19

OPTIMAL AVAILABILITY ALLOCATION IN SERIES-PARALLEL MAINTAINED SYSTEMS

2115-5574A

by

Chang Hoon Lie

B.S. (Nuclear Engineering), Seoul National University

Seoul, Korea, 1970

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

Kansas State University

Manhattan, Kansas

1974

Approved by :

Frank A. Tillman
Co-Major Professor

Ching-Lai Hwang
Co-Major Professor

LD
2668
T4
1974
L538
C.2
Document

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to his major professors, Dr. F.A. Tillman and Dr. C.L. Hwang for their valuable guidance and helpful advice in the course of this work. He also acknowledges the support and encouragement from Dr. D.L. Grosh and Dr. N.D. Eckhoff.

The author wishes to extend his deep gratitude to his parents for their constant help and inspiration ; to his wife, Kyunghee, for her continued support and patience throughout this work.

THIS BOOK CONTAINS NUMEROUS PAGES WITH MULTIPLE PENCIL AND/OR PEN MARKS THROUGHOUT THE TEXT.

THIS IS THE BEST IMAGE AVAILABLE.

Chapter 1

INTRODUCTION

Increasing complexity of modern-day equipment, both in the military and commercial areas, has brought with it new types of engineering problems involving high performance, reliability, and maintainability. Reliability engineering appeared on the scene in the late 1940s and early 1950s and first applied to the fields of communication and transportation. Much of the early reliability work was confined to making trade-offs between certain performance and reliability aspects of systems. Ever since, reliability has always been considered during system design. However, as systems have become increasingly complex, the reliability problem has become more acute.

Despite the fact that the reliability programs were effective in prolonging the life of systems, the concept of maintainability was recognized as a result of the reliability programs conducted in the late 1940s and early 1950s, which indicated that 100% reliability of system was an unobtainable goal. Therefore, although problems in reliability continue, when maintenance is possible, another aspect of system performance - that of maintenance and repair - is now becoming a major discipline from the point of view of engineering design and management. The objective of maintainability is to design and develop systems which can be maintained in the least time, at the least cost, and with a minimum expenditure of support resources, without adversely affecting the each unit's performance or safety charateristics.

If maintenance is possible, reliability is an incomplete measure for the system effectiveness in that it only considers the mean time to failures. Thus, an appropriate single measure for the system effectiveness which takes into account the duration of repairs as well as the frequency of failures are required. This single measure of effectiveness for the maintained system is availability which is of primary concern in this study.

Availability models for the series-parallel systems consisting of subsystems in series, where each subsystem has identical units in parallel are developed assuming various probability density functions for failure and repair times of each unit. In developing the availability models, two types of maintenance policies for each subsystem are considered : the corrective maintenance is performed when the subsystem fails due to the failure of all redundant units and the preventive maintenance is scheduled at a fixed age of the subsystem and is actually performed only if the subsystem has not failed before this fixed age. Preventive maintenance action consists of replacing or reparing only the failed units if each unit has a constant failure rate and replacing both failed and unfailed units if each unit has an increasing failure rate with time. Thus, each subsystem is assumed to be fully restored after the completion of either corrective or preventive maintenance. The cost of the system consists of three cost components : the cost for designing the mean time between maintenance and mean corrective and preventive maintenance time, the cost for corrective maintenance, and the cost for preventive maintenance.

The optimal availability allocation problem, then, is to determine individual units' detailed availability specifications which will minimize the total cost of the system under the constraint of meeting the system availability requirement. Both, the cost function and the availability equation of the system, are highly nonlinear, the optimization methods employed to solve this problem are both generalized reduced gradient (GRG) method and sequential unconstrained minimization technique (SUMT). This availability allocation technique is applicable in the early stages of maintained system design to determine individual units' detailed availability specifications that will achieve a specified level of system availability with the least cost for the system. This technique may also be applied in the latter stages of system design when modifications and improvements for the initial specifications are required.

Chapter 2

BASIC CONCEPTS

2.1 INTRODUCTION

Reliability is defined as the probability of a system performing its purpose adequately for the period of time intended under the operating conditions encountered. If f(t) is the probability density function of failure times of a system or a unit, then the reliability function R(t) is given by

$$R(t) = P(T>t) = \int_t^\infty f(s)ds \tag{2.1}$$

where T = time to failure or life length

P = the probability

Reliability has always been considered during system design. However, as the high degree of complexity is involved in many of the modern large - scale electronic systems which are required to give continuous service, e.g., computing equipment used to monitor and regulate continuous processes such as commercial power distribution, certain types of communication systems, and military defense systems on continuous alert, etc., it is difficult even with the best design technique to obtain long mean operating periods between failures. Therefore several practices have been adopted to offset the high failure rates. Redundancy is ordinarily employed in the various subsystems of the system so that a subsystem failure occurs only when all units are down. However, when maintenance is possible, reliability is an incomplete measure for the system effictiveness, thus other appropriate measures which take account of the duration of repairs as well as the frequency of failures are required.

Dependability is an appropriate measure when a system is assigned to a mission with a specified duration. It is defined as the probability that a system either does not fail or fails and is repaired in an allowable time interval during a mission period [94]. It considers operating time and active corrective maintenance time during a mission period. If a system is intended for continuous use for a long period of time and preventive maintenance is considered as well as corrective maintenance, then availability or fractional uptime is a better measure for the system effectiveness [72]. The definition and concepts of availability will be discussed in the later sections.

2.2 CORRECTIVE AND PREVENTIVE MAINTENANCE

All recoverable systems which are used for continuous or intermittent service for some period of time are subject to maintenance at one time or another. In general, maintenance actions can be classified in two categories : First, there is unscheduled or corrective maintenance, necessitated by system in - service failure or malfunction. Its purpose is to restore system operation as soon as possible by replacing, repairing, or adjusting the unit or units which cause interruption of service. Second, there is scheduled or preventive maintenance actions. Its purpose is to keep the system in a condition consistent with its built - in levels of performance, reliability, and safety. According to Bazovsky[14], preventive maintenance fulfills this purpose by servicing, inspections, and minor or major overhauls during which

"1. regular care is provided to normally operating subsystems and units which require such attention (lubrication, refueling, cleaning, adjustment, alignment, etc.),
2. failed redundant units are checked, replaced, or repaired if the system contains redundancy, and
3. units which are nearing a wearout condition are replaced or overhauled."

These actions are performed to prevent unit and system failure rates from increasing over and above the design levels.

2.3 MAINTAINABILITY INDEXES

Let's examine the maintainability indexes in some detail. The following indexes are the means for determining whether or not the maintainability requirement stated in the overall specification for a system has been complied with, and are defined in [17] and [29].

Mean time to repair (MTTR)

Mean active corrective maintenance time ($\bar{M}_{ct}$) is often construed as being synonymous with MTTR. It is the statistical mean of the times required to repair a unit or a system, and as such, represents the summation of all repair times, divided by the total number of failures that occurred during a given period. It is expressed by the following equation :

$$\text{MTTR} = \bar{M}_{ct} = \frac{\sum_{i=1}^{f_c} (M_{ct})_i}{f_c} \qquad (2.2)$$

where f_c = number of failures

= number of corrective actions

M_{ct} = active maintenance time per corrective maintenance task.

<u>Mean preventive maintenance time</u> ($\bar{M}_{pt}$)

In order to reduce the probability that a system will require corrective action, it normally is taken out of operation from time to time for preventive action. Because the time required for this type of action represents a portion of the total period of a system's inoperability, it must be calculated as contributing to total system down-time. Mean preventive maintenance time thus is defined as the statistical mean of the summation of periods required for preventive action, divided by the total number of preventive actions scheduled for a period as follows :

$$\bar{M}_{pt} = \frac{\sum_{i=1}^{f_p} (M_{pt})_i}{f_p} \qquad (2.3)$$

where f_p = number of preventive maintenance actions

M_{pt} = active maintenance time per preventive maintenance.

<u>Mean active corrective and preventive maintenance time ($\bar{M}$)</u>

This index is established to represent all system down-time resulting from both corrective and preventive activities ; as such, it represents active down-time, thereby excluding the down-time for which administrative actions, unavailability of tools, etc., are responsible. It is the statistical mean of the periods during which corrective and preventive work is performed on a system during a given period, divided by the total number of all such maintenance actions. It is calculated by use of the following equation :

$$\bar{M} = \frac{\bar{M}_{ct} f_c + \bar{M}_{pt} f_p}{f_c + f_p} \tag{2.4}$$

or

$$\bar{M} = \frac{\bar{M}_{ct}(1/MTBM_u) + \bar{M}_{pt}(1/MTBM_s)}{1/MTBM_u + 1/MTBM_s} \tag{2.5}$$

where $MTBM_u$ = mean interval of unscheduled or corrective maintenance

$MTBM_s$ = mean interval of scheduled or preventive maintenance

Mean down-time (MDT)

It is the sum of mean active corrective and preventive maintenance time ($\bar{M}$) and mean delay time for that system during a specified period. Because delay time is determined by administrative and supply factors that cannot accurately be anticipated, they are beyond a designer's control, and accordingly, can play little part in maintainability design.

2.4 AVAILABILITY

This is the principal measure of the effectiveness of maintained systems and is of primary concern in this work. Availability is defined as the fraction of the total desired operating time that the system is actually operable, or it can be defined as the ratio of uptime to total time [26]:

$$A = \frac{\text{Uptime}}{\text{Total Time}} \tag{2.6}$$

This equation can be rewritten as

$$A = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} \tag{2.7}$$

where MTBF = mean time between failures

Equation (2.7) is frequently called the inherent availability.

To examine how equation (2.6) can be written as equation (2.7), let's introduce system utilization factor U defined as follows [14].

$$U = \frac{t}{t + T_o + T_p + T_r} \tag{2.8}$$

where t = system operating time

T_o = total corrective maintenance time

T_p = total preventive maintenance time

T_r = downtime other than T_o and T_p such as administrative time, supply time, etc.

If the system is in an ideal support environment without consideration for preventive maintenance action and T_r, then we have

$$U = \frac{t}{t + T_o} \tag{2.9}$$

This is a measure of the system's availability because it gives the percentage of time the system will be available for operation. If for the system operating time we select its mean time between failures MTBF, which may be some fraction or some multiple of t in U, we can then derive the average maintenance time MTTR which is required for every MTBF system operating hours.

$$MTTR = T_o \frac{MTBF}{t} \tag{2.10}$$

Now if we use MTBF and MTTR instead of t and T_o in the utilization factor, we obtain a value which is numerically identical with U, which is by definition called system availability A as given by equation (2.7). It gives the same percentage of average time the system will be available for service as obtained from U.

2.5 THREE CONCEPTS OF AVAILABILITY

In general there are three concepts of availability, i.e., inherent availability, achieved availability, and operational availability. Blanchard and Lowery [17] define them as follows.

Inherent availability (A_i)

The probability that a system or a unit, when used under stated conditions, without consideration for any scheduled or preventive action, in an ideal support environment (i.e., available tools, spares, manpower, data, etc.), shall operate satisfactorily at a given point in time. It excludes ready time, preventive maintenance downtime, logistics time, and waiting or administrative downtime. It is a function of the reliability and the mean active corrective maintenance time characteristics of the system. It can be expressed as

$$A_i = \frac{MTBF}{MTBF + MTTR} \tag{2.11}$$

Achieved availability (A_a)

The probability that a system or a unit, when used under stated conditions in an ideal support environment, shall operate satisfactorily at a given point in time. It excludes logistics time and waiting or administrative downtime. It includes active preventive and corrective maintenance downtime and is a function of the frequency of maintenance and the mean maintenance time. It can be expressed as

$$A_a = \frac{MTBM}{MTBM + \bar{M}} \tag{2.12}$$

where $\bar{M}$ is mean active corrective and preventive maintenance time given in equation (2.5) and MTBM is the mean time between maintenance or mean interval of all maintenance requirements which can be expressed as

$$MTBM = \frac{1}{1/MTBM_u + 1/MTBM_s} \tag{2.13}$$

Operational availability (A_o)

The probability that a system or a unit, when used under stated conditions in an actual operational environment, shall operate satisfactorily at a given point in time. It includes ready time, logistics time, and waiting or administrative downtime. It can be expressed as

$$A_o = \frac{\text{MTBM + ready time}}{\text{(MTBM + ready time) + MDT}} \qquad (2.14)$$

where ready time = operational cycle - (MTBM + MDT) .
The operational cycle is the total of all operating time, ready time, and down-time. This is illustrated in Figure 2.1.

Now we have defined three concepts of availability. Of the three concepts of availability, achieved availability is the major concern in this work. The system considered in this work is assumed to be used for continuous service for some period of time. Hence both the corrective and preventive maintenance actions are assumed to be taken during the duty time. If the system is used for intermittent service for some period of time and the preventive or corrective maintenance is done during off duty time, then this sould be reflected in f_c, f_p or $MTBM_u$ and $MTBM_s$. In the later chapters, achieved availability for the particular system will be developed assuming various probability density functions for the failure and repair times of each unit.

For those terms not defined in this section, refer to Appendix A1.1.

2.6 PROFITABILITY OF PREVENTIVE MAINTENANCE

Preventive maintenance is described as a particular category of maintenance designed to optimize the related

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

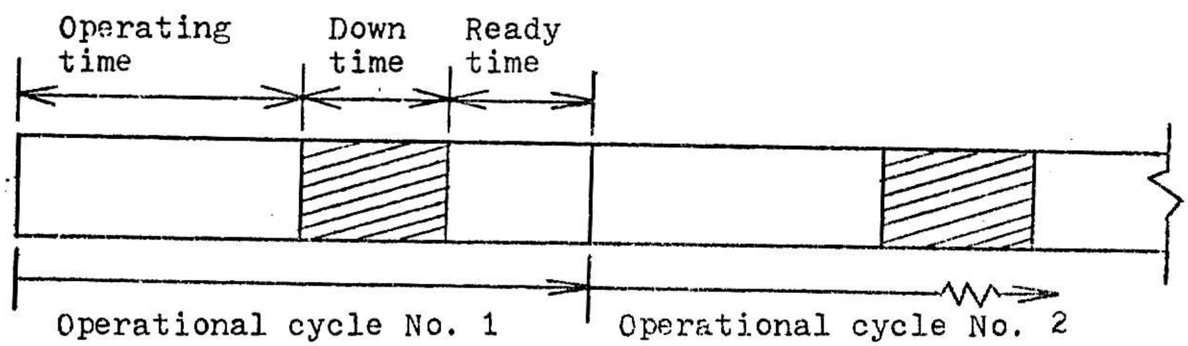

Figure 2.1. Operational cycle

concepts of availability and the costs associated with the repair actions. By performing preventive maintenance it is expected to reduce the operational failures, thus it may be possible either to increase the availability of the system or to reduce the total maintenance costs. To achieve a balance between availability and maintenance costs for any system, several factors must be weighed simultaneously. According to ARINC [84], the various factors to be considered are :

" 1. the reliability index and time duration desired ;
2. the cost of an in-service failure ;
3. the cost of replacement before failure ;
4. the most economical point in the equipment life to effect this replacement ; and
5. the predictability of the failure pattern of the equipment under consideration. "

To make preventive maintenance worthwhile, the failure rate of the systems and/or units must increase with time, or the preventive maintenance of the system must cost less than the corrective maintenance. The preventive maintenance is also advantageous for those systems which exhibit probability density functions with coefficient of variation of failure times less than that of the exponential distribution.

With regard to the cost required for the maintenance action, Bell, Kamins, and McCall [15] show three reasons for expecting this cost to be higher for a corrective maintenance than for a preventive maintenance. Because of the unexpectedness of a corrective maintenance, the reaction to a demandis not immediate, thus relatively long periods are spent awaiting service in case of a corrective maintenance, thereas this waiting time can be reduced to a minimum in case of preventive maintenance. The second reason is that actual repair or replacement time is often greater for a corrective maintenance than for a preventive maintenance because it is more difficult to repair or replace a failed unit than to repair or replace an unfailed unit, and the failure of a unit sometimes causes damage to other units. Finally, since more resources required to perform the maintenance action are needed for a corrective maintenance than for a preventive maintenance the value per unit time of the output foregone during a corrective maintenance action often exceeds the same measure for a preventive maintenance action.

The profitability of preventive maintenance with regard to the failure rate is discussed below : Let us assume that the system can be restored to its original condition after the completion of a preventive maintenance action and the preventive maintenance is scheduled at age T. Then, for a system having an increasing failure rate over time, the failure rate r(t) drops back to the original level at age T as a result of the preventive maintenance as illustrated in Figure 2.2.(a). Hence, the actual failure of a system can be reduced, thus an increase in the mean life or the mean time between unscheduled maintenance can be attained. In this situation, preventive maintenance is worthwhile.

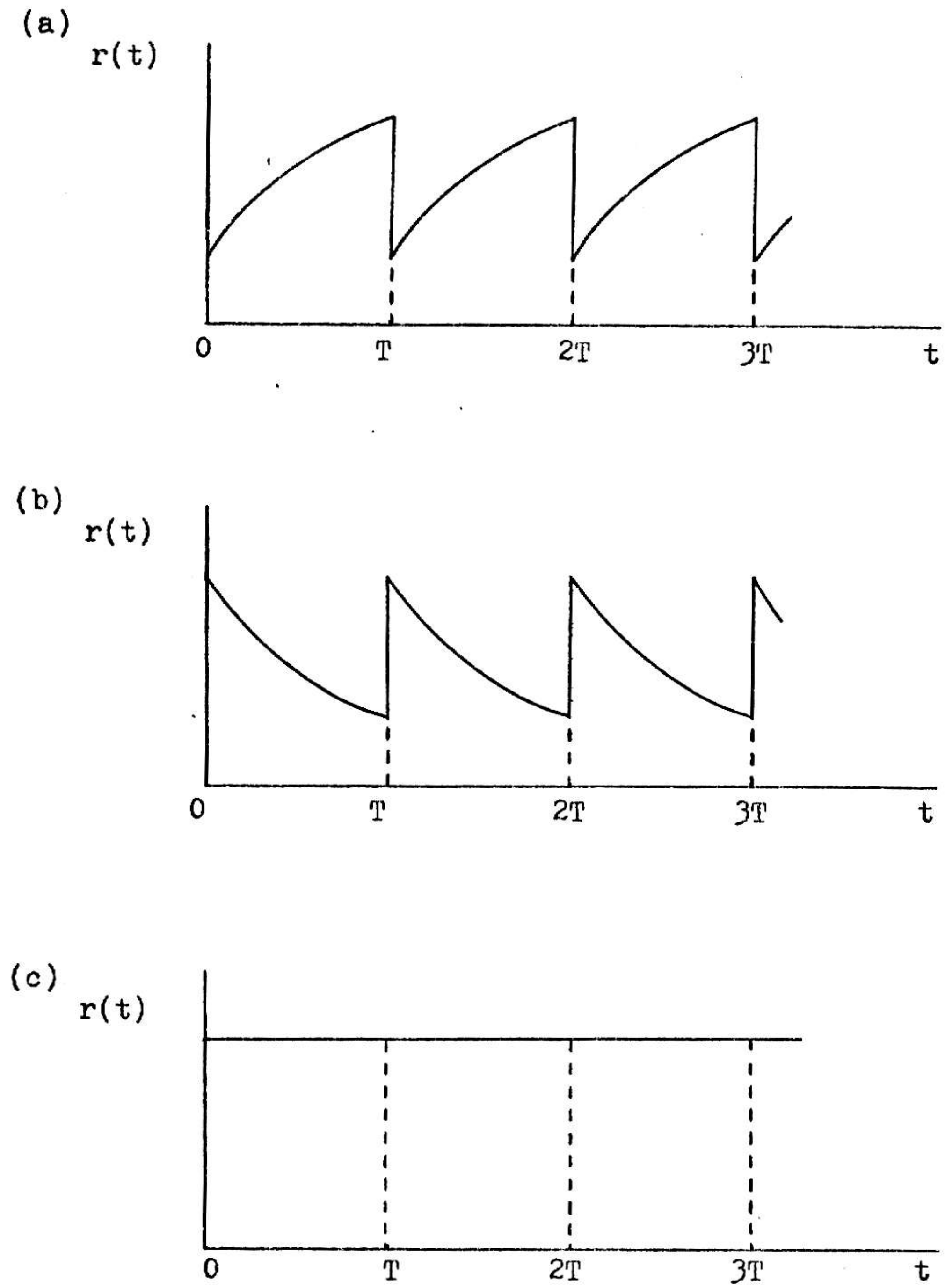

Figure 2.2. The failure rate r(t) versus time t when preventive maintenance is performed at age T: (a) increasing failure rate, (b) decreasing failure rate, and (c) constant failure rate.

As shown in Figure 2.2.(b), if the failure rate of a system decreases with time, the performance of the preventive maintenance makes a system more unreliable. In this case, since the older system is more reliable, preventive maintenance is not worthwhile. For a system having a constant failure rate, the age of a system has nothing to do with its failure rate as shown in Figure 2.2.(c). Thus, we gain nothing by applying preventive maintenance to this system.

The coefficient of variation, $\sqrt{V(t)}/E(t)$, is also closely related to the failure rate. Barlow and Proschan [11] show that the coefficient of variation of a failure distribution having an increasing failure rate over time is less than that of the exponential distribution. For a distribution having a decreasing failure rate over time, the inequality is reversed. Hence, the coefficient of variation can be used as an alternative criterion to test the profitability of the preventive maintenance. In other words, the preventive maintenance is worthwhile for those systems which exhibit probability density functions with coefficient of variation of failure times less than that of the exponential distribution, since this implies that the systems have increasing failure rates over time.

If the failure rate of system decreases with time, preventive maintenance is not worthwhile. This corresponds to the case of greater coefficient of variation than that of the exponential distribution. Examples of probability density functions for failure times which belong to this category are :

1. Gamma distribution for $\alpha < 1$ and $\lambda = 1$

$$f(t) = \frac{\lambda}{\Gamma(\alpha)} (\lambda t)^{\alpha-1} e^{-\lambda t} \quad , t > 0 \qquad (2.15)$$

2. Weibull distribution for $\alpha < 1$ and $\lambda = 1$

$$f(t) \doteq (\lambda\alpha) t^{\alpha-1} e^{-\lambda t^{\alpha}} \quad , t > 0 \qquad (2.16)$$

where λ and α are scale and shape parameters respectively. If the failure rate increases with time, the preventive maintenance is worthwhile. This corresponds to the case of smaller coefficient of variation than that of the exponential distribution. Examples of probability density functions for failure times which belong to this category are :

1. Normal distribution

$$f(t) = \frac{2}{\sqrt{2\pi(V(t))}} e^{-\frac{(t-E(t))^2}{2(V(t))}} \qquad (2.17)$$

2. Gamma distribution for $\alpha > 1$ and $\lambda = 1$
3. Weibull distribution for $\alpha > 1$ and $\lambda = 1$
4. Erlang distribution for $k > 1$

$$f(t) = \frac{(k\lambda)^k (\lambda t)^{k-1} e^{-k\lambda t}}{(k-1)!} \qquad (2.18)$$

If the failure rate of a system is constant, it presents the border-line case for which preventive maintenance may be or may not be advisable. The following distributions are belong to this case :

1. Exponential distribution

$$f(t) = \lambda e^{-\lambda t} \qquad , t > 0 \tag{2.19}$$

where λ = failure rate (constant)

2. Gamma distribution for $\alpha = 1$, $\lambda = 1$

3. Weibull distribution for $\alpha = 1$, $\lambda = 1$

4. Erlang distribution for $k = 1$

The above four distributions are identical because they reduce to exponential distribution for the specified values of parameters.

A few examples are shown below to illustrate this coefficient of variation characteristics. Now, the coefficient of variation of the exponential distribution is 1 as shown below : Since exponential distribution is given by equation (2.19), expected value of failure times E(t) is

$$E(t) = \int_0^\infty tf(t)dt = \frac{1}{\lambda} \tag{2.20}$$

and

$$E(t^2) = \int_0^\infty t^2 f(t)dt = \frac{2}{\lambda^2} \tag{2.21}$$

Then the variability V(t) is

$$V(t) = E(t^2) - (E(t))^2 = \frac{1}{\lambda^2} \tag{2.22}$$

Hence, the coefficient of variation is

$$\sqrt{V(t)} \;/\; E(t) = 1 \tag{2.23}$$

Let us consider a system with two identical units in parallel whose failure times are exponentially distributed with parameter λ. Then the reliability of each unit $R_a(t)$ is

$$R_a(t) = \int_t^\infty f(s)ds = e^{-\lambda t} \tag{2.24}$$

thus the reliability of the system $R_s(t)$ is

$$R_s(t) = R_a(t) + R_a(t) - (R_a(t))(R_a(t))$$

$$= 2e^{-\lambda t} - e^{-2\lambda t} \tag{2.25}$$

The density function for failure times of the system $f_s(t)$ is

$$f_s(t) = -\frac{dR_s(t)}{dt}$$

$$= 2\lambda e^{-\lambda t} - 2\lambda e^{-2\lambda t} \tag{2.26}$$

The expected value of failure times of the system $E(t)$ is

$$E(t) = \int_o^\infty tf_s(t)dt = \frac{3}{2\lambda} \tag{2.27}$$

and

$$E(t^2) = \int_o^\infty t^2 f_s(t)dt = \frac{7}{2\lambda^2} \tag{2.28}$$

Then the variability $V(t)$ is

$$V(t) = E(t^2) - (E(t))^2 = \frac{5}{4\lambda^2} \tag{2.29}$$

Hence, the coefficient of variation of the two-unit redundant system is

$$\sqrt{V(t)}\ /\ E(t) = \sqrt{\frac{5}{4\lambda^2}} \Big/ \frac{3}{2\lambda} = \frac{\sqrt{5}}{3} = 0.743 \tag{2.30}$$

Since the coefficient of variation of the two unit redundant system, 0.743, is less than that of a single unit system with the exponential failure distribution,1, the preventive maintenance would be worthwhile.

Let us now consider a system with k functional subsystems connected in series. Each subsystem is assumed to have approximately identical failure rate $k\lambda$ and individually characterized by exponential failure distribution.

$$f(t) = k\lambda e^{-k\lambda t} \tag{2.31}$$

Then the Erlang k density function for this system is given by

$$f(t) = \frac{(k\lambda)^k(\lambda t)^{k-1}e^{-k\lambda t}}{(k-1)!} \tag{2.32}$$

Further we can obtain

$$R(t) = e^{-k\lambda t} \sum_{n=0}^{k-1} \frac{(k\lambda t)^n}{n!} \tag{2.33}$$

$$E(t) = \frac{1}{\lambda} \tag{2.34}$$

$$V(t) = \frac{1}{k\lambda^2} \tag{2.35}$$

Hence, the coefficient of variation of the system is

$$\sqrt{V(t)}\ /\ E(t) = \sqrt{\frac{1}{k\lambda^2}} \Big/ \frac{1}{\lambda} = \frac{1}{\sqrt{k}} \tag{2.36}$$

Since

$$\frac{1}{\sqrt{k}} < 1 \qquad \text{for} \quad k > 1 \tag{2.37}$$

i.e., the system coefficient of variation of failure times is less than that of the exponential case, preventive maintenance would be worthwhile.

Chapter 3

LITERATURE SURVEY

3.1 RELIABILITY AND AVAILABILITY MODELS FOR THE SYSTEM WITH CORRECTIVE MAINTENANCE.

Reliability models for systems with repair have been discussed in a number of articles [31, 36, 39, 46, 56, 67]. Exponential distributions are frequently assumed for failure times and repair times. This assumption allows us to employ a Markovian approach which, in turn, permits us to work with linear homogeneous differential equations with constant coefficients as a result of Laplace transforms. The Markovian approach in the formulation of reliability models of the system is developed by Barlow and Hunter[8,9], Epstein and Hosford[31], and Htun[50]. deMercado[28] develops methods for predicting the reliability and moments of the first time to failure of complex systems having many failed states by using discrete Markov processes. He shows that once the matrix of the constant failure and repair rates of the subsystems is known, and the state assignment is made, then we can obtain the probabilistic description of the complex system. Sandler[76] and Shooman[80] also demonstrate the use of Markov process in developing both reliability models for the non-maintained system and availability models for the maintained system. They consider systems with a single unit, units in series, units in parallel and standby under various repair policies. To apply Markov processes in formulating availability models, exponential distribution must be assumed for failure and repair times because it enables us to have constant failure rate and repair rate, thus a lack of memory property is satisfied.

To justify the use of exponential failure law, experimental and operational data have been collected. One of the earliest reports of a statistical nature was made by Carhart[21], and subsequent studies by Davis[27] and Boodman[18] indicate that this distribution adequately fits failure experience. With regard to repair times, Rohn[72] maintains that the essential characteristic of repair times of complex electronic equipment is stated as a high frequency of short repair times and a few long repair times, thus this sort of behavior suggests representtation by an exponential distribution. Howard, Howard and Hadden[48] presents the operational data of ground equipment for surface-to-air missiles systems and heavy radar systems. They show that there is a strong tendency for down times to follow the log-normal distribution. Several other studies on airborne radar equipment have also indicated observed repair time distributions to fit the log-normal distribution best[90], but it has been usually approximated by an exponential distribution for analytic convenience and computational purposes [91]. Since it is not always possible to describe systems' failure and repair times by an exponential distribution, this limits the applicability of Markov processes.

McGregor [60] has developed an approximation formula for reliability with repair for the system with n-identical subsystems in parallel. Arms and Goodfriend [6] presents graphical information for making reliability and maintainability analyses at both unit and system levels. Myers [66] suggests the use of Monte Carlo technique whenever the problem is extremely complex and/or experimentation is desirable but costly, and illustrates a few examples of this solution technique. However, Faragher and Watson [33] maintains that availability analyses of complex systems utilizing Monte Carlo simulation technique has revealed lack of realism because it is inflexible with respect to configuration changes, thus making it unsuitable for study of optimization of availability through unit redundancy. Some other approaches concentrate on the mathematical aspects of the simulation and neglect the engineering aspects. By incorporating engineering and mathematical analysis, he presents realistic methodology which involves engineering description of the system, formulation of the simulation model and programming it for the computer, and computer exercises and engineering analysis. Finkelstein and Schafer [36] and Wohl [94] have developed models for repairable systems using dependability as a measure of system effectiveness instead of using availability.

For analytic and computational reason, not much work has been done when failure and repair times are other than exponential. Branson and Shah [19] demonstrate the reliability analysis of two-unit redundant systems with exponential failure times and general repair-time probability laws using a semi-Markov process. Hall, Dubner, and Adler [45] have developed the reliability formulae for redundant configurations when failure times and repair times follow combinations of the exponential, Weibull and log-normal distributions. They illustrate the use of Fourier series for evaluating the inverse Laplace transformation. When failure distribution is not of an exponential form, non-Markov process or the usual definition of availability

$$A = \frac{MTBF}{MTBF + MTTR} \tag{3.1}$$

may be employed. This definition assumes a steady state condition which is of an expected value function. Though non-Markovian processes have not been studied as widely as Markov processes, Sandler [76] shows that it is often possible to treat a stochastic process of the non-Markovian type by reducing it to a Markov process by increasing the number of states, each being described by a constant transition rate. He illustrates an example for a single-unit system where the failure distribution is Gamma function

$$F(t) = 1 - e^{-\lambda t} - \lambda t e^{-\lambda t} \tag{3.2}$$

and the repair distribution is exponential.

$$G(t) = 1 - e^{-\mu t} \tag{3.3}$$

He assumes that the unit goes through two exponential phases each of length $\frac{1}{\lambda}$ since

$$\int_0^\infty e^{-\lambda t}\, dt = \frac{1}{\lambda} \tag{3.4}$$

and

$$\int_0^\infty \lambda t e^{-\lambda t}\, dt = \frac{1}{\lambda} \tag{3.5}$$

Thus, he reduces this process to a Markov process by adding one more state and obtains the steady state availability for this system. The definition given by equation (3.1) has been used as a main design criterion for maintained systems [17, 29, 40], though there is no probabilistic guarantee that specified availability value will ever be reached in practice except roughly on the average [59]. Using the definition given by equation (3.1), Wohl [94] has developed availability of a single-unit system with Weibull-distributed time to failure and repair. Martz [59] provides a definition of single-cycle availability that incorporates a probabilistic guarantee that the availability value will be reached in practice. Single-cycle availability is defined as the value A_ν such that

$$P(A \geq A_\nu) = \nu \qquad (3.6)$$

where $0 \leq \nu \leq 1$ is specified. To illustrate the use of this definition, he presents a few examples with exponential, uniform and Rayleigh distributions for failure and repair times, and shows that the median cycle availability $A_\nu = 0.5$ is equivalent to the steady state availability given by equation (3.1) in all his examples.

3.2 RELIABILITY AND AVAILABILITY MODELS FOR THE SYSTEM WITH BOTH CORRECTIVE AND PREVENTIVE MAINTENANCE

So far, in the previous section literature on reliability and availability models, only corrective maintenance has been considered. The literature on the effects of preventive maintenance in the formulation of availability model and the cost associated with it will now be discussed. The earliest approach to the planned replacement problem has been made by Campbell [20]. He has discussed the comparative advantages of replacing a number of street lamps either all at once or as they failed. Though his paper is of some interest as a precursor of many recent investigations, his problem differs from most problems of current interest in that he does not require immediate replacement to be made when a failure occurs. Welker [89] is also concerned with mass replacement, and develops a method for determining optimum replacement intervals for certain vacuum tubes. Savage [77] studies optimum block replacement policies for an infinite time span within a more general setting. However, his formulation does not seem readily applicable since he leaves the expression for the cost as a function of the replacement interval in general form. Barlow, Hunter and Proschan[10] treat a somewhat less general version of this problem by specifying a form for the cost function. The situation is described in terms of a checking problem. The optimal checking times are chosen to minimize the expected loss, given that a cost for a single check and a cost for a failure in the system in the interval between inspections. A theory of optimum sequential replacement policies for the case of a finite time horizon has been developed by Barlow and Proschan [12]. They show that for a finite time horizon there exists policies which require that after each removal the next planned replacement interval is selected to minimize expected expenditure during the remaining time, and that these policies will be more effective than a fixed replacement policy.

However, periodic or preventive maintenance policies assuming an infinite usage horizon seems to have received the most attention in the literature. When an optimum interval exists, Morse [65] shows how to determine the replacement interval minimizing expected cost per unit of time. Zelen [95] discusses that most of the periodic maintenance policies have the relation for the expected cost as a function of time

$$C(t, \delta) = C_1 E[N_1(t, \delta)] + C_2 E[N_2(t, \delta)] \quad (3.7)$$

where C_1 is the cost of a replacement or preventive maintenance, C_2 the cost of a corrective maintenance due to a failure, $N_1(t,\delta)$ the number of preventive maintenance actions or replacement in time t, $N_2(t,\delta)$ the number of failures in time t, and δ is the maintenance period which must be determined. The criterion usually chosen is to set $dD/d\delta = 0$ where

$$D(\delta) = \lim_{t \to \infty} C(t,\delta) / t \quad (3.8)$$

The existence of the limit is guaranteed by the fundamental theorems of renewal theory. A comprehensive study of this type of theory is found in [7].

Renewal theory is an application of the analysis of recurrent events to problems concerning the duration of life in aggregates of physical system. Such aggregates are sometimes referred to as self-renewing when the failure of any unit results in its replacement. The renewal density, $m(t)$, is given by

$$m(t) = f(t) + \int_0^t m(t-x)f(x)dx \tag{3.8a}$$

As shown by this equation, the probability of a renewal occurring in $[t, t+dt]$, $m(t)dt$, is the sum of the probability, $f(t)dt$, that the first renewal is in $[t, t+dt]$ and the sum over x of the probability that there is a renewal near t-x followed by a failure-time of length x. Hence, the process at time t is dependent of its past. A Markov process is defined as a stochastic process such that the conditional probability distribution for the state at any future instant, given the present state, is unaffected by any additional knowledge of the past history of the system. Hence, the future states of the process are independent of its past. Moreover, the behavior of the entire process for all values of the time parameter is studied in a Markov process, whereas the study of renewal processes is restricted to renewal points. One of the advantages of this restriction is that we do not make any assumptions regarding the behavior of the process during a renewal period. Whenever a process has the property that its present state is independent of its past, this implies that the exponential distribution describes the failure process. Whenever the failure process follows some other distribution and is time dependent, it sometimes can be approximated by the exponential distribution and can be simply analyzed as a Markovian process. If the process cannot be approximated by the exponential distribution, the renewal theory approach must be utilized.

