INFORMAL REPORT

RELIABILITY IN THE DESIGN PHASE

ALI SYYED SIAHPUSH
STEVEN W. HILLS
HOANG PHAM
DEBU MAJUMDAR
Informal Technical Report

RELIABILITY IN THE DESIGN PHASE

Ali S. Siahpush
Steven W. Hills
Hoang Pham
Debu Majumdar

Published October 1991

Idaho National Engineering Laboratory
EG&G Idaho, Inc.
Idaho Falls, ID 83415

Prepared for the
U.S. Department of Energy
through the
EG&G Idaho LDRD Program
Under DOE Field Office, Idaho
Contract DE-AC07-76ID01570

* EG&G Idaho, Inc.
# DOE-ID

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED
ABSTRACT

A study was performed to determine the common methods and tools that are available to calculate or predict a system's reliability. A literature review and software survey are included. The desired product of this developmental work is a tool for the system designer to use in the early design phase so that the final design will achieve the desired system reliability without lengthy testing and rework. Three computer programs were written which provide the first attempt at fulfilling this need. The programs are described and a case study is presented for each one. This is a continuing effort which will be furthered in FY-1992.
# TABLE OF CONTENTS

**LIST OF TABLES**

**LIST OF FIGURES**

**NOMENCLATURE**

## 1.0 INTRODUCTION

## 2.0 LITERATURE REVIEW

2.1 Cost-Reliability Data Very Important in designing Optimum System

2.2 Catalog of Optimization Methods
   2.2.1 Optimum Reliability allocation for a Series System
   2.2.2 Optimum Redundancy Allocation for Series-Parallel System
   2.2.3 Minimize Total Cost of a System Subject to a Constraint
   2.2.4 System Reliability Maximization for a Non-Series-Parallel System

2.3 Various Algorithms Exist for Reliability Allocations
   2.3.1 Equal Share of Reliability by All Subsystems
   2.3.2 The ARINC Method
   2.3.3 Minimum Effort Method

2.4 An Example of a Heuristic Method

2.5 Review of Several reliability Software Packages

## 3.0 ENGINEERING RELIABILITY IN DESIGN PHASE

3.1 Brute Force Method
3.2 Subsystem Elimination Logic
3.3 Effort Minimization Method
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS (cont)</td>
<td></td>
</tr>
<tr>
<td>4.0 A CASE STUDY</td>
<td></td>
</tr>
<tr>
<td>4.1 Designing with ERDP-BF</td>
<td>50</td>
</tr>
<tr>
<td>4.2 Designing with ERDP-SEL</td>
<td>57</td>
</tr>
<tr>
<td>4.3 Designing with ERDP-EMM</td>
<td>66</td>
</tr>
<tr>
<td>5.0 CONCLUSION AND RECOMMENDATION</td>
<td>69</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>71</td>
</tr>
<tr>
<td>APPENDICES</td>
<td></td>
</tr>
<tr>
<td>Appendix A. Glossary of reliability modeling terminology</td>
<td>75</td>
</tr>
<tr>
<td>Appendix B. ERDP-EMM output</td>
<td>82</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.1 System with eight subsystems for BF/SEL example</td>
<td>40</td>
</tr>
<tr>
<td>3.3.2 Results of BF/SEL comparison for eight subsystems</td>
<td>41</td>
</tr>
<tr>
<td>4.1 Reliability values for the pump system</td>
<td>52</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Reliability evaluation process in engineering design</td>
<td>3</td>
</tr>
<tr>
<td>2.2.1 A system with N subsystems in series</td>
<td>5</td>
</tr>
<tr>
<td>3.1 A typical system with five subsystems</td>
<td>22</td>
</tr>
<tr>
<td>3.1.1 Subsystem reliability as a function of redundancy</td>
<td>24</td>
</tr>
<tr>
<td>3.1.2 Sample input screen for ERDP-BF</td>
<td>26</td>
</tr>
<tr>
<td>3.1.3 Sample output screen for ERDP-BF</td>
<td>27</td>
</tr>
<tr>
<td>3.1.4 System configurations for controller/actuator example</td>
<td>28</td>
</tr>
<tr>
<td>3.1.5 Reliability vs. cost for ERDP-BF controller/actuator example</td>
<td>30</td>
</tr>
<tr>
<td>3.2.1 Sample input screen for ERDP-SEL</td>
<td>34</td>
</tr>
<tr>
<td>3.2.2 Sample output screen for ERDP-SEL</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3 Subsystem configurations for controller/actuator example</td>
<td>36</td>
</tr>
<tr>
<td>3.2.4 Reliability vs. cost for ERDP-SEL controller/actuator example</td>
<td>38</td>
</tr>
<tr>
<td>3.3.1 Aircraft electronics system diagram</td>
<td>44</td>
</tr>
<tr>
<td>3.3.2 Conservative solution</td>
<td>46</td>
</tr>
<tr>
<td>3.3.3 Flow Diagram of the ERDP-EMM computer code</td>
<td>48</td>
</tr>
<tr>
<td>3.3.4 One redundancy of subsystem one</td>
<td>49</td>
</tr>
<tr>
<td>3.3.5 One redundancy of subsystem two</td>
<td>49</td>
</tr>
<tr>
<td>4.1.a Diagram of the original system</td>
<td>53</td>
</tr>
<tr>
<td>4.1.b Photograph of the Isokinetic Pump System and Backup System</td>
<td>54</td>
</tr>
<tr>
<td>4.2 Original pump system</td>
<td>55</td>
</tr>
<tr>
<td>4.3 Maximum available pump system</td>
<td>55</td>
</tr>
</tbody>
</table>
LIST OF FIGURES (cont)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Parallel original system</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Results of ERDP-BF using only standard components</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Results of ERDP-BF using only optional components</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Results of ERDP-BF using both standard and optional components</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Results of ERDP-BF when the desired reliability is increased to 0.969</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Results of ERDP-SEL using only standard components</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Results of ERDP-SEL using only optional components</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Results of ERDP-SEL using both standard and optional components</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Conservative solution for the original system</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Redundancy of two original components</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Conservative solution for the max. system</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Two options for max. reliability system</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>$b$</td>
<td>Total amount of the resources available for the whole system</td>
</tr>
<tr>
<td>$C_j$</td>
<td>Cost of one component at stage $j$</td>
</tr>
<tr>
<td>$C_s$</td>
<td>Total cost of a system</td>
</tr>
<tr>
<td>$g_{ij}$</td>
<td>Resource $i$ (such as cost, volume, mass, etc) used at stage $j$</td>
</tr>
<tr>
<td>$G(R_j, R_j')$</td>
<td>Effort function: Amount of effort needed to increase the reliability of the $j$th subsystem from its reliability of $R_j$ to $R_j'$.</td>
</tr>
<tr>
<td>$R_j$</td>
<td>Average subsystem reliability</td>
</tr>
<tr>
<td>$R_j'$</td>
<td>New average subsystem reliability</td>
</tr>
<tr>
<td>$R_D$</td>
<td>Desired system reliability</td>
</tr>
<tr>
<td>$R_D'$</td>
<td>Effective desired reliability after the reliability contribution of the previously eliminated subsystem is taken into account.</td>
</tr>
<tr>
<td>$R_j$</td>
<td>Reliability at stage $j$</td>
</tr>
<tr>
<td>$R_j'$</td>
<td>Assigned subsystem, $j$, new reliability</td>
</tr>
<tr>
<td>$R_k$</td>
<td>Product of the reliabilities of the subsystems that were eliminated</td>
</tr>
<tr>
<td>$R_s$</td>
<td>System reliability</td>
</tr>
</tbody>
</table>
### NOMENCLATURE (cont)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$W$</td>
<td>Total weight of a subsystem</td>
</tr>
<tr>
<td>$W_{ij}$</td>
<td>Weight of one component at stage $j$</td>
</tr>
<tr>
<td>$X_j$</td>
<td>Number of components in parallel at stage $j$</td>
</tr>
</tbody>
</table>

#### Greek letters

| $\lambda_j$ | Failure rate of the $j^{th}$ component/subsystem |

#### Subscripts

| i | resource |
| j | stage |
| k | Number of subsystems at stage $j$ |
| N | Total number of stages |
| s | system |

#### Superscript

| * | Quantities that are updated each time a subsystem is eliminated |
1.0 INTRODUCTION

Design engineers have always, to their best ability, designed their product to be reliable. However, design engineers must also work within given constraints. These constraints can be budget, volume, mass, and other requirements imposed by the customer. The designer's goal is to stay within the constraints as close as possible.

Traditionally, when the design is completed, the reliability engineer analyzes the system. If the system's reliability does not meet the requirements, the system has to be redesigned. The cost of redesign is high and involves several support groups (drafting, vendors, design reviewers, etc.) in addition to the design engineer. Redesign may have several iterations to obtain the final desired reliability.

In some cases (e.g., limited design budget), the engineer intends to meet the reliability requirements in the prototype phase of the project. The prototype system would then undergo extensive testing and, based on the data generated, the reliability would be determined. Unsatisfactory results would send the designer back to the drawing board and cost might increase many times before good results are obtained. Therefore, the earlier the deficiencies are identified and corrected, the more capable and inexpensive the end product would be.

Over the years, various techniques have been used to optimize reliability in the design phase. These methods are usually labor-intensive, require specialized skills, and are not applicable to design factors other than reliability. A simple and cost effective engineering tool is needed to optimize reliability in parallel with other major design factors.
There are a few major attributes that would maximize the use of this tool. It should be amenable to use early in the design process. It should be amenable to identify potential faults and weaknesses in a design during the critical planning and development phase. The designer could then create alternatives and implement changes before costly tooling and production. If the tool is a computer program, it should be user friendly, simple to operate, flexible, and require very little training. This tool should also produce technically reliable results.

To incorporate reliability into a design phase we have developed the approach illustrated by the flow chart in Figure 1.1. This approach recommends making the reliability analysis (process) a design responsibility with analytical tools appropriate to the level of design. The designer is given a reliability requirement, and it is his or her responsibility to incorporate reliability into the early design phase.

This report contains our ideas and concepts towards providing this engineering tool to help designers build reliability into the initial design phase of a project. This report is organized into three sections:

(1) State of the Art Literature Review - A review of previous work on cost versus reliability, optimization methods and algorithms for reliability allocation, and reliability software availability.

(2) Engineering Reliability in Design Phase - A description of three computer programs we have developed to optimized reliability based on a requirements of a design.

(3) A Study Case - The three computer codes were tested and an existing INEL system was analyzed to gain insight into how implementation of reliability might be achieved early in the design phase.

Appendix A contains some background information on reliability and a glossary of reliability modeling terminology.
Figure 1.1. Reliability Evaluation Process in engineering Design
2.0 LITERATURE REVIEW

The idea that reliability should be considered at the very beginning of a design is not new. However, it was found that suitable simple tools that engineers could use at the initial design phase do not exist. A good description of a reliability process to follow for an engineering system from design to production has been described in the 1976 Reliability Design Handbook [1]. Reliability prediction techniques for application during the early conceptual design phases of a system development have also been developed by the defense industries. One such work was performed by James et al. [2] of Hughes Aircraft Company (1974), which predicts the reliability of the design based on design specifications, detailed parts summaries, handbook predictions from existing systems and statistical linear regression analysis. Their main objective was to become capable of predicting the failure rate for the system.

Two classes of work have generally been done that are relevant to the problem addressed in this report. In one class, the researchers have discussed ways to optimize the design of a system from a reliability point of view. In the other class of work, techniques and computer programs have been developed which can calculate component failure rates and hence ultimately the system failure rate, or the reliability, of the design. We briefly describe these two separately.

In the last two decades several optimization techniques have been developed to improve the reliability of a system. These include methods to reduce the complexity of a system, increase the reliability of its components, use redundant components and structures, and practice a planned maintenance and repair schedule [3]. The problem considered in the present work addresses only two basic means of increasing the reliability of an initial design, namely to use more reliable components and to use redundancy.
2.1 Cost-Reliability Data Very Important in Designing Optimum System

The use of highly reliable components has one large drawback - the cost of a component increases tremendously with its reliability. The cost of a more reliable component is often much more expensive than that of a two component parallel system. As an example, two identical components with reliability of 0.9 for each, can be combined in parallel to produce a system reliability of 0.99. The total cost of these two components would be less than manufacturing one component with reliability of 0.99. Misra and Ljubojevic [4] studied a method for optimal reliability design of a system with redundant components where the cost of the components varied exponentially with reliability. They used a Lagrange function for optimization. The authors stressed that knowledge of the cost versus reliability data of the components are crucial for designing an optimum system, and once the cost-reliability function for the components are known, the design problem reduces to a mixed-integer programming problem, and can be solved.

