

NOTICE TO USERS

Portions of this document have been judged by the Clearinghouse to be of proper production quality and not fully legible. However, in an effort to $n_{a,c}$ as much information as possible available to the public, the Clearinghouse sells this document with the understanding that if the user is not satisfied, the document may be returned for refund.

If you return this document, please include this notice together with the IBM order card (label) to:

e entre tid i die ein die Kuidelichendelichendelichendelichen die Studie die Stationer

dietscheinen und die eine

2 1

Clearinghouse Attn: 152,12 Spingfield, Va. 22151

Best Available Copy

COMPARISON OF RELIABILITY ALLOCATION OPTIMIZATION METHODS

To see the second set as a definition [[b]]¹

Thesis

	William Z.	Spivey
GRE/EE/69-2	2d Lt.	USAF

This document is subject to special export controls and each transmittal to foreign governments of foreign nationals may be made only with prior approval of the Dean, School of Engineering, Air Force Institute of Technology (AFIT-SE), Wright-Patterson Air Force Base, Ohio 45433.

COMPARISON OF RELIABILITY ALLOCATION OPTIMIZATION METHODS

THESIS

Presented to the Faculty of the School of Engineering of the Air Force Institute of Technology

Air University

en de la cale

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

William Z. Spivey, B.S.M.E.

Second Lieutenant USAF

Graduate Reliability Engineering

December 1969

This document is subject to special export controls and each transmittal to foreign governments of foreign governments of foreign nationals may be made only with prior approval of the Dean, School of Engineering, Air Force Institute of Technology (AFIT-SE), Wright-Patterson Air Force Base, Ohio 45433.

PREFACE

This study is the result of a suggestion by Professor T. L. Regulinski, who later became my thesis advisor, that the methods of reliability allocation needed further development.

While I was searching through the literature dealing with reliability allocation, I discovered that allocation models had been developed in each of the areas in which I was interested. This reduced my problem from one of developing and comparing an allocation model in each of the three areas of Lagrange multipliers, mathematical programming, and dynamic programming to one of merely comparing the three allocation models that had been previously developed.

The dynamic programming allocation model was developed by John D. Kettle (Ref 10:249) and the Lagrange multiplier allocation model was developed by Hugh Everett III (Ref 4:399). The algorithm presented here for mathematical programming is based on a heuristic argument that was suggested by Peter J. Kolesar (Ref 11:317).

I wish to express my thanks to Professor Regulinski for the helpful advice and constructive criticism without which I probably would never have completed this study.

ii

Preface (Contd)

I also wish to express thanks to my recent lride for her encouragement during the times when I felt that I was making no progress. Finally, I claim responsibility for any errors that may be found in this work.

William Z. Spivey

યન્ય પ્રેક્ષેન્ટ

ar a Sollah da bilar i "ata ana aning aning aning

Contents

Preface	Page
List of Figures	vi
List of Tables	vi
Abstract	vii
I. Introduction	1
Problem Scope Assumptions Standards Approach	2 2 2 2 2 2 2 2 2 2 2 3 4 4
II. Lagrange Multipliers	6
Theory	6
III. Mathematical Programming	
Theory and Applicat	ion 12
IV. Dynamic Programming	14
Theory	· · · · · · · · · · · · · · · · · · 14 · · · · · · · · · · · · · · · · · 14
V. Conclusions	
Results	· · · · · · · · · · · · · · · 21 · · · · · · · · · · · · · · · 26
VI. Recommendations for Furt	her Study 27
Bibliography	
Appendix I	

the subsection of the

<u>Contents</u> (Contd)

and Price Balladies Jacob

etter solution

Appendix	II	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	•	•	•	•	Page 36
Appendix	III	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	•	•	•	41
Appendix	IV	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	48
Vita	•	•		•		•		•		•	•	•	•		•	•	•	•	•	•	50

List of Figures

n tille att to

Figure		Page
1.	Combining Subsystems 1 and 2	17
2.	Combining Subsystems 1 and 2 with 3	18
3.	Combining Subsystems 1, 2, and 3 with 4 .	19

List of Tables

<u>Table</u>		Page
I	Component Reliability and Cost for Each Subsystem	16
II	Final Allocation of Example	20
111	Component Reliability and Cost for Each Sub- system in the Representative System	23
IV	Final Allocation for Each Method	24
v	Symbols in the Lagrange Multiplier Method	30
VI	Symbols Used in Mathematical Programming	36
VII	Symbols Used in the Dynamic Programming Method	41
VIII	Symbol: Used in the Random Number Generator	48

ABSTRACT

The reliability allocation problem has previously been solved by methods other than the classical optimization methods. This thesis brings together and compares three allocation models that are based on optimization methods.

It was found that dynamic programming gives the optimal solution to the allocation problem. The model based on Lagrange multipliers gave a solution that was within 0.5% of the optimal solution given by dynamic programming. The mathematical programming algorithm gave a solution that was within 0.6% of the optimal solution.

COMPARISON OF RELIABILITY ALLOCATION OPTIMIZATION METHODS

I. INTRODUCTION

Reliability allocation models that are now in use are based on assignment of reliability goals to the various subsystems to fulfill the requirement that the system failure rate equal the sum of the failure rates of the various subsystems. These methods of allocation are based on the assumptions that times to integrant failure are exponentially distributed. The two models most frequently used are the complexity and the failure rate weighting models. Differention of the state of the state of the second state of the

a subhalana'

When the complexity model is used, the most complex subsystem is assigned the highest failure rate. This failure rate will be equal to the ratio of the number of integrants in the subsystem under consideration to the number of integrants in the entire system multiplied times the desired system failure rate. The end result of this model is to assign lower reliability goals to the more complex subsystems and higher reliability goals to the more simple subsystems.

The failure rate weighting model assigns failure rates to the various subsystems based on failure rates that have

been achieved on similar subsystems in the past. Parts that have had low failure rates are assigned reliability goals that are harder to achieve than parts that have had high failure rates. the solution of the share

PROBLEM

The problem is to develop three reliability allocation models, one using Lagrange multipliers, one using mathematical programming, and one using dynamic programming. The allocation will be made in such a way that one of the system parameters will be minimized subject to a constraint placed on the system reliability. These system parameters are such as cost, weight, development time, and test cost.

These three models will be compared and the advantages and disadvantages of each will be discussed.

SCOPE

The study will apply the mathematical methods of Lagrange multipliers, mathematical programming, and dynamic programming to the allocation problem. Any discussion of the theory behind these methods will be limited to that necessary for the application of the method.

The comparison will point out the advantages and disadvantages of each method and will also discuss, for each

method, any changes in assumptions that could be made to make the method more general.

ASSUMPTIONS

The main assumptions are that the operation of any subsystem is independent of the operation of any other subsystem and that the system will operate as long as all of the subsystems are operating.

These assumptions imply that the reisbility of the system will be the product of the reliabilities of all sub-systems.

$$RS = \iint_{i=1}^{M} R_{i}$$
(1)

Furthermore, each subsystem is assumed to be composed of N identical components, N-1 of which are redundant. The reliability of each subsystem will be

$$R_{i} = 1 - (1 - r_{i})^{N_{i}}$$
(2)

The subsystem cost is assumed to be a linear function of the number of components in the subsystem.

$$C_i = N_i C_i$$
(3)

STANDARDS

The comparison of the allocation models will be made on the basis of optimality of the solution, flexibility of the algorithm, and simplicity of computation.

Optimality of the solution was chosen as a basis for comparison because the main objective of allocation is to do it in the most optimal manner. If the algorithm does not give a solution that is at least near optimal, then it is of no value.

A flexible algorithm is desirable because an algorithm that can be used for a general reliability allocation problem is better than an algorithm that is limited to more restricting assumptions. In particular, a good algorithm should be able to handle the situation where the cost function is not linear.

Simplicity of computation was chosen as a basis for comparison because a very long algorithm that gives an optimal solution is not necessarily superior to a shorter algorithm that gives a near optimal solution. This is especially true in the situation where the model is only a rough approximation of the actual situation.