Earlier works on restricted forms of the periodic maintenance problems are found in [86]. In a series of report, Weiss [85, 86, 87] considers the effects on system reliability and on maintenance costs of both strictly periodic and random periodic maintenance or replacement policies for an essentially infinite usage period. The operating characteristic of random periodic policies were determined by Flehinger [37]. Derman and Sacks [30] obtain the optimal replacement policy for a piece of equipment in which the decision to replace depends on the observed state of equipment deterioration at specified points in time. The derivation of an optimum periodic maintenance interval corresponding to a given finite span is basically much more difficult problem. Barlow and Proschan [13] prove the existence of such an optimal policy. Further, they carefully expose the strictly periodic and random periodic maintenance problems, and have shown that for an infinite time horizon there always exists a strictly periodic maintenance policy which is superior to a random policy [12].

In practice random preventive maintenance policy or sequential replacement policy may be quite difficult to find analytically and it is therefore of some interest to restrict our attention to the preventive maintenance policy such that the preventive maintenance is scheduled at age T and preventive maintenance is actually performed only if the system has not failed before age T. If the system has failed before age T, the system is assumed to be restored to its original good condition as a result of the corrective maintenance and the preventive maintenance is rescheduled at time T from this point. In this case, T is taken to be fixed. Bell, Kamins, and McCall [15] have investigated replacement policies for aircraft and missile parts, and have obtained specific replacement policies for parts which fail according to one of the following probability distributions : normal, log-normal, and Weibull. The relationship which gives the average hourly costs in terms of two costs, K_1 and K_2, and the failure distribution of the unit has been developed by Weissbaum [88]. His model is

$$C_A(T) = \frac{K_1 - (K_1 - K_2)G(T)}{\int_0^T G(t)dt} \tag{3.9}$$

where $C_A(A)$ = the average hourly cost,

K_1 = the total cost of an in-service failure,

K_2 = the total cost of a preventive maintenance or replacement,

$G(T)$ = the probability that a new unit will last at least T hours before failure,

T = the fixed time between preventive maintenance or replacement, and

$\int_0^T G(t)dt$ = mean interval of all maintenance requirements.

The ratio of K_1 to K_2 is the critical factor in arriving at a decision regarding preventive maintenance or replacement policy. As the ratio increases, the lowest average hourly cost is realized by replacing the unit after a short life. Welker [89] has also considered policies which minimize the average hourly operating cost on a single unit. The effects of scheduled maintenance on availability for a system composed of a similar units of which at least n out of m units must operate for the system to be functioning have been studied by Meyers and Dick [62]. Cho [25] has introduced distribution of prolongation U(x)

$$U(x) = \frac{\int_x^\infty R(s)ds}{\int_0^\infty R(s)ds} \tag{3.10}$$

where R(s) is the reliability function, and has formulated a preventive maintenance objective function which maximizes system availability.

If T_f = the mean interval of corrective maintenance

T_a = the mean interval of preventive maintenance

T_m = mean corrective maintenance time

T_p = mean preventive maintenance time

then, in general, $T_a < T_f$, $T_p < T_m$, and T_p is more likely to be nearly constant in duration than is T_m because of its scheduled nature [25]. Morse [65] has shown that an optimum T_a exists which will maximize the system availability A expressed as

$$A = (1 + \frac{aR}{1-U} + b\frac{1-R}{1-U})^{-1} \tag{3.11}$$

where $a = T_p/T_f$ and $b = T_m/T_f$, and has obtained optimum T_a by using the chart with known T_p / T_m.

Rosenheim [73] has developed an expression for mean life or mean time between unscheduled maintenance of a renewable system m(T) when preventive maintenance is scheduled every T hours.

$$m(T) = \frac{\int_0^T R(t)\,dt}{Q(T)} \qquad (3.12)$$

where T is the fixed interval for preventive maintenance, R(t) is the reliability function for the system, and Q(T) is the probability of failure. It has been shown that if redundancy exists the increase in mean life and reliability can be achieved by a preventive maintenance policy even when all units have constant failure rates [84]. According to Bazovsky [14], equation (3.12) is valid regardless of the failure distribution of units. If the renewal of the system is possible either by corrective maintenance or preventive maintenance, equation (3.12) can be applied to any failure time distributions. In this thesis, equation (3.12) is used to find the mean time between unscheduled maintenance of the system under the assumption of the following maintenance policies : The corrective maintenance policy is such that repair or replacement of units begins only after the system has failed, thus the renewal of the system is assumed to be possible as a result of the corrective maintenance. The preventive maintenance policy is such that the preventive maintenance is scheduled at age T and the preventive maintenance is actually performed only if the system has not failed before age T. If the system has failed before age T, the system is restored to its original good condition as a result of the corrective maintenance, thus the preventive maintenance is rescheduled at time T from this point.

3.3 OPTIMIZATION OF RELIABILITY AND AVAILABILITY ALLOCATION PROBLEM IN MULTISTAGE SYSTEMS

As a high degree of complexity is involved in many of the modern systems, much interest have been shown in allocating the reliability or availability parameters such as failure rates, mean time to repair, and/or preventive maintenance period to the various units that make up a system in the early stages of system design. The practical problem is to determine those parameters from a design, redesign or operating point of view such that some measures like cost or weight of the system is minimized while a system reliability or availability requirement is met.

A number of authors has discussed optimization of reliability allocation problems in multistage systems. Among them, Bellman and Dreyfus [16] applied dynamic programming for solving the problem of maximizing reliability subject to the two linear constraints of cost and weight. Kettelle [55] has developed an algorithm which utilizes dynamic programming for solving the problem of maximizing reliability subject to a single cost constraint. By extending the work of Kettelle, Proschan and Bray [71] have developed a procedure for solving the problem of maximizing reliability subject to multiple linear constraints, which is a special case of the more general problem treated by Tillman and Littschwager [83]. They investigate the reliability optimization problems which are subject to linear separable constraints by using integer programming. Tillman [81] has again employed integer programming to determine the optimum number and location of redundant units for the system which has subsystems with units, where the subsystems and the units within the same subsystem are subject to more than two modes of failure. Mizukami [64] also demonstrates the applicability of convex and integer programming to the problem of determiming optimum redundancy. He describes a design method to maximize system reliability subject to several constraints on total cost, weight, volume, etc. Rudd [74] uses dynamic programming to determine the optimal parallel redundancy of chemical processing system which maximizes the profit of the system, and illustrates an numerical example for the three-stage process system. Fan, Wang, Tillman, and Hwang [32] develop the computational procedure for the same chemical system considered by Rudd by the use of discrete maximum principle, and present numerical examples for the three-stage and eight-stage systems. Whenever the redundant units cannot be reduced to a purely parallel or series configuration in a complex system, Tillman, Hwang, Fan, and Lai [82] use the Bayes' theorem to obtain the reliability of this system, then employ the sequential unconstrained minimization technique (SUMT) for optimizing the reliability with nonlinear constraint.

Some of the relatively recent papers have treated the optimization of availability allocation problems. Goldman and Whitin [41] discuss the trade-off technique between reliability and maintainability, and show how the availability parameters consistent with minimum cost operation and the specified system availability can be calculated. Kabak [54] has used geometric programming to determine the optimal design parameters which minimize total system cost. Johnson [53] presents a methodology for finding the optimum number of redundant units. Dynamic programming is proposed for optimizing the cost function under the predetermined availability level. McNichols and Messer, Jr. [61] have developed a cost-based procedure for allocating the availability parameters to the various units of the system. The allocation problem is expressed as the minimization of the total improvement cost, subject to the constraint of meeting the system availability goal, and is solved using Lagrange multipliers method. Shershin [79] has dealt with mathematical means for optimizing the simultaneous apportionments of reliability and maintainability by means of Lagrange multipliers and dynamic programming. Wilknson and Walvekar [92] have used dynamic programming for allocating availability optimally to a multicomponent system. They determine the MTBF and MTTR which minimize the system cost under the minimum availability requirement. As an extension of this study, Lambert, Walvekar, and Hirmas [58] present a method for determining the optimum MTBF, MTTR, and the number of redundant units to use in a multistage system to achieve a given availability at minimum cost. A three-stage example is illustrated by the use of dynamic programming.

Chatterjee [24] has studied the problem of allocating the availability parameters which consist of failure and repair rate of each unit and the preventive maintenance period to each unit of the system consists of n subsystems in series where each subsystem has two identical units in parallel. Assuming exponential distribution for failure and repair times, he applied Markov process to obtain the availability expression for the two unit redundant system. Since the expression obtained by using Markov process reflects only the corrective maintenance he, under the assumption made on an intuitive basis that the decrease in the probability of the systems being down as a result of the introduction of preventive maintenance is directly proportional to the increase in the mean life achieved by introducing preventive maintenance,has developed the availability model which reflects both the corrective and preventive maintenance. His model may well be applied, however the principal assumption is based on an intuitive basis and the use of Markovian approach limits the applicability of the model. Besides, the availability model does not include the time required for the preventive maintenance. Therefore some different approaches are desired which could eliminate those difficiencies.

3.4 AVAILABILITY ALLOCATION PROBLEM IN THIS THESIS

No one in the literature reviewed has developed a mathematical availability model for the general series-parallel system, which reflects both the corrective and preventive maintenance. If the system can be restored to its original good condition after preventive maintenance action, the model is applicable regardless of the failure time distribution of each unit. In addition, no one has treated the problem of allocating the availability parameters which consists of failure rate, mean corrective maintenance time, mean preventive maintenance time, and preventive maintenance period to each unit of the system. The system considered consists of N subsystems in series and each subsystem has n_j identical units in parallel. The availability model which reflects both the corrective and preventive maintenance has been developed for the n unit redundant system which is equivalent to the subsystem in this study using the definition given by equation (2.12) assuming various probability distributions for the failure and repair times of each unit. The corrective maintenance begins only when the system fails due to the failure of all redundant units. The preventive maintenance is scheduled at age T and is actually performed only if the system has not failed before age T. If the system has failed before age T, the system is renewed as a result of the corrective maintenance and the preventive maintenance is rescheduled at time T from this point. The cost structure of the system consists of three cost components : cost for designing mean time between maintenance and mean maintenance time, cost for corrective maintenance, and cost for preventive maintenance. The availability allocation problem is to determine the optimum availability parameters which minimize the cost of the system under the constraint of the specified availability requirement for the system. Two numerical examples are shown for the system with three subsystems in series where each subsystem consists of two identical units in parallel. Exponential distributions are assumed for failure and repair times in the first example and Weibull failure time and general repair time distributions are assumed in the second example. Since the nature of both the objective function and the constraint is nonlinear, the optimization techniques used for solving these problems are the generalized reduced gradient (GRG) method and the sequential unconstrained minimization technique (SUMT).

Chapter 4

DEVELOPMENT OF THE MODEL

4.1 INCREASE IN MEAN TIME BETWEEN UNSCHEDULED MAINTENANCE OR MEAN LIFE DUE TO PREVENTIVE MAINTENANCE

The effects of the preventive maintenance policy on the mean life or mean time between unscheduled maintenance of redundant systems will be considered. The mean life of the system, m, without preventive maintenance is defined as

$$m = \int_0^\infty t f(t) dt \tag{4.1}$$

where f(t) is the failure density function of the system. It can alternatively be defined as [80]

$$m = \int_0^\infty R(t) dt \tag{4.2}$$

where R(t) is the reliability function of the system. Thus, on the average the system will fail once every m hours if failed redundant units are not replaced until system failure. However, if the preventive maintenance policy is adopted which allows for the repair or replacement of failed redundant units before the system fails, system failure can be postponed depending on how often the system is inspected and maintained if inspection reveals the presence of failed units. With this preventive maintenance policy the system will fail less frequently than it would without preventive maintenance because it is assured that after every preventive maintenance action full redundancy is restored. The mean life or the mean time between unscheduled maintenance with preventive maintenance thus becomes longer than m, and theoretically it will become infinitely long if failed redundant units are immediately replaced. The relationship between the preventive maintenance period T and the mean time between unscheduled (corrective) maintenance when preventive maintenance is scheduled at age T will now be derived.

Rosenheim [73] has shown that the mean life or the mean time between unscheduled (corrective) maintenance of a system having redundant units can be increased by scheduling preventive maintenance. To derive the general reliability and mean life equations, the following maintenance procedure is assumed : Corrective maintenance policy is such that repair or replacement begins only when the system fails due to failure of all redundant units. Preventive maintenance is scheduled at age T, starting at time 0, and is actually performed only if the system has not failed before age T. Every unit is checked, and any one which has failed is replaced by a new and statistically identical unit if the exponential failure law is assumed for all units, thus the system is restored to new condition after each preventive maintenance action. To derive the reliability function, a time period of t hours can be written as

$$t = jT + s \qquad j = 0, 1, 2, \cdots ; \; 0 \leq s \leq T \tag{4.3}$$

Let us denote the reliability function of a redundant system in which preventive maintenance is scheduled at age T by $R_T(t)$, then for a time period such that j = 1 and s = 0

$$R_T(t = T) = R(T) \tag{4.4}$$

If $j=2$ and $s=0$, the system has to operate the first T hours without failure of the system. After replacement of all failed units, another T hours of failure-free system operation is required. Hence,

$$R_T(t=2T) = R(T)R(T) = [R(T)]^2 \qquad (4.5)$$

If $0<s<T$, an additional s hours of failure-free system operation is required. Hence,

$$R_T(t=2T+s) = [R(T)]^2 R(s) \qquad (4.6)$$

In general, the reliability function of a redundant system in which preventive maintenance is scheduled at age T can be written as

$$R_T(t=jT+s) = [R(T)]^j R(s) \qquad j=0,1,2,\cdots;\ 0\leq s<T \qquad (4.7)$$

Therefore, the mean life of a redundant system in which preventive maintenance is scheduled at age T, m(T), is

$$m(T) = \int_0^\infty R_T(t)dt \qquad (4.8)$$

The integral over the range $0<t<\infty$ can be expressed as the sum of integrals over intervals of T, or

$$m(T) = \sum_{j=0}^{\infty} \int_{jT}^{(j+1)T} R_T(t)dt \qquad (4.9)$$

Since $t = jT + s$, $dt = ds$ and the limits of the integral become 0 to T.

Hence

$$m(T) = \sum_{j=0}^{\infty} \int_{o}^{T} R_{T}(t)ds$$

$$= \sum_{j=0}^{\infty} \int_{o}^{T} [R(T)]^{j} R(s)ds$$

$$= \sum_{j=0}^{\infty} [R(T)]^{j} \int_{o}^{T} R(s)ds$$

When $x < 1$

$$\sum_{j=0}^{\infty} x^{j} = \frac{1}{1-x} \tag{4.10}$$

Substitution of R(T) in place of x gives

$$\sum_{j=0}^{\infty} [R(T)]^{j} = \frac{1}{1-R(T)} \quad , \qquad R(T) < 1 \tag{4.11}$$

Therefore

$$m(T) = \frac{\int_{o}^{T} R(s)ds}{1 - R(T)} \tag{4.12}$$

If we denote the unreliability of the system by Q(T), then

$$Q(T) = 1 - R(T) \tag{4.13}$$

Using this notation, equation (4.12) can be rewritten as

$$m(T) = \frac{\int_{o}^{T} R(s)ds}{Q(T)} \tag{4.14}$$

This is the mean time between unscheduled maintenance of the redundant system in which preventive maintenance is scheduled at age T. Now, let us denote the numerator of the equation (4.14) by MTBM

$$MTBM = \int_0^T R(s)ds \qquad (4.15)$$

Then, equation (4.15) represents the mean time between both scheduled (preventive or periodic) and unscheduled (corrective) maintenance, in other words, it is the mean time at which the system is restored to its original condition [14]. Of these system maintenance actions which put the system back in a state of fully restored redundancy, 100 [Q(T)] percent are caused by unscheduled or corrective maintenance, whereas 100 [R(T)] or 100 [1 - Q(T)] percent are caused by scheduled or preventive maintenance actions. Thus, the mean time between unscheduled maintenance m(T) given by equation (4.14) is expressed as the ratio of the mean time between both scheduled and unscheduled maintenance MTBM to the fraction of maintenance caused by actual failure of the system Q(T). Similarly, since 100 R(T) percent of maintenance actions are caused by preventive maintenance, the mean time between scheduled maintenance $MTBM_s$ can be written as

$$MTBM_s = \frac{\int_0^T R(s)ds}{R(T)} \qquad (4.16)$$

An example which shows the increase in mean life that can be achieved by a preventive maintenance policy is illustrated in [84] for a system having two identical units in parallel. Each individual unit is assumed to have an exponential failure distribution with parameter λ. Preventive maintenance is scheduled at age T, starting at time 0. The reliability function of the two unit redundant system is

$$R(t) = 2e^{-\lambda t} - e^{-2\lambda t} \qquad (4.17)$$

Using equation (4.14), the mean life of the system with preventive maintenance is

$$m(T) = \frac{\int_0^T (2e^{-\lambda s} - e^{-2\lambda s})\, ds}{1 - (2e^{-\lambda T} - e^{-2\lambda T})}$$

$$= \frac{\frac{3}{2\lambda} - \frac{2}{\lambda} e^{-\lambda T} + \frac{1}{2\lambda} e^{-2\lambda T}}{1 - (2e^{-\lambda T} - e^{-2\lambda T})} \qquad (4.18)$$

If preventive maintenance is not performed, i.e., $T=\infty$, m(T) becomes

$$m(T) = \frac{3}{2\lambda} \qquad (4.19)$$

which is equivalent to m

$$m = \frac{3}{2\lambda} \qquad (4.20)$$

For the specified value of λ, $\lambda = .01$ failures/hour, the mean life of the system with preventive maintenance for the various values of T is compared below :

$T = \infty$:	m(T) = 150 hrs
T = 150hrs :	m(T) = 179 hrs

$T = 100$ hrs : $m(T) = 208$ hrs

$T = 50$ hrs : $m(T) = 304$ hrs

$T = 10$ hrs : $m(T) = 1097$ hrs

Figure 4.1[14] shows the mean time between unscheduled maintenance or mean life $m(T)$ of a system with preventive maintenance as a function of the preventive maintenance period T. The shorter is T, the longer will be the $m(T)$. Conversely, the longer T is made, the shorter becomes its $m(T)$, and in the limit, when T = infinity, $m(T)$ reduces to m.

$$m(T) = m = \int_0^\infty R(t)dt \tag{4.21}$$

For the redundant system in which failure times of each individual unit is exponentially distributed, the preventive maintenance policy can achieve an increased mean life of the system if the corrective maintenance policy is such that repair begins only when the system has failed due to failure of all redundant units. When preventive maintenance is scheduled under this corrective maintenance policy, the system might have been working with some redundant units in the failed state, and these failed units can be replaced or restored to new condition. However, if the corrective maintenance policy is to replace a failed unit the instant it fails, then the system will be always in a state of fully restored redundancy, thus the application of preventive maintenance will not increase the mean life of the system.

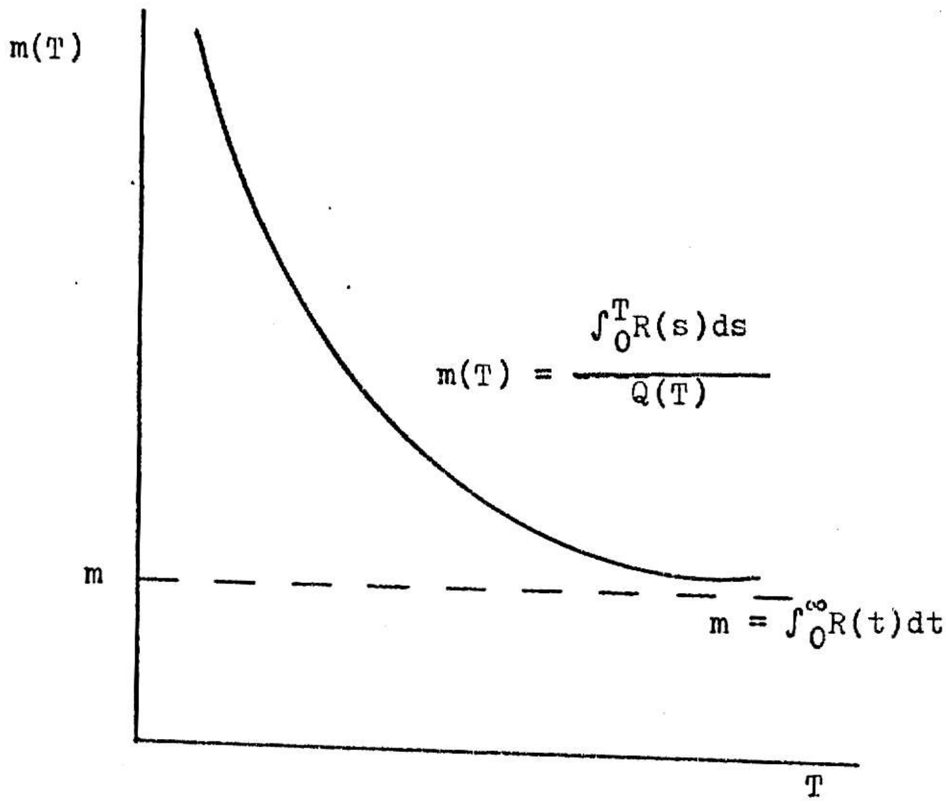

Figure 4.1. Mean time between unscheduled maintenance of a preventively maintained redundant system where scheduled preventive maintenance period is T.

Since the exponential failure law is assumed for each of the redundant units, each unit has a constant failure rate over time, i.e., the age of a unit has nothing to do with its failure rate. An old unit and a brand new one are equally likely to go on operating for some particular time period. Due to this constant failure rate characteristic the system can be in a state of its original good condition if only the failed units are replaced. As discussed above, we gain nothing by performing preventive maintenance for a single unit system having an exponential failure law since the unit we install is no better than the one we take out. This can be seen by comparing the mean life of a system with and without preventive maintenance. The reliability of a single unit system is

$$R(t) = e^{-\lambda t} \tag{4.22}$$

The mean life of a system without preventive maintenance is

$$m = \int_0^\infty e^{-\lambda t}\, dt = \frac{1}{\lambda} \tag{4.23}$$

The mean life of a system in which preventive maintenance is scheduled at age T is

$$\begin{aligned} m(T) &= \frac{\int_0^T e^{-\lambda s}\, ds}{1 - e^{-\lambda T}} \\ &= \frac{\frac{1}{\lambda}(1-e^{-\lambda T})}{1 - e^{-\lambda T}} \\ &= \frac{1}{\lambda} \end{aligned} \tag{4.24}$$

Thus, regardless of T, m(T) is always constant and is equal to m for a single unit system having an exponential failure law.

We have seen the effects of preventive maintenance on the mean time between unscheduled maintenance of the redundant system in which the redundant units have the exponential failure distribution. Although equation (4.14) is derived under the assumption of the exponential failure law for each of the redundant units, according to Bazovsky [14], it is valid regardless of the failure distribution of the redundant units if the system can be restored to its original good condition after each preventive maintenance action. For the system whose redundant units have increasing failure rates over time, if the unit is known to fail because of wearout and if it is not replaced on schedule, it will fail with a mean life equal to its mean wearout life. However, if the units are replaced on schedule before wearout can affect them, we can expect an increase in the mean life or the mean time between unscheduled maintenance. To apply equation (4.14) to the system whose redundant units have increasing failure rates over time, we assume that the corrective maintenance policy is such that replacement begins only when system fails due to failure of all the redundant units. When the redundant units have constant failure rates, the system can be restored to its original good condition only if failed units are replaced or overhauled during the preventive maintenance action. However, if the redundant units have increasing failure rates over time, we assume that the preventive maintenance policy is such that both failed and unfailed units are replaced at age T only if the system has not failed before age T, thus the system is renewed after each preventive maintenance action. In summary, the following maintenance policies will be assumed throughout this thesis : Corrective maintenance begins only when the system fails due to failure of all redundant units, thus the system is renewed after each corrective maintenance action. Preventive maintenance is scheduled at age T and is actually performed only if the system has not failed before age T. If the system has failed before age T, the system is renewed as a result of the corrective maintenance, thus the preventive maintenance is rescheduled at time T from this point. If the redundant units have constant failure rates, only failed units are replaced or overhauled, whereas if the redundant units have increasing failure rates over time, both failed and unfailed units are replaced during the preventive maintenance action. Under this preventive maintenance policy, the system can be restored to its original good condition regardless of the failure distribution of the redundant units.

4.2 MEAN MAINTENANCE TIME FOR CORRECTIVE AND PREVENTIVE MAINTENANCE

In the previous section, we have obtained the expressions for the mean time between unscheduled (corrective) maintenance $MTBM_u$ or m(T) and the mean time between scheduled (preventive) maintenance $MTBM_s$ for the redundant system in which preventive maintenance is scheduled at age T. We also have seen that the more frequently the system is scheduled for preventive maintenance, the longer will be the $MTBM_u$. Thus the probability that a system will require corrective maintenance action is reduced. If the reliability is considered as a measure of system effectiveness, then more frequent performance of the preventive maintenance will give us a higher value of the reliability. However, if the availability which takes account of the reliability as well as the maintainability is a measure of primary concern to us, then more a frequent schedule of the preventive maintenance will not necessarily give us a higher value of the availability. For the system intended for continuous service, since both the corrective and preventive maintenance actions must be taken during the duty time, the time required for both the corrective and preventive maintenance actions represents the period of a system's inoperability.

Now, let us consider the mean corrective maintenance time of the system with n identical units in parallel. If it takes t_c hours for one repairman to repair a failed unit and the corrective maintenance policy is such that repair begins only when the system fails due to failure of all redundant units, then the mean corrective maintenance time of the n unit redundant system, $\overline{M}_{ct}$, with one repairman is

$$\overline{M}_{ct} = nt_c \tag{4.25}$$

Under the same corrective maintenance policy, if n repairmen are assigned to the n unit redundant system, then the mean corrective maintenance time of the system $\overline{M}_{ct}$ is

$$\overline{M}_{ct} = t_c \tag{4.26}$$

therefore, mean corrective maintenance time of the redundant system $\overline{M}_{ct}$ can be determined by the repair time distribution of the unit and the number of repairmen.

Similarly, if the mean preventive maintenance time of a unit is t_p hours for one repairman, then the mean preventive maintenance time of the system $\overline{M}_{pt}$, with one repairman, is

$$\overline{M}_{pt} = nt_p \tag{4.27}$$

If n repairmen are ssigned to the n unit redundant system, then the mean preventive maintenance time of the system $\overline{M}_{pt}$ is

$$\bar{M}_{pt} = t_p \tag{4.28}$$

In general, t_p is less than t_c and is more likely to be nearly constant in duration than is t_c because of its schedule nature [25]. In the following section, various probability distributions will be assumed for the repair time of a unit requiring corrective maintenance. However, for the preventive maintenance time of a unit, a general repairtime distribution will be assumed.

4.3 AVAILABILITY MODEL FOR THE n - UNIT REDUNDANT SYSTEM WITH EXPONENTIAL DISTRIBUTION FOR FAILURE AND REPAIR TIMES

Let us consider a redundant system with n identical units in parallel. The system failure occurs only when all units are down. The corrective maintenance policy is such that repair or replacement begins only when the system fails due to failure of all redundant units. n repairmen are assigned to the system and every repairman is assumed to be equally capable. If the exponential distribution is assumed for the failure and repair times of each individual unit with a failure rate λ and a repair rate μ respectively, then the probability density function (pdf) for the failure time for each unit is given by

$$f(t) = \lambda e^{-\lambda t}, \qquad t > 0 \tag{4.29}$$

where $\lambda > 0$ is the constant failure rate and the pdf for the repair times of each unit is

$$g(t) = \mu e^{-\mu t}, \qquad t > 0 \tag{4.30}$$

where $\mu > 0$ is the constant repair rate.

The reliability of a unit $R_a(t)$ is

$$R_a(t) = \int_t^\infty f(s)ds$$

$$= \int_t^\infty \lambda e^{-\lambda s}\, ds = e^{-\lambda t} \tag{4.31}$$

The unreliability of a unit $Q_a(t)$ is

$$Q_a(t) = \int_o^t f(s)ds$$

$$= 1 - R_a(t) = 1 - e^{-\lambda t} \tag{4.32}$$

Now, consider a system with two identical units in parallel. Since a system failure occurs only when both units are down, the reliability of a system $R(t)$ is

$$R(t) = 1 - \text{(Probability that both units will fail)}$$

$$= 1 - Q_a(t)\, Q_a(t)$$

$$= 1 - [Q_a(t)]^2$$

$$= 1 - (1 - e^{-\lambda t})^2 = 2e^{-\lambda t} - e^{-2\lambda t} \qquad (4.33)$$

The reliability of a two-unit redundant system given by equation (4.33) can also be obtained by structuring it as a Markov process. The general concepts of a Markov process are presented and equation (4.33) is obtained by the use of the Markovian approach in Appendix A 1.2. Similarly, for a three-unit redundant system, $R(t)$ is

$$R(t) = 1 - [Q_a(t)]^3$$

$$= 1 - (1 - e^{-\lambda t})^3 \qquad (4.34)$$

In general, the reliability of a n-unit redundant system $R(t)$ is

$$R(t) = 1 - [Q_a(t)]^n$$

$$= 1 - (1 - e^{-\lambda t})^n \qquad (4.35)$$

If we denote the unreliability of a system by $Q(t)$, then

$$Q(t) = [Q_a(t)]^n = (1 - e^{-\lambda t})^n \qquad (4.36)$$

where $R(t) + Q(t) = 1$

If preventive maintenance is scheduled at age T, then the mean time between unscheduled (corrective) maintenance of the n-unit redundant system $MTBM_u$ (or $m(t)$ as defined in section 4.1) is (refer to equation (4.14))

$$\mathrm{MTBM}_u = \frac{\int_0^T R(t)dt}{Q(T)}$$

$$= \frac{\int_0^T [1 - (1 - e^{-\lambda t})^n]\, dt}{(1 - e^{-\lambda T})^n} \qquad (4.37)$$

and the mean time between scheduled (preventive) maintenance of the system MTBM_s is (refer to equation (4.16))

$$\mathrm{MTBM}_s = \frac{\int_0^T R(t)dt}{R(T)}$$

$$= \frac{\int_0^T [1 - (1 - e^{-\lambda t})^n]dt}{1 - (1 - e^{-\lambda T})^n} \qquad (4.38)$$

Hence, the mean time between maintenance or mean interval of both scheduled and unscheduled maintenance MTBM is

$$\mathrm{MTBM} = \frac{1}{1/\mathrm{MTBM}_u + 1/\mathrm{MTBM}_s}$$

$$= \int_0^T R(t)dt$$

$$= \int_0^T [1 - (1 - e^{-\lambda t})^n]dt \qquad (4.39)$$

The mean corrective maintenance time of a unit t_c is obtained from equation (4.30)

$$t_c = E(t) = \int_0^\infty tg(t)dt$$

$$= \int_0^\infty t\mu e^{-\mu t}\, dt = \frac{\Gamma(2)}{\mu} = \frac{1}{\mu} \qquad (4.40)$$

where E(t) is the expected value of repair time t. If the corrective maintenance policy is such that repair or replacement begins only when the system fails, and if n repairmen are available, then the mean corrective maintenance time of the system $\overline{M}_{ct}$ is

$$\overline{M}_{ct} = t_c = \frac{1}{\mu} \qquad (4.41)$$

If we assume a general repair-time distribution for the preventive maintenance time of each unit and denote the mean preventive maintenance time of a unit by t_p, then the mean preventive maintenance time of the system $\overline{M}_{pt}$, with n repairmen, is

$$\overline{M}_{pt} = t_p \qquad (4.42)$$

Hence, the mean corrective and preventive maintenance time $\overline{M}$, which represents all the system down-time resulting from both corrective and preventive maintenance is, (refer to equation(2.5))

$$\overline{M} = \frac{\overline{M}_{ct}\ (1/MTBM_u) + \overline{M}_{pt}\ (1/MTBM_s)}{1/MTBM_u \ + 1/MTBM_s}$$

$$= \overline{M}_{ct}\ Q(T) + \overline{M}_{pt}\ R(T)$$

$$= (\frac{1}{\mu})(1 - e^{-\lambda T})^n + t_p[1 - (1 - e^{-\lambda T})^n] \qquad (4.43)$$

Therefore, the achieved availability of the n-unit redundant system A (which is defined as A_a in Chapter 2) is (refer to equation (2.12))

$$A = \frac{MTBM}{MTBM + \overline{M}}$$

$$= \frac{\int_o^T R(t)dt}{\int_o^T R(t)dt + \overline{M}_{ct}\ Q(T) + \overline{M}_{pt}\ R(T)}$$

$$= \frac{\int_o^T[1-(1-e^{-\lambda t})^n]dt}{\int_o^T [1-(1-e^{-\lambda t})^n]dt + (\frac{1}{\mu})(1-e^{-\lambda T})^n + t_p[1-(1-e^{-\lambda T})^n]}$$

$$(4.44)$$

Equation (4.44) represents the general expression of the achieved availability for the n-unit redundant system with n repairmen when the exponential distribution is assumed for the failure and repair times of each individual unit.

Under the same corrective maintenance policy, if one repairman is assigned to the system, then

$$\bar{M}_{ct} = nt_c = \frac{n}{\mu} \tag{4.45}$$

and

$$\bar{M}_{pt} = nt_p \tag{4.46}$$

Thus, the achieved availability of the system becomes

$$A = \frac{\int_0^T [1-(1-e^{-\lambda t})^n]dt}{\int_0^T[1-(1-e^{-\lambda t})^n]dt + (\frac{n}{\mu})(1-e^{-\lambda T})^n + nt_p[1-(1-e^{-\lambda T})^n]} \tag{4.47}$$

For the evaluation of the integral term in equations (4.44) and (4.47), it is possible to expand $(1-e^{-\lambda t})^n$ using the binomial theorem.

However, especially when the failure time distribution is assumed to be other than exponential, it is difficult, if not impossible, to solve analytically. Therefore, numerical integration by the use of trapezoidal rule will be employed to evaluate this integral term (refer to Appendix A1.3).

4.4 AVAILABILITY MODEL FOR THE n-UNIT REDUNDANT SYSTEM WITH FAILURE AND REPAIR TIME DISTRIBUTIONS OTHER THAN EXPONENTIAL

Let us consider a n-unit redundant system whose redundant units have increasing failure rates over time. The assumptions on the state of system failure, corrective maintenance policy, and number of repairmen are identical with those considered in the previous section. However, since the redundant units have increasing failure rates over time, we assume that the preventive maintenance policy is such that both failed and unfailed units are replaced at age T only if the system has not failed before age T. The achieved availability of the n-unit redundant system is developed assuming the following combinations of failure time - repair time distributions : Gamma - Gamma, Weibull - Weibull, Rayleigh - Rayleigh, Normal - Normal, and Weibull - general. For the preventive maintenance time of each redundant unit, a general repair time distribution is assumed. The mean time between maintenance MTBM and mean corrective and preventive maintenance time $\bar{M}$ are derived for each failure and repair time distribution, note that other combinations can be used to derive the achieved availability of the system. In this section, however, only the above combinations will be considered.

