum_{i=1}^{4} c_i \leq C$$

$$R_{i,min} \leq R_i \leq 1.0$$

where

$$c_i = K_i \left[\tan\left(\frac{\pi}{2} R_i\right)\right]^{\alpha_i} \qquad , \; i=1,2,3,4 \quad [10]$$

The numerical values of parameters are

$$k_1=25, \qquad k_2=25, \qquad k_3=50, \qquad k_4=37.5,$$

$$R_{i,min} = 0.50, \quad \alpha_i = 1.0, \quad \text{for } i=1,2,3,4$$

$$C = 800.$$

We now apply the GREG computer code to solve this example. This problem will be reformulated to

Maximize

$$f_0(\bar{x}) = 1 - R_3 [(1-R_1)(1 - R_4)]^2 - (1 - R_3)\{1 - R_2[1 - (1 - R_1)(1 - R_4)]\}^2$$

subject to

$$f_1(\bar{x}) = \sum_{i=1}^{4} k_i\left[\tan\left(\frac{\pi}{2} R_i\right)\right]^{\alpha_i} - 800 \leq 0$$

$$R_{i,min} \leq R_i \leq 1.0 \; , \qquad i=1,2,3,4$$

Then, four external, user-supplied subroutines will be used in which PH1X defines the objective function, CPHI defines the constraint functions, JACOB defines the gradient of the constraint functions, and GRADF1 defines the gradient of the objective function.

By starting with the initial point of $[R_1, R_2, R_3, R_4]^0 = [0.52, 0.52, 0.52, 0.52]$, the solutions are determined to be

$[R_1, R_2, R_3, R_4]^* = [0.902968, 0.948525, 0.813532, 0.851429]$

with the maximum system reliability, R_s, of 0.9990396 and total consumed cost of 799.949076. By starting with the initial point of $[R_1, R_2, R_3, R_4]^0 = [0.60, 0.60, 0.60, 0.60]$, the solutions are determined to be $[R_1, R_2, R_3, R_4]^* = [0.896629, 0.949567, 0.830751, 0.842901]$ with the maximum system reliability, R_s of 0.9990471 and total consumed cost of 799.999023.

<u>Example 2</u>

The numerical example of Example 1 is restated below. The objective is to find the optimal R_i's which minimize the system cost [10]

$$C_s = \sum_{i=1}^{4} K_i \left[\tan\left(\frac{\pi}{2} R_i\right)\right]^{\alpha_i}$$

subject to the constraints

$$R_{s,min} \le 1 - R_3 [(1 - R_1)(1 - R_4)]^2 - (1 - R_3)\{1 - R_2[1 - (1 - R_1)(1 - R_4)]\}^2$$

$$R_{i,min} \le R_i \le 1.0 \quad , i=1,2,3,4$$

The numerical values of parameters are

$k_1 = 25, \quad k_2 = 25, \quad k_3 = 50, \quad k_4 = 37.5$

$R_{i,min} = 0.50, \quad \alpha_i = 1.0, \quad$ for i=1,2,3,4

$R_{s,\ min} = 0.99$

The problem should also be reformulated as

Maximize

$$f_0(\bar{x}) = -\sum_{i=1}^{4} k_i \left[\tan\left(\frac{\pi}{2} R_i\right)\right]^{\alpha_i}$$

subject to

$$f_1(\bar{x}) = -0.01 + R_3[(1 - R_1)(1 - R_4)]^2 + (1 - R_3)\{1 - R_2[1 - (1 - R_1)(1 - R_4)]\}^2$$

$$R_{i,min.} \leq R_i \leq 1.0 \quad , i=1,2,3,4$$

Also, four external user-supplied subroutines, namely, PHIX, CPHI, JACOB, and GRADFI, will be used. By starting with the initial point of $[R_1, R_2, R_3, R_4]^0$ = [0.52, 0.52, 0.52, 0.52], the solutions are determined to be $[R_1, R_2, R_3, R_4]^*$ = [0.827672, 0.891787, 0.634234, 0.732349] with the minimum total cost of 396.85345 and the system reliability of 0.9904930. By starting with the initial point of $[R_1, R_2, R_3, R_4]^0$ = [0.60, 0.60, 0.60, 0.60], the solutions are determined to be $[R_1, R_2, R_3, R_4]^*$ = [0.829047, 0.892711, 0.638432, 0.734509] with the minimum total cost of 400.79110 and the system reliability of 0.9907858.

Example 3

To demonstrate the technique of GRG, the five-stage reliability problem is solved. The problem is

Maximize

$$R_s = \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=1}^{5} p_j(x_j)^2 \leq P$$

$$g_2 = \sum_{j=1}^{5} c_j(x_j + \exp(x_j/4)) \leq C$$

$$g_3 = \sum_{j=1}^{5} w_j x_j \exp(x_j/4) \leq W$$

where $x_j \geq 1$, $j=1,2,\ldots,5$, are integers.

The constants associated with the five stage problem are

j	R_j	p_j	P	c_j	C	w_j	W
1	0.80	1		7		7	
2	0.85	2		7		8	
3	0.90	3	110	5	175	8	200
4	0.65	4		9		6	
5	0.75	2		4		9	

It is noted that, in optimizing the system reliability, the decision variables, namely, the number of components used at each stage, are considered as continuous variables. The nearest integer numbers are assigned to them eventually.

We now apply the GREG computer code to solve this problem. The example will be reformulated as

Maximize

$$f_0(\bar{x}) = \sum_{j=1}^{5} \ell n \; [1 - (1 - R_j)^{x_j}]$$

subject to

$$f_1(\bar{x}) = \sum_{j=1}^{5} p_j \cdot x_j^{2} - 110 \leq 0$$

$$f_2(\bar{x}) = \sum_{j=1}^{5} c_j(x_j + \exp\,(x_j/4)) - 175 \leq 0$$

$$f_3(\bar{x}) = \sum_{j=1}^{5} w_j \cdot x_j \exp\,(x_j/4) - 200 \leq 0 \quad .$$

Then, four external, user-supplied subroutines will be used in which PHIX defines the objective function, CPHI defines the constraint functions, JACOB defines the gradient of the constraint functions, and GRADFI defines the gradient of the objective function.

The five-stage reliability problem solved by GREG program gives the optimal system configuration, $(x_1, x_2, x_3, x_4, x_5) = (2.678, 2.353, 2.070, 3.531, 2.792)$. The system reliability with this configuration is 0.9235. Since all x_j, $j=1,2,\ldots,5$, should be positive integers, the above results should be rounded off to the nearest integer as

(A). $(x_1, x_2, x_3, x_4, x_5) = (3, 2, 2, 4, 3)$, or

(B). $(x_1, x_2, x_3, x_4, x_5) = (3, 2, 2, 3, 3)$

Configuration (A), although will result in a higher system reliability, consumes 111 of g_1 which is greater than the available resource, 110. Configuration (B) gives system reliability, R_s, 0.9045 and consumes $g_1 = 83$, $g_2 = 146.1$, and $g_3 = 194.5$. Therefore, configuration (B) shows the optimal results.

REFERENCES

1. Abadie, J. and J. Guigou, Numerical experiments with the GRG method, in Integer and Nonlinear Programming, e.d. J. Abadie, Amsterdam: North Holland Publishing Co. (1970).

2. Abadie, J., and J. Carpenter, Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints, in Optimization, ed. R. Fletcher, N.Y.: Academic Press (1969).

3. Abadie, J., Solution des questions de dégénérescence dans la methode GRG, EDF* note HI 143/100 due 25 (September 1969).

4. Abadie, J. and J. Guigou, Gradient réduit généralisé, EDF* note HI 069/2 du Avril (1969).

5. Guigou, J., Presentation et utilisation du code GRG, EDF* note HI 102/02 du 9 Juin (1969).

6. Guigou, J., Presentation et utilisation du code GREG, EDF* note HI 582/2 due 25 Mai (1971).

7. Hwang, C. L., J. L. Williams, and L. T. Fan, Introduction to the Generalized Reduc Gradient Method, Institute for Systems Design and Optimization, Report No. 39, K.S.U., Manhattan, Kansas 66502.

8. Wolfe, P., Methods of nonlinear programming, in: Recent Advances in Mathematical Programming, eds. R. Graves and P. Wolfe, McGraw-Hill, New York, 1963.

9. Wolfe, P., Methods for linear constraints, in: Nonlinear Programming, ed. J. Abad North Holland Publishing Co., Amsterdam, 1967.

10. Aggarwal, K. K., and J. S. Gupta, "On minimizing the cost of reliable systems", IEEE Transactions on Reliability, Vol. R-24, No. 3, pp. 205 (August 1975).

3.6 METHOD OF LAGRANGE MULTIPLIERS AND THE KUHN-TUCKER CONDITIONS IN OPTIMAL SYSTEM RELIABILITY

1. Introduction

The general nonlinear programming problem can be solved by the method of Lagrange multipliers when the problem has characteristics that (1) no inequalities appear in the constraints, (2) no non-negativity or discreteness restrictions are imposed on the variables, (3) the number of equality constraints is less than the number of variables, and (4) the objective and constraint functions are continuous and posses partial derivatives at least through second order. The necessary and sufficient conditions are developed from Taylor's series expansion.

The method of Lagrange multipliers can be generalized to handle problems involving inequality constraints and non-negative variables. The necessary conditions for optimizing the problems are the so called Kuhn-Tucker conditions. These necessary conditions are also sufficient for a global minimum, if the objective function is a convex function and the constraints form a convex set of a feasible region, and for a global maximum if the objective function is a concave function and the constraints form a convex set of a feasible region.

Wolfe (1959) introduced, based on the Kuhn-Tucker conditions, the modified simplex method for quadratic programming problems which is widely used and is simple to apply. Many authors have derived the necessary and sufficient conditions for different cases of nonlinear programming problem form the Kuhn-Tucker conditions. For details, see [6].

Several papers have presented the application of method of Lagrange multipliers and the Kuhn-Tucker conditions to the following system reliability optimization problems:

Example 1   Single Constraint Problem

Given a system reliability requirement $R_{s,min}$, the problem is to determine a least-cost allocation of an N-stage series system that yields $R_s \geq R_{s,min}$. The example is from Kettelle [1962]. As an example, consider the following four-stage system with a system reliability requirement of $R_{s,min} = 0.99$ and total cost less than $b_1 = 61$:

Stage, j	4	3	2	1
C_j	1.2	2.3	3.4	4.5
R_j	0.8	0.7	0.75	0.85

The problem is

Maximize

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

subject to

$$g_1 = \sum_{j=i}^{N} c_j x_j \leq b_1$$

and

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

Example 2   Two Linear Constraints Problem

Consider an example of a series system of four stages. The component reliability, cost, and weight data are:

Stage, j	1	2	3	4
Component reliability, R_j	0.80	0.70	0.75	0.85
Cost, c_j	1.2	2.3	3.4	4.5
Weight, w_j	5	4	8	7

The system cost and weight are 56 and 120, respectively.

The problem is

Maximize

$$R_s = \prod_{j=i}^{4} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=i}^{4} c_j x_j \leq 56$$

$$g_2 = \sum_{j=i}^{4} w_j x_j \leq 120$$

where $x_j \geq 1$, j = 1, 2, 3, 4, are integers.

A simple Lagrange multiplier method may be used to solve a single constraint problem, e.g., Example 1. In this approach, the attempts employ a trial and error approach until all resouces are consumed, and assume that the degree of redundancy is continuous even though it must be discrete. However, it is very difficult to use Lagrange multipliers with multiple constraints, e.g., Example 2. To solve Example 2, the Kuhn-Tucker condition will be used to generate a set of simultaneous equations which can be solved by Newton's method. The solution obtains unique value of the Lagrange multipliers. Theoretically, a nonlinear constraint problem can also be solved by the Lagrange multiplier method and the Kuhn-Tucker conditions.

2. Lagrange Multiplier Method for Single Constraint Problem

Example 1

Example 1 in the Introduction section is considered. For this single constraint problem, one Lagrange multiplier, λ, should be introduced to form an unconstrained maximum of the new function

$$L(\bar{x}) = R_s - \lambda \sum_{j=i}^{N} c_j x_j \tag{1}$$

This solution is a solution to that constrained maximization problem where constraints are, in fact, the amount of the resource expended in achieving the unconstrained solution. In general, different choices of the λ's lead to different resource levels, and it may be necessary to adjust them by trial and error to achieve the maximum allowable resource, b_1. Therefore, the adjustment of the λ's is required [4].

Since that maximization of the logarithm of the system reliability maximizes the objective function, we take our payoff to be the log of the reliability

$$H = \ln R_s$$
$$= \sum_{j=i}^{N} \ln\left[1 - (1 - R_j)^{x_j}\right] \tag{2}$$

For a given λ, the Lagrange multiplier function will be formed as

$$L(\bar{x}) = \sum_{j=i}^{N} \ln\left[1 - (1 - R_j)^{x_j}\right] - \lambda \sum_{j=i}^{N} c_j x_j \tag{3}$$

over the integers $x_j \geq 1$. $j = 1,2,\ldots,N$.

Eq. (3) can be maximized by differentiation with respect to x_j and equating to zero to obtain the optimal x_j, then rounding off the values to the nearest integers. Namely,

$$\frac{dL(\bar{x})}{dx_j} = 0 \tag{4}$$

or

$$\frac{-(1 - R_j)^{x_j} \ln(1 - R_j)}{\left[1 - (1 - R_j)^{x_j}\right]} - \lambda c_j = 0 \tag{5}$$

leads to the solution (real) for x_j:

$$x_j = \frac{\ln\{1/[1 - \ln(1 - R_j)/\lambda c_j]\}}{\ln(1 - R_j)} \tag{6}$$

which is applied to each stage. The rounding off procedures to x_j, $j = 1, 2, 3, 4$, to upper and lower nearest integers are tested to determine which maximizes the R_s, and the payoffs and costs summed to produce an optimum solution.

Referring to Example 1, application of the Lagrange multiplier method as previously developed for a series of values of λ produces the solutions shown in Table 1. Inspection of the results shows that in all but one case the changes in allocation from one solution to the next consists of at most one additional component in at most one stage. Therefore the reliability and cost are monotonic increasing with the number of components used and there is no λ which could produce new solutions between these solutions. However, the transition from $\lambda = 0.0003$ to 0.0002 produced a change in three stages, and we can expect further solutions in this interval for intermediate λ values. Additional exploration of this region yields two more solutions, as given in Table 2.

Since there are no longer any changes by more than one component between successive solutions, the optimal allocation is $[x_1, x_2, x_3, x_4,]^* = [5,7,6,4]$ with system reliability, $R_s = 1 - 0.001288 = 0.998712$, and cost $\sum_{j=i}^{4} c_j x_j = 60.5$, which is the same result obtained by the dynamic programming approach.

Table 1

λ	Cost	System unreliability	Allocation			
			Stage 1	Stage 2	Stage 3	Stage 4
0.0009	44.6	0.009997	5	5	4	3
0.0008	48.0	0.007086	5	5	5	3
0.0007	50.3	0.005392	5	6	5	3
0.0006	54.8	0.002530	5	6	5	4
0.0005	54.8	0.002530	5	6	5	4
0.0004	54.8	0.002530	5	6	5	4
0.0003	54.8	0.002530	5	6	5	4
0.0002	61.7	0.001033	6	7	6	4

Table 2

λ	Cost	System unreliaiblity	Allocation			
			Stage 1	Stage 2	Stage 3	Stage 4
0.000225	54.8	0.002530	5	6	5	4
0.000220	57.1	0.002020	5	7	5	4
0.000215	60.5	0.001288	5	7	6	4
0.000210	61.7	0.001033	6	7	6	4

3. Kuhn-Tucker Conditions

The Kuhn-Tucker conditions can be stated as follows [6]:

A point $(x_1, x_2, \ldots, x_n)$ which optimizes a function

$$S = f(x_1, x_2, \ldots, x_n) \tag{7}$$

subject to the inequality constraints

$$g_j\ (x_1, x_2, \ldots, x_n) \leq 0, \qquad j = 1,2,\ldots,r \tag{8}$$

exists if there is a set of $\lambda_1, \lambda_2, \ldots, \lambda_m$, that satisfies the following set of conditions.

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=i}^{r} \lambda_j \frac{\partial g_j}{\partial x_i} = 0, \qquad i = 1,2, \ldots, n \tag{9}$$

$$\lambda j g_j = 0, \qquad j = 1,2, \ldots, r \tag{10}$$

$$g_j \leq 0, \qquad j = 1,2, \ldots, r \tag{11}$$

$$\lambda_j \geq 0, \qquad j = 1,2, \ldots, r \text{ (for maximization)} \tag{12a}$$

or

$$\lambda_j \leq 0, \qquad j = 1,2, \ldots, r \text{ (for maximization)} \tag{12b}$$

These conditions are also sufficient for a global minimum if f and g_j, $j = 1, 2, \ldots, r$, are all convex and differentiable functions and for a global maximum if f is concave and g_j, $j = 1, 2, \ldots, r$, are all convex and differentiable functions.

Similarly, the necessary conditions for optimization of the function equation (7), subject to the inequality constraints, equation (8) and the constraint of non-negative x are

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=i}^{r} \lambda_j \frac{\partial g_j}{\partial x_i} \leq 0, \qquad i = 1,2,\ldots, n \text{ (for maximization)} \quad (13a)$$

or

$$\frac{\partial L}{\partial x_i} = \frac{\partial f}{\partial x_i} - \sum_{j=i}^{r} \lambda_j \frac{\partial g_i}{\partial x_i} \geq 0, \qquad i = 1,2,\ldots, n \text{ (for minimization)} \quad (13b)$$

$$x_i \frac{\partial L}{\partial x_i} = 0, \qquad i = 1,2,\ldots, n \quad (14)$$

$$\lambda_j g_j = 0, \qquad j = 1,2,\ldots, r \quad (15)$$

$$g_j \leq 0, \qquad j = 1,2,\ldots, r \quad (16)$$

$$x_i \geq 0, \qquad i = 1,2,\ldots, n \quad (17)$$

$$\lambda_j \geq 0, \qquad j = 1,2,\ldots, r \text{ (for maximization)} \quad (18a)$$

or

$$\lambda_j \leq 0, \qquad j = 1,2, \ldots, r \text{ (for minimization)} \quad (18b)$$

Equations (12a), (12b), (18a) and (18b) are based on the fact that if $\lambda > 0$, the stationary point cannot be a minimum, and if $\lambda < 0$, it can not be a maximum [kuhn-Tucker (1951)]. Note that the sign of λ will be affected by factors such as the nature of the optimization problem (whether maximization or minimization), the type of inequality constraints [whether $g_j(x) \leq 0$ or $g_j(x) \geq 0$], and the form of the Lagranian function [whether $L(x,\lambda) = f(x) - \sum_j \lambda_j g_j(x)$ or $L(x,\lambda) = f(x) + \sum_j \lambda_j g_j(x)$]. Recall that equations (12a), (12b), (18a) and (18b) are based on the inequality constraints given by equation (8) [$g_j(x) \leq 0$], and the Lagrangian function of the form, $L(x,\lambda) = f(x) - \sum_j \lambda_j g_j(x)$.

4. Method of Lagrange multipliers and the Kuhn-Tucker Conditions for Two Linear Constraints Problem

Example 2

When more than one constraint is imposed to the problem, the trial and error procedure for searching λ's associated with each constraint is not practical. In this example, the Kuhn-Tucker conditions are applied to simplify the problem [9,10].

For an N-stage series system, the problem can be restated as

Maximize

$$R_s = \prod_{j=i}^{N} [1 - (1 - R_j)^{x_j}] \qquad (19)$$

subject to

$$\sum_{j=i}^{N} a_{ij} x_j \leq b_i \qquad i = i,2,\ldots, r \qquad (20)$$

If we denote $(1 - R_j)$ by Q_j, $(1 - r_j)^{x_j}$ by Q'_j, then eq. (19) becomes

$$R_s = \prod_{j=i}^{N} (1 - Q'_j)$$

Since maximization of the logarithm of the system reliability maximizes the objective function [1], we can denote the objective function by

$$\ell n R_s = \sum_{j=i}^{N} \ell n(1 - Q'_j) \qquad (21)$$

Also, by

$$Q_j^{x_j} = Q'_j,$$

We obtain

$$x_j = \frac{\ell n Q'_j}{\ell n Q_j} \qquad (22)$$

Substituting x_j into eq. (20),

$$\sum_{j=i}^{N} \alpha_{ij} \frac{\ell n Q_j'}{\ell n Q_j} \leq b_i, \qquad i = 1,2,\ldots, r$$

or

$$\sum_{j=i}^{N} \alpha_{ij} \ \ell n Q_j' \leq b_i, \qquad i = 1,2,\ldots, r \tag{23}$$

where

$$\alpha ij = \frac{a_{ij}}{\ell n Q_j} \tag{24}$$

Since the objective (eq. (21)) and constraints (eq. (23)) functions are all separable concave and convex of Q_j' respectively, this guarantees the global maximum [5].

The Lagrange function, whose stationary point is to be found, is

$$L(R', \lambda) \equiv \sum_{j=i}^{N} \ell n R_j - \sum_{i=1}^{r} \lambda_i \left[\sum_{j=i}^{N} (\alpha_{ij} \ \ell n(1 - R_j')) - b_i \right] \tag{25}$$

The Kuhn-Tucker condition can be written as

$$\frac{\partial L}{\partial R_j} = \frac{1}{R_j} + \sum_{i=1}^{r} \lambda_i \alpha_{ij} / (1 - R_j') = 0, \quad i = 1,2,\ldots,N \tag{26}$$

$$\lambda_i \left[\sum_{j=i}^{N} (\alpha_{ij} \ \ell n(1 - R_j')) - b_i \right] = 0 \tag{27}$$

$$\sum_{j=i}^{N} \alpha_{ij} \ \ell n(1 - R_j') - b_i \leq 0 \tag{28}$$

$$\lambda_i \geq 0, \qquad i = 1,2,\ldots,r \tag{29}$$

Eqs. (26) - (29) form the basis of a solution for optimization. A set of N + r equations represented by eq. (26) and eq. (27) can be solved by Newton's method. Actually, eq. (29) will form the stopping criterion for the iteration process.

After the solution is obtained from the simultaneous equations formed from the Kuhn-Tucker conditions, we combine the solution with eqs. (21) and (22) to get the optimal system reliability and optimal redundant numbers for each stage in which the rounding off procedure to the nearest integer number is required.

Referring to Example 2, the problem can be stated as

Maximize

$$\ell n R_s = \sum_{j=1}^{4} \ell n R_j'$$

subject to

$$\sum_{j=1}^{4} \frac{c_j}{\ell n(1-R_j)} \ell n(1 - R_j') - C \leq 0$$

$$\sum_{j=1}^{4} \frac{w_j}{\ell n(1-R_j)} \ell n(1 - R_j') - W \leq 0$$

The Lagrange function is

$$L(R',\lambda) = \sum_{j=1}^{4} \ell n\ R_j' - \lambda[\sum_{j=1}^{4} \frac{c_j}{\ell n(1-R_j)} \ell n(1 - R_j') - C]$$
$$- \lambda_2[\sum_{j=1}^{4} \frac{w_j}{\ell n(1-R_j)} \ell n(1 - R_j') - W]$$

The Kuhn-Tucker conditions are

$$\frac{\partial L}{\partial R_j'} = \frac{1}{R_j'} + \frac{\lambda_1 c_j + \lambda_2 w_j}{(1 - R_j')\ \ell n(1 - R_j)} = 0, \qquad i = 1,2,3,4 \tag{30}$$

$$\lambda_1[\sum_{j=1}^{4} \frac{c_j}{\ell n(1-R_j)} \ell n(1 - R_j') - C] = 0 \tag{31}$$

$$\lambda_2[\sum_{j=1}^{4} \frac{w_j}{\ell n(1-R_j)} \ell n(1 - R_j') - W] = 0 \tag{32}$$

$$\sum_{j=1}^{4} \frac{c_j}{\ell n(1-R_j)} \ell n(1 - R_j') - C \leq 0 \tag{33}$$

$$\sum_{j=1}^{4} \frac{w_j}{\ln(1-R_j)} \ln(1 - R_j') - W \leq 0 \tag{34}$$

$$\lambda_1, \lambda_2 \leq 0 \tag{35}$$

This problem solved by the Lagrange multipliers method and the Kuhn-Tucker conditions following eqs. (30) - (35) gives the results, $[R_1, R_2, R_3, R_4]^* = [0.999735, 0.999494, 0.999294, 0.999388]$, $[\lambda_1, \lambda_2]^* = [0.00019994, -0.00003730]$, and the system reliability R_s, 0.997914. If use eq (22), we find the optimal allocation $[x_1, x_2, x_3, x_4]^* = [5.11, 6.30, 5.23, 3.90]$ which have to be rounded off to the nearest integers as [5, 6, 5, 4]. The system reliability under the allocation of [5, 6, 5, 4] is 0.997471, the consumed g_1 is 54.8 and g_2 is 117.