2.2 Catalog of Optimization Methods

An exhaustive catalog of work done by various authors up to 1977 on the subject of optimization techniques for system reliability with redundancy has been compiled by Tillman, Hwang and Kuo [3]. This catalog includes references for the following major category of problems.

2.2.1 Optimum Reliability Allocation for a Series System (Figure 2.2.1): In this case the functional operation depends on the proper operation of all the components.

\[ \text{Figure 2.2.1 } \text{A System with N Subsystems in series.} \]
The problem is formulated as follows: Maximize the system reliability 
\[ R_s = \prod_{j=1}^{N} R_j \] subject to \[ \sum_{j=1}^{N} g_{ij}(R_j) \leq b_i \] where, \( R_j \) is the reliability at stage \( j \), and \( g_{ij} \) is the resource \( i \) (such as cost, space, mass, etc.) used at stage \( j \), and \( b_i \) is the total amount of the resource available for the whole system.

2.2.2 Optimum Redundancy Allocation for Series-Parallel and Parallel-Series Systems with Maximization of System Reliability Subject to Cost Constraints: A general formulation of the problem is:
Maximize \( R_s = f(R_i(X_j)) \), subject to \[ \sum_{j=1}^{N} g_{ij}(X_j) \leq b_i \]. Here \( X_j \) is the number of components at stage \( j \).

2.2.3 Minimize Total Cost of the System, \( C_s \) subject to a System Reliability Constraint: Mathematically the problem can be stated as Minimize
\[ C_s = \sum_{j=1}^{N} C_j(X_j) \] subject to \( R_s \geq \) Required System reliability of \( R_0 \)

2.2.4 System Reliability Maximization for a Non Series-Parallel System: (Determine reliability allocation and redundancy allocation.) The problems stated above are nonlinear integer programming problems. Many techniques and algorithms have been explored by different authors. The different techniques employed include integer programming, dynamic programming, linear programming, geometric programming, sequential unconstrained minimization technique, modified sequential simplex pattern search, Lagrange multipliers and the Kuhn-Tucker conditions, Generalized Lagrangian function, heuristic approaches, parametric approaches, etc. Reference 3 contains references to these works and more.

2.3 Various Algorithms Exist for Reliability Allocations
The function f mentioned above is a well known function for series and parallel systems. The function is complex for other system configurations (such as a bridge network system). In the present work we address only those systems which can be reduced to a series configuration. Then we address the problem of reliability allocation among the subsystems in order to achieve a certain reliability goal.

The reliability allocation, even for a simple series system is not easy, and generally requires a compromise among several objectives. Often the designer must consider reliability with other factors such as weight, cost, space, etc. Kapur and Lamberson have given a brief exposure to reliability allocation techniques in their book [5]. Some of these methods are described below for a series system.

2.3.1 Equal Share of Reliability by All Subsystems: The easiest solution is to assign equal reliability to each of the N components that make the total system. In that case, each subsystem must have a reliability equal to the Nth root of the desired system reliability.

2.3.2 The ARINC Method: A sophistication of this technique would be to allocate a reliability to a component based on its failure rate data. In this method, known as the ARINC method, a failure vulnerability is determined for each subsystem from the existing data or knowledge of the subsystem. This could simply be a ratio of the failure rate of the subsystem to the total failure rate of the system. Then a reliability is assigned to each subsystem such that its new failure rate is the failure vulnerability fraction of the desired total failure rate of the system.

Let the component failure rate, \( \lambda_j \), of a component j at time t be given by \( R_j = \exp(-\lambda_j t) \). The reliability of the system is then given by \( R_s = \exp(-\lambda t) \), where \( \lambda = \sum \lambda_j \). Let the desired reliability be given by \( R_d = \exp(-\lambda^* t) \). In this method, each subsystem j is assigned a new
reliability $R_j'$ such that $R_j' = \exp(-\lambda_j^* t)$, where $\lambda_j^* = \left(\frac{\lambda_j}{\lambda}\right) \lambda^*$.

This method compensates for the less reliable components by enhancing its reliability more in comparison to the others. This method can be made further sophisticated by utilizing the component or subsystem complexity and the relationship between component and system failure, as is done in the AGREE allocation method [5].

2.3.3 Minimum Effort Method: An easier and relatively simple method is the Effort Minimization Algorithm. Let us consider a function $G(R_i, R_i')$, which is called the effort function, and is defined to be the amount of effort needed to increase the reliability of the $j$th subsystem from its reliability of $R_j$ to $R_j'$. The following characteristics can be assumed for $G$. For a given initial $R_j$, more effort is needed to increase $R_j'$ to $R_j' + DR_j^*$. That is, $G(x, y + Dy) > G(x, y)$.

Similarly, it will take less effort to improve a component reliability from a subsystem which already has a higher initial reliability. That is $G(x + Dx, y) \leq G(x, y)$.

It is also assumed that effort needed to improve reliability in an incremental fashion is additive. For a series system the problem then reduces to minimization of the total effort, namely

$$\text{Minimize } \sum_{j=1}^{N} G(R_j, R_j^*) \text{ subject to } \prod_{j=1}^{N} R_j \geq R_D \text{ (desired system reliability)}.$$

With reasonable assumptions about the effort function, it can be shown that the above problem has a unique solution:

$$R_i' = \bar{R}_0 \text{ for } i \leq k$$
$$R_j \text{ for } i > k,$$
where, the subsystems have been rearranged with increasing reliabilities. The number $k$ is the maximum value of $j$ such that

$$R_j < \left[ \frac{R_D}{\prod_{j\neq j+1} R_{j+1}} \right]^{\frac{1}{2}}, \quad J=1, 2, \ldots, N$$

and $R_{N+1}=1$ by definition. $R_0$ can be obtained from the relation

$$R_D = (R_0)^{\frac{1}{k}} \prod_{j=k+1}^{N} R_j$$

Intuitively, for the subsystems numbered $k+1, k+2, \ldots, k+N$, the reliabilities are acceptable and are not changed. The reliabilities of the $k$ subsystems, whose initial reliabilities are considered low, are changed to a higher reliability value, same for all of them (namely $R_0$). An iteration scheme is required here to determine the number of subsystems, $k$, whose reliabilities are to be enhanced. The equally apportioned reliability, $R_0$, for these subsystems can then be easily computed from the desired reliability and the reliabilities of those that remain unchanged.

This method will be utilized as an algorithm in our development of a simple computer program for design optimization. In this method, however, we assumed that the same effort function $G$ is applicable to all components. In practice, there will be different $G_i$ for different subsystems, and the problem would be to minimize the sum of all these $G_i$'s. This becomes more complicated, but can be solved with dynamic programming [6].

2.4 An Example of a Heuristic Method
Several authors have developed heuristic methods to arrive at the optimum system reliability [7]. An elaborate method proposed by Tillman, Hwang and Kuo [8] that uses a combined heuristic and a pattern search technique is described below because it shows how a computer program can be developed rather well to solve these types of optimization problems. The problem addressed is an usual constrained reliability optimization problem, but here both the optimal component reliability and the optimum number of redundancies are to be determined in order to achieve the best overall system reliability. A similar problem was discussed in reference 4. These are mixed integer nonlinear programming problems.

Consider an N-stage parallel-series system, where both the component reliability, $R_j$, and the number of components $X_j$ at the stage $j$ are to be determined subject to certain constraints. The system reliability is given by

$$R_s = f(R_1, X_1; R_2, X_2; \ldots; R_N, X_N) \quad (2.4.1)$$

The constraint equations can be written in a general form

$$\sum_{j=1}^{N} G_{ij}(R_j, X_j) \leq b_i \quad (2.4.2)$$

For simplicity, we consider the case where there are limits on the system design for total weight $W$ and total cost $C$ such that

$$\sum_{j=1}^{N} g_{ij}(X_j, R_j) = \sum_{j=1}^{N} W_{ij}X_j \leq W \quad (2.4.3)$$

and
Here $W_j$ and $c_j$ are the weight and the cost for one component at the stage $j$. In this case $b_1=W$ and $b_2=C$. Simple expressions for $C_j(R_j)$ such as $C_j=(A+B)R_j$ or $C_j=E \exp(D \cdot R_j)$ can be assumed, which include the important fact that more reliable components will cost more.

The problem now is to determine $(R_1, X_1; R_2, X_2; \ldots, R_N, X_N)$ such that Equation (2.4.1) will be maximized subject to Eqs. (2.4.3) and (2.4.4). Most optimization problems are of this nature.

In Tillman et al.'s method, a pattern search technique (sequential search routine) is used for $(R_1, R_2, \ldots, R_N)$ until a maximum system reliability is reached. A heuristic approach is then applied to each value of $(R_1, R_2, \ldots, R_N)$ to obtain the optimal number of redundancies $(X_1, X_2, \ldots, X_N)$, which maximizes the system reliability while satisfying the nonlinear constraints. Their heuristic approach is based on the concept that a component would be added to the stage where its addition produces the greatest ratio, $F_j$, of "increment increases in reliability" to the "product of decrement in slacks."

To explain their method further, we describe below their computation procedure:

1. Assume an initial set of $R=(R_1, R_2, \ldots, R_N)$.
2. Substitute this $R$ into equations (2.4.1) and (2.4.4). Assume $X=(1, 1 \ldots, 1, 1)$
3. Calculate $F_j$, the heuristic ratio, for all $j$ using their algorithm for $F$. Select the stage that produces the maximum $F$. This stage is then considered to have an additional component.
4. Add one component to this stage. Check if constraints (2.4.3) and (2.4.4) are violated. If it is a feasible solution, then modify the new $X_j$, and repeat step 3.
Note that, if at least one constraint is exactly satisfied, the current $X$ set is an optimal solution corresponding to the $R$. Also, if at least one constraint is violated, cancel the proposed addition of the redundant component; remove that stage from further consideration. The current $X$ is the optimal solution with respect to the $R$. The authors have successfully tested the program with an IBM 370/158 computer.

The main point of discussing this method is to give an example and the idea that a computer program can be developed with heuristic iteration schemes to find a solution for an optimum system. Other examples can be found in references [9] and [10].

2.5 Review of Reliability Software Packages

Many computer programs have been developed which aid in the study or prediction of the reliability of a system. These have originated because of the fact that there is a demand everywhere to produce a highly reliable product and to remain competitive in the marketplace. In the following paragraphs we discuss several software packages, most of which are commercially available.

(1) J. B. Bowles, et al., of the Center for Machine Intelligence at the University of South Carolina made a comparison of six commercially available programs that they thought were useful to predict the reliability of electronic systems [11]. The following packages were included in their study:

(a) The Failure RATE (FRATE) prediction program [12] was originally developed by the British Telecom Research Laboratories for their own internal use. FRATE provides designers of electronic equipment with a common basis for system reliability prediction from the early design phase onward. The prediction is based on the HRD4 model from the British Telecom Handbook of Reliability Data [13]. It supports both parts count and parts stress reliability analysis and allows the inclusion of mechanical
components as well as electrical components. FRATE runs on IBM PC-XT, AT or compatible computers. The input interface can be either menu driven or command line. The database includes failure rates for most common electronic parts, including discrete components and integrated circuits. The user can enter information for components that are not in the existing database. The FRATE model gives a prediction of the FITS (Failures/10^6 hours) and the MTTF (Mean Time To Failure) for the system.

(b) The Military Reliability Stress (MilStress) analysis program [14] was developed to give an up-to-date picture of the reliability of an electronic product from the early design stages to a fully operational system. It is based on the models found in the MIL-HDBK-217E reliability standard [15]. MilStress is a menu driven program that uses a system of displays to lead the user through the procedure for defining the hierarchy and component blocks of the system under study. The program contains a large library of electrical components plus 1000 standard mechanical parts with default reliability data. MilStress has several output options, including a print out of the total system in tree structure form and a listing of all the components in the system, showing each component’s contribution to the overall system failure rate. MilStress runs on IBM PC-XT, AT or compatible computers or on DEC VAX computers. Another similar program, BellStress [16], is available from the same vendor and does the same reliability calculations but it is based on the Belcore Reliability Prediction Procedure [17] instead of the military model.