Gamma distributions for failure and repair times

Let us consider Gamma distributions for failure and repair times of each redundant unit. The pdf for failure times of each unit is given by

$$f(t) = \frac{\lambda}{\Gamma(\alpha)}(\lambda t)^{\alpha-1}e^{-\lambda t} \qquad , \ t > 0 \qquad (4.48)$$

where $\lambda > 0$: scale parameter

$\alpha \geq 1$: shape parameter

Since we are interested in the increasing failure rate over time, we will restrict our attention to the case where $\alpha > 1$. The pdf for repair times of each unit is

$$g(t) = \frac{\mu}{\Gamma(\beta)} (\mu t)^{\beta-1} e^{-\mu t} \qquad , t > 0 \qquad (4.49)$$

where $\mu > 0$: scale parameter

$\beta \geq 1$: shape parameter

The reliability of a unit $R_a(t)$ is

$$R_a(t) = \int_t^\infty f(s)ds$$

$$= \int_t^\infty \frac{\lambda}{\Gamma(\alpha)} (\lambda s)^{\alpha-1} e^{-\lambda s} ds \qquad (4.50)$$

By transformation of variable, i.e., let

$$\lambda s = u \qquad (4.51)$$

the limits of integral become λt to ∞. Hence

$$R_a(t) = \int_{\lambda t}^\infty \frac{u^{\alpha-1} e^{-u}}{\Gamma(\alpha)} du \qquad (4.52)$$

If α is a positive integer

$$\Gamma(\alpha) = (\alpha - 1)! \tag{4.53}$$

Thus, equation (4.52) becomes

$$R_a(t) = \int_{\lambda t}^{\infty} \frac{u^{\alpha-1} e^{-u}}{(\alpha-1)!} du \tag{4.54}$$

Equation (4.54) can be rewritten as

$$(\alpha-1)!\ R_a(t) = \int_{\lambda t}^{\infty} u^{\alpha-1}\ e^{-u}\ du \tag{4.55}$$

The right hand side of equation (4.55) can be integrated by parts by letting

$$x = u^{\alpha-1} \quad , \quad dy = e^{-u}\ du \tag{4.56}$$

Then, we obtain

$$dx = (\alpha-1)u^{\alpha-2}du \quad , \quad y = -e^{-u} \tag{4.57}$$

Hence, equation (4.55) becomes

$$(\alpha-1)!R_a(t) = e^{-\lambda t}\ (\lambda t)^{\alpha-1} + (\alpha-1)\int_{\lambda t}^{\infty} e^{-u}\ u^{\alpha-2}du \tag{4.58}$$

Continuing to integrate by parts, we obtain

$$(\alpha-1)!R_a(t) = e^{-\lambda t}[(\lambda t)^{\alpha-1} + (\alpha-1)(\lambda t)^{\alpha-2} + (\alpha-1)(\alpha-2)(\lambda t)^{\alpha-3}$$

$$+ \ldots\ldots + (\alpha-1)!] \qquad (4.59)$$

Therefore

$$R_a(t) = e^{-\lambda t}\left[1 + \lambda t + \frac{(\lambda t)^2}{2!} + \ldots\ldots + \frac{(\lambda t)^{\alpha-1}}{(\alpha-1)!}\right]$$

$$= \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!} \qquad (4.60)$$

Note that equation (4.60) represents the cumulative density function (cdf) of the Poisson distribution. The reliability of the n-unit redundant system R(t) is

$$R(t) = 1 - [1 - R_a(t)]^n$$

$$= 1 - \left[1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!}\right]^n \qquad (4.61)$$

and the unreliability of the system Q(t) is

$$Q(t) = 1-R(t) = \left[1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!}\right]^n \qquad (4.62)$$

If preventive maintenance is scheduled at age T, the mean time between unscheduled maintenance of the system is (refer to equation (4.14))

$$MTBM_u = \frac{\int_o^T R(t)dt}{Q(T)}$$

$$= \frac{\int_o^T [1 - (1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!})^n] \, dt}{[1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda T}(\lambda T)^k}{k!}]^n} \qquad (4.63)$$

and the mean time between scheduled maintenance of the system is (refer to equation (4.16))

$$MTBM_s = \frac{\int_o^T R(t)dt}{R(T)}$$

$$= \frac{\int_o^T [1 - (1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!})^n]dt}{1 - [1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda T}(\lambda T)^k}{k!}]^n} \qquad (4.64)$$

The mean time between maintenance MTBM is

$$MTBM = \int_o^T R(t)dt = \int_o^T [\ 1 - (1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!})^n\ ]dt \qquad (4.65)$$

The mean corrective maintenance time of a unit t_c is

$$t_c = E(t) = \int_0^\infty tg(t)dt$$

$$= \frac{\mu^\beta}{\Gamma(\beta)} \int_0^\infty t^\beta e^{-\mu t}\, dt \qquad (4.66)$$

If we let

$$x = \mu t \qquad (4.67)$$

equation (4.66) becomes

$$t_c = \frac{\mu^\beta}{\Gamma(\beta)\ \mu^{\beta+1}} \int_0^\infty x^\beta e^{-x}\, dx$$

$$= \frac{\Gamma(\beta+1)}{\Gamma(\beta)\mu} = \frac{\beta}{\mu} \qquad (4.68)$$

The mean corrective maintenance time of the system, with n repairmen, is

$$\overline{M}_{ct} = t_c = \frac{\beta}{\mu} \qquad (4.69)$$

Since general repair time distribution is assumed for the preventive maintenance time of each unit, the mean preventive maintenance time of the system, with n repairmen, is

$$\bar{M}_{pt} = t_p \tag{4.70}$$

Using equations (2.5), (4.63), (4.64), (4.69), and (4.70), the mean corrective and preventive maintenance time of the system $\bar{M}$ is

$$\bar{M} = \left(\frac{\beta}{\mu}\right)\left[1 - \sum_{k=0}^{\alpha-1} \frac{e^{-\lambda T}(\lambda T)^k}{k!}\right]^n + t_p\left[1-\left(1-\sum_{k=0}^{\alpha-1} \frac{e^{-\lambda T}(\lambda T)^k}{k!}\right)^n\right] \tag{4.71}$$

Using equations (2.12), (4.65) and (4.71) the achieved availability of the system is obtained as

$$A = \left\{\int_0^T\left[1-\left(1-\sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!}\right)^n\right]dt\right\} \Bigg/ \left\{\int_0^T\left[1-\left(1-\sum_{k=0}^{\alpha-1} \frac{e^{-\lambda t}(\lambda t)^k}{k!}\right)^n\right]dt + \left(\frac{\beta}{\mu}\right)\left[1-\sum_{k=0}^{\alpha-1} \frac{e^{-\lambda T}(\lambda T)^k}{k!}\right]^n + t_p\left[1-\left(1-\sum_{k=0}^{\alpha-1} \frac{e^{-\lambda T}(\lambda T)^k}{k!}\right)^n\right]\right\} \tag{4.72}$$

Equation (4.72) represents the achieved availability for the n-unit redundant system with n repairmen when Gamma distributions are assumed for failure and repair times of each redundant unit.

For comparison, the achieved availability of the system with one repairman can be obtained by replacing $\frac{\beta}{\mu}$ and t_p by $\frac{n\beta}{\mu}$ and nt_p respectively in equation (4.72).

<u>Weibull distributions for failure and repair times</u>

Let us consider Weibull distributions for failure and repair times of each redundant unit. The pdf for failure times of each unit is given by

$$f(t) = (\lambda\alpha)t^{\alpha-1}\, e^{-\lambda t^{\alpha}} \quad , \quad t > 0 \qquad (4.73)$$

where $\lambda > 0$: scale parameter

$\alpha > 0$: shape parameter

The pdf for repair times of each unit is

$$g(t) = (\mu\beta)t^{\beta-1}\, e^{-\mu t^{\beta}} \quad , \quad t > 0 \qquad (4.74)$$

where $\mu > 0$: scale parameter

$\beta > 0$: shape parameter

The reliability of a unit $R_a(t)$ is

$$R_a(t) = \int_t^{\infty} (\lambda s)s^{\alpha-1}\, e^{-\lambda s^{\alpha}} ds = e^{-\lambda t^{\alpha}} \qquad (4.75)$$

The failure rate r(t) is obtained as [11]

$$r(t) = \frac{f(t)}{R_a(t)} = \lambda\alpha t^{\alpha-1} \tag{4.76}$$

Thus, if $\alpha > 1$, the failure rate increases with time.

The reliability of the n-unit redundant system is

$$R(t) = 1 - (1 - e^{-\lambda t^{\alpha}})^n \tag{4.77}$$

and the unreliability of the system is

$$Q(t) = (1 - e^{-\lambda t^{\alpha}})^n \tag{4.78}$$

The mean time between unscheduled maintenance of the system in which preventive maintenance is scheduled at age T is (refer to equation (4.14))

$$MTBM_u = \frac{\int_o^T R(t)dt}{Q(T)}$$

$$= \frac{\int_o^T [1 - (1 - e^{-\lambda t^{\alpha}})^n]dt}{(1 - e^{-\lambda T^{\alpha}})^n} \tag{4.79}$$

and the mean time between scheduled maintenance of the system is (refer to equation (4.16))

$$MTBM_s = \frac{\int_0^T R(t)dt}{R(T)}$$

$$= \frac{\int_0^T [1 - (1 - e^{-\lambda t^\alpha})^n]dt}{1 - (1 - e^{-\lambda T^\alpha})^n} \tag{4.80}$$

The mean time between maintenance is

$$MTBM = \int_0^T R(t)dt = \int_0^T [1 - (1 - e^{-\lambda t^\alpha})^n]dt \tag{4.81}$$

The mean corrective maintenance time of a unit t_c is

$$t_c = \int_0^\infty tg(t)dt = \mu^{-\frac{1}{\beta}} \Gamma(\frac{1}{\beta} + 1) \tag{4.82}$$

Thus, the mean corrective maintenance time of the system, with n repairmen, is

$$\bar{M}_{ct} = t_c = \mu^{-\frac{1}{\beta}} \Gamma(\frac{1}{\beta} + 1) \tag{4.83}$$

The mean preventive maintenance time of the system, with n repairmen, is

$$\bar{M}_{pt} = t_p \tag{4.84}$$

Using equations (2.5), (4.79), (4.80), (4.83), and (4.84), the mean corrective and preventive maintenance time of the system $\bar{M}$ is

$$\bar{M} = [\mu^{-\frac{1}{\beta}}\Gamma(\tfrac{1}{\beta} + 1)](1-e^{-\lambda T^{\alpha}})^{n} + t_{p}[1-(1-e^{-\lambda T^{\alpha}})^{n}] \tag{4.85}$$

Using equations (2.12), (4.81), and (4.85), the achieved availability for the n-unit redundant system with n repairmen when Weibull distributions are assumed for failure and repair times of each unit is

$$A = \left[\int_{o}^{T}[1-(1-e^{-\lambda t^{\alpha}})^{n}]dt\right] \Big/ \left[\int_{o}^{T}[1-(1-e^{-\lambda t^{\alpha}})^{n}]dt + [\mu^{-\frac{1}{\beta}}\Gamma(\tfrac{1}{\beta}+1)](1-e^{-\lambda T^{\alpha}})^{n} + t_{p}[1-(1-e^{-\lambda T^{\alpha}})^{n}]\right] \tag{4.86}$$

The achieved availability of the system with one repairman can be obtained by replacing t_c and t_p by nt_c and nt_p respectively in equation (4.86).

Rayleigh distributions for failure and repair times

Let us consider Rayleigh distributions for failure and repair times of each redundant unit with parameters λ and μ

respectively. The pdf for failure times of each unit is

$$f(t) = \lambda t e^{-\lambda t^2/2} \quad , \lambda > 0 \, , \, t \geq 0 \tag{4.87}$$

and the pdf for repair times of each unit is

$$g(t) = \mu t \, e^{-\mu t^2/2} \quad , \mu > 0 \, , \, t \geq 0 \tag{4.88}$$

The reliability of a unit $R_a(t)$ is

$$R_a(t) = \int_t^\infty \lambda s \, e^{-\lambda s^2/2} \, ds = e^{-\lambda t^2/2} \tag{4.89}$$

The failure rate $r(t)$ is

$$r(t) = \frac{f(t)}{R_a(t)} = \lambda t \tag{4.90}$$

Thus, the failure rate linearly increases with time for $\lambda > 0$. The reliability of the n-unit redundant system is

$$R(t) = 1 - (1 - e^{-\lambda t^2/2})^n \tag{4.91}$$

and the unreliability of the system is

$$Q(t) = (1 - e^{-\lambda t^2/2})^n \tag{4.92}$$

If preventive maintenance is scheduled at age T, using equations (4.14), (4.16), and (4.15), $MTBM_u$, $MTBM_s$ and MTBM of the system are respectively

$$MTBM_u = \frac{\int_o^T [1-(1-e^{-\lambda t^2/2})^n]dt}{(1-e^{-\lambda T^2/2})^n} \tag{4.93}$$

$$MTBM_s = \frac{\int_o^T [1-(1-e^{-\lambda t^2/2})^n]dt}{1 - (1-e^{-\lambda T^2/2})^n} \tag{4.94}$$

$$MTBM = \int_o^T [1-(1-e^{-\lambda t^2/2})^n]dt \tag{4.95}$$

The mean corrective maintenance time of the system, with n repairmen, is

$$\overline{M}_{ct} = t_c = \int_o^\infty tg(t)dt = \sqrt{\pi/(2\mu)} \tag{4.96}$$

The mean preventive maintenance time of the system, with n repairmen, is

$$\overline{M}_{pt} = t_p \tag{4.97}$$

Using equations (2.5), (4.93), (4.94), (4.96), and (4.97), the mean corrective and preventive maintenance time of the system $\overline{M}$ is

$$\bar{M} = \sqrt{\pi/(2\mu)}\ (1-e^{-\lambda T^2/2})^n + t_p[1-(1-e^{-\lambda T^2/2})^n] \qquad (4.98)$$

Using equations (2.12), (4.95), and (4.98), the achieved availability for the n-unit redundant system with n repairmen when Rayleigh distributions are assumed for failure and repair times of each unit is

$$A = \left\{\int_0^T[1-(1-e^{-\lambda t^2/2})^n]dt\right\} \Big/ \left\{\int_0^T[1-(1-e^{-\lambda t^2/2})^n]dt + \sqrt{\pi/(2\mu)}(1-e^{-\lambda T^2/2})^n + t_p[1-(1-e^{-\lambda T^2/2})^n]\right\} \qquad (4.99)$$

It is also possible to obtain the achieved availability of the system with one repairman by replacing t_c and t_p by nt_c and nt_p respectively.

Normal distributions for failure and repair times

Let us consider Normal distributions for failure and repair times of each redundant unit. The pdf for failure times of each unit is

$$f(t) = \frac{1}{\sqrt{2\pi}\ \sigma} e^{-\frac{1}{2}(\frac{t-\alpha}{\sigma})^2} \tag{4.100}$$

where α = mean and σ = standard deviation

and the pdf for repair times of each unit is

$$g(t) = \frac{1}{\sqrt{2\pi}\ \sigma'} e^{-\frac{1}{2}(\frac{t-\beta}{\sigma'})^2} \tag{4.101}$$

where β = mean and σ' = standard deviation.

The reliability of a unit $R_a(t)$ is

$$R_a(t) = \int_t^\infty f(s)ds$$

$$= 1 - \int_{-\infty}^t f(s)ds$$

$$= 1 - h(\frac{t-\alpha}{\sigma}) \tag{4.102}$$

where h is the tabulated normal cumulative distribution function.

Thus, the reliability of the n-unit redundant system is

$$R(t) = 1 - [h(\frac{t-\alpha}{\sigma})]^n \tag{4.103}$$

and the unreliability of the system Q(t) is

$$Q(t) = [h(\frac{t-\alpha}{\sigma})]^n \tag{4.104}$$

The $MTBM_u$, $MTBM_s$, and MTBM of the system in which preventive maintenance is scheduled at age T are respectively (refer to equations (4.14), (4.16), and (4.15))

$$MTBM_u = \frac{\int_o^T [1-[h\,(\frac{t-\alpha}{\sigma})]^2]dt}{[h\,(\frac{T-\alpha}{\sigma})]^n} \tag{4.105}$$

$$MTBM_s = \frac{\int_o^T [1-[h\,(\frac{t-\alpha}{\sigma})]^n]dt}{1 - [h\,(\frac{T-\alpha}{\sigma})]^n} \tag{4.106}$$

$$MTBM = \int_o^T [1-[h\,(\frac{t-\alpha}{\sigma})]^n]dt \tag{4.107}$$

The mean corrective maintenance time of the system, with n repairmen, is

$$\bar{M}_{ct} = t_c = \beta \tag{4.108}$$

The mean preventive maintenance time of the system, with n repairmen, is

$$\bar{M}_{pt} = t_p \tag{4.109}$$

The mean corrective and preventive maintenance time of the system $\bar{M}$ is (refer to equations (2.5), (4.105), (4.106), (4.108), and (4.109))

$$\bar{M} = \beta[h(\frac{T-\alpha}{\sigma})]^n + t_p[1-[h(\frac{T-\alpha}{\sigma})]^n] \quad (4.110)$$

Therefore, the achieved availability for the n-unit redundant system with n repairmen is (refer to equations (2.12), (4.107), and (4.110))

$$A = \frac{\int_o^T[1-[h(\frac{t-\alpha}{\sigma})]^n]dt}{\int_o^T[1-[h(\frac{t-\alpha}{\sigma})]^n]dt + \beta[h(\frac{T-\alpha}{\sigma})]^n + t_p[1-[h(\frac{T-\alpha}{\sigma})]^n]}$$

(4.111)

Weibull failure-time distribution and general repair time distribution.

If the Weibull distribution is assumed for failure times of each redundant unit, from equations (4.79), (4.80), and (4.81), the $MTBM_u$, $MTBM_s$, and MTBM are respectively

$$MTBM_u = \frac{\int_o^T [1-(1-e^{-\lambda t^{\alpha}})^n]dt}{(1-e^{-\lambda T^{\alpha}})^n} \quad (4.112)$$

$$MTBM_s = \frac{\int_o^T [1-(1-e^{-\lambda t^{\alpha}})^n]dt}{1-(1-e^{-\lambda T^{\alpha}})^n} \quad (4.113)$$

$$MTBM = \int_{o}^{T'} [1- (1-e^{-\lambda t^{\alpha}})^{n}]dt \tag{4.114}$$

If general repair-time distributions are assumed for both corrective and preventive maintenance times of each redundant unit, then $\overline{M}_{ct}$ and $\overline{M}_{pt}$ of the system, with n repairmen, are respectively

$$\overline{M}_{ct} = t_{c} \tag{4.115}$$

$$\overline{M}_{pt} = t_{p} \tag{4.116}$$

Using equations (2.5), (4.112), (4.113), (4.115), and (4.116), the mean corrective and preventive maintenance time of the system $\overline{M}$ is

$$\overline{M} = t_{c}(1-e^{-\lambda T^{\alpha}})^{n} + t_{p}[1-(1-e^{-\lambda T^{\alpha}})^{n}] \tag{4.117}$$

Using equations (2.12), (4.114), and (4.117), when Weibull failure-time and general repair-time distributions of each unit are assumed, the achieved availability for the n-unit redundant system with n repairmen is

$$A = \frac{\int_{o}^{T}[1-(1-e^{-\lambda t^{\alpha}})^{n}]dt}{\int_{o}^{T}[1-(1-e^{-\lambda t^{\alpha}})^{n}]dt + t_{c}(1-e^{-\lambda T^{\alpha}})^{n} + t_{p}[1-(1-e^{-\lambda T^{\alpha}})^{n}]} \tag{4.118}$$

In addition to the combinations treated in this section, it is possible to consider other combinations of failure time - repair time distributions. By using the already derived expressions for $MTBM_u$, $MTBM_s$, MTBM, $\overline{M}_{ct}$, and $\overline{M}_{pt}$ for various distributions for failure and repair times, we can obtain $\overline{M}$ and achieved availabilities for other combinations of failure and repair time distributions. The expressions for the above quantities are summarized in Table 4.1 for the combinations treated in Sections 4.3 and 4.4.

4.5 COST STRUCTURE

A fundamental objective in the building of a system is that it be capable of performing its intended function at minimum total cost. The primary reason for developing mathematical availability models for maintained systems is to compare alternate designs and select the one that best satisfies the objective. To make cost predictions, ARINC [84] suggests " (1) break the expenditures down into rather small categories, (2) collect as much past experience on expenditures in each category as possible, and (3) predict from this information how much is likely to be spent in each category for the project being costed ". Thereafter, all the categories must again be put together to obtain the total cost of the system. General cost information with regard to the reliability and the maintainability is available in [5] and [84].

Table 4.1. Summary of theexpressions for the $MTBM_u$, $MTBM_s$, MTBM, $\overline{M}_{ct}$, $\overline{M}_{pt}$, $\overline{M}$, and A for the n-unitredundant system.

		$R(t)$				$A = \dfrac{MTBM}{MTBM + \overline{M}}$
	pdf for failure times		$MTBM_u = \dfrac{\int_0^T R(t)dt}{Q(T)}$	$MTBM_s = \dfrac{\int_0^T R(t)dt}{R(T)}$	$MTBM = \int_0^T R(t)dt$	
	pdf for repair times		$\overline{M}_{ct}$	$\overline{M}_{pt}$	$\overline{M} = \dfrac{\overline{M}_{ct}(1/MTBM_u)+\overline{M}_{pt}(1/MTBM_s)}{1/MTBM_u + 1/MTBM_s}$	
Exponential	$\lambda e^{-\lambda t}$	$1-(1-e^{-\lambda t})^n$	$\dfrac{\int_0^T[1-(1-e^{-\lambda t})^n]dt}{(1-e^{-\lambda T})^n}$	$\dfrac{\int_0^T[1-(1-e^{-\lambda t})^n]dt}{1-(1-e^{-\lambda T})^n}$	$\int_0^T[1-(1-e^{-\lambda t})^n]dt$	$\dfrac{\int_0^T[1-(1-e^{-\lambda t})^n]dt}{\int_0^T[1-(1-e^{-\lambda t})^n]dt+(\frac{1}{\mu})(1-e^{-\lambda T})^n+t_p[1-(1-e^{-\lambda T})^n]}$
	$\mu e^{-\mu t}$		$\dfrac{1}{\mu}$	t_p	$(\frac{1}{\mu})(1-e^{-\lambda T})^n+t_p[1-(1-e^{-\lambda T})^n]$	
Gamma	$\dfrac{\lambda}{\Gamma(\alpha)}(\lambda t)^{\alpha-1}e^{-\lambda t}$	$1-(1-\sum_{k=0}^{\alpha-1}\dfrac{e^{-\lambda t}(\lambda t)^k}{k!})^n$	$\dfrac{\int_0^T[1-(1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda t}(\lambda t)^k}{k!})^n]dt}{[1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda T}(\lambda T)^k}{k!}]^n}$	$\dfrac{\int_0^T[1-(1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda t}(\lambda t)^k}{k!})^n]dt}{1-[1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda T}(\lambda T)^k}{k!}]^n}$	$\int_0^T[1-(1-\sum_{k=0}^{\alpha-1}\dfrac{e^{-\lambda t}(\lambda t)^k}{k!})^n]dt$	$\dfrac{\int_0^T[1-(1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda t}(\lambda t)^k}{k!})^n]dt}{\int_0^T[1-(1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda t}(\lambda t)^k}{k!})^n]dt+(\frac{\beta}{\mu})[1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda T}(\lambda T)^k}{k!}]^n+t_p[1-(1-\sum_{k=0}^{\alpha-1}\frac{e^{-\lambda T}(\lambda T)^k}{k!})^n]}$
	$\dfrac{\mu}{\Gamma(\beta)}(\mu t)^{\beta-1}e^{-\mu t}$		$\dfrac{\beta}{\mu}$	t_p	$(\frac{\beta}{\mu})[1-\sum_{k=0}^{\alpha-1}\dfrac{e^{-\lambda T}(\lambda T)^k}{k!}]^n + t_p[1-(1-\sum_{k=0}^{\alpha-1}\dfrac{e^{-\lambda T}(\lambda T)^k}{k!})^n]$	

Table 4.1. (Continued)

Weibull	$(\lambda\alpha)t^{\alpha-1}e^{-\lambda t^{\alpha}}$	$1-(1-e^{-\lambda t^{\alpha}})^n$	$\frac{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt}{(1-e^{-\lambda T^{\alpha}})^n}$	$\frac{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt}{1-(1-e^{-\lambda T^{\alpha}})^n}$	$\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt$	$\frac{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt}{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt+[\mu^{-1/\beta}\Gamma(\frac{1}{\beta}+1)](1-e^{-\lambda T^{\alpha}})^n+t_p[1-(1-e^{-\lambda T^{\alpha}})^n]}$
	$(\mu\beta)t^{\beta-1}e^{-\mu t^{\beta}}$		$\mu^{-1/\beta}\Gamma(\frac{1}{\beta}+1)$	t_p	$[\mu^{-1/\beta}\Gamma(\frac{1}{\beta}+1)](1-e^{-\lambda T^{\alpha}})^n$ $+t_p[1-(1-e^{-\lambda T^{\alpha}})^n]$	
Rayleigh	$\lambda t e^{-\lambda t^2/2}$	$1-(1-e^{-\lambda t^2/2})^n$	$\frac{\int_0^T[1-(1-e^{-\lambda t^2/2})^n]dt}{(1-e^{-\lambda T^2/2})^n}$	$\frac{\int_0^T[1-(1-e^{-\lambda t^2/2})^n]dt}{1-(1-e^{-\lambda T^2/2})^n}$	$\int_0^T[1-(1-e^{-\lambda t^2/2})^n]dt$	$\frac{\int_0^T[1-(1-e^{-\lambda t^2/2})^n]dt}{\int_0^T[1-(1-e^{-\lambda t^2/2})^n]dt+\sqrt{\pi/(2\mu)}(1-e^{-\lambda t^2/2})^n+t_p[1-(1-e^{-\lambda T^2/2})^n]}$
	$\mu t e^{-\mu t^2/2}$		$\sqrt{\pi/(2\mu)}$	t_p	$\sqrt{\pi/(2\mu)}(1-e^{-\lambda T^2/2})^n$ $+t_p[1-(1-e^{-\lambda T^2/2})^n]$	
Normal	$\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}(\frac{t-\alpha}{\sigma})^2}$	$1-[h(\frac{t-\alpha}{\sigma})]^n$	$\frac{\int_0^T[1-[h(\frac{t-\alpha}{\sigma})]^n]dt}{[h(\frac{T-\alpha}{\sigma})]^n}$	$\frac{\int_0^T[1-[h(\frac{t-\alpha}{\sigma})]^n]dt}{1-[h(\frac{T-\alpha}{\sigma})]^n}$	$\int_0^T[1-[h(\frac{t-\alpha}{\sigma})]^n]dt$	$\frac{\int_0^T[1-[h(\frac{t-\alpha}{\sigma})]^n]dt}{\int_0^T[1-[h(\frac{t-\alpha}{\sigma})]^n]dt+\beta[h(\frac{T-\alpha}{\sigma})]^n+t_p[1-[h(\frac{T-\alpha}{\sigma})]^n]}$
	$\frac{1}{\sqrt{2\pi}\sigma'}e^{-\frac{1}{2}(\frac{t-\beta}{\sigma'})^2}$		β	t_p	$\beta[h(\frac{T-\alpha}{\sigma})]^n+t_p[1-[h(\frac{T-\alpha}{\sigma})]^n$	
Weibull	$(\lambda\alpha)t^{\alpha-1}e^{-\lambda t^{\alpha}}$	$1-(1-e^{-\lambda t^{\alpha}})^n$	$\frac{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt}{(1-e^{-\lambda T^{\alpha}})^n}$	$\frac{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt}{1-(1-e^{-\lambda T^{\alpha}})^n}$	$\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt$	$\frac{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt}{\int_0^T[1-(1-e^{-\lambda t^{\alpha}})^n]dt+t_c(1-e^{-\lambda T^{\alpha}})^n+t_p[1-(1-e^{-\lambda T^{\alpha}})^n]}$
general			t_c	t_p	$t_c(1-e^{-\lambda T^{\alpha}})^n+t_p[1-(1-e^{-\lambda T^{\alpha}})^n]$	

* General repair-time distribution is assumed for mean preventive maintenance time.

Let us divide the total cost of the n-unit redundant system into three components : the cost of design for the mean time between maintenance and the mean maintenance time, the cost of corrective maintenance, and the cost of preventive maintenance. Shershin [79] suggests that such a breakdown of the cost is justified since the data for each component can be estimated. Thus, the cost functions for each component can be stated as follows :

1. the cost of design for the mean time between maintenance and the mean maintenance time, C_d, is

$$C_d = a(\mathrm{MTBM}) + \frac{b}{\overline{M}} - c \tag{4.119}$$

2. the cost of corrective maintenance, C_c, is

$$C_c = \left(\frac{z}{\mathrm{MTBM}_u} \right)(d\overline{M}_{ct})^2 \tag{4.120}$$

3. the cost of preventive maintenance, C_p, is

$$C_p = \left(\frac{z}{\mathrm{MTBM}_s} \right)(u\overline{M}_{pt} - v) \tag{4.121}$$

where MTBM_u, MTBM_s, MTBM, $\overline{M}_{ct}$, $\overline{M}_{pt}$, and $\overline{M}$ are derived in the previous section for the various probability distributions.

The parameters a, b, c, d, u, and v are cost coefficients which must be estimated from the data, and z is the total mission time of the system.

As the MTBM of the system increases, the system will operate longer without either scheduled or unscheduled down time of the system. Similarly, the decrease in $\bar{M}$ implies that the system can be repaired in a shorter time. Hence, the increase in MTBM and/or the decrease in $\bar{M}$ will require more effort in the research and development of each unit of the system. Thus, the design cost component is expected to increase as the MTBM increases and/or the $\bar{M}$ decreases. The corrective maintenance cost component decreases as the $\bar{M}_{ct}$ decreases since the system can be repaired in a shorter time as $\bar{M}_{ct}$ decreases. This cost component is weighted by the number of system failures during the total mission time z, $z/MTBM_u$. The interrelationship between corrective and preventive maintenance is reflected in this weighting factor since the length of $MTBM_u$ is affected by the preventive maintenance period T. Similarly, the preventive maintenance cost component decreases as $\bar{M}_{pt}$ decreases. This cost component is weighted by the number of preventive maintenance actions during the total mission time z, $z/MTBM_s$. If the preventive maintenance is scheduled more frequently, $MTBM_s$ will be smaller, thus this cost component will increase. To avoid duplication of the maintenance cost, it is assumed that the overlapping of the maintenance actions is negligible.

Now, consider a series-parallel system consisting of N subsystems in series where each subsystem consists of n_j identical units in parallel. Due to the series connection, the entire system is down if any one of subsystems fails. Using the subscript j, the three cost components of j^{th} subsystem can be written as

$$(C_d)_j = a_j(MTBM)_j + \frac{b_j}{(\bar{M})_j} - c_j \tag{4.122}$$

$$(C_c)_j = \frac{z}{(MTBM_u)_j} \left[d_j(\bar{M}_{ct})_j\right]^2 \tag{4.123}$$

$$(C_p)_j = \frac{z}{(MTBM_s)_j} \left[u_j(\bar{M}_{pt})_j - v_j\right] \tag{4.124}$$

Finally, the total cost of the series-parallel system, C_T, is

$$C_T = \sum_{j=1}^{N} \left[(C_d)_j + (C_c)_j + (C_p)_j\right] \tag{4.125}$$

4.6 MATHEMATICAL STATEMENT OF PROBLEM

Consider a series-parallel system with N subsystems in series where each subsystem consists of n_j identical units in parallel as shown in Figure 4.2. The subsystems are assumed to be statistically independent of each other. Due to the series connection, the entire system is down if any one of

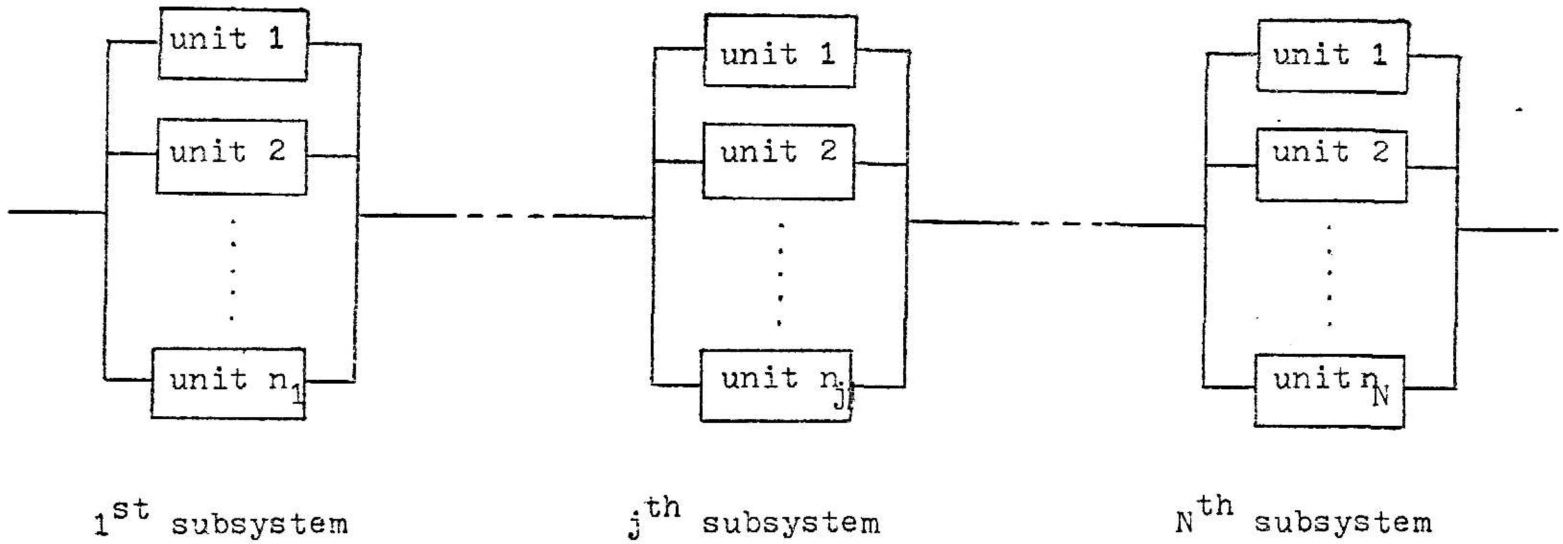

Figure 4.2. Series-parallel system with N subsystems in series where each subsystem consists of n_j identical units in parallel.

subsystems fails. The corrective maintenance policy is such that repair or replacement of each unit of the subsystem begins only when the subsystem fails due to failure of all redundant units. The preventive maintenance for the j^{th} subsystem is scheduled at age T_j and is actually performed only if the j^{th} subsystem has not failed before age T_j. If the j^{th} subsystem has failed before age T_j, this subsystem can be renewed as a result of the corrective maintenance, thus the preventive maintenance for this subsystem is rescheduled at time T_j from this point on. The number of repairmen is equal to that of units for each subsystem and they are assumed to work independently of each other. If we denote the achieved availability of the j^{th} subsystem by A_j, then the achieved availability of the series-parallel system, A_s, is expressed as

$$A_s = \prod_{j=1}^{N} A_j \tag{4.126}$$

The problem, then, is to determine T_j, $(t_p)_j$, $j=1, 2, \cdots, N$, and some particular parameters of the probability distributions for the failure and repair times of each unit for each subsystem which minimize the total cost of the system

$$C_T = \sum_{j=1}^{N} [(C_d)_j + (C_c)_j + (C_p)_j] \tag{4.127}$$

subject to

$$A_s \geq A_o \tag{4.128}$$

where A_o is the system availability requirement to be met.

Additional constraints are boundary conditions for each of the decision variables.

This optimization problem is formulated below more specifically for the combinations of exponential-exponential and Weibull-general distributions for failure time and repair time distributions.