APPROACH

The approach to solving the problem will be to first

GRE EE/69-2

develop the necessary algorithms. Then these algorithms will be applied to solving an allocation problem for a representative system with component reliabilities and costs that are randomly selected. From the results of this allocation, the limitations of each algorithm will be determined and will be compared with the limitations of the other algorithms. Then the advantages and disadvantages of each model will be discussed.

ebiliti Betti ni Meno. In et in designa (in 1990) (in Minimus designa (administrativa)) esti a

II. LAGRANGE MULTIPLIERS

As most often used, Lagrange multipliers are introduced in context with differentiable functions. In these cases they are used to produce constrained stationary points.

Another technique involving the use of Lagrange multipliers has been practiced at the Weapons Systems Evaluation Division, Institute for Defense Analysis. The objective of this technique is the maximization of a function with constraints, rather than the location of stationary points (Ref 4:399-417).

THEORY

The basic problem can be described by defining an arbitrary set, S, which is the set of all possible actions. Defined on this set, S is a payoff function, H. There are N real valued functions, $C^{k}(x)$ (k=1,...,N), defined on S which are called the resource functions. The problem is to find

subject to $C^{k}(X) \leq C^{k}$ (5) for all k, where C^{k} is the maximum permissible value of this resource function.

This technique is based on the following theorem (Ref 4:401).

Theorem 1.

Z^k, k=1,...,N, are nonnegative real numbers.
 X* contained in S maximizes the function

$$H(X) = \sum_{k=1}^{N} Z^{k} C^{k}(X)$$

over all X contained in S.

3. X* maximizes H(X) over all those X contained in S such that $C^k(X^*) = C^k$ for all k.

This theorem says, that for any choice Z^k, k=1,...N, that if an unconstrained maximum of the new function

$$H(X) - \sum_{k=1}^{N} Z^{k}C^{k}(X)$$
 (6)

can be found, then this solution is a solution to that constrained maximization problem with constraints equal to the amount of each resource expended in achieving the unconstrained solution.

According to this theorem, it is possible to select an arbitrary, nonnegative set of Z's, find an unconstrained maximum for equation (6) and this gives the solution to a constrained problem. The only difficulty is that there is no way of knowing in advance which constrained problem will be solved. This diffuculty may be avoided through the use of an iterative procedure that will examine the spectrum of

Z's and isolate the desired constrained solution.

For the general problem this procedure does not guarantee a solution in every case. It does guarantee that if a solution can be found, then it will be optimal.

A subclass of the general problem is one that may be called the cell problem. The cell problem is the problem in which there are a number, M, of independent areas into which the resources may be committed and for which the total payoff is the sum of the payoffs from each independent area.

For this type problem there exists for each cell a strategy set, S_i ; a payoff function, H_i defined on S_i ; and N resource functions C_i^k defined on S_i . The problem is to find a strategy set, one element for each cell, that maximizes the total payoff subject to constraints on the total resource expended. That is, find:

Max $\sum_{i=1}^{M} H_{i}(X_{i})$ (7) All choices of $[X_{i}]$ i=1 $H_{i}(X_{i})$ (7) X_{i} contained in S_{i}

subject to $\sum_{i=1}^{M} C_{i}^{k} c^{k}$ for all k. (8)

This problem may be put in the form of equation (6) by noting that

$$s = \sum_{i=1}^{M} s_i$$
 (9)

and that
$$H(X) = \sum_{i=1}^{M} H_i(X)$$
 (10)

and that
$$C^{k}(X) = \sum_{i=1}^{M} C^{k}_{i}(X_{i})$$
 for all k. (11)

By applying the theorem to equations (9), (10), and (11) it is seen that finding the maximum of the constrained function (7) is equivalent to finding

Max M M M
$$\prod_{i=1}^{M} s_i \left[\sum_{i=1}^{M} H_i(X_i) \right] - \sum_{k=1}^{N} z^k \left[\sum_{i=1}^{M} c_i^k(X_i) \right].$$
 (12)

By interchanging the order of summation, the function (12) becomes

$$\begin{array}{ccc} & \text{Max} & \text{M} & \text{M} \\ \text{X contained in} & & \mathcal{I}_{i=1}^{\text{M}} S_{i} & \sum_{i=1}^{M} [H_{i}(X_{i}) - \sum_{k=1}^{N} Z^{k} C_{i}^{k}(X_{i}) \end{array}$$
(13)

$$=\sum_{i=1}^{M} S_{i}$$
(9)

and street as he is the street of the substrate of the substrate dual by the substrate of the substrate of the

and the order

A. F. F. Bersh, et C.B.R.R.B.B.B.B. Advised A. F. M.C.

the function (13) is maximized by maximizing

S

$$H_{i}(X_{i}) - \sum_{k=1}^{N} Z^{k} C_{i}^{k}(X_{i})$$
(14)

independently for each cell.

Since

The theorem guarantees that if the Lagrangian in each cell has been correctly maximized then the result is a global maximum for the entire problem.

APPLICATION

Applying this to the problem of reliability allocation, the system reliability is given by

$$RS = \frac{M}{n_{i=1}} [1 - (1 - r_{i})]$$
(15)

<u>նեն՝ երդելու հանձները հանձները</u>

Since the logarithm is a monotonic increasing function, maximizing the logarithm is the same as maximizing the function itself. So the payoff is taken to be the logarithm of the reliability.

$$H = Ln(R) = \sum_{i=1}^{M} Ln[1-(1-r_i)]$$
(16)

Now the equation is in the form of the cell problem. To maximize equation (16), it is only necessary to maximize independently for each cell or subsystem the expression

$$H_{i}(N_{i}) = Ln[1-(1-r_{i})^{N_{i}}] - Z c_{i}N_{i}$$
 (17)

Since the functions, $H_i(N_i)$, are concave, each of them can be maximized analytically by differentiation. The integer maximum for all integers equal to or greater than 1 can be found by checking the integers on each side of the real maximum to see which gives the larger payoff. The result is obtained as

$$\frac{d H_{i}(N_{i})}{d N_{i}} = \frac{-(1-r_{i})^{N_{i}} Ln(1-r_{i})}{[1-(1-r_{i})^{N_{i}}]} Z c_{i}$$
(18)

or

$$N_{i} = \frac{Ln (Z c_{i}) - Ln (Z c_{i} - Ln (1 - r_{i}))}{Ln (1 - r_{i})}$$
(19)

for real N.

This formula is applied for each subsystem, $[N_i]$ and $[N_i + 1]$ are examined to find which integer maximizes, then the reliabilities and costs are summed to produce an optimal solution.

This procedure is repeated for a series of Z's greater than zero. For each Z, the maximum reliability for a given cost is found. The series of reliabilities is then examined to find which solution gives the reliability nearest to but greater than the required system reliability. This solution gives the allocation for minimum cost that gives the required system reliability.

Appendix I contains a listing of a computer program to perform the necessary calculations.

III. <u>MATHEMATICAL PROGRAMMING</u> Sequential Integer Programming

2000 - an sha shore as sha anna an an a

Mathematical programming is defined as a programming procedure in which the goal is to optimize an objective Junction subject to one or more constraints.

THEORY AND APPLICATION

Consider the problem of trying to improve the reliability of a system by adding one more redundant component to one subsystem. Assume each subsystem already has N_k components. The increase in reliability of the system by adding one component to the <u>i</u>th subsystem will be

$$E_{i} = [_{k=i} 1 - (1 - r_{k})^{N_{k}}] [1 - (1 - r_{i})^{N_{i}+1}] - \frac{M_{k=1}}{1 - (1 - r_{k})^{N_{k}}}.$$
(20)

If the object is to get the greatest reliability improvement per unit cost, the additional redundant component should be added to the subsystem where the quantity A_i is the largest where A_i is defined as in equation (21).

$$A_{i} = \frac{E_{i}}{c_{i}}$$
(21)

The problem is then expanded to the problem of trying to achieve a system reliability, RS>R. It is intuitively appealing that this can be done in an optimal manner by an iterative scheme composed of an unknown number of steps if at each step the set, A_i , (i=1,...,M), is generated and the one additional component is added to the subsystem where the quantity A_i is the greatest. The iterative scheme continues until a system reliability, R≥RS, is achieved.

For the reliability allocation problem at hand, the initial solution was selected such that each subsystem would have a reliability equal to or greater than the desired system reliability, RS.