(c) PC Predictor [18] is a program that permits the designer to evaluate assemblies or equipment reliability using the stress analysis method of MIL-HDBK-217. There are two versions of the program. One uses the component failure models from MIL-HDBK-217D/Notice 1 and the other uses the MIL-HDBK-217E model. With the exception of the variations of input and output formats and
the fact that it will not support any mechanical components, the PC Predictor program gives results similar to the MilStress program for electronic systems.

(d) REAPmate [19] is designed to perform reliability predictions on electronic and electro-mechanical components using the algorithms and techniques found in MIL-HDBK-217. REAPmate requires very little information when entering component data because generic component parameters such as the packaging type and technology used are contained in the component library. One feature of this package is that it can be run on a workstation as well as a personal computer. An additional advantage of REAPmate is that it will generate graphical output such as a plot of failure rate of the system as a function of ambient temperature.

(e) ReCalc2 [20] is another program based on the MIL-HDBK-217 models and data. It automates the parts stress reliability prediction procedures of MIL-HDBK-217E and can provide reliability predictions for any design from a single circuit board to very large multi-circuit board systems. By using the integrated system functions, the user can define a modular, hierarchical system model, including redundancy.

(f) RELEX [21] is a complete MIL-HDBK-217 parts stress analysis reliability prediction package, supporting both MIL-HDBK-217D/Notice 1 and MIL-HDBK-217E. This program is another one that has graphical output options. An additional feature that makes this program useful is that the output reports can be sorted by various categories.

(2) For mechanical parts reliability analysis, the Design Reliability (DEREL) and the Design Reliability Optimization (DERELOP) programs [22] are being developed at the University of Sherbrooke, Canada. DEREL uses the design governing stress equation and the failure governing strength equation to calculate a design reliability number. Material strength
data and design parameter data are input in the form of individual points, histographic data, or statistical distributions (uniform, normal, log normal, Weibull, or Gamma). In the case of individual or histographic data, the program directs the information to a curve fitting routine to fit the data to a probability density function. A numerical routine is then used to perform the double integration expression of the reliability interference of the stress and strength curves. The design and material information which has been input to DEREL can then be optimized. DERELOP uses standard nonlinear optimization algorithms and the probabilistic design procedure to iterate the stress variables and strength variables until the optimal solution is found for a specified minimum reliability. These programs are still under development and are not available commercially.

(3) The Event Time Availability Reliability Analysis (ETARA) system [23] was developed at the NASA Lewis Research Center to evaluate the performance of the Space Station Freedom electrical power system, but the methodology and software can be modified to simulate any system that can be represented by a block diagram. ETARA is an interactive, menu-driven reliability, availability, and maintainability simulation program. From a reliability block diagram of a system the program simulates the behavior of the system over a specified period of time using Monte Carlo methods to generate block failure and repair times as a function of exponential and/or Weibull distributions. The block failures are tabulated both individually and by block type, as well as total downtime, repair time, and time waiting for spares. Maintenance man-hours per year and system reliability, with or without repair, at or above a particular output capability can also be calculated. ETARA is an IBM PC (or compatible) based program.

(4) Failure Rate Analysis Tools (FRATOOLS) is a design aid being developed by the Hughes Aircraft Company [24] to support the "concurrent engineering" concept of integrating the design, analysis, and manufacturing functions from the very beginning of a project. FRATOOLS consists of several stand-alone programs integrated together to work as
a single software package. These tools were successfully used by
designers to analyze the F18 Radar Upgrade System. The FRATOOLS
methodology fully complies with the methods contained in MIL-HDBK-217E.
Some of the novel elements of FRATOOLS include the use of "smart"
software for automating the decision making process, the capability to
compute the failure rates for advanced technology devices which are not
covered in MIL-HDBK-217E, and integrating the methodology with the
design process in a manner which fully supports concurrency in the
design of radar systems.

(5) The Fault Tree Analysis (FTA) program is being developed by the Xerox
Corporation [25]. If used throughout the design phase of product
development, FTA can help isolate and resolve potential failure modes
prior to the completion of the design. FTA provides a user friendly
interface for graphical generation of fault trees. An editor allows
additions, deletions, and changes to the fault events, the links between
fault events, or the numerical information stored at each fault event.
Failure rates can be assigned to events in the tree and propagated using
AND and OR logic for intermediate, basic, and incomplete event failure
rates and the determination of minimal cutsets. FTA software runs on a
Xerox 1100 series workstation and is written in Interlisp-D. The FTA
system is almost completely menu driven, where a mouse is used to select
menu options that appear on the screen. The graphical fault tree is
generated on the screen and can be output to a hardcopy device. The
analysis is integrated with the creation and modification of the fault
tree.

(6) The Hybrid Automated Reliability Predictor (HARP) is a package developed
at Duke University under the sponsorship of the NASA Langley Research
Center [26]. It is a program that implements advanced reliability
modeling techniques characterized by a decomposition of the overall
model into fault-occurrence and fault-handling submodels. The fault-
occurrence model is a nonhomogeneous Markov chain which is solved
analytically, while the fault-handling model is a Petri net which is
simulated. In addition, HARP provides automated analysis of
sensitivities to uncertainties in the input parameters and the initial state specifications. HARP accepts the description of the system being modeled either as a fault tree or as a continuous time Markov chain. If the system is described as a fault tree, it is internally converted to the corresponding Markov chain. The fault/error handling behavior is then automatically inserted into the Markov chain representation. HARP provides a variety of alternatives for the fault/error handling model. The parameters may be described in terms of a range of values or as a single point estimate, which HARP then uses to produce a set of bounds on the time dependent reliability measure reported. In the case of nonrepairable systems, failure distributions may be exponential or Weibull.

(7) The Reliability Block Diagram Analysis (RBDA) Workstation [27] was designed to provide an environment for the development of small or large multi-system models. Capabilities include on-screen development of block diagram logic, probabilistic data, and event information. Models may also contain capacity data for multi-state analysis. Results may be quantified and reviewed for accuracy in a cut set editor, and hard copies can be plotted. RBDA integrates reliability block diagram and fault tree methodology. RBDA restructures block diagram models as fault trees. In this form, the block diagram model may be quantified to obtain cut sets. The results may then be loaded into a cut set editor for review and sensitivity analysis. Models are constructed with a block diagram to represent the functional relationship between components in a system at various levels. Blocks may represent single components or nested block diagrams which contain further logic in that event. A nested block diagram can itself contain another nested block diagram and so on, until a bottom level is reached. RBDA handles parallel and serial logic, both simple and complex, as well as combination logic to accommodate cases where m of n components are required for system success.

(8) The Reliability/Availability Analysis Program (RELAV) was developed by the California Institute of Technology and the NASA Jet Propulsion
Laboratory [23] to determine the reliability or availability of any general system which can be modeled as imbedded $k$ out of $n$ groups of components and/or subgroups. RELAV can assess current system performance during the later testing phases of a system design, as well as model candidate designs or validate and form predictions during the early phases of a design. From a reliability block diagram, it calculates the success probability of each group of items within the system assuming $k$ out of $n$ operating rules apply for each group. The program operates on a folding basis. It works its way toward the system level from the most embedded level by folding related groups into single components. The entire folding process involves probabilities; therefore, availability problems are performed in terms of the probability of success, and reliability problems are performed for specific mission lengths. RELAV operates on an IBM PC or compatible computer.

(9) The Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) program [28] is being developed jointly at Duke University and Gould Computer Systems. SHARPE uses a hierarchical modeling technique that makes it possible to use mixtures of different kinds of reliability models at different levels. This flexibility is used to decompose large state-space matrix problems. The SHARPE technique provides a result that is symbolic in the mission time variable, so each submodel need be analyzed only once for each set of model parameters. Basic, built-in model types can be combined hierarchically in a flexible manner, with the number and types of models at each level and the particular information carried between the models left up to the modeler. Components in each model type are assigned functions that are symbolic in the time variable $t$. The analysis of each model type is carried out symbolically, resulting in another function that is symbolic in $t$. The SHARPE framework provides seven model types:

1. series-parallel reliability block diagrams
2. fault trees without repeated nodes

18
3. acyclic Markov chains
4. irreducible cyclic Markov chains
5. cyclic Markov chains with absorbing states
6. acyclic semi-Markov chains
7. series-parallel directed (acyclic) graphs

Block diagram and fault tree models are specialized for modeling reliability and availability. The other model types can be used to model performance as well. SHARPE is written in the C language and is designed to operate in the UNIX or VMS environment.

(10) SPOCUS-PC [29] is a program that can analyze the reliability of a system by calculating and ranking the time-dependent unavailability, time-dependent unreliability, unavailability importance, and unreliability importance of basic events (component failures). It calculates the unavailability, failure rate, and expected number of failures for the main system failure of interest. The required input consists of control information, basic event information (failure rates, repair times, and optional names), times when the characteristics are calculated, and the list of minimal cut sets. SPOCUS-PC runs on IBM PC, XT, AT, or compatible computers.

(11) The Software Tool for the Analysis of Reliability and Safety (STARS) project [30] is being developed by the Commission of the European Communities Joint Research Center, Systems Engineering and Reliability Division in Italy. The purpose of STARS is to become an integrated set of computer aided reliability modeling and analysis tools for the various tasks involved in systems safety and reliability analysis for large plants (specifically, nuclear power plants). STARS is an expert system which includes a knowledge base and an inference engine. The knowledge base has information on the problem domain and data about the specific problem to solve. The inference engine is the processor which applies the rules and other data in the knowledge base to infer new knowledge and solve the problem. STARS has been conceived to give interactive advice and support to analysts during the execution of
safety and reliability assessment of process plants and is developed to perform a number of functions. Among these are the identification of events and event sequences leading to hazards, the construction of event trees, fault trees, and Markov models, the logical and/or probabilistic analysis of the models, the identification and treatment of common mode and common cause failures, and the identification of reliability dependency.

There are several other software programs available that are applicable in the design phase. We selected only a few representative samples that cover the methods being used at this time. Most of the others either have specific limited application or are variations of the ones discussed. Descriptions of some other programs can be found in references [31] and [32].

All the programs which have been reviewed can give the user a prediction of the system's reliability. Many of them include complex methods for manipulating the available data. However, none of them have the capability to integrate component reliability numbers with other design parameters and determine the optimum design. The basis for our methodology will be a program which uses component reliability data in conjunction with, cost data, weight data, or other design parameters and determines the system configuration which gives the optimum configuration for the chosen parameter.

3.0 ENGINEERING RELIABILITY IN DESIGN PHASE

One goal of the reliability engineer is to find the best way to increase systems reliability. Reliability is the performance characteristic of a product that reflects its ability to operate satisfactorily in order to complete the product’s assigned mission. A more precise definition of reliability is: the probability of successful operation for a specified period of time under specified conditions and environments of operation [33]. Therefore, the probability that a system successfully performs as designed is called "system reliability".
System reliability is a measure of how well a system meets its design objective, and it is usually expressed in terms of the reliabilities of the components or subsystems.

One basic method of enhancing reliability is the use of redundancy. Redundancy can be used at the hardware or software level. But it is now well accepted that many critical systems cannot achieve the required reliability without employing redundancy in their structures [34]. Although the number of required active and hot standby components in redundant systems may be dictated mostly by the reliability requirements, it is desirable to include the cost of the components in the analysis which strikes a balance between the reliability and the total cost of the redundant system.