Exponential distributions for failure and repair times

Using equations (4.39), (4.37), (4.38), (4.41), (4.42), (4.43), and (4.44), the mean time between maintenance, the mean time between unscheduled maintenance, the mean time between scheduled maintenance, the mean corrective maintenance time, the mean preventive maintenance time, the mean corrective and preventive maintenance time, and the achieved availability for the j^{th} subsystem can respectively be written as

$$(MTBM)_j = \int_0^{T_j} [1 - (1 - e^{-\lambda_j t})^{n_j}]dt \tag{4.129}$$

$$(MTBM_u)_j = \frac{(MTBM)_j}{(1 - e^{-\lambda_j T_j})^{n_j}} \tag{4.130}$$

$$(MTBM_s)_j = \frac{(MTBM)_j}{1 - (1 - e^{-\lambda_j T_j})^{n_j}} \tag{4.131}$$

$$(\overline{M}_{ct})_j = \frac{1}{\mu_j} \tag{4.132}$$

$$(\overline{M}_{pt})_j = (t_p)_j \tag{4.133}$$

$$(\overline{M})_j = (\overline{M}_{ct})_j(1-e^{-\lambda_j T_j})^{n_j} + (\overline{M}_{pt})_j[1-(1-e^{-\lambda_j T_j})^{n_j}] \tag{4.134}$$

$$A_j = \frac{(MTBM)_j}{(MTBM)_j + (\overline{M})_j} \tag{4.135}$$

By substituting equations (4.129), (4.130), (4.131), (4.132), (4.133), and (4.134) into equations (4.122), (4.123), and (4.124), the three cost components for the j^{th} subsystem $(C_d)_j$, $(C_c)_j$, and $(C_p)_j$ can respectively be obtained, where a_j, b_j, c_j, d_j, u_j, and v_j are cost coefficients for the j^{th} subsystem. Then, for the known total mission time z, the problem may be stated as follows:

Determine λ_j, μ_j, $(t_p)_j$, and T_j, $\quad j = 1, 2, \ldots, N$

which minimize the total cost of the system, C_T

$$C_T = \sum_{j=1}^{N} [(C_d)_j + (C_c)_j + (C_p)_j] \tag{4.136}$$

subject to

$$A_s = \prod_{j=1}^{N} A_j \geq A_o \tag{4.137}$$

and

$$B_j \leq \lambda_j \leq D_j\,, \qquad j = 1, 2, \cdots, N$$

$$E_j \leq \mu_j \leq F_j\,, \qquad j = 1, 2, \cdots, N$$

$$G_j \leq (t_p)_j \leq H_j\,, \qquad j = 1, 2, \cdots, N \qquad (4.138)$$

$$L_j \leq T_j \leq M_j\,, \qquad j = 1, 2, \cdots, N$$

where B_j, D_j, E_j, F_j, G_j, H_j, L_j, and M_j for $j= 1, 2, \cdots, N$ and A_o are known constants.

Weibull failure-time distribution and general repair-time distribution

Similarly, using equations (4.114), (4.112), (4.113), (4.115), (4.116), (4.117), and (4.118), MTBM, $MTBM_u$, $MTBM_s$, $\overline{M}_{ct}$, $\overline{M}_{pt}$, $\overline{M}$, and the achieved availability for th j^{th} subsystem, when the Weibull failure-time distribution and the general repair-time distribution are assumed for each unit of each subsystem, can respectively be given by

$$(MTBM)_j = \int_0^{T_j} [1 - (1-e^{-\lambda_j t^{\alpha_j}})^{n_j}]dt \qquad (4.139)$$

$$(MTBM_u)_j = \frac{(MTBM)_j}{(1 - e^{-\lambda_j T_j^{\alpha_j}})^{n_j}} \qquad (4.140)$$

$$(MTBM_s)_j = \frac{(MTBM)_j}{1 - (1-e^{-\lambda_j T_j^{\alpha_j}})^{n_j}} \qquad (4.141)$$

$$(\overline{M}_{ct})_j = (t_c)_j \tag{4.142}$$

$$(\overline{M}_{pt})_j = (t_p)_j \tag{4.143}$$

$$(\overline{M})_j = (\overline{M}_{ct})_j(1 - e^{-\lambda_j T_j^{\alpha_j}})^{n_j} + (\overline{M}_{pt})_j[1 - (1-e^{-\lambda_j T_j^{\alpha_j}})^{n_j}] \tag{4.144}$$

$$A_j = \frac{(MTBM)_j}{(MTBM)_j + (\overline{M})_j} \tag{4.145}$$

The three cost components for the j^{th} subsystem can be obtained by substituting equations (4.139), (4.140), (4.141), (4.142), (4.143), and (4.144) into equations (4.122), (4.123), and (4.124). Then, for the known total mission time z and the known shape parameters α_j, $j=1, 2, \cdots, N$, the problem may be stated as follows:

Determine λ_j, $(t_c)_j$, $(t_p)_j$, and T_j , $j=1, 2, \cdots, N$ which minimize

$$C_T = \sum_{j=1}^{N} [(C_d)_j + (C_c)_j + (C_p)_j] \tag{4.146}$$

subject to

$$A_s = \prod_{j=1}^{N} A_j \geq A_o \tag{4.147}$$

and

$$
\begin{aligned}
&B_j \le \lambda_j \le D_j \,, && j=1,2,\cdots, N \\
&E_j \le (t_c)_j \le F_j \,, && j=1,2,\cdots, N \\
&G_j \le (t_p)_j \le H_j \,, && j=1,2,\cdots, N \\
&L_j \le T_j \le M_j \,, && j=1,2,\ldots, N
\end{aligned}
\tag{4.148}
$$

where B_j, D_j, E_j, F_j, G_j, H_j, L_j, and M_j for $j=1,2,\cdots, N$ and A_o are known constants.

The optimization techniques employed for solving these problems are both the generalized reduced gradient (GRG) method and sequential unconstrained minimization technique (SUMT). The concepts and the computational procedures of the GRG and SUMT will be discussed in the following Chapter.

Chapter 5

GENERALIZED REDUCED GRADIENT (GRG) METHOD AND SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE (SUMT)

5.1 GENERALIZED REDUCED GRADIENT (GRG) METHOD

The generalized reduced gradient (GRG) method has been proposed by Abadie and Carpentier [4, 38] by extending the Wolfe reduced gradient method [2, 42]. The Wolfe method solves problems with a nonlinear objective function and linear constraints, whereas the GRG method concerns itself with the case of nonlinear constraints. The GRG method has been coded in FORTRAN by Abadie [3], Abadie and Guigou [1], and Guigou [43, 44]. Three generations of programs have been developed. The first is an experimental code called GRG 66 which is followed by the second code, GRG 69. An improved code, GREG, is the outgrowth of the first two codes and is regarded as the highly promising nonlinear programming procedure.

The general nonlinear programming problem may be stated in the form of maximize

$$f_o(\underline{X}), \qquad \underline{X} = (X_j \mid j = 1, 2, \cdots, M) \tag{5.1}$$

subject to the constraints

$$\underline{f}\,(\underline{X}) = 0, \qquad \underline{f} = (f_i \mid i = 1, 2, \cdots, m) \tag{5.2}$$

$$a_j \leq X_j \leq b_j\,, \qquad j = 1, 2, \cdots, M \tag{5.3}$$

where the underbar denotes a vector.

Note that the inequality constraints can be reduced to the equality constraints by the addition of slack variables, thus any nonlinear programming problem may be put into this form.

The GRG algorithm is based on a basic optimization procedure which transforms a constrained optimization problem into one that is unconstrained. This is accomplished by partitioning the solution vector $\underline{X}$ into m-dimensional dependent variables, $\underline{y}$ and (M-m)-dimensional independent variables, $\underline{x}$. The dependent variables, $\underline{y}$, then, are solved in terms of the independent variables, $\underline{x}$, via the constraints. If a feasible point $\underline{X}^o$ be given in such a way as to satisfy the non-degeneracy assumption, i.e., there exists a partition of $\underline{X}$ into $\underline{x}$ and $\underline{y}$ such that

$$a_j \le y_j^o \le b_j \quad , \qquad j=1, 2, \cdots, m \tag{5.4}$$

and $\partial \underline{f}/\partial \underline{y}^o$ is non-singular, the GRG algorithm may, then, be briefly summarized as follows [1,52] :

Step 1. Compute the reduced gradient, $\underline{g}^o$, and the projected reduced gradient, $\underline{p}^o$, at the starting point $\underline{X}^o = [\underline{x}^o, \underline{y}^o]$. Then, the direction of movement for the independent variable $\underline{x}$, $\underline{h}^o$, may be $\underline{h}^o = \underline{p}^o$. It may be modified by conjugate directions, where the restriction is that $\underline{h}^o \cdot \underline{p}^o > 0$.

Step 2. Compute θ which maximizes $f_o(\underline{x}^o + \theta \underline{h}^o, \underline{y}^o + \theta \underline{k}^o)$ by applying a one-dimensional search technique, where

$\underline{k}^o$ represents the direction of movement for $\underline{y}^o$.

Step 3. Compute $\tilde{\underline{x}}^1 = \underline{x}^o + \theta\underline{h}^o$ and $\tilde{\underline{y}}^1 = \underline{y}^o + \theta\underline{k}^o$, and project the values for the independent variables onto the bounds, $a_j \leq x_j \leq b_j$, $j=1, 2, \cdots, M-m$, to obtain $\underline{x}^1$. Since $\tilde{\underline{y}}^1$ usually do not satisfy the feasibility conditions, it is used as the starting point for finding $\underline{y}^1$ iteratively at Step 4.

Step 4. A feasible solution is obtained by solving $\underline{f}(\underline{x}^1, \underline{y}^1) = 0$ by an iterative method. If no speedy convergence is observed, decrease θ (for instance, set $\theta=\theta/2$) and go to Step 3. Otherwise, let $\underline{y}^1$ be the solution obtained if the new solution, $\underline{X}^1 = [\underline{x}^1, \underline{y}^1]$, improves the objective function. If the objective function is not improved, θ is reduced by $\theta/2$ and the procedure is returned to Step 3.

Step 5. Set $\underline{X}^o = \underline{X}^1$ and repeat the algorithm.

Theoretically, the stopping condition for the GRG algorithm is when $p_j^o = 0$, $j = 1, 2, \cdots, M-m$. In practice, the following stopping criteria are employed:

$$\| \underline{p}^o \| = \sqrt{\sum_{j=1}^{M-m} (p_j^o)^2} < \epsilon \tag{5.5}$$

$$p_j^o < \epsilon, \qquad j = 1, 2, \cdots, M-m \tag{5.6}$$

$$|f_o(\underline{x}^1)-f_o(\underline{x}^o)| < \epsilon \tag{5.7}$$

where $\epsilon \geq 10^{-7}$ are recommended.

Details of the GRG algorithm, computational procedures, flow diagrams, and numerical examples may be seen in [1] and [52].

The GREG program developed by Abadie and his associates of Electricité de France has been coded in FORTRAN IV. It consists of a main program, nine permanent or internal subroutines, and four user supplied external subroutines. The main program and the permanent subroutines have been compiled and stored in a partitioned data set. The four user supplied subroutines are called in the following order.

<u>Subroutine PHIX</u>

PHIX defines the objective function to the GREG program. This value is stored in the FORTRAN variable PHI, and is described in terms of the FORTRAN vector array, XC(J), J = 1, 2, ···, NV. Only the original problem variables are used to describe PHI. The code is dimensioned with the constraint of NV $\leq$ 100.

<u>Subroutine CPHI</u>

CPHI defines the inequality and/or equality constraint functions. The values are stored in the vector array VC(I), I = 1, 2, ···, NC, where NC $\leq$ 50, and in terms of the original problem variables, XC(J), J = 1, 2, ···, NV. The constraints must be ordered with inequalities first and equalities second.

Subroutine JACOB

JACOB defines the gradients of the constraint functions. The partial derivative $\partial f_i / \partial x_j$ is stored in the matrix array A(i, j). The rows of the matrix represent each constraint function, $f_i(\underline{X})$, i = 1, 2, ···, NC, in the same order as sequenced in CPHI.

Subroutine GRADFI

GRADFI defines the gradient of the objective function in terms of the array XC(J), J = 1, 2, ···, NV. The component values are stored in the vector array C(J), J = 1, 2, ···, NV.

To use the GREG program, values for nineteen parameters, a starting point, a lower bound, and an upper bound must be established. The list of parameters and their definitions are given in Table 5.1 [81]. Each parameter is given a default value which is used if it is not changed in the parameter input list. The stopping criterion is recommended to be greater than 10^{-7}. Details of the single precision arithmetic and double precision arithmetic for GRG may respectively be found in [93]and [78].

Table 5.1. Parameters

FORTRAN Program Symbols	Explanations	Default Value
1. NV	the number of original problem variables.	0
2. NIN	the number of inequality constraints.	0
3. NEG	the number of equality constraints.	0
4. NEVL	the maximum number of iterations for Newton's Method.	20*
5. NTØ (NT zero)	the maximum number of bisections in the parabolic interpolation process when maximizing a function of a single variable.	6*
6. ITET	the number of previous iterations used to help determine a maximum value for θ.	20*
7. ICONJ	equals 1 if conjugate directions are desired when maximizing, otherwise zero.	1
8. IDIAG	equals 1 if diagonal directions are desired when maximizing, otherwise zero.	0
9. ITMAX	the maximum number of iterations.	50
10. KFIL	equals 1 if the cost function is linear, otherwise, zero.	0
11. KLIN	equals 1 if <u>all</u> of the constraints are linear, otherwise, zero.	0
12. NCØ	equals zero for linear programming problems and is $\geq$ the number of constraints for nonlinear programming problems.	10

Table 5.1. (continued)

13.	ITSOR	the iteration which recording of the intermediate output starts. ($1 \leq$ ITSOR $\leq$ ITMAX)	1*
14.	ISOLSR	the solution is printed out every ISOLSR iterations. For small problems, ISOLSR = ITMAX. For large problems, ISOLSR < ITMAX.	50
15.	EPSIL	used as a criterion for the choice of a pivot in the changing and inversion of a basis. ($10^{-2} \leq$ EPSIL $\leq 10^{-1}$)	0.1E0*
16.	EPSILØ (EPSIL zero)	is used as a stopping criteria for Newton's Method. (EPSILØ $\geq 10^{-7}$)	0.1E-02
17.	EPSILI	is used as a stopping criteria if the problem is declared convex. (EPSILI $\geq 10^{-7}$)	0.1E-02
18.	EPSIL2	is used as a stopping criterion by using the norm of the reduced gradient. (EPSIL2 $\geq 10^{-7}$)	0.1E-02
19.	PC	equals zero if the problem is non-convex; equals one if the problem is convex. This parameter affects only the stopping criteria.	0.0

* - it is recommended that these values are not changed.

5.2 SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE(SUMT)

The sequential unconstrained minimization technique (SUMT) was proposed by Carroll [22, 23] and further developed by Fiacco and McCormick [34, 35]. This technique solves a constrained minimization problem by transforming it into a sequence of unconstrained minimization problems which, then, can be solved by the use of any available unconstrained minimization techniques.

The general nonlinear programming problem with nonlinear and/or linear inequality and/or equality constraints may be formulated as the problem of finding the M-dimensional column vector $\underline{X}$, $\underline{X} = (X_1, X_2, \cdots, X_M)^T$, which minimizes

$$f(\underline{X}) \tag{5.8}$$

subject to

$$g_i(\underline{X}) \geq 0, \qquad i = 1, 2, \cdots, m \tag{5.9}$$

$$h_j(\underline{X}) = 0, \qquad j = 1, 2, \cdots, l \tag{5.10}$$

where superscript T denotes transposition.

The SUMT technique for solving this problem is based on the minimization of a function

$$S(\underline{X}, r_k) = f(\underline{X}) + r_k \sum_{i=1}^{m} \frac{1}{g_i(\underline{X})} + r_k^{-1/2} \sum_{j=1}^{l} h_j^2(\underline{X}) \tag{5.11}$$

over a strictly monotonic decreasing sequence $\{r_k\}$.

Under certain restrictions, the sequence of values of the S function, $S(\underline{X}, r_k)$, is respectively minimized by a sequence of $\{\underline{X}(r_k)\}$ over a strictly monotonic decreasing sequence $\{r_k\}$, converging to the constrained optimum values of the original objective function, $f(\underline{X})$. The essential requirement is the convexity of the S function.

The intuitive concept of the S function can be described as follows:

The second term of the S function, $r_k \sum_{i=1}^{m} \frac{1}{g_i(\underline{X})}$, can be considered as a penalty factor attached to the objective function. By adding this penalty term, the minimization of S function will assure a minimum to be in the interior of the inequality constrained region. Since this term will approach infinity as the value of $\underline{X}$ approaches any one of the boundaries of the inequality constraints, $g_i(\underline{X}) \geq 0$ for $i = 1, 2, \cdots, m$, the value of $\underline{X}$ will tend to remain inside the inequality constrained feasible region. The third term of the S function, $r_k^{-1/2} \sum_{j=1}^{l} h_j^2(\underline{X})$, will approach infinity as r_k approaches zero unless $h_j(\underline{X}) = 0$ for all $j = 1, 2, \cdots, l$. Hence, this consideration will force all equality constraints to be zero.

The computational procedure is started by selecting an arbitrary starting point inside the feasible region bounded by the inequality constraints and selecting a value of r_k either arbitrarily or using the formula. Minimization of S function for the current value of r_k is made by the use of any unconstrained minimization technique (e.g., the second-order gradient method or Hooke and Jeeves pattern search method). After a minimum value of S function is reached, the value of r_k is reduced and a search is repeated starting from the previous minimum point of S function. To obtain any meaningful optimal solution, the procedure must satisfy two stopping criteria. The first criterion is needed to terminate the minimization of S function for each value of r_k. When Hooke and Jeeves pattern search method is used, this criterion is the predetermined limit, and if the step size is reduced below this limit convergence is assumed. The criteria used for the second-order gradient method may be seen in [49].

The second stopping criterion such as

$$\left| \left| \frac{f[\underline{X}(r_k)]}{G[\underline{X}(r_k)]} \right| -1 \right| < \epsilon \tag{5.12}$$

is needed for terminating overall minimization of $f[\underline{X}(r_k)]$, where the dual value, $G[\underline{X}(r_k)]$, is defined as [34]

$$G[\underline{X}(r_k)] = f[\underline{X}(r_k)] - r_k \sum_{i=1}^{m} \frac{1}{g_i(\underline{X})} + r_k^{-1/2} \sum_{j=1}^{l} h_j^2(\underline{X}) \tag{5.13}$$

In general, ϵ is ranging from 10^{-3} to 10^{-5}.

By employing a strictly monotonic decreasing sequence of $\{r_k\}$, a monotonic decreasing sequence $\{S_{min}(\underline{X}, r_k)\}$ inside the feasible region is obtained. As r_k approaches zero the second term of S function approaches zero and the equality constraints, $h_j(\underline{X}) = 0$ for $j = 1, 2, \cdots, l$, are forced to be satisfied, thus the third term of S function is forced to approach zero. Therefore, as r_k approaches zero $S(\underline{X}, r_k)$ approaches $f(\underline{X})$, where $\underline{X}$ is the optimum point which yields the minimum $S(\underline{X}. r_k)$ as well as the minimum $f(\underline{X})$. For details of the SUMT algorithm, computational procedures, flow diagrams, and numerical examples, refer to [49] and [57].

Currently available computer program for the SUMT is "RAC Computer Program Implementing the SUMT for Nonlinear Programming", IBM SHARE number 3189, developed by McCormick, Mylander, and Fiacco. This computer program uses a second-order gradient search method as the unconstrained minimization technique. To use a second-order gradient search method, one has to find the first- and second-order derivatives of the converted objective function. This, often, arises difficulties whenever the nonlinear programming problem is a highly complex one. To bypass this difficulty, a modified version was developed by Lai [57]. The modified version incorporates the Hooke and Jeeves pattern search method [47, 51] which requires no derivatives. The direction of search in the gradient method is the steepest descent direction, whereas in the Hooke and Jeeves pattern search technique it is determined by a direct comparison of two values of the objective function at two points separated from each other by a finite step. For this reason, when the pattern search is close to the boundary of inequality constraints, it falls into the infeasible region. A heuristic technique developed by Paviani and Himmelblau [68] is then used to direct the search back into the feasible region.

The program designed by Lai [93] consists of the following routines:

Main Program

Subroutine BACK - used to pull back infeasible point.

Subroutine PENAT - used to compute penalty terms.

Subroutine WEIGH - used to compute the weight of violations.

Subroutine READIN - used to read in additional data if needed.

Subroutine OUTPUT - used to print additional information if needed.

Subroutine OBRES - used to compute the objective function and constraints.

Lai's original program uses the WATFOR compiler, however, in this work some statements have been changed to use the FORTRAN H level since this compiler is faster than WATFOR compiler. The list of information which the program requires is shown in Table 5.2. If the objective function is considered to be flat, the double precision procedure is recommended. As discussed earlier the optimum $\underline{X}$ value is obtained when the S functional value approaches the f functional value. The program computes a final stopping criteria, YSTOP, at the end of each stage of the monotonically decreasing sequence of R. If YSTOP becomes less than THETA at any stage, the computation stops, and the value of $\underline{X}$ at that stage is the final optimal point.

Details of Lai's modified version may be seen in [57].

Table 5.2. List of information

FORTRAN Program Symbols	Explanations
N	total number of decision variables.
MG	total number of inequality constraints.
MH	total number of equality constraints.
R	the penalty coefficient, r_k.
RATIO	reducing rate for reducing R.
INCUT	stopping criterion for stopping each k-iteration.
THETA	final stopping criterion.
X(I)	initial starting point.
D(I)	step size in the Hooke and Jeeves pattern search.
OX(I)	estimated optimum point.
NOPM	number of input problem sets.
ITMAX	specified maximum number of calculating f-functional values within each k-iteration.
MAXP	specified maximum number of k-iterations.
ISIZE	input option code for initial step size set-up.
ICUT	input option code for the step size in each of the stage.
Y	function of X(I) for the objective function.
G(J)	function of X(I) for the j^{th} inequality constraint.
H(K)	function of X(I) for the k^{th} equality constraints

Chapter 6

NUMERICAL EXAMPLES

6.1 EXAMPLE 1 : EXPONENTIAL DISTRIBUTIONS FOR FAILURE AND REPAIR TIMES

Problem Statement

Consider a series-parallel system with three subsystems in series where each subsystem consists of two identical units in parallel. Due to the series connection, the entire system is down if any one of subsystems fails. Let the failure times and repair times of each unit of the j^{th} subsystem be exponentially distributed with failure rate λ_j and repair rate μ_j. Then, the following assumptions are made to formulate the problem :

1. The subsystems are statistically independent of each other.
2. The number of repairmen is equal to that of units for each subsystem. Every repairman is equally capable and works independently of each other.
3. The corrective maintenance policy is such that repair or replacement of each unit of the j^{th} subsystem begins only when the j^{th} subsystem fails due to failure of both redundant units. Hence, the subsystem redundancy is fully restored after the completion of the corrective maintenance action.
4. The preventive maintenance for the j^{th} subsystem is scheduled at age T_j and is actually performed only if

the j^{th} subsystem has not failed before age T_j. If the j^{th} subsystem has failed before age T_j, this subsystem can be renewed as a result of the corrective maintenance, thus the preventive maintenance for this subsystem is rescheduled at time T_j from this point. The preventive maintenance action consists of replacing or overhauling only failed units. Since redundant units have constant failure rate, the subsystem can be restored to its original good condition under this preventive maintenance policy.

5. General repair time is assumed for the mean preventive maintenance time of each unit, $(t_p)_j$, for the j^{th} subsystem.

The cost of each subsystem consists of three cost components: the cost of design for the mean time between maintenance and mean maintenance time, the cost of corrective maintenance, and the cost of preventive maintenance. The total cost of the series-parallel system is the summation of the cost of each subsystem.

The problem, then, is to determine the failure rate λ_j, the repair rate μ_j, the mean preventive maintenance time $(t_p)_j$, and the scheduled preventive maintenance period T_j, for $j = 1, 2, 3$, which minimize the total cost of the system under the constraint of the system availability requirement.

Problem Formulation

The following values are assumed for the following constants:

Number of subsystems ;

$$N = 3 \tag{6.1}$$

Number of identical units for each subsystem ;

$$n_j = 2 \quad , \qquad j = 1, 2, 3 \tag{6.2}$$

Total mission time ;

$$z = 1500. \tag{6.3}$$

System availability requirement ;

$$A_o = .97 \tag{6.4}$$

Cost coefficients for each subsystem ;

$$\begin{array}{lll} a_1 = .6 & a_2 = .5 & a_3 = .8 \\ b_1 = 400. & b_2 = 500. & b_3 = 600. \\ c_1 = 5. & c_2 = 5. & c_3 = 5. \\ d_1 = 1.8 & d_2 = 2. & d_3 = 1.7 \\ u_1 = 20. & u_2 = 15. & u_3 = 50. \\ v_1 = 3. & v_2 = 4. & v_3 = 2. \end{array} \tag{6.5}$$

Boundary values for each variable ;

$$
\begin{aligned}
&B_j = .001 && D_j = .02 && , j = 1, 2, 3 \\
&E_j = .02 && F_j = .6667 && , j = 1, 2, 3 \\
&G_j = .5 && H_j = 25. && , j = 1, 2, 3 \\
&L_j = 100. && M_j = 800. && , j = 1, 2, 3
\end{aligned}
\qquad (6.6)
$$

By substituting equations (4.129), (4.130), (4.131), (4.132), (4.133), and (4.134) with equation (6.2) and (6.3) into equations (4.122), (4.123), and (4.124), the three cost components of each subsystem $(C_d)_j$, $(C_c)_j$, and $(C_p)_j$, for $j = 1, 2, 3$, are respectively given by

$$
(C_d)_j = a_j \int_o^{T_j} [1-(1-e^{-\lambda_j t})^2] dt + \frac{b_j}{(\frac{1}{\mu_j})(1-e^{-\lambda_j T_j})^2 + (t_p)_j [1-(1-e^{-\lambda_j T_j})^2]} - c_j
$$

$$
, j = 1, 2, 3 \qquad (6.7)
$$

$$
(C_c)_j = \frac{1500}{\dfrac{\int_o^{T_j} [1-(1-e^{-\lambda_j t})^2] dt}{(1-e^{-\lambda_j T_j})^2}} [d_j (\frac{1}{\mu_j})]^2
$$

$$
, j = 1, 2, 3 \qquad (6.8)
$$

$$(C_p)_j = \frac{1500}{\dfrac{\int_0^{T_j}[1-(1-e^{-\lambda_j t})^2]dt}{1-(1-e^{-\lambda_j T_j})^2}} [u_j(t_p)_j - v_j] , \quad j=1,2,3$$

(6.9)

where the values for the cost coefficients of each subsystem, a_j, b_j, c_j, d_j, u_j, and v_j, for $j = 1, 2, 3$, are given by equation (6.5). By substituting equations (4.129) and (4.134) with equation (6.2) into equation (4.135), the achieved availability of each subsystem, A_j, $j = 1, 2, 3$, is given by

$$A_j = \left[\int_0^{T_j}[1-(1-e^{-\lambda_j t})^2]dt\right] \Bigg/ \left[\int_0^{T_j}[1-(1-e^{-\lambda_j t})^2]dt + (\frac{1}{\mu_j})(1-e^{-\lambda_j T_j})^2 + (t_p)_j[1-(1-e^{-\lambda_j T_j})^2]\right] , \quad j = 1, 2, 3$$

(6.10)

The total cost of the system, C_T, which is a function of λ_j, μ_j, $(t_p)_j$, and T_j, for $j = 1, 2, 3$, is then given by

$$C_T = \sum_{j=1}^{3}[(C_d)_j + (C_c)_j + (C_p)_j] \quad (6.11)$$

where $(C_d)_j$, $(C_c)_j$, and $(C_p)_j$ are respectively given by equations (6.7), (6.8), and (6.9).

Since the three subsystems are in series, the system is operational only when all three subsystems are operational. Hence, the achieved availability of the system, A_s, is given by

$$A_s = \prod_{j=1}^{3} A_j \tag{6.12}$$

where A_j is given by equation (6.10).

Then, for the total mission time z = 1500 hours, the problem is to determine λ_j, μ_j, $(t_p)_j$, and T_j, for j=1,2,3, which minimize the total cost of the system, C_T, given by equation (6.11) under the constraint of the system availability requirement

$$A_s \geq A_o = .97 \tag{6.13}$$

with the boundary conditions for each of variables

$$\begin{aligned} .001 &\leq \lambda_j \leq .02 \quad , \; j=1,2,3 \\ .02 &\leq \mu_j \leq .6667 \quad , \; j=1,2,3 \\ .5 &\leq (t_p)_j \leq 25. \quad , \; j=1,2,3 \\ 100. &\leq T_j \leq 800. \quad , \; j=1,2,3 \end{aligned} \tag{6.14}$$

Problem Definition for the GRG program

The nonlinear programming problem in the GRG format is stated as follows :

maximize $\quad - C_T$

subject to

$$
\begin{aligned}
.97 - A_s &\leq 0 \\
A_s - 1. &\leq 0
\end{aligned}
\tag{6.15}
$$

As discussed in section 5.1, in order to use the GREG program the individual variables are described in terms of the FORTRAN vector array XC(j), j=1,2,···,12, i.e.,

$$
\begin{aligned}
\lambda_j &= XC(j) \quad , \; j=1,2,3 \\
\mu_j &= XC(j+3) \quad , \; j=1,2,3 \\
(t_p)_j &= XC(j+6) \quad , \; j=1,2,3 \\
T_j &= XC(j+9) \quad , \; j=1,2,3
\end{aligned}
\tag{6.16}
$$

Using these original problem variables, the objective function PHI is defined in the subroutine PHIX. Since the problem must be defined in the form of maximizing the objective function, we set

$$
PHI = -C_T \tag{6.17}
$$

Constraints are defined in subroutine CPHI using vector array VC(i) , i=1,2, i.e.,

$$
VC(1) = .97 - A_s \tag{6.18}
$$

$$VC(2) = A_s - 1.$$

In subroutine JACOB, the numerical partial derivatives of the constraints with respect to each variable are defined using the matrix array A(i,j). The numerical partial derivatives of the objective function with respect to each variable are defined using vector array C(j), j=1,2,···,12, in subroutine GRADFI. The reason we take the numerical partial derivatives is due to the fact that both the objective function and the constraints are of highly nonlinear. In the data cards, the following parameter values are specified:

NV = 12

NIN = 2

ISOLSR = 1

Other parameters not listed above are given default values as shown in Table 5.1. After the parameter data, a starting point, a lower bound, and a upper bound must follow. For details of the user supplied subroutines, refer to Appendix 2.

Problem Definition for the SUMT Program

The nonlinear programming problem in the SUMT format is stated as follows :

minimize C_T

subject to

$$g(j) = \lambda_j - .001 > 0 \quad , j=1,2,3$$

$$
\begin{aligned}
g(j+3) &= .02 - \lambda_j > 0 &&, j=1,2,3\\
g(j+6) &= \mu_j - .02 > 0 &&, j=1,2,3\\
g(j+9) &= .6667 - \mu_j > 0 &&, j=1,2,3\\
g(j+12) &= (t_p)_j - .5 > 0 &&, j=1,2,3\\
g(j+15) &= 25. - (t_p)_j > 0 &&, j=1,2,3\\
g(j+18) &= T_j - 100. > 0 &&, j=1,2,3\\
g(j+21) &= 800. - T_j > 0 &&, j=1,2,3\\
g(25) &= A_s - .97 > 0 &&\\
g(26) &= 1. - A_s > 0 &&
\end{aligned}
\tag{6.19}
$$

To use the SUMT program, FORTRAN vector array X(j), j=1,2,···,12, is used to represent the individual variables, i.e.,

$$
\begin{aligned}
\lambda_j &= X(j) &&, j=1,2,3\\
\mu_j &= X(j+3) &&, j=1,2,3\\
(t_p)_j &= X(j+6) &&, j=1,2,3\\
T_j &= X(j+9) &&, j=1,2,3
\end{aligned}
\tag{6.20}
$$

The objective function and constraints are respectively defined using the FORTRAN variable Y and vector array G(J), J = 1, 2,···, 26, in subroutine OBRES. In the data cards, the following parameter values are specified :

NOPM = 1

NAME = SUMTAV

N = 12

MG = 26

MH = 0

R = 0

RATIO = 0

ITMAX = 500

INCUT = 4

THETA = 10^{-3}

MAXP = 30

ISIZE = 0

ICUT = 1

After the parameter data, a starting point, a step size, and the estimated optimum values must follow. For details of the SUMT computer program, refer to Appendix 2.

GRG Results

A GRG solution obtained by starting from a set of initial starting values, $[\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3, (t_p)_1, (t_p)_2, (t_p)_3, T_1, T_2, T_3] = [\ .005, .005, .005, .04, .04, .04, 2., 2., 2., 500., 500., 500.]$, is shown in Table 6.1a. The stopping criterion used to terminate the program is

$$\left| f_o(\bar{x}^o+\theta\bar{h}^o, \bar{y}^o+\theta\bar{k}^o) - f_o(\bar{x}^o, \bar{y}^o)\right| < 10^{-12}\left|f_o(\bar{x}^o, \bar{y}^o)\right| \tag{6.21}$$

where f_o, $\bar{x}^o$, $\bar{y}^o$, $\bar{h}^o$, $\bar{k}^o$, and θ are defined in section 5.1. It is worth noting that the first six variables, λ_j's and μ_j's, are more sensitive than the remaining variables.