5. Conclusion

The Kuhn-Tucker conditions do provide valuable clues about the characteristics of the optimal solution, and they also permit the determination of the optimal solution. However, it is usually difficult, if not impossible to derive the optimal solution for a large scale nonlinear programming problem directly from the conditions. Also, it is not necessarily true that every point which is a solution to the Kuhn-Tucker conditions will be a point at which the objective function takes on a relative maximum or minimum for all $\bar{x}$ which satisfy the constraints. But every point at which the objective function assumes a relative maximum or minimum for $\bar{x}$ satisfying the constraints must be a solution to the Kuhn-Tucker conditions. There are many valuable indirect applications of the Kuhn-Tucker conditions. An example is quadratic programming.

REFERENCES

1. R. E. Barlow, F. Proschan, and L. C. Hunter, Mathematical Theory of Reliability, New York: Wiley, 1965.

2. Black, G., and F. Proschan, "On Optimal redundancy", Operations Research, Vol. 7, pp. 581-588 (1959).

3. Bodin, L. D., "Optimization procedure for the analysis of coherent structures", IEEE Transactions on Reliability, Vol. R-18, No. 3, pp. 118-126 (August 1969).

4. Everett, H. III, "Generalized Lagrange Multiplier Method for solving problems of optimum allocation of resources", Operations Research, Vol. 11, No. 3, pp. 399-417 (May-June 1963).

5. Hadley, G., Nonlinear and Dynamic Programming, Reading, Mass.: Addison-Wesley, 1964.

6. Hwang, C. L., P. K. Gupta, and L. T. Fan, Method of Lagrange Multipliers and the Kuhn-Tucker-Conditions, Institute for Systems Design and Optimization, Report No. 60, Kansas State University (1974).

7. Kuhn, N. W. and Tucker, A. W., Non-Linear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statics and Probability, J. Neyman, Ed., University of California Press, Berkeley, California, pp. 481-492 (1951).

8. Messinger, M., and Shooman, H., "Technique for optimum spares allocation: a tutorial review", IEEE Transactions on Reliability, Vol. R-19, pp. 156-166 (November 1970).

9. Misra, K. B., and M. D. Ljubojevic, "Optimal Reliability design of a system: a new look", IEEE Transactions on Reliability, Vol. R-22, pp. 255-258 (December 1973).

10. Misra, K. B., "Reliability optimization of a series-parallel system", IEEE Transactions on Reliability, Vol. R-21, No. 4, pp. 230-238 (November 1972).

11. Shershin, A. C., "Mathematical optimization techniques for the simultaneous apportionments of reliability and maintainability", Operations Research, Vol. 18, No. 1, pp. 95-106 (January-February 1970).

12. Tucker, A. W., Linear and Nonlinear Programming, Operations Research, Vol. 5, pp. 244-257 (1957).

3.7 THE GENERALIZED LAGRANGIAN FUNCTION METHOD APPLIED TO OPTIMAL SYSTEMS RELIABILITY

1. Introduction

A general mathematical programming problem can be stated as

$$\text{Problem (A): minimize } f(\bar{x}) \tag{1}$$

subject to

$$g_i(\bar{x}) \geq 0, \qquad i = 1, \ldots, m, \tag{2}$$

$$x \varepsilon \Omega \tag{3}$$

where $x \varepsilon E^n$, and Ω is a subset of n-Euclidean space E^n. It is assumed that $f(\bar{x})$, and $g_1(\bar{x})$, $g_2(\bar{x})$, $\ldots, g_m(\bar{x})$ are real valued functions on Ω and twice continuously differentiable.

Problem (A) can be solved by methods which are based on transformation of a given constrained problem into a sequence of unconstrained problems. There are two classes of such methods, namely, the penalty and Lagrangian methods. The penalty methods (e.g., sequential unconstrained minimization technique) have been studied extensively and applied to many practical problems [3],[5]. However, they suffer from numerical instabilities. The Lagrange multipliers method has been used mostly for the analysis of economic systems [2]. Recently, augmented Lagrangian functions have been proposed to solve the problems with equality [6,7,10] and inequality constraints [1,11,12,13].

In this section a new type of the generalized or augmented Lagrangian function proposed by Sayama et al. [12,13] for finding the solution of a non-linear programming problem with inequality constraints is applied to optimal systems reliability problems. The function is twice continuously differentiable and closely related to the generalized penalty function which includes the interior and exterior penalty functions as special cases.

The theoretical properties of the function and the computational algorithm are presented in [13]. The method has been proved to be locally convergent to the saddle points of the generalized Lagrangian. By using the method, we can find the Lagrange multipliers associated with the solution of problem (A), which play an important part for design and synthesis in the fields of engineering and economics.

2. The generalized Lagrangian function and the computational procedures

The classical Lagrangian function associated with Problem (A) is defined as

$$L(\bar{x},\bar{\lambda}) = f(\bar{x}) - \sum_{i=1}^{m} \lambda_i g_i(\bar{x}) \tag{4}$$

where λ_i, $i = 1,2, \ldots, m$ are the Lagrange multipliers. The literature on the penalty method and the method of Lagrange multipliers is well reviewed in Fiacco and McCormick [5], Lootsma [8], and Rockafellar [11].

Although several examples have been suggested to satisfy the properties of the generalized Lagrangian, a proper choice of the function is of utmost importance in obtaining efficient methods of solution. A class of the generalized or augmented Lagrangian proposed by Sayama et al. [12,13] is

$$L(\bar{x},\bar{\lambda};t) = f(\bar{x}) - \sum_{i=1}^{m} \begin{cases} \lambda_i g_i(\bar{x}) - t g_i^2(\bar{x}), & g_i(\bar{x}) \leq 0 \\ \dfrac{(\lambda_i)^2 g_i(\bar{x})}{\lambda_i + t g_i(\bar{x})}, & g_i(\bar{x}) \geq 0 \end{cases} \tag{5}$$

or in a similar form to the classical Lagrangian

$$L(\bar{x},\bar{\lambda};t) = f(\bar{x}) - \sum_{i=1}^{m} \lambda_i g_i(\bar{x}) + \sum_{i=1}^{m} \begin{cases} t g_i^2(\bar{x}), & g_i(\bar{x}) \leq 0 \\ \dfrac{\lambda_i t g_i^2(\bar{x})}{\lambda_i + t g_i(\bar{x})}, & g_i(\bar{x}) \geq 0 \end{cases} \tag{6}$$

where λ_i, i = 1, 2, ..., m are multipliers and t>0, a penalty parameter. $L(\bar{x},\bar{\lambda};t)$ is termed the multiplier function, and the computational algorithm using the function is called the multiplier method. $L(\bar{x},\lambda;t)$ is constructed in a such a way that it is twice continuously differentiable if $f(\bar{x})$, and $g_i(\bar{x})$, i = 1,2, ... m are twice continuously differentiable. This property is very important to the computational procedure for finding the unconstrained minimum of the generalized Lagrangian.

It is worth noting that by letting t = 0 in $L(\bar{x},\bar{\lambda};t)$, equation (5) is reduced to the classical Lagrangian equation (4). The multiplier function can also be interpreted as an exterior penalty function if $\lambda_i = 0$, i = 1,2, ..., m in $L(\bar{x},\bar{\lambda};t)$.

A computational algorithm which makes use of the multiplier function associated with Problem (A) is considered. The penalty parameter, t, if choosen sufficiently large (say 10^5), is kept constant. Let $\bar{\lambda}1$ be an initial estimation of $\bar{\lambda}$, and let $\bar{x}^k$ denotes a point minimizing $L(\bar{x},\bar{\lambda}^k;t)$; i.e.,

$$L_x(\bar{x}^k,\bar{\lambda}^k;t) = \nabla f(\bar{x}^k) - \sum_{i=1}^{m} \left\{ \begin{array}{l} [\lambda_i^k - 2tg_i(\bar{x}^k)]\nabla g_i(\bar{x}^k) \\ \\ \dfrac{(\lambda_i^k)^3}{[\lambda_i^k + tg_i(\bar{x}^k)]^2}\nabla g_i(\bar{x}^k) \end{array} \right\} = 0 \qquad (7)$$

This suggests that we take

$$\lambda_i^{k+1} = \begin{cases} \lambda_i^k - 2tg_i(\bar{x}^k) & , \quad g_i(\bar{x}^k) \leq 0, \\ \\ \dfrac{(\lambda_i^k)^3}{[\lambda_i^k + tg_i(\bar{x}^k)]^2} & , \quad g_i(\bar{x}^k) \geq 0, \end{cases} \qquad i = 1, \ldots, m, \qquad (8)$$

so that $(\bar{x}^k,\bar{\lambda}^{k+1})$ satisfies the following equation

$$L_x(\bar{x}^k,\bar{\lambda}^{k+1}) = \nabla f(\bar{x}^k) - \sum_{i=1}^{m} \lambda_i^{k+1} \nabla g_i(\bar{x}^k) = 0. \qquad (9)$$

If $\bar{\lambda}^{1} \geq 0$, then $\bar{\lambda}^{k}$ is kept non-negative according to the correction of (8). Eq. (8) may be represented as follows:

$$\lambda_i^{k+1} = \begin{cases} \lambda_i^k - 2\xi_i t g_i(\bar{x}^k) & , \quad g_i(\bar{x}^k) \leq 0 \\ \lambda_i^k - \xi_i \dfrac{t\lambda_i^k g_i(\bar{x}^k)\ [2\lambda_i^k + tg_i(\bar{x}^k)]}{[\lambda_i^k + tg_i\ (\bar{x}^k)]^2} & , \quad g_i(\bar{x}^k) \geq 0, \end{cases}$$

$$i = 1,\ldots,m, \tag{10}$$

where $1 \geq \xi_i > 0$. If $\xi_i = 1$, (10) is equivalent to (8). By using the multiplier function (10) can be written as follows:

$$\lambda_i^{k+1} = \lambda_i^k + cL_{\lambda_i}\ (\bar{x}^k, \bar{\lambda}^k; t), \qquad i = 1,\ldots,m, \tag{11}$$

where c is a scalar,

$$C = \begin{cases} 2\xi_i t \\ 2\xi_i t\ \dfrac{\lambda_i^k + 0.5tg_i(\bar{x}^k)}{\lambda_i^k + 2tg_i(\bar{x}^k)} \end{cases}$$

The computational procedure by the multiplier method may be summarized as follows:

1) Choose a penalty parameter $t > 0$ and initial values of multiplier $\bar{\lambda}1 \geq 0$.

2) Find x^k that minimizes $L(\bar{x}, \lambda^k; t)$. Any multidimensional search technique, e.g., the sequential simplex pattern search may be used.

3) Stop the iterations when one of the following criteria is satisfied.

$$|\lambda_i^k g_i(\bar{x}^k)| \le \varepsilon, \quad i = 1, \ldots, m$$

or

$$|f(\bar{x}^k) - L(\bar{x}^k, \bar{\lambda}^k; t)| \le \varepsilon$$

where ε is a sufficiently small positive number.

4) Select $\bar{\lambda}^{k+1}$ by (8) or (10), and return to step 2).

3. Numerical Examples

Example 1

To demonstrate the generalized Lagrangian function method, the five-stage reliability problem is solved. The problem is

Maximize

$$R_s = \prod_{j=1}^{5} [1 - (1-R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=1}^{5} p_j (x_j)^2 \le P$$

$$g_2 = \sum_{j=1}^{5} c_j (x_j + \exp(x_j/4)) \le C$$

$$g_3 = \sum_{j=1}^{5} w_j x_j \exp(x_j/4) \le W$$

where $x_j \ge 1$, $j = 1, 2, \ldots, 5$, are integers.

The constants associated with the five-stage problem are

j	R_j	p_j	P	c_j	C	w_j	W
1	0.80	1		7		7	
2	0.85	2		7		8	
3	0.90	3	110	5	175	8	200
4	0.65	4		9		6	
5	0.75	2		4		9	

It is noted that in optimizing the system reliability, the decision variables, namely, the number of components used at each stage, are considered as continuous variables. The nearest integer numbers are assigned to them eventually.

To solve this problem, we first of all have to reformulate it as

Minimize

$$f_o(\bar{x}) = -R_s = -\sum_{j=1}^{5} [1 - (1-R_j)^{x_j}]$$

subject to

$$g_1(\bar{x}) = P - \sum_{j=1}^{5} p_j (x_j)^2 \geq 0$$

$$g_2(\bar{x}) = C - \sum_{j=1}^{5} c_j (x_j + \exp(x_j/4)) \geq 0$$

$$g_3(\bar{x}) = W - \sum_{j=1}^{5} w_j x_j \exp(x_j/4) \geq 0$$

$$g_{3+j}(x_j) = x_j - 1.0 \geq 0 \qquad , \quad j = 1, 2, \ldots, 5$$

$$g_9(\bar{x}) = 1.0 + f_o(\bar{x}) \geq 0$$

The constant penalty parameter, $t = 1.0 \times 10^5$, and initial estimate of multiplier $\bar{\lambda}^1 = (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0)^T$ are chosen [Step 1]. The generalized Lagrangian function is given by

$$L(\bar{x},\bar{\lambda};t) = f_o(\bar{x}) - \sum_{i=1}^{9} \begin{cases} 1 \cdot g_i(\bar{x}) - 1.0 \times 10^5 \cdot g_i^{\,2}(\bar{x}), & g_i(\bar{x}) \leq 0 \\ \dfrac{(1)^2 \cdot g_i(\bar{x})}{1 + 1.0 \times 10^5 \cdot g_i(\bar{x})}, & g_i(\bar{x}) > 0 \end{cases} \quad (12)$$

which is a function of $\bar{x} = (x_1,x_2,x_3,x_4,x_5)^T$ only. The sequential simplex pattern search method starting from $\bar{x}^o = (1.0, 1.0, 1.0, 1.0, 1.0)^T$ is applied to find the minimum $L(\bar{x},\bar{\lambda};t)$ of eq. (7) [Step 2] $\cdot$ $\bar{x}^1 = (x_1^1,x_2^1,x_3^1,x_4^1,x_5^1)^T$ is found to be $(2.0594, 2.5178, 2.7202, 3{,}4299, 2.6118)^T$ which gives

$L'(\bar{x}^1,\bar{\lambda};t) = -0.9014323$

$f_o(\bar{x}^1) = -0.9013423$

$g_1(\bar{x}^1) = 13.39341$

$g_2(\bar{x}^1) = 26.07391$

$g_3(\bar{x}^1) = 15.60743$

$g_4(\bar{x}^1) = 1.0594$

$g_5(\bar{x}^1) = 1.5178$

$g_6(\bar{x}^1) = 1.7202$

$g_7(\bar{x}^1) = 1.4299$

$g_8(\bar{x}^1) = 1.6118$

$g_9(\bar{x}^1) = 0.0986577$

The stopping criteria are $\varepsilon_1 = 1.0 \times 10^{-3}$ and $\varepsilon_2 = 1.0 \times 10^{-4}$. Since

$$|\lambda_i' \, g_i(\bar{x}')| = |1 \cdot g_i(\bar{x}')| \not\leq 1.0 \times 10^{-3}, \quad \text{for} \quad i = 1, 2, \ldots, 9,$$

and

$$\left| \frac{L(\bar{x}', \bar{\lambda}^1; t) - L(\bar{x}^o, \bar{\lambda}^o; t)}{L(\bar{x}^1, \bar{\lambda}^1; t)} \right| = \left| \frac{0.9013426 - 0.9}{0.9013426} \right|$$

$$= 0.0015 \not\leq 1 \times 10^{-4} \qquad \text{[Step 3]} .$$

We should go to Step 4 by choosing new $\bar{\lambda}^2$'s using eq. (8). Then go to Step 2.

The iterative procedures are carried out until the stopping criteria are satisfied [Step 3]. The results are presented in Table 1.

The optimal results are $\bar{X} = (x_1, x_2, x_3, x_4, x_5)^T = (2.408, 2.376, 2.019, 3.632, 2.898)$, $f_0(\bar{x}) = -0.9193794$, $g_1 = 12.85325$, $g_2 = 23.77317$, $g_3 = 0.18543$.

Since the number of components used at each stage should be positive integers, the optimal results of $\bar{X}$ shall be rounded off to the nearest integers, and check the constraints. The possible system configurations are:

(A) $(x_1, x_2, x_3, x_4, x_5) = (2,2,2,4,3)$

(B) $(x_1, x_2, x_3, x_4, x_5) = (3,2,2,4,3)$

(C) $(x_1, x_2, x_3, x_4, x_5) = (3,2,2,3,3)$

Configuration (A) results in system reliability, $R_s = 0.9007759$, $g_1(\bar{x}) = 4.0$, $g_2(\bar{x}) = 24.74182$, $g_3(\bar{x}) = 1.7710$. Configuration (B) results in system reliability, $R_s = 0.9308028$, $g_1(\bar{x}) = -1.0$, $g_2(\bar{x}) = 14.467$, $g_3(\bar{x}) = -19.6139$ in which constraint 1, and 3 are violated. Configuration (C) results in system reliability, $R_s = 0.9044667$, $g_1(\bar{x}) = 27$, $g_2(\bar{x}) = 28.879$, $g_3(\bar{x}) = 7.319$. Configuration (B) consumes the costs exceeding the total available resources. Configuration (A) gives the system reliability less than that from configuration (C), therefore configuration (C) is the optimal solution.

Table 1. Computational results of the system reliability maximization problem (Example 1).

$\varepsilon_1 = 10^{-3}$, $\varepsilon_2 = 10^{-4}$, $t = 10^5$ $\lambda_i^1 = 1.0$, $x_i^o = 1.0$, $i = 1,2, \ldots, 5$

Iteration

k	x_1	x_2	x_3	x_4	x_5	$-L$	$R_s = -f_o$	g_1^k	g_2^k	g_3^k
0	1.0	1.0	1.0	1.0	1.0	0.9	0.29835	98.0	101.91	151.21
1	2.0594	2.5178	2.7207	3.4299	2.6118	0.9013426	0.9013423	13.39	26.07	1.32
2	2.2845	2.4699	2.2402	3.3416	3.0026	0.9166631	0.9166632	14.83	25.08	0.20
3	2.3672	2.4509	1.9734	3.7306	2.7458	0.9182468	0.9182469	9.95	22.27	2.04
4	2.3808	2.3486	2.0810	3.6134	2.8736	0.9182124	0.9182125	11.57	23.36	1.49
5	2.3928	2.3606	2.0930	3.5964	2.9090	0.9192970	0.9192970	11.33	23.06	0.001
6	2.4079	2.3757	2.0187	3.6316	2.8983	0.9193794	0.9193794	11.13	22.83	0.20

Computation Time = 5.36 sec.

Example 2

The objective of this example is to find the optimal R_i's which minimize

$$C = 2K_1R_1^{\alpha_1} + 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4}$$

subject to the constraints

$$R_{s,min} \leq 1 - R_3[(1-R_1)(1-R_4)]^2 - (1-R_3)\{1-R_2[1 - (1-R_1)(1-R_4)]\}^2$$

$$R_i \geq R_{i,min}$$

The numerical values of parameters are

$K_1 = 100,\quad K_2 = 100,\quad K_3 = 200,\quad K_4 = 150$

$\alpha_i = 0.6,\quad i = 1,2,3,4.$

$R_{s,min} = 0.9,\ R_{i,min} = 0.5,\ i = 1,2,3,4.$

The problem is formulated in the generalized Lagrangian function format as follows:

Minimize

$$f_o(\bar{R}) = C$$

$$= 2K_1R_1^{\alpha_1} + 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4}$$

subject to the constraints

$$g_1(\bar{R}) = 1 - R_3[(1-R_1)(1-R_4)]^2 - (1-R_3)\{1 - R_2[1 - (1-R_1)(1-R_4)]\}^2 - R_{s,min} \geq 0$$

$$g_{i+1}(\bar{R}) = R_i - R_{i,min} \geq 0, \qquad i = 1,2,3,4.$$

$$g_{i+5}(\bar{R}) = 1 - R_i \geq 0, \qquad i = 1,2,3,4$$

$$g_{10}(\bar{R}) = R_3\ [(1-R_1)(1-R_4)]^2 - (1-R_3)\{1-R_2[1-(1-R_1)(1-R_4)]\}^2 \geq 0$$

where $1-g_{10}(\bar{R})$ is the system reliability of the complex configuration shown in Fig. 1 , and $\bar{R} = [R_1,R_2,R_3,R_4]^T$. R_1,R_2,R_3, and R_4 are the reliabilities of blocks 1,2,3, and 4 respectively.

For this example, the SUMT-RAC program fails to satisfy the special requirement that the violable non-negativity constraints should never be violated during the search. The results by applying the formulation of generalized Lagrangian function method are presented in Table 2.

The optimal solutions obtained from the starting point of $(R_1,R_2,R_3,R_4)^o$ = (0.7,0.7,0.7,0.7) are $(R_1,R_2,R_3,R_4)^*$ = (0.50001, 0.84062, 0.5, 0.5), optimal minimum cost, C = 642.0446, and the system reliability, R_s = 0.9005. The penalty parameter, $t = 1.0 \times 10^5$, $\varepsilon_1 = 1.0 \times 10^{-3}$, and $\varepsilon_2 = 1.0 \times 10^{-4}$ are used.

It is noted that the penality parameter, say $t = 1.0 \times 10^5$, is large enough, we will finally reach the optimal solution in the feasible region. Comparing with the results obtained by SUMT-Lai, the cost is almost the same (C = 642,0446 by this method, and 642.428 by SUMT-Lai), but the system reliability, R_s, is slightly higher by this method (R_s = 0.9005) than that by SUMT-Lai (R_s = 0.900021). The multiplier method also exhibits much faster convergence (11.53 sec. for this problem) than SUMT-Lai (about 45 sec. for the same problem) using an IBM 370/158 computer.

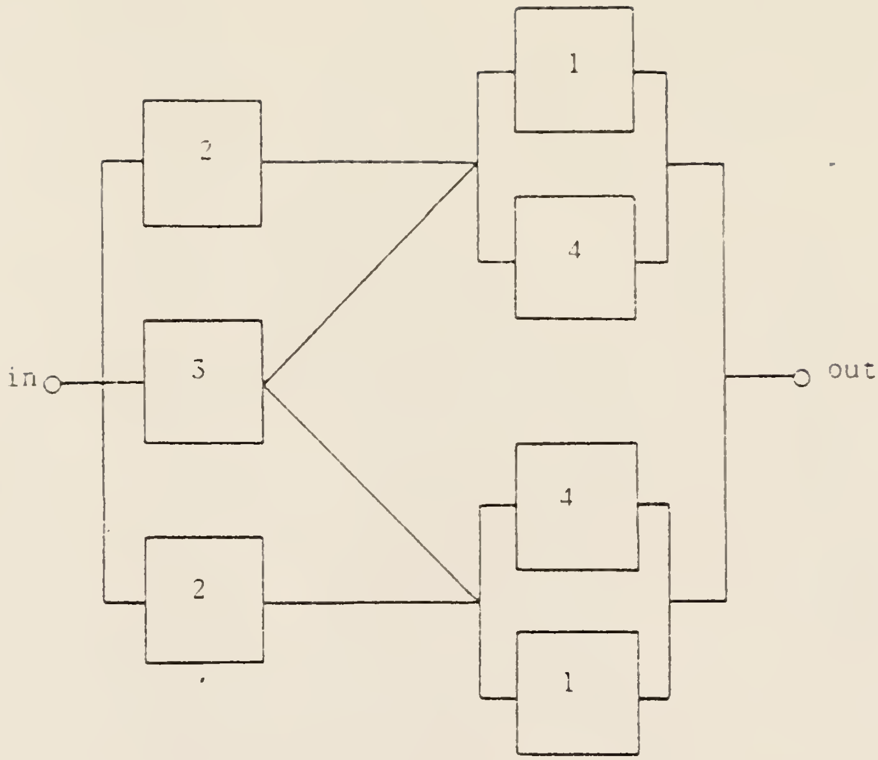


Fig. 1 A schamatic diagram of a complex system

Table 2 Computer results of the cost function minimization problem (Example 2)

$\varepsilon_1 = 10^{-3}$ $\varepsilon_2 = 10^{-4}$, $t = 10^5$,

$\lambda_i = 1.0$, $R_i = 0.7$ $i = 1,2,3,4$

Iteration k	R_1	R_2	R_3	R_4	R_5	L	f_o
0	0.7	0.7	0.7	0.7	0.8274	700.	726.61
1	0.5	0.83076	0.49992	0.50021	0.8979	641.32	640.8625
2	0.50004	0.83811	0.5	0.50001	0.8999	641.82	641.7678
3	0.50001	0.84062	0.5	0.5	0.9005	641.85	642.0446

Computation Time = 7.65 sec.