Appendix II contains a listing of a computer program to make these calculations.

IV. DYNAMIC PROGRAMMING

n october 1 litteriotoria (ING Ariji

Dynamic programming has been developed since 1950. The idea is that starting at the last stage of an allocation process, it is possible to ennumerate all of the alternatives. Thus, it is possible to select the alternative that gives the optimum policy for the last stage. Now, since the optimal policy for the last stage is known, it is possible to find the optimum policy for the next to last stage. Then the process is repeated for each stage working from the last stage toward the first stage.

THEORY

The basic theorem of dynamic programming is "The Principle of Optimality." It states that "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision". A proof of this theorem is given by Bellman and Dreyfus (Ref 2:15).

APPLICATION

The application of this theory to a reliability allocation process is as follows:

(1) Plan a sequence for combining the stages or

subsystems. Continue until all subsystems have been combined.

The plan used for the process discussed here is to combine the first and second subsystems, then join this combination with the third subsystem. In general, the <u>J</u>th subsystem is joined with the combination of the J-1 previous subsystems.

(2) Determine the minimum number of components for each subsystem that will give the subsystem under consideration a reliability greater than or equal to the required system reliability, RS. This will be the base requirement. The optimal policy will consider only the extra costs and reliabilities above the base requirements. (3) Prepare the cost-reliability sequence for each subsystem.

(4) Combine the first grouping and select the dominating sequence. The dominating sequence is the optimal policy such that for any given entry, R_i, C_i (i=1),...,M), that entry is the cheapest entry in the group with reliability exceeding R_i -1.

(5) Proceed in this same manner until all of the subsystems have been combined.

The allocation is found from the final entry that gives a reliability greater than or equal to the required

system reliability, RS.

The procedure is demonstrated by the following example. Example: Given a system of 4 subsystems, a required system reliability of .99, and component costs and reliabilities as given in Table I, find the optimum allocation of reliability.

Table I Component Reliability and Cost for Each Subsystem

Subsystem		Integra	Base		
-	Cost	Reliability	Unreliability	Requirements	
 1	10	. 90	. 10	2	
2	15	.95	.05	2	
3	13	.93	.07	2	
4	17	. 92	. 08	2	

		,		
Sequence Number	1	2	З	4
Cost	20	30	40	50
Unreliability	.01	. 001	. 000 1	.00001
·	(1)-	-(2)-	(3)	
3.0	50	60	,70	80
.0025	.0125	.0035	.0026	.0025
		(4)		+ (6)
2			l	
4.5	65	75	85	95
2 .000125	.0101	.001125	.000225	.000135
3			4(7)-	(8)
60	80	90	100	110
.00000625	.010	.0010	.000/06	.000016
4				
75	95	105	115	125
.000003125	.010	.0010	. 00010	. 0000103

USE CONTRACTOR

whether the first

A United as

e balanter e

Fig. 1. Combining Subsystems 1 and 2

The order of combination will be 1 and 2, the combination of 1 and 2 with 3, and finally, the combination of 1, 2, and 3 with 4.

The unreliability of the combination of the entries of column \underline{i} and row \underline{j} is given by equation (22). When the reliability is close to 1, the unreliability of the

combination is approximately the sum of the unreliability of each. The cost of the combination of the entries in column \underline{i} and row \underline{j} is given by equation (23).

			182			
Sequend	ce Number	1	٤	3	4	5
Ca	ost f	50 ,	ر <i>ه</i> ه	70 1	75 1	85
Unrel	iability	.0125	.0035	.002.6	.001125	.000225
		(i)_		(3)		
	1					
	26	76	86	96	101	111
	.00.49	,0174	.0084	.0075	.00602.5	.005125
			/ (4)-	>(5)		
	2					
	39	69	99	109	114	124
2	.000343	.012843	.003843	.002943	.001468	.000568
3						
	3		1			
	52	102	112	122	127	137
	.000024	.01250	.003524	.002624	.001149	.000249
	4					
	65	115	125	135	140	150
	.00000.168	.01250	.00350	.00260	.001126	.00022458



$$UR_{ij} = (1-R_i) + (1-R_j) - (1-R_i)(1-R_j)$$
(22)

$$C_{ij} = C_{i} + C_{j}$$
(23)

After all subsystems have been combined, the allocation is found by examining the last combination to find the entry that meets the required reliability and has the lowest cost, then working back through each subsystem to find the allocation for that subsystem. For this example the final allocation is given in Table II.

H H

[ı, z	Ę 3		
Sequence Number	1	2	3	4	5
Cost	76	86	96	99	109
Unreliability	.0174	.0084	.0075	.003843	.002943
	()-	(E).	(3).	→ (4)	
, I I I I I I I I I I I I I I I I I I I					
34	110	120	130	133	143
.0064	.0238	.0149	. 0139	.01024	.00934
4	[(5)			
2					
51	127	137	147	150	160
.000512	.017912	. 008912	.008012	. 004355	.003455
3					
68	/44	154	164	167	177
. 0 000 10 96	.01744	.00849	.00754	.003883	.002983

Fig. 3 Combining Subsystems 1, 2, and 3 with 4

Final Allocation of Example							
System Cost 137 System Reliability .9911							
Subsystem	Cost	Unreliability	Reliability				
1 2 3 4	30 30 26 51	.001 .0025 .0049 .00051	.999 .9975 .9951 .99949				

There are two refinements for this procedure that will reduce the time required. These improvements restrict the number of entries that must be examined in order to find the next element in the dominating sequence.

(1) A row or column for which the unreliability is greater than the unreliability of the present entry in the dominating sequence need not be examined to find the next element in the dominating sequence.

(2) If the present entry in the dominating sequence is in column \underline{i} and row \underline{j} , then the next entry will either be in column $\underline{k \leq i}$ or row $\underline{1 \leq j}$.

A computer listing of a program to make these calculations is given in Appendix III.

V. CONCLUSIONS

After each of the allocation models had been developed, the vehicle with which to compare them had to be selected. The decision was made to compare them using a system composed of 20 subsystems. The component reliabilities and costs for each of these 20 subsystems were selected at random. The component reliabilities were selected from the uniform distribution from 0.5 to 1.0. The component costs were selected from the uniform distribution from 0.0 to 1000.0. (See Appendix IV) The system reliability requirement was selected to be .998.

H-childenalmerte

ADD. I. CORPORED D.

addressed and a set 1

्रधानुबहित्ति चहान्य सिन्धानिक ब्रोतिब्दाक्षेण्ठां स्थानियोज्ञा । जन्मन्ते कः १९९० कः अन्त क्रम्

RESULTS

As seen from Table IV, the solution given by the dynamic programming method was optimal. The system cost from the allocation made by the Lagrange multiplier method was within 0.5 per cent of the optimal solution. The solution given by the mathematical programming algorithm was within 0.6 per cent of the optimal solution. From this it is seen that all of the allocation algorithms give solutions that are either optimal or close to optimal.

According to the theory of dynamic programming, an optimal solution is guaranteed, so the result from this part

of the problem was as expected.

Now consider the result of the Lagrange multiplier algorithm. The theory guarantees that the allocation given by this method is optimal. However, it is an optimal solution of a reliability allocation problem where the desired system reliability is equal to .99808056. A gap exists around the area that gives a system reliability equal to .998. The Lagrange multiplier technique is unable to find the optimal solution in this area. This gap is due to a nonconcavity in the function of optimum payoff in terms of resource constraints (Ref 4:407).

The mathematical programming method is based on a heuristic argument and does not guarantee an optimal solution. The solution given, which is within 0.6 per cent of the optimal solution is as good as expected.

The major disadvantage of the dynamic programming algorithm is the large computer memory required to make the calculations. In the problem considered here, at stage 19 the optimal sequence contained 667 entries. Each entry requires a storage space for cost and one for unreliability. In addition, if the program is to print out a final solution, enough space is required to store the location of each entry in the optimal sequence for each stage in dynamic programming process. In this problem a matrix of more than 20 x 667

6.1.4. 1-1

is required to store the location of each entry in the optimal sequence for each stage. The total memory required for the dynamic programming algorithm was 22695 bits. This is approximately 50 times the memory required for either the Lagrange multiplier technique or the mathematical programming technique.