As mentioned in the introduction, the purpose of this work is to develop a simple computer program that can help engineers to study and improve the reliability of a system in the early phase of the design. Here we first assume that the reliabilities of the components that comprise the subsystems are known. This allows one to compute reliabilities of the subsystems and hence the reliabilities of the system. Without this information the present work will not be of any benefit. A crucial assumption is then made that the system is such that it can be represented as a series structure in block diagram logic. This enables one to express the reliability of the system as a product of its subsystem reliabilities. This is expressed mathematically as

\[ R_s = \prod_{j=1}^{N} R_j \]  

In most of the previous work reviewed in this area, the authors have developed techniques to determine the optimum way to achieve a higher system reliability of an existing system. However, the goal of this work is to help the design engineer design the system to a predetermined desired reliability. The initial design of a system often has a lower reliability than what is desired. Two ways to improve its reliability will be considered; increasing the reliability of individual components and increasing the reliability of the subsystems by adding redundant components to the subsystem.
As for the nomenclature used in this development, the term "subsystem" refers to a single block as drawn in a logical reliability block diagram. The block may consist of a single component or redundant components. For example, Figure 3.1(a) depicts a system of five subsystems in series. Figure 3.1(b) shows the components which may comprise each subsystem; subsystems 1 and 4 consist of a single component, subsystems 2 and 5 have two components in parallel (single redundancy), and subsystem 3 has three components in parallel (double redundancy). The system reliability is calculated by equation (3.1) and each subsystem reliability, \( R_j \), is calculated by

\[
R_j = 1 - (1 - R_j)^{k_j}
\]  

(3.2)

where \( R_j \) is the reliability of the individual components of subsystem \( j \), and \( k_j \) is the number of parallel components in subsystem \( j \).

![Figure 3.1 A typical system with five subsystems.](image)

Three computer programs called ERDP (Engineering Reliability in the Design Phase) have been developed to examine three different approaches for achieving the desired reliability of a system. Each of these programs has its advantage, and the designer should try out all three for the initial design to gain insight into possible combinations to reach the desired reliability. The user can also exercise options of using initial components, improved components, or a combination of initial and improved components.

The first two versions of the ERDP program were developed with the idea of
minimizing a parameter of the system configuration while reaching or exceeding a specified desired reliability of the system. One parameter which the designer usually wants to minimize is cost. Therefore, the program refers to this optimization parameter as "cost". However, other parameters such as weight, volume, availability, desirability, or total number of components could be used. This cost factor for the system is calculated by first summing the costs for each component of each subsystem and then summing the costs of all the subsystems. Therefore, any parameter which is additive for the system can be used. If the exact numbers are not available, relative numbers can be used for the input data or, if each component cost factor is specified as "1", the program will minimize the number of components. The idea is that the computer program will return one configuration which is the optimum configuration for the available data. In this case, "optimum" means that the selected configuration has the minimum cost factor of all the ones that meet or exceed the desired reliability. Several iterations can be quickly run to investigate different possibilities.

The third version of the ERDP program utilizes the effort minimization method and the only design constraint is the final desired reliability. Cost of the system is calculated but it does not affect the reliability optimization process. This program is described in section 3.3.

3.1 Brute Force Method

The most obvious and least elegant method of determining the optimum configuration is to look at every possible combination and choose the one which best fits the specified criteria. This method will be called the "Brute Force" method and the program is called ERDP-BF (Engineering Reliability in the Design Phase - Brute Force).

This program will evaluate a system consisting of up to eight subsystems in series. The components can have standard and optional reliability values. For each standard and optional component, ERDP-BF will consider redundant components of up to three in parallel. Therefore, there will be six different possibilities for each subsystem. That means, for a system of eight
subsystems, ERDP-BF would have to evaluate 6⁶ or 1,679,616 different combinations. This takes several minutes on an IBM-AT personal computer. For configurations with fewer subsystems or without optional components the time lag is significantly less.

It was decided to limit the scope of the program to double redundancy for practical reasons. Most engineering components have a reliability in the range of 0.7 or higher and Figure 3.1.1 shows that, for these components, redundancy greater than 2 results in very little subsystem reliability gain.

![Figure 3.1.1 Subsystem Reliability as a Function of Redundancy](image)

To run the program, the user inputs the number of subsystems, the desired system reliability, the reliability of each component, and the cost factor of each component. The cost factor can be any parameter which the user wants to minimize, as discussed above. The program requires a cost factor to be input or else it cannot choose one "optimum" solution. The program will ask if there is an optional component for each subsystem as the data is being input. It is possible to have some subsystems with optional components and some without. See Figure 3.1.2 for a sample input screen for ERDP-BF.

After the data is entered, ERDP-BF calculates the system reliability with standard single component values and compares that to the desired reliability. If the initial reliability is less than the desired reliability it will begin
evaluating all possible subsystem combinations of standard and optional components with zero, single, and double redundancy.

When ERDP-BF evaluates a configuration, it calculates the system reliability and system cost and determines if the system reliability meets or exceeds the desired reliability. Of all the configurations which meet or exceed the desired reliability, it will choose the one with the lowest cost as the optimum configuration. This configuration will then be displayed to the user. At this point, the user will be asked to change the desired reliability and run the program again. If the answer is yes, a new value for the desired reliability will be entered and ERDP-BF will again evaluate every combination and choose the lowest cost configuration which meets or exceeds the desired reliability. See Figure 3.1.3 for a sample output screen of ERDP-BF.

The following example will illustrate the logic which is used in this program. Assume a system with two subsystems; subsystem 1 is a control circuit and subsystem 2 is an electric actuator. The standard control circuit has been shown to have a reliability of 0.80 and its cost is $100. An optional circuit with a reliability of 0.85 is available that costs $200. The standard component for subsystem 2 has been shown to have a reliability of 0.75 and its cost is $150. An optional actuator is available with a reliability of 0.85 and it costs $250. Figure 3.1.4 shows the 36 possible configurations of this system and lists the system reliability and cost for each one. The reliability and cost are plotted in Figure 3.1.5.

The plot of Figure 3.1.5 shows the configurations that ERDP-BF will choose for various desired system reliabilities. The initial system reliability is 0.600. If the desired reliability is greater than 0.600 and less than 0.720, configuration (4) will be chosen. This figure illustrates that configurations (19), (10), and (4) are the same cost but, since configuration (4) has a higher reliability, it would be chosen by ERDP-BF as the optimum. Similarly, configuration (2) would be chosen if the desired reliability is greater than
Enter the number of serial subsystems in the system: 2

Enter the desired reliability of the system: 0.80

For Component # 1

Enter the initial reliability: 0.800
Enter the initial cost: 100.00

Enter the optional reliability: 0.850
Enter the optional cost: 200.00

Figure 3.1.2  Sample Input Screen for ERDP-BF
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.800. The final system reliability is 0.816 and the system cost is 450.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 3.1.3 Sample Output Screen for ERDP-BF
### Reliability vs. Cost Table

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 Std.</td>
<td>0.80</td>
<td>100</td>
</tr>
<tr>
<td>#1 Opt.</td>
<td>0.85</td>
<td>200</td>
</tr>
<tr>
<td>#2 Std.</td>
<td>0.75</td>
<td>150</td>
</tr>
<tr>
<td>#2 Opt.</td>
<td>0.85</td>
<td>250</td>
</tr>
</tbody>
</table>

36 Possible Configurations:
(S denotes standard components and O denotes optional components)

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Reliability</th>
<th>Cost</th>
<th>Configuration</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>0.600</td>
<td>250</td>
<td>(10)</td>
<td>0.680</td>
<td>350</td>
</tr>
<tr>
<td>(2)</td>
<td>0.750</td>
<td>400</td>
<td>(11)</td>
<td>0.782</td>
<td>600</td>
</tr>
<tr>
<td>(3)</td>
<td>0.787</td>
<td>550</td>
<td>(12)</td>
<td>0.797</td>
<td>850</td>
</tr>
<tr>
<td>(4)</td>
<td>0.720</td>
<td>350</td>
<td>(13)</td>
<td>0.816</td>
<td>450</td>
</tr>
<tr>
<td>(5)</td>
<td>0.900</td>
<td>500</td>
<td>(14)</td>
<td>0.938</td>
<td>700</td>
</tr>
<tr>
<td>(6)</td>
<td>0.945</td>
<td>650</td>
<td>(15)</td>
<td>0.957</td>
<td>950</td>
</tr>
<tr>
<td>(7)</td>
<td>0.744</td>
<td>450</td>
<td>(16)</td>
<td>0.843</td>
<td>550</td>
</tr>
<tr>
<td>(8)</td>
<td>0.930</td>
<td>600</td>
<td>(17)</td>
<td>0.970</td>
<td>800</td>
</tr>
<tr>
<td>(9)</td>
<td>0.977</td>
<td>750</td>
<td>(18)</td>
<td>0.989</td>
<td>1050</td>
</tr>
</tbody>
</table>

**Figure 3.1.4** System Configurations for Controller/Actuator Example
<table>
<thead>
<tr>
<th>Subsystem #1</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std.</td>
<td>0.80</td>
<td>100</td>
</tr>
<tr>
<td>Opt.</td>
<td>0.85</td>
<td>200</td>
</tr>
<tr>
<td>Subsystem #2</td>
<td>Std.</td>
<td>0.75</td>
</tr>
<tr>
<td>Opt.</td>
<td>0.85</td>
<td>250</td>
</tr>
</tbody>
</table>

36 Possible Configurations:
(S denotes standard components and O denotes optional components)

<table>
<thead>
<tr>
<th>(19)</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.638</td>
<td>350</td>
</tr>
<tr>
<td>(20)</td>
<td>.797</td>
<td>500</td>
</tr>
<tr>
<td>(21)</td>
<td>.837</td>
<td>650</td>
</tr>
<tr>
<td>(22)</td>
<td>.733</td>
<td>550</td>
</tr>
<tr>
<td>(23)</td>
<td>.916</td>
<td>700</td>
</tr>
<tr>
<td>(24)</td>
<td>.962</td>
<td>850</td>
</tr>
<tr>
<td>(25)</td>
<td>.747</td>
<td>750</td>
</tr>
<tr>
<td>(26)</td>
<td>.934</td>
<td>900</td>
</tr>
<tr>
<td>(27)</td>
<td>.981</td>
<td>1050</td>
</tr>
<tr>
<td>(19)</td>
<td>.638</td>
<td>350</td>
</tr>
<tr>
<td>(20)</td>
<td>.797</td>
<td>500</td>
</tr>
<tr>
<td>(21)</td>
<td>.837</td>
<td>650</td>
</tr>
<tr>
<td>(22)</td>
<td>.733</td>
<td>550</td>
</tr>
<tr>
<td>(23)</td>
<td>.916</td>
<td>700</td>
</tr>
<tr>
<td>(24)</td>
<td>.962</td>
<td>850</td>
</tr>
<tr>
<td>(25)</td>
<td>.747</td>
<td>750</td>
</tr>
<tr>
<td>(26)</td>
<td>.934</td>
<td>900</td>
</tr>
<tr>
<td>(27)</td>
<td>.981</td>
<td>1050</td>
</tr>
</tbody>
</table>

Figure 3.1.4 (continued) System Configurations for Controller/Actuator Example
Figure 3.1.5  Reliability vs. Cost for ERDP–BF Controller/Actuator Example
0.720 and less than 0.750. The dashed line on the plot connects the optimum configurations that would be selected by the program as the desired reliability is increased.

3.2 Subsystem Elimination Logic

Another method of optimizing the system configuration was developed which does not require every possible combination to be calculated. This heuristic method is based on the fact that the reliability of a system of serial subsystems is the product of the reliabilities of each of the subsystems as described by equation (3.1). The relative importance of one subsystem's reliability on the whole system reliability is directly proportional to the first derivative of the system reliability with respect to that subsystem's reliability. Differentiating equation (3.1) with respect to $R_i$ gives

$$\frac{\partial R_s}{\partial R_j} = \prod_{i=1}^{N} R_i ; i \neq j. \tag{3.2.1}$$

It is evident from this equation that the partial derivative of the system reliability with respect to an individual subsystem is the product of the reliabilities of all the rest of the subsystems. This product would be greatest for the subsystem that has the lowest reliability. Therefore, the subsystem that has the most effect on the system reliability is the one with the lowest reliability. Also, because of this relationship, when the reliability of the least reliable subsystem is improved by using more reliable components or by redundant components to the point where it is above the reliability of another subsystem, it is no longer the one that has the most effect on the system reliability. Therefore, the most effective way to increase the reliability of a system is to increase the reliability of each subsystem to the same level.

Therefore, the objective of Subsystem Elimination Logic (SEL) is to systematically improve the reliability of each subsystem to a common level and eliminate each one from further consideration by performing the following
steps:

1. Calculate the average subsystem reliability that would be required to meet the desired system reliability. This is defined by

\[ \overline{R} = (R_D)^{\frac{1}{N}} \]  

(3.2.2)

where \( \overline{R} \) is the average subsystem reliability, \( R_D \) is the desired system reliability, and \( N \) is the total number of subsystems.