Table 6.1a. GRG solution for the first set of starting values (numerical example 1)

Iteration No.	failure rate			repair rate			mean preventive maintenance time			scheduled preventive maintenance period			cost of design			cost of corrective maintenance			cost of preventive maintenance			total cost	system availability
	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_3$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
starting point	.005	.005	.005	.04	.04	.04	2.	2.	2.	500.	500.	500.	174.41	152.31	237.34	9555.39	11796.80	8523.17	32.62	-22.92	86.40	30581.35	.79423
1	.001	.001	.001	.04717	.04856	.04652	2.	2.	2.	500.	500.	500.	368.67	346.87	505.95	526,99	587.39	493.81	98.73	69.38	261.50	3914.44	.97217
2	.001	.001	.001	.05194	.05376	.05096	1.99998	1.99998	1.99996	500.	500.	500.	375.12	354.41	515.34	418.86	474.94	394.59	98.73	69.38	261.49	3446.60	.97396
3	.001	.001	.00220	.05194	.05376	.16299	1.99998	1.99998	1.99996	500.	500.	500.	365.70	341.96	476.83	586.96	676.46	178.55	98.73	69.38	200.27	2994.83	.97166
4	.001	.001	.00220	.05194	.05376	.25484	1.99998	1.99998	1.99996	500.	500.	500.	365.70	341.96	644.72	566.96	676.46	13.18	98.73	69.38	200.27	2943.16	.97639
5	.001	.001	.00392	.05231	.05417	.25484	1.99998	1.99998	1.99996	500.	500.	500.	366.26	342.72	442.28	575.12	661.77	137.20	98.73	69.38	139.33	2823.05	.97128
6	.001	.001	.00685	.14557	.15676	.26059	1.99964	1.99974	1.99964	500.	500.	500.	412.68	403.31	333.62	116.17	124.99	269.13	98.72	69.37	52.38	1820.70	.97131
7	.001	.001	.00642	.14557	.15676	.26059	1.99964	1.99974	1.99964	500.	500.	500.	425.33	419.28	334.89	74.73	79.56	265.12	98.71	69.37	52.53	1819.52	.97247
8	.001	.001	.00639	.14557	.15676	.26059	1.99964	1.99974	1.99964	500.	500.	500.	425.33	419.28	335.31	74.73	79.56	264.39	98.71	69.37	52.84	1819.51	.97251
9	.001	.001	.00629	.14559	.15678	.26064	1.99964	1.99974	1.99964	500.	500.	500.	425.33	419.28	337.14	74.72	79.55	261.20	98.71	69.37	54.17	1819.46	.97267
10	.001	.001	.00622	.14563	.15681	.26072	1.99964	1.99974	1.99963	500.	500.	500.	425.34	419.29	339.28	74.69	79.52	257.51	98.71	69.37	55.71	1819.43	.97286
11	.001	.001	.00594	.14606	.15720	.26170	1.99963	1.99975	1.99961	500.	500.	500.	425.52	419.47	348.54	74.23	79.11	241.60	98.71	69.37	62.18	1818.74	.97365
12	.001	.001	.00758	.15265	.16313	.27640	1.99951	1.99996	1.99932	500.	500.	500.	427.98	421.99	317.99	67.96	73.47	241.60	98.71	69.38	34.18	1793.68	.97094
13	.001	.001	.00689	.15265	.16313	.27640	1.99951	1.99996	1.99932	500.	500.	500.	427.98	421.99	332.63	67.96	73.47	254.77	98.71	69.38	44.14	1791.03	.97247
14	.001	.001	.00684	.15265	.16313	.27640	1.99951	1.99996	1.99932	500.	500.	500.	427.98	421.99	333.37	67.96	73.47	253.50	98.71	69.38	44.66	1791.02	.97254
15	.001	.001	.00675	.15267	.16315	.27645	1.99951	1.99996	1.99932	500.	500.	500.	427.99	421.99	335.17	67.95	73.46	250.42	98.71	69.38	45.92	1790.98	.97217
16	.001	.001	.00669	.15270	.16317	.27651	1.99951	1.99996	1.99931	500.	500.	500.	428.00	422.00	336.86	67.93	73.44	247.54	98.71	69.38	47.10	1790.96	.97287
17	.001	.001	.00653	.15284	.16331	.27686	1.99951	1.99996	1.99931	500.	500.	500.	428.04	422.05	340.30	67.83	73.35	241.71	98.71	69.38	49.46	1790.80	.97318
18	.001	.001	.00701	.15302	.16347	.27726	1.99950	1.99997	1.99930	500.	500.	500.	428.12	422.13	330.42	67.64	73.17	257.94	98.70	69.38	42.23	1789.72	.97226
19	.001	.001	.00681	.15305	.16349	.27733	1.99950	1.99997	1.99930	500.	500.	500.	428.12	422.14	333.60	67.62	73.15	252.37	98.70	69.38	44.44	1789.51	.97257

Table 6.1a. (continued)

Iteration No.	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_3$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
20	.001	.001	.00675	.15307	.16351	.27737	1.99950	1.99997	1.99930	500.	500.	500.	428.13	422.14	335.96	67.60	73.14	248.34	98.70	69.38	46.09	1789.49	.97279
21	.001	.001	.00668	.15310	.16354	.27747	1.99950	1.99997	1.99930	500.	500.	500.	428.14	422.16	337.67	63.57	73.11	245.44	98.70	69.38	47.28	1789.45	.97295
22	.001	.001	.00584	.15795	.16799	.28892	1.99941	2.00018	1.99907	500.	500.	500.	429.87	423.97	367.81	63.48	69.28	194.24	98.70	69.39	64.55	1781.29	.97549
23	.001	.001	.01354	.24108	.24428	.48495	1.99790	2.00366	1.99523	500.	500.	500.	451.68	447.40	374.51	27.25	32.76	166.21	98.62	69.53	3.04	1671.00	.97215
24	.001	.001	.01142	.24462	.24752	.49328	1.99784	2.00383	1.99506	500.	500.	500.	452.37	448.16	395.68	26.47	31.91	135.25	98.62	69.53	7.43	1665.41	.97520
25	.00266	.00211	.01060	.24462	.24752	.49328	1.99784	2.00383	1.99506	500.	500.	500.	351.06	375.37	403.53	115.58	101.04	125.41	66.84	54.47	10.39	1603.70	.97099
26	.00278	.00195	.01065	.24464	.24753	.49328	1.99784	2.00383	1.99506	500.	500.	500.	346.04	383.17	403.10	121.48	92.01	125.92	65.01	56.30	10.22	1603.23	.97098
27	.00291	.00191	.01050	.24468	.24756	.49327	1.99783	2.00383	1.99506	500.	500.	500.	339.08	387.65	404.50	129.94	87.02	124.28	62.44	57.33	10.80	1603.04	.97098
28	.00291	.00191	.01050	.24468	.24756	.49327	1.99783	2.00383	1.99506	500.	500.	500.	339.09	387.68	404.49	129.93	86.99	124.29	62.45	57.33	10.79	1602.89	.97093
Final	.00291	.00191	.01050	.24468	.24756	.49327	1.99783	2.00383	1.99506	500.	500.	500.	337.85	387.67	404.50	130.67	87.00	124.29	62.80	57.33	10.79	1602.89	.97093

If we recall that the direction of movement in the GRG is along the projected reduced gradient and the magnitude of the reduced gradient for each independent variable, i.e., the magnitude of the movement for each variable is determined by the magnitude of the partial derivatives of both objective function and constraints with respect to each variable, then this phenomenon can be explained to be caused due to the great differences between the magnitude of the numerical partial derivatives of both objective function and constraints with respect to each variable in this particular problem. To illustrate these differences, let us investigate the approximate values of the partial derivatives of objective function (f_o) and two constraints (f_1 and f_2) with respect to each variable at one particular iteration, i.e., at 19th iteration.

$$\frac{\partial f_o}{\partial \lambda_1} = -0.18\times10^5 , \quad \frac{\partial f_o}{\partial \lambda_2} = -0.28\times10^5 , \quad \frac{\partial f_o}{\partial \lambda_3} = -0.15\times10^4$$

$$\frac{\partial f_o}{\partial \mu_1} = 0.46\times10^3 , \quad \frac{\partial f_o}{\partial \mu_2} = 0.42\times10^3 , \quad \frac{\partial f_o}{\partial \mu_3} = 0.11\times10^4$$

$$\frac{\partial f_o}{\partial (t_p)_1} = -0.83\times10 , \quad \frac{\partial f_o}{\partial (t_p)_2} = 0.19\times10^2 , \quad \frac{\partial f_o}{\partial (t_p)_3} = -0.20\times10^2$$

$$\frac{\partial f_o}{\partial T_1} = -0.25 , \quad \frac{\partial f_o}{\partial T_2} = -0.22 , \quad \frac{\partial f_o}{\partial T_3} = 0.25 \tag{6.22}$$

$$\frac{\partial f_i}{\partial \lambda_1} = \pm 0.30\times 10 , \quad \frac{\partial f_i}{\partial \lambda_2} = \pm 0.28\times 10 , \quad \frac{\partial f_i}{\partial \lambda_3} = \pm 0.22\times 10$$

$$\frac{\partial f_i}{\partial \mu_1} = \mp 0.13\times 10^{-1}, \quad \frac{\partial f_i}{\partial \mu_2} = \mp 0.11\times 10^{-1}, \quad \frac{\partial f_i}{\partial \mu_3} = \mp 0.53\times 10^{-1}$$

$$\frac{\partial f_i}{\partial (t_p)_1} = \pm 0.17\times 10^{-2}, \quad \frac{\partial f_i}{\partial (t_p)_2} = \pm 0.17\times 10^{-2}, \quad \frac{\partial f_i}{\partial (t_p)_3} = \pm 0.30\times 10^{-3}$$

$$\frac{\partial f_i}{\partial T_1} = \mp 0.52\times 10^{-5}, \quad \frac{\partial f_i}{\partial T_2} = \mp 0.54\times 10^{-5}, \quad \frac{\partial f_i}{\partial T_3} = \mp 0.15\times 10^{-5} \tag{6.23}$$

where upper sign corresponds to the first constraint, i=1, and lower sign corresponds to the second constraint, i=2. The values of the partial derivatives vary from one iteration to another. However, almost the same magnitude of difference has been maintained throughout iterations. Note that the magnitudes of the partial derivatives of both objective function and constraints with respect to T_j's are negligible compared with those with respect to λ_j's and μ_j's. This is why T_j's are remained almost unchanged throughout iterations whereas λ_j's and μ_j's are relatively sensitive. This type of difficulty sometimes makes the computation inefficient and may lead the program terminated at a false optimum.

One possible alleviation from this difficulty is to employ the inverse of those sensitive variables as original problem variables. This approach is not guaranteed to work, but this has helped solve some problems of this type. Since λ_j's and μ_j's are sensitive, we employ $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ as original problem variables, thus we can expect to lessen the differences between the magnitudes of the partial derivatives. Using this method, the same set of starting points as used in Table 6.1a is again tried. The converted initial starting values are $[\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}, \frac{1}{\mu_1}, \frac{1}{\mu_2}, \frac{1}{\mu_3}, (t_p)_1, (t_p)_2, (t_p)_3, T_1, T_2, T_3] = [200., 200., 200., 25., 25., 25., 2., 2., 2., 500., 500., 500.]$. As shown in Table 6.1b, an improved solution is obtained. However, the same difficulty persists in this approach, i.e., $\frac{1}{\mu_j\text{'s}}$ and $(t_p)_j$'s appear as sensitive variables while others are remained almost unchanged.

Therefore, the fundamental alleviation from this type of difficulty is to modify the direction of movement so that each of the variables has about the same sensitivity to a given movement. Since this modification must be made within the main program stored in the computer and requires a lot of time, this is not attempted in this study. Without this modification, the only way to get an optimal solution is to try both methods as we did for the first set of starting values and select the best result as a global optimum.

To test whether or not further improved solution can be obtained, another set of starting values, $[\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3, (t_p)_1, (t_p)_2, (t_p)_3, T_1, T_2, T_3] = [.005, .004, .003, .4, .3, .4, 2., 2., 1.5, 400., 300., 300.]$, is

Table 6.1b. GRG solution for the first set of starting values (numerical example 1) ; using $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ as original problem variables

Iteration No.	failure rate λ_1	failure rate λ_2	failure rate λ_3	repair rate μ_1	repair rate μ_2	repair rate μ_3	mean preventive maintenance time $(t_p)_1$	mean preventive maintenance time $(t_p)_2$	mean preventive maintenance time $(t_p)_3$	scheduled preventive maintenance period T_1	scheduled preventive maintenance period T_2	scheduled preventive maintenance period T_3	cost of design $(C_d)_1$	cost of design $(C_d)_2$	cost of design $(C_d)_3$	cost of corrective maintenance $(C_c)_1$	cost of corrective maintenance $(C_c)_2$	cost of corrective maintenance $(C_c)_3$	cost of preventive maintenance $(C_p)_1$	cost of preventive maintenance $(C_p)_2$	cost of preventive maintenance $(C_p)_3$	total cost C_T	system availability
starting point	(200.)* .005	(200.) .005	(200.) .005	(25.) .04	(25.) .04	(25.) .04	2.	2.	2.	500.	500.	500.	174.41	152.31	237.34	9555.39	11796.80	8523.17	32.62	22.92	86.40	30581.35	.79422
1	(201.44995) .00496	(201.70576) .00496	(201.32007) .00497	(5.39353) .18534	(1.5) .66667	(7.19589) .13897	1.06388	1.15855	.50172	499.98093	499.97580	499.98842	458.42	506.86	662.31	33.45	41.12	29.90	6.38	3.21	2.09	3114.57	.98547
2	(201.49169) .00496	(201.70991) .00496	(201.38122) .00497	(3.11859) .32066	(1.5) .66667	(4.72349) .21171	.73915	1.08236	.50172	499.97961	499.97681	499.98535	455.67	485.14	658.31	33.99	41.93	30.33	6.23	8.87	20.55	1673.20	.98502
3	(201.49169) .00496	(201.70991) .00496	(201.38122) .00497	(2.49503) .40080	(1.5) .66667	(4.72349) .21171	.73915	1.08236	.50172	499.97961	499.97681	499.98535	330.42	479.07	359.18	102.64	41.93	300.96	10.49	10.91	20.53	1654.87	.97173
4	(201.49169) .00496	(201.70991) .00496	(201.38122) .00497	(2.49503) .41036	(1.5) .66667	(4.72349) .21171	.73915	1.08236	.50172	499.97961	499.97681	499.98535	341.80	479.07	359.18	89.75	41.93	300.96	10.49	10.91	20.53	1654.61	.97224
5	(201.49169) .00496	(201.70991) .00496	(201.38122) .00497	(2.41399) .41425	(1.5) .66667	(4.72349) .21171	.73915	1.08236	.50172	499.97961	499.97681	499.98535	342.65	479.07	359.18	88.88	41.93	300.96	10.49	10.91	20.53	1654.59	.97227
6	(201.48555) .00496	(201.70296) .00496	(201.38665) .00497	(1.92446) .51963	(2.24666) .44511	(3.13767) .31871	.67898	1.40755	.50172	499.98017	499.97738	499.98539	396.51	357.39	446.92	52.34	102.36	115.93	9.28	15.83	20.53	1515.73	.97663
7	(201.48467) .00496	(201.70250) .00496	(201.38581) .00497	(1.92808) .51865	(2.19887) .45478	(3.07501) .32520	.68278	1.41297	.40172	499.98026	499.97743	499.98553	388.43	371.36	436.25	56.18	90.10	127.54	9.49	15.33	20.53	1515.23	.97644
8	(201.48425) .00496	(201.70233) .00496	(201.38541) .00497	(1.92963) .51823	(2.17792) .45915	(3.04693) .32820	.68453	1.41573	.50172	499.98030	499.97746	499.98559	388.32	372.40	437.29	56.23	89.22	126.35	9.50	15.35	20.53	1515.17	.97652
9	(201.48419) .00496	(201.70230) .00496	(201.38535) .00497	(1.92985) .51817	(2.17496) .45978	(3.04281) .32864	.68480	1.41623	.5	499.98030	499.97747	499.98560	388.19	373.66	438.59	56.29	88.15	124.89	9.52	15.38	20.46	1515.12	.97661
Final	(201.48419) .00496	(201.70231) .00496	(201.38535) .00497	(1.92985) .51817	(2.17496) .45978	(3.04281) .32864	.68480	1.41623	.5	499.98030	499.97747	499.98560	388.19	373.68	438.60	56.29	88.14	124.87	9.52	15.38	20.46	1515.12	.97661

* Figures in parentheses respectively represent $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ values

tried. Table 6.2a shows the result. Using $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ as variables in place of λ_j's and μ_j's, the same set of starting points is again tried. The converted starting values are $[\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}, \frac{1}{\mu_1}, \frac{1}{\mu_2}, \frac{1}{\mu_3}, (t_p)_1, (t_p)_2, (t_p)_3, T_1, T_2, T_3]$ = [200., 250., 333.3333, 2.5, 3.3333, 2.5, 2., 2., 1.5, 400., 300., 300.] and the result is shown in Table 6.2b. Since the results obtained in Table 6.1a, 6.2a, and 6.2b are inferior to that obtained in Table 6.1b, we conclude that the solution obtained in Table 6.1b is the global optimum.

SUMT Results

To compare the SUMT results with the GRG results, the identical two sets of starting values as used for GRG are used. The result for the first set of starting values is shown in Table 6.3. Since this starting point is in infeasible region, a new feasible starting point is selected by the computer program before the minimization of S-function is started. Seven (k=7) iterations for the minimization of S-function and 3481 calculations for the objective functional values are required to reach the optimal solution. When the number of cut-down step-size operation is 4, the minimization of S-function at each k-iteration is terminated. The final stopping criterion used to terminate the program is $\epsilon = 10^{-3}$. The result for the second set of starting values is shown in Table 6.4. Five(k=5) iterations for S-function minimization and 2223 calculations for the objective functional values are required to reach the optimal solutiin.

Table 6.2a. GRG solution for the second set of starting values (numerical example 1)

Iteration No.	failure rate			repair rate			mean preventive maintenance time			scheduled preventive maintenance period			cost of design			cost of corrective maintenance			cost of preventive maintenance			total cost	system availability
	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_3$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
starting point	.005	.004	.003	.4	.3	.4	2.	2.	1.5	400.	300.	300.	312.12	301.47	524.15	91.68	138.10	37.20	56.54	84.65	276.56	1822.47	.97247
1	.00523	.00374	.00421	.40000	.30001	.39999	2.00000	2.00000	1.50000	400.	300.	300.	308.00	308.25	479.20	96.41	124.82	59.83	53.15	87.62	228.34	1745.61	.97133
2	.00539	.00355	.00504	.40000	.30002	.39999	2.00000	2.00000	1.50000	400.	300.	300.	298.42	321.91	410.36	107.42	99.12	109.34	46.77	95.22	147.11	1702.18	.96822
3	.00545	.00349	.00531	.40000	.30002	.39999	2.00000	2.00000	1.50000	400.	300.	300.	301.45	317.26	430.26	103.83	107.52	92.38	48.78	92.69	172.17	1689.51	.96933
4	.00453	.00349	.00618	.40000	.30002	.39999	2.00000	2.00000	1.50000	400.	300.	300.	323.15	314.18	422.15	80.76	113.27	98.96	63.29	90.98	162.10	1668.84	.97001
5	.00387	.00100	.00893	.40005	.30012	.40002	2.00000	1.99999	1.49998	400.	300.	300.	287.65	382.08	409.73	121.07	15.14	109.93	39.68	123.09	142.27	1618.58	.97001
6	.00472	.00100	.00823	.40006	.30012	.40007	1.99999	1.99999	1.49997	400.	300.	300.	318.89	382.08	381.75	84.96	15.14	139.59	60.42	123.09	109.17	1615.08	.97001
7	.00458	.00100	.00835	.40010	.30013	.40012	1.99999	1.99999	1.49997	400.	300.	300.	321.98	382.08	379.84	81.89	15.14	141.93	62.49	123.09	106.54	1614.97	.97001
8	.00578	.00100	.00783	.41105	.30182	.41936	1.99893	1.99987	1.49715	400.00001	300.00001	300.00004	301.50	382.24	399.57	102.62	14.97	119.78	46.22	123.08	118.11	1608.08	.97001
9	.00881	.00100	.01087	.55651	.32416	.60000	1.98492	1.99827	1.45976	400.00013	300.00011	300.00054	311.34	383.94	462.86	91.94	13.21	85.27	21.96	122.98	60.60	1553.67	.97001
10	.00732	.00100	.01236	.55653	.32417	.59999	1.98492	1.99826	1.45976	400.00013	300.00011	300.00054	329.71	384.18	451.25	73.68	12.98	97.44	29.99	122.96	43.93	1546.12	.97001
11	.00792	.00142	.01204	.55654	.32417	.59997	1.98492	1.99827	1.45975	400.00013	300.00012	300.00054	329.74	376.08	453.35	73.56	21.94	95.07	30.27	119.22	46.79	1545.87	.97001
12	.00725	.00135	.01215	.55654	.32418	.59997	1.98492	1.99827	1.45975	400.00013	300.00012	300.00054	330.51	376 19	452.77	72.90	21.81	95.72	30.58	119.27	46.00	1545.74	.97000
13	.00725	.00133	.01217	.55654	.32418	.59997	1.98492	1.99827	1.45975	400.00013	300.00012	300.00054	330.58	376.51	452.62	72.82	21.45	95.88	30.64	119.42	45.80	1545.72	.97000
14	.00725	.00133	.01217	.55654	.32418	.59997	1.98492	1.99827	1.45975	400.00013	300.00012	300.00054	330.59	376.57	452.60	72.81	21.38	95.91	30.65	119.45	45.77	1545.72	.97000
Final	.00725	.00133	.01217	.55654	.32418	.59997	1.98492	1.00827	1.45975	400.00013	300.00012	300.00054	330.59	376.58	452.59	72.81	21.38	95.91	30.65	110.45	45.76	1545.72	.97000

Table 6.2b. GRG solution for the second set of starting values (numerical example) : using $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ as original problem variables

Iteration No.	failure rate			repair rate			mean preventive maintenance time			scheduled preventive maintenance period			cost of design			cost of corrective maintenance			cost of preventive maintenance			total cost	system availability
	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_3$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
starting point	(200.)* .005	(250.) .004	(333.3333) .003	(2.5) .4	(3.3333) .3	(2.5) .4	2.	2.	1.5	400.	300.	300.	312.12	301.47	524.15	91.68	138.10	37.20	56.54	84.65	275.56	1322.47	.97247
1	(199.99700) .00500	(250.00151) .00400	(333.32831) .00300	(2.06144) .48510	(2.72640) .36678	(2.64764) .37769	1.87526	1.85609	.64434	400.00138	300.00201	300.00905	405.36	438.09	642.80	32.87	28.25	51.16	44.92	63.15	86.47	1760.40	.98229
2	(199.99210) .00500	(250.00173) .00400	(333.31795) .00300	(1.72234) .58061	(2.14981) .46516	(3.19423) .31306	1.76946	1.76851	.5	400.00298	300.00426	300.01989	405.25	430.57	553.52	32.88	27.78	90.92	45.00	67.20	86.46	1729.48	.98054
3	(199.97307) .00500	(249.99617) .00400	(333.27654) .00300	(1.91109) .52326	(1.55527) .64298	(4.74494) .21075	1.80460	2.21253	.5	400.00674	300.01005	300.05537	336.12	344.33	421.99	68.36	27.79	274.33	51.77	123.12	86.43	1684.41	.97245
4	(199.9637) .00500	(249.99539) .00400	(333.26976) .00300	(1.84601) .54171	(1.54255) .64828	(4.71863) .21193	1.77794	2.35237	.5	400.00785	300.01081	300.06196	360.35	372.55	503.29	51.19	29.57	132.15	49.86	98.52	86.45	1683.85	.97703
5	(199.96641) .00500	(249.99519) .00400	(333.26745) .00300	(1.82419) .54819	(1.54505) .64723	(4.64932) .21509	1176648	2.39765	.5	400.00830	300.01108	300.06436	364.15	367.44	505.86	49.16	29.44	129.49	49.39	102.38	86.45	1683.71	.97711
6	(199.96514) .00500	(249.99506) .00400	(333.26589) .00300	(1.81496) .55098	(1.54505) .64723	(4.60965) .21694	1.76392	2.42318	.5	400.00858	300.01127	300.06594	365.70	365.14	508.74	48.37	29.48	126.60	49.18	104.03	86.45	1683.68	.97715
7	(199.96450) .00500	(249.99500) .00400	(333.26508) .00300	(1.81135) .55207	(1.54505) .64723	(4.59108) .21781	1.76196	2.43511	.5	400.00872	300.01137	300.06677	366.57	363.59	510.83	47.95	29.48	124.56	49.06	105.21	86.45	1683.66	.97718
8	(199.96338) .00500	(249.99494) .00400	(333.26430) .00300	(1.80810) .55307	(1.54505) .64723	(4.57362) .21865	1.76015	2.44634	.5	400.00886	300.01146	300.06756	366.75	363.22	511.32	47.86	29.48	124.08	49.03	105.48	86.45	1683.66	.97718
9	(199.96318) .00500	(249.99487) .00400	(333.26341) .00300	(1.80507) .55400	(1.54505) .64723	(4.55498) .21954	1.75832	2.45832	.5	400.00901	300.01158	300.06846	367.10	362.48	512.34	47.69	29.48	123.10	48.97	106.05	86.44	1683.65	.97719
Final	(199.96318) .00500	(249.99487) .00400	(333.26341) .00300	(1.80507) .55400	(1.54505) .64723	(4.55498) .21954	1.75832	2.45832	.5	400.00901	300.01158	300.06846	367.26	362.11	512.86	47.61	29.48	122.61	48.94	106.33	86.44	1683.65	.97720

* Figures in parentheses respectively represent $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ values

Table 6.3. SUMT solution for the first set of starting values (numerical example 1)

Iteration k	cumulative No. of f-value calculations up to iteration k	value of r_k	failure rate λ_1	λ_2	λ_3	repair rate μ_1	μ_2	μ_3	mean preventive maintenance time $(t_p)_1$	$(t_p)_2$	$(t_p)_3$	scheduled preventive maintenance period T_1	T_2	T_3	cost of design $(C_d)_1$	$(C_d)_2$	$(C_d)_3$	cost of corrective maintenance $(C_c)_1$	$(C_c)_2$	$(C_c)_3$	cost of preventive maintenance $(C_p)_1$	$(C_p)_2$	$(C_p)_3$	total cost C_T	S functional value	system availability A_s
estimated optimum values			.003	.003	.003	.4	.4	.4	2.	2.	2.	400.	400.	400.												
initial step-size			.0003	.0003	.0003	.04	.04	.04	.2	.2	.2	40.	40	40.												
starting point		6.84200	.005	.005	.005	.04	.04	.04	2.	2.	2.	500.	500.	500.	174.41	152.31	237.34	9555.39	11796.80	8523.17	32.62	22.92	86.40	30531.35	35230.00	.79422
selected feasible starting point		6.84200	.0032	.0032	.0032	.28	.2	.2	1.2	1.2	1.2	340.	340.	340.	340.05	308.07	426.26	98.11	237.40	171.52	63.52	42.35	175.45	1563.00	16630.00	.97151
1	503	6.84200	.00431	.00463	.00519	.39101	.36141	.44591	1.12503	1.31907	1.48287	628.710	628.710	632.427	354.45	337.15	489.39	87.08	136.13	73.45	11.80	8.33	28.93	1526.69	9985.85	.97631
2	1004	.85520	.00709	.00661	.00739	.58985	.57721	.54788	1.51384	2.06865	1.20387	788.710	788.710	792.427	357.47	395.72	486.07	65.84	79.01	71.04	1.45	1.96	2.46	1461.02	2307.75	.97555
3	1505	.21380	.00735	.00661	.00777	.57460	.56196	.53263	1.43761	2.24442	1.12764	788.710	797.10	792.43	347.00	387.99	469.10	72.00	83.38	79.05	1.15	2.03	1.79	1443.50	1663.[illegible]	.97[illegible]
4	2006	.05345	.00840	.00645	.00789	.59534	.53722	.49054	1.15405	2.25485	.84409	796.52	797.10	796.65	340.32	378.33	441.77	76.70	89.04	94.59	.42	2.26	1.19	1424.61	1489.57	.9723[illegible]
5	2510	.01336	.00843	.00645	.00792	.59134	.53683	.49054	.98565	2.63292	.51800	796.52	797.10	796.65	338.40	377.53	441.39	78.02	89.15	94.93	.34	2.69	.69	1423.15	1440.46	.97215
6	2975	.00334	.00843	.00645	.00792	.59134	.53683	.49054	.98565	2.63292	.51800	796.52	797.10	796.65	338.40	377.53	441.39	78.02	89.15	94.93	.34	2.69	.69	1423.15	1427.43	.97215
7 (Final)	3481	.00084	.00843	.00645	.00797	.59134	.53683	.49054	.98565	2.63292	.51800	796.52	797.10	796.65	338.40	377.53	441.39	78.02	89.15	94.93	.34	2.69	.69	1423.15	1424.23	.97215

Table 6.4. SUMT solution for the second set of starting values (numerical example 1)

Iteration	cumulative No. of f-value calculations up to iteration k	value of r_k	failure rate λ_1	λ_2	λ_3	repair rate μ_1	μ_2	μ_3	mean preventive maintenance time $(t_p)_1$	$(t_p)_2$	$(t_p)_3$	scheduled preventive maintenance period T_1	T_2	T_3	cost of design $(C_d)_1$	$(C_d)_2$	$(C_d)_3$	cost of corrective maintenance $(C_c)_1$	$(C_c)_2$	$(C_c)_3$	cost of preventive maintenance $(C_p)_1$	$(C_p)_2$	$(C_p)_3$	total cost C_T	G functional value	system availability A_s
estimated optimum values			.003	.003	.003	.4	.4	.4	2.	2.	2.	400.	400.	400.												
initial step-size			.0003	.0003	.0003	.04	.04	.04	.2	.2	.2	40.	40.	40.												
starting point		.26220	.005	.004	.003	.4	.3	.4	2.	2.	1.5	400.	300.	300.	312.12	301.47	524.15	91.68	138.10	37.20	56.54	84.65	276.56	1822.47	2273.00	.97247
1	501	.26220	.00562	.00462	.00514	.43529	.40193	.42633	1.28810	2.15813	.99600	653.367	577.463	615.805	327.86	347.44	484.58	94.42	108.82	79.19	6.62	19.23	21.49	1489.65	1502.21	.97535
2	810	.03277	.00592	.00462	.00514	.44529	.40193	.42633	1.28810	2.15813	.99600	653.367	577.463	615.805	320.20	347.44	484.58	99.79	108.82	79.19	5.71	19.23	21.49	1486.45	1473.37	.97492
3	1314	.00819	.00632	.00475	.00564	.44670	.40577	.42692	1.01615	2.16784	.67764	667.157	591.398	629.595	316.28	346.59	466.94	101.55	110.34	87.54	3.26	17.17	10.57	1460.24	1470.13	.97390
4	1748	.00102	.00708	.00550	.00564	.44670	.40577	.42692	1.01615	2.16784	.67764	667.157	591.398	629.595	301.15	329.31	466.94	114.18	129.99	87.54	2.20	12.50	10.57	1454.37	1455.86	.97177
5 (Final)	2223	.00026	.00708	.00550	.00564	.44670	.40577	.42692	1.01615	2.16784	.67764	667.157	591.398	629.595	301.15	329.31	466.94	114.18	129.99	87.54	2.20	12.50	10.57	1454.37	1454.74	.97177

The same stopping criterion is applied to terminate the program. Since the solution obtained in Table 6.4 is inferior to that in Table 6.3, we conclude that the solution in Table 6.3 is the global optimum.

Comparison Between GRG and SUMT Results

Both GRG and SUMT final results for the first and second sets of starting values are respectively summarized in Table 6.5a and 6.5b. There is approximately 6% difference between the global optimum values obtained by GRG and SUMT. The difficulty persisted in GRG might have caused this difference. In the Lai's modified version of SUMT which incorporates the Hooke and Jeeves pattern search, the direction of search is determined by a direct comparison of two values of the objective function at two points separated from each other by a finite step. This requires a large number of evaluation of functional values, thus increases the computing time. However, such difficulty as persisted in GRG can be alleviated in SUMT. As far as the computing time is concerned, GRG has an advantage over SUMT as shown in Table 6.5a and 6.5b. In general, if some modifications in the main program of GRG are provided to move each variable at about the same rate, then GRG is expected to give us a further improved solution which will converge to the SUMT solution with the advantage of computing time.

6.2 EXAMPLE 2 : WEIBULL FAILURE TIME AND GENERAL REPAIR TIME DISTRIBUTIONS

Table 6.5a. Summary of GRG and SUMT final results for the first set of starting values (numerical example 1)

	failure rate			repair rate			mean preventive maintenance time			scheduled preventive maintenance time			total cost	system availability	No. of iteration	execution time(min.)
	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	C_t	A_s		
starting point	.005	.005	.005	.04	.04	.04	2.	2.	2.	500.	500.	500.	30581.35	.79422		
GRG	.00291	.00191	.01050	.24468	.24756	.49327	1.99783	2.00383	1.99506	500.	500.	500.	1602.89	.97093	28	.982
GRG(using $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ as variables)	.00496	.00496	.00497	.51817	.45978	.32864	.68480	1.41623	.5	499.98030	499.97747	499.98560	1515.12*	.97661	9	.523
SUMT	.00843	.00645	.00797	.59134	.53683	.49054	.98565	2.63292	.51800	796.52	797.10	796.65	1423.15**	.97216	k=7 (3481)	2.543

* global optimum obtained by GRG

** global optimum obtained by SUMT

Table 6.5b. Summary of GRG and SUMT final results for the second set of starting values (numerical example 1)

	failure rate			repair rate			mean preventive maintenance time			scheduled preventive maintenance time			total cost	system availability	No. of iteration	execution time (min.)
	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	C_T	A_s		
starting point	.005	.004	.003	.4	.3	.4	2.	2.	1.5	400.	300.	300.	1822.47	.97247		
GRG	.00725	.00133	.01217	.55654	.32418	.59997	1.98492	1.99827	1.45975	400.00013	300.00012	300.00054	1545.72	.97000	.14	.714
GRG (using $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ as variables)	.005	.004	.003	.55400	.64723	.21954	1.75832	2.45832	.5	400.00901	300.01158	300.06846	1683.65	.97720	9	.502
SUMT	.00708	.00550	.00564	.44670	.40577	.42692	1.01615	2.16784	.67764	667.157	591.398	629.595	1454.37	.97177	k=5 (2223)	1.759

Problem Statement

Consider the same configuration of the system as considered in section 6.1. Let Weibull failure-time distribution with scale parameter λ_j and shape parameter α_j and general repair-time distribution with mean corrective maintenance time $(t_c)_j$ be assumed for each unit of the j^{th} subsystem. Then, assumptions 1, 2, and 5 are identically made as in section 6.1. However, assumptions 3 and 4 are modified as follows :

3'. Since the failure rate increases with time, for $\alpha_j > 1$, the corrective maintenance policy is such that the replacement of each unit of the j^{th} subsystem begins only when the j^{th} subsystem fails due to failure of both redundant units. Hence, the subsystem redundancy is fully restored after the completion of the corrective maintenance action.

4'. The same preventive maintenance policy as in section 6.1 is scheduled. The preventive maintenance action, however, consists of replacing both failed and unfailed units. Under this preventive maintenance policy, the subsystem can be restored to its original good condition even if each unit of it has a increasing failure rate with time.

Then, using the same cost structure as in section 6.1, for the known total mission time z and the known shape parameter α_j, j=1,2,3, the problem is to determine the scale parameter λ_j, the mean corrective maintenance time

$(t_c)_j$, the mean preventive maintenance time $(t_p)_j$, and the scheduled preventive maintenance period T_j, for j=1,2,3, which minimize the total cost of the system under the constraint of the system availability requirement.

Problem Formulation

The values for N, n_j, j=1,2,3, and z respectively given by equations (6.1), (6.2), and (6.3) are also used in this section with the following assumed values for the following constants ; Shape parameter ;

$$\alpha_j = 2 \qquad , j=1,2,3 \tag{6.24}$$

System availability requirement ;

$$A_o = .93 \tag{6.25}$$

Cost coefficients for each subsystem ;

$$\begin{array}{lll} a_1 = 1.8 & a_2 = 1.3 & a_3 = 2. \\ b_1 = 200. & b_2 = 170. & b_3 = 250. \\ c_1 = 5. & c_2 = 5. & c_3 = 5. \\ d_1 = 2. & d_2 = 2.5 & d_3 = 3. \\ u_1 = 40. & u_2 = 100. & u_3 = 50. \\ v_1 = 3. & v_2 = 4. & v_3 = 2. \end{array} \tag{6.26}$$

Boundary values for each variable ;

$$\begin{aligned} B_j &= .0001 & D_j &= .0007 &&, \ j=1,2,3 \\ E_j &= .5 & F_j &= 20. &&, \ j=1,2,3 \\ G_j &= .1 & H_j &= 10. &&, \ j=1,2,3 \\ L_j &= 50. & M_j &= 150. &&, \ j=1,2,3 \end{aligned} \tag{6.27}$$

By substituting equations (4.139), (4.140), (4.141), (4.142), (4.143), and (4.144) with equations (6.2), (6.3), and (6.24) into equations (4.122), (4.123), and (4.124), the three cost components of each subsystem $(C_d)_j$, $(C_c)_j$, and $(C_p)_j$, for j=1,2,3, are respectively given by

$$(C_d)_j = a_j \int_o^{T_j} [1-(1-e^{-\lambda_j t^2})^2]dt + \frac{b_j}{(t_c)_j(1-e^{-\lambda_j T_j^2})^2 + (t_p)_j[1-(1-e^{-\lambda_j T_j^2})^2]} - c_j$$

$$, \ j = 1, \ 2, \ 3 \tag{6.28}$$

$$(C_c)_j = \frac{1500}{\dfrac{\int_o^{T_j}[1-(1-e^{-\lambda_j t^2})^2]dt}{(1-e^{-\lambda_j T_j^2})^2}} [d_j(t_c)_j]^2$$

$$, \ j = 1, \ 2, \ 3 \tag{6.29}$$

$$(C_p)_j = \frac{1500}{\dfrac{\int_o^{T_j}[1-(1-e^{-\lambda_j t^2})^2]dt}{1-(1-e^{-\lambda_j T_j^2})^2}} [u_j(t_p)_j - v_j]$$

$$, \ j = 1, \ 2, \ 3 \tag{6.30}$$

where the values for the cost coefficients of each subsystem, a_j, b_j, c_j, d_j, u_j, and v_j, for $j = 1, 2, 3$, are given by equation (6.26). By substituting equations (4.139) and (4.144) with equations (6.2) and (6.24) into equation (4.145), the achieved availability of each subsystem, A_j, $j = 1, 2, 3$, is given by

$$A_j = \left[\int_o^{T_j} [1-(1-e^{-\lambda_j t^2})^2]dt\right] \Big/ \left[\int_o^{T_j} [1-(1-e^{-\lambda_j t^2})^2]dt + (t_p)_j[1-(1-e^{-\lambda_j T_j^2})^2]\right] \quad , \; j = 1, 2, 3 \tag{6.31}$$

The total cost of the system, C_T, which is a function of λ_j, $(t_c)_j$, $(t_p)_j$, and T_j, for $j = 1, 2, 3$, is then given by

$$C_T = \sum_{j=1}^{3} [(C_d)_j + (C_c)_j + (C_p)_j] \tag{6.32}$$

where $(C_d)_j$, $(C_c)_j$, and $(C_p)_j$ are respectively given by equations (6.28), (6.29), and (6.30). Since the three subsystems are in series, the achieved availability of the system, A_s, is given by

$$A_s = \prod_{j=1}^{3} A_j \tag{6.33}$$

where A_j is given by equation (6.31).