Subsystem	Cost	Reliability
1	593	. 50000069
2	641	.50001587
3	750	.50036501
4	253	.50839523
5	824	.69309029
6	956	.94107663
7	984	.64476239
8	641	.82953494
9	752	.47390355
10	285	.82398152
11	554	.95175489
12	744	.88622 ³⁷
13	101	.88311443
14	314	.81163169
15	228	.66752880
16	245	.8531637
17	635	.62273432
18	599	.82288922
19	767	.92645199
20	630	.80839568

Table III Component Reliability and Cost for Each Subsystem in the Representative System

THE PARTY INTO A REPORT OF A DESCRIPTION OF

	Yethod
	Each
le IV	for
Tab	Allocation
	Final

Subsystem	Dynamic	Programming	Lagrange	Multipliers	Mathemati	lcal Programing
	Cost	Reliability	Cost	Reliability	Cost	Reliability
-	6077	.99987793	7709	.99987793	7709	.99987793
7	7692	.99975595	7692	.99975595	7692	.99975595
	0006	.99975798	0006	.99975798	0006	.99975798
4	3542	.99995184	3289*	* 60202*	3289*	.99990205*
5	6592	.99992128	6592	.99992128	6592	.99992128
و	3824	499998794	3824	. 99998794	3824	.99998794
2	7872	.99974640	7872	.99974640	7872	.99974640
ŝ	3205	.99985605	3205	.99985605	3205	.99985605
6	7520	.99983634	7520	.99982634	7520	.99982634
10	1710	.99997026	1710	.99997026	1710	.99997026
11	1662	.99983644	1662	• 99988644	1662	• 99988644
12	2976	.99983241	3720*	*£6086666*	3720*	*6086666
13	606	*****	505*	*81826666	606	4466666
14	1884	.99995533	1884	.99995533	1884	.99995533
15	2052	.99995036	2052	.99995036	2052	.99995036
16	1470	199998997	1470	79999897	1470	16686666°
17	5715	.99984518	5715	.99984518	5715	.99984518
18	3594	£1696666°	3594	£1696666°.	3594	.99996913
19	3068	£7079920.	3068	.99997073	3068	.99997073
20	3780	.999995051	3780	.99995051	3780	.99995051
Total	85473	.99800140	85863	.99808056	85964	08660866.
* Indicates	subsystem in whi	ich allocation by	given method	differs from opl	timum.	

As a further comparison of the relative length of computation for the three algorithms, consider the computer time required for each. The dynamic programming algorithm required 9 minutes and 55 seconds, the mathematical programming algorithm required 50 seconds, and the Lagrange multiplier technique required only 36 seconds.

One other area of comparison is the area of flexibility. All three of the allocation methods could, with little effort, be made to handle the situation where the cost function is nonlinear.

The change required in the Lagrange multiplier technique would be to substitute the new cost function in equation (17). The change required of the dynamic programming algorithm would be to read into the computer the new reliability cost sequence for each stage. The change in the mathematical programming algorithm would require a method for calculating the increase in system cost by adding one component at the <u>i</u>th subsystem similar to the method for calculating the increase in reliability as given in equation (20).

SUMMARY

From the results of the problem considered here, it is felt that the Lagrange multiplier technique is best. Although the solution obtained by the use of dynamic programming was optimal, it is felt that the large time requirement offsets this fact.

VI. <u>RECOMMENDATIONS</u> FOR FURTHER STUDY

Z

The next direction to be followed in this course of study should be to consider the problem of maximizing the reliability of the system subject to multiple constraints. Bellman and Drefus (Ref 2) and Proschan and Bray (Ref 14) have examined the dynamic programming aspects of the problem involving multiple constraints. Everett (Ref 4) has formed the basis for study in this area using Lagrange multipliers. Koelesar (Ref 11) has considered the solution of this problem by the use of linear programming.

It is felt that each of these areas is worth further examination.

ารสร้างในการการจะ ขุณฑรารรรรรรรรรรษณฑราร

BIBLIOGRAPHY

- 1. Aris, Rutherford. <u>Discrete Dynamic Programming</u>. New York: Blaisdel Publishing Company, 1964.
- Bellman, Richard E. and Stuart E. Dreyfus. <u>Applied</u> <u>Dynamic Programming</u>. Princeton, New Jersey: Princeton University Press, 1962.
- 3. Dantzig, George B. <u>Linear Programming and Extensions</u>. Princeton, New Jersey: Princeton University Press, 1963.
- Everett, Hugh III. "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources." <u>Operations Research</u>, 10: 399-417 (May-June, 1963).
- Glover, Fred. "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem." <u>Operations Re</u>search, 13:879-919 (November-December, 1965).
- Heyne, J. B. and G. H. Sandler. "System and Subsystem Peliability." in <u>Reliability Engineering for Electronic</u> <u>Systems</u>. Edited by Richard H. Meyers, <u>et al</u>. New York: John Wiley and Sons, Inc., 1964
- 7. Hildebrand, F. B. <u>Methods of Applied Mathematics</u>. New York: Prentice Hall, Inc., 1952.
- Hillier, Frederick S. and Gerald J. Lieberman. <u>Intro-</u> <u>duction to Operations Research</u>. San Francisco: Holdenday, Inc., 1967.
- 9. Hinely, J. T. and B. F. Shelley. <u>Method of Allocating</u> <u>Reliability Goals to Aircraft Subsystems and Components</u>. Aerospace Reliability and Maintainability Conference. AIAA/SAE/ASME, 1963.
- Kettelle, John D. Jr. "Least Cost Allocation of Reliability Investment." <u>Operations Research</u>, 10:249-265 (March-April, 1962).

Bibliography (Contd)

- 11. Kolesar, Peter J. "Linear Programming and the Reliability of Multicomponent Systems." <u>Naval Research</u> <u>Logistics Quarterly</u>, 14:317-327 (September, 1967).
- Mroz, G. J. and E. A. Yerman. <u>Reliability Apportionment and Optimization Techniques</u>. Technical Memorandum 31-024-43-ORA-10 Downey, California: Autonetics, 1963.
- 13. Nemhauser, George L. <u>Introduction to Dynamic Program-</u> ming. New York: John Wiley and Sons, Inc., 1966.
- 14. Proschan, Frank and T. A. Bray. "Optimum Redundancy Under Multiple Constraints." <u>Operations Research</u>, 13:800-814 (September-October 1965).
- Zelan, Marvin, editor. <u>Statistical Theory of Reliabil-</u> <u>ity</u>. Madison, Wisconsin: University of Wisconsin Press, 1963.

APPENDIX I

alle grad and states and the states of the second second

Lagrange Multiplier Method

This appendix contains a description of the program that was used for the Lagrange multiplier allocation method. The program was written using Fortran IV for the IBM

7044/7094 system.

Symbol	Description of Meaning
Input Variables	
RS	The required system reliability.
ML	Number of subsystems in the system. ML must be less than or equal to 20.
RC(1)	Reliability of the components used in the Ith subsystem.
CC(I)	Cost of the components used in the <u>I</u> th subsystem.
Program Variable	s
ZL	The Lagrange multiplier.
US	Required system unreliability.
UC(I)	Unreliability of components used in the Ith subsystem.
N(I) K(I)	Number of components allocated to the <u>I</u> th subsystem.
^N _K]	Matrix that contains the number of com- ponents of each subsystem.
XZL	Increment to find the Lagrange multiplier
	that gives a system reliability less than RS.
T.	Level of the Lagrange multiplier increment

Table V Symbols Used in the Lagrange Multiplier Method

Table V (Contd)

1.0.1

oliche de la compañía

्रात्रीय करण्डाः कारक्रम् विदियम्भित्रीक्षितिक्षितिक्षिति विद्याप्र । किन्द्रिया किन्द्रियो कि

Symbol	Description of Meaning		
Program Variable	:S		
DZ(L)	Lagrange multiplier increment.		
TZL	DZ(1) = XZL/10 DZ(2) = XZL/100. Lower limit of the Lagrange multiplier.		
Program Subrouti	nes		
BEST(ZL,K)	Program to calculate the optimal number of components for each subsystem, K, given the Lagrange multiplier, Z,		
URS (K)	Program to calculate the system unreliabil- ity, given K.		
COST(K)	Program to calculate the system cost given K.		
COSTS(I,J)	Cost of the <u>I</u> th subsystem when it consists of J components.		
R(I,J)	Reliability of the <u>I</u> th subsystem when it consists of J components.		