REFERENCES

1. K. J. Arrow, F. J. Gould and S. M. Howe, "A general saddle point result for constrained optimization," Institute of Statistics Mimeo Series No. 774, Univ. of N. Carolina (Chapel Hill), 1971.

2. K. J. Arrow, L. Hurwicz and H. Uzawa: Studies in Linear and Non-linear Programming, Stanford Univ. Press, Stanford, 1958.

3. J. Bracken and G. P. McCormick: Selected Applications of Nonlinear Programming, Wiley, 1968.

4. H. Everett, "Generalized Lagrange multiplier method for solving problems of optimal allocation of resources," Operat. Res., 11, 399-417, 1963.

5. A. V. Fiacco and G. P. McCormick: Nonlinear Programming, Sequential Unconstrained Minimization Techniques, Wiley, 1968.

6. P. C. Haarhoff and J. D. Buys, "A new method for the optimization of a nonlinear function subject to nonlinear constraints," Comput. J., 13, 178-184, 1970.

7. M. R. Hestenes, "Multiplier and gradient method," J. Optim. Theory and Applns., 4, 303-320, 1969.

8. F. A. Lootsma, "A survey of methods for solving constrained minimization problems via unconstrained minimizations," in Numerical Methods for Nonlinear Optimization, F. A. Lootsma (editor), Academic Press, 1973.

9. B. T. Polyak, "Iteration methods using Lagrange multipliers for solving extremal problems with constraints of the equation type," USSR Comput. Math. and Math. Phys., 10-5, 42-52 (1970).

10. M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in Optimization, R. Fletcher (editor), Academic Press, 1969.

11. R. T. Rockafellar, "Penalty method and augmented Lagrangians in nonlinear programming," Proceedings of 5th IFIP Conference on Optimization Techniques, Rome, May, 1973.

12. H. Sayama, Y. Kameyama, H. Nakayama, and Y. Sawaragi, "Iteration Process in Lagrange Multipliers for Multiplier Method", System and Control (Japan), Vol. 17, pp. 775-778 (1973).

13. H. Sayama, Y. Kameyama, H. Nakayama, and Y. Sawaragi, "The Generalized Lagrangian Functions for Mathematical Programming Problems", Report 55, Institute for Systems Design and Optimization, Kansas State University, 1974.

3.8 GEOMETRIC PROGRAMMING APPLIED TO OPTIMAL SYSTEM RELIABILITY

1. Introduction

By employing the well-known inequality which states that the arithmetic mean is at least as great as the geometric mean, dual problems for a variety of optimal design problems, i.e., primary problems, may be formulated. Geometric programming exploits this inequality and the relationships between the primal and dual problems to facilitate solution of optimization problems. The primal problems must be expressed in terms of a class of functions which are called positive polynomials or posynomials for short.

In a primal problem, posynomial S is minimized subject to constraints of the posynomial type. Because of the inequality relating the arithmetic and geometric means, there exists a related problem which requires maximization of the so-called dual function v subject to certain linear constraints [1,2,6].

Geometric programming differs from other optimization techniques in that it gives the minimum values $S(\bar{x})$ of posynomial S (primary function) without first locating the point $\bar{x}$ where S is minimum. It solves the dual problem first, then the optimal solution of the primal problem can be obtained by the corresponding relation (see the following sections).

2. Formulation of the Problem

A more general primal minimization problem involving posynomials subject to r inequality constraints and the corresponding dual problem of maximizing the dual function subject to its constraints can be stated as follows:

<u>Primal Problem</u>

Minimize

$$S = \sum_{j=1}^{n_0} u_j \tag{1}$$

subject to

$$g_i \leq 1 \qquad , \; i=1,2,\ldots,r \tag{2}$$

where

$$g_i = \sum_{j=m_i}^{n_i} u_j \qquad , \tag{3}$$

Here

$$m_i = n_{i-1} + 1 \qquad , \; i=1,2,\ldots,r \tag{4}$$

and the u_j are numbered consecutively from 1 to $n_r = n$. The u_j are defined:

$$u_j = c_j \prod_{i=1}^{m} x_i^{a_{ji}} \qquad , \qquad j=1,2,\ldots,n \tag{5}$$

where

$$x_1, x_2, \ldots, x_m > 0 \tag{6}$$

The components a_{jk} are arbitrary real numbers, but the coefficients c_j assumed to be positive.

The posynomial S which is to be minimized is a function of m independent variables $x_1, x_2, \ldots, x_m$. The inequality constraints, eq. (2), are called forced constraints, where the inequality constraints given in eq. (6) are considered to be natural constraints. The matrix (a_{jk}) is called the exponent matrix. It has n rows and m columns.

The dual problem that corresponds to the primal problem is as follows:

Dual Problem

Maximize

$$v = \left[\prod_{j=1}^{n} \left(\frac{c_j}{\delta_j}\right)^{\delta_j} \right] \prod_{i=1}^{r} \lambda_i^{\lambda_i} \quad . \tag{7}$$

where

$$\lambda_i = \sum_{j=m_i}^{n_i} \delta_j \qquad , \; i=1,2, \ldots, r \tag{8}$$

Here

$$m_1 = n_0 + 1, \qquad m_2 = n_1 + 1, \ldots, m_r = n_{r-1} + 1$$

The constants c_j are assumed to be positive and the weights $\delta_1, \delta_2, \ldots, \delta_n$ are subject to the linear constraints:

$$\delta_1 \geq 0, \quad \delta_2 \geq 0, \ldots, \delta_n \geq 0 \tag{9}$$

$$\sum_{j=1}^{n_0} \delta_j = 1 \tag{10}$$

$$\sum_{j=1}^{n} a_{jk} \delta_j = 0, \qquad k = 1,2, \ldots, m \tag{11}$$

where the coefficients, a_{jk}, are all real numbers.

The dual function, v, is a function of the variables, $\delta_1, \delta_2, \ldots, \delta_n$, and the linear constraints of positivity condition (eq. (9)), the normality condition (eq. (10)), and the orthogonality condition (eq. (11)) are imposed on these variables.

Note the manner in which the dual problem is generated from its corresponding primal problem. The positive constants, c_j, appearing in the dual function, v, are the coefficients of the posynomials whose terms are given by eq. (5). Each δ_j is associated with the j-th term, u_j, of the primal problem, and hence, each u_j of the posynomials is associated with one and only one of the dual variables, $\delta_1, \delta_2, \ldots, \delta_n$. Each λ_i in the dual problem comes from a forced constraint, $g_i \leq 1$, of the primal problem. Because the normality condition forces the weights of the objective function to sum to unity, the λ_0 corresponding to the objective function itself is unity, and thus it does not appear in eq. (7). This normality condition is the only part of the dual problem that distinguishes between the objective function, S, and a set of the inequality constraints, $g_i < 1$. The coefficient matrix (a_{jk}) that appears in the orthogonality condition, eq. (11), is the exponent matrix of the primal problem.

Since the optimal redundancy allocation problems under consideration have positive coefficients with the variables in the objective function, if the objective function can be transformed to a polynomial form, and all the resources requirement associated with each component of the ith constraint and the jth resource have positive values, then the geometric programming with the type of "minimization of posynomial subject to inequality constraints" will be considered here [3,4].

Referring to an N-stage parallel redundant system which has linear and separable constraints, the system reliability can be stated as

Maximize

$$R_s = \prod_{j=1}^{N} (1 - Q_j^{x_j}) \tag{12}$$

subject to

$$\sum_{j=1}^{N} p_{ij} x_j \leq b_i \,, \qquad i = 1, 2, \ldots, r \tag{13}$$

when $Q_j \leq 0.5$, which is a reasonable assumption for the component unreliability, then eq. (12) can be approximated as

maximize

$$R_s \simeq 1 - \sum_{j=1}^{N} Q_j^{x_j}$$

or, equivalently,

minimize

$$S = \sum_{j=1}^{N} Q_j^{x_j} \tag{14}$$

Since the stage unreliability is defined as

$$Q_j' = Q_j^{x_j} \tag{15}$$

hence eq. (14) can be expressed as

$$S = \sum_{j=1}^{N} Q'_j \tag{16}$$

Also, from the definition of eq. (15),

$$\ell n Q'_j = x_j \ell n Q_j ,$$

or

$$x_j = \frac{\ell n Q'_j}{\ell n Q_j} \quad , \; j = 1, 2, \ldots, N \tag{17}$$

Substituting x_j into eq. (13), we obtain

$$\sum_{j=1}^{N} p_{ij} \frac{\ell n Q'_j}{\ell n Q_j} \leq b_i , \qquad i = 1, 2, \ldots, r$$

Divided both side by $b_i > 0$,

$$\sum_{j=1}^{N} \frac{p_{ij}}{b_i \ell n Q_j} \ell_n Q'_j \leq 1,$$

or

$$\sum_{j=1}^{N} \left(- \frac{p_{ij}}{b_i \ell n Q_j}\right) (-1) \ell n Q'_j \leq 1 \tag{18}$$

If we define

$$k_{ij} = - \frac{p_{ij}}{b_i \ell n Q_j} \tag{19}$$

then

$$\sum_{j=1}^{N} (-k_{ij}) \ell n Q'_j \leq 1$$

or

$$\sum_{j=1}^{N} \ell n Q'^{\,-k_{ij}}_j \leq 1 \tag{20}$$

or

$$\ln \prod_{j=1}^{N} Q_j'^{\,-K_{ij}} \leq 1 \qquad (21)$$

Taking exponential on both side, then

$$\prod_{j=1}^{N} Q_j^{\,-K_{ij}} \leq e$$

or

$$e^{-1} \prod_{j=1}^{N} Q_j'^{\,-K_{ij}} \leq 1, \; i = 1, 2, \ldots, r \qquad (22)$$

The primal geometric programming problem is therefore formulated as minimize eq. (16) subject to eq. (22).

Assuming x_j to be continuous variables, the dual geometric programming formulation is

Maximize

$$v = \prod_{j=1}^{N} \left[\frac{1}{\delta_j}\right]^{\delta_j} \prod_{i=N+1}^{N+r} \left[\frac{e^{-1}}{\delta_i}\right]^{\delta_i} \prod_{i=1}^{r} (\lambda_i)^{\lambda_i} \qquad (23)$$

subject to

$$\sum_{j=1}^{N} \delta_j = 1 \qquad (24)$$

and

$$\delta_j - \sum_{i=N+1}^{N+r} K_{ij}\delta_i = 0, \; j = 1, 2, \ldots, N \qquad (25)$$

$$\lambda_i = \sum_{j=m_i}^{n_i} \delta_j \qquad ,i = 1, 2, \ldots, r \qquad (26)$$

$$\delta_i \geq 0, \; i = 1, 2, \ldots, r \qquad (27)$$

where δ_i, i = 1, 2, ..., r, are the dual variables corresponding to eq. (22) and δ_{r+j}, j =1, 2, ..., N, are the dual variables corresponding to eq. (16).

Eqs. (24) and (25) can be simultaneously solved to get δ^*_j, j = 1, 2, ..., N + r. Substitution of these results into eq. (23) gives rise to $v(\overset{*}{\delta})$. It has been proved that [2]

$$S(Q^*) = V(\delta^*), \tag{28}$$

and

$$u_j(\bar{Q}) = \bar{\delta}_j\ S(\bar{Q}'),\ j = 1, 2, \ldots, N+r \tag{29}$$

where

$$u_j = C_j \prod_{i=1}^{m} Q'_k{}^{a_{ik}},\ k = 1, 2, \ldots, m \tag{30}$$

From eqs. (28) and (29), Q'_k, k = 1, 2, ..., m can be optimally obtained.

Finally, we will apply eq. (15) to find the optimal allocations, x_j, j = 1, 2, ..., N.

3. <u>A Numerical Example</u>

Consider the problem in which N stages are connected in series and redundant components, $x_j - 1$, are added in parallel at each stage. The objective is to determine x_j at each stage, such that the system reliability is maximized and the cost and the weight constraints are not exceeded. The problem is

Maximize

$$R_s = \prod_{j=1}^{N} [1 - (1 - R_j)^{x_j}] \tag{31}$$

subject to

$$g_1 = \sum_{j=1}^{N} c_j x_j \leq C \tag{32}$$

$$g_2 = \sum_{j=1}^{N} w_j x_j \leq W$$

The constants associated with this problem are given as

Stage j	Cost c_j	Weight w_j	Probability of Survival R_j
1	1.2	1.0	0.80
2	2.3	1.0	0.70
3	3.4	1.0	0.75
4	4.5	1.0	0.85

$C = 56.0, \qquad W = 30.0$

The objective function can also be stated as

Minimize

$$Z = \sum_{j=1}^{4} Q_j' \tag{33}$$

subject to

$$e^{-1} \prod_{j=1}^{4} Q_j'^{\,-K_{ij}} \leq 1, \quad i = 1, 2 \tag{34}$$

where

$$Q_j' = Q_j^{x_j} = (1 - R_j)^{x_j} \tag{35}$$

$$K_{1j} = - \frac{c_j}{C \ell n Q_j} \quad , j = 1, 2, 3, 4 \tag{36}$$

$$K_{2j} = - \frac{w_j}{W \ell n Q_j} \quad , j = 1, 2, 3, 4 \tag{37}$$

The dual function is

$$v = \left(\frac{1}{\delta_1}\right)^{\delta_1} \left(\frac{1}{\delta_2}\right)^{\delta_2} \left(\frac{1}{\delta_3}\right)^{\delta_3} \left(\frac{1}{\delta_4}\right)^{\delta_4} \left(\frac{e^{-1}}{\delta_5}\right)^{\delta_5} \left(\frac{e^{-1}}{\delta_6}\right)^{\delta_6}$$

$$\cdot \; (\lambda_1)^{\lambda_1} (\lambda_2)^{\lambda_2} \tag{38}$$

where

$$\lambda_1 = \delta_5,\ \lambda_2 = \delta_6.$$

The normality condition becomes

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 = 1 \tag{39}$$

and the orthogonality conditions becomes

$$\delta_1 - [K_{11}\delta_5 + K_{21}\delta_6] = 0$$

$$\delta_2 - [K_{12}\delta_5 + K_{22}\delta_6] = 0$$

$$\delta_3 - [K_{13}\delta_5 + K_{23}\delta_6] = 0$$

$$\delta_4 - [K_{14}\delta_5 + K_{24}\delta_6] = 0 \tag{40}$$

Various methods can be applied to get the optimal δ_1^*, δ_2^*, δ_3^*, δ_4^*, δ_5^*, δ_6^*, and v^*.

From eqs. (39) - (40), we can express δ_1, δ_2, δ_3, δ_4, and δ_5 in terms of δ_6:

$$\delta_j = \frac{K_{1j}}{\sum_{j=1}^{4} K_{1j}} + \left(K_{2j} - \frac{K_{1j}\sum_{j=1}^{4} K_{2j}}{\sum_{j=1}^{4} K_{1j}}\right)\delta_6 \qquad j = 1, 2, 3, 4 \tag{41}$$

$$\delta_5 = \frac{1}{\sum_{j=1}^{4} K_{1j}} - \frac{\sum_{j=1}^{4} K_{2j}}{\sum_{j=1}^{4} K_{1j}}\ \delta_6 \tag{42}$$

Substituting egs (41) and (42) into eg. (38), the objective function is a one-dimensional function (in term of δ_6) to be maximized. The Golden Section method can be applied to find

$$V^* = 0.00207$$

$$\delta_6{}^* = 0.0046$$

Also, we can obtain, by substituting into egs (41) and (42),

$$\delta_1{}^* = 0.09969$$

$$\delta_2{}^* = 0.25537$$

$$\delta_3{}^* = 0.32786$$

$$\delta_4{}^* = 0.31707$$

$$\delta_5{}^* = 7.48310$$

Therefore,

$$Q_1' = 0.2^{x_1} = \delta_1 V^* = 0.0002064$$

$$Q_2' = 0.3^{x_2} = \delta_2 V^* = 0.0005286$$

$$Q_3' = 0.25^{x_3} = \delta_3 V^* = 0.0006787$$

$$Q_4' = 0.15^{x_4} = \delta_4 V^* = 0.0006563$$

From these equations, the optimal x_j, j = 1, 2, 3, 4, are found to be

$$x_1 = 5.27248$$

$$x_2 = 6.26699$$

$$x_3 = 5.26248$$

which gives the system reliability, R_s, 0.99793. After rounding off to the nearest integers, we get the optimal allocation:

$x_1 = 5$

$x_2 = 6$

$x_3 = 5$

$x_4 = 4$

The system reliability is 0.99747 with a cost slack of 1.4 and a weight slack of 10.

REFERENCES

1. Duffin, R. J., E. L. Peterson, and C. Zener, Geometric Programming, N.Y.: Wiley (1967).

2. Fan, L. T., L. E. Erickson, and C. L. Hwang, Geometric Programming, Institute for System Desing and Optimization, Vol. 5, K.S.U., Manhattan, Kansas 66502.

3. Federowicz, A. J., and M. Mazumdar, "Use of geometric programming to maximize reliability achieved by redundancy", Operations Research Vol. 16, No. 5, pp. 948-954 (Sept. - Oct. 1968).

4. Misra, K. B., and J. Sharma, "A new geometric programming formulation for a reliability problem", International Journal of Quality Control, Vol. 18, No. 3, pp. 497 - 503 (1973).

5. Nijkamp, P., Planning of Industrial Complexes by Means of Geometric Programming, Rotterdam Universtiy Press, Rotterdam, Netherlands, (1972).

6. Zener, C., Engineering Design by Geometric Programming, N.Y.: Wiley-Interscience, (1971).

3.9 INTEGER PROGRAMMING APPLIED TO OPTIMAL SYSTEM RELIABILITY

1. Introduction

In many problems the decision variables make sense only if they have integer values. Redundancy allocation in a system reliability optimization problem is a good example. If all the variables are integer, we have an integer programming problem which can be solved by an integer programming algorithm. A problem in which some, of the variables are required to be integers is a mixed integer programming problem. For example, if both of the redundancy allocations and element reliability at each stage are regarded as decision variables in a series system, we have a mixed integer programming problem.

In some situations, the decision variables are (assumed to be) continuous, even though they must be integers. The solution is obtained by rounding the fractional values of the optimal solution to integer values. This approach has, however, its risks. Although this is one approach there are pitfalls.

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:

Example 1 Linear Objective Function

The problem is to minimize a linear cost function

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

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

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

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

where

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

The constants associated with the problem are given as

$N = 2, \quad R_{s,min} = 0.9903, \quad W = 40$

$R_1 = 0.91, \; R_2 = 0.96,$

$c_1 = 5, \quad c_2 = 8,$

$w_1 = 9, \quad w_2 = 6$

Example 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 weight 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

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

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

Two sets of data are widely associated with this problem. The cost, weight, and probability of survival for the redundancies at each stage. The two data sets are listed below.

A. Consider the set of known data originally presented in [25, 26]

Stage j	Cost c_j	Weight w_j	Probability of Survival R_j
1	5	8	0.90
2	4	9	0.75
3	9	6	0.65
4	7	7	0.80
5	7	8	0.85

$C = 100, \quad W = 104$

The problem is also restricted that

$$0 \leq m_j \leq 4, \qquad j = 1, 2, \ldots, 5$$

B. Consider the set of known data used in [4, 15, 16, 29]

Stage j	Cost c_j	Weight w_j	Probability of Survival 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 5 Nonlinear Objective Function and Nonlinear Constraint Functions

In this example [17, 25, 26], 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

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

subject to

$$g_1 = \sum_{j=1}^{N} p_{j1} m_j + p_{j2} m_j^2 + p_{j3} \leq P$$

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

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

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

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

The constants assigned to this problem are:

$$N = 2,\ R_{s,min} = 0.85,\ m_{j,max} = 4,\ P = 37,\ C = 81,\ W = 38$$

j	v_j	p_{j1}	p_{j2}	p_{j3}	c_j	α_j	w_j
1	3	3	1	0	30	0	30
2	2	3	1	1	30	4	30

Example 4

The "object" is to maximize nonlinear system reliability subject to 3 nonlinear constraints with redundant components in each stage that are subject to type 1 failures [24].

Maximize

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

subject to

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

$$G_2(m) = 20(m_1 + \exp(-m_1)) + 20(m_2 + \exp(-m_2))$$

$$+ 20(m_3 + \exp(-m_3)) \geq 120,$$

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

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

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

The subsystems are subject to four failure models ($s_i = 4$) with one O 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 1.

Table 1

The type of failue and its failure probability for each element

subsystem	type of failure	failure probability
i	u	q_{iu}
1	O	0.01
	A	0.05
	A	0.10
	A	0.18
2	O	0.08
	A	0.02
	A	0.15
	A	0.12
3	O	0.04
	A	0.05
	A	0.20
	A	0.10

There are then at least five methods of solving these binds of problems in integer programming: Partial enumeration - Lawer & Bell, Implicit enumeration - Lemke & Spielberg, cutting plane method - Gomory, Branch and bound, and Implicit enumeration - Geoffrion. They are classified in Table 2.

Table 2 Classification of Examples and Approaches

Examples	Methods applied to the examples	References
Example 1 Min. Linear cost function s.t. $R_s \geq R_{s,min}$ Linear weight constraint	Partial enumeration - Lawler & Bell Implicit enumeration - Lemke & Spielberg	17 9
Example 2 Max. R_s s.t. 2 linear cost constraints	Cutting plane method - Gomory Branch and bound Partial enumeration - Lawler & Bell Partial enumeration Enumeration - Balas or Glover	25,26 4, 15, 16, 18 17 14 10
Example 3 Min. Nonlinear cost function s.t. $R_s \geq R_{s,min}$ 3 nonlinear constraints	Cutting plane method - Gomory Partial enumeration - Lawler & Bell	25,26 17
Example 4 Max. R_s with 2 classes of failure modes s.t. 3 nonlinear constraints	Cutting plane method - Gomory Implicit enumeration - Geoffrion	24 8

2. Partial Enumeration Method

Example 1

The integer programming problem of 0-1 type variables due to Lawler and Bell [11] is used to find the solution of the example. Lawler and Bell describe a programmed algorithm for solving discrete optimization problems with a monotonic objective function and arbitrary constraints.

A brief review of the Lawler-Bell method is provided in the 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 r constraints of the form

$$g_{i1}(x) - g_{i2}(x) \geq 0, \qquad i = 1,\ldots.,r \tag{1}$$

where

$$x \equiv \{x_1, x_2,\ldots,x_n\}$$

$$x_j = 0 \text{ or } 1, \quad j = 1,\ldots,n$$

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 $j = 1,\ldots,n$. This is the vector partial ordering. There is also the lexicographic or numerical ordering of these vectors obtained by identifying with each x, 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 $x \leq y$.

The method is basically a search method, which starts with $x = (0, 0,\ldots0)$ and examines the 2^n solution vectors in the numerical ordering described above. Further, the labor of examination is

subject to

$$g_1(m) = \sum_{j=1}^{2} \ln[1 - (1 - R_j)^{m_j+1}] - \ln R_{s,min} \geq 0$$

$$g_2(m) = W - \sum_{j=1}^{2} w_j m_j \geq 0 \tag{2}$$

Before m_1 and m_2, both ≥ 0, can be transformed to the variable of 0-1 type, it is necessary to estimate their upper bounds. This is done by substituting zero for all variables in the constraints, except the one for which the maximum value is to be found. Denote these by m^*_{ij}, then $m^u_j = \min (m^*_{ij})$, i = 1, 2,...r, is the upper bound for m_j. It is easy to show that both m^u_1 and m^u_2 are 3. We therefore can make the following substitution

$$m_1 = x_{11} + 2x_{12}$$

$$m_2 = x_{21} + 2x_{22} \tag{3}$$

where, x_{ij}, i = 1, 2, j = 1, 2, is either 0 or 1.

Now the problem is reformulated as

$$f(x) = 5x_{11} + 10x_{12} + 8x_{21} + 16x_{22}$$

$$g_{11}(x) = \ln[1 - 0.09^{x_{11}+2x_{12}+1}] + \ln[1 - 0.04^{x_{21}+2x_{22}+1}] - \ln 0.9903$$

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

$$g_{22}(x) = 9x_{11} + 18x_{12} + 6x_{21} + 12x_{22} - 25 \tag{4}$$

consideralby cut down by following certain 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 indicate the conditions under which certain vectors may be skipped.[1]

1) Test if $g_0(x) \leq g_0(\hat{x})$. If YES, skip to x* and repeat the operation; otherwise proceed to step 2).
2) Examine whether $g_{i1}(x^* - 1) - g_{i2}(x) \leq 0$ for i = 1,...,r. 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,...,r), 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 [11] 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,...1,), is found. In case one has skipped to a vector having numerical order higher the (1,..,1), designate this state by "overflow" and terminate the procedure. The least "costly" vector recorded provides the optimum solution.

One should not be overwhelmed by the number of trials. In practice the number of vectors to be examined may be quite small. For example, in an 11-variable problem with a total of 2^{11}-solution vectors, only 42 vectors were examined.