Then, for the total mission time z=1500 hours and for the shape parameter $\alpha_j = 2$, $j=1,2,3$, the problem is to determine λ_j, $(t_c)_j$, $(t_p)_j$, and T_j, for $j=1,2,3$, which minimize the total cost of the system, C_T, given by equation (6.32) under the constraint of the system availability requirement

$$A_s \geq A_o = .93 \tag{6.34}$$

with the boundary conditions for each of variables

$$\begin{aligned}
.0001 &\leq \lambda_j \leq .0007 &&, j=1,2,3 \\
.5 &\leq (t_c)_j \leq 20. &&, j=1,2,3 \\
.1 &\leq (t_p)_j \leq 10. &&, j=1,2,3 \\
50. &\leq T_j \leq 150. &&, j=1,2,3
\end{aligned} \tag{6.35}$$

<u>Problem Definition for the GRG Program</u>

The problem in the GRG format is stated as follows :

maximize $\quad - C_T$

subject to

$$\begin{aligned}
.93 - A_s &\leq 0 \\
A_s - 1. &\leq 0
\end{aligned} \tag{6.36}$$

To use the GREG program, the individual variables are described in terms of the array XC(j), j=1,2,$\cdots$, 12, i.e.,

$$\begin{aligned}
\lambda_j &= XC(j) &&, j=1,2,3 \\
(t_c)_j &= XC(j+3) &&, j=1,2,3 \\
(t_p)_j &= XC(j+6) &&, j=1,2,3
\end{aligned}$$

$$T_j = XC(j+9) \qquad ,j=1,2,3 \qquad (6.37)$$

Using these original problem variables, the objective function, the constraints, the partial derivatives of the objective function, and the partial derivatives of the constraints are similarly defined as in section 6.1. The same parameter values as specified in section 6.1 are used.

Problem Definition for the SUMT Program

The problem in the SUMT format is stated as follows :

minimize C_T

subject to

$$
\begin{aligned}
g(j) &= \lambda_j - .0001 > 0 && ,j=1,2,3 \\
g(j+3) &= .0007 - \lambda_j > 0 && ,j=1,2,3 \\
g(j+6) &= (t_c)_j - .5 > 0 && ,j=1,2,3 \\
g(j+9) &= 20. - (t_c)_j > 0 && ,j=1,2,3 \\
g(j+12) &= (t_p)_j - .1 > 0 && ,j=1,2,3 \\
g(j+15) &= 10. - (t_p)_j > 0 && ,j=1,2,3 \\
g(j+18) &= T_j - 50. > 0 && ,j=1,2,3 \\
g(j+21) &= 150. - T_j > 0 && ,j=1,2,3 \\
g(25) &= A_s - .93 > 0 && \\
g(26) &= 1. - A_s > 0 &&
\end{aligned}
\qquad (6.38)
$$

To use the SUMT program, $X(j)$, $j=1,2,\cdots, 12$, is used to describe the individual variables, i.e.,

$$
\begin{aligned}
\lambda_j &= X(j) && , j=1,2,3 \\
(t_c)_j &= X(j+3) && , j=1,2,3 \\
(t_p)_j &= X(j+6) && , j=1,2,3
\end{aligned}
$$

$$T_j = X(j+9) \qquad , j = 1, 2, 3 \qquad (6.39)$$

The same parameter values as specified in section 6.1 are used. With these informations the problem can, similarly, be defined in the SUMT format as in section 6.1.

GRG Results

A GRG solution for a set of starting values, $[\lambda_1, \lambda_2, \lambda_3, (t_c)_1, (t_c)_2, (t_c)_3, (t_p)_1, (t_p)_2, (t_p)_3, T_1, T_2, T_3]$ $= [.0002, .0002, .0002, 2., 2., 2., 1., 1., 1., 100., 100., 100.]$, is shown in Table 6.6a. This indicates that only λ_j's are sensitive while others are remained unchanged. This can be explained by the same reason as discussed in section 6.1. It is, therefore, highly probable that this solution might result in a false optimum. Since the same difficulty as persisted in the previous section has been encountered, the same approach as we did in section 6.1 will be followed without repeating discussions. Using $\frac{1}{\lambda_j\text{'s}}$ as variables in place of λ_j's, the same set of starting points, $[\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \frac{1}{\lambda_3}, (t_c)_1, (t_c)_2, (t_c)_3, (t_p)_1, (t_p)_2, (t_p)_3, T_1, T_2, T_3]$ $= [5000., 5000., 5000., 2., 2., 2., 1., 1., 1., 100., 100., 100.]$, is tried. The solutions obtained and shown in Table 6.6b indicates much improvement. To test whether or not further improved solutions could be obtained, another set of starting points was tried. As we did for the first set of starting points, we tried both methods. The solutions for a set of starting values, $[\lambda_1, \lambda_2, \lambda_3, (t_c)_1, (t_c)_2, (t_c)_3, (t_p)_1, (t_p)_2, (t_p)_3,$

Table 6.6a. GRG solution for the first set of starting values (numerical example 2).

Iteration No.	scale parameter			mean corrective maintenance time			mean preventive maintenance time			scheduled preventive maintenance period			cost of design			cost of corrective maintenance			cost of corrective maintenance			total cost	system availability
	λ_1	λ_2	λ_3	$(t_c)_1$	$(t_c)_2$	$(t_c)_3$	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_3$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
starting point	.0002	.0002	.0002	2.	2.	2.	1.	1.	1.	100.	100.	100.	245.38	190.46	289.10	237.59	371.23	534.57	185.45	481.16	240.58	2775.51	.93367
1	.000215	.000252	.000192	2.	2.	2.	1.	1.	1.	100.	100.	100.	230.54	160.69	301.51	285.21	588.02	467.39	129.59	135.70	287.88	2611.38	.92437
2	.000217	.000257	.000191	2.	2.	2.	1.	1.	1.	100.	100.	100.	238.32	174.52	294.91	259.86	477.61	502.95	157.43	282.80	261 29	2663.74	.92934
3	.000217	.000257	.000191	2.	2.	2.	1.	1.	1.	100.	100.	100.	239.61	176.61	293.87	255.79	462.61	508.68	162.13	307.32	257.10	2663.71	.93001
4	.000217	.000257	.000191	2.	2.00003	2.	1.	1.	1.	100.	100.	100.	239.59	176.57	293.91	255.85	462.90	508 47	162.05	306.87	257.25	2663.58	.92999
5	.000217	.000257	.000191	2.	2.00004	2.	1.	1.	1.	100.	100.	100.	239.60	176.58	293.90	255.83	462.81	508.53	162.08	307.90	257.21	2663.56	.93000
6	.000217	.000257	.000191	2.	2.00004	2.	1.	1.	1.	100.	100.	100.	239.60	176.58	293.90	255.83	462.79	508 55	162.08	307.04	257.19	2663.56	.93000
7	.000217	.000257	.000161	2.	2.00004	2.	1.	1.	1.	100.	100.	100.	239.60	176.58	328.36	255.83	462.78	339.47	162.08	307.04	391.26	2659.60	.93402
8	.000217	.000257	.000159	2.	2.00004	2.	1.	1.	1.	100.	100.	100.	239.60	176.58	310.25	255.83	462.78	422.95	162.08	307.04	322.48	2659.59	.93206
9	.000229	.000284	.000158	2.	2.00004	2.	1.	1.	1.	100.	100.	100.	232.16	166.91	312.05	279.81	535.78	414.15	135.31	197.78	329.51	2626.81	.92749
10	.000229	.000284	.000158	2.	2.00006	2.	1.	1.	1.	100.	100.	100.	235.72	171.35	311.36	268.13	501.13	417.46	148.08	246.53	326.85	2626.71	.93002
11	.000229	.000284	.000158	2.	2.00007	2.	1.	1.	1.	100.	100.	100.	235.73	171.36	311.36	268.12	501.10	417.48	148.09	246.57	326.84	2626.68	.93002
12	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	235.73	171.36	311.36	268.12	501.10	417.48	148.09	246.57	326.84	2626.66	.93002
13	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	235.72	171.35	311.36	268.14	501.14	417.49	148.06	246.50	326,83	2626.63	.93001
14	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	235.72	171.36	311.36	268.13	501.12	417.49	148.07	246.53	326.83	2626.62	.93002
15	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	235.72	171.36	311.36	268.13	501.12	417.49	148.08	246.54	326.83	2626.62	.93002
16	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	235.72	171.36	311.36	268.13	501.12	417.49	148.08	246.54	326.83	2626.62	.93002
17	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	235.72	171.36	311.36	268.13	501.12	417.49	148.08	246 54	326.83	2626.62	.93002
Final	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	235.72	171.36	311.36	268.13	501.11	417.49	148.08	246.54	326.83	2626.62	.93002

Table 6.6b. GRG solution for the first of starting values (numerical example 2) ; using $\frac{1}{\lambda_j\text{'s}}$ as original problem variables

Iteration No.	scale parameter			mean corrective maintenance time			mean preventive maintenance time			scheduled preventive maintenance period		
	λ_1	λ_2	λ_3	$(t_c)_1$	$(t_c)_2$	$(t_c)_3$	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3
starting point	(5000.)* .0002	(5000.) .0002	(5000.) .0002	2.	2.	2.	1.	1.	1.	100.	100.	100.
1	(4999.99992) .00020	(4999.99972) .00020	(5000.00004) .00020	1.34713	.87725	.5	.41158	.1	.26447	100.02010	100.05866	100.02232
2	(4999.99989) .00020	(4999.99969) .00020	(4999.99998) .00020	1.27891	.92334	1.09035	.24294	.1	.32527	100.02289	100.06164	100.02678
3	(4999.99984) .00020	(4999.99962) .00020	(4999.99991) .00020	1.16180	.98034	1.07295	.1	.1	.1	100.02842	100.06913	100.03591
4	(4999.99983) .00020	(4999.99961) .00020	(4999.99990) .00020	1.16340	.97807	1.06097	.1	.1	.1	100.02863	100.06948	100.03620
5	(4999.99983) .00020	(4999.99961) .00020	(4999.99990) .00020	1.16387	.97741	1.05751	.1	.1	.1	100.02870	100.06959	100.03629
Final	(4999.99983) .00020	(4999.99961) .00020	(4999.99990) .00020	1.16387	.97741	1.05751	.1	.1	.1	100.02870	100.06959	100.03629

Iteration No.	cost of design			cost of corrective maintenance			cost of preventive maintenance			total cost	system availability
	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_3$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
starting point	245.38	190.46	289.10	237.59	371.23	534.57	185.45	481.16	240.58	2775.51	.93367
1	504.01	519.08	772.79	27.70	23.15	33.31	4.99	29.83	14.96	1733.41	.98259
2	358.77	310.07	284.59	77.29	95.45	687.22	4.99	29.89	101.47	1552.74	.95555
3	388.68	301.94	455.02	59.89	103.29	146.85	4.99	29.88	14.96	1493.38	.96894
4	354.79	318.09	451.90	80.15	88.54	149.99	4.99	29.89	14.96	1493.29	.96809
5	354.75	318.16	452.39	80.18	88.48	149.49	4.99	29.89	14.96	1493.29	.96811
Final	354.70	318.24	452.91	80.21	88.42	148.97	4.99	29.89	14.96	1493.29	.96812

* Figures in parentheses respectively represent $\frac{1}{\lambda_j\text{'s}}$ values

T_1, T_2, T_3] = [.00015, .00015, .00015, 2., 2., 2., 1.5, 1.5, 1.5, 110., 110., 110.], and a set of converted starting values, [$\frac{1}{\lambda_1}$, $\frac{1}{\lambda_2}$, $\frac{1}{\lambda_3}$, $(t_c)_1$, $(t_c)_2$, $(t_c)_3$, $(t_p)_1$, $(t_p)_2$, $(t_p)_3$, T_1, T_2, T_3] = [6666.667, 6666.667, 6666.667, 2., 2., 2., 1.5, 1.5, 1.5, 110., 110., 110.], are respectively shown in Table 6.7a and 6.7b.. The same stopping criterion given by equation (6.21) is applied to terminate the program. The solutions obtained in Table 6.6a, 6.7a, and 6.7b are inferior to that obtained in Table 6.6b. Hence, we conclude that the solution obtained in Table 6.6b is the global optimum.

SUMT Results

To compare the SUMT results with GRG results, the identical two sets of starting values as used for the GRG were tried. The SUMT results for the first set of starting values and for the second set of starting values are respectively shown in Table 6.8 and 6.9. Five (k=5) iterations for S-function minimization and 2252 calculations for the objective functional values, and k=4 iterations and 1729 objective functional value calculations are respectively required for the first and second set of starting values to reach the optimal solutions. In both cases, when the number of cut-down step-size operation is 4, the minimization of S-function at each k-iteration is terminated and the final stopping criterion used to terminate the program is $\epsilon=10^{-3}$. Since the optimum solution obtained in Table 6.9 is somewhat inferior to that in Table 6.8, we conclude that

Table 6.7a. GRG solution for the second set of starting values (numerical example 2)

Iteration No.	scale parameter			mean corrective maintenance time			mean preventive maintenance time						cost of design			cost of corrective maintenance			cost of preventive maintenance			total cost	system availability
	λ_1	λ_2	λ_3	$(t_c)_1$	$(t_c)_2$	$(t_c)_3$	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_2$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
starting point	.00015	.00015	.00015	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	257.94	198.71	302.17	195.51	305.49	439.91	297.32	761.54	380.77	3139.36	.93815
1	.000166	.000196	.000158	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	251.53	185.85	298.55	216.05	393.45	462.62	257.38	496.28	355.26	2916.95	.93415
2	.000179	.000227	.000164	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	238.16	167.92	290.43	260.57	529.06	514.68	179.41	196.57	299.54	2809.36	.92627
3	.000185	.000240	.000167	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	241.84	173.24	292.75	248.06	486.88	499.60	200.04	273.79	315.27	2772.92	.92875
4	.000185	.000282	.000129	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	244.61	168.98	312.69	238.79	520.55	375.58	216.00	210.92	456.69	2744.80	.93038
5	.000185	.000285	.000127	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	244.57	168.22	314.85	238.94	526.67	362.68	215.73	200.53	472.53	2744.66	.93038
6	.000186	.000285	.000126	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	244.31	168.54	314.31	239.79	524.06	365.86	214.24	204.93	468.60	2744.64	.93038
Final	.000186	.000285	.000126	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	244.18	168.57	314.44	240.23	523.86	365.12	213.48	205.25	469.51	2744.64	.93038

Table 6.7b. GRG solution for the second set of starting values (numerical example 2) : using $\frac{1}{\lambda_j\text{'s}}$ as original problem variables

Iteration No.	scale parameter			mean corrective maintenance time			mean preventive maintenance time			scheduled preventive maintenance period			cost of design			cost of corrective maintenance			cost of preventive maintenance			total cost	system availability
	λ_1	λ_2	λ_3	$(t_c)_1$	$(t_c)_2$	$(t_c)_3$	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	$(C_d)_1$	$(C_d)_2$	$(C_d)_3$	$(C_c)_1$	$(C_c)_2$	$(C_c)_3$	$(C_p)_1$	$(C_p)_2$	$(C_p)_3$	C_T	A_s
starting point	(6666.667)* .00015	(6666.667) .00015	(6666.667) .00015	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	257.94	198.71	302.17	195.51	305.49	439.91	297.31	761.54	303.77	3139.36	.93815
1	(6666.66690) .00015	(6666.66671) .00015	(6666.66695) .00015	1.57330	1.26411	.94284	1.01153	.20644	.88941	110.01975	110.05653	110.02097	364.79	553.36	759.84	61.64	19.11	27.51	86.57	31.17	51.60	1859.31	.98014
2	(6666.66685) .00015	(6666.66663) .00015	(6666.66690) .00015	1.38512	1.02166	.68208	.72709	.1	.58367	110.02922	110.07091	110.03138	362.51	414.70	761.48	66.94	42.38	27.51	68.14	31.20	50.58	1736.88	.97810
3	(6666.66678) .00015	(6666.66658) .00015	(6666.66681) .00015	1.24018	1.04397	1.11245	.29770	.1	.34691	110.03930	110.07706	110.04201	437.08	321.08	342.94	44.17	90.58	431.67	5.21	31.20	15.67	1590.61	.96693
4	(6666.66672) .00015	(6666.66653) .00015	(6666.66675) .00015	1.17902	1.05172	1.200031	.1	.1	.1	110.04677	110.08305	110.05090	386.07	327.95	449.30	66.36	84.23	163.94	5.21	31.21	15.62	1523.63	.97148
5	(6666.66670) .00015	(6666.66651) .00015	(6666.66673) .00015	1.21828	1.04674	1.11267	.1	.1	.1	110.04839	110.08534	110.05256	377.90	329.06	469.74	71.38	83.95	141.81	5.21	31.21	15.62	1525.62	.97186
6	(6666.66669) .00015	(6666.66650) .00015	(6666.66672) .00015	1.23000	1.04548	1.09493	.1	.1	.1	110.04907	110.08639	110.05349	374.89	329.46	478.35	73.37	83.63	133.78	5.21	31.21	15.62	1525.51	.97200
7	(6666.66669) .00015	(6666.66650) .00015	(6666.66672) .00015	1.23442	1.04491	1.08920	.1	.1	.1	110.04936	110.08685	110.05392	373.56	329.63	481.33	74.27	83.50	131.16	5.21	31.21	15.62	1525.49	.97204
8	(6666.66668) .00015	(6666.66650) .00015	(6666.66672) .00015	1.23725	1.04464	1.08576	.1	.1	.1	110.04956	110.08717	110.05422	372.95	329.71	482.55	74.69	83.43	130.11	5.21	31.21	15.62	1525.48	.97204
Final	(6666.66668) .00015	(6666.66650) .00015	(6666.66672) .00015	1.23725	1.04464	1.08576	.1	.1	.1	110.04956	110.08717	110.05422	372.44	329.77	483.52	75.05	83.39	129.28	5.21	31.20	15.62	1525.48	.97204

* Figures in parentheses respectively represent $\frac{1}{\lambda_j\text{'s}}$ values

Table 6.8. SUMT solution for the first set of starting values (numerical example 2)

Iteration k	cumulative No. of f-value calculations up to iteration k	value of r_k	scale parameter λ_1	λ_2	λ_3	mean corrective maintenance time $(t_c)_1$	$(t_c)_2$	$(t_c)_3$	mean preventive maintenance time $(t_p)_1$	$(t_p)_2$	$(t_p)_3$
estimated optimum value			.00015	.00015	.00015	1.5	1.5	1.5	1.	1.	1.
initial step-size			.000015	.000015	.000015	.15	.15	.15	.1	.1	.1
starting point		.01912	.0002	.0002	.0002	2.	2.	2.	1.	1.	1.
1	501	.01912	.000234	.000234	.000234	1.66452	1.66452	1.66452	.77635	.77635	.77635
2	813	.00239	.000249	.000234	.000234	1.66452	1.66452	1.66452	.77635	.77635	.77635
3	1314	.00060	.000268	.000253	.000215	1.27903	1.16468	1.23386	.39700	.39700	.39700
4	1749	.00007	.000305	.000253	.000215	1.27903	1.16468	1.23386	.39700	.39700	.39700
5 (Final)	2252	.00002	.000305	.000253	.000215	.67892	1.16468	.90380	.73807	.24353	.55048

Iteration k	scheduled preventive maintenance period T_1	T_2	T_3	cost of design $(C_d)_1$	$(C_d)_2$	$(C_d)_3$	cost of corrective maintenance $(C_c)_1$	$(C_c)_2$	$(C_c)_3$	cost of preventive maintenance $(C_p)_1$	$(C_p)_2$	$(C_p)_3$	total cost C_T	functional value	system availability A_s
estimated optimum value	110.	110.	110.												
initial step-size	11.	11.	11.												
starting point	100.	100.	100.	245.38	190.46	289.10	237.59	371.23	534.57	185.45	481.26	240.58	2775.51	3469.00	.93357
1	124.602	131.185	132.233	252.23	195.95	296.97	212.20	336.27	485.12	29.79	52.80	24.75	1886.07	2441.65	.93712
2	124.602	131.185	132.233	247.70	195.95	296.97	220.89	336.27	485.12	24.36	52.80	24.75	1834.81	1545.83	.93541
3	140.849	143.268	143.296	278.31	235.59	356.68	138.93	174.76	257.65	2.72	8.30	8.30	1461.29	1477.97	.95179
4	140.849	143.268	143.296	269.80	235.59	356.68	149.12	174.76	257.65	1.38	8.30	8.30	1461.57	1463.59	.95059
5 (Final)	140.849	135.946	143.296	407.40	236.57	429.94	42.02	173.66	138.24	2.83	7.93	11.86	1450.45	1450.95	.96311

Table 6.9. SUMT solution for the second set of starting values (numerical example 2)

Iteration k	cumulative No. of f-value calculations up to iteration k	value of r_k	scale parameter λ_1	λ_2	λ_3	mean corrective maintenance time $(t_c)_1$	$(t_c)_2$	$(t_c)_3$	mean preventive maintenance time $(t_p)_1$	$(t_p)_2$	$(t_p)_3$	scheduled preventive maintenance time T_1	T_2	T_3	cost of design $(C_d)_1$	$(C_d)_2$	$(C_d)_3$	cost of corrective maintenance $(C_c)_1$	$(C_c)_2$	$(C_c)_3$	cost of preventive maintenance $(C_p)_1$	$(C_p)_2$	$(C_p)_3$	total cost C_T	S functional value	system availability A_s
estimated optimum value			.00025	.00025	.00025	1.5	1.5	1.5	1.5	1.5	1.5	110.	110.	110.												
initial step-size			.000025	.000025	.000025	.15	.15	.15	.15	.15	.15	11.	11.	11.												
starting point		.01196	.00015	.00015	.00015	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	257.94	198.71	302.17	195.51	305.49	439.91	297.32	761.54	380.77	3139.36	3924.00	.93515
1	501	.01196	.000206	.000206	.000206	1.66452	1.66452	1.66452	1.16452	1.16452	1.16452	134.602	136.161	134.255	259.34	201.45	305.65	200.11	313.69	449.91	39.14	92.60	51.47	1913.34	2326.33	.94029
2	792	.00150	.000222	.000206	.000206	1.66452	1.66452	1.66452	1.16452	1.16452	1.16452	134.602	136.161	134.255	254.09	201.45	305.65	209.65	313.69	449.91	30.54	92.60	51.47	1909.03	1958.94	.93955
3	1294	.00037	.000316	.000262	.000237	1.25019	1.17294	1.16948	.59780	.59780	.59780	144.319	144.895	144.415	271.16	232.49	358.66	145.17	180.83	245.33	1.35	9.71	7.96	1455.64	1462.07	.95018
4 (Final)	1729	.00005	.000379	.000262	.000237	1.25019	1.17294	1.16948	.59780	.59780	.59780	144.319	144.895	144.415	260.95	232.49	358.66	159.26	180.83	245.33	.40	9.71	7.96	1455.56	1456.72	.94545

the solution obtained in Table 6.8 is the global optimum.

Comparison Between GRG and SUMT Results

Both GRG and SUMT final results for the first and second set of starting values are respectively summarized in Table 6.10a and 6.10b. There is approximately 2.9% difference between the global optimum values obtained by GRG and SUMT. This difference might have been caused by the difficulty discussed in section 6.1. Since other comparisons between the results can, similarly, be made as was done in section 6.1, these will not be repeated in this section.

Table 6.10a. Summary of GRG and SUMT final results for the first set of starting values (numerical example 2)

	scale parameter			mean corrective maintenance time			mean preventive maintenance time			scheduled preventive maintenance period			total cost	system availability	No. of iteration	execution time(min.)
	λ_1	λ_2	λ_3	$(t_c)_1$	$(t_c)_2$	$(t_c)_3$	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	C_T	A_s		
starting point	.0002	.0002	.0002	2.	2.	2.	1.	1.	1.	100.	100.	100.	2775.51	.93367		
GRG	.000229	.000284	.000158	2.	2.00008	2.	1.	1.	1.	100.	100.	100.	2626.62	.93002	17	1.061
GRG(using $\frac{1}{\lambda_j}$'s as variables)	.0002	.0002	.0002	1.16387	.97741	1.05751	.1	.1	.1	100.02870	100.06959	100.03629	1493.29*	.96812	5	.439
SUMT	.000305	.000253	.000215	.67892	1.16468	.90380	.73807	.24353	.55048	140.849	135.946	143.296	1450.45**	.96311	k=5 (2252)	1.681

* global optimum obtained by GRG

** global optimum obtained by SUMT

Table 6.10b. Summary of GRG and SUMT final results for the second set of starting values (numerical example 2)

	scale parameter			mean corrective maintenance time			mean preventive maintenance time			scheduled preventive maintenance period			total cost	system availability	No. of iteration	execution time(min.)
	λ_1	λ_2	λ_3	$(t_c)_1$	$(t_c)_2$	$(t_c)_3$	$(t_p)_1$	$(t_p)_2$	$(t_p)_3$	T_1	T_2	T_3	C_T	A_s		
starting point	.00015	.00015	.00015	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	3139.36	.93515		
GRG	.000186	.000285	.000126	2.	2.	2.	1.5	1.5	1.5	110.	110.	110.	2744.64	.93038	6	.413
GRG(using $\frac{1}{\lambda_j}$'s as variables)	.00015	.00015	.00015	1.23725	1.04464	1.08576	.1	.1	.1	110.04956	110.08717	110.05422	1525.48	.97204	8	.442
SUMT	.000379	.000262	.000237	1.25019	1.17294	1.16948	.59780	.59780	.59780	144.319	144.895	144.415	1455.56	.94845	k=4 (1729)	1.292

Chapter 7

DISCUSSION AND CONCLUSIONS

This study deals with the optimal availability allocation problem for maintained systems. The introduction of availability as a single measure of system effectiveness is of primary concern in this study. Since availability reflects both the reliability and maintainability of a system, it appears to be an appropriate measure from an engineering design and management viewpoint.

Availability models are developed for systems which contain subsystems in series where each subsystem has identical units in parallel. The definition of availability employed in this study assumes a steady state condition. The models developed herein enable us to assume various probability density functions for failure and repair times, whereas the normal Markovian approach uses only exponential failure distributions.

In developing the availability models, the corrective maintenance policy assumed is such that repair or replacement for the subsystem begins only when the subsystem fails due to the failure of all redundant units. This assumption requires the subsystem to be fully restored after the completion of corrective maintenance. This policy, however, is applicable to those subsystems where the subsystem's output is monitored. For those subsystems in which the status of individual units can be monitored, some variations of the corrective maintenance policy may also be considered. Under a policy such as to repair each individual unit as it fails, the cost associated with corrective maintenance is expected to increase due to the increased frequent maintenance. Hence, the latter policy might be preferrable only if the reduction in the costs associated with both design and preventive maintenance exceeds the increase in the cost of corrective maintenance. In this thesis, however, only the former policy has been considered because it seems to be preferred from an administrative point of view and seems to be the case most often encountered in practice.

The preventive maintenance policy assumed in this study is more realistic than strictly periodic maintenance policy in that preventive maintenance action for each subsystem need not necessarily be performed every T_j. Thus, the number of actual preventive maintenance actions under this policy is expected to be less than that under a strictly periodic maintenance policy. In this respect, the cost associated with preventive maintenance will be reduced with this policy

The proposed model is inadequate if a sequentially determined preventive maintenance policy is assumed. The development of model with a sequentially determined preventive maintenance policy seems to be much more complex and is not attempted in this study. However, if such a study is conducted at a later date a similar conceptual approach used in this study may be employed.

Under both corrective and preventive maintenance policies assumed in this thesis, each subsystem redundancy can be fully restored after the completion of either corrective or preventive maintenance. These assumptions enable us to develop availability models which reflect the effects of both corrective and preventive maintenance as proposed in this thesis. If subsystem redundancy cannot be fully restored either by corrective or preventive maintenance, the problem of developing availability models analytically is much more complex. The simulation approach, however, is expected to solve this type of problem and is suggested for further work.

The number of repairmen assigned to each subsystem is assumed to be either one or equal to that of redundant units. It is possible, however, to develop models under the assumption of various number of repairmen.

The availability equations contain a integral term. If exponential failure distribution is assumed, this can be evaluated analytically with the use of binomial theorem. However, when the failure time distribution is other than exponential, it is difficult, if not impossible, to evaluate it analytically. Therefore, numerical integration by the use of trapezoidal rule is employed to evaluate this integral term in numerical examples.

In numerical examples, the number of redundant units assumed for each subsystem is two, but different number of units for each subsystem can be assumed. Although this is treated as a given constant, future study on this subject will be able to treat it as a variable.

The cost function for the system consists of three cost components : the cost for design, the cost for corrective maintenance, and the cost for preventive maintenance. Each of the individual cost components are interrelated and are an approximation of real world situations. In numerical examples, a typical set of constants is assumed for the cost coefficients, however, they can be estimated if operational data is available for any particular system.

Both GRG and SUMT are employed to solve availability allocation problems. The results obtained by these two methods are compared. In GRG, the direction of movement is along the projected reduced gradient and the magnitude of movement for each variable is determined by the magnitudes of the partial derivatives of both the objective function and the constraints. Due to the great differences between the values of the partial derivatives, only some variables having large values of partial derivatives have significant movement to improve the value of the objective function while the others with small values of the partial derivatives remained unchanged. One possible alleviation from this difficulty is to employ the inverse of those variables having large values of the partial derivatives as variables in the problem. This, sometimes, enables us to lessen the difference between the values of partial derivatives. As shown in the numerical examples, this method has helped to obtain improved solutions, however, fundamental alleviation from this type of difficulty still remains unsolved. In Lai's modified version of SUMT which incorporates the Hooke and Jeeves pattern search, the direction of search is determined by a direct comparison of two values of the object function at two points. This requires a large number of evaluations of the functional values, thus increasing the computing time.

The availability models developed in this thesis are more general and extensive than any others developed in the past in that they reflect the effects of both corrective and preventive maintenance. This study provides the basis for a procedure to allocate the availability parameters to the individual units of the subsystem. The availability allocation is treated as a cost minimization problem, subject to the constraint of satisfying the system availability requirement. This allocation technique is valuable in the early stages of maintained system design. This technique is also useful in the latter stages of system design when modifications and improvements for the initial specifications are required.

REFERENCES

1. Abadie,J.(ed.), Integer and Nonlinear Programming, North Holland Publishing Co., Amsterdam, 1970.

2. Abadie,J.(ed.), Nonlinear Programming, North Holland publishing Co., Amsterdam, 1967.

3. Abadie,J., "Solution des questions de dégénérescence dans la methode GRG", Électricité de France note HI 143/00 du 25 Septembre 1969.

4 Abadie,J. and Carpendier,J., Généralisation de la méthode du gradient réduit de Wolfe au cas de constrintes non-lineaires, Électricité de France note HR 6678, du 27 Octobre 1965.

5. Ankenbrandt,F.L.(ed.), Maintainability Design, Engineering Publishers, Elizabeth, N.J., 1963.

6. Arms,R.L. and Goodfriend,R.D., "Some Useful Reliability Graphs for Units and Simple Repairable Systems", 11th Nat. Sym. on Rel. and Q.C., pp.408-418, Miami beach, Florida, Jan. 1965.

7. Barlow,R.E. and Hunter,L.C., "System Efficiency and Reliability", Technometrics, Vol. 2, No.1, pp. 43-53,1960.

8. Barlow,R.E. and Hunter,L.C., "Mathematical Models for Systems Reliability", The Sylvania Technologist, Vol. XIII, Nos.1 and 2, Jan. and Apr. 1960.

9. Barlow,R.E. and Hunter,L.C., "Reliability Analysis of One Unit System", J. of the ORSA,Vol. 9 pp. 200-208, Mar.-Apr. 1961.

10. Barlow,R.E., Hunter,L.C., and Proschan,F., "Optimum Checking Procedures", 7th Nat. Sym. on Rel. and Q.C., pp. 485-495, Philadelphia, Pa., Jan. 1961.

11. Barlow,R.E. and Proschan,F., Mathematical Theory of Reliability, John Wiley, N.Y., 1965.

12. Barlow,R.E. and Proschan,F., "Planned Replacement", EDL-M 296, Electronic Defense Laboratories, Mountain View, Cal., 1960.

13. Barlow,R.E. and Proschan,F., "Planned Replacement", Studies in Applied Probability and Management Science (edited by Arrow, Karlin, and Scarf), Stanford University Press, Stanford, Calif., 1962.

14. Bazovsky, I., Reliability Theory and Practice, Prentice-Hall, Englewood Cliffs, N.J., 1961.

15. Bell, C.F., Kamins, M., and McCall, J.J., "Some Elements of planned Replacement Theory", 1966 Annual Sym. on Rel., pp. 98-117, San Francisco, Calif., Jan. 1966.

16. Bellman, R.E. and Dreyfus, S.E., "Dynamic Programming and the Reliability of Multicomponent Devices", Operations Research, Vol.6, No.2, pp. 200-206, Mar.-Apr. 1958.

17. Blanchard, B.S., Jr. and Lowery, E.E., Maintainability, McGraw-Hill, 1969.

18. Boodman, D.M., "The Reliability of Airborne Radar Equipment", J. of ORSA, Vol. 1, No. 2, pp. 39-45, Feb. 1953.

19. Branson, M.H. and Shah, B., "Reliability Analysis of Systems Comprised of Units with Arbitrary Repair-Time Distribution", IEEE Trans. on Rel., Vol. R-20, No.4, pp. 217-223, Nov. 1971.

20. Campbell, N.R., "The Replacement of Perishable Members of a Continually Operating System", J. Roy. Stat. Soc., Vol.7, pp. 110-130, 1941.

21. Carhart, R.R., "A Survey of the Current Status of the Reliability Problem", Rand Corp., Research Memo. RM-1131, Aug. 1953.

22. Carroll, C.W., "An Operations Research Approach to the Economic Optimization of a Kraft Pulping Process", Ph.D. Dissertation, Institute of Paper Chemistry, Applitown, Wisc., 1959.

23. Carroll, C.W., "The Created Response Surface Technique for Optimizing Nonlinear Restrained Systems", Operations Research, Vol.9, pp. 169-184, 1961.

24. Chatterjee, S., "Availability Models of Maintained Systems", A Master's Thesis, Dep't of I.E., Kansas State University, 1971.

25. Cho, H.H., "On Proper Preventive Maintenance", 9th Nat. Sym. on Rel. and Q.C., pp. 431-438, San Francisco, Calif., Jan. 1963.

26. Cunningham, C.E. and Cox, W., Applied Maintainability Engineering, John Wiley & Sons, 1972.

27. Davis, D.J., "An Analysis of Some Failure Data", J. of the American Statistical Assoc., Vol. 47, No. 258, pp. 113-150, June 1952.