2. Check to see if any of the subsystems already meet or exceed \( \overline{R} \). If so, they will be eliminated from further consideration and a new \( \overline{R} \) for the remaining subsystems will be calculated after accounting for the system reliability contribution of those subsystems that were eliminated. The new average subsystem reliability requirement is calculated by

\[ \overline{R'} = \left( \frac{R_D}{R_k} \right)^{\frac{1}{N-k}} \]  

(3.2.3)

where \( R_k \) is the product of the reliabilities of the subsystems that were eliminated and \( k \) is the number of subsystems that were eliminated.

3. Select the least reliable subsystem and determine what redundancy level of standard or optional components will meet or exceed the current value of \( \overline{R} \) or \( \overline{R'} \) and choose the alternative with the lowest cost. If there are two or more alternatives that are equal cost, the one with the highest reliability is chosen. In the unlikely event that none of the available alternatives meet the required reliability, the one with the highest reliability is selected. Eliminate this subsystem from further consideration and recalculate \( \overline{R'} \) for the remaining subsystems. \( \overline{R'} \) will be updated
each time a subsystem is eliminated from contention by

\[
\bar{R}' = \left( \frac{R_D^*}{R_j} \right)^{-\frac{1}{N'-1}}
\]  

(3.2.4)

In this equation, the "*" superscript denotes quantities that are updated each time a subsystem is eliminated. \( R_D^* \) is the effective desired reliability after the reliability contribution of the previously eliminated subsystem is taken into account. If no subsystems have been eliminated, then \( R_D^* = R_D \); if subsystems have been eliminated by step #2, then \( R_D^* = (R_D)/R_k \); if this step (#3) is being repeated, then \( R_D^* = (\text{the previous } R_D^*)/R_i \), where \( R_i \) is the reliability of the subsystem that was previously eliminated. Similarly, \( N' \) is the number of subsystems that have not yet been eliminated. If no subsystems have been eliminated, then \( N' = N \); if subsystems have been eliminated by step #2, then \( N' = N - k \); if this step (#3) is being repeated, then \( N' = (\text{the previous } N') - 1 \).

4. Select the next least reliable subsystem and repeat the procedure of step #3. Keep repeating until all subsystems have been eliminated.

The computer program which performs this procedure has been named ERDP-SEL. The data input is the same as described for ERDP-BF and the output display is similar, as shown in Figures 3.2.1 and 3.2.2. The main difference is that the optimization technique is much faster, so a system with more components can be quickly analyzed. ERDP-SEL will evaluate a system of up to 20 serial subsystems. This limit is applied only because the graphical display fills the whole computer screen.

As an example of how this logic works, consider the controller/actuator example cited for the ERDP-BF program. Figure 3.2.3 shows the six possible
Enter the number of serial subsystems in the system: 8

Enter the desired reliability of the system: 0.70

For Component # 1
Enter the initial reliability: 0.800
Enter the initial cost: 100.00
Enter the optional reliability: 0.850
Enter the optional cost: 200.00

Figure 3.2.1 Sample Input Screen for ERDP-SEL
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.700. The final system reliability is 0.759 and the system cost is 2850.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 3.2.2 Sample Output Screen for ERDP-SEL
alternatives for each subsystem. There are zero, single, and double redundancy configurations for each of the standard and optional components of each subsystem. These configurations are the same as shown in Figure 3.1.4 except these are for the individual subsystems, whereas Figure 3.1.4 is for the total system.

<table>
<thead>
<tr>
<th>Subsystem #1</th>
<th>Std.</th>
<th>0.80</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opt.</td>
<td>0.85</td>
<td>200</td>
</tr>
<tr>
<td>Subsystem #2</td>
<td>Std.</td>
<td>0.75</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Opt.</td>
<td>0.85</td>
<td>250</td>
</tr>
</tbody>
</table>

**6 Possible Configurations for each Subsystem:**
(S denotes standard components and O denotes optional components)

<table>
<thead>
<tr>
<th>Subsystem 1</th>
<th>Reliability</th>
<th>Cost</th>
<th>Subsystem 2</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>S</td>
<td>.800</td>
<td>(2,1)</td>
<td>S</td>
<td>.750</td>
</tr>
<tr>
<td>(1,2)</td>
<td>S</td>
<td>.960</td>
<td>(2,2)</td>
<td>S</td>
<td>.938</td>
</tr>
<tr>
<td>(1,3)</td>
<td>S</td>
<td>.992</td>
<td>(2,3)</td>
<td>S</td>
<td>.984</td>
</tr>
<tr>
<td>(1,4)</td>
<td>O</td>
<td>.850</td>
<td>(2,4)</td>
<td>O</td>
<td>.850</td>
</tr>
<tr>
<td>(1,5)</td>
<td>O</td>
<td>.978</td>
<td>(2,5)</td>
<td>O</td>
<td>.978</td>
</tr>
<tr>
<td>(1,6)</td>
<td>O</td>
<td>.997</td>
<td>(2,6)</td>
<td>O</td>
<td>.997</td>
</tr>
</tbody>
</table>

*Figure 3.2.3 Subsystem Configurations for Controller/Actuator Example*

The first thing ERDP-SEL does is calculate $\bar{R}$. In order to calculate $\bar{R}$, the program must know the desired reliability. We'll choose a value slightly greater than the initial value so that the program will search for the first
optimum configuration that is better than the initial configuration. Let $R_D = 0.601$. Then calculate $\bar{R}$ from equation (3.2.1).

$$\bar{R} = (R_D)^{\frac{1}{N}} = 0.601^{\frac{1}{3}} = 0.775$$

The second step is to check the standard component values to see if they already meet $\bar{R}$. We see that the standard component of subsystem 1 is 0.800, which is greater than 0.775. Therefore, set configuration (1,1) as the final configuration for subsystem 1 and calculate $R'$ from equation (3.2.3) for choosing the remaining subsystem configuration.

$$R' = \left(\frac{R_D}{R_k}\right)^{\frac{1}{N'}} = \left(\frac{0.601}{0.800}\right)^{\frac{1}{3-1}} = 0.751$$

Subsystem 2 is the only one remaining and we see in Figure 3.2.3 that all configurations except (2,1) meet the reliability requirement of 0.751. Configuration (2,4) is the lowest cost of these and it will be selected as the final configuration for subsystem 2. ERDP-SEL will then calculate the resultant system reliability ($0.800 \times 0.850 = 0.680$) and return subsystem configurations (1,1) and (2,4) as the optimum system configuration to the user. This is the same as configuration (10) in Figures 3.1.4 and 3.1.5 and it is apparent that the true optimum configuration was not found by ERDP-SEL because configuration (4) is the same cost with higher reliability.

Figure 3.2.4 is the same reliability versus cost plot as Figure 3.1.5 with the dashed line showing the configurations that will be selected by ERDP-SEL as the desired reliability is increased. This figure illustrates that the true optimum will not always be found by ERDP-SEL. It shows that configurations (4) and (8) will not be selected and that the program skips over configuration (2) and then back again in an illogical manner. This is the result of the SEL method which selects a subsystem configuration without considering the combined effects of all the subsystems. Because the selection of each
Figure 3.2.4 Reliability vs. Cost for ERDP-SEL Controller/Actuator Example
subsystem configuration depends on the current value of $\overline{R}$, the final configuration is also affected by what value the user selects for the desired system reliability. In Figure 3.2.4, ERDP-SEL chose configuration (10) for $R_D$ values up to 0.580, configuration (13) for values of $R_D$ between 0.681 and 0.722, configuration (2) for values between 0.723 and 0.750, and then configuration (5) for values between 0.751 and 0.878. This points out the fact that the user must try several iterations to get an idea of what configuration would be best. It also shows that this logic is not ideal for systems with a small number of subsystems.

A hypothetical system with eight subsystems was used to further compare the Brute Force method to the Subsystem Elimination Logic method. In this example all the subsystems had optional components except subsystem # 6. Therefore, there were 839,808 possible configurations for ERDP-BF to analyze. Each iteration took about 7½ minutes on the IBM-AT computer but only about 1 minute on a Model 70-386 computer. For ERDP-SEL, however, each iteration was almost instantaneous on either computer. Table 3.2.1 describes the system and Table 3.2.2 lists the results of each program for various values of desired system reliability.

The results in Table 3.2.2 show that the SEL method did not return the true optimum configuration for each value of $R_D$. This points out the fact that the user must run several iterations to fully understand the intricacies of the system under study. It also suggests that a more sophisticated logic should be developed to analyze systems with large numbers of subsystems. The Brute Force method can be effective for systems with smaller numbers of subsystems.
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Component</th>
<th>Reliability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard</td>
<td>0.80</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>0.85</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>Standard</td>
<td>0.75</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>0.85</td>
<td>250</td>
</tr>
<tr>
<td>3</td>
<td>Standard</td>
<td>0.85</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>0.90</td>
<td>350</td>
</tr>
<tr>
<td>4</td>
<td>Standard</td>
<td>0.82</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>0.88</td>
<td>300</td>
</tr>
<tr>
<td>5</td>
<td>Standard</td>
<td>0.90</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>0.95</td>
<td>200</td>
</tr>
<tr>
<td>6</td>
<td>Standard</td>
<td>0.98</td>
<td>500</td>
</tr>
<tr>
<td>7</td>
<td>Standard</td>
<td>0.85</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>0.95</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td>Standard</td>
<td>0.75</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Optional</td>
<td>0.90</td>
<td>400</td>
</tr>
</tbody>
</table>

Initial System Reliability = 0.235
Initial System Cost = 1550
Maximum System Reliability = 0.988
Maximum System Cost = 7500

Table 3.2.1. System with Eight Subsystems for BF/SEL Example
<table>
<thead>
<tr>
<th>$R_D$</th>
<th>ERDP-BF</th>
<th>ERDP-SEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reliability</td>
<td>Cost</td>
</tr>
<tr>
<td>0.30</td>
<td>0.325</td>
<td>1750</td>
</tr>
<tr>
<td>0.40</td>
<td>0.406</td>
<td>1900</td>
</tr>
<tr>
<td>0.50</td>
<td>0.507</td>
<td>2100</td>
</tr>
<tr>
<td>0.60</td>
<td>0.658</td>
<td>2400</td>
</tr>
<tr>
<td>0.70</td>
<td>0.757</td>
<td>2600</td>
</tr>
<tr>
<td>0.80</td>
<td>0.821</td>
<td>2850</td>
</tr>
<tr>
<td>0.90</td>
<td>0.903</td>
<td>3350</td>
</tr>
<tr>
<td>0.95</td>
<td>0.953</td>
<td>4350</td>
</tr>
<tr>
<td>0.97</td>
<td>0.974</td>
<td>5050</td>
</tr>
<tr>
<td>0.98</td>
<td>0.983</td>
<td>5650</td>
</tr>
</tbody>
</table>

Table 3.2.2 Results of BF/SEL Comparison for Eight Subsystems
3.3 Effort Minimization Method (EMM)

The mathematical basis of this method has been discussed in the literature review section. Here the subsystems are first positioned in an increasing order of reliabilities. Let the reliabilities of this new ordered system be denoted by $R_i$, then

$$R_1 \leq R_2 \leq R_3 \leq \ldots \leq R_N \quad (3.3.1)$$

The most difficult part of this program is to keep track of the subsystem number and its order. For example $R_i$ may be the subsystem 3, and $R_N$ could be subsystem 1.

If the desired reliability $R_D$ is more than the initial system reliability $R_s = R_1 R_2 \ldots R_N$, then the reliabilities of the first $K$ subsystems will be increased to the same value $\overline{R}_0$ such that

$$(\overline{R}_0)^K \prod_{j=K+1}^{N} R_j = R_D \quad (3.3.2.a)$$
or

$$\overline{R}_0 = \left( \frac{R_D}{\prod_{j=K+1}^{N}} \right)^{\frac{1}{K}} \quad (3.3.2.b)$$

Note that the reliabilities of the $(K+1)$ to the $N$th subsystems in the new order are not changed. The major task here is to determine the $K$ value.

The number $K$ is determined in a way that
\[ K = \max_i \left\{ \frac{R_D}{\prod_{j=i+1}^{N+1} R_j} \right\}^{\frac{1}{2}} = R_i \]

such that

\[ R_i < \left( \frac{R_D}{\prod_{j=i+1}^{N+1} R_j} \right)^{\frac{1}{2}} = R_i \]

where \( R_{N+1} = 1 \) by definition.