This example should first of all be formulated as

Minimize

$$f = \sum_{j=1}^{2} c_j m_j$$

Now the problem conforms to the Lawler-Bell algorithm. The solution is arrived at after examining only six vectors out of the 16 generated by the four binary variables of (3). The sequence of examination and the different rules applied are indicated in Table 3. The vector ordering used is also shown, viz., $x \equiv \{x_{12}, x_{22}, x_{11}, x_{21}\}$. 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 proceeds.

To begin with Table 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 $g_0(\hat{x})$. The true optimum is shown by the arrow in Table 3. Therefore, $m_1 = m_2 = 1$ from (3).

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 [27] requiring 11 binary variables, the solution was obtained by examining 42 of the 2^{11} solutions.

Table 3

x_{12}	x_{22}	x_{11}	x_{21}		
0	0	0	0	$g_{11}(x^* - 1) - g_{12}(x) < 0$	skip to x^* through step 2)
0	0	0	1	$g_{11}(x^* - 1) - g_{12}(x) < 0$	skip to x^* through step 2)
0	0	1	0	$g_{11}(x) - g_{12}(x) < 0$	change $x \to x + 1$ through step 3)
0	0	1	1	feasible, $g_0(x) = 13$	skip to x^* through step 3)
0	1	0	0	$g_0(x) < g_0(\hat{x})$	skip to x^* through step 1)
1	0	0	0	$g_{21}(x^* - 1) - g_{22}(x) < 0$	skip to x^* through step 2)

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

3. The Gomory Cutting Plane Method

Example 2

This problem is to be solved by Gomory's cutting plane method [5, 6]. It is noted that 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.

The reliability optimization problem can be formally stated as:

Optimize

$$\left.\begin{array}{ll} Z = \sum_{j=i}^{N} f_j(m_j) & \\ \text{subject to} & \\ \sum_{j=1}^{N} g_{ij}(m_j) \leq b_i & i = 1, 2, \ldots r \\ \prod_{j=1}^{N} R_j' \geq M & \end{array}\right\} \qquad (5)$$

where $m_j = 0.1, \ldots, m_j'$ $\qquad j = 1, 2, \ldots, N.$

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

Z = the objective to be maximized

N = the number of subsystems or stages

$f_j(m_j)$ = the objective function at stage j as a function of m_j

$g_{ij}(m_j)$ = 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

r = 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

M = the minimum acceptable reliability of the system

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. (5) can be solved as the following integer programming problem expressed by eq. (6):

$$
\left.
\begin{array}{ll}
\text{max/min} & \\
\quad Z = \sum_{j=1}^{N} \sum_{k=0}^{m_j'} \Delta f_{jk} m_{jk} & \\
\text{subject to} & \\
\quad \sum_{j=1}^{N} \sum_{k=0}^{m_j'} \Delta g_{ijk} m_{jk} \le b_i \quad , & i = 1, 2, \ldots r \\
\quad \sum_{j=1}^{N} \sum_{k=0}^{m_j'} \Delta \ln R_{jk} m_{jk} \ge \ln M & \\
\text{and} & \\
\quad m_{jk} = 1 & \text{for } k = 0 \\
\quad m_{jk} - m_{j,k-1} \le 0 & \text{for } k=1,\ldots m_j' \\
 & \quad j=1,\ldots N \\
\quad m_{jk} \ge 0 & \text{for all } j \text{ and } k
\end{array}
\right\} \quad (6)
$$

where in addition to the same notations used in problem (5),

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

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

$$\Delta f_{jk} = f_{jk} \qquad \text{for } k = 0$$
$$= f_{jk} - f_{j,k-1} \qquad \text{for } k = 1,\ldots,m_j$$

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.

$$\Delta g_{ijk} = g_{ijk} \qquad \text{for } k = 0$$
$$= g_{ijk} - g_{ij,k-1} \qquad \text{for } k = 1,\ldots,m_j' \quad i = 1,\ldots,r$$

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.

$$\Delta \ell nR_{jk} = \ell nR_{jk} \qquad \text{for } k = 0$$
$$= \ell nR_{jk} - nR_{j,k-1} \qquad \text{for } k = 1,\ldots,m_j$$

is the change in ℓnR_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.

The equivalence of (5) and (6) is easily illustrated using the quantities and terms which are defined above and assuming that m_j units are used at stage j. By substituting these quantities into the objective function Z of (5), yields

$$Z = \sum_{j=1}^{N} f_i(m_j) = \sum_{j=1}^{N} \sum_{k=0}^{m_j} \Delta f_{jk}$$

Likewise the restriction equation of (5) becomes

$$\sum_{j=1}^{N} g_{ij}(m_j) = \sum_{j=1}^{N} \sum_{k=0}^{m_j} \Delta g_{ijk}$$

Now by taking logarithms of the reliability restriction of (5) and with the appropriate substituting, the follwoing equivalent restriction is obtained.

$$\ell nM \leq \sum_{j=1}^{N} \ell nR_j = \sum_{j=1}^{N} \sum_{k=0}^{m_j} \Delta\ \ell n\ R_{jk}$$

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

$$m_{jk} = 1 \qquad \text{for } 0 \leq k \leq m_j$$
$$= 0 \qquad m_j < k \leq m_j' \tag{7}$$

with the obvious result that

$$m_j = \sum_{k=1}^{m'_j} m_{jk} \tag{8}$$

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 (7) as restrictions. Equation (7) includes the requirement that each subsystem shall contain at least one component. This is accomplished by including the following restrictions

$$m_{jk} = 1 \qquad \text{for } k = 0 \quad j = 1,\ldots,N. \tag{9}$$

The remaining part of (7) 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

$$m_{jk} \leq m_{j,k-1} \qquad k = 1,\ldots,m_j' \quad j = 1,\ldots,N. \tag{10}$$

Thus including (9) and (10) completes the formulation of the problem (5) as an integer programming problem as stated by (6).

After applying the set A data of Example 2 shown in the Introduction section, the problem can be illustrated in Fig. 1 in the required integer programming formulation. The equations in Group I insure that one basic unit is in each stage. The Group II equations allow the kth redundant unit to be in the solution only if the (k-1)th redundant unit is included and require the m_{jk} variables to be either zero or one. The system restrictions on cost and weight are in Group III. The c_j equation, representing the Δ in R_{jk} values, is the objective function to be maximized. This problem was solved by an integer programming algorithm and the solution is as follows

$m_{10} = 1$	$m_{20} = 1$	$m_{30} = 1$	$m_{40} = 1$	$m_{50} = 1$
$m_{11} = 1$	$m_{21} = 1$	$m_{31} = 1$	$m_{41} = 1$	$m_{51} = 1$
$m_{12} = 1$	$m_{22} = 1$	$m_{32} = 1$	$m_{42} = 1$	$m_{52} = 1$
	$m_{23} = 1$	$m_{33} = 1$	$m_{43} = 1$	
		$m_{34} = 1$		

			Stage I					Stage II					Stage III					Stage IV					Stage V				
			m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{20}	m_{21}	m_{22}	m_{23}	m_{24}	m_{30}	m_{31}	m_{32}	m_{33}	m_{34}	m_{40}	m_{41}	m_{42}	m_{43}	m_{44}	m_{50}	m_{51}	m_{52}	m_{53}	m_{54}
Group I	1	=	1																								
	1	=						1																			
	1	=											1														
	1	=																1									
	1	=																					1				
Group II	0	≥	-1	1																							
	0	≥		-1	1																						
	0	≥			-1	1																					
	0	≥				-1	1																				
	0	≥						-1	1																		
	0	≥							-1	1																	
	0	≥								-1	1																
	0	≥									-1	1															
	0	≥											-1	1													
	0	≥												-1	1												
	0	≥													-1	1											
	0	≥														-1	1										
	0	≥																-1	1								
	0	≥																	-1	1							
	0	≥																		-1	1						
	0	≥																			-1	1					
	0	≥																					-1	1			
	0	≥																						-1	1		
	0	≥																							-1	1	
	0	≥																								-1	1
Group III	100	≥		5	5	5	5		4	4	4	4		9	9	9	9		7	7	7	7		7	7	7	7
	104	≥		8	8	8	8		9	9	9	9		6	6	6	6		7	7	7	7		8	8	8	8
c_j		=	-0.10536	0.09531	0.00904	0.00090	0.00009	-0.28768	0.22314	0.04879	0.01183	0.00294	-0.43078	0.30010	0.08686	0.02870	0.00985	-0.22314	0.18232	0.03279	0.00643	0.00128	-0.16252	0.13976	0.01938	0.00287	0.00043

Fig. 1

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

$m_1 = 2$ redundant units at stage 1

$m_2 = 3$ redundant units at stage 2

$m_3 = 4$ redundant units at stage 3

$m_4 = 3$ redundant units at stage 4

$m_5 = 2$ redundant units at stage 5

This configuration has a cost of 93 units where the limit is 100 units and weighs 104 units which is equal to the limit. The system reliability $R_s = 0.985$ or $\ln R_s = -0.015175$.

4. The Branch and Bound Method

<u>Example 2</u>

Example 2 can also be solved by the branch and bound method [4, 28], which is briefly introduced as follows.

Problem A: Maximize total system reliaibility

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

subject to the constraints:

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

If we make the following transformations:

$$c_{ik} = \ell n(1 - p_i^{k+1}) - \ell n(1 - p_i^{k}),$$

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

then Problem A can be identically formulated as.

Problem B: Maximize

$$Z = \sum^{i=m} \sum^{k=\infty} c \quad x$$

subject to constraints:

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

$$x_{ik} = 0 \text{ or } 1; \qquad \text{and} \qquad x_{ik} = 0 \text{ implies } x_{il} = 0 \text{ if } l > k.$$

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

Let $X = (x_{ik})$ 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.

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

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

$$\sum_{i=1}^{i=m} a_{ij}(k_i + 1) \le d_j.$$

Hence $N = (n_i | n_i = k_i + 1)$ 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

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

$$= \sum_{i=1}^{i=m} \sum_{k=1}^{k=k_i} \{\ell n(1 - p_i^{k+1}) - \ell n(1 - p_i^{k})\}$$

$$= \sum_{i=1}^{i=m} \{\ell n(1 - p_i^{k_i+1}) - \ell n(1 - p_i)\}$$

$$= \ell n R - \sum_{i=1}^{i=m} \ell n(1 - p_i).$$

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.

Bounding Procedure

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

Maximize

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

Subject to a single constraint

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

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

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

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

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 $r_{ik} > r_{i,k+1}$, or $\max_{i,k}\{r_{ik}\} = \max_i\{r_{i1}\}$. Hence $Z \leq \max_i\{r_{i1}\} \cdot b_j$.

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

$$Z \leq \max_i\{r_{i1}\} \cdot b_j \text{ for any } j \leq \min_j[\max_i\{r_{i1}\} \cdot b_j].$$

Consequently, the optimal feasible solution Z^* is bounded by the quantity $\min_j[\max_i(r_{i1}) \cdot b_j]$. 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 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 to 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 bounds for the inclusive and exclusive branches (subsets) can be computed as follows.

Inclusive branch:

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

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

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

Hence the upper bound on the inclusive branch equals

$$\sum_{i=1}^{i=m} \sum_{k=1}^{k=k_i} c_{ik} + \min_j \left(\max_{i \notin I} (r_{i,k_i+1}) \cdot b_j'\right). \tag{11}$$

Exclusive branch:

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

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

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

Hence the upper bound on the exclusive branch equals

$$\sum_{i=1}^{i=m} \sum_{k=1}^{k=k_i} c_{ik} + \min_j \left(\max_{i \notin I'} (r_{i,k_i+1}) \cdot b_j'\right]. \tag{12}$$

The first forward solution is obtained by selecting the component for a branching decision that yields the highest upper bound on the inclusive branch and always branches into the inclusive branch. During the forward procedure, the bounds for the exclusive branch are stored as temporary bounds. The bounds for the inclusive branch are not stored explicitly. After a complete solution is reached (i.e. at least one of the resources is depleted completely giving a 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 index such that $X_{ik} = 1$ in the solution X^0, for all $i \neq i^*$. These revised upper bounds are then compared with the objective function $Z(X^0)$. Only those branches need be explored further for which the upper bound exceeds $Z(X^0)$. The method of branch and bound is used to solve this example where the set B data of Example 2 shown in the Introduction are assigned. By eq. (6), b_j=(35.6, 16). The forward procedure, during which the initial upper bounds are computed, is shown in Table 4. When compared with the upper bounds, Z = 1.02167 is shown to be optimal, yielding the results shown in Table 5. the same result as obtained by Proschan and Bray. [29]

Table 4

Illustration of the Forward Procedure

Level	Stage selected	Upper bound
1	2	0.67575
2	1	0.80308
3	3	0.74449
4	4	0.87040
5	2	0.93613
6	1	0.98437
7	3	0.96546
8	2	1.00108
9	4	1.00627
10	1	1.01451
11	3	1.01061
12	2	1.01833
13	1	1.02039
14	2	1.02048

Initial solution Z = 1.02167

Tabel 5

The Optimal Configuration

Stage i	No. of parallel components n_i
1	5
2	6
3	4
4	3
Total system reliability = 0.99169	

5. The Gormory's Cutting Plane Method

Example 3

To solve this problem by Gomery's cutting plane method [5,6], we have to transfer the constraint,

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

into

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

This can be written as

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

$$0.1625 \geq -\sum_{j=1}^{2} \sum_{k=0}^{4} \Delta \ln R_{jk} m_{jk}.$$

In this example, the units have the same reliability as the units in exmaple 2, thus the $\Delta \ln R_{jk}$ values are the same as the objective function c_j values of example 2.

The objective is to determine the m_j, the number of redundant units at 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. Problems with few linear constraints such as example 1 can be readily solved by methods presented in [20, 27], but it seems that these methods are inadequate for solving this second example which includes multiple nonlinear restraints. The integer programming formulation of this problem is illustrated in Fig. 2.

The Group I equations represent the greater-than-restrictions, 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 the m_{jk} variables to be either zero or one. A minimization problem is converted to a maximization problem by multiplying the objective function (-1). This problem was converted, therefore 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 $\ln R_s = -0.1065$.

			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	$\le$	30	11	23	27	29	26	11	23	27	29
	38	$\le$	0	23	13	6	2	0	23	13	6	2
Group II	1	=	1									
	1	=						1				
Group III	0	$\ge$	-1	1								
	0	$\ge$		-1	1							
	0	$\ge$			-1	1						
	0	$\ge$				-1	1					
	0	$\ge$						-1	1			
	0	$\ge$							-1	1		
	0	$\ge$								-1	1	
	0	$\ge$									-1	1
Group IV	37	$\ge$	0	4	6	8	10	1	4	6	8	10
	0.1625	$\ge$	0.1054	-0.0953	-0.0090	-0.0009	-0.0001	0.2877	-0.2231	-0.0488	-0.0118	-0.0029
	c_j	=	0.0	-1.81959	-0.38768	0.19911	0.38415	0.0	-1.21306	-0.25846	0.13274	0.25510

Fig. 2

Example 3

To solve this problem by the integer programming due to the Lawler-Bell algorithm, we reformulate the problem as:

Minimize the cost function

$$g_0(m) \equiv [3m_1 \exp(-m_1/2)] + [2m_2 \exp(-m_2/2)]$$

subject to the constraints

$$g_1(m) \equiv 37 - [3m_1 - m_1^2] - [3m_1 + m_2^2 + 1] \geq 0$$

$$g_2(m) \equiv [30m_1 + \exp(-m_1))] + [30(m_2 + \exp(-m_2)) - 4] - 81 \geq 0$$

$$g_3(m) \equiv [30m_1 \exp(-m_1/4)] + [30m_2 \exp(-m_2/4)] - 38 \geq 0$$

$$g_4(m) \equiv [\ln(1 - 0.1^{m_1+1})] + [\ln(1 - 0.25^{m_2+1})] + 0.1625 \geq 0. \tag{13}$$

$g_4(m)$ refers to the reliability constraint.

The problem can be transformed to the type (1) by substituting $m_1 \equiv x_{11} + 2x_{12} + 4x_{13}$ and $m_2 \equiv x_{21} + 2x_{22} + 4x_{23}$. The maximum value of either m_1 or m_2 does not exceed 5 from the scrutiny of the constraints of (13).

The different functions in (1) are defined as follows

$$g_{11}(x) = g_{22}(x) = g_{32}(x) = g_{42}(x) = 0$$

$$\begin{aligned}-g_{12}(x) = 36 &- 3(x_{11} + 2x_{12} + 4x_{13}) \\ &- (x_{11} + 2x_{12} + 4x_{13})^2 - 3(x_{21} + 2x_{22} + 4x_{23}) \\ &- (x_{21} + 2x_{22} + 4x_{23})^2\end{aligned}$$

$$\begin{aligned}g_{21}(x) = 30[(x_{11} &+ 2x_{12} + 4_{13}) \\ &+ \exp(-(x_{11} - 2x_{12} + 4x_{13}))] \\ &+ 30[(x_{21} + 2x_{22} + 4x_{23}) \\ &+ \exp(-(x_{21} + 2x_{22} + 4x_{23}))] - 85\end{aligned}$$

$$g_{31}(x) = 30[(x_{11} + 2x_{12} + 4x_{13})$$

$$\cdot \exp(-(\frac{x_{11} + 2x_{12} + 4x_{13}}{4}))]$$

$$+ 30[(x_{21} + 2x_{22} + 4x_{23})$$

$$(-(\frac{x_{21} + 2x_{22} + 4x_{23}}{4}))] - 38$$

$$g_{41}(x) = \ln(1 - 0.1^{x_{11}+2x_{12}+4x_{13}+1})$$

$$+ \ln(1 - 0.25^{x_{21}+2x_{22}+4x_{23}+1}) + 0.1625. \tag{14}$$

The solution with variable ordering indicated in Table 5 is obtained in nine steps only, whereas the complete set consists of 64 vectors. The minimum $g_0(x)$ recorded is 1.082680 (shown by the arrow in Table 6) for which allocation is $m_1 = 0, m_2 = 4$, $\ln R_s = -0.106563$, and $R_s = 0.89892$.

Table 6

x_{13}	x_{23}	x_{12}	x_{22}	x_{11}	x_{21}		
0	0	0	0	0	0	$g_{21}(x^* - 1) - g_{22}(x) < 0$	skip to x^* through step 3)
0	0	0	0	0	1	$g_{21}(x^* - 1) - g_{22}(x) < 0$	skip to x^* through step 3)
0	0	0	0	1	0	$g_{21}(x^* - 1) - g_{22}(x) < 0$	skip to x^* through step 3)
0	0	0	2	0	0	$g_{31}(x) - g_{32}(x) \quad 0$, $x \rightarrow x + 1$	
0	0	0	1	0	1	feasible, $g_0(\hat{x}) = 1.338780$	skip to x^* through step 3)
0	0	0	1	1	0	$g_0(x)Kg_0(\hat{x})$	skip to x^* through step 1)
0	0	1	0	0	0	$g_0(x)Kg_0(\hat{x})$	skip to x^* through step 1)
0	1	0	0	0	0	feasible, $g_0(\hat{x}) = 1.082680$	skip to x^* through step 3)
1	0	0	0	0	0	$g_0(x)Kg_0(\hat{x})$	skip to x^* through step 1)

$x^* = 1(000000)$, i.e., overflow.

Example 4

Geoffrion's implicit enumeration method is to solve the problem with two classes of failure modes. A formulation by 0-1 linear programming [3, 8] is introduced. We state the original system reliability problem as

Problem A

Maximize

$$R_s = \prod_{i=1}^{N} [1 - Q_i(m_i)] \tag{15}$$

subject to

$$G_t(m) = \prod_{i=1}^{N} g_{ti}(m_i) \leq b_t, \quad t = 1,2,\ldots,T. \tag{16}$$

where

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

$Q^O(m_i)$ and $Q^A(m_i)$ are the unreliabilities of subsystem i obtained for class O failure modes and for class A failure modes, respectively. [24]

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

$$x_{ij} = \begin{cases} 1; \text{ allocate } j \text{ elements to subsystem } i, \\ 0; \text{ otherwise}. \end{cases} \tag{17}$$

When we introduce this 0 - 1 variable the nonlinear system reliability (Example 4) of the NIP - m problem, we get the following linearized objective function:

$$f(X) = \sum_{i=1}^{N} \sum_{j=r_i}^{v_i} c_{ij} x_{ij}, \tag{18}$$

where, for all i and j,

$$c_{ij} \equiv \ln \left[1 - \left\{ \sum_{u=1}^{h_i} Q^p_{iu}(j) + \sum_{u=h_i+1}^{s_i} Q^s_{iu}(j) \right\} \right]$$

$$Q^p_{iu}(j) \equiv 1 - (1 - q_{iu})^j, \quad Q^s_{iu}(j) \equiv (q_{iu})^j \tag{19}$$

When we introduce the 0-1 variable into the T nonlinear constraints (16), we get

$$g_t(X) = \sum_{i=1}^{N} \sum_{j=r_i}^{v_i} a_{tij} x_{ij} \leq b_t, \qquad t = 1,2,\ldots,T. \tag{20}$$

$$a_{tij} \equiv g_{ti}(j), \text{ for all } t, i, \text{ and } j. \tag{21}$$

By definition of 0-1 variable (17), we add the following N linear constraints to the constraints (20):

$$g_{T+i}(X) \equiv 1 - \sum_{j=r_i}^{v_i} x_{ij} = 0, \qquad i = 1,2,\ldots,N. \tag{22}$$

By introducing the 0-1 variable, we have thereby reformulated problem A into a ZOLP problem which maximizes the linear objective function (18)-(19) subject to the T + N linear constraints (20)-(22). This is the ZOLP-m problem. It is proved in the next page that there is a one-to-one correspondence between the NIP-m and the ZOLP-m problem proposed here.

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

$$= \sum_{i=1}^{N} \left\{ \sum_{j\varepsilon Z_i^+} \ln[1 - \{Q_i^O(j) + Q_i^A(j)\}]x_{ij} + \sum_{j\varepsilon \bar{Z}_i^+} \ln[1 - \{ Q_i^O(j) + Q_i^A(j)\}]x_{ij} \right\}, \tag{23}$$

where $Q_i^O(j) \equiv \sum_{u=1}^{h_i} Q_{iu}^p(j)$, $Q_i^A(j) \equiv \sum_{u=h_i+1}^{s_i} Q_{iu}^s(j)$, and $Z_i = \{j;\ r_i,\ r_i+1, \ldots, v_i\}$

is set of subsystem i and a direct sum of the Z_i and $\bar{Z}_i$ (which are a partitioning of Z_i). Let X* be a feasible solution to the ZOLP-m problem. Then, (23) is as follows:

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

$$= \sum_{i=1}^{N} \ln[1 - \{ Q_i^O(j_i^*) + Q_i^A(j_i^*)\}] = \ln R(j^*), \tag{24}$$

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

Now we prove that (20) and (21) are correct. It is obvious that (21) is necessary. In order to prove that it is sufficient, substitute (21) into (20).

$$g_t(X) = \sum_{i=1}^{N} \sum_{j=r_i}^{u_i} g_{ti}(j)x_{ij} = \sum_{i=1}^{N} \left\{ \sum_{j\varepsilon Z_i^+} g_{ti}(j)x_{ij} + \sum_{j\varepsilon \bar{Z}_i^+} g_{ti}(j)x_{ij} \right\}, \qquad t = 1, 2, \ldots, T. \tag{25}$$

Let X* be a feasible solution to the ZOLP-m problem, then (25) is

$$g_t(X) = \sum_{j=1}^{N} \sum_{j \varepsilon Z_i^+} g_{ti}(j)x_{ij} = \sum_{i=1}^{N} g_{ti}(j_i^*)$$
$$= G_t(j^*) \leq b_t, \qquad t = 1,2,\ldots,T. \tag{26}$$

The Zolp-m problem is to maximize the linear objective function (18)-(19) subject to the linear constraints (20)-(22) for N = 3 and T = 3, where the coefficients a_{tij} of (11) for j = 1, 2, 3, 4 are:

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

$$a_{2ij} = -20(j + \exp(-j)), \; a_{3ij} = 20j \exp(-j/4), \text{ for } i = 1, 2, 3,$$

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

The ZOLP-m problem is illustrated in Table 7 in the required ZOLP-m formulations which 1000 times the coeffcient c_{ij} for all i and j of the linear objective function (18). The variables are (for j = 1, 2, 3, 4):

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

The feasible and optimal solutions of the ZOLP-m example are shown in Table 8; the optimal solutions are $x_2 = 1$, $x_5 = 1$, and $x_{11} = 1$.

Table 7

Objective Function, x_{ij}

i j:	1	2	3	4
1	24.7	23.3	38.7	50.2
2	123.3	243.4	332.7	417.0
3	26.3	112.1	161.3	204.6

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 8

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				

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

REFERENCES

1. Garfinkel, R. S., and G. L. Nemhauser, Integer Programming, N.Y.: Wiley (1972).

2. Gen, M., H. Okuno, and S. Shinofuji, "An optimizing method in system reliability with failure-modes by implicit enumeration algorithm", J. of the Operations Research of Japan, Vol. 19, pp. 99-116 (June 1976).