PROGRAM SUMMARY

After the input data is received, the program starts an iterative procedure (statements 7 through 12) to find a Lagrange multiplier that will give a system reliability less than required. This is done because the system reliability is a monotonic decreasing function of the Lagrange multiplier (Ref 4:406).

The next section (statements 15 through 17) define the limits to be searched and the Lagrange multiplier increment.

The third section compares the allocation for two

فالطحما المؤتقل مديره مواجدا متعاصينا والمراجع

different values of the Lagrange multiplier. Statements 25 through 29 check for one change in the allocation. If no changes occur, ZL is decreased and the search is made again. If a change does occur, the five statements after statement 30 search for a second change. If two changes have occurred, statements 40 through 50 reduce the interval and the machine is sent back to find the level where only one change has occurred. If two changes have not occurred, the machine prints out the result, then continues the search. stantin et stat bi**ktä**hensa (**Kii**le

000/JE/60-2

·	EUNG REXECUTE REPUNE	, 0 929200<u>0</u> 10 0 0	65-326	LT SPIVLY	AF11-52	
. 1	TOPTO SATA		• *	-	•	
- 	01 (56 00/2/08	210M (20), ACG (00, 110, 111)	20) +CC (20)	+UC(20)+DZ(5	()+X(3)+K(20)	-
• • • • • •	1 7540 (5.001) PS.11L 5.002 1 (RC(1)	•T=1•*L)		• • •	
	2 PEAD(5 HS =]	•203) (CC(I)•I= •∩= RS	=1•%L)	••••		
· · · • · • ·	5_00_6_1 6_000(1)	=],ML =].0 -RC(I)			ang	·· · ·
terana a ar a cora-	$\frac{7^{2} Z = 0}{11} = \Lambda$	Logiuszr Loat (Pa))/ALOGIU	(1))		
· · · · · · · · · · · · · · · · · · ·	- XZL = 7 7L = 7	(<u>)(()</u> +XZL L +XZL	vroe (nc (i)))/(CC(<u>1</u>)+(1	•0-UC(1)**(:1))
·	0 CALL 3 11 JF(1)05	EST (ZL+K) (K) +GE+ US) GG	TO 15	1999 - 1995 - 1996 - 1996 - 1996 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -		
	. <u>12_GC_TC</u>	۹				
		US STATEMENTS C LIERAZLA AND TH OTHE MENTALIES	ALCULATE	NT TO BE ADD	ED TC FOR.	
•	15_TZL = . L=1	ZL - XZL	· · · · · · · · · · · · · · · · · · ·	••••••••••••••••••••••••••••••••••••••		-
	X(L) =	1.0				-
to contact to a second second	GO TO	- •/**** 190	معدية معرفه المعرفة		 <u>-</u>	
<u> </u>	2^ 7L = 21 25 CALL FI	- PZ(L) "ST_(7L+K)		NOTR	EPRODUCIBLE	
•	26 15 (E)	1. ME.V(1))00	TO 20	, ·		
Marina andre anges para - an	27 IF(1.C	-"L) 60 TO 50	····			
	27 J = 1+ 27 GO TO 3	26			• • • • • • • • • • • • • • • • • • •	
	THE AL	OCATION.	ZY CHECK	FOR A COPIN		· · · ·
	30 1 = 1+.		00		the anti-contraction in a statistical card attraction of the state of	•
		.MF. K(I)) GO	TO 40	- • 、 .	. .	
		MTS AFTER 30 C	HECK FOR	N 2ND CHANGE	IN THE	
	$\underline{42} = \underline{42}$	· • · · · · · · · · · · · · · · · · · ·		**********		
	7L = ZL	L + 07(L-1)		u . 10	-	
	0Z(L) = 	= •1*0Z(L−1) 20			and and a substantian of the sub-	
	50 (F(ZL IF 2 C	+LE+ X(L)) GC ANGES OCCUR+ S	TO 52 TATECENTS	40 THEOUGH	SO REDUCE THE	
•	TALKS S TATERV COLTO	L Sherration and the 2	CHANGES OC			
•	52 IF (L 6 53 IF(ZL 6	GT• 1) L = L-1 LT• TZL) CALL	FXIT		-	
	<u></u>			• • • • • • • • • • • • • • • • • • •		

PTF./51/61-2 100 MPITE (6+204) 101 MPITE (6+205) (4(1)+1 = 1 •"L) 100 MPITE (6+205))____ - = COST (K) REL = 1.0 - URG(K)103 HEITE (6,207) ZL,C,REL (RITE (6,300) -----00 104 I = 1,FL C3 = COSTS (1+K(1)) $PT = R(T_{2} < \{T\})$ 104 MEITE (6,301) 1, C5, RT PO 105 I = 1,"L $105 \quad \Psi(1) = \zeta(1)$ SO TO 20 201 FORMAT (F12.6,13) . . . 202 FORMAT (SE10.8) 213 FORMAT (5510.0) 204 FORMATIONISX, 41HTHE NUMBER OF COMPONENTS IN EACH STUDE IS) 215 FORMAT (180,6X,5(13,4X)) 206 FORMAT (7X, FULANSDA, 10X, 4HCOST, 10X, 11HRELIAUILITY) ____207 FORMAT (5X; F10.6;4X;F10.0;5X;F10.6) 300 FOR AT (100,10X,57ETASE,6X,4HCOUT,4X,11HRELIAULLITY) 301 FORDAT (100+11X+13+2X+F9+0+4X+F10+6) FND -----TINELC CONT SHOPOHTINE (EST(2+K)) STUPRUSRAT TO FIRD THE OPTIMES BUDGER OF WEDDADANT 0 CO PONENTS. DIMENSION K(20) +UC(20)+ CC(20) CONTRA CO, UC, NL S(N+S+C)=ALOG(1+C-U**N)-Z*C*FLOAT(N) 1 00 10 T= 1+4L 2 PH=(NLOG(Z*CC(T))-/LOC(Z*CC(T)-/LOG(DCT)))/NLOC(LC(T)) 2 VI = 01 ····· = ?1, +1 1, TE ("L .LT. 1) FO TO B 5 TE (PC) - C(T) - C(T) - AT-H(RL-UC(T) - CC(T)) - GU TO -(1) = "L SC TO 10 8 ×(1) = 3日 10 CONTINUE PETURN ריח 34

·	
	and the second
· · · · ·	CONTRACTOR AND TOTAL SYSTEM COST
A 1	
·	$= \frac{1}{2} \left[\frac{1}{2}$
•	
•	$4 \cos(1) = \cos(1) + \cos(1) $
·····	oution:
	م به الم
······································	na su a su an anna an an an an an an anna anna
a an martin a di di di di	
	i de la companya de l
	4100TC 0100
	C SUMPROBAT TO FIND SYSTEM UNDELIABILITY
· · · ·	FUNCTION URS(N)
	D1 ENSION N(20) +UC(20) + CC(20)
·*********	CONTRICCOUCONA
	$1 - 25 = 1 \cdot 0$
	2.005 f = 1.001 k
	5 55 - DS#(1-0-1/C(1)##H(1))
	1125 = 1.0 - 0.5
•	n ting/

•	the second se
•	
	LINFIC SUM
	A BARGAN. TA ANLAULATE CUBSYSTER CUST
	FUNCTION COSTS(I)J
	D1. ENSION CC(20)+RC(20)+R(20)
	CONCORCORCORL
	COSTS = CC(T)*FLOAT(J)
	n na ser en
	TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
	PHUGRAN TE CALCULATE SUBSYSTER RELIACILITY
	FUNCTION R(1.1)
	INTERNATION COLON + UCLON
·····	
	Z = Y a Z = U X I J Z = U OFTUDAL
بالالالية المتحد المستنابي	
•	
•	
	and a second
•	
ay menen a marin mam	
	26
	50
·	والمعام معقفة معقد المنافع والمعادي المعادي المحاد المعادي المحاد المحاد المحاد المحاد المحاد المحاد المحاد الم

.