The new system reliability can then be computed as

\[ R_s = (\overline{R}_0)^K \prod_{i=K+1}^{N} R_i \]

This algorithm is used in the Effort Minimization Method computer program. The program is named ERDP-EMM and it is written in the FORTRAN language. The inputs to this program are number of components, desired reliability, and individual component reliability. Following the aforementioned algorithms, the computer program locates \( \overline{R}_0 \), and subsystems that need higher reliabilities.

This method is best explained by an example. Consider an aircraft electrical system with six subsystems in series called the sensor, guidance, computer, fire control, search radar, and shoot. The system can operate successfully if and only if all these six subsystem operate. This is shown in Figure 3.3.1.
For simplicity, we have already assigned the reliabilities into an increasing form. These are given by $R_1=0.75$, $R_2=0.80$, $R_3=0.87$, $R_4=0.90$, $R_5=0.95$, $R_6=0.99$.

The initial system reliability is then $R_0 = \prod_{j=1}^{6} R_j = 0.44$. Suppose the desired reliability is $R_D=0.53$. Starting from $i=6$, $r_1$ can be determined as

$$r_6 = \left( \frac{0.53}{1} \right)^{\frac{1}{6}} = 0.90 < R_6$$

$$r_5 = \left( \frac{0.53}{0.99} \right)^{\frac{1}{5}} = 0.88 < R_5$$

$$r_4 = \left( \frac{0.53}{0.95 \times 0.99} \right)^{\frac{1}{4}} = 0.87 < R_4$$

$$r_3 = \left( \frac{0.53}{0.90 \times 0.95 \times 0.99} \right)^{\frac{1}{3}} = 0.86 < R_3$$

Therefore at $i=2$, maximum value of $i$ where Equation (3.3.4) is satisfied. Hence from Equation (3.3.3), $K=2$. Once $K$ is determined, the rest follows easily. In this example,

$$\overline{R}_0 = \left( \frac{0.53}{0.87 \times 0.90 \times 0.95 \times 0.99} \right)^{\frac{1}{2}} = 0.85$$
Hence the first two components will have to improve their reliabilities to 0.85. Then the new system reliability

\[ R_s = 0.85 \times 0.85 \times 0.87 \times 0.90 \times 0.95 \times 0.99 = 0.53 \]

The next step deals with how the new reliability \( R_0 \) can be achieved by the subsystems.

We know that \( R_j \) (where \( j \leq K \)) has to meet \( R_0 \). One option that provides this required reliability is to use redundant components. Assuming that all parallel components in a subsystem have the same reliability, \( R_i \), then the reliability of this subsystem is \( R_0 = 1 - (1 - R_j)^{X_j} \) where \( X_j \) is the total number of parallel components in the subsystem. Therefore

\[ X_j = \frac{\log(1 - R_0)}{\log(1 - R_j)} \]  

(3.3.6.a)

The \( X_j \) value must be a positive real number, since non-integer redundancy cannot be achieved. We choose \( X_j \) as

1. \( X_j = X_j \) if \( X_j \) is a positive integer number;
2. \( X_j = \lfloor X_j \rfloor + 1 \) if \( X_j \) is a positive real number.  

(3.3.6.b)

It should be noted that in condition (2), the result for the number of redundancies is a conservative solution, and \( \lfloor X_j \rfloor \) itself may result in a satisfactory result.

To demonstrate this method, we continue with the aircraft electronic system. As mentioned earlier, the first two subsystems require higher reliability (\( R_0 = 0.85 \)). At this point, the program considers the initial subsystem reliability. Then, it uses Equation (3.3.6) to determine the number of redundancies required for the subsystems to improve their reliability to meet
For $j=1$, Equation (3.3.6) yields

$$X_j = \frac{\log(1-0.85)}{\log(1-0.75)} = 1.368 + 1 = 2$$

and for $j=2$,

$$X_j = \frac{\log(1-0.85)}{\log(1-0.80)} = 1.17 + 1 = 2$$

The conservative result is shown below (Figure 3.3.2).

![Conservative solution diagram](image)

Then the final system reliability can be calculated as

$$R_s = (1-(1-0.75)^2) (1-(1-0.8)^2) (0.87) (0.90) (0.95) (0.99) = 0.66$$
In this case $R_{\text{new}}$ is much larger than the desired reliability, $R_D=0.53$.

As mentioned earlier, this algorithm provides a conservative solution. The program, then investigates the solutions for fewer redundancies for the subsystems. That is, the program considers $|x_j|$ versus the $|x_j|+1$ option to determine number of redundancies required to reach the desired system reliability. After going through this process, the program duplicates this effort for optional available redundancies. When this process is completed, the program combines the initial and optional component redundancies. It substitutes one initial reliability at a time with its maximum available reliability and performs the analysis. The configurations that meet the desired system reliability will be recorded into an external file and also printed on the computer screen. A flow diagram of the ERDP-EMM program is shown in Figure 3.3.3.

To demonstrate this process we continue with the aircraft electrical system. After the conservative solution is reached, the following two steps will be conducted.
Figure 3.3.3 Flow diagram of the ERDP-EMM computer program.
(1) For subsystem one, \( \lfloor x_j + 1 \rfloor = 2 \), and subsystem two, \( \lfloor x_j \rfloor = 1 \) yield one redundancy of subsystem one, and no redundancy for the rest (Figure 3.3.4):

\[
R_{S-new} = \left[ 1 - (1 - 0.7502) \right] (0.8) (0.87) (0.9) (0.95) (0.99) = 0.55
\]

Figure 3.3.4. One redundancy of subsystem one.

(2) For subsystem one, \( \lfloor x_j \rfloor = 1 \), and subsystem two, \( \lfloor x_j + 1 \rfloor = 2 \) yield one redundancy of subsystem two, and no redundancy for the rest (Figure 3.3.5):

\[
R_{S-new} = (0.750) \left[ 1 - (1 - (0.8)^2) \right] (0.87) (0.9) (0.95) (0.99) = 0.53
\]

Figure 3.3.5. One redundancy of subsystem two.

In this system there are no optional component reliabilities, so the process stops at this point. Otherwise the program would perform the analysis with optional reliabilities only, then combine the initial and optional together. (Replacing original with optional, one subsystem at a time.)
4.0 A CASE STUDY

This case study will demonstrate the benefit of incorporating reliability techniques into a preliminary engineering design. By utilizing reliability techniques, engineers can save the time and money spent on performing the rework. This particular system has been previously redesigned to improve its reliability. However, the reliability improvement was made by simply duplicating the entire system. It was not optimized and the actual reliability improvement was not calculated.

The following paragraphs describe the system and the previous work that was done. Then the three programs that have been described earlier in this document were used to see how they could have been used to improve the design process. The results were satisfactory, and we gained insight in understanding reliability in the design phase and how it interfaces with design constraints (volume, mass, price, etc.).

**Statement of the Problem:** Design a Backup Isokinetic Sample Pump System on the existing pump system (P-NCM-271-1) in the New Waste Calciner Facility (NWCF) area of CPP-659 at the Idaho Chemical Processing Plant. The existing pump, P-NCM-271, provides an isokinetic sample stream of the NWCF HVAC stack. This pump is necessary for emission monitoring. There was no backup pump installed. Failure of the single pump system would constitute a single point failure for monitoring stack emissions and, therefore, a more reliable design was needed.

**Requirements:** The system shall be functional for 8000 hours (333 days) continuously with 10 demands over that period of time. The desired system reliability requested by the customer was 70 percent. This number for reliability is based on the fact that, if there is 70% chance that the system will function for approximately 8000 hours and 10 demands over that period of time, the customer is satisfied with the system. Note that engineering requirements (codes, material, and engineering standards) are not our concern for this case.
Original Solution Suggested by the Customer: Design and install a backup system with a pressure regulating valve, intake air filtration system, pump, motor, sensors, and isolation ball valves in parallel with the existing system.

The sensors consist of pressure, temperature, and flow sensors. These sensors monitor the air flow through the pump. If there is a change in the pressure, temperature, or velocity of the air, before and after the pump, that will suggest a change in the air velocity which in turn results in non-isokinetic flow. (Isokinetic flow is defined as a flow which has a constant velocity throughout the piping system.)

Design and Analysis (ERDP): The backup system was designed by Ali Siahpush from the Mechanical Engineering Department of EG&G. At the time of design, the designer had no experience in reliability analysis and based on the "traditional engineering way", the design followed the ASME, ANSI, and other appropriate codes and standards. The customer requested that the entire system be duplicated, and the design of the backup system was based on that request.

The diagram and photograph of the original system is depicted in Figure 4.1.a, and 4.1.b, respectively. To simplify the analysis, the pressure, temperature, and flow sensors are ignored because they have very high reliability. The reliability numbers for the components are presented in Table 4.1.
Table 4.1. Reliability values for the Isokinetic Pump System

<table>
<thead>
<tr>
<th>COMPONENTS</th>
<th>FAILURE MODE</th>
<th>FAILURE RATE</th>
<th>COMPONENT RELIABILITY</th>
<th>PRICE ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety Valve</td>
<td>Failure to Open</td>
<td>(1) 3.E-4/D</td>
<td>(1) 0.967</td>
<td>(1) 225</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 2.1E-4/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure to Close</td>
<td>(1) 3.E-3/D</td>
<td>(2) 0.978</td>
<td>(2) 350</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 2.E-3/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vacuum Breaker</td>
<td>Failure to Open</td>
<td>(1) 1.E-2/D</td>
<td>(1) 0.90</td>
<td>(1) 345</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 0.85E-2/D</td>
<td>(2) 0.91</td>
<td>(2) 560</td>
</tr>
<tr>
<td>AC Motor</td>
<td>Failure to Start</td>
<td>(1) 3.E-5/D</td>
<td>(1) 0.786</td>
<td>(1) 250</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 2.E-5/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure to Run</td>
<td>(1) 3.E-5/H</td>
<td>(2) 0.852</td>
<td>(2) 375</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 2.E-5/H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pump</td>
<td>Failure to Start</td>
<td>(1) 3E-3/D</td>
<td>(1) 0.73</td>
<td>(1) 400</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 2E-3/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failure to Run</td>
<td>(1) 3E-5/H</td>
<td>(2) 0.834</td>
<td>(2) 655</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 2E-5/H</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>External Leakage</td>
<td>(1) 3E-6/H</td>
<td>(2) 0.834</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2) 3E-6/H</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1) Original Components  (2) Max. Available Reliable Components
Figure 4.1.a. Schematic Diagram of the Isokinetic Pump System
Figure 4.1.b. Photograph of the Pump System.
These reliability values were not provided by the manufacturer but were obtained from Reference 35. All components are in series and failure of one component will result in system failure. For clarification, the block diagram of the system and the component reliability values are depicted in Figure 4.2.

![Block Diagram](Figure 4.2. Original pump system)

The original system reliability of the pump system can be defined as

$$R_{\text{orig}} = (0.97)(0.9)(0.78)(0.73)(0.97) = 0.48 \text{ or } 48\%$$

The system reliability of 0.48 is less than the desired reliability (0.7). The next question is how do we go about increasing system reliability? The first option is to change the manufacturer and procure better (higher reliability) components. The optional components' reliabilities are also listed in Table 4.1. The block diagram of the optional components is shown in Figure 4.3.

![Block Diagram](Figure 4.3. Maximum available pump system)

With the new component reliabilities, the system reliability is
The system reliability of 0.61 is still less than the desired reliability (0.7). If the customer requested the same material and components for interchangeability, this option would not be acceptable. The next option, if possible, is redundancy. In the original backup pump design, all components were duplicated. This resulted in two parallel systems, each with reliability of 0.48 (original system). The block diagram of this system is shown in Figure 4.4.

The reliability of the parallel system can be defined as

\[ R_{parallel} = (1 - (1 - 0.48)^2) = 0.73 \text{ or } 73\% \]

So, by duplicating the entire system, the desired reliability was reached but it may not be the optimum configuration.