3. 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).

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

5. Gomory, R. E., "All integer programming algorithm", IBM Res. Center Rept. RC-189, Jan. 1960, or in "Industrial Scheduling" edited by J. F. Muth and G. L. Thompson, Prentice Hall, (1963).

6. Gomory, R. E., "Outline of an algorithm for integer solutions to linear problem," Bulletin of the American Mathematical Society, Vol. 64, No. 5, pp. 275-278 (1958).

7. Hu, T. C., Integer Programming and Network Flows, Mass.: Addison-Wesley Publishing Comp. (1969).

8. 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).

9. 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).

10. Kolesar, P. J., "Linear programming and the reliability of multicomponent systems", Naval Research Logistics Quarterly, Vol. 15, pp. 317-327 (Sept. 1967).

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

12. Lawler, E. L. and D. E. Wood, "Branch-and-bound-methods: A survey", Operations Research, Vol. 14, pp. 699-719, (July-Aug. 1966).

13. Lemke, C. E., and Spielberg, K., "Direct Search Algorithms for Zero-One and Mixed Integer Programming," Operations Research, 15, 5, 1967, pp. 892-914.

14. 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).

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

16. 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).

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

18. 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).

19. Mizukami, K., "Optimum redundancy for maximum system reliability by the method of convex and integer programming". Operations Research, Vol. 16, pp. 392-406 (March-April 1968).

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

21. Salkin, H. M., Integer Programming, Mass.: Addison Wesley (1975).

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

23. Taha, H. A., Integer Programming - Theory, Applications and Computations, N. Y.: Academic Press (1975).

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

25. 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).

26. Tillman, F. A., and J. M. Liittschwager, "Integer programming formulation of constained reliability problems", Management Science, Vol. 13, No. 11, pp. 887-899 (July 1967).

27. Bellman, R. E. and S. E. Dreyfus, "Dynamic programming and the reliability of multi-component devices," Operations Research, Vol. 6, pp. 200-206 (Mar.-Apr. 1958

28. Balas, E., "A note on the branch-and-bound principle," Operations Research, Vol. 1 pp. 442-445 (1968).

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

3.10 OTHER METHODS APPLIED TO THE SYSTEM RELIABILITY OPTIMIZATION PROBLEMS

1. Introduction

In addition to the methods presented in the previous sections, there are several other methods that have been used for the system reliability optimization problems. A classical approach [1, 7, 11, 13, 15, 16] is to maximize the system reliability without considering the "cost". Minimum effort to increase the system reliability is of primal interest. Parametric method involving a transformation of the objective function into a simplified form so that either the method of Lagrange multiplier and the Kuhn-Tucker conditions [4] or modified Box's method [3] can be applied to solve the transformed problem.

Linear programming has sometimes been included in reliability optimization techniques for solving (a) an optimization problem with a linear form of non-negative variables subject to a system of linear inequalities [9, 17], or (b) an original nonlinear optimization problem having been transformed to a standard linear form which can be solved by linear programming. Separable programming [21, 23] is a typical technique to handle this formulation.

Stochastic method has also been used in reliability problems to maximize system reliability subject to cost restraints [10]. The method is based on a stochastic approach in which probability distributions are attached to families of allocations. Random search technique [5] and other miscellaneous optimization techiniques [6, 8, 12, 14, 19, 20] are sometimes applied to system reliability optimization problems.

Illustrations are given in the following by the classical approach, parametric method, linear programming, and separable programming.

2. A Classical Approach

Gordon [11], and Moskowitz and McLean [13] may be the first two groups using the graphical techniques in optimum component redundancy for maximum system relialbility. Their objective was also to develop a general mathematical solution for the optimum number of redundant elements in a system, while the reliabilities of the individual components are known, but without considering "cost" constraints. The figures which show the overall reliability as a function of complexity, reliability of components, and redundancy are presented so that the optimal solution can be pointed out from the figures.

Basing on a theorem of Albert [1], Lloyd and Lipow [11] introduced an effort function, which is required to accomplish the system reliability of a series configuration from the present reliability, R_s, to a desired higher level, $\bar{R}_s$. Let $R_1, R_2, \ldots, R_n$ denote the subsystem reliabilities, the system reliability can be given by

$$R_s = \prod_{i=1}^{n} R_i \tag{1}$$

Since $\bar{R}_s > R_s$, it is required to increase at least one of the R_i's to the point that the required reliability, $\bar{R}_s$, will be met, in accordance with eq. (1). To accomplish such an increase takes a certain effort, which is to be alloted in some way among the subsystems. The desired system reliability achieved with minimum effort is given as follows.

(A) Order the known reliabilities $R_1, R_2, \ldots, R_n$ in nondecreasing order (we assume now that such an ordering is implicit in the notation) so that

$$R_1 \leq R_2 \leq \cdots \leq R_n \tag{2}$$

(B) Increase each of the reliabilities $R_1, R_2, \ldots, R_{K_0}$ to the same value $\bar{R}_0$; but do not attempt to increase the reliabilities $R_{K_0+1}, \ldots, R_n$.

The number K_0 is determined as

K_0 = maximum value of j such that

$$R_j < \left(\frac{\bar{R}_s}{\prod_{i=j+1}^{n+1} R} \right)^{1/j} = r_j \text{ (say)} \tag{3}$$

where $R_{n+1} = 1$ by definition.

The number $\bar{R}_0$ is determined as

$$\bar{R}_0 = \left(\frac{\bar{R}_s}{\prod_{j=K_o+1}^{n+1} R_j} \right)^{1/K_0} \tag{4}$$

(C) It is evident that the system reliability will then be $\bar{R}_s$, since

$$\text{new reliaiblity} = \bar{R}_0^{K_0} R_{K_0+1} \ldots R_n = \bar{R}_0^{K_o} \prod_{j=K_0+1}^{n+1} R_j \tag{5}$$

and by using eq. (4) we immediately obtain

$$\text{new reliability} = \bar{R}_s$$

<u>A Numerical Example</u>

Let $(R_1, R_2, R_3, R_4, R_5, R_6) = (0.75, 0.80, 0.87, 0.90, 0.95, 0.99)$, then

$$R_s = \prod_{j=1}^{6} R_j = 0.4418.$$

The required value of system reliability is $\bar{R}_s = 0.53$. Suppose that we did not consider the selection of K_o by eq. (3) but arbitrarily decided to set $K_o = 1$ and use eq. (4). We would then obtain

$$\bar{R}_o = \left(\frac{0.53}{\prod_{j=2}^{6} R_j \quad x\ 1} \right)^{1/1} = 0.8996 .$$

and we would have

$$\bar{R}_s = 0.53 \quad = \quad 0.8996 \text{ x } 0.80 \text{ x } 0.87 \text{ x } 0.90 \text{ x } 0.95 \text{ x } 0.99$$

as desired. However, the theorem tells us that the effect to increase reliability has not been allotted in an optimum manner; i.e., more effort has been used than is necessary. Rather, we should determine K_o by eq. (3). To do this we calculate the quantities:

$$r_6 = (\frac{0.53}{1})^{1/6} = 0.8996$$

which is smaller than $R_6 = 0.99$. Therefore the 6th component is good enough. Similarly

$$r_5 = (\frac{0.53}{0.99x1})^{1/5} = 0.8825$$

which is smaller than $R_5 = 0.95$;

$$r_4 = (\frac{0.53}{0.99 \text{ x } 0.95 \text{ x } 1})^{1/4} = 0.8664$$

which is smaller than $R_4 = 0.90$; and

$$r_3 = (\frac{0.53}{0.99 \text{ x } 0.95 \text{ x } 0.90 \text{ x } 1}) = 0.8551$$

which is also smaller than $R_3 = 0.87$; therefore, components of stages 5, stages 5, 4, and 3 are good enough. However,

$$r_2 = \left(\frac{0.53}{0.99 \times 0.95 \times 0.90 \times 0.87 \times 1}\right)^{1/2} = 0.8484$$

which is greater than $R_2 = 0.80$. Therefore the 2nd component is not good. Since 2 is the largest subscript j such that $R_j < r_j$, then $K_0 = 2$, which means to achieve the system reliability, $R_s = 0.53$, the minimum effort to be allotted is to increase the 1st and 2nd component from 0.70 and 0.80 to the same level. $\bar{R}_0 = 0.8484$; whereas the rest components are left at their original level. The resulting reliability of the entire system is, as required,

$$\bar{R}_s = 0.53 = (0.8484)^2 \times 0.87 \times 0.90 \times 0.95 \times 0.99.$$

Effort Function Minimization

Effort function G(x, y), of a system is defined as the amount of effort required to increase the system reliability, x, to a higher level, y. Any cost, weight, volume, or power demand can be regarded as special kind of effort function, whether they are mathematically well described or not. Therefore, the cost minimization problem is an effort function minimization problem. The effort function always satisfies the following requirements:

1. $G(x, y) \geq 0$, which means the increasing of reliability from lower level, x, to higher level, y, will always need at least zero effort.
2. $G(x, y)$ is nondecreasing in y for fixed x and nonincreasing in x for fixed y; eg.

$$G(0.7, 0.8) \leq G(0.7, 0.85)$$

$$G(0.6, 0.8) \geq G(0.7, 0.8)$$

3. If $x \leq y \leq z$, $G(x, y) + G(y, z) = G(x, z)$, which states that the amount of effort to increase the reliability from x to z is equal to the sum of efforts to increase the reliability from x to y, then from y to z. Namely, $G(x, y)$ is additive.
4. $G(0, y)$ has a derivative $h(y)$ such that $yh(y)$ is strictly increasing in y, $0 < y < 1$.

For an N-stage series system, we denote R_i and R_s the reliabilities of the ith stage and the system respectively. If $\bar{R}_s$ is the minimum requirement of the system reliability and $\bar{R}_i$ the optimal ith stage reliability, then we can readily define the effort function minimization problem as

Minimize

$$\sum_{i=1}^{N} G(R_i, \bar{R}_i)$$

subject to

$$\prod_{i=1}^{N} \bar{R}_i \geq \bar{R}_s$$

To solve this optimization problem, R_i, $i = 1, 2, \ldots, N$, $\bar{R}_s$, and the effort function $G(R_i, \bar{R}_i)$ should be given, then various optimization, eg. dynamic programming, the method of Lagrange multiplier and the Kuhn-Tucker conditions, GRG, etc. can be applied to reach the optimal solution.

3. Parametric Method

Principle and Historical Background

Parametric approach was originally used in evaluating system reliability, especially when the number of components in a system was large or the system configuration complex. Probability was treated as a point in a Cartesian frame and formulas were derived to evaluate the system reliability by assigning a parametric value to it [2].

If the probability of success of any event is x, hence the probability of failure is $y = 1 - x$, then the parametric ϕ and θ associated with x and y are defined by

$$\phi \equiv \tan\theta = \frac{y}{x} = \frac{y}{1-y} = \frac{1-x}{x} . \tag{6}$$

By this transformation, the complex system , whether in the form of bridges, delta-star, or star-delta, can be expressed by the combinations of these parameters assigned in each subsystem. Then the system reliability can be automatically obtained by transforming back from eq. (6).

Parametric method are just an intermediate step to transfer the objective function in terms of component reliability to the one in terms of the parameters, ϕ, subject to "cost" constraints, therefore, the objective function having been formulated in parametric forms can be solved by any applicable nonlinear programming technique. The method of Lagrange multipliers and the Kuhn-Tucker conditions [4] and the modified Box method [3] are two applicable ones.

Formulation of the Problem

The problem formulation by the parametric approach is mainly on the transformation of the objective function by eq. (6).

For an N-stage series configuration, the system reliability is known as

$$R_s = \prod_{j=1}^{N} R_j' \tag{7}$$

where

$$R_j' = [1 - (1 - R_j)^{x_j}] \tag{8}$$

If the parameters ϕ_s and ϕ_j' are defined as

$$\phi_s = \frac{1-R_s}{R_s} \qquad \left(\text{or } R_s = \frac{1}{1+\phi_s}\right) \tag{9}$$

and

$$\phi_j' = \frac{1 - R_j'}{R_j'} \qquad \left(\text{or } R_j' = \frac{1}{1+\phi_j'}\right) \tag{10}$$

respectively, then eq. (7) can be represented as

$$\phi_s + 1 = \prod_{j=1}^{N} (1 + \phi_j') \tag{11}$$

In most reliability studies, we are dealing with components having a relatively high value of R_j'. Using this fact, eq. (11) can be expressed as

$$\phi_s \simeq \sum_{j=1}^{N} \phi_j' \tag{12}$$

From definition in eq. (6), we have

$$\phi_j' = \tan \theta_j' = \frac{Q_j'}{1 - Q_j'} \tag{13}$$

and

$$\phi_j = \tan \theta_j = \frac{Q_j}{1 - Q_j} \tag{14}$$

Then from eqs. (13) and (14), it is easy to find that

$$\phi_j' = \frac{1}{1 + \cot \theta_j'} \tag{15}$$

and

$$\phi_j = \frac{1}{1 + \cot \theta_j} \tag{16}$$

Since $R_j' = 1 - Q_j'$, and $R_j = 1 - Q_j$, then eq. (8) becomes

$$Q_j' = Q_j^{x_j} \tag{17}$$

Substituting eqs (15) and (16) into eq. (17), we obtain

$$1 + \cot \theta_j' = (1 + \cot \theta_j)^{x_j} \tag{18}$$

By eqs (12) and (13), eq (17) can be expressed in terms of ϕ_j' and ϕ_j as

$$1 + \frac{1}{\phi_j'} = (1 + \frac{1}{\phi_j})^{x_j}$$

or equivalently

$$\frac{\phi_j' + 1}{\phi_j'} = (\frac{\phi_j + 1}{\phi_j})^{x_j}$$

If both ϕ_j' and ϕ_j are much smaller than 1, then

$$\phi_j' \approx \phi_j^{x_j}$$

Substituting into eq. (12), then we reformulate the objective function as

$$\phi_s = \sum_{j=1}^{N} \phi_j^{x_j} \tag{19}$$

to be minimized subject to linear constraints

$$\begin{aligned} &\sum_{j=1}^{N} c_j x_j \leq C \\ &\sum_{j=1}^{N} w_j x_j \leq W \end{aligned} \tag{20}$$

To solve eqs. (19) and (20), the Lagrange function is introduced as

$$L = \sum_{j=1}^{N} \phi_j^{x_j} + \lambda_1 [\sum_{j=1}^{N} c_j x_j - C] + \lambda_2 [\sum_{j=1}^{N} w_j x_j - W] \tag{21}$$

The Kuhn-Tucker conditions are

$$\frac{\partial L}{\partial x_j} = \phi_j^{x_j} \ln\phi_j + \lambda_1 c_j + \lambda_2 w_j = 0 \tag{22}$$

$$\sum_{j=1}^{N} c_j x_j = C \tag{23}$$

$$\sum_{j=1}^{N} w_j x_j = W \tag{24}$$

From eq. (22).

$$x_j = \frac{1}{\ln\phi_j} [\ln(a_j \lambda_1 + b_j \lambda_2)] \qquad , \; j=1,2,\ldots,N \tag{25}$$

where

$$a_j \equiv - \frac{c_j}{\ell n \phi_j}$$

$$b_j \equiv - \frac{w_j}{\ell n \phi_j}$$

Substituting (25) into (23) and (24),

$$- \sum_{j=1}^{N} a_j [\ell n(a_j \lambda_1 + b_j \lambda_2)] = C \tag{26}$$

$$- \sum_{j=1}^{N} b_j [\ell n(a_j \lambda_1 + b_j \lambda_2)] = W \tag{27}$$

Solving the simultaneous eqs (26) and (27) to get λ_1 and λ_2. Once λ_1 and λ_2 are obtained, x_j, j=1,2,...,N, can be found from eq. (25)

<u>A Numerical Example</u>

Consider the five stage problem [18]:

Maximize

$$R_s = \sum_{j=1}^{5} [1 - (1 - R_j)^{x_j}]$$

subject to

$$g_1 = \sum_{j=1}^{5} c_j x_j \leq C$$

$$g_2 = \sum_{j=1}^{5} w_j x_j \leq W$$

The constraints associated with the problem are:

Stage j	Cost c_j	Weight w_j	Reliability R_j
1	5	8	0.90
2	4	9	0.75
3	9	6	0.65
4	7	7	0.80
5	7	8	0.85

$C = 100$, $W = 104$

The objective function is transformed by eq. (19) as

Minimize

$$\phi_s = \sum_{j=1}^{N} \phi_j^{x_j}$$

where

$$\phi_j = \frac{1 - R_j}{R_j} \qquad j = 1,2,3,4,5$$

By using the method of Lagrange multipliers and the Kuhn-Tucker conditions, we introduce multipliers λ_1 and λ_2 to obtain the solution shown in Table 1. The result is identical to that in [8].

Table 1 Numerical result of the example

Number of components at each stage x_1	x_2	x_3	x_4	x_5	Used cost g_1	Used weight g_2	R_s	λ
2	3	3	2	2	77	91	0.87529	0.005
2	3	4	3	2	93	104	0.93080	0.004
2	3	4	3	2	93	104	0.93080	0.003 *
2	3	4	3	3	100	112	0.94901	0.002

where

$\lambda = \lambda_1 + \lambda_2$

* is the optimal solution

4. Linear Programming

A linear programming problem arises whenever two or more candidates or activities are competing for limited resources and when it can be assumed that all relationships within the problem are linear.

Since the reliability optimization problem usually has nonlinear objective function and/or nonlinear constraint functions, unless we linearized the objective and/or constraint functions or we do encounter specific case, linear programming is not applicable. In the next section we are to introduce separable programming, which is a special class of nonlinear programming and is usually fit for the system optimization problems adaptable to linear programming.

A special case of reliability allocation problem solved by linear programming is presented here.

Problem statement and formulation

The problem is about reliability least cost apportionment which says:[17]

A company has a system to build which is composed of two subsystems in series. The reliability requiremnt for the system is 0.90. Initial evaluation of the two subsystems yields a reliability of 0.85 for subsystem 1 and 0.87 for subsystem 2. The product of these two subsystem reliabilities is approximately 0.74. It is clear that both subsystems' reliabilities must be improved to meet the 0.90 reliability requirement. The relative additional program cost for incremental reliability improvements is determined to be in a ratio of 0.3 to 0.7 (normalized) for subsystems 1 and 2 respectively. The reliability improvement tradeoff factor between subsystems 1 and 2 is 0.9 and 0.1 respectively, i.e., subsystem 1 approaches the constraint at a rate of 0.9 per incremental increase in reliaibility. The problem is to minimize the costs to meet the 0.90 reliability requirement.

Notation	Definition
R_o	The initial system reliability (predicted)
R_r	The design goal (the required reliability)
R_j'	The initial assessed reliability of subsystem j
$\Delta \alpha_j$	The reliaiblity improvement increment of subsystem j
α_j	The exponent corresponding to the reliability R_j', $\alpha_j < 1$ if subsystem j is part of a redundant system whose reliability is written as $R_j'^{\alpha_j}$
N	The total number of serially connected subsystems
ln(x)	The natural logarithm of x
K_j	The reliability improvement difficulty factor for subsystem j $0 \leq K_j \leq 1$ $\sum_{j=1}^{N} K_j = 1$
M_i	Number of structural variables in the ith equation
C_{ij}	The reliability improvement tradeoff factor for the ith tradeoff between a subset of $M_i \leq N$ subsystems $0 \leq C_{ij} \leq 1; \quad \sum_{j=1}^{M_i} C_{ij} = 1$
β_i	The minimal tradeoff requirement (in terms of the total reliability improvement increment) for the ith tradeoff

The methodology for pointing out areas for design improvement to meet design goals using a linear programming is as follows:

If initially

$$R_o < R_r; \quad R_o = \prod_{j=1}^{N} R_j'^{\,\alpha_j}$$

maximize the function,

$$\sum_{j=1}^{N} K_j \Delta\alpha_j$$

subject to the constraints

$$\sum_{j=1}^{N} \Delta\alpha_j \ \ln R_j' \leq \ln R_o - \ln R_r$$

$$\sum_{j=1}^{M_i} C_{ij} \Delta\alpha_j \leq \beta_i$$

(some C_{ij} may be 0)

(For simplification it is assumed in the example $\alpha_1 = \alpha_2 = 1$)

We seek to maximize

$$.3\ \Delta\alpha_1 + .7\ \Delta\alpha_2$$

Which therefore minimizes (since $\Delta\alpha_j$ are negative) additional reliability program costs subject to the constraints:

	Type of Constraints
(1) $.1625\ \Delta\alpha_1 + .1393\ \Delta\alpha_2 \leq -.1964$	Reliability Requirement constraint
(2) $.9\ \Delta\alpha_1 + .1\ \Delta\alpha_2 \leq \beta$	Tradeoff constraint
(3) $\Delta\alpha_1 > -1$	Implied constraint
(4) $\Delta\alpha_2 > -1$	Implied constraint

Solutions are generated as a function of β in Fig. 1 and 2. One can see graphically the following situations relative to the feasibility of solutions and the value of p.

For $\beta \geq -.4161$ - the tradeoff constraint does not influence the external solution to the problem. This case represents situations where there is no problem in meeting a given tradeoff constraint.

For $-.9243 \leq \beta < -.4161$ - there exists a feasible solution, the solution of which is influenced by both the minimal reliability requirement and the tradeoff constraint.

For $\beta < -.9243$ - no feasible solution exists since the constraint $\Delta\alpha_1 > -1$ imposes that the boundary is open on the left side in Fig. 1.