28. deMercado,J.B., "Reliability Prediction Studies of Complex Systems having Many Failed States', IEEE Trans. on Rel., Vol. R-20, No. 4, pp. 223-230, Nov. 1971.

29. Dep't of the Army Pamphlet 705-1, Maintainability Engineering, Headquarters, Dep't of the Army, June 1966.

30. Derman,C. and Sacks,J., "Replacement of Periodically Inspected Equipment", Naval Research Logistics Quarterly, Vol. 7, No. 4, pp. 597-607, 1960.

31. Epstein,B. and Hosford,J. "Reliability of Some Two Unit Redundant Systems", 6th Nat. Sym. on Rel. and Q.C., pp. 469-476, Washington, D.C., Jan. 1960.

32. Fan,L.T., Wang,C.S., Tillman,F.A., and Hwang,C.L., "Optimization of Systems Reliability", IEEE Trans. on Rel., Vol. R-16, No. 2, pp. 81-86, Sept. 1967.

33. Faragher,W.E. and Watson,H.S., "Availability Analyses-A Realistic Methodology", 10th Nat. Sym. on Rel. and Q.C., pp. 365-378, Washington,D.C., 1964.

34. Fiacco,A.V., and McCormick,G.P., Nonlinear Programming: Sequential Unconstrained Minimization Techniques, John Wiley & Sons, N.Y., 1968.

35. Fiacco,A.V. and McCormick,G.P., "The Sequential Unconstrained Minimization Technique for Nonlinear Programming: A Primal-Dual Method", Management Sci., Vol. 10, pp. 360-366, 1964.

36. Finkelstein,J. and Schafer,R., "Dependability Models for a System of N Parallel Elements", 8th Nat. Sym. on Rel. and Q.C., pp. 434-441, Washington, D.C., 1962.

37. Flehinger,B.J., "A General Model for the Reliability Analysis of Systems Under Various Preventive Maintenance Policies", Annals of Math. Stat., Vol. 33, No. 1, pp. 137-156, 1962.

38. Fletcher,R.(ed.), Optimization, Academic Press, N.Y.,1969.

39. Gaver,D.P., Jr., "Time to Failure and Availability of Paralleled Systems with Repair", IEEE Trans. on Rel., Vol. R-12, No. 2, pp. 30-38, June 1963.

40. Goldman,A.S. and Slattery,T.B., Maintainability, John Wiley & Sons, N.Y., 1964.

41. Goldman,A.S. and Whitin,T.M., "Optimizing the Trade-off Between Reliability and Maintainability Design", 10th Nat. Sym. on Rel. and Q.C., pp, 19-32, Washington, D.C.,

1964.

42. Graves,R. and Wolfe,P.(eds.), Recent Advances in Mathematical Programming, McGraw-Hill, N.Y., 1963.

43. Guigou,J., Presentation et utilisation du code GRG, Électricité de France note HI 102/02 du 9 Juin 1969.

44. Guigou,J., Presentation et utilisation du code GREG, Électricité de France note HI 582/2 du Mai 1971.

45. Hall,R.A., Dubner,H., and Adler,L.B., "Reliability of Nonexponential Redundant Systems", 1966 Annual Symposium on Rel., pp. 594-608, San Francisco, Calif., Jan. 1966.

46. Hall,K.M. and McDonald,R.H., "Improving System Reliability", 7th Nat. Sym. on Rel. and Q.C., pp. 214-228, Philadelphia, Pa., Jan. 1961.

47. Hooke,R. and Jeeves,T.A., "Direct Search Solution of Numerical and Statistical Problems", J. Assoc. Compt. Mach., Vol. 8, 1961.

48. Howard,R.R., Howard,W.J., and Hadden,F.A., "Study of Down Time in Military Equipment", 5th Nat. Sym. on Rel. and Q.C., pp. 402-408, Philadelphia, Pa, 1959.

49. Hsu,F.T., Fan,L.T., and Hwang,C.L., "Sequential Unconstrained Minimization Technique (SUMT) for Optimal Production Planning", Institute for Systems Design and Optimization, Kansas State University,

50. Htun,L.T., "Reliability Prediction Techiques for Complex Systems", IEEE Trans. on Rel., Vol. R-15, pp. 58-69, Aug. 1966.

51. Hwang,C.L., Fan,L.T., and Kumar,S., "Hooke and Jeeves Pattern Search Solution to Optimal Production Planning Problems", Report No. 18, Institute of Systems Design and Optimization, Kansas State University, 1969.

52. Hwang,C.L., Williams,J.L., and Fan,L.T., "Introduction to the Generalized Reduced Gradient Method", Report No. 39, Institute for Systems Design and Optimization, Kansas State University, July 1972.

53. Johnson,P.A., "A Proposed Methodelogy for Designing Real-Time InformationSystems with Availability Constraints", IEEE Trans. on Rel., Vol. R-21, No. 4, pp. 220-223, Nov. 1972.

54. Kabak,I.W., "System Availability and Some Design Implications", Operations Research, Vol. 17, pp. 827-837, 1969.

55. Kettelle,J.D., Jr., "Least-Cost Allocation of Reliability Investment", Operations Research, Vol. 10, No. 2, pp. 249-265, 1962.

56. Kneale,S.G., "Reliability of Parallel Systems with Repair and Switching", 7th Nat. Sym. on Rel. and Q.C., pp. 129-133, Philadelphia, Pa., Jan. 1961.

57. Lai,K.C., "Optimization of Industrial Management Systems by the Sequential Unconstrained Minimization Technique", A Master's Report, Dep't of I.E., Kansas State Univ., 1970.

58. Lambert,B.K., Walvekar,A.G., and Hirmas,J.P., "Optimal Redundancy and Availability Allocation in Multistage Systems", IEEE Trans. on Rel., Vol. R-20, No. 3, pp. 182-185, Aug. 1971.

59. Martz,H.F., Jr., "On Single-Cycle Availability", IEEE Trans. on Rel., Vol. R-20, No. 1, pp. 21-23, Feb. 1971.

60. McGregor,M.A., "Approximation Formulas for Reliability with Repair", IEEE Trans. on Rel., Vol. R-12, No. 4, pp. 64-91, Dec. 1963.

61. McNichols,R.J. and Messer,G.H., Jr., "A Cost-Based Availability Allocation Algorithm", IEEE Trans. on Rel., Vol. R-20, No. 3, pp. 178-182, Aug. 1971.

62. Meyers,R. and Dick,R.S., "Some Considerations of Scheduled Maintenance", 8th Nat. Sym. on Rel. and Q.C., pp. 343-356, Washington, D.C., Jan. 1962.

63. Meykar,O.A., "Maintainability Terminology Supports the Effectiveness Concepts", IEEE Trans. on Rel., Vol. R-16, No. 1, pp. 10-15, May 1967.

64. Mizukami,K., "Optimum Redundancy for Maximum System Reliability by the Method of Convex and Integer Programming", ORSA, Vol. 16, pp. 392-406, Mar.-Apr. 1968.

65. Morse,P.M., Queues, Inventories, and Maintenance, John Wiley & Sons, N.Y., 1958.

66. Myers,P.J., "Monte Carlo : Reliability Tool for Design Engineers", 9th Nat. Sym. on Rel. and Q.C., pp. 487-492, San Francisco, Calif., Jan. 1963.

67. Nagy,G., "The Reliability of Repairable Systems", 9th Nat. Sym. on Rel. and Q.C., pp. 93-108, San Francisco, Calif., Jan. 1963.

68. Paviani,D.A. and Himmelblau,D.M., "Constrained Nonlinear

Optimization by Heuristic Programming", AICHE meeting in New Orleans, March 1969.

69. Pennington, R.H., Introductory Computer Methods and Numerical Analysis, The Macmillan Co., N.Y., 1967.

70. Peterson, E.L., "Maintainability Application to System Effectiveness Quantification", IEEE Trans. on Rel., Vol. R-20, No. 1, pp. 3-7, Feb. 1971.

71. Proschan, F. and Bray, T., "Optimum Redundance under Multiple Constraints", ORSA, Vol. 13, pp. 800-814, Sept.-Oct. 1965.

72. Rohn, W.B., "Reliability Prediction for Complex Systems", 5th Nat. Sym. on Rel. and Q.C., pp. 381-388, Philadelphia, Pa., Jan. 1959.

73. Rosenheim, D.E., "Analysis of Reliability Improvement through Redundancy", Prodeedings of the New York Conference on Reliability Theory, June 1958.

74. Rudd, D.F., "Reliability Theory in Chemical Systems Design", I & EC Fundamentals, Vol. 1, No. 2, pp. 138-143, May 1962.

75. Ryerson, C.M., "Definitions Panel", 5th Nat Sym. on Rel. and Q.C., pp. 161-178, Philadelphia, Pa., Jan. 1959.

76. Sandler, G.H., System Reliability Engineering, Prentice-Hall, Englewood Cliffs, N.J., 1963.

77. Savage, I.R., "Cycling", Naval Research Logistics Quarterly, Vol. 3, No. 3, pp. 163-175, 1956.

78. Shafii, Y., "A Study of the Generalized Reduced Gradient Method", A Master's Report, Dep't of I.E., Kansas State University, 1973.

79. Shershin, A.C., "Mathematical Optimization Techniques for the Simultaneous Apportionments of Reliability and Maintainability", Operations Research, Vol. 18, No. 1, pp. 95-106, 1970.

80. Shooman, M.L., Probabilistic Reliability - An Engineering Approach, McGraw-Hill, 1968.

81. Tillman, F.A., "Optimization of Constrained Reliability Problems with Several Modes of Failures by Integer Programming", IEEE Trans. on Rel., Vol. R-18, pp. 47-53, May 1969.

82. Tillman, F.A., Hwang, C.L., Fan, L.T., and Lai, K.C.,

"Optimal Reliability of a Complex System", IEEE Trans. on Rel., Vol. R-19, No.3, pp. 95-100, Aug. 1970.

83. Tillman,F.A. and Littschwager,J., "Integer Programming Formulation of Constrained Reliability Problems", Management Science, Vol. 13, pp. 887-899, July 1967.

84. Von Alven,W.H. (ed.), Reliability Engineering, ARINC Research Corp., Prentice-Hall, Englewood Cliffs, N.J., 1964.

85. Weiss,G.H., "On the Theory of Replacement of Machinery with a Random Filure Time", Naval Research Logistics Quarterly, Vol. 3, No. 4, pp. 279-294, 1956.

86. Weiss,G.H., "On Some Economic Factors Influencing a Reliability Program", NAVORD Report 4256, U.S. Naval Ordance Lab., White Oak, Md., 1956.

87. Weiss,G.H., "The Reliability of a Redundant System which Operaters Repetitively", NAVORD Report 4348, U.S. Naval Ordance Lab., White Oak, Md., 1956.

88. Weissbaum,W.E., "Probability - Theoretic Solution of Some Maintenance Problems", Proc. Fouth Signal Maintenance Symposium, 1960.

89. Welker,E.L., "Relationship Between Equipment Reliability, Preventive Maintenance Policy, and Operating Costs", ARINC Research Corp., Publication No. 101-9-135, Feb. 1959.

90. Welker,E.L. and Horne,R.C., "Concepts Associated with System Effectiveness", ARINC Monograph No. 9, ARINC Research Corp., July, 1960.

91. Westland,R.A., Hanifan,D.T., and Sacks,J., "A Reliability Maintainability Trade-Off Procedure", 10th Nat. Sym. on Rel. and Q.C., pp. 600-611, Washington, D.C., Jan. 1964.

92. Wilkinson,R.E. and Walvekar,A.G., "Optimal Availability Allocation in a Multicomponent System", AIIE Trans., Vol. 2, No. 3, pp. 270-272, Sept. 1970.

93. Williams,J.L., "Optimization of Industrial Systems with the Separable Programming and the Generalized Reduced Gradient Methods", A Master's Thesis, Dep't of I.E., Kansas State University, 1972.

94. Wohl,J.G., "System Operational Readiness and Equipment Dependability", IEEE Trans. on Rel., Vol. R-15, No. 1, pp. 1-6, May 1966.

95. Zelen, M. (ed.), Statistical Theory of Reliability, Publication No. 9 of the Mathematics Research Center, U.S. Army, The University of Wisconsin Press, Madison, Wisc., 1964.

APPENDIX 1

A1.1 GLOSSARY OF TERMS IN RELIABILITY AND MAINTAINABILITY

This glossary is intended to clarify those technical terms and definitions used throughout this thesis and other related literatures on reliability and maintainability. These terms are defined in [17, 63, 75, 84].

<u>Active Repair Time</u>

That portion of down time during which one or more repairmen are working on the system to effect a repair. This time includes preparation time, fault-location time, fault-correction time, and final check-out time for the system.

<u>Chance Failure</u>

A chance failure is a failure which occurs at random within the operational time of a system after all efforts have been made to eliminate design defects and unsound units, and before wearout becomes predominant.

<u>Dependability</u>

According to Peterson [70], dependability accounts for reliability, maintainability, and alternate operational modes. The mathematical definition of dependability can be written as

$$D = R + M_o(1 - R) \tag{A1.1}$$

where D is the dependability which is the probability that a system's mission will be successfully completed within the mission time t_1, provided that a down time per failure not exceeding a given time t_2 will not adversely affect the overall mission. R is the reliability which is the probability that a system will operate without failure for the mission time t_1. M_o is the operational maintainability of the system - the probability that when a failure occurs it will be repaired in a time not exceeding the allowable downtime t_2.

Down Time

The total time during which the system is not in acceptable operating condition. This can be subdivided into active repair time, logistics or supply time, and wait or administrative time.

Failure

The inability of a system to perform within previously specified limits.

Failure Rate

The failure rate or hazard rate r(t) associated with the random variable T is defined as

$$r(t) = \frac{f(t)}{R(t)} \tag{A1.2}$$

where f(t) is the pdf of T and R(t) is the reliability function. To interpret r(t), consider the conditional probability, i.e., the probability that the system will fail during the next δt time units, given that the system is functioning properly at time t. Applying the definition of conditional probability, we may write this as

$$P(t \leq T \leq t + \delta t \mid T > t) = \frac{P(t < T \leq t + \delta t)}{P(T > t)}$$

$$= \frac{\int_t^{t+\delta t} f(x)dx}{P(T > t)} = \frac{\delta t f(\epsilon)}{R(t)} \qquad \text{(A1.3)}$$

where $t \leq \epsilon \leq t + \delta t$.

For small δt and supposing that f is continuous at 0^+, the last expression in equation (A1.3) is approximately equal to $\delta t r(t)$. Thus, $\delta t r(t)$ represents the approximate probability of failure occurring between time t and $t + \delta t$. Note that the pdf of T, f, uniquely determines the failure rate r(t), or conversely, r(t) uniquely determines the pdf f by the following equation :

$$f(t) = r(t)e^{-\int_0^t r(s)ds} \qquad \text{(A1.4)}$$

Logistics or Supply Time

That portion of down time during which maintenance is delayed solely because a required item is not immediately available.

Mean Time Between Failures (MTBF)

The total measured operating time of a population of equipments divided by the total number of failures within the population during the measured period of time. Alternatively, mean time between failures of a repairable equipment is defined as the ratio of the total operating time to the total number of failures. The measured operating time of the equipments of the population which did not fail must be included. This measurement is normally made during that period of time between the early life and wearout failures. In the case of exponentially distribution time between failures this ratio is the reciprocal of failure rate.

Meam Time to Failure (MTTF)

The measured operating time of a single piece of equipment divided by the total number of failures of the equipment during the measured period of time. This measurement is normally made during that period of time between the early life and wearout failures.

Mean Time to First Failure (MTTFF)

The average time to first failure of several equipments. It is used to determine the apparent approach of the equipment life characteristic to its random failure rate and is accomplished during the manufacturing phase of a program.

Mission Time

The period of time in which a device must perform specified mission task in a specified environment.

Operating Time

The time during which the system is operating in a manner acceptable to the operator. This includes the time when the operator may be somewhat dissatisfied with the manner of operation, but is not sufficiently dissatisfied to shut the system down and request repair action.

Operational Readiness

The probability that a product will perform satisfactorily at any point in calendar time.

Probability of Survival

The probability of a given system of performing its intended function for the given Use Cycle.

Redundancy

The existence of more than one means for accomplishing a given task, where all means must fail before there is an over-all failure to the system. Parallel redundancy applies to systems where both means are working at the same time to accomplish the task, and either of the systems is capable of handling the job itself in case of failure of the other system. Standby redundancy applies to a system where there is an alternate means of accomplishing the task that is switched in by a malfunction sensing device when the primary system fails.

Repair Time

The time measured from the beginning of correction of a malfunction to the completion of such correction. It is assumed that the cause of malfunction is known. Repair time is distinguished from repair effort which is measured in man-hours.

System Effectiveness

A measure of the degree to which a system can be expected to achieve a set of specific mission requirements, and which may be expressed as a function of availability, dependability, and capability.

Uptime

That elements of active time during which a system is either alert, reacting, or performing a mission.

Uptime Ratio

The quotient of uptime, divided by uptime plus downtime.

Wait or Administrative time

That portion of down time not included in active repair time and logistics or supply time. This includes both necessary administrative actions and unnecessarily wasted time.

Wearout

The process of attrition which results in an increase of the failure rate with increasing age.

Wearout Failures

Those failures which occur as a result of deterioration processes or mechanical wear, and whose probability of occurrence normally increases with time.

A1.2 MARKOV PROCESSES

When a sequence of experiments or trials constitutes a Markov process, it is assumed that the outcome on any trial depends on the outcome of the directly preceding trial. Hence a conditional probability associated with every pair of outcomes is required to be introduced. Space and time concepts are also needed to be introduced. For example, we may define the states of a machine as operating or failed, and consider how transitions are made back and forth from each of the possible states. It is possible to consider processes discrete in both space and time, processes discrete in space and continuous in time, and processes continuous in both space and time. Most reliability and availability problems are of processes discrete in space and continuous in time. The important feature of a Markov process is that the future states of the process depend only on its immediate past history, therefore we say that there is a lack of memory. If the conditional transition probability is constant, a process is called statioary. If the conditional probabilities vary with time, a process is called non-stationary or non-Markovian. To apply Markov processes in the formulation of reliability and availability models, exponential distribution is assumed for failure times. This assumption enables us to have a constant failure rate, thus a lack of memory property of a Markov process is satisfied.

To illustrate the use of Markovian approach, the reliability function for a two-unit redundant system given by equation (4.33) is obtained below by applying Markov process [31]. Under the same assumptions assumed in section 4.3, the possible states of the system are defined as

state 0 : both units operating

state 1 : one unit failed and is not repaired, the other operating

state 2 : both units failed.

The Markov graph for this system is shown in Figure A1.1.
The transition matrix in this case is

$$\begin{array}{cc} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} \\ \text{state} \end{array} \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 1-2\lambda dt & 2\lambda dt & 0 \\ 0 & 1-\lambda dt & \lambda dt \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad (A1.5)$$

To develop the system of differential equations we must first enumerate the probabilities of being in each state at time t+dt.
These are :

$$\begin{aligned} P_0\,(t+dt) &= P_0\,(t)(1-2\lambda dt) \\ P_1\,(t+dt) &= P_0\,(t)(2\lambda dt) + P_1\,(t)(1-\lambda dt) \\ P_2\,(t+dt) &= P_1\,(t)(\lambda dt) + P_2(t) \end{aligned} \qquad (A1.6)$$

where $P_i(t)$ represents the probability of being in i^{th} state at time t. From equation (A1.6), we obtain

$$P_0'(t) = -2\lambda P_0(t)$$

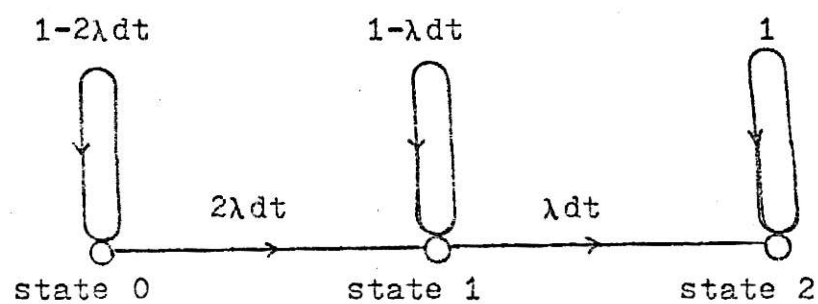

Figure A1.1. Markov graph for a two-unit redundant system.

$$P_1'(t) = 2\lambda P_0(t) - \lambda P_1(t) \qquad \text{(A1.7)}$$

$$P_2'(t) = \lambda P_1(t)$$

where $P_i'(t)$ denotes the first derivative.

If the system is in state 0 at time 0, the initial conditions become

$$P_0(0) = 1, \quad P_1(0) = 0, \quad P_2(0) = 0 \qquad \text{(A1.8)}$$

Taking Laplace transforms of equation (A1.7) we have

$$\begin{aligned} (s + 2\lambda)q_0(s) &= 1 \\ -2\lambda q_0(s) + (s + \lambda)q_1(s) &= 0 \qquad \text{(A1.9)} \\ -\lambda q_1(s) + sq_2(s) &= 0 \end{aligned}$$

Solving equation (A1.9) for $q_2(s)$ we obtain

$$q_2(s) = \frac{2\lambda^2}{s(s + \lambda)(s + 2\lambda)} \qquad \text{(A1.10)}$$

By partial fraction expansion

$$q_2(s) = \frac{1}{s} - \frac{2}{s+\lambda} + \frac{1}{s+2\lambda} \tag{A1.11}$$

Taking inverse transforms of $q_2(s)$ gives

$$P_2(t) = 1 - 2e^{-\lambda t} + e^{-2\lambda t} \tag{A1.12}$$

Therefore, the reliability of the system at time t is

$$\begin{aligned} R(t) &= 1 - P_2(t) \\ &= 2e^{-\lambda t} - e^{-2\lambda t} \end{aligned} \tag{A1.13}$$

A1.3 THE TRAPEZOIDAL RULE

Let $y = f(x)$ be a function defined between $x = a$ and $x = b$. Now divide the interval $a \leq x \leq b$ into n subintervals by the points $a < x_1 < x_2 < \cdots < x_{i-1} < x_i < \cdots < x_{n-1} < b$ and set

$$\delta x_i = x_i - x_{i-1} \tag{A1.14}$$

If we consider the following sum

$$\sum_{i=1}^{n} f(\gamma_i)\delta x_i \tag{A1.15}$$

where γ_i be any point between x_{i-1} and x_i, then as the number of intervals n approaches infinity in such a manner that all the lengths of the intervals δx_i approach zero, the quantity given by equation(A1.15) approaches a limit. This limit is called the definite integral of f(x) from a to b and is denoted by

$$\int_a^b f(x)dx \qquad \text{(A1.16)}$$

Equation (A1.16) can be considered to be the area lying between the curve f(x) and the x axis, and between the lines x=a and x=b. If the function f(x) is sufficiently simple that its antiderivative F(x), whose derivative F'(x) is equal to f(x), can be determind analytically, then equation(A1.16) can be evaluated by using the following equation :

$$\int_a^b f(x)dx = F(b) - F(a) \qquad \text{(A1.17)}$$

However, if it is difficult or impossible to find the F(x) analytically, as is often the case, it is necessary to employ the trapezoidal rule or some other numerical method of approximation to evaluate equation (A1.16). Such methods are quite natural and useful when digital computers are available [69].

The numerical integration by the use of trapezoidal rule can be made by dividing the interval a to b into n equal parts of length $\delta x = (b-a)/n$, erecting an ordinate line to the curve at each of the points of division, and connecting the end points of these ordinate lines to form trapezoids, as in Figure A1.2. The areas of n trapezoids, A_1, A_2 $\cdots$, A_n, are

$$A_1 = \tfrac{1}{2}[f(a) + f(x_1)]\delta x$$

$$A_2 = \tfrac{1}{2}[f(x_1) + f(x_2)]\delta x \tag{A1.18}$$

$$\vdots$$

$$A_n = \tfrac{1}{2}[f(x_{n-1}) + f(b)]\delta x$$

The sum of the areas of n trapezoids, A, is

$$A = A_1 + A_2 + \cdots + A_n$$

$$= \delta x[f(a)/2 + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + f(b)/2] \tag{A1.19}$$

This can be seen to approximate the area under the curve, in other words, this approximates the definite integral of f(x) between a and b. Therefore

$$\int_a^b f(x)dx \approx A \tag{A1.20}$$

The approximation can be made as close as desired by taking a sufficient number of intervals. The FORTRAN subroutine

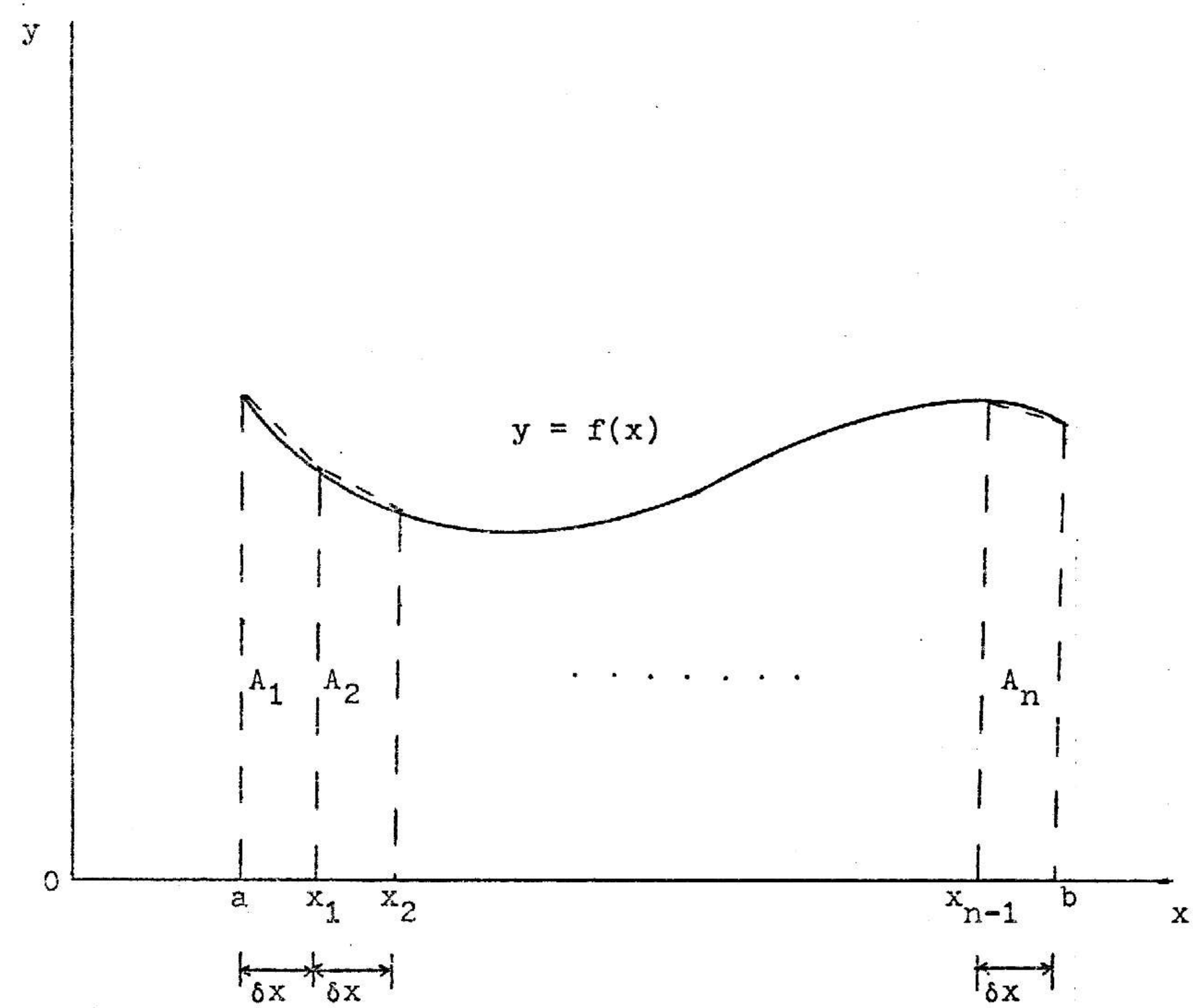

Figure A1.2. A function

INTEG which employes the trapezoidal rule to evaluate definite integral terms in both equations (4.47) and (4.118) is listed in Appendix 2.

APPENCIX 2

COMPUTER PROGRAM LISTINGS

A2.1 GRG : USER SUPPLIED SUBROUTINES FOR EXAMPLE 1

These subroutines use λ_j's, μ_j's, $(t_p)_j$'s, and T_j's as original problem variables. To use $\frac{1}{\lambda_j\text{'s}}$ and $\frac{1}{\mu_j\text{'s}}$ as variables, only a few modifications within these listed subroutines are required.