4.1 Designing with ERDP-BF

The parameters of the Isokinetic Pump system described above were used in the ERDP-BF program to determine the optimum configuration based on redundant components instead of duplicating the whole system. Three runs were made. The first one considered only the standard components; the second one considered only the optional components; and the third one considered the use of both standard and optional components in any combination. The results are
shown in the following figures. Figure 4.1.1 shows that the desired reliability could have been met by duplicating the standard components in subsystems 3 and 4. Figure 4.1.2 shows that, with the optional components, the desired reliability could have been met by duplicating only the subsystem 3 components. However, this configuration has a cost of $2665, whereas the system of standard components in Figure 4.1.1 was $2095. The configuration in Figure 4.1.3, for which all combinations of components were considered, is the same as when only the standard components were considered. This shows that the optional components for this system are not cost effective in this reliability range.

Out of curiosity, several more runs were made with increasing values for desired reliability just to see how high the system reliability would have to be in order for the optional components to be cost effective. The result of this exercise is shown in Figure 4.1.4. This figure shows that when the desired reliability reaches 0.969, it is cost effective to use the optional components in subsystem 3.

4.2 Designing with ERDP-SEL

The ERDP-SEL program was also used to analyze the Isokinetic Pump system to compare the results to the ERDP-BF results. Again, three runs were made; one with only standard components, one with only optional components, and one with both types of components. Figure 4.2.1 shows that, for standard components, the third and fourth subsystems would have duplicated components. This result is the same as given by ERDP-BF. Figure 4.2.2 shows that, for optional components, the fourth subsystem would have duplicated components. This result is different than ERDP-BF because the SEL method requires that the subsystem with the least reliable components will be changed to meet the average component reliability requirement before the next subsystem is changed. Therefore, subsystem 4 is changed first because it is the one with the least reliable components. The result is a system which meets the desired reliability, but is more costly than the configuration of Figure 4.1.2. When all components are considered, ERDP-SEL determines that standard components are the best and returns the configuration shown in Figure 4.2.3, which is the
same as Figure 4.2.1. This is the same result as ERDP-BF.
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.700. The final system reliability is 0.744 and the system cost is $2095.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 4.1.1 Results of ERDP-BF for Isokinetic Pump System Using Only Standard Components
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.700. The final system reliability is 0.710 and the system cost is 2665.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 4.1.2 Results of ERDP-BF for the Isokinetic Pump System Using Only Optional Components
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.700. The final system reliability is 0.744 and the system cost is 2095.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 4.1.3 Results of ERDP-BF for Isokinetic Pump System Using Both Standard and Optional Components
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.969. The final system reliability is 0.974 and the system cost is 4260.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 4.1.4 Results of EDP-BF for the Isokinetic Pump System When the Desired Reliability is Increased to 0.969
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.700. The final system reliability is 0.744 and the system cost is 2095.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 4.2.1 Results of ERDP-SEL for Isokinetic Pump System Using Only Standard Components
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.700. The final system reliability is 0.721 and the system cost is 2945.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 4.2.2 Results of ERDP-SEL for Isokinetic Pump System Using Only Optional Components
The following sketch is the optimum configuration for the system in order to reach or exceed your desired reliability of 0.700. The final system reliability is 0.744 and the system cost is 2095.00.

Would you like to change the desired reliability and run the program again? (Y/N)

Figure 4.2.3 Results of ERDP-SEL for Isokinetic Pump System Using Both Standard and Optional Components
4.3 Designing with ERDP-EMM

The computer program, ERDP-EMM, was also utilized to optimize the number of redundancies for each component. The information on the original system was input to the program and the computer results (print out) are presented in Appendix B.

The first step of the program provides the conservative solution. This solution is that the first three lowest reliability components need to have a higher reliability (0.73, 0.78, 0.9) and each requires a single redundancy in order to achieve the desired reliability. This option results in $R_s=0.822$ and price of $2440$, and is shown in Figure 4.3.1.

![Figure 4.3.1. Conservative solution for the original system.](image)

In the next step, the program reduces one redundancy at a time and checks the configuration to see if it meets the desired reliability. With original components, the program determines that, in order to reach the desired reliability, the two least reliable components must be improved and each will need a redundancy. If the redundancy is provided, as shown in Figure 4.3.2, the system reliability will be $R_s=0.75$, and its price $2095$. 
The reliability of this configuration exceeds the desired reliability and is even better than duplicating the whole system.

Next, the program does the same exercise with the optional component reliabilities. The conservative solution yields two redundancies for the two least reliable components. This option provides $R_e=0.83$ and price of $3320$. (Figure 4.3.3).

At this point the program reduces one redundancy at time and searches for the configurations that meet the reliability requirement. Two options meet the desired reliability and they are shown in Figure 4.3.4.
After exhausting all these options, the program combines initial (original) and maximum component reliabilities. It substitutes one initial reliability at a time with its maximum available reliability and performs the analysis. The configurations that meet the desired reliability are recorded into an external file. A copy of the output is presented in Appendix B.

In this project the customer requested us to use the same material and components for interchangeability. By using reliability calculations in the design phase, we have shown that there is no need to duplicate the entire system, and by duplicating a few components, one can achieve higher reliability and reduce the cost. The cost could have been reduced by over $50,000 if we had known to use the configuration of Figure 4.3.2. The cost is associated with engineering, drafting, project management, a design review team, construction, and material.
5.0 CONCLUSION AND RECOMMENDATION

The goal of this project was to develop a methodology or tool which would improve current design practices by incorporating reliability principles throughout the design phase. The purpose of developing a reliability methodology for designing a system is twofold. First, applying reliability calculations will aid the engineer to design a better system during the early phase of a project. Second, it will eliminate rework during the testing phase.

The ERDP (FY-1991) project evolved from component reliability to system reliability as it was discovered that system design is where the greater need for a methodology exists. However, most engineering system components can be analyzed by sub-dividing them into a system of sub-components. The original intent of researching the reliability of basic components with respect to material properties, failure mechanisms, and environment was not pursued.

In FY-1991, we accomplished the followings:

(1) State of the art literature review on previous work on cost versus reliability, optimization methods and algorithms for reliability allocation, and reliability software availability.
(2) Developed algorithms on how to optimize reliability in the design phase of a simple system.
(3) Developed three pilot computer programs to incorporate these algorithms into the design process.
(4) Investigated reliability of the "Back-up Isokinetic Sampling Pump System" located at ICPP-659 to verify the methodology.

In addition to this ERDP, sophisticated tools for analyzing reliability of a high technology system exist at INEL. The methods and techniques are available through the Nuclear Reactor Research & Technology (NRR&T) Department (EG&G). However, these reliability tools are not routinely incorporated into the design process and are generally used only after a system has been
designed.

In FY-1991, we considered simple systems to analyze, and for FY-1992, the methodology will be developed by completing the following tasks:

1. Develop optimization techniques to determine the desired number of component/subsystem redundancy in complex systems (mixed parallel-series, mixed series-parallel, bridge, standby, etc.) to reach the final desired system reliability. This technique maximizes the reliability of subsystems subjected to the design constraints.

2. Integrate the algorithms of the above developed techniques and develop computer codes to obtain the desired overall system reliability.

3. Collect available data on reliability and failure rate prediction techniques for mechanical components in complex systems (each technique provides a prediction for a specific component), and integrate it into the computer code. This integration will provide another option for the designer to predict the failure rate of a component based on data provided by a vendor (operating temperature, pressure, material, etc.).

4. Study and apply an artificial intelligence or expert system to the problem of reliability optimization techniques in order to provide the best decision making tool within the design constraints.

5. Finally, apply the process developed in FY-1992 to a system design performed by the Engineering Department to verify the methodology and the documentation.
REFERENCES


[14]. MilStress, Mitchell & Gauthier Associates, 73 Junction Square Drive, Concord MA 01742-3096.


[16]. BellStress, Mitchell & Gauthier Associates, 73 Junction Square Drive, Concord MA 01742-3096.


[20]. ReCalc2, T-Cubed Systems, Inc., 31220 La Baya Drive, Suite 110, Westlake Village CA 91362.


[23]. COSMIC Reliability Collection, 382 East Broad Street, Athens, GA 30602.


[29]. SPOCUS-PC, JBF Associates, Inc., Technology Dr., 100 Technology Park Center, Knoxville, TN 37392.


APPENDIX A

Glossary of Reliability Modeling Terminology
GLOSSARY OF RELIABILITY MODELING TERMINOLOGY

In this section a glossary of reliability modeling terminology is given to help the reader understand the report:

A fault tree is a logic diagram depicting the failure combinations among system elements that lead to specified kinds of system performance.

A cut set for a fault tree is a set of basic events whose occurrence causes the top event to occur. A cut set is said to be a minimal cut set if, when any basic event is removed from the set, the remaining events collectively are no longer a cut set [A.1].

Event tree is an inductive logic method for identifying the various possible outcomes of a given initiating event. In risk analysis applications, the initiating event of an event tree is typically a system failure [A.2].

Reliability Block Diagram is prepared from the functional schematic diagram of a system, indicating the functional reliability arrangement of all system components. This diagram shows the series, parallel, standby, or other arrangement of the various components.

Common-cause failures are multiple failures attributable to a common cause [A.3].

MIL-HDBK-217E - entitled: "Military Standardization Handbook - Reliability of Electronic Equipment", was developed by Rome Air Development Center and issued on October 27, 1986. This MIL-HDBK-217E provided procedures for determining the actual failure rates of various electronic components. The actual failure rate of a component may be determined by obtaining the base, or generic, failure rate and multiplying it by the appropriate application and operation stress.
factors, environment factors, and complexity factors.

The parts count technique provides an estimate of reliability based on a count by part type (resistor, capacitor, transistor, etc.). This method is applicable during proposal and early design studies where the degree of design detail is limited. It involves counting the number of parts of each type, multiplying this number by a generic failure rate for each part type and summing up the products to obtain the failure rate of each functional circuit, assembly and/or block depicted in the system block diagram. This technique is available in MIL-HDBK-217B [A.4].

The part stress analysis technique involves the same basic steps as the part count technique. However, the stress analysis technique requires the use of detailed part models plus calculation of circuit stress values for each part prior to determining its failure rate. Each part is evaluated in its electrical circuit and mechanical assembly application based on an electrical and thermal stress analysis. Once part failure rates are established, a combined failure rate for each functional block in the reliability block diagram can be determined. This technique is available in MIL-HDBK-217B [A.4].

The concepts of Markov Model are those of the state of a system and transitions between such states. The system under consideration is said to occupy a certain state whenever it satisfies the conditions that define the state. There are two basic classes of Markov models. The first is the class of discrete time models, or Markov chain models. The second is the class of continuous time models, or Markov process models [A.5].

A state in a Markov chain is an absorbing state if it is impossible to leave it. Consequently, a Markov chain is said to be an absorbing chain if it has at least one absorbing state and from every state it is possible to go on to an absorbing state.

Petri net is an abstract, formal graph model useful for representing
systems that exhibit concurrent, asynchronous, or nondeterministic behavior [A.6].

A set $S$ of states is said to be irreducible if every pair of states in $S$ communicates. If the entire state space of a Markov chain is irreducible, then the chain is said to be an irreducible Markov chain. An irreducible Markov chain is one in which every state can be reached from every other state.

A fault/error-handling model (FEHM) is designed to capture the sequence of events that occur within the system once a fault occurs. The general structure of the FEHM is a single-entry up to four exit model entered when a fault occurs. The four exits represent the four possible outcomes of the attempted system recovery. The transient restoration exit $R$ represents the correct recognition of a transient fault. The permanent coverage exit $C$ represents the successful reconfiguration of the system to eliminate a fault. The third and fourth exits from the FEHM represent system failure [A.7].

The fault-occurrence/repair model (FORM) contains information about the structure of the system (how many components of what type interconnected in what way) and about the fault arrival and repair processes (how often does each component type fail and how long does it take to fix it) [A.7].

A linear program is an optimization problem in which the objective and constraint functions are linear.

A integer program is a linear program with the additional restriction that some variables be integers.

A nonlinear program is an optimization problem in which any functions (objective and constraint) are nonlinear.

Dynamic programming is a mathematical technique dealing with the
optimization of multistage processes. A multistage processes is a process that can be separated into a number of sequential steps or stages, which may be completed in one or more ways [A.8].

Geometric programming is based on the inequality concept that the arithmetic mean is greater than or equal to the geometric mean [A.9].

The principle of sequential unconstrained minimization technique is a transformation of a constrained minimization problem into a sequence of unconstrained minimization problems. This transformation enables one to use well-established unconstrained optimization techniques to solve the constrained problem without inventing a new technique [A.10].