5. Separable Programming

Seaprable programming is a special class of nonlinear programming that is adaptable to linear programming. The problems are constructed of separable functions which have the form

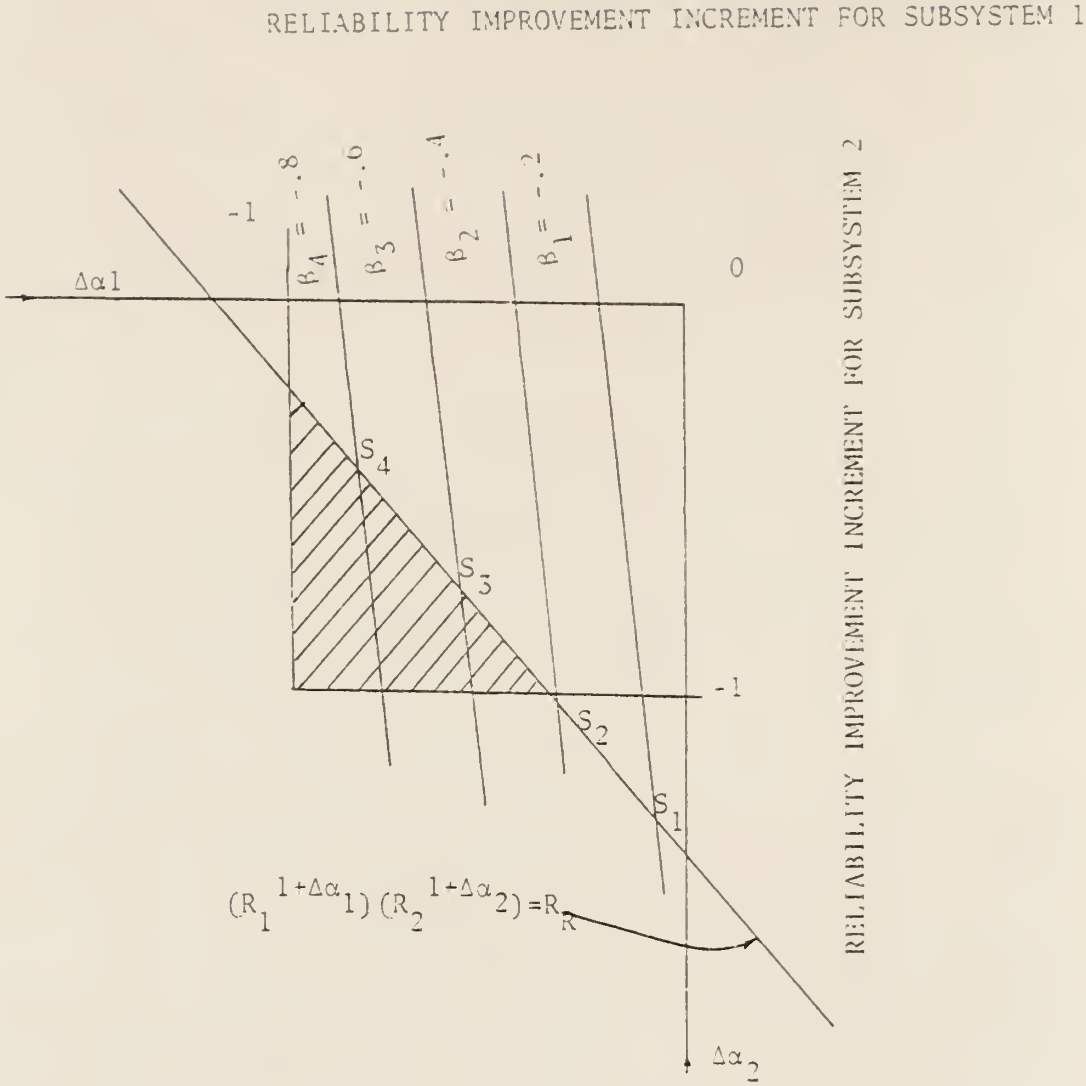
$$\phi(\bar{x}) = \sum_{i=1}^{m} h_i(x_i)$$

The separable programming problem can be defined as finding a set of x_i, i=1,2,...,m which maximizes (or minimizes)

$$c(\bar{x}) = \sum_{i=1}^{m} f_i(x_i)$$

subject to the constraints

$$\sum_{i=1}^{n} g_{ki}(x_i) \leq b_k, \qquad k=1,\ldots,p$$



CONSTRAINTS:

$.9\Delta\alpha_1 + .1\Delta\alpha_2 \leq \beta$

$-(\Delta\alpha_1 \ln R_1 + \Delta\alpha_2 \ln R_2) \leqq \ln R_o - \ln R_R$

or equivalently

$(R_1^{1+\Delta\alpha_1})(R_2^{1+\Delta\alpha_2}) \geqq R_R$

$\Delta\alpha_1 > -1$ IMPLIED

$\Delta\alpha_2 > -1$ IMPLIED

	$\Delta\alpha_1$	$\Delta\alpha_2$	$\beta=$
S_1 =	(-.0755,	-1.32)	-.2
S_2 =	(-.331,	-1.02)	-.4
S_3 =	(-.586,	-.73)	-.6
S_4 =	(-.842,	-.43)	-.8

Fig. 1

OPTIMAL VALUES $R_1^{(1 + \Delta\alpha_1)}$ $R_2^{(1 + \Delta\alpha_2)}$ AS A FUNCTION OF BETA

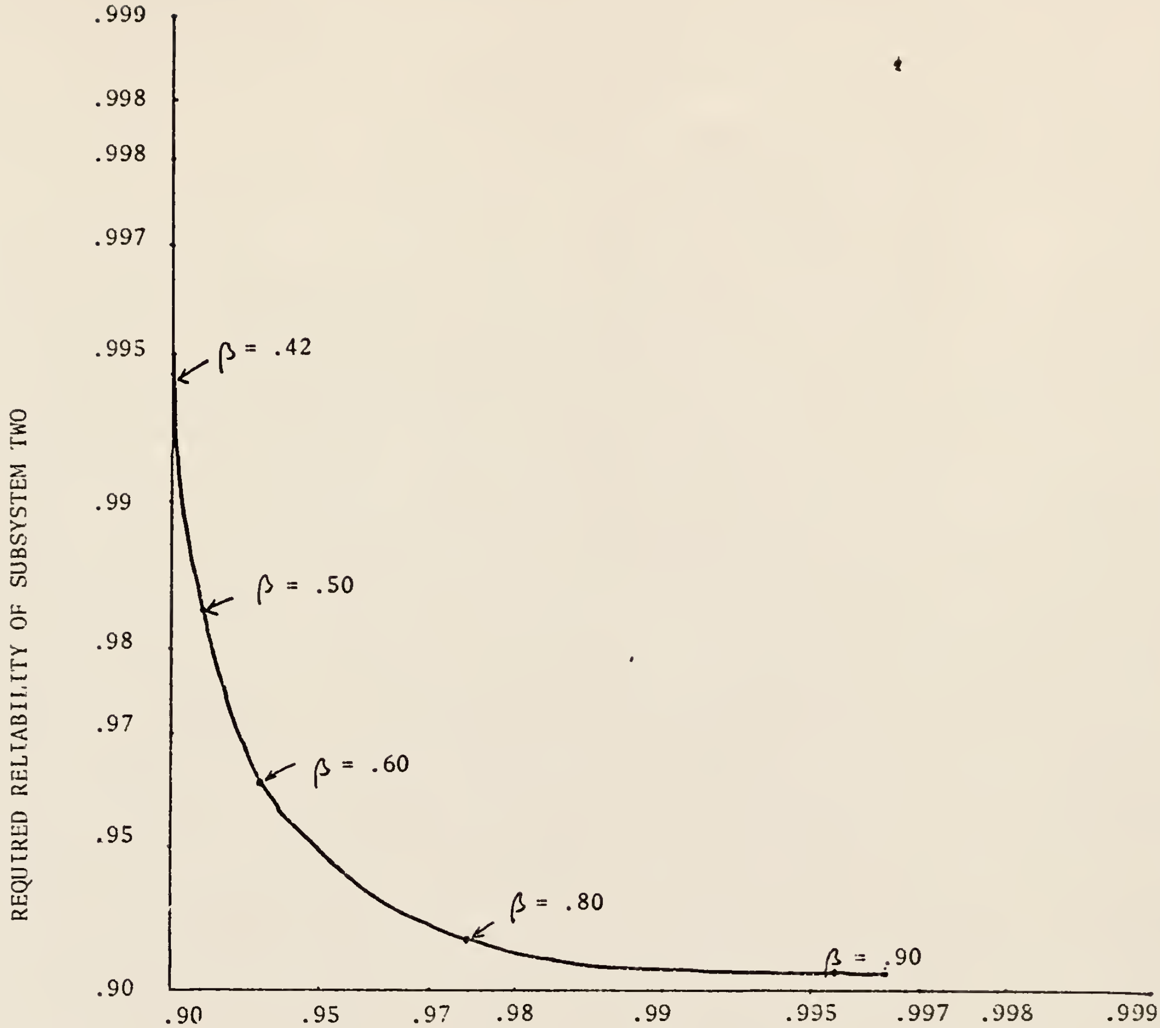


Fig. 2 REQUIRED RELIABILITY OF SUBSYSTEM ONE

and

$$x_i \geq 0$$

By approximating a nonlinear function of one variable by a piecewise linear function, the problem becomes a restricted linear programmming problem, and can be solved by a slightly revised simplex method. MPS/360 has this revision[23].

Formulation of the Problem

A continuous nonlinear function of a single variable, x_i, can be approximated by a piecewise linear function over a specified interval domain. This is done by partitioning this intercal domain into n_i disjoint, but continuous, intercals. The $(n_i + 1)$ points of the partitions are represented by the set

$$S = \left\{ x_i^0, x_i^1, x_i^2, \ldots, x_i^{n_i} \right\}$$

There are two methods of representing the piecewise linear spproximation of a continuous nonlinear function of one variable. The method employed here is known as the "delta method." Both methods are developed in G. Hadley's Nonlinear and Dynamic Programming [22]. The "delta method" uses the differences of adjacent points of the set, S, and the differences of the functional values at the adjacent points in developing the approximating equation of a function, $f_i(x_i)$. The differences are represented by

$$\Delta x_i^j = x_i^j - x_i^{j-1}, \qquad i = 1,2,\ldots,m$$

$$\Delta f_i^j = f_i(x_i^k) - f_i(x_i^{j-1}), \qquad j = 1,2,\ldots,n_i \tag{28}$$

where the subscript refers to a function and/or variable such as x_i, $f_i(x_i)$, and $g_{ki}(x_i)$, and the superscript refers to a partitioning of a variable. That is, $f_i(x_i^j)$ is the value of $f_i(x_i)$ at $x_i = x_i^j$. The differences for adjacent points and the corresponding functional values for a function with $n_i = 4$ are shown in Fig. 3.

To represent the variable x_i and the approximation of $f_i(x_i)$, a set of variables, n_i^j, $j = 1,2, \ldots, n_i$, is created that follows what is known as the "restricted-basis-entry-rule." The rule is satisfied for any one of the following conditions.

(i) $0 \leq D_i^1 \leq 1$ iff $n_i^j = 0, \quad j = 2, 3, \ldots, n_i$

(ii) $0 \leq D_i^j \leq 1$ iff $D_i^\ell = 1, \quad \ell = 1, 2, \ldots, j-1$

and

$D_i^k = 0, \quad k = j + 1, \ldots, n_i$

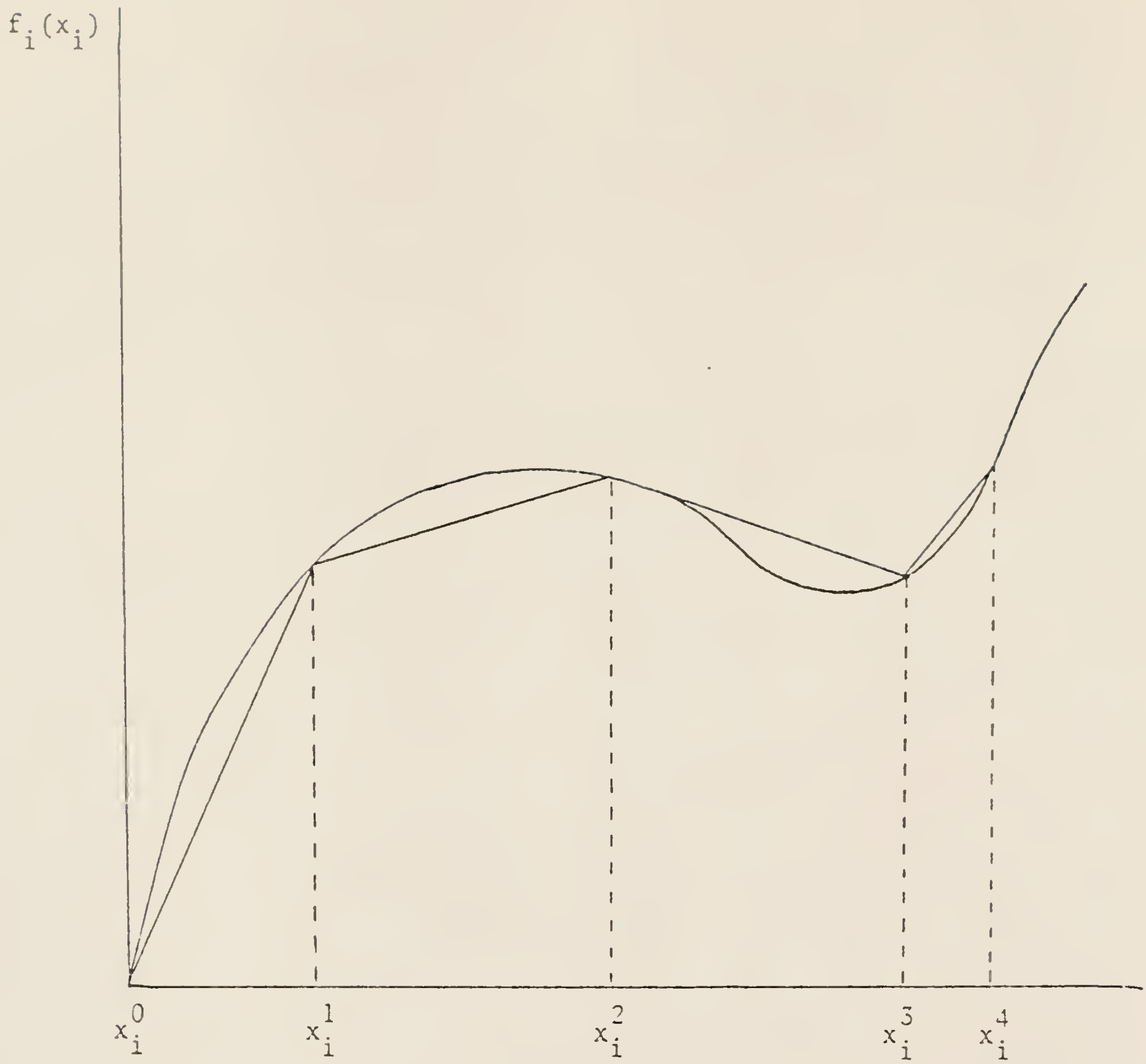
(iii) $0 \leq D_i^{n_i}$ iff $D_i^j = 1, \quad j = 1, 2, \ldots, n_i-1$

where n_i is the number of partitioning intervals for a variable x_i. D_i^j represents a variable created for the <u>jth</u> partition of variable x_i. Intuitively, for any $0 \leq D_i^j \leq 1$ all previous D_i^ℓ variables ($\ell = 1, \ldots, j-1$) must have a value of one and all following values ($= j+1, \ldots, n_i$) must be zero.

<u>A Numerical Example</u> [21]

Maximize

$$R_s = \prod_{j=1}^{5} [1 - (1 - R_j)^{x_j}]$$



$$\Delta x_i^1 = x_i^1 - x_i^0 \qquad \Delta f_i^1 = f_i(x_i^1) - f_i(x_i^0)$$

$$\Delta x_i^2 = x_i^2 - x_i^1 \qquad \Delta f_i^2 = f_i(x_i^2) - f_i(x_i^1)$$

$$\Delta x_i^3 = x_i^3 - x_i^2 \qquad \Delta f_i^3 = f_i(x_i^3) - f_i(x_i^2)$$

$$\Delta x_i^4 = x_i^4 - x_i^3 \qquad \Delta f_i^4 = f_i(x_i^4) - f_i(x_i^3)$$

Fig. 3. Linear Approximation of $f_i(x_i)$.

subject to

$$g_1 = \sum_{j=1}^{5} p_j(x_j)^2 \leq P$$

$$g_2 = \sum_{j=1}^{5} c_j(x_j + \exp(x_j/4)) \leq C$$

$$g_3 = \sum_{j=1}^{5} w_j x_j \exp(x_j/4) \leq W$$

where $x_j \geq 1$, $j=1,2,\ldots,5$. are integers.

The constraints associated with the five stage problem are

j	R_j	p_j	P	c_j	C	w_j	W
1	0.80	1		7		7	
2	0.85	2		7		8	
3	0.90	3	110	5	175	8	200
4	0.65	4		9		6	
5	0.75	2		4		9	

It is noted that, in optimizing the system reliability, the decision variables namely, the number of components used at each stage, are considered as continuous variables. The nearest integer numbers are assigned to them eventually.

The objective function was transformed to maximize

$$S = \ell n\, R_s = \sum_{j=1}^{5} \ell n[1 - (1 - R_j)^{x_j}],$$

then the MPS/360 [23] was applied to solve the problem.

The procedure is recommended in MPS/360 to determine the existance of a local optimum solution, if it exists. Separable programming, at its best, will guarantee only a local optimum. One reason is that unlike linear inequality constraints nonlinear inequality constraints do not necessarily form a convex set. A second reason is that a nonlinear function is not necessarily concave or convex. The only way to guarantee a stationary point in a global maximum is for a function to be concave, or if it is a gloabal minimum the function must be convex. Sine the linear approximation function of a separable nonlinear function will reflect its particular concave and convex properties, separable programming will, at its best, produce a local optimum solution.

The solution to the problem is

$x_1 = 2.70000$

$x_2 = 2.32929$

$x_3 = 2.10000$

$x_4 = 3.50000$

$x_5 = 2.80000$

Following the similar rounding off procedures discussed in Example 3 of GRG section, the configuration of (3, 2, 2, 3, 3) will give the optimal solution with system reliability, R_s, 0.9045 and consumes $g_1 = 83$, $g_2 = 146.1$ and $g_3 = 194.5$. It is noted that separable programming is an approximate method depending on the fineness of the grid equations for accuracy. The uniform grid for this solution is only 0.10. The effects of grid size on problem accuracy is dependent on the properties of the approximated functions.

REFERENCES

1. Albert, A., "A Measure of the Effort Required to Increase Reliability," Technical Report No. 43, November, 1958, Applied Mathematics and Statistics Laboratory, Stanford University, Contract No. N6onr-25140 (NR 342-022).

2. Banerjee, S.K., and K. Rajamani, "Parametric representation of probability in two dimensions - a new approach in system reliability evaluation", IEEE Transactions on Reliability, Vol. R-21, pp. 56-60 (February 1972).

3. Banerjee, S.K., K. Rajamani, and S.S. Deshpande, "Optimal redundancy allocation for non series-parallel networks", IEEE Transactions on Reliability, Vol. R-25, No. 2, pp. 115-117 (June 1976).

4. Banerjee, S.K., and K. Rajamani, "Optimization of system reliability using a parametric approach", IEEE Transactions on Reliability, Vol. R-22, pp. 35-39 (April 1973).

5. Beraha, D. and K.B. Misra, "Reliablity optimization through random search algorithm", Microelectronics and Reliability, Vol. 13, pp. 295-297 (August 1974)

6. Bodin, L.D., "Optimization procedure for the analysis of coherent structures", IEEE Transactions on Relaibility, Vol. R-18, No. 3, pp. 118-126 (August 1969).

7. Gordon, R., "Optimum component redundancy for maximum system reliability", Operations Research, Vol. 5, pp. 229-243 (April 1957).

8. Hees, Ir.R.N.v., and Ir.H.W.v.d. Meerendonk, "Optimal reliability of parallel multi-component systems", Operations Research Quarterly, Vol. 12, No. 1, pp. 16-26 (May 1961).

9. Kolesar, P.J., "Linear programming and the reliability of multicomponent systems", Naval Research Logistics Quarterly, Vol. 15, pp. 317-327 (Sept. 1967).

10. Lientz, B.P., "A stochastic method of allocation of components to maximize reliability," IEEE Transactions on Reliability, Vol. R-23, No. 2, (June 1974).

11. Lloyd, D.K., and M. Lipow, Reliability: Management, Methods, and Mathematics, Englewood Cliffes, N.J.: Prentice-Hall (1962).

12. Morrison, D.F., "The optimum allocation of spare components in system", Technometrics, Vol. 3, No. 3, pp. 399-406 (August 1961).

13. Moskowitz, F., and J.B. McLean, "Some reliability aspects of system design", IRE Transactions on Reliability and Quality Control, Vol. PGROC-8, pp. 7-35 (September 1956).

14. Neuner, G.E., and R.N. Miller, "Resource allocation for maximum reliability", Proceedings of 1966 Annual Symposium on Reliability, pp. 332-346 (January 1966).

15. Sasaki, M., "A simplified mthod of obtaining highest system reliaiblity", Proceedings of the Eighth National Symposium on Reliability and Quality Control, pp. 489-502 (1962).

16. Sasaki, M., "An easy allotment method achieving maximum system reliability", Proceedings of the Ninth National Symposium on Reliability and Qualtiy Control pp. 109-124 (1963).

17. Selman, V., and N.T. Grisamore, "Optimum system analysis by linear programming", 1966 Processing Annual Symposium on Reliability, American Society for Quality Control, pp. 696-705 (1966)

18. Tillman, F.A., and J.M. Liittschwager, "Integer programming formulation of constrained reliability problems", Management Science, Vol. 13, No. 11, pp. 887-899 (July 1967).

19. Webster, L.R., "Optimum system reliability and cost effectiveness", Proceedings of 1967 Annual Symposium on Reliability, pp. 489-500 (January 1967).

20. Webster, L.R., "Choosing optimum system configurations", Proceedings of the Tenth National Symposium on Reliability & Quality Control, Washington, D.C., pp. 345-359 (1964).

21. Williams, J., "Optimization of industrial systems with the separable programming and the generalized reduced gradient methods", Master Thesis, Kansas State University, Manhattan, Kansas, (1972).

22. Hadley, G., Nonlinear and Dynamic Programming, Reading, Mass: Addison-Wesley (1964).

23. Mathematical Programming System/360, Version 2, Linear and Separable Programming User's Manual, GH20-0476-2, International Business Machines Corp., 1968.

24. Kapur, K. C., and L. R. Lamberson, Reliability in Engineering Design, N.Y.: Wiley (1977).

25. Smith, C. O., Introduction to Reliability in Design, N.Y.: McGraw-Hill (1976).

CHAPTER 4 DETERMINATION OF COMPONENT RELIABILITY AND REDUNDANCY FOR OPTIMUM SYSTEMS RELIABILITY

1. INTRODUCTION

In the design process, a system must not only be designed to meet its functional requirement but must also be designed to perform its function successfully. This latter requirement involves designing reliability into the system. Often this involves designing to meet the reliability requirements within the framework of several system constraints. In some optimum system reliability problems the element reliability is assumed to be fixed, and the optimal number of redundancies at each stage is determined where the system is subject to constraints. A number of optimization techniques have been successfully applied to solve this class of problems [7]. However, a more general problem is one where both the optimal component reliability and the optimum number of redundancies are to be determined in order to obtain the best overall systems reliability [3]. Specifically the problem is one where the designer must not only determine the number of redundancies but also the reliability of each component. This is a mixed integer non-linear programming problem.

In general, problems of this type are difficult to solve by the normal system optimization techniques, for example, by the method of Lagrange multipliers [3], sequential unconstrained minimization technique (SUMT) or generalized reduced gradient technique [7] because these techniques do not provide integer solutions. The available integer programming techniques do not guarantee an optimal solution. Hence a technique that provides an integer solution as well as the optimal level of component reliability is required. The suggested procedure is one such technique.

A series system with active component redundancy is considered in this study. A combination of the well-known Hooke and Jeeves pattern search [2]

Notation

b_i = the available resource for the i^{th} constraint

C = the available cost limitation in dollars

$C_j(R_j)$ = the cost of one element at the j^{th} stage as a function of R_j

g_{ij} = the amount of the i^{th} resource consumed at the j^{th} stage

p_j = the product of the weight per element and the volume per element at the j^{th} stage

P = the limitation of the product of the volume times the weight constraints

N = the total number of stages in the system of interest

R_j^o = the initial component reliability at the j^{th} stage

R_j, Q_j = the reliability and unreliability of one element at the j^{th} stage, respectively

R_s, Q_s = the system reliability and unreliability, respectively

r = the total number of constraints

v_j = the volume of one component at the j^{th} stage

w_i = the weight of one component at the j^{th} stage

W = the limitation on weight

X_j = the number of components used at stage j

X_j^o = the initial number of elements used at the j^{th} stage

$\bar{X}^*(\bar{R})$ = a vector of optimal number of elements at each stage as a function of the component reliability at each stage

λ_j = the components failure rate at the j^{th} stage

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

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

and the suggested heuristic approach by Aggarwal, et al. [1] is proposed as a stepwise optimization technique for solving this problem. The procedure is simple and efficient; a component reliability is assumed and the optimal number of redundancies is determined by the heuristic technique. A sequential search routine for maximizing the overall system reliability is carried out by using the Hooke and Jeeves pattern search.

2. STATEMENT OF THE PROBLEM

The system reliability of an N-stage parallel-series system, where both the component reliability, R_j, and the number of components, X_j, at the jth stage are to be determined, is expressed by

$$R_s(\bar{R},\bar{X}) = \prod_{j=1}^{N} [1-(1-R_j)^{X_j}] \tag{1}$$

subject to

$$\sum_{j=1}^{N} g_{ij}(R_j,X_j) \leq b_i, \qquad i = 1,2, \ldots, r \tag{2}$$

where the system reliability, $R_s = R_s(R_1,R_2, \ldots, R_N; X_1,X_2, \ldots, X_N)$, R_j, $j = 1,2, \ldots, N$, are all real numbers between 0 and 1, and X_j, $j = 1,2, \ldots, N$, are all positive integers.

To set up equations (1) and (2), five assumptions are made, they are: (1) Each stage is in series and is considered to be essential for the overall operational success of the mission of the system. (The system is denoted as a 1-out-of-N: F configuration). (2) All the stages as well as all the parallel elements used at each stage are s-independent. All components in parallel in the same stage have the same probability of failure.

(3) All the components at each stage are simultaneously working, and for a stage to fail all the elements in that stage must fail (Each stage is denoted as a 1-out-of-X_j: G configuration). (4) A short circuit failure will not be considered, that is, only a single mode of failure is assumed. (5) The costs are additive between stages.

Both the number of redundancies and the component reliability improvement will incur a "cost", which may be stated in dollars, weight, volume or a combination of all three. In order to be specific, three such constraints are assumed. These constraints have been used often to test and demonstrate optimization techniques. (4,5,6)

The first constraint is a combination of weight and volume and is stated as follows:

$$\sum_{j=1}^{N} g_{1j}(X_j) = \sum_{j=1}^{N} w_j v_j (X_j)^2 = \sum_{j=1}^{N} P_j (X_j)^2 \leq P \qquad (3)$$

It is noted that the component reliability does not usually affect the weight nor the volume, hence g_{1j} is not a function of R_j.
The second constraint is expressed in dollars, and is a function of X_j and R_j. It is stated as:

$$\sum_{j=1}^{N} g_{2j}(X_j, R_j) = \sum_{j=1}^{N} C_j(R_j)(X_j + \exp(X_j/4)) \leq C \qquad (4)$$

where $C_j(R_j)$ is the cost per component at the jth stage. The cost is an increasing function of R_j or conversely a decreasing function of the component failure rate expressed by

$$C_j(\lambda_j) = \alpha_j \left\{\frac{1}{\lambda_j}\right\}^{\beta_j}$$

where α_j and β_j are constants representing the inherent characteristics of each component at the jth stage, $\beta_j > 1$. If each component follows the negative exponential failure law, i.e.,

$$R_j = e^{-\lambda_j t}$$

for all j, then the component cost at the jth stage is

$$C_j(R_j) = \alpha_j \left(\frac{-t}{\ell n R_j}\right)^{\beta_j} \tag{5}$$

where t is the operating time during which the component at stage j will not fail. Usually α_j and β_j and t are given.