APPENDIX II

Mathematical Programming Method

This appendix contains a description of the program that was used for mathematical programming. The program was written in Fortran IV for the IBM 7044/7094 operating system.

> Table VI Symbols Used in Mathematical Programming

Symbol

Description of Meaning

Input Variables

RS	The required system reliability.
ML	The number of subsystems in the system.
RC(1)	Reliability of components in the Ith subsystem.
CC(I)	Cost of Components in the I subsystem.

Program Variables

N(I)	Number of components in the Ith subsystem.
N1(I)	N(I) + 1, $NI(J) = N(J)$ for all J unequal
	to I.
A(I)	Increase in reliability per unit cost by
	adding one component to the 1th subsystem.

Program Subroutines

RST(N)	System reliability given N(I) components
	at each stage.
R(I,J)	Reliability of the Jth subsystem then it
	consists of J components.
COST(N)	System cost given N(I) components at each
	stage.
C(I,J)	Cost of the <u>I</u> th subsystem when it contains
	J components.

PROGRAM SUMMARY

After the input data has been read, the five statements before statement 10 calculate the initial solution. The next section has the machine print out the present solution. The third section (statements 13 through 20) calculate A(I), the increase in reliability per unit cost for the <u>I</u>th subsystem. Statements 25 through 31 add one component to the subsystem for which the value of A(I) is the largest. Next, the machine is sent back to statement 12 to check if the present solution gives a system reliability as large as required. If it does, the results is printed out and the program ends. If it does not, the procedure is repeated until the solution does meet the reliability requirement.

. .

1	5.JO * * # YE 2117 //	* • 2 • 2 000 * • 305	60-13F	Fi UDINEM VII	7_7F
	* * * 3 ** * * * * * * * * *	·			
		T - D1 1.1 - 47 	nn: cht u T () sh(??) • 8	190 3 (200) • 10 (200)	
		"•\$€•"L			
	1 0530 (5.)	201) REAT	- 1 1/1 1		
• • • • •	n neth (1+	202) (PC(1)+1 203) (CC(1)+1	= 1, 1		
	с <u>т</u> ч рят.	95 = 0FS1RFD	SYLITE PU	LINGLITY. CL	두 대한 부분
				T () (T N) T () () ()	THE TRACE
1 	<pre>C (1) = (CC(1) = (</pre>	COT OF COMPON	くだい パンチョン (それてい リンデう)	TN GREEN	
	1- T =	=]•"".			
		MECHLATE INIT	TAL SOLUT	ICL.	
· · · · · ·	T = 1	-LT.C) 8(1) =	.ust]•/=~\\. ⊨1	(1))	
	(1) = i	:(T)	-		
	IN CONTINUE		a 1 • •		
· ·- ·	11 IF(201(0)	E •58•85) 60 1 (Min	0 100		
	cy = cost	(**)			
	10175 (6.	2741 20.07	• • • •		
		(205) (M(])+] Mit Dut to 540	= 1.4ML) M (22062852	•	
	13 PO 20 I =	1931 - 1931 - 1938 - 1938 - 1932		•	
	LOOP TO C	ALCULATE A(I)	•		ALE
	$Y_1(1) = 0$	(1) + 1 - T(21) =	AN/COUN	our cour	
	20 21 (1) = 1	(T)		CEPROL	
•	$T \Box^{*} U = \Lambda($	1)		of Re	
-	i = 1	2.14		H.	۶
	25 00 40 L =	6 2911L - 1855 - 1865-757 - 18	(!).		
-	TE (TESP	•CF. *(1)) GO	TG 31		
	TE"P = 4(1)			
	U = 1 no construction				
• ·	21 P(L) = P(L) + 1	•		
	STATT DO T	TO ADD CXE IN	ose co pro	ENT TO COUSYET	Ell (III)
-	. <u>the Fron</u>	ST 5(I).	•		
	100 90179 15.	2101		· • • · · • • • · · · • •	
	CUTPUT	«			
	20 = 317	(3)			
	CX = CCST	(V) CONTRIP	utvo tur i	omasten latt on stat	··· · · · · · · · · · · · · · · · · ·
	e excla conce	SMITTA CONTRA Veteva		an gran the Court	× 44 − € 54 − 6
	""IT" (/,	2111 (1.92	·····		<u> </u>
	MOITE (6.	212) - 10X - 65 COLT			
	217 FOR ALLE 211 FOR ALLE	•F1/*C•SX•F10	≥×91178261 •8}	······································	
	00 11º I	= 1,4L,2			
	TI=I+1 *				· / •-

20

*

.

C	<pre></pre>
	1 PCL (12+6+13) 2 format(PC: +R) 2 format(PC: +R) 4 roputt(PC: +4X+F)(+P+4X+F)(+) 4 roputt(PC: +4X+F)(+)
21 21 22	- μα μιντ (μμη κΑ(ΟΥ.ΤΥΙ)) - μα μιντ (μμη κΑ(ΟΥ.ΤΥΙ)) - ΕΥΙΔΙΑΤ (μμη ΚΥ.ΕΟΥ.ΤΥΙΟΥ.ΤΕΊΩΟΥ.ΙΝΗΡΕLΙΑΝΙΔΙΤΥΥΖΑΚΗΡΟΝΙΟΧΟΟ - ΕΥΔΙΑΤ (ΓΕΙΑ.Ε) - ΓΟΡΟΛΤΟ (ΓΕΙΑ.Ε) - ΓΝΟΡ - ΕΥΔ - ΕΥΔ
	e e e e e e e e e e e e e e e e e e e
	NOT REPRODUCIRIE
• •	
	IC SULT $P \approx CRATTO CALCULATE SYSTEM RELTABILITY.$ $P \approx CRATTO CALCULATE SYSTEM RELTABILITY.$
· · ·	
····· · · · · · · · · · · · · · · · ·	
<u>t 177</u>	C #14.2 PROGRAM TO CALCULATE FURSYOT SURRELT STLETTY. FURCTION D(1+J) DTURNATENCE(20) +DE(20) CONTON EC+PE+11
	$P = 1 \cdot 1 - (1 \cdot 1 - RC(1)) \times J$
•	
• -	
·••• · · · · · · · · · · · · · · · · ·	
	3)

.

•

······ - -.

.

4.2

NOT REPRODUCIBLE

-- - - - - -~ _ C - PROGRAM FOR CALCULATE SOUDY/TERE COUT. FUELTION COLOUR PRODUCY/TERE COUT. - -- -. PICTOLINU (CC)+RC(20)+RC(20) COMINE CC,RC+ME COMINE CC,RC+ME COMINE CC(1) R FLOAT(J) -----. - -DETUDN <u>– 195</u> TONTO · • •

>DITION PLINE PL . $\overline{}$ 77117N $\pm s_{\rm e}$ ·----

:

•

.

---- / ·- / · · · · ·

 σ (i.e.

and a second second

.....

• • • • •

. . . .

-

APPENDIX III

Dynamic Programming Method

This appendix contains the program description for the dynamic programming algorithm. The program was written in Fortran IV for use on the IBM 7044/7094 operating system.

ł

L. Post I.

ŝ

	Table VII
Symbols u	ised in the Dynamic Programming Method
Symbol	Description of Meaning
Input Variable	:s
RS ML RC(I) CC(I)	The required system reliability. Number of subsystems in the system. Reliability of the components used in the Ith subsystem. Cost of the components used in the Ith sub- system.
Program Variat	ples
U(I)	Unreliability of components used in the <u>I</u> th subsystem.
U(I,J) C(I,J)] M	Unreliability-cost sequence for the <u>I</u> th subsystem. Number of the subsystem that is being that is being combined with the combination of the M-1 previous subsystems.
YU(I)] YC(I)	Unreliability-cost sequence for the Mth subsystem.
XU(I) XC(I)	Unreliability-cost sequence for the combi- nation of the M-l previous subsystems.
AU(I) AC(I)	Dominating sequence for the combination of the Mth subsystem with the M-1 previous subsystems.
L(M,I)	Location of the two elements that were com- bined to give the Ith entry in the dominat- ing sequence for the M subsystem combination

Table VII (Contd) Symbol Description of Meaning Program Variables EUX ECX Candidate for the next entry in the dominating sequence. EUY 1 Candidate for the next entry in the dominat-ECY ing sequence. Program Subroutines OUTPUT(N) Subprogram to print out the dominating sequence, which is of length N, resulting from the combination of the Mth subsystem with the M-1 previous subsystems.