The parametric approach is an intermediate step in transforming the objective function terms of component reliability to terms involving some new parameters subject to constraints. Hence, the objective function formulated in parametric forms can be solved by any applicable nonlinear programming technique [A.10].

The necessary Kuhn-Tucker conditions can be stated as follows: a point which optimizes the objective function subject to the inequality constraints exists if there is a set of lagrange multipliers that satisfies the set of differentiable functions and inequality conditions.

The Lagrange multipliers in conjunction with the Kuhn-Tucker conditions technique can be generalized to solve problems involving inequality constraints and non-negative variables. The necessary Kuhn-Tucker conditions are sufficient for a global minimum, if the objective function is a convex function and the constraints form a convex set in a feasible region; and for the global maximum if the objective function is a concave function and the constraints form a convex set in a feasible region [A.9].

The generalized Lagrangian function technique is used for finding the solution of a nonlinear programming problem with inequality constraints.
and is often applied to solving optimal system reliability problems [A.9].

Sequential simplex pattern search techniques start with a finite interval in which the objective function is presumed unimodal; the function is said to be unimodal if over a given region a function increases (decreases) to a certain point and then decreases (increases) monotonically [A.10].
References


APPENDIX B

ERDP-EMM output
THE OUTPUT OF THIS PROGRAM IS SUMMARIZED IN EXTERNAL FILE NAME OUTPUT.DAT

THIS PROGRAM WILL EVALUATE THE RELIABILITY OF A SERIES COMPONENTS. CALCULATIONS STARTS WITH:
(1) RELIABILITY OF THE SYSTEM USING ONLY INITIAL COMPONENT RELIABILITY
(2) RELIABILITY OF THE SYSTEM USING ONLY MAX. COMPONENT RELIABILITY
(3) RELIABILITY OF THE SYSTEM COMBINING ONE MAX. AND THE REST INITIAL COMPONENT RELIABILITY

NOTE ON REDUNDANCY:
2 REDUNDANCY: 3 COMPONENTS IN PARALLEL
1 REDUNDANCY: 2 COMPONENTS IN PARALLEL
0 REDUNDANCY: NO REDUNDANCY

ENTER THE DESIRED SYSTEM RELIABILITY: .7

ENTER # OF COMPONENTS: 5

initial reliability calculation starts
ENTER INITIAL RELIABILITY OF COMPONENTS AND ITS $:
ONE PAIR AT A TIME. EXAMPLE: .75 50 <Enter> THEN NEXT PAIR. .97 225
.9 345
.78 250
.73 400
.97 225

OPTION:
YOU HAVE THE OPTION TO DESIGN THE SYSTEM BASED ON A MAX. PRICE, AND A DESIRED SYSTEM REL. YOU WANT TO EXPLORE THIS OPTION? (Y/N): n

REL. OF COMPON. A\' CHANGED TO AN INCREASING FORM
COMPONENTS RELIABILITY AND THEIR PRICES ($) ARE:
(.730 ($) 400.00), (.780 ($) 250.00), (.900 ($) 345.00), (.970 ($) 225.00),
SYSTEM RELIABILITY AND ITS COST ARE: 0.482 $ 1445.00
PRESS <Enter> TO CONTINUE:

THE FOLLOWING OPTION WILL PROVIDE THE CONSERVATIVE SOLUTION
OTHER OPTIONS WILL BE EXPLORED LATER
IN ORDER TO REACH DESIRED REL. THESE COMPONENTS REQUIRE THE FOLLOWING REDUN:

COMP # 1 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.927 800.00
COMP # 2 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.952 500.00
COMP # 3 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.990 690.00
IF THE ABOVE IS DONE, THEN, THE SYSTEM RELIABILITY WILL BE: 0.822
AND ITS PRICE IS ($) : 2440.00

RELIABILITY OF EACH COMPONENT IS:
0.730 0.780 0.900 0.970 0.970
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 2 WITH 1 REDUNDANCY WILL RESULT IN SYSTEM RELIABILITY OF 0.747 AND A PRICE OF ($) 2095.00

You have the option to change component reliability values to higher values. Do you want to exercise this option? (Y/N): y

max. reliability calculation starts
ONE PAIR AT A TIME. EXAMPLE: .75 50 <Enter> THEN NEXT PAIR. .98 350
.91 560
.85 375
.83 655

83
OPTION:
YOU HAVE THE OPTION TO DESIGN THE SYSTEM BASED ON A MAX. PRICE, AND A DESIRED SYSTEM REL. DO YOU WANT TO EXPLORE THIS OPTION? (Y/N): n

REL. OF COMPONENTS ARE CHANGED TO AN INCREASING FORM
COMPONENTS RELIABILITY AND THEIR PRICES ($) ARE:
(.830 ($) 655.00), (.850 ($) 375.00), (.910 ($) 560.00), (.980 ($) 350.00),
SYSTEM RELIABILITY AND ITS COST ARE: 0.617 $ 2230.00
PRESS <Enter> TO CONTINUE:

THE FOLLOWING OPTION WILL PROVIDE THE CONSERVATIVE SOLUTION
OTHER OPTIONS WILL BE EXPLORED LATER
IN ORDER TO REACH DESIRED RELI. THESE COMPONENTS REQUIRE THE FOLLOWING REDUN:

COMP # 1 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.971 1110.00
COMP # 2 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.979 750.00
IF THE ABOVE IS DONE, THEN, THE SYSTEM RELIABILITY WILL BE: 0.830
AND ITS PRICE IS ($) : 3320.00

RELIABILITY OF EACH COMPONENT IS:
0.830 0.850 0.910 0.980 0.980
IF COMPONENT # 1 HAS 1 REDUNDANCY, THEN THE SYSTEM RELIABILITY AND ITS
PRICE ($) WILL BE: 0.721 2945.00

RELIABILITY OF EACH COMPONENT IS:
0.830 0.850 0.910 0.980 0.980
IF COMPONENT # 2 HAS 1 REDUNDANCY, THEN THE SYSTEM RELIABILITY AND ITS
PRICE ($) WILL BE: 0.709 2665.00

-------->one max. & the rest original components:--------

RELIABILITY OF EACH COMPONENT IS:
.780 .830 .900 .970 .970
SYSTEM RELIABILITY AND ITS COST ARE: 0.548 $ 1445.00
PRESS <Enter> TO CONTINUE:

THE FOLLOWING OPTION WILL PROVIDE THE CONSERVATIVE SOLUTION
OTHER OPTIONS WILL BE EXPLORED LATER
IN ORDER TO REACH DESIRED RELI. THESE COMPONENTS REQUIRE THE FOLLOWING REDUN:

COMP # 1 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.952 800.00
COMP # 2 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.971 500.00
COMP # 3 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.990 690.00
IF THE ABOVE IS DONE, THEN, THE SYSTEM RELIABILITY WILL BE: 0.861
AND ITS PRICE IS ($) : 2440.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
0.780 0.830 0.900 0.970 0.970
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 2 WITH 1 REDUNDANCY WILL
RESULT IN SYSTEM RELIABILITY OF 0.783 AND A PRICE OF ($) 2095.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
0.780 0.830 0.900 0.970 0.970
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 3 WITH 1 REDUNDANCY WILL
RESULT IN SYSTEM RELIABILITY OF 0.736 AND A PRICE OF ($) 2190.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
COMPONENT # 2 WITH 1 REDUNDANCY AND COMPONENT # 3 WITH 1 REDUNDANCY WILL RESULT IN SYSTEM RELIABILITY OF 0.706 AND A PRICE OF ($): 2040.00

RELIABILITY OF EACH COMPONENT IS:
0.730 0.350 0.900 0.970 0.970
SYSTEM RELIABILITY AND ITS COST ARE: 0.525  $ 1445.00
PRESS <Enter> TO CONTINUE:

THE FOLLOWING OPTION WILL PROVIDE THE CONSERVATIVE SOLUTION
OTHER OPTIONS WILL BE EXPLORED LATER
IN ORDER TO REACH DESIRED RELI. THESE COMPONENTS REQUIRE THE FOLLOWING REDUNDANCY:

COMP # 1 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($): 0.937 300.00
COMP # 2 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($): 0.973 300.00
COMP # 3 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($): 0.990 300.00
IF THE ABOVE IS DONE, THEN, THE SYSTEM RELIABILITY WILL BE: 0.844
AND ITS PRICE IS ($): 2440.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
0.730 0.350 0.900 0.970 0.970
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 2 WITH 1 REDUNDANCY WILL RESULT IN SYSTEM RELIABILITY OF 0.757 AND A PRICE OF ($): 2095.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
0.730 0.350 0.900 0.970 0.970
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 2 WITH 1 REDUNDANCY WILL RESULT IN SYSTEM RELIABILITY OF 0.754 AND A PRICE OF ($): 2190.00

----- one max. & the rest original components: <-----

RELIABILITY OF EACH COMPONENT IS:
0.730 0.780 0.910 0.970 0.970
SYSTEM RELIABILITY AND ITS COST ARE: 0.488  $ 1445.00
PRESS <Enter> TO CONTINUE:

THE FOLLOWING OPTION WILL PROVIDE THE CONSERVATIVE SOLUTION
OTHER OPTIONS WILL BE EXPLORED LATER
IN ORDER TO REACH DESIRED RELI. THESE COMPONENTS REQUIRE THE FOLLOWING REDUNDANCY:

COMP # 1 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($): 0.927 800.00
COMP # 2 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($): 0.952 500.00
IF THE ABOVE IS DONE, THEN, THE SYSTEM RELIABILITY WILL BE: 0.755
AND ITS PRICE IS ($): 2095.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
0.730 0.780 0.910 0.970 0.970
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 2 WITH 1 REDUNDANCY WILL RESULT IN SYSTEM RELIABILITY OF 0.755 AND A PRICE OF ($): 2095.00

----- one max. & the rest original components: <-----

RELIABILITY OF EACH COMPONENT IS:
0.730 0.780 0.900 0.970 0.980
SYSTEM RELIABILITY AND ITS COST ARE: 0.487  $ 1445.00
PRESS <Enter> TO CONTINUE:
THE FOLLOWING OPTION WILL PROVIDE THE CONSERVATIVE SOLUTION.
OTHER OPTIONS WILL BE EXPLORED LATER
IN ORDER TO REACH DESIRED RELI. THESE COMPONENTS REQUIRE THE FOLLOWING REDUN:

COMP # 1 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.927 800.00
COMP # 2 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.952 500.00
COMP # 3 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.990 690.00
IF THE ABOVE IS DONE, THEN, THE SYSTEM RELIABILITY WILL BE: 0.830
AND ITS PRICE IS ($) : 2440.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
0.730 0.780 0.900 0.970 0.980
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 2 WITH 1 REDUNDANCY WILL RESULT IN SYSTEM RELIABILITY OF 0.755 AND A PRICE OF ($) 2095.00
-------->one max. & the rest original components: <--------

RELIABILITY OF EACH COMPONENT IS:
0.730 0.780 0.900 0.970 0.980
SYSTEM RELIABILITY AND ITS COST ARE: 0.487 $ 1445.00
PRESS <Enter> TO CONTINUE:

THE FOLLOWING OPTION WILL PROVIDE THE CONSERVATIVE SOLUTION
OTHER OPTIONS WILL BE EXPLORED LATER
IN ORDER TO REACH DESIRED RELI. THESE COMPONENTS REQUIRE THE FOLLOWING REDUN:

COMP # 1 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.927 800.00
COMP # 2 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.952 500.00
COMP # 3 NEEDS 1 REDUN. NEW REL & PRICE OF THE COMP. ARE ($) : 0.990 690.00
IF THE ABOVE IS DONE, THEN, THE SYSTEM RELIABILITY WILL BE: 0.830
AND ITS PRICE IS ($) : 2440.00

ONE MAX. AND THE REST INITIAL COMPONENT CALCULATION:
RELIABILITY OF EACH COMPONENT IS:
0.730 0.780 0.900 0.970 0.980
COMPONENT # 1 WITH 1 REDUNDANCY AND COMPONENT # 2 WITH 1 REDUNDANCY WILL RESULT IN SYSTEM RELIABILITY OF 0.755 AND A PRICE OF ($) 2095.00

WOULD YOU LIKE TO RUN THE PROGRAM AGAIN (Y/N):

END  DATE
1.10.92