Thus, $C_j(R_j) \cdot X_j$ is the cost of the components at the jth stage as a function of R_j and X_j. An additional cost $C_j(R_j)\exp(X_j/4)$ is included, as the cost for interconnecting parallel elements.

Substitute (5) into (4), one obtains a dollar constraint as

$$\sum_{j=1}^{N} \alpha_j \left(\frac{-t}{\ell n R_j}\right)^{\beta_j} (X_j + \exp(X_j/4)) \leq C, \tag{6}$$

Similarly a weight constraint is stated as

$$\sum_{j=1}^{N} g_{3j}(X_j) = \sum_{j=1}^{N} w_j X_j \quad \exp(X_j/4) \leq W \tag{7}$$

where w_jX_j is the weight of all of the components at the jth stage. Again an additional factor is multiplied, which is exp (X_j/r), due to the hardware for interconnecting the links. Also note that the weight constraint is not a function of the component reliability.

Now, the problem can be stated as one where the $R_1, R_2, \ldots, R_N$; $X_1, X_2, \ldots, X_N$ are selected so that equation (1) will be maximized subject to (3), (6) and (7), where $R_1, R_2, \ldots, R_N$ are real numbers between 0 and 1; and $X_1, X_2, \ldots, X_N$ are positive integers.

3. AN OPTIMIZATION PROCEDURE

The combination of the Hooke and Jeeves pattern search [2] and the heuristic approach of Aggarwal, et al. [1] is employed for solving the previously stated mixed integer nonlinear programming problem. The descriptive flow diagram is shown in Fig. 1.

The Hooke and Jeeves pattern search technique is a sequential search routine for maximizing the function, $R_s(\bar{R},\bar{X})$. The argument in the Hooke and Jeeves pattern search is the component reliability, $\bar{R}$, which is varied until the maximum of $R_s(\bar{R},\bar{X})$ is obtained. The heuristic approach is applied to each value of $\bar{R}$ to obtain the optimal number of redundancies, $X_1, X_2, \ldots, X_N$, which maximizes $R_s(\bar{R},\bar{X})$ while satisfying the nonlinear constraints. This heuristic approach is based on the concept that a component is added to the stage where its addition produces the greatest ratio of "increment increases in reliability" to the "product of decrements in slacks". This ratio is defined by

$$F_j(X_j) = \frac{\Delta(1-R_j)^{X_j}}{\prod_{i=1}^{3} \Delta g_{ij}(X_j)} \tag{8}$$

where

$$\Delta(1-R_j)^{X_j} = (1-R_j)^{X_j} - (1-R_j)^{X_j+1} = R_j(1-R_j)^{X_j}$$

and

$$\Delta g_{ij}(X_j) = g_{ij}(X_j+1) - g_{ij}(X_j)$$

The computational procedures for evaluating the functional value of the system reliability, $R_s(\bar{R},\bar{X})$, at any point is:

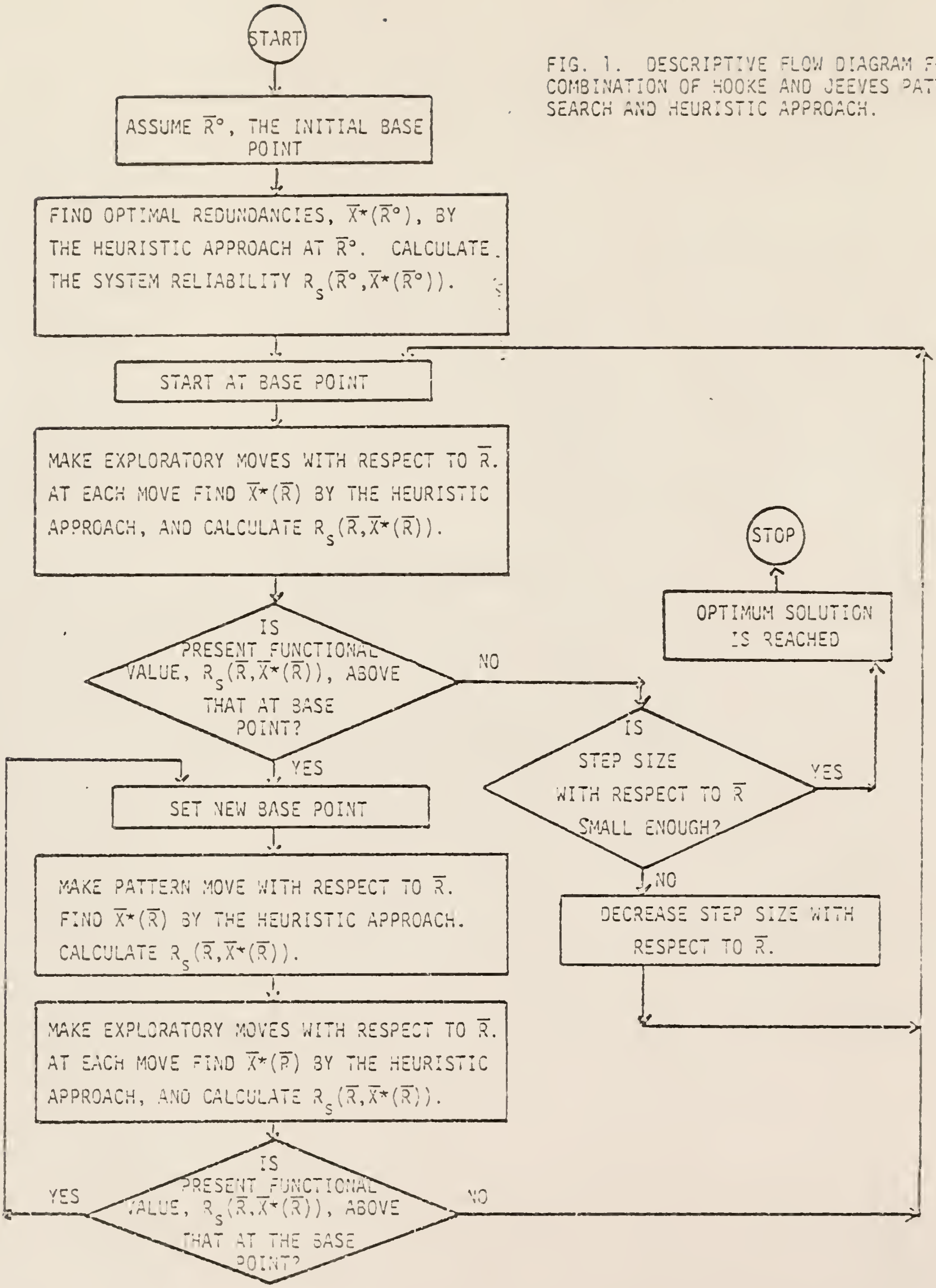


FIG. 1. DESCRIPTIVE FLOW DIAGRAM FOR COMBINATION OF HOOKE AND JEEVES PATTERN SEARCH AND HEURISTIC APPROACH.

1) For an initial starting point the component reliability, $\bar{R} = (R_1, R_2, \ldots, R_N)$, is given.

2) (a) Substitute the value $(R_1, R_2, \ldots, R_N)$ into (1) and (6), then the problem is to find $(X_1, X_2, \ldots, X_N)$, a straightforward redundancy problem where the heuristic approach can be applied.

(b) Let $\bar{X} = (X_1, X_2, \ldots, X_N) = (1, 1, \ldots, 1)$.

3) (a) Calculate $F_j(X_j)$ for all j using (8).

(b) Select the stage having the highest $F_j(X_j)$. A redundant component is proposed to be added to that stage.

4) Check to see if the constraints are violated.

(a) If the solution is still feasible, add one redundant component. Modify the value of X_j and repeat step 3.

(b) If at least one constraint is exactly satisifed; the current value of $\bar{X}$ is an optimal solution corresponding to $(R_1, R_2, \ldots, R_N)$. Go to step 5.

(c) If at least one constraint is violated, cancel the proposed addition of the redundant component; remove that stage from further consideration and repeat step 3. When all the stages are excluded from further consideration, the current values of $\bar{X}$ are the optimal solution with respect to $\bar{R} = (R_1, R_2, \ldots, R_N)$.

5. Calculate the system reliability, R_s, the functional value, for the assigned $\bar{R}$ and the optimum $\bar{X}^*$.

4. NUMERICAL EXAMPLES

Example 1. A five stage problem was solved with the values given in Table 1. The optimal solution is presented in Table 2. The optimum system reliability is 0.91494 at the point $(R_1,R_2,R_3,R_4,R_5; X_1,X_2,X_3,X_4,X_5)$ = (0.7582, 0.8000, 0.9000, 0.8000, 0.7500; 3,3,2,2,3). Using the starting values of (R_1,R_2,R_3,R_4,R_5) = (0.70,0.70,0.70,0.70,0.70), the computation took 23 sec. to reach the optimum solution on an IBM 370/158 computer.

Example 2. A similar five stage problem as Example 1 was solved, but where the limitations on the constraints were P = 220, C = 350, W = 400. The optimal solution obtained from using the following two sets of starting values of $\bar{R}^o$ = (0.7,0.7,0.7,0.7,0.7) and $\bar{R}^o$ = (0.8, 0.8, 0.8, 0.8, 0.8) are presented in Tables 3 and 4. The optimal system reliabilities for these two set of solutions are 0.993657 and 0.994767. The difference is about 0.11%. However, the optimum component reliabilities and redundancies are (0.900, 0.850, 0.856, 0.750, 0.850; 3,4,4,4,4) and (0.850, 0.863, 0.902, 0.700, 0.900; 4,4,3,3,3), respectively. It seems that the functional value of the systems reliability, at the optimum is quite flat, therefore, there is a flexibility to select various values of component reliabilities and redundancies which have nearly the same optimal system reliability.

5. CONCLUDING REMARKS

The determination of the optimal number of redundancies as well as the optimal component reliability level in each of stages are carried out by a combination of the well-known Hooke and Jeeves pattern search technique and a heuristic approach. The optimal system reliability problem is an

Table 1. Constants used in Example 1.

j	α_j	p_j	w_j	P	C	W
1	2.33×10^{-5}	1	7			
2	1.45×10^{-5}	2	8	110	175	200
3	5.41×10^{-6}	3	8			
4	8.05×10^{-5}	4	6			
5	1.95×10^{-5}	2	9			

$\beta_j = 1.5, \quad j = 1,2,3,4,5$

$t = 1000$

Table 2. Optimal solution for Example 1.

	R_1	R_2	R_3	R_4	R_5
Starting point	0.70	0.70	0.70	0.70	0.70

Initial step size 0.05

Final step size 0.00039

	R_1	R_2	R_3	R_4	R_5	X_1	X_2	X_3	X_4	X_5
Optimal point	0.7582	0.8000	0.9000	0.8000	0.7500	3	3	2	2	3

Optimal system reliability 0.91494

Slack for the first constraint = 28

Slack for the second constraint = 0.033727

Slack for the third constraint = 1.4118

Table 3. Optimal solution for Example 2

	R_1	R_2	R_3	R_4	R_5	X_1	X_3	X_3	X_4	X_5
Starting point	0.7	0.7	0.7	0.7	0.7					
Optimal point	0.900	0.850	0.856	0.750	0.850	3	4	4	4	4

Optimal system reliability, R_5 = 0.993657

Slack for the first constraint = 35,

Slack for the second constraint = 0.033247

Slack for the third constraint = 18.476

Initial step size = 0.05

Final step size = 0.0002

Table 4. An alternate optimal solution for Example 2

	R_1	R_2	R_3	R_4	R_5	X_1	X_2	X_3	X_4	X_5
Starting point	0.8	0.8	0.8	0.8	0.8					
Optimal point	0.850	0.863	0.902	0.700	0.900	4	4	3	5	3

Optimal system reliability, R_s = 0.994767

Slack for the first constraint = 27

Slack for the second constraint = 0.006542

Slack for the third constraint = 24.226

Initial step size = 0.05

Final step size = 0.0002

extension of the usual reliability optimization problem and is a mixed integer nonlinear programming problem. The heuristic approach insures the integer number of redundancies with nonlinear constraints, while the Hooke and Jeeves pattern search optimizes the component reliability level. This procedure seems to be very efficient in solving this problem.

ACKNOWLEDGEMENT

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

REFERENCES

1. Aggarwal, K.K., J.S. Gupta, and K.B. Misra, "A New heuristic criterion for solving a redundancy optimization problem", IEEE Transactions on Reliability, Vol. R-24, No. 1, pp. 86-87 (April 1975).
2. Hooke, R., and T.A. Jeeves, "Direct search solutions of numerical and Statistical problems", J. Assoc. Compt. Mach., Vol. 8, pp. 212-224 (1961).
3. Misra, K.B., and M.D. Ljubojevic, "Optimal reliability design of a system: a new look", IEEE Transactions on Reliability, Vol. R-22, pp. 255-258, (Dec. 19
4. Sharma, J., and K.V. Venkateswaran, "A direct method for maximizing the system reliability", IEEE Transactions on Reliability, Vol. R-20, No. 4, pp. 256-259 (November 1971).
5. Tillman, F.A., and J.M. Littschwager, "Integer programming formulation of constrained reliability problems", Management Science, Vol. 13, No.11, pp. 887-899 (July 1967).
6. 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. R-17, No. 3, pp. 153-157 (September 1968).
7. Tillman, F.A., C.L. Hwang, and W. Kuo, "Optimization techniques for systems reliability with redundancy - a review", IEEE Transactions on Reliability, (this same issue, 1977).

APPENDIX

OUTLINE OF SEVERAL OPTIMIZATION TECHNIQUES

A1. DYNAMIC PROGRAMMING

Dynamic Programming provides a powerful tool for solving multistage decision processes which arise in various fields. It is based on the so-called "principle of optimality" and employs the techniques of invariant imbedding. The essential notions of dynamic programming are linked to a serial structure. As mentioned, its cornerstone is the principle of optimality founded Bellman (1957). It states, "An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

Consider a multistage process for which x_n denotes a state vector which represents a set of variables from stage n, and θ^n is a decision (or control) vector which stands for a set of decision (or control) variables at stage n.

The notion of stage is actually an abstract one and the function of each stage is to transform the state variables from the input state to the output state. This transformation can generally be expressed as

$$x_n = T_n(x_{n+1};\ \theta_n), \qquad n = N, N-1, \ldots, 2, 1. \tag{1}$$

Equation (1) is of vector form. If there are s state variables and one decision variable, equation (1) can be written as

$$x_{i,n} = T_{i,n}(x_{1,\ n+1},\ x_{2,\ n+1},\ \ldots,\ x_{s,n+1},\ \theta_n). \tag{2}$$

The objective of optimization of a multistage process is to seek a set of admissible values of θ_1, θ_2, ..., θ_N so that a desired performance criterion or a return function which is the objective function is maximized (or minimized). The characteristic feature of a multistage decision process is that there is an interval profit or return associated with each stage of the process. The objective function can be expressed as the summation of the interval profits,

$$S(x_{N+1};\ \theta_N,\ \ldots,\ \theta_2,\ \theta_1) = \sum_{n=N}^{1} g_n(x_{n+1};\ \theta_n). \tag{3}$$

The value of the objective function depends on the initial state and a sequence of the decisions, $\theta_N, \ldots, \theta_2, \theta_1$. If we represent the maximum return function or the maximum objective function by $f_N(x_{N+1})$, then

$$\begin{aligned} f_N(x_{N+1}) &= f_N(x_{1,N+1},\ x_{2,N+1},\ \ldots,\ x_{s,N+1}) \\ &= \max S(x_{N+1};\ \theta_N,\ \ldots,\ \theta_1) \\ &= \max_{\{\theta^n\}} \sum_{n=N}^{1} g_n(x_{n+1};\ \theta_n) \end{aligned} \tag{4}$$

Thus, in general, $f_n(x_{n+1})$ is the maximum return obtainable from the operation of an n-stage process if an optimal policy is followed starting with the initial state, x_{n+1}.

If there is one decision variable in each stage, equation (4) expresses an N-dimensional optimization problem because this problem must be optimized with respect to all the N decision variables. The dynamic programming technique is to deal with this problem as N one-dimensional problems. For a one-stage process, equation (4) becomes

$$f_1(x_2) = \max_{\theta_1} \{g_1(x_2;\ \theta_1)\}$$

which is the simplest optimization problem among the sequence of problems for $n = 1, 2, \ldots, N$. The other members of this sequence can be obtained by writting equation (4) in the form,

$$f_1(x_{n+1}) = \max_{\theta_n} \max_{\theta_{n-1}} \ldots \max_{\theta_1} \{g_n(x_{n+1};\theta_n) + \ldots + g_1(x_2;\theta_1)\}.$$

Since the inputs to stages following stage n are all affected by θ_n and the state of stage n is not affected by decisions made at stages following it, we can rewrite this as

$$f_n(x_{n+1}) = \max_{\theta_n} \{g_n(x_{n+1};\theta_n) + \max_{\theta_{n-1}} \ldots \max_{\theta_1} \left(g_{n-1}(x_n;\theta_{n-1}) + \ldots + g_1(x_2;\theta_1)\right)\}. \tag{6}$$

The expression,

$$\max_{\theta_{n-1}} \ldots \max_{\theta_1} \left(g_{n-1}(x_n;\theta_{n-1}) + \ldots + g_1(x_2;\theta_1)\right),$$

stands for the maximum return (the objective function) from an (n-1)-stage process with initial state x_n. Hence, we can also write

$$f_{n-1}(x_n) = \max_{\theta_{n-1}} \ldots \max_{\theta_1} \left(g_{n-1}(x_n;\theta_{n-1}) + \ldots + g_1(x_2;\theta_1)\right). \tag{7}$$

Thus, equation (6) can be simplified to

$$f_n(x_{n+1}) = \max_{\theta_n} \left(g_n(x_{n+1};\theta_n) + f_{n-1}(x_n)\right)$$

or (8)

$$f_n(x_{n+1}) = \max_{\theta_n} \left(g_n(x_{n+1};\theta_n) + f_{n-1}(T(x_{n+1};\theta_n))\right).$$

This is the so-called functional equation and, in essence, a mathematical statement of the principle of optimality. It gives a recursive relationship between an n stage process and an n-1 stage process. The solution of the functional equation yields the value of the maximum return and the corresponding optimal policy which belongs to the set $[0_n]$.

Further detials concerning dynamic programming as an optimization tool are available in the texts by Bellman (1957) and Bellman and Dreyfus (1962).

REFERENCES

Bellman, R., Dynamic Programming, Princeton University Press, Princeton (1957).

Bellman, R., and S. E. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton (1962).

A2. THE DISCRETE MAXIMUM PRINCIPLE

Consider a simple multistage process consisting of N stages connected in series. The state of the process stream denoted by an s-dimensional vector, $x = (x_1, x_2, \ldots, x_s)$, is transformed at each stage according to an r-dimensional decision vector, $\theta = (\theta_1, \theta_2, \ldots, \theta_r)$, which represents the decisions made at that stage. The transformation of the process stream at the nth stage is described by a set of performance equation in vector form.

$$x^n = T^n(x^{n-1}; \theta^n), \qquad (n = 1, 2, \ldots, N) \tag{1}$$

$$x^0 = \alpha$$

A typical optimization problem associated with such a process is to find a sequence of θ^n, $n = 1, 2, \ldots, N$, subject to the constraints

$$\psi_i^n \, [\theta_1^n, \theta_2^n, \ldots, \theta_r^n] \leq 0 \tag{2}$$

$$(n = 1, 2, \ldots, N; \quad i = 1, 2, \ldots, r)$$

which makes a function of the state variable of the final stage

$$S = \sum_{i=1}^{s} c_i x_i^N, \qquad (c_i = \text{constant}) \tag{3}$$

an extremum when the initial condition $x^0 = \alpha$ is given. The function, S, which is to be maximized (or minimized), is the objective function of the process.

The procedure for solving such an optimization problem by the discrete maximum principle is to introduce an s-dimensional adjoint vector z^n and a Hamiltonian function H^n, which satisfy the following relations:

$$H^n = (z^n)^T x^n = \sum_{i=1}^{s} z_i^n T_i^n (x^{n-1}; \theta^n), \tag{4}$$

$$(n = 1, 2, \ldots, N)$$

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}} \qquad (i = 1, 2, \ldots, s;\ n = 1, 2, \ldots, N) \tag{5}$$

and

$$z_i^N = c_i, \qquad (i = 1, 2, \ldots, s) \tag{6}$$

If the optimal decision vector function $\bar{\theta}^n$, which makes the objective function S an extremum, is interior to the set of admissible decisions θ^n, the set given by equation (2), a necessary condition for S to be (local) extremum with respect to θ^n is

$$\frac{\partial H^n}{\partial \theta^n} = 0, \qquad (n = 1, 2, \ldots, N) \tag{7}$$

If $\bar{\theta}^n$ is at a boundary of the set, it can be determined from the condition that H^n is (locally) extremum.

Further detials on the discrete maximum principle can be found in the text by Fan and Wang (1964).

REFERENCE

Fan, L. T., and C. S. Wang, The Discrete Maximum Principle, Wiley, New York (1964).

A3. THE GENERALIZED REDUCED GRADIENT METHOD (GRG)

The generalized reduced gradient method is a method of nonlinear programming proposed by Abadie and Carpentier (1965, 1966, 1969). The method is essentially a generalization of the Wolfe reduced gradient technique [Wolfe, 1963], which solves problems having a nonlinear objective function and linear (equality) constraints. In the Wolfe method, the variables are classed as independent and dependent. From the set of linear (equality) constraints, the dependent variables are obtained in terms of the independent variables, and the expressions thus obtained are substituted into the objective function. The original problem, therefore, is reduced to an unconstrained one with reduced dimension. A variety of optimization techniques may then be used to find the optimum solution. Applying the same concepts to problems with nonlinear constraints adds to the computational difficulties, but is not altogether impossible.

The general nonlinear programming problem with nonlinear equality constraints is defined as follows:

Determine vector $\bar{X}$ so as to maximize

$$f_0(\bar{X}) \tag{A1}$$

subject to the constraints:

$$\bar{f}(\bar{X}) = 0 \tag{A2}$$

and the boundary conditions:

$$\bar{a} \leq X \leq \bar{b}, \tag{A3}$$

where $\bar{X}$, $\bar{a}$, and $\bar{b}$ are N-dimensional column vectors, and $\bar{f}(\bar{X})$ is an M-dimensional column vector of constraint functions in terms of vector $\bar{X}(N>M)$. Inequality constraints may be employed by the appropriate addition of slack variables.

The problem is solved by partitioning the vector of variables into the independent and dependent sets of variables, of N-M and M dimensions, respectively. Let

$$\bar{X} = [\bar{x},\bar{y}] \tag{A4}$$

where $\bar{x}$ is the (N-M)-dimensional set of independent (basic) variables, and $\bar{y}$ the M-dimensional set of dependent (nonbasic) variables. If the constraint functions satisfy the requirements of the Implicit Function Theorem (Apostol, 1957), then the non-degeneracy assumption is that the dependent variables can be expressed as functions of the independent variables, i.e.,

$$\bar{y} = \bar{\phi}(\bar{x}), \tag{A5}$$

such that $\bar{y}$ is within the boundary:

$$\bar{a} \leq \bar{X} = [\bar{x},\bar{y}] \leq \bar{b}. \tag{A6}$$

When this condition does not hold, the basis is changed until a feasible solution is obtained.