ANSWER(N) Subprogram to print out the final allocation given the dominating sequence of length N after all subsystems have been combined.

PROGRAM SUMMARY

After the data has been read, statements 11 through 20 calculate the base requirements for each subsystem. Statements 21 through 39 calculate the reliability-cost sequence for each subsystem and write out this information. Statements after 39, through 40 make a block transfer of the reliability-cost sequence for the first subsystem into the location (XU, XC) for the dominating sequence of the M-1 previous subsystems. Since M equals 2, this combination is the reliability-cost sequence of subsystem 1.

The next section (statements 50 through 55) is a block transfer to move the reliability-cost sequence of the <u>M</u>th subsystem into the location to be combined with the combination of the M-1 previous subsystems.

The following section (statements 60 through 110) find the dominating sequence, one element at a time, of the combination of the <u>Mth</u> subsystem and the M-1 previous subsystems and write out this sequence.

The last section makes a block transfer of the dominating sequence obtained in the previous section into the location to be combined with the next subsystem. It also increments M to the next subsystem and tests to see if all subsystems have been combined. If they have, the final allocation is printed out. If they have not, the machine is sent back to statement 50 to continue the procedure.

1117 Z 117 Z 5 - T

	• . • • •	0,8.4000 65.266 LT CRIVER AFT(-0)
	< 7 Y T ()	
	- LEON)	
	+105-6	5 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
		CONTRACTOR SCIE
		CONTRACT AND
		(2, 3) = (
	. –	$ \begin{array}{c} (10) \\ ($
	· 1	
	1	
	*	(2) (2) (2) (2) (2) (2) (2) (2) (2) (2)
		274V(5,203) (((()),1=1,4VL))
	- <u>-</u>	CCULY = COST OF COPONENTS USED IN SUSSYSTEM I.
	f	BC(I) = DEFIVATEITA OF CONDONEVIS DOED IN SUBSACES IN 1.
		$\mu = 1 \cdot 1 = 3 \cdot 1 = $
	S,	00 10 I= 1,1L
	1.0	$\cup C(\mathbf{I}) = 1 \cdot 0 + \mathbf{B}C(\mathbf{I})$
	11	DO(20) T = 1.00L
		K = 1
	12	Y='!((T) ***
	14	TF(115-X) 15,16,16
	15	
	~	LOOP TO CALCHEATE THE NUMBER OF COMPONENTS IN SACH
- • • •	~	CH SYSTEM RECHIRED TO MEET THE PASE RECHIREMENT.
	10	(1) A start st Start start st Start start st Start start st Start start st Start start st Start start st Start start st Start start st Start start st Start start sta
	10	
	2''	
	21	
	Ċ	LOOP TO CALCULATE THE RELIABILITY COST SEQUENCE FOR
	С	FACH SUBSYSTEN.
		Y = NL(I)
		C(1,1) = Y * CC(1)
		$DC_{35} = 2.10$
		$U(1,\mathbf{J}) = U(1,\mathbf{J}-1) * U(1)$
	35	C(1,J) = C(1,J-1) + CC(1)
		MT = 10
		60 39 F = 1,14
		WRITE (6,210) I
	C	LOOP TO BRINT OUT THE RELINGTLINY COST SEQUENCE FOR
	~ ·	FACH SUBTYSIEN.
		PO(35 1 = 1.10.7)
	•	
	~ ~	
	23	
		$X_{A}(1) = E(1, 1)$
	40	$\gamma_{\mathcal{C}(1)} = \gamma_{\mathcal{C}(1,1)}$
		·= 2 REPRODUCT
		VI = 10
	57	DO 55 1 = 1,10
		Y((i)) ≅ (((i ,• i))
	55	$MC(\mathbf{I}) = C(M,\mathbf{I})$
		1177 = 1577 [(567 (51 + 1 - 1)

1997/1997/1994-19

AT THIS TE CALCULATE THE UNRELINETETY OF CACH STATE TO THE ILL GUTTORT THE SYSTEM MERCHANISM. $Y(1) = X_0(1) + Y_0(1) - X_0(1) + Y_0(1)$ C $f((1) = \chi_{0}(1) + \chi_{0}(1)$ NTO FOR ALL TO REPORT ANTIAL CATENCES, INCLUDED INTERS TELLOND, FUN SCH STACT. U(1,1) = 1.1r = 1 NOT REPRODUCIBLE .! = 1 11 = 1 ある (1) = 次日(月月) THE # 244(19) For = 2.. (16) 1 (X = 5554560) 138 = 0.1500000 TTE TER ALS 71 THROUGH 80 EXAMILE ALL ASHS LIDS THAT OA CODE TO US TO FIND A CANDIDATE FOR THE MEAT BED EAT IN THE OCHNYTHE SEQUENCE. TO BY C = 1.91 **,** -1. (XU(K) . 1. 4J) 60 TO 01 20 80 61 = 1.1 (F. (YU(L1)+GF+ MJ) OU TO 80 $\mathbb{P}\left[1\right] = X^{-1}(X) + Y^{-1}(L1) - X^{-1}(Y) + Y^{-1}(L1)$ 10(711 .01.11) 50 TO RA $(c_{1}) = xc(c_{1}) + xc(c_{1})$ は長くのでは、ようてよりの人が、いび、手付いるり、 10((101+7)+504)+400 +(EC1+E0+ECX)) 60 TU 30 11X = 111 Sinx = E.C.D. LEX = 1 (0×K + L1 60.00071000 et contracts STATISTIC BE THREEGHESD EXAMPLE ALL COLD HE LEDG THERE (R FINAL TO I THE FILL A CANCIDATE FOR THE NEXT REFERENT IN 1 THE TOTAL SECRETOR. 50 01 H = 1,1 TE (X0(3)) .SF. EM) 60 TO 91 00 40 Ll = 1,10 FF (YU(L1) • # • 1.0) 60 TO 50 「りん」= スポイベ) + マワ(L主) ー スレ(K)ッイレ(L主) $[f \in J] = N f(K) + N f(LJ)$ JE(FCI +LT+ECY) (0 TO 90 HE CONVERSENTEMENTS (ECLERGELCY)) CONTRACTOR FCY = FC1 LEY = JITAN + LI AN CONTINUE AL CONTINUE STATESTERS OF THREE OF 160 SELFCE THE SEXT FRINT IN 5 THE DEPARTED STATEMENT, FUR ENDER LOX CALED. Th = [14+] INC LCC .LT. (CX) 00 TU 100 10 (Con . LIFERA) - The (LET (CARLER)) CO TO 100 NUCLINE - 25%

÷.

÷

100/5-/AP-1

NOT REPRODUCIBLE $TC(T_X) = T_X X$ $L(\rightarrow i \Box) = LEX$ Sec. 7 110 THE REPORT 11. SE(11) 4 FEY し(+1/1) = 147 77(1)(1)) (I), USY) (0 TO 120 11. stations to the terms is manned and TO INALINA STRUTTER FOR THE DUPPED FOR RELIAVIELING AN UNION ONT REAL STARLING. 1 = L(+14)/100 J = L(+IC) - 100×1 CC 76 69 . . 120 / YER OUTPUT (15) 51 = 11a 100 1241 = 1+NI XC(1) = 20(1) 12^{12} XC(1) = AC(1) PLACE TRANSFER TO SET XU(1) = AU(1) PRIOR TO US DIMESS 5 YE (1) RETERING THE REAL PROTECTION STRUCTURE. IF(+ +LI+"L) 3 / 7/ 56 TINT TO THE THE ALL STAGES HAVE FIELD COMMENTS. CALL ANDER (DI) 2 1 16+147 (712+6+ 13) 2 2 FOR AT (5F10.6) 214 ECREAT (SE10.0) 1.4X, JACCAT (, 5X.)) 211 FORMAR(148,2(19,7X,F)^,6+2X,F)^,6,6Y)) BUG FORMAT(10X,110 ETATEDENT ,18) 101 FORTAT(HX,2(TA,4X,F10,8,2X,F10,1,6X)) 0-71:04 ⊊ I.(1.) STRETC THAT SPERGRATINE GENERAL (*1) 1.1.1.1.1.1.1. (J(no.)) \$AC(500) \$L(25) \$99(c) Controle , AD, AT, L VETTE (A,205) 5711- (5+277) DF- 150 1 = 3,111,2 TT = 1+1 Z = 1 + - 70(1)Z1 = Jar = Ad(1;) >3110 (0.52 6) 1.52.5°C(1).5L(1.51).5L(-21.5°C(11).5L(-.51)) (135) CCMTREAL 206 FTX MT(1492(12+549042(+0912+54)+(964913972)) 207 Sear AI (AA, 2000L) CAI 108, 28, 1108, LIACILITY, D.S. 4000L 1, 28, 13366 XT LUCATION, JX)) C 1 1 1 1 1 1 CALC

21 - 1 - 1 St - 2

```
111 D. D.
                         19(39(1) .LS. (5) 30 TO 226
        2. CONTING
        2 · 4 · 1 = 1.
                         5- <del>-</del> 1
         227 J = L( ,1) /100
                          \mathcal{C} = \mathcal{L}(\mathcal{C}, \mathbf{I}) - \mathsf{J} + \mathbf{I}^{\circ} \mathcal{C}
                          PA(1) = 1.1 -U(1.4)
                          ^*(**) = C(**,∀)
                         1 = J
                         14 = 1-}
                          \gamma \wedge (1,1) = 1 \wedge \gamma - U(1,1)
                          r^{*}(1) = r(1, 0)
                         979 [TF (3,301)
                         7 = 1 • - 所(税)
                         11 = 1 + 1
         240 AITE (A.S.S) I, 26(1), CA(1), II, PA (II), CA(1), CA(1), A (II), CA (II
         WAY E DENAT(10) . X. BRICSYSTER OFLIGE (LITES X. 1000 TOUR CONF.)
         9 (1 000001() 4X (E10 .8 .] (Y (E10 . ))
         SAN TER ATELE 4X+2(DEREVATER, >X, 110KEL1/DILLIY, 6X, 4000ST (DX))
         -

= = = = (2(7X, 1), 7X, 1 10, 8, 3X, F10, (2X))
                         977 - K. H
                            7 ···>
• T -
```

1

. • * .-

NOT REPRODUCIBLE

...

1.1

APPENDIX IV

the offer have not be

<u>...</u>

Random Number Generator

This appendix describes the program that was used to generate the component cost and reliability for each subsystem. The program was written in the Forgo version of Fortran II. In the particular compiler used, the arctangent subroutine had been replaced with a random number generator. This program was written for the IEM 1620.

Table VIII Symbols Used in Random Number Generator

Symbol	Meaning	
R(I) C(I) X	Component reliability. Component cost. Temporary location.	

(*** /-* /6)+,

•

.

į

ε

	<pre>A 715 246K SF THES FLOODA THES FLOODA DECEMPTHES R(NO 10 T = 1) X = DAT(0.000</pre>	21VEN TREETS 2 NEAR AUBLE 2 ALEXANNA AUB	NG GULR SC 9 Pandon Tu 9 Blitheig A 2ARD:	URCE SHERS UTTHERE •1 ARY 1000•0•	e Mary Talak
č	· * = (X→ • =) · A, · X = X + • • 5	••••			
<u>.</u>	P(I) = X			·· <u>-</u>	
↓ *.	-CONTINGE -PO-29-1 = 1,	20			
21	C(I) = 1000. PUNCH 201.(2	* RAG((0+0) (J)• J = 1•28	· · · ···· · · · · · · · ·	· • · • • • • • • • •	
	PRIMCH 202. (C	(J), J = 1,20	·)		
.	EOD.VI(eLJU	₽) ●C)			
			,		
		· · · · · ·	··· ·		
				· · ·	
			··· •		· .
- .	· · ·	- •··			
		• .	· -		
			· · ·		
· -·	• • •	· · · · · ·	NOT RE	PRODUCIBLE	
		• -			
		···· · ·	· · · · ·		
· · •· · • · · · ·	· · · · ·	••••••••••••••••••••••••••••••••••••••	·-· .	• • • • • • •	
		•			
· · ····		· · · · · · · · · · · · · · · · · · ·		·· · · · · · · · · · · ·	
	•	· · · .			
		· · · ·	· · · ·	· ······	
	• • ••• • • •	· · · · · · · · · · · · · · · · · · ·			
		· .	• • ·		
· .		· · · · · · · · · · · · · · · · · · ·			
				• • • • • • • • • • • • • • • • • • •	• • • •
			43		

station in the

-

•••••••••

1. da 1.

• •

VITA

William Zachary Spivey was born on 30 March 1945 in Ruston, Louisiana. He was graduated from Delhi High School in Delhi, Louisiana in 1963. He attended Louisiana Polytechnic Institute and upon graduation in 1968, he was awarded a Bachelor of Science in Mechanical Engineering and a commission in the United States Air Force.

Permanent Address: Rt 2, Box 56

Ruston, Louisiana 71270

This thesis was typed by Bobbie Thompson

Unclassified Security Classification				
DOCUM	ENT CONTROL DATA -	R&D	overall tenast is classified)	
(Security classification of fifte, body of adatter ORIGINA TING ACTIVITY (Corporate author)	and incraing annoiation must c	2. REPORT 3	ECURITY CLASSIFICATION	
Air Force Institute of Technology (AFIT-SE) Wright-Patterson AFB, Ohio 45433		Unclas	sified.	
REPORT TITLE				
Comparison of Reliability Allocation	ion Optimization Me	thods		
4. DESCRIPTIVE NGTES (Type of report end inclusive de	tes)			
AFIT THASIS 5. AUTHORISI (First name, middle Initial, last name)				
William Z. Spivey 2/Lt USAF				
REPORT DATE	70. TOTAL NO	OF PAGES	TE. NO. OF REFS	
December 1969	Da. ORIGINATO	A'S REPORT NUN	15 IBER(\$)	
D. PROJECT NO.	GRE/EE	GRE/EE/69-2		
c. N/A	96. OTHER RE this report)	95. OTHER REPORT NO(5) (Any other numbers that may be assign this report)		
d.				
approval of the Dean, School of Eng <u>Iright-Patterson Air Force Base, Ol</u> II SUPPLEMENTARY NOTES	gineering, Air Forc	e Institute	of Technology (AFIT-SE	
approval of the Dean, School of Eng	gineering, Air Forc	e Institute	of Technology (AFIT-SE	
Approval of the Dean, School of Englight Patterson Air Force Base, Of The reliability allocation pro- than the classical optimization met three allocation models that are ba- It was found that dynamic pro- allocation problem. The model base was within 0.5% of the optimal solu- brogramming algorithm gave a solut:	roblem has previous thods. This thesis ased on optimizatio ogramming gives the ed on Lagrange mult ation given by dyna ion that was within	e Institute Iy been sol brings tog n methods. optimal so ipliers gav mic program 0.6% of th	ved by methods other ether and compares (U) lution to the e a solution that ming. The linear s optimal solution. (

, , ,

¥

•. 22

.

			- D			
KEY WORDS		* *		K B	BOLE	K C
······································						
Lagrange multipliers					i	
Dynamic programming					1	
Reliability allocation			Ì			
Mathematical programming			1		i	
Waitematers high manne	1				Ì	
		1	1			
		}	}			
		Į	ļ	l		ļ
				, 1	1	[
		}	{	1	1	1
						[
		E I	}	' i	į	1
		1	1		1	
		1		ł		1
		1	i	1	1	1
		1		1	:	İ
		i			}	
		1	1	ł	1	
		1	}		1	
	.	ł	1	l ł		ł
]]			
			ć I			
					Ì	
		1			Ì	
	1		ļ			
			Į	l l		
			1	1	}	
		1		}	}	
			ļ		1	ĺ
	1	i	1	1		
	{	1	ł			
	l	i		1		
		:				
		i				
]				
		1	l I			
		1	ł		1	
		1	ł			
]	}		1	
		1	ł			
	ļ		l			
			ł			
	-					

erikiteraanna biloo saaanda olaabii

• .

Unclassified Security Classification