By substituting the vector $\bar{y}$ into the objective function, the problem may now be simply defined as:

$$\text{Maximize} \qquad f_0(\bar{X}) = f_0(\bar{x},\bar{y}) = f_0(\bar{x},\bar{\phi}(\bar{x})) \equiv F(\bar{x}) \tag{A7}$$

subject to:

$$\bar{a} \leq \bar{x} \leq \bar{b} \tag{A8}$$

Computational Procedure

The procedure for using the GRG method is summarized below [Abadie, 1970; Hwang, et al., 1972]:

Step 1. Compute the direction of movement, $\bar{h}^o$, at the starting point $\bar{X}^o = [\bar{x}^o, \bar{y}^o]$, by computing the "reduced gradients" at this point:

$$\bar{g}^{oT} = \frac{\partial F(\bar{x})}{\partial \bar{x}^o} = \frac{\partial f_o}{\partial \bar{x}^o} + \frac{\partial f_o}{\partial \bar{y}^o} \cdot \frac{\partial \bar{y}^o}{\partial \bar{x}^o} . \tag{A9}$$

But from (A2) we have:

$$\frac{\partial \bar{f}}{\partial \bar{x}} + \frac{\partial \bar{f}}{\partial \bar{y}} \cdot \frac{\partial \bar{y}}{\partial \bar{x}} = 0 . \tag{A10}$$

Solving for $\partial\bar{y}/\partial\bar{x}$, we obtain

$$\frac{\partial \bar{y}}{\partial \bar{x}} = - \left(\frac{\partial \bar{f}}{\partial \bar{y}}\right)^{-1} \cdot \frac{\partial \bar{f}}{\partial \bar{x}} . \tag{A11}$$

Substituting (A11) into (A9) gives:

$$\bar{g}^{oT} = \frac{\partial f_o}{\partial \bar{x}^o} - \frac{\partial f_o}{\partial \bar{y}^o} \left(\frac{\partial \bar{f}}{\partial \bar{y}^0}\right)^{-1} \cdot \frac{\partial \bar{f}}{\partial \bar{x}^o} . \tag{A12}$$

Now the "projected reduced gradinets", $\bar{P}^o$, for each component of the independent vector $\bar{x}$, are computed in the following manner:

$$P_i^o = \begin{cases} 0 & \text{if } x_i = \text{lower bound and } g_i^o \leq 0 \\ 0 & \text{if } x_i = \text{upper bound and } g_i^o \geq 0 \\ g_i^o & \text{otherwise.} \end{cases} \tag{A13}$$

$i = 1, 2, \ldots, N\text{-}M$

Now

$$\bar{h}^o = \bar{P}^o \tag{A14}$$

Step 2. It is desirable to stay in the feasible region, or at least close to it, by selecting a proper direction of movement. The steps for vectors $\bar{x}^o$ and $\bar{y}^o$ are $\bar{x}^o + \theta\bar{h}^o$, and $\bar{y}^o + \theta\bar{K}^o$, respectively. The desired movement is along the surface of the constraints. It is accomplished by finding the tangent to $\bar{f}(\bar{x}^o + \theta\bar{h}^o, \bar{y}^o + \theta\bar{K}^o) = 0$ at point $(\bar{x}^o, \bar{y}^o)$, that is:

$$\frac{\partial \bar{f}}{\partial \bar{x}^o} \cdot \bar{h}^o + \frac{\partial \bar{f}}{\partial \bar{y}^o} \cdot \bar{K}^o = 0 \tag{A15}$$

This yields

$$\bar{K}^o = - \left(\frac{\partial \bar{f}}{\partial \bar{y}^o}\right)^{-1} \left(\frac{\partial \bar{f}}{\partial \bar{x}^o}\right) \bar{h}^o \tag{A16}$$

Now,

$$\bar{f}(\bar{x}^o + \theta\bar{h}^o, \bar{y}^o + \theta\bar{K}^o)$$

is to be optimized for θ using a one-dimensional search technique.

Step 3. After calculating:

$$\bar{x}^1 = \bar{x}^o + \theta\bar{h}^o$$

$$\bar{y}^1 = \bar{y}^o + \theta\bar{K}^o$$

$$f_o(\bar{x}^1, \bar{y}^1),$$

the values of the independent variables are projected into the bounds $\bar{a} \le \bar{x} \le \bar{b}$ as follows:

$$x_j^1 = \begin{cases} \text{lower bound} & \text{if } x_j^o + \theta h_j^o \leq \text{lower bound} \\ \text{upper bound} & \text{if } x_j^o + \theta h_j^o \geq \text{upper bound} \\ x_j^o + \theta h_j^o & \text{otherwise} \end{cases} \tag{A17}$$

$j = 1,2, \ldots, N\text{-}M$

Step 4. A feasible solution is developed by solving the following by an iterative method:

$$\bar{f}(\bar{x}^1, \bar{y}^1) = 0.$$

The existence of $\bar{y}^1 = \bar{\phi}(\bar{x}^1)$ is insured by the Implicit Function Theorem as mentioned before. If a component of $\bar{y}^1$ violates a boundary condition (degeneracy), a change of basis occurs. Two cases may arise at the end:

a) if the iterative procedure does not converge to a $\bar{y}^1$, then $\bar{x}^1$ is out of the functional domain. This is alleviated by reducing θ and returning to step 3.

b) if the solution obtained is $\bar{y}^1$, then the solution vector is $\bar{X}^1 = [\bar{x}^1, \bar{y}^1]$. If the solution vector does not improve the objective function, θ is reduced by half and the procedure is returned to step 3.

Step 5. At this step, $\bar{X}^o$ is set equal to $\bar{X}^1$ and the algorithm is repeated. However, if a better value for θ can be somehow determined, a return to step 3 is made before the iteration proceeds.

The termination criterion for the GRG method is, theoretically, when:

$$\bar{P}_i^o = 0, \qquad i = 1,2, \ldots, N\text{-}M.$$

In practice, however, the following three stopping criteria are used:

$$1)\quad ||\bar{P}^o|| = \sqrt{\sum_{i=1}^{N-M} (P_i^o)^2} < \varepsilon_1,$$

$$2)\quad P_i^o < \varepsilon_2 \qquad (A18)$$

$$3)\quad |f_o(\bar{X}^1) - f_o(\bar{X}^o)| < \varepsilon_3.$$

The GRG method has been studied extensively, and coded in FORTRAN by Abadie (1969), Abadie and Guigou (1969, 1970), and Guigou (1969, 1971). Three generations of the program have been developed. The first, called GRG 66, was an experimental code, followed by the second one, GRG 69. An improved code, GREG, is the outgrowth of the first two, and by far, the most improved one. It is obtainable through J. Abadie, Electricité de France, Paris, France.

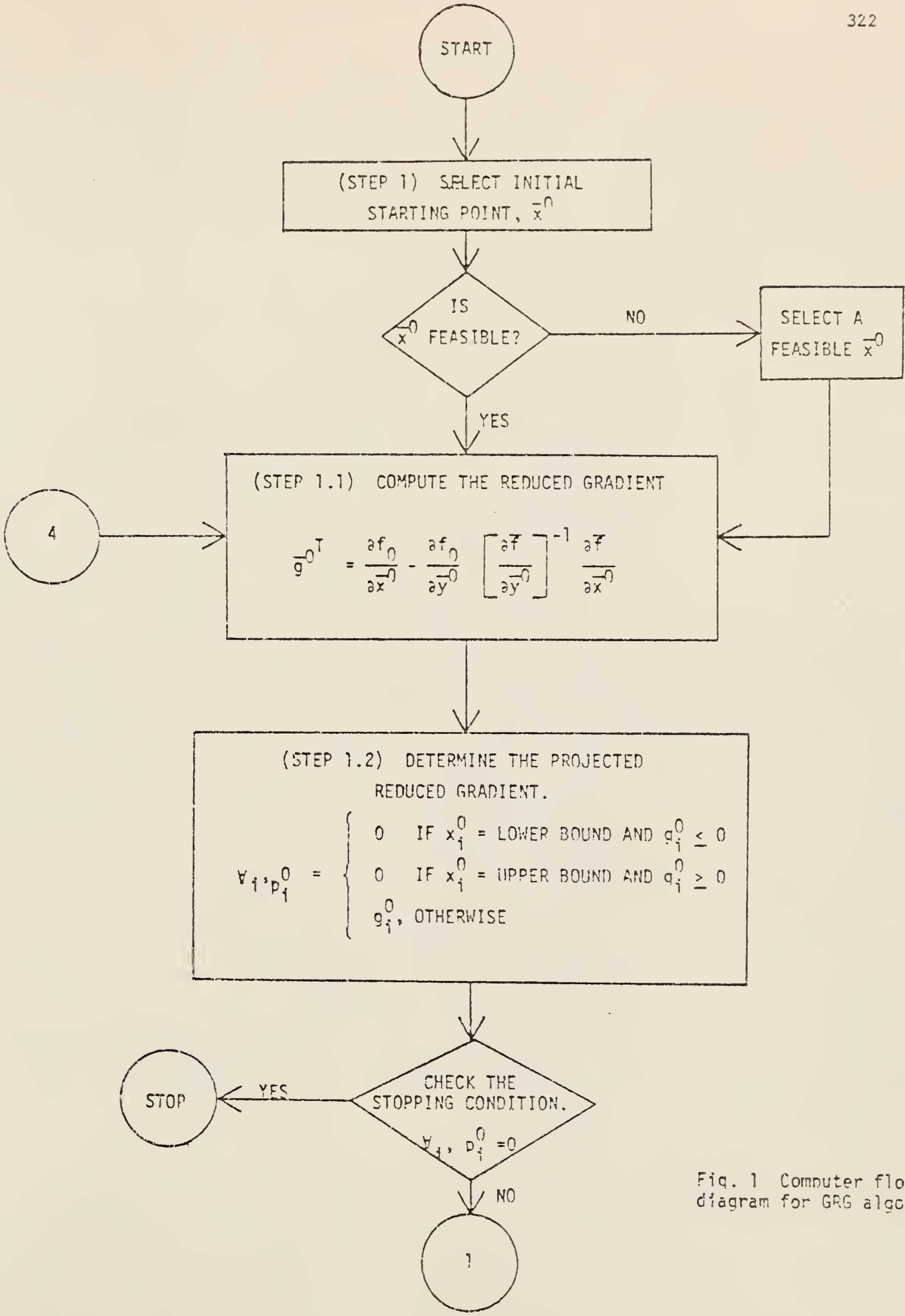


Fig. 1 Computer flow diagram for GRG algorithm

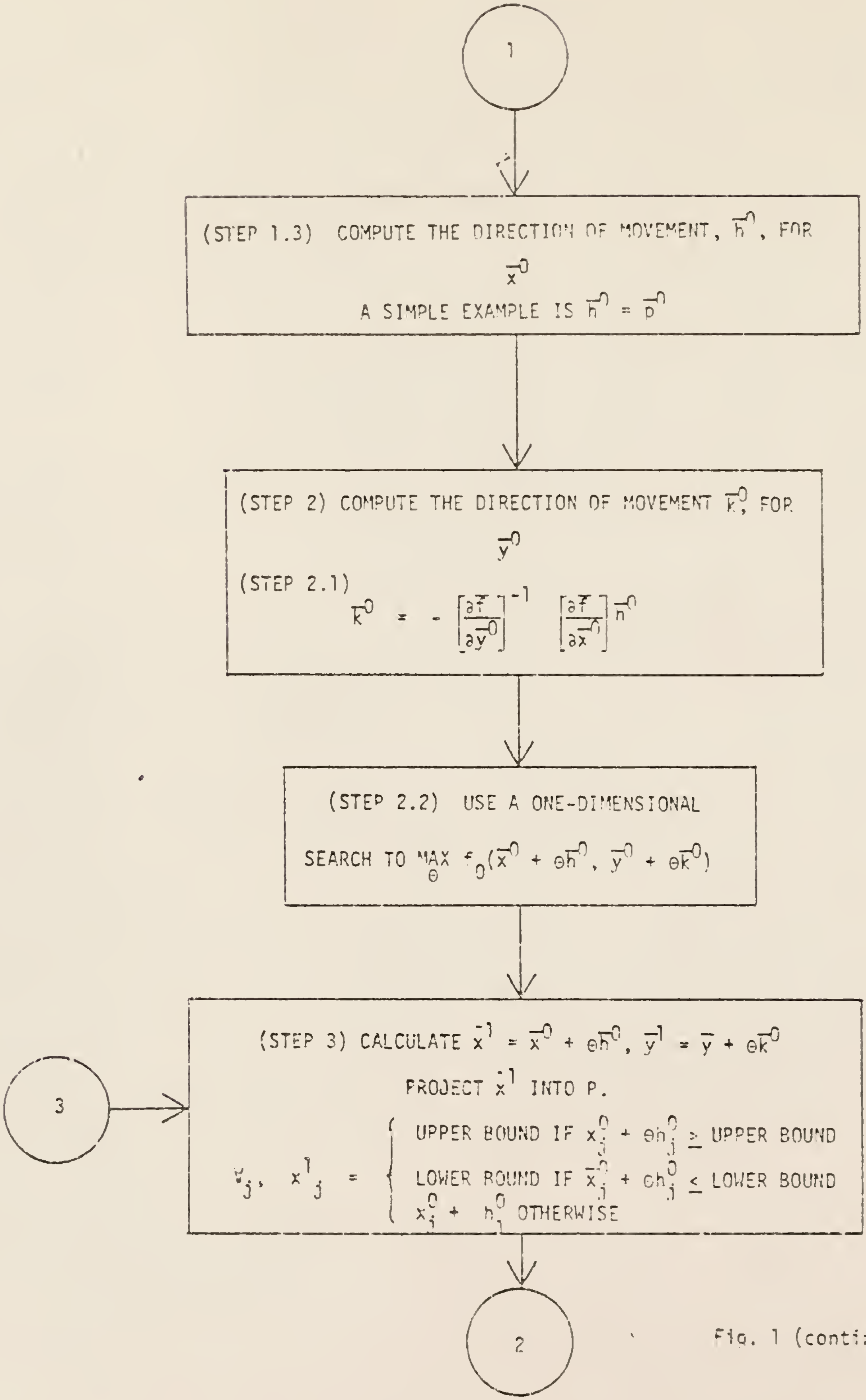


Fig. 1 (continued)

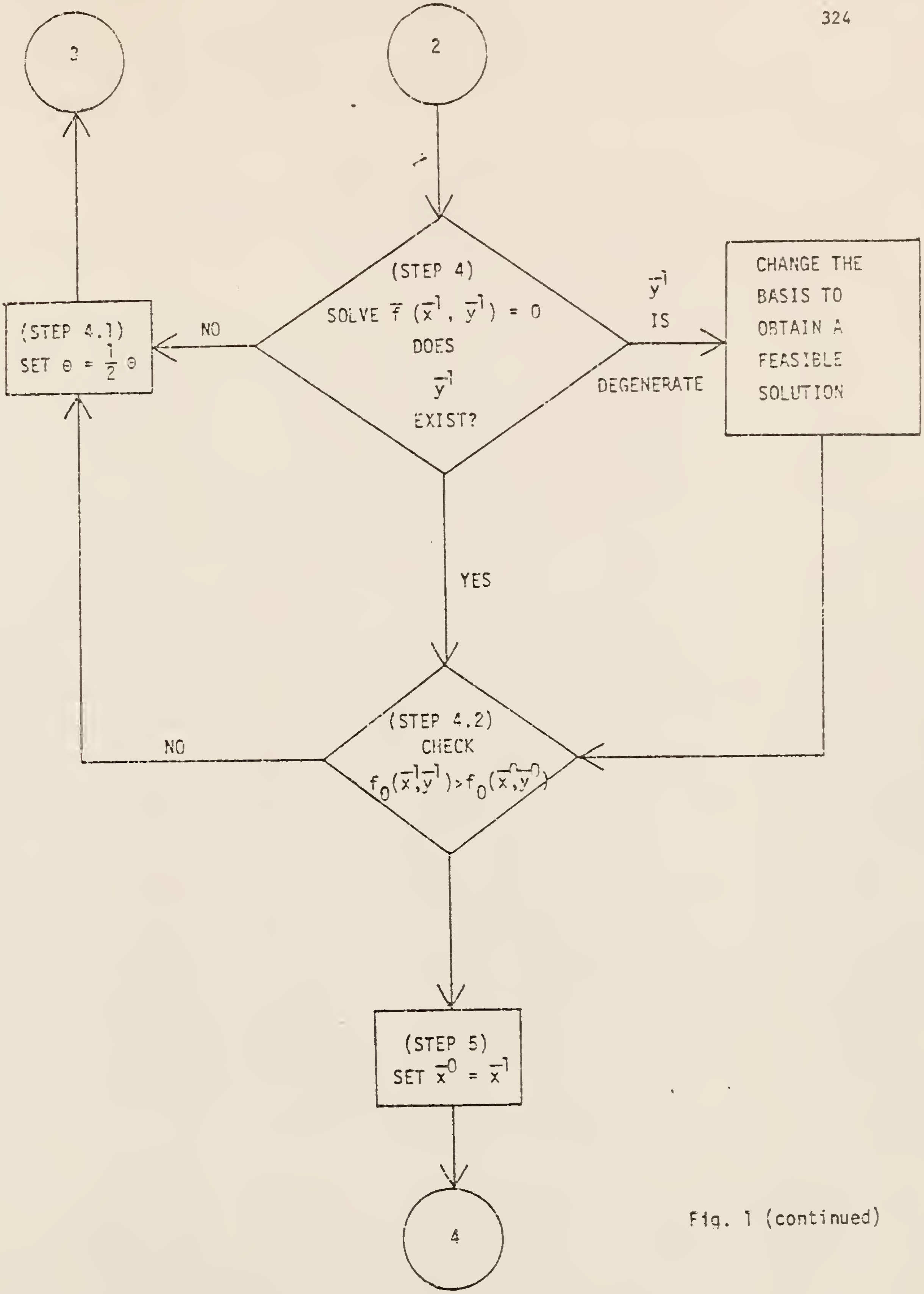


Fig. 1 (continued)

A4. GEOMETRIC PROGRAMMING

In an earlier section we saw how linear programming problems could be formulated in terms of both primal and dual problems. By employing the inequality which states that the arithmetic mean is at least as great as the geometric mean, a dual problem for many optimal design problems may be formulated. Geometric programming uses this inequality and the relationsips of the primal and dual problems to solve optimization problems. The primal problem is expressed in terms of a class of functions which we call positive polynomials, or posynomials for short.

The primal problem is that of minimizing a posynomial S subject to constraints of a certain type. Let M denote the constrained minimum value of the primal function S. Because of the inequality relating the arithmetic and geometric means, there is a related maximization problem concerning a function v which is the dual function. It will be shown that the dual problem is one of maximizing v subject to certain linear constraints. We will also show that M is the constrained maximum vlaue of v as well as the constrained minimum value of S.

Geometric programming is based primarily on the arithmetic mean - geometric mean inequality which states that the arithmetic mean is at least as great as the geometric mean. For the general case, the weighted arithmetic and geometric means satisfy the relation

$$\sum_{i=1}^{n} \delta_i U_i \geq \prod_{i=1}^{n} U_i^{\delta_i} \qquad (1)$$

where the δ_i are the weights which must sum to unity, that is, the normality condition,

$$\delta_1 + \delta_2 + \ldots . + \delta_n = 1, \tag{2}$$

must be satisfied. Equation (1) is an equality if and only if all of the U_i are equal.

Now, suppose we wish to find the minimum vlaue of the objective function

$$S = \frac{x_2}{x_1^2} + x_2 + 2\,\frac{x_1}{x_2} \tag{3}$$

we have

$$S = \frac{1}{4}\left(\frac{4x_2}{x_1^2}\right) + \frac{1}{4}(4x_2) + \frac{2}{4}\left(\frac{4x_1}{x_2}\right) \geq \left(\frac{4x_2}{x_1^2}\right)^{\frac{1}{4}} (4x_2)^{\frac{1}{4}} \left(\frac{4x_1}{x_2}\right)^{\frac{2}{4}} = 4 \tag{4}$$

From this equation we find that 4 is a lower bound for S, that is,

$$S \geq 4 \tag{5}$$

Using differential calculus, we can show that 4 is the minimum vlaue of S and that this occurs at $x_1 = x_2 = 1$.

The preceding example has shown that we can obtain the minimum value of an objective function directly by properly choosing the weights of each term in the posynomial. If the geometric mean is properly weighted, it is independent of its variables, and it is not necessary to determine the values of the variables prior to finding the minimum value of the objective function. It is this unique property of the geometric mean that makes it easy to minimize certain posynomials.

The weighted arithmetic mean - gemoetric mean inequality with the normality condition can be written as

$$u_1 + u_2 + \ldots + u_n \geq \left(\frac{u_1}{\delta_1}\right)^{\delta_1} \left(\frac{u_2}{\delta_2}\right)^{\delta_2} \ldots \left(\frac{u_n}{\delta_n}\right)^{\delta_n} \tag{6}$$

if we let $u_i = \delta_i U_i$ for $i = k, 2, \ldots, n$.

The left side of this inequality is the posynomial S that is to be minimized. For simplicity, we shall refer to equation (6) as _the geometric inequality_, and we shall call the left side _the primal function_ and the right side _the predual function_. Using V to denote the predual function, the inequality, equation (6) becomes

$$S \geq V \tag{7}$$

If the primal function is a posynomial and the u_j are given by

$$u_j = C_j \prod_{i=1}^{m} x_i^{a_{ji}} \tag{8}$$

Substituting the above into

$$V = \left(\frac{u_1}{\delta_1}\right)^{\delta_1} \left(\frac{u_2}{\delta_2}\right)^{\delta_2} \cdots \left(\frac{u_n}{\delta_n}\right)^{\delta_n} \tag{9}$$

gives

$$V = \left(\frac{C_1}{\delta_1}\right)^{\delta_1} \left(\frac{C_2}{\delta_2}\right)^{\delta_2} \cdots \left(\frac{C_n}{\delta_n}\right)^{\delta_n} x_1^{D_1} \; x_2^{D_2} \cdots x_m^{D_m} \tag{10}$$

where

$$D_j = \sum_{i=1}^{n} \delta_i \, a_{ij}, \; j = 1, 2, \ldots, m \tag{11}$$

Sometimes it is possible to choose the weights δ_i in such a way that all of the exponents D_j are zero. When this is possible, the predual function, V, does not depend on the variables $x_1, x_2, \ldots, x_n$. When all of the D_j are zero, equation (10) becomes

$$v = \left(\frac{C_1}{\delta_1}\right)^{\delta_1} \left(\frac{C_2}{\delta_2}\right)^{\delta_2} \cdots \left(\frac{C_n}{\delta_n}\right)^{\delta_n} \qquad (12)$$

which we shall refer to as the dual function, v.

From inequality (7), we know that out objective function, S has a minimum point. We shall use M to designate this positive greatest lower bound of S which must satisfy the inequality

$$S \geq M \geq v \qquad (13)$$

Further details on how to solve optimization problems with constraints using geometric programming are provided in the texts by Duffin et al. (1966) and Wilde and Beightler (1967).

REFERENCES

Duffin, R. J., E. L. Peterson, and C. Zener, Geometric Programming, John Wiley and Sons, Inc., New York (1966).

Wilde, D. J. and C. S. Beightler, Optimization Theory, Prentice-Hall, Englewood Cliffs, New Jersey (1967).

OPTIMIZATION TECHNIQUES FOR SYSTEMS RELIABILITY WITH REDUNDANCY

by

WAY KUO

B.S. (Nuclear Engineering) National Tsing-Hua University, Taiwan, 1972
M.S. (Nuclear Engineering), University of Cincinnati, Ohio, 1975

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

1978

The objectives of this thesis are:

(1) to make a critical review and classification of all system reliability optimization problems and various optimization techniques which have been used;

(2) to study the optimization techniques of the generalized reduced gradient method (GRG) and a generalized Lagrangian functions method applied to solve the system reliability optimization problems. Both of the algorithms have not been applied in this field yet;

(3) to extend the regular system reliability optimization problems for simple reliability allocation problem or simple redundancy allocation problem to the one taking both of them in consideration.

(4) to propose new methods for determining integer solution, particularly, heuristic methods.

The rationale is that this "topic" is another step in the collection, classification, presentation and testing of new problems and new techniques that is vital to solving the system reliability optimization problems.

A state-of-the-art review of the literature related to optimal system reliability with redundancy is presented in Chapter 2.

In Chapter 3, the optimization techniques are presented, which are (a) to maximize the system reliability of various system configurations subject to the 'cost' constraints, or (b) to minimize any specific 'cost' while satisfying the minimum requirement of the system reliability. In the chapter, literature published on optimal system reliability is classified and critically reviewed. Various problems are also classified and resolved by heuristic approach, dynamic programming, and integer programming.

These optimization techniques always give solution of integer numbers which meet the integer requirement of redundancy allocation problems.

Sequential Unconstrained Minimization Technique (SUMT) has been widely used in solving many optimization problems. The problem with a tangent form cost function is successfully solved by SUMT. Generalized Reduced Gradient method (GRG) and generalized Lagrangian functions method have been used in solving the system reliability optimization problems in Chapter 3.

The maximum principle, the method of Lagrange multipliers and the Kuhn-Tucker conditions, geometric programming, and several miscellaneous optimization techniques (e.g. linear programming and separable programming) are also presented in Chapter 3 to cover the comprehensive discussion of optimization techniques having been used in system reliability optimization problems.

The extension to the usual reliability optimization problems is presented in Chapter 4. The problem is to include the determination of optimal level of component reliability and the number of redundancies in each of the stages simultaneously. The Hooke and Jeeves pattern search technique in combination with a heuristic approach is proposed to solve this mixed integer nonlinear programming problem.