ILLEGIBLE DOCUMENT

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

THIS IS THE BEST COPY AVAILABLE

P-474

```
      SUBROUTINE PHIX
      DIMENSION CC(6,3)
      DIMENSION  B(100)
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
     1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFA,DTP,DELTA
     1X,DI,D,EPSIL,EPSIL0,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSILU,EPSIL6,EPSIL9,EPS,EPSI1,FI1,FI0,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHIC,PHI1,PHI2,PSIT,TOP,PSI4,PENAL,PS1,PS2,PS3,
     4RAPG,ROB,RO,RB,RA,SCAL,TB,TD,TC,TREN,TRE,TR,TR1,TETA,TETAT,TOTG,TQ
     5TG1,TETA1,TET2,TET3,TET4,TET5,TET,TETA2,TO,TO1,TO0,TET1,T1
      DOUBLE PRECISION  TRB,T2,T22,TEX,TE,U,VC,VO,V2,VI,V3,V,VCI,VCL,XI,
     1XNV,XN0,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIE1,XIE2
     2,XSE1,XSE2,XTR1,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
     3,Z,ZI,X,XC,XS,X1,X2,XR,B
      DOUBLE PRECISION DIT,DISOLS
      DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVD,DESIGN,CORRCT,PREVNT,DES,COR,PRE,
     1SUP,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,RTM3,RTM4,VAVO,AU,UC1,UC2,DEXP
      COMMON/LIE1/CC
      COMMON    B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
      COMMON  FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSIL0,EPSIL2,YSORT
      COMMON       NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NTO,NIN1,NIN2,NIN3,NI
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,II,IK,IK1,IS,IS1,IT,IBP,ICDB,JCDB,KC
     2DB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON       KFONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,K
     1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JKO,LC
      IF(IT) 10,11,11
   10 CC(1,1)=.60
      CC(1,2)=.50
      CC(1,3)=.80
      CC(2,1)=400.
      CC(2,2)=500.
      CC(2,3)=600.
      CC(3,1)=5.
      CC(3,2)=5.
      CC(3,3)=5.
      CC(4,1)=1.8
      CC(4,2)=2.0
      CC(4,3)=1.7
      CC(5,1)=20.
      CC(5,2)=15.
      CC(5,3)=50.
      CC(6,1)=3.
      CC(6,2)=4.
      CC(6,3)=2.
   11 DESIGN=0.
      CORRCT=0.
      PREVNT=0.
      DO 100 J=1,3
      JC=J+3
      JP=JC+3
      JQ=JP+3
      UNREL=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL=1.-UNREL
      CALL INTEG(XC,J,JQ,RMTBM)
      UMTBM=RMTBM/UNREL
      SMTBM=RMTBM/REL
```

```
      CTM=1./XC(JC)
      PTM=XC(JP)
      RTM=CTM*UNREL+PTM*REL
      WRITE(6,130)RMTBM,UMTBM,SMTBM,RTM
  130 FORMAT(/' ',4D15.6)
      DES=CC(1,J)*RMTBM+CC(2,J)/RTM-CC(3,J)
      COR=(1500./UMTBM)*(CC(4,J)*CTM)**2
      PRE=(1500./SMTBM)*(CC(5,J)*PTM-CC(6,J))
      SUB=DES+COR+PRE
      WRITE(6,140)DES,COR,PRE,SUB
  140 FORMAT(' ',4D15.6)
      DESIGN=DESIGN+DES
      CORRCT=CORRCT+COR
      PREVNT=PREVNT+PRE
  100 CONTINUE
      WRITE(6,150)DESIGN,CORRCT,PREVNT
  150 FORMAT(' ',5X,'THREE COST COMPONENTS ARE'/' ',3D24.16)
      PHI=DESIGN+CORRCT+PREVNT
      PHI=-PHI
      RETURN
      END
      SUBROUTINE CPHI
      DIMENSION CC(6,3)
      DIMENSION  B(100)
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)    GEGF  20
     1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)           GEGF  30
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFA,DTP,DELTA
     1X,DE,D,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSILU,EPSIL6,EPSIL9,EPS,EPSI1,FI1,FI0,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
     4KAPG,ROB,RO,RB,RA,SCAL,TB,TD,TC,TREN,TRS,TR,TR1,TETA,TETAT,TQTG,TQ
     5TG1,TETA1,TET2,TET3,TET4,TET5,TET,TETA2,TO,TO1,TQO,TET1,T1
      DOUBLE PRECISION  TRB,T2,T22,TEX,TF,U,VC,VO,V2,VI,V3,V,VCI,VCL,XI,
     1XMV,XNO,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIE1,XIE2
     2,XSE1,XSE2,XTR1,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
     3,Z,ZI,X,XC,XS,X1,X2,XR,B
      DOUBLE PRECISION DIT,DISOLS
      DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVD,DESIGN,CORRCT,PREVNT,DES,COR,PRE,
     1SUB,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,PTM3,RTM4,VAVO,AU,UC1,UC2,DEXP
      COMMON/LIE1/CC
      COMMON    B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
      COMMON   FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSILO,EPSIL2,YSORT
      COMMON        NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NTO,NIN1,NIN2,NIN3,NIGEGA  50
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,I1,IR,IR1,IS,IS1,IT,IBP,ICDB,JCDB,KCGEGA  60
     2DB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON        KFONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEGA  80
     1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JKO,LC
      VAVO=1.
      DO 100 J=1,3
      JC=J+3
      JP=JC+3
      JQ=JP+3
      UNREL=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL=1.-UNREL
      CALL INTEG(XC,J,JQ,RMTBM)
      CTM=1./XC(JC)
      PTM=XC(JP)
```

```
      RTM=CTM*UNREL+PTM*REL
      AU=RMTBM/(RMTBM+RTM)
      VAVO=VAVO*AU
  100 CONTINUE
      WRITE(6,150)VAVO
  150 FORMAT(' ',5X,'SYSTEM AVAILABILITY=',D24.16)
      VC(1)=.97-VAVO
      VC(2)=VAVO-1.
      RETURN
      END
      SUBROUTINE JACOB
      DIMENSION CC(6,3)
      DIMENSION DT(4)
      DIMENSION AV(3),AVD(12)
      DIMENSION B(100)
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
     1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFA,DTP,DELTA
     1X,DI,D,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSILU,EPSIL6,EPSIL9,EPS,EPSI1,FI1,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
     4RAPG,ROB,RO,RB,RA,SCAL,TB,TD,TC,TREN,TPB,TR,TR1,TETA,TETAT,TQTG,TQ
     5TG1,TETA1,TET2,TET3,TET4,TET5,TET,TETA2,TQ,TQ1,TQ0,TET1,T1
      DOUBLE PRECISION  TRB,T2,T22,TCX,TS,U,VC,VO,V2,VI,V3,V,VCI,VCL,XI,
     1XNV,XNO,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIC1,XIC2
     2,XSC1,XSC2,XTR1,XIR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
     3,Z,ZI,X,XC,XS,X1,X2,XR,B
      DOUBLE PRECISION DIT,DISCLS
      DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVD,DESIGN,CORRCT,PREVNT,DES,COR,PRE,
     1SUB,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,RTM3,RTM4,VAVO,AU,UC1,UC2,DEXP
      COMMON/LIB1/CC
      COMMON/LIB2/DT
      COMMON   B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
      COMMON  FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSILO,EPSIL2,YSORT
      COMMON      NV,NC,NK,NEG,NIN,NTV,NV1,NOV,NEVL,NTO,NIN1,NIN2,NIN3,NIGEGA
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,II,IR,IR1,IS,IS1,IT,IBP,ICDB,JCDB,KCGEGA
     2DB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON      KFONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEGA
     1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JKO,LC
      IF(IT)100,101,101
  100 DT(1)=.0004
      DT(2)=.01
      DT(3)=.09
      DT(4)=15.
  101 VAVO=1.
      DO 150 J=1,3
      JC=J+3
      JP=JC+3
      JQ=JP+3
      UNREL=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL=1.-UNREL
      CALL INTEG(XC,J,JQ,RMTBM)
      CTM=1./XC(JC)
      PTM=XC(JP)
      RTM=CTM*UNREL+PTM*REL
      AV(J)=RMTBM/(RMTBM+RTM)
      VAVO=VAVO*AV(J)
```

```
      XC(J)=XC(J)+DT(1)
      UNREL1=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL1=1.-UNREL1
      CALL INTEG(XC,J,JQ,RMTBM1)
      RTM1=CTM*UNREL1+PTM*REL1
      AVD(J)=RMTBM1/(RMTBM1+RTM1)
      XC(J)=XC(J)-DT(1)
      XC(JC)=XC(JC)+DT(2)
      CTM1=1./XC(JC)
      RTM2=CTM1*UNREL+PTM*REL
      AVD(JC)=RMTBM/(RMTBM+RTM2)
      XC(JC)=XC(JC)-DT(2)
      XC(JP)=XC(JP)+DT(3)
      PTM1=XC(JP)
      RTM3=CTM*UNREL+PTM1*REL
      AVD(JP)=RMTBM/(RMTBM+RTM3)
      XC(JP)=XC(JP)-DT(3)
      XC(JQ)=XC(JQ)+DT(4)
      UNREL2=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL2=1.-UNREL2
      CALL INTEG(XC,J,JQ,RMTBM2)
      RTM4=CTM*UNREL2+PTM*REL2
      AVD(JQ)=RMTBM2/(RMTBM2+RTM4)
      XC(JQ)=XC(JQ)-DT(4)
  150 CONTINUE
      UC1=.97-VAV3
      UC2=VAV3-1.
      A(1,1)=(.97-AVD(1)*AV(2)*AV(3)-UC1)/DT(1)
      A(1,2)=(.97-AV(1)*AVD(2)*AV(3)-UC1)/DT(1)
      A(1,3)=(.97-AV(1)*AV(2)*AVD(3)-UC1)/DT(1)
      A(1,4)=(.97-AVD(4)*AV(2)*AV(3)-UC1)/DT(2)
      A(1,5)=(.97-AV(1)*AVD(5)*AV(3)-UC1)/DT(2)
      A(1,6)=(.97-AV(1)*AV(2)*AVD(6)-UC1)/DT(2)
      A(1,7)=(.97-AVD(7)*AV(2)*AV(3)-UC1)/DT(3)
      A(1,8)=(.97-AV(1)*AVD(8)*AV(3)-UC1)/DT(3)
      A(1,9)=(.97-AV(1)*AV(2)*AVD(9)-UC1)/DT(3)
      A(1,10)=(.97-AVD(10)*AV(2)*AV(3)-UC1)/DT(4)
      A(1,11)=(.97-AV(1)*AVD(11)*AV(3)-UC1)/DT(4)
      A(1,12)=(.97-AV(1)*AV(2)*AVD(12)-UC1)/DT(4)
      A(2,1)=(AVD(1)*AV(2)*AV(3)-1.-UC2)/DT(1)
      A(2,2)=(AV(1)*AVD(2)*AV(3)-1.-UC2)/DT(1)
      A(2,3)=(AV(1)*AV(2)*AVD(3)-1.-UC2)/DT(1)
      A(2,4)=(AVD(4)*AV(2)*AV(3)-1.-UC2)/DT(2)
      A(2,5)=(AV(1)*AVD(5)*AV(3)-1.-UC2)/DT(2)
      A(2,6)=(AV(1)*AV(2)*AVD(6)-1.-UC2)/DT(2)
      A(2,7)=(AVD(7)*AV(2)*AV(3)-1.-UC2)/DT(3)
      A(2,8)=(AV(1)*AVD(8)*AV(3)-1.-UC2)/DT(3)
      A(2,9)=(AV(1)*AV(2)*AVD(9)-1.-UC2)/DT(3)
      A(2,10)=(AVD(10)*AV(2)*AV(3)-1.-UC2)/DT(4)
      A(2,11)=(AV(1)*AVD(11)*AV(3)-1.-UC2)/DT(4)
      A(2,12)=(AV(1)*AV(2)*AVD(12)-1.-UC2)/DT(4)
      WRITE(6,200)((A(I,J),J=1,12),I=1,2)
  200 FORMAT(' ','THE PARTIAL DERIVATIVES OF THE CONSTRAINTS ARE'/
     1(' ',6D15.6))
      RETURN
      END
      SUBROUTINE GRADEI
      DIMENSION CC(6,3)
      DIMENSION DT(4)
      DIMENSION B(100)
```

```
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
     1,Y(150),C(150),VC(50),IPAS(50),IHB(100),IVC(50),IVA(100)
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFA,DTP,DELTA
     1X,DT,D,EPSIL,EPSIL0,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSIL8,EPSIL6,EPSIL9,EPS,EPSI1,FI1,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHIC,PHI1,PHI2,PSIT,TOP,PSI4,PENAL,PE1,PE2,PE3,
     4KAPG,ROB,RO,RB,RA,SCAL,TB,TD,TC,TRIN,TR2,TR,TR1,TETA,TETAT,TOTG,TQ
     5TG1,TETA1,TET2,TET3,TET4,TET5,TET,TETA2,TQ,TQ1,TOD,TET1,TI
      DOUBLE PRECISION  TRB,T2,T22,TEX,TE,U,VC,VO,V2,VI,V3,V,VCI,VCL,XI,
     1XNV,XND,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XJC,XSB1,XSB2,XMK,XIC1,XIC2
     2,XSC1,XSC2,XTR1,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORT0,YRO,YR
     3,Z,ZI,X,XC,XS,X1,X2,XP,B
      DOUBLE PRECISION DIT,DISOLS
      DOUBLE PRECISION DMOD,DSORT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVO,DESIGN,CORRCT,PREVNT,DES,COR,PRE,
     1SUB,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,RTM3,PTM4,VAVO,AU,UC1,UC2,DEXP
      COMMON/LIB1/CC
      COMMON/LIB2/DT
      COMMON    B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
      COMMON  FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSIL0,EPSIL2,YSORT
      COMMON       NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NTO,NIN1,NIN2,NIN3,NIGEGA
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,II,IR,IR1,IS,IS1,IT,IBP,ICOB,JCOB,KCGEGA
     2DB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON      KFONC,KGRAD,KCONT,KINV1,KINV2,KCOBA,KJACO,KMAX1,KMAX2,KGEGA
     1REN1,KREN2,KINV,KCOBA1,KREN11,KREN21,IDIREC,JKO,LC
      DO 100 J=1,3
      JC=J+3
      JP=JC+3
      JQ=JP+3
      UNREL=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL=1.-UNREL
      CALL  INTEG(XC,J,JQ,RMTBM)
      UMTBM=RMTBM/UNREL
      SMTBM=RMTBM/REL
      CTM=1./XC(JC)
      PTM=XC(JP)
      RTM=CTM*UNREL+PTM*REL
      XC(J)=XC(J)+DT(1)
      UNREL1=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL1=1.-UNREL1
      CALL  INTEG(XC,J,JQ,RMTBM1)
      UMTBM1=RMTBM1/UNREL1
      SMTBM1=RMTBM1/REL1
      RTM1=CTM*UNREL1+PTM*REL1
      C(J)=CC(1,J)*(RMTBM1-RMTBM)+CC(2,J)*(1./RTM1-1./RTM)+1500.*(
     1CC(4,J)*CTM)**2*(1./UMTBM1-1./UMTBM)+1500.*(CC(5,J)*PTM-CC(6,J))
     2*(1./SMTBM1-1./SMTBM)
      C(J)=-C(J)/DT(1)
      XC(J)=XC(J)-DT(1)
      XC(JC)=XC(JC)+DT(2)
      CTM1=1./XC(JC)
      RTM2=CTM1*UNREL+PTM*REL
      C(JC)=CC(2,J)*(1./RTM2-1./RTM)+(1500./UMTBM)*((CC(4,J)*CTM1)**2
     1-(CC(4,J)*CTM)**2)
      C(JC)=-C(JC)/DT(2)
      XC(JC)=XC(JC)-DT(2)
      XC(JP)=XC(JP)+DT(3)
      PTM1=XC(JP)
```

```
      RTM3=CTM*UNREL+PTM1*REL
      C(JP)=CC(2,J)*(1./RTM3-1./RTM)+(1500./SMTBM)*CC(5,J)*(PTM1-PTM)
      C(JP)=-C(JP)/DT(3)
      XC(JP)=XC(JP)-DT(3)
      XC(JQ)=XC(JQ)+DT(4)
      UNREL2=(1.-DEXP(-XC(J)*XC(JQ)))**2
      REL2=1.-UNREL2
      CALL INTEG(XC,J,JQ,RMTBM2)
      UMTBM2=RMTBM2/UNREL2
      SMTBM2=RMTBM2/REL2
      RTM4=CTM*UNREL2+PTM*REL2
      C(JQ)=CC(1,J)*(RMTBM2-RMTBM)+CC(2,J)*(1./RTM4-1./RTM)+1500.*(
     1CC(4,J)*CTM)**2*(1./UMTBM2-1./UMTBM)+1500.*(CC(5,J)*PTM-CC(6,J))
     2*(1./SMTBM2-1./SMTBM)
      C(JQ)=-C(JQ)/DT(4)
      XC(JQ)=XC(JQ)-DT(4)
  100 CONTINUE
      WRITE(6,200)(C(I),I=1,12)
  200 FORMAT(' ','THE PARTIAL DERIVATIVES OF THE OBJ. FN. ARE'/(' ',
     16D15.6))
      RETURN
      END
      SUBROUTINE INTEG(XA,I,IQ,FSUB)
      DIMENSION XA(150)
      DOUBLE PRECISION XA,ZERO,RI,DINTVL,RM,RF,FSUB,DEXP
      ZERO=0.
      RI=.5
      DINTVL=(XA(IQ)-ZERO)/100.
   10 ZERO=ZERO+DINTVL
      RM=1.-(1.-DEXP(-XA(I)*ZERO))**2
      RI=RI+RM
      IF(ZERO.LT.(XA(IQ)-DINTVL)) GO TO 10
      RF=1.-(1.-DEXP(-XA(I)*XA(IQ)))**2
      FSUB=DINTVL*(RI+RF/2.)
      RETURN
      END
```

A2.2 GRG : USER SUPPLIED SUBROUTINES FOR EXAMPLES 2

In these subroutines, λ_j's, μ_j's, $(t_p)_j$'s, and T_j's are used as original problem variables.

```
      SUBROUTINE PHIX
      DIMENSION CC(6,3)
      DIMENSION  B(100)
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)   GEGF  20
     1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)          GEGF  30
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFA,DIP,DELTA
     1X,DI,D,EPSIL,EPSILO,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSIL8,EPSIL6,EPSIL9,EPS,EPSI1,FI1,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSIT,TOP,PS14,PENAL,PE1,PE2,PE3,
     4RAPG,ROB,RO,RB,RA,SCAL,TB,TD,TC,TREN,TRE,TR,TR1,TETA,TETAT,TQTG,TQ
     5TGL,TETA1,TET2,TET3,TET4,TET5,TLT,TETA2,TQ,TQ1,TQO,TET1,T1
      DOUBLE PRECISION  TRB,T2,T22,TEX,TS,U,VC,VO,V2,VI,V3,V,VCI,VCL,XI,
     1XNV,XNO,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIP1,XIP2
     2,XST1,XST2,XTR1,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
     3,Z,ZI,X,XP,XS,X1,X2,XR,B
      DOUBLE PRECISION DIT,DISOLS
      DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVD,DESIGN,CORRCT,PREVNT,DES,COR,PRE,
     1SUB,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,RTM3,RTM4,VAVO,AU,UC1,UC2,DEXP
      COMMON/LISI/CC
      COMMON    B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
      COMMON   FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSILO,EPSIL2,YSORT
      COMMON       NV,NC,NK,NEG,NIN,NIV,NV1,NEV,NEVL,NTO,NIN1,NIN2,NIN3,NIGEGA  50
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,II,IR,IR1,IS,IS1,IT,IBP,ICOB,JCOB,KCGEGA  60
     2DB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON       KFONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEGA  80
     1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JKO,LC
      IF(IT) 10,11,11
   10 CC(1,1)=1.8
      CC(1,2)=1.3
      CC(1,3)=2.
      CC(2,1)=200.
      CC(2,2)=170.
      CC(2,3)=250.
      CC(3,1)=5.
      CC(3,2)=5.
      CC(3,3)=5.
      CC(4,1)=2.
      CC(4,2)=2.5
      CC(4,3)=3.
      CC(5,1)=40.
      CC(5,2)=100.
      CC(5,3)=50.
      CC(6,1)=3.
      CC(6,2)=4.
      CC(6,3)=2.
   11 DESIGN=0.
      CORRCT=0.
      PREVNT=0.
      DO 100 J=1,3
      JC=J+3
      JP=JC+3
      JQ=JP+3
      UNREL=(1.-DEXP(-XC(JQ)**2*XC(J)))**2
      REL=1.-UNREL
      CALL INTEG(XC,J,JQ,RMTBM)
      UMTBM=RMTBM/UNREL
      SMTBM=RMTBM/REL
```

```
      CTM=XC(JC)
      PTM=XC(JP)
      RTM=CTM*UNREL+PTM*REL
      WRITE(6,130)RMTBM,UMTBM,SMTBM,RTM
  130 FORMAT(/' ',4D15.6)
      DES=CC(1,J)*RMTBM+CC(2,J)/RTM-CC(3,J)
      COR=(1500./UMTBM)*(CC(4,J)*CTM)**2
      PRE=(1500./SMTBM)*(CC(5,J)*PTM-CC(6,J))
      SUB=DES+COR+PRE
      WRITE(6,140)DES,COR,PRE,SUB
  140 FORMAT(' ',4D15.6)
      DESIGN=DESIGN+DES
      CORRCT=CORRCT+COR
      PREVNT=PREVNT+PRE
  100 CONTINUE
      WRITE(6,150)DESIGN,CORRCT,PREVNT
  150 FORMAT(' ',5X,'THE THREE COST COMPONENTS ARE'/' ',3D24.16)
      PHI=DESIGN+CORRCT+PREVNT
      PHI=-PHI
      RETURN
      END
      SUBROUTINE CPHI
      DIMENSION CC(6,3)
      DIMENSION  B(100)
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)    GEGF
     1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)           GEGF
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFA,DTP,DELTA
     1X,DI,D,EPSIL,EPSIL0,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSILU,EPSIL6,EPSIL9,EPS,EPSI1,FI1,FI0,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHI0,PHI1,PHI2,PSIT,TQP,PSI4,PENAL,PE1,PE2,PE3,
     4RAPG,ROB,RO,RB,RA,SCAL,TB,TD,TC,TREN,TRE,TR,TR1,TETA,TETAT,TQTG,TQ
     5TG1,TETA1,TET2,TET3,TET4,TET5,TET,TETA2,TQ,TQ1,TQ0,TET1,T1
      DOUBLE PRECISION  TRB,T2,T22,TEX,TE,U,VC,V0,V2,VI,V3,V,VCI,VCL,XI,
     1XNV,XN0,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIE1,XIE2
     2,XSE1,XSE2,XTR1,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORT0,YR0,YR
     3,Z,ZI,X,XC,XS,X1,X2,XR,B
      DOUBLE PRECISION DIT,DISOLS
      DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVD,DESIGN,CORRCT,PREVNT,DES,COR,PRE,
     1SUB,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,RTM3,RTM4,VAV0,AU,UC1,UC2,DEXP
      COMMON/LIE1/CC
      COMMON    B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
      COMMON   FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSIL0,EPSIL2,YSORT
      COMMON       NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NT0,NIN1,NIN2,NIN3,NIGEGA
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,I1,IR,IR1,IS,IS1,IT,I3P,ICOB,JCOB,KCGEGA
     2DB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON      KFONC,KGRAD,KCONT,KINV1,KINV2,KCDBA,KJACO,KMAX1,KMAX2,KGEGA
     1REN1,KREN2,KINV,KCDBA1,KREN11,KREN21,IDIREC,JK0,LC
      VAV0=1.
      DO 100 J=1,3
      JC=J+3
      JP=JC+3
      JQ=JP+3
      UNREL=(1.-DEXP(-XC(JQ)**2*XC(J)))**2
      REL=1.-UNREL
      CALL INTEG(XC,J,JQ,RMTBM)
      CTM=XC(JC)
      PTM=XC(JP)
```

```
      RTM=CTM*UNREL+PTM*REL
      AU=RMTBM/(RMTBM+RTM)
      VAVO=VAVO*AU
  100 CONTINUE
      WRITE(6,150)VAVO
  150 FORMAT(' ',5X,'SYSTEM AVAILABILITY=',D24.16)
      VC(1)=.95-VAVO
      VC(2)=VAVO-1.
      RETURN
      END
      SUBROUTINE JACO3
      DIMENSION CC(6,3)
      DIMENSION DT(4)
      DIMENSION AV(3),AVD(12)
      DIMENSION  B(100)
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)    GEGF  20
     1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)           GEGF  30
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TRA,C,DELTFI,DELTFA,DTP,DELTA
     1X,DI,D,EPSIL,EPSIL0,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSILU,EPSIL6,EPSIL9,EPS,EPSI1,FI1,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHIO,PHI1,PHI2,PSII,TOP,PSI4,PENAL,PE1,PE2,PE3,
     4RAPG,RO3,RO,RB,RA,SCAL,TB,TD,TC,TREN,TRE,TF,TR1,TETA,TETAT,TQTG,TQ
     5TG1,TETA1,TET2,TET3,TET4,TET5,TET,TETA2,TQ,TQ1,TQ0,TET1,T1
      DOUBLE PRECISION  TRB,T2,T22,TEX,TE,U,VC,V0,V2,VI,V3,V,VCI,VCL,XI,
     1XNV,XMO,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XIE1,XIE2
     2,XSE1,XSE2,XTR1,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORTO,YRO,YR
     3,Z,ZI,X,XC,XS,X1,X2,XR,B
      DOUBLE PRECISION DIT,DISOLS
      DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVD,DESIGN,CORRCT,PREVNT,DES,COR,PRE,
     1SUB,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,RTM3,RTM4,VAVO,AU,UC1,UC2,DEXP
      COMMON/LIE1/CC
      COMMON/LIE2/DT
      COMMON    B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,FTA
      COMMON  FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSIL0,EPSIL2,YSORT
      COMMON        NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NTO,NIN1,NIN2,NIN3,NIGEGA  50
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,II,IR,IR1,IS,IS1,IT,IEP,ICOB,JCOB,KCGEGA  60
     2OB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON        KFONC,KGRAD,KCONT,KINV1,KINV2,KCOBA,KJACO,KMAX1,KMAX2,KGEGA  80
     1REN1,KREN2,KINV,KCOBA1,KREN11,KREN21,IOIREC,JKO,LC
      IF(IT)100,101,101
  100 DT(1)=.000012
      DT(2)=.4
      DT(3)=.2
      DT(4)=2.
  101 VAVO=1.
      DO 150 J=1,3
      JC=J+3
      JP=JC+3
      JO=JP+3
      UNREL=(1.-DEXP(-XC(JO)**2*XC(J)))**2
      REL=1.-UNREL
      CALL INTEG(XC,J,JO,RMTBM)
      CTM=XC(JC)
      PTM=XC(JP)
      RTM=CTM*UNREL+PTM*REL
      AV(J)=RMTBM/(RMTBM+RTM)
      VAVO=VAVO*AV(J)
```

```
      XC(J)=XC(J)+DT(1)
      UNREL1=(1.-DEXP(-XC(JQ)**2*XC(J)))**2
      REL1=1.-UNREL1
      CALL  INTEG(XC,J,JQ,RMTBM1)
      RTM1=CTM*UNREL1+PTM*REL1
      AVD(J)=RMTBM1/(RMTBM1+RTM1)
      XC(J)=XC(J)-DT(1)
      XC(JC)=XC(JC)+DT(2)
      CTM1=1./XC(JC)
      RTM2=CTM1*UNREL+PTM*REL
      AVD(JC)=RMTBM/(RMTBM+RTM2)
      XC(JC)=XC(JC)-DT(2)
      XC(JP)=XC(JP)+DT(3)
      PTM1=XC(JP)
      RTM3=CTM*UNREL+PTM1*REL
      AVD(JP)=RMTBM/(RMTBM+RTM3)
      XC(JP)=XC(JP)-DT(3)
      XC(JQ)=XC(JQ)+DT(4)
      UNREL2=(1.-DEXP(-XC(JQ)**2*XC(J)))**2
      REL2=1.-UNREL2
      CALL  INTEG(XC,J,JQ,RMTBM2)
      RTM4=CTM*UNREL2+PTM*REL2
      AVD(JQ)=RMTBM2/(RMTBM2+RTM4)
      XC(JQ)=XC(JQ)-DT(4)
  150 CONTINUE
      UC1=.93-VAVD
      UC2=VAVD-1.
      A(1,1)=(.93-AVD(1)*AV(2)*AV(3)-UC1)/DT(1)
      A(1,2)=(.93-AV(1)*AVD(2)*AV(3)-UC1)/DT(1)
      A(1,3)=(.93-AV(1)*AV(2)*AVD(3)-UC1)/DT(1)
      A(1,4)=(.93-AVD(4)*AV(2)*AV(3)-UC1)/DT(2)
      A(1,5)=(.93-AV(1)*AVD(5)*AV(3)-UC1)/DT(2)
      A(1,6)=(.93-AV(1)*AV(2)*AVD(6)-UC1)/DT(2)
      A(1,7)=(.93-AVD(7)*AV(2)*AV(3)-UC1)/DT(3)
      A(1,8)=(.93-AV(1)*AVD(8)*AV(3)-UC1)/DT(3)
      A(1,9)=(.93-AV(1)*AV(2)*AVD(9)-UC1)/DT(3)
      A(1,10)=(.93-AVD(10)*AV(2)*AV(3)-UC1)/DT(4)
      A(1,11)=(.93-AV(1)*AVD(11)*AV(3)-UC1)/DT(4)
      A(1,12)=(.93-AV(1)*AV(2)*AVD(12)-UC1)/DT(4)
      A(2,1)=(AVD(1)*AV(2)*AV(3)-1.-UC2)/DT(1)
      A(2,2)=(AV(1)*AVD(2)*AV(3)-1.-UC2)/DT(1)
      A(2,3)=(AV(1)*AV(2)*AVD(3)-1.-UC2)/DT(1)
      A(2,4)=(AVD(4)*AV(2)*AV(3)-1.-UC2)/DT(2)
      A(2,5)=(AV(1)*AVD(5)*AV(3)-1.-UC2)/DT(2)
      A(2,6)=(AV(1)*AV(2)*AVD(6)-1.-UC2)/DT(2)
      A(2,7)=(AVD(7)*AV(2)*AV(3)-1.-UC2)/DT(3)
      A(2,8)=(AV(1)*AVD(8)*AV(3)-1.-UC2)/DT(3)
      A(2,9)=(AV(1)*AV(2)*AVD(9)-1.-UC2)/DT(3)
      A(2,10)=(AVD(10)*AV(2)*AV(3)-1.-UC2)/DT(4)
      A(2,11)=(AV(1)*AVD(11)*AV(3)-1.-UC2)/DT(4)
      A(2,12)=(AV(1)*AV(2)*AVD(12)-1.-UC2)/DT(4)
      WRITE(6,200)((A(I,J),J=1,12),I=1,2)
  200 FORMAT(' ','THE PARTIAL DERIVATIVES OF THE CONSTRAINTS ARE'/
     1(' ',6D15.6))
      RETURN
      END
      SUBROUTINE GRADFI
      DIMENSION CC(6,3)
      DIMENSION DT(4)
      DIMENSION  B(100)
```

```
      DIMENSION  A(50,100),ALFA(50,50),X(150),XC(150),XI(150),XS(150)
     1,Y(150),C(150),VC(50),IBAS(50),IHB(100),IVC(50),IVA(100)
      DOUBLE PRECISION  A,ALFA,ALC,ALA,ALB,TBA,C,DELTFI,DELTFA,DTP,DELTA
     1X,D1,D,EPSIL,EPSIL0,EPSIL2,ETA,EPSIL1,EPSIL3,EPSIL4,EPSIL5,EPSIL7,
     2EPSIL8,EPSIL6,EPSIL9,EPS,EPSI1,FI1,F10,F20,F1,F2,G,GAMA,GNORM,GKS,
     3GK,PHI,PSI,PSI3,PC,PHI0,PHI1,PHI2,PSIF,FOP,PSI4,PENAL,PE1,PE2,PE3,
     4RAPG,ROB,R0,RB,RA,SCAL,TB,TD,TC,TREN,TRE,TE,TR1,TETA,TETAT,TOTG,TO
     5TG1,TETA1,TET2,TET3,TET4,TET5,TET,TETA2,T0,T01,T00,TET1,T1
      DOUBLE PRECISION   TPB,T2,T22,TEX,TE,U,VC,V0,V2,VI,V3,V,VC1,VCL,XI,
     1XNV,XN0,XSB,XSA,XNORM,XNOR,XINORM,XMAJ,XSC,XSB1,XSB2,XMK,XI31,XI32
     2,XSS1,XSS2,XTR1,XTR,XTR2,XT,XIT,Y,YSORT,YSORT1,YNORM,YSORT0,YR0,YR
     3,Z,ZI,X,XC,XS,X1,X2,XR,B
      DOUBLE PRECISION DIT,DISOLS
      DOUBLE PRECISION DMOD,DSQRT,DABS,DMIN1,DMAX1
      DOUBLE PRECISION CC,DT,AV,AVD,DESIGN,CORRCT,PREVNT,DFS,COR,PRE,
     1SUB,UNREL,UNREL1,UNREL2,REL,REL1,REL2,RMTBM,RMTBM1,RMTBM2,UMTBM,
     2UMTBM1,UMTBM2,SMTBM,SMTBM1,SMTBM2,CTM,CTM1,PTM,PTM1,RTM,RTM1,
     3RTM2,RTM3,RTM4,VAV0,AU,UC1,UC2,DEXP
      COMMON/LIE1/CC
      COMMON/LIE2/DT
      COMMON    B,A,ALFA,X,XC,XI,XS,Y,C,VC,DELTFI,ETA
      COMMON  FI1,PHI,PSI,PSI3,TB,TD,TC,EPSIL,EPSIL0,EPSIL2,YSORT
      COMMON       NV,NC,NK,NEG,NIN,NTV,NV1,NEV,NEVL,NT0,MIN1,MIN2,MIN3,NIGEGA
     1N4,NVNIN1,NVNIN2,NVNIN3,INDEX,II,IR,IR1,IS,IS1,IT,IBP,ICDB,JCDB,KCGEGA
     2DB,KFIL,KLIN,KREN,KD,IBAS,IHB,IVC,IVA,IVB
      COMMON       KFONC,KGRAD,KCONT,KINV1,KINV2,KCOBA,KJACO,KMAX1,KMAX2,KGEGA
     1REN1,KREN2,KINV,KCOBA1,KREN11,KREN21,IDIREC,JK0,LC
      DO 100 J=1,3
      JC=J+3
      JP=JC+3
      JQ=JP+3
      UNREL=(1.-DEXP(-XC(JQ)**2*XC(J)))**2
      REL=1.-UNREL
      CALL  INTEG(XC,J,JQ,RMTBM)
      UMTBM=RMTBM/UNREL
      SMTBM=RMTBM/REL
      CTM=XC(JC)
      PTM=XC(JP)
      RTM=CTM*UNREL+PTM*REL
      XC(J)=XC(J)+DT(1)
      UNREL1=(1.-DEXP(-XC(JQ)**2*XC(J)))**2
      REL1=1.-UNREL1
      CALL  INTEG(XC,J,JQ,RMTBM1)
      UMTBM1=RMTBM1/UNREL1
      SMTBM1=RMTBM1/REL1
      RTM1=CTM*UNREL1+PTM*REL1
      C(J)=CC(1,J)*(RMTBM1-RMTBM)+CC(2,J)*(1./RTM1-1./RTM)+1500.*
     1(CC(4,J)*CTM)**2*(1./UMTBM1-1./UMTBM)+1500.*(CC(5,J)*PTM-CC(6,J))
     2*(1./SMTBM1-1./SMTBM)
      C(J)=-C(J)/DT(1)
      XC(J)=XC(J)-DT(1)
      XC(JC)=XC(JC)+DT(2)
      CTM1=XC(JC)
      RTM2=CTM1*UNREL+PTM*REL
      C(JC)=CC(2,J)*(1./RTM2-1./RTM)+(1500./UMTBM)*((CC(4,J)*CTM1)**2-
     1(CC(4,J)*CTM)**2)
      C(JC)=-C(JC)/DT(2)
      XC(JC)=XC(JC)-DT(2)
      XC(JP)=XC(JP)+DT(3)
      PTM1=XC(JP)
```

```
      RTM3=CTM*UNREL+PTM1*REL
      C(JP)=CC(2,J)*(1./RTM3-1./RTM)+(1500./SMTBM)*CC(5,J)*(PTM1-PTM)
      C(JP)=-C(JP)/DT(3)
      XC(JP)=XC(JP)-DT(3)
      XC(JQ)=XC(JQ)+DT(4)
      UNREL2=(1.-DEXP(-XC(JQ)**2*XC(J)))**2
      REL2=1.-UNREL2
      CALL INTEG(XC,J,JQ,RMTBM2)
      UMTBM2=RMTBM2/UNREL2
      SMTBM2=RMTBM2/REL2
      RTM4=CTM*UNREL2+PTM*REL2
      C(JQ)=CC(1,J)*(RMTBM2-RMTBM)+CC(2,J)*(1./RTM4-1./RTM)+1500.*(
     1CC(4,J)*CTM)**2*(1./UMTBM2-1./UMTBM)+1500.*(CC(5,J)*PTM-CC(6,J))
     2*(1./SMTBM2-1./SMTBM)
      C(JQ)=-C(JQ)/DT(4)
      XC(JQ)=XC(JQ)-DT(4)
  100 CONTINUE
      WRITE(6,200)(C(I),I=1,12)
  200 FORMAT(' ','THE PARTIAL DERIVATIVES OF THE OBJ. FN. ARE'/(' ',
     16D15.6))
      RETURN
      END
      SUBROUTINE INTEG(XA,I,IQ,FSUB)
      DIMENSION XA(150)
      DOUBLE PRECISION XA,ZERO,RI,DINTVL,RM,RF,FSUB,DEXP
      ZERO=0.
      RI=.5
      DINTVL=(XA(IQ)-ZERO)/100.
   10 ZERO=ZERO+DINTVL
      RM=1.-(1.-DEXP(-ZERO**2*XA(I)))**2
      RI=RI+RM
      IF(ZERO.LT.(XA(IQ)-DINTVL)) GO TO 10
      RF=1.-(1.-DEXP(-XA(IQ)**2*XA(I)))**2
      FSUB=DINTVL*(RI+RF/2.)
      RETURN
      END
```

A2.3 SUMT : LAI'S VERSION WITH USER SUPPLIED SUBROUTINES FOR EXAMPLE 1

```
C                                                                       HJS00010
C                    ****************************                       HJS00020
C                                                                       HJS00030
C     THIS PROGRAM IS FOR OPTIMIZING CONSTRAINED MINIMIZATION PROBLEM HJS00040
C BY A COMBINATIONAL USE OF HOOKE AND JEEVES PATTERN SEARCH TECHNIQUE HJS00050
C AND SUMT FORMULATION . WHEN THE SEARCH GETS OUT OF THE FEASIBLE      HJS00060
C REGION , IT WILL BE PULLED BACK BY A HEURISTIC PROGRAMMING TECHNIQUEHJS00070
C EXECUTED BY THE SUBROUTINE BACK .                                     HJS00080
C     THE ORIGINAL IDEALS CAME FROM ..                                  HJS00090
C          SEARCH TECHNIQUE ... HOOK AND JEEVES .                       HJS00100
C          SUMT FORMULATION ... FIACCO AND MCCORMICK .                  HJS00110
C          PULL BACK TECHNIQUE ...PAVIANI AND HIMMELBLAU .              HJS00120
C     THE NECESSARY REFERENCE DOCUMENTS CAN BE SEEN IN MY MASTER        HJS00130
C REPORT .                                                              HJS00140
C                                       K. C. LAI , IE , KSU .          HJS00150
C                                                                       HJS00160
C     ************************************************************      HJS00170
C                                                                       HJS00180
C     **INPUT-OUTPUT VARIABLES ...                                      HJS00190
C          NOPM .. NO. OF SUBPROBLEMS INPUT .                           HJS00200
C          NAME1,NAME2,NAME3 .. 3 PARTS OF PROBLEM NAME, USER MAY USEHJS00210
C                                 ANY 6 CHARACTERS TO NAME THE PROBLEM.HJS00220
C          N .. NO. OF VARIABLES OF THE PROBLEM .                       HJS00230
C          MG .. NO. OF INEQUALITY CONSTRAINTS G(J) .GE. 0. .           HJS00240
C          MH .. NO. OF EQUALITY CONSTRAINTS H(K) .EQ. 0. .             HJS00250
C          R .. PENALTY COEFFICIENT FOR SUMT FORMULATION .              HJS00260
C               OPTION --R .LE. 0.0, WILL USE A COMPUTED VALUE .        HJS00270
C          RATIO .. REDUCING RATE FOR R FROM STAGE TO STAGE .           HJS00280
C                   OPTION -- RATIO .LE. 0.0,  WILL USE RATIO=4.0 .     HJS00290
C          ITMAX .. INPUT WITHIN-STAGE ITERATION MAXIMUM NO.            HJS00300
C          INCUT .. STOPPING CRITERION FOR STAGE ITERATION, NO. OF      HJS00310
C                   CUT-DOWN STEP-SIZE OPERATION .                      HJS00320
C          THETA .. FINAL STOPPING CRITERION, SUJESTED VALUE 10**(-4)HJS00330
C                   OR ABOUT .                                          HJS00340
C          MAXP .. INPUT MAXIMUM NO. OF STAGES , IF EXCEEDED, STOP .    HJS00350
C          X(I) .. (I)TH DIMENSION OF DECISION VARIABLE .               HJS00360
C          D(I) .. (I)TH DIMENSION OF STEP SIZE .                       HJS00370
C          OX(I) .. (I)TH DIMENSION OF (ESTIMATED VALUE) OPTIMUM.       HJS00380
C          ISIZE .. OPTION CODE FOR INITIAL STEP-SIZE SET UP ..         HJS00390
C                     0 -- USE INPUT D(I) VALUES.                       HJS00400
C                     1 -- USE COMPUTED D(I) =0.02*OX(I).               HJS00410
C          ICUT .. OPTION CODE FOR STAGE STARTING STEP-SIZE             HJS00420
C                 SET UP .. 0 -- ALL USE INPUT D(I) VALUE.              HJS00430
C                           1 -- USE INITIAL D(I)/K FOR (K)TH STAGE.HJS00440
C          P .. P FUNCTION VALUE .                                      HJS00450
C          Y .. F FUNCTION VALUE .                                      HJS00460
C          YSTOP .. COMPUTED VALUE OF FINAL-STOPPING DETERMINATOR .     HJS00470
C          IDPM .. SEQUENCE NO. OF SUBPROBLEMS OUTPUT .                 HJS00480
C          NOR .. NO. OF STAGES UP TO CURRENT STAGE .                   HJS00490
C          B .. TOLERANCE LIMIT FOR VIOLATIONS .                        HJS00500
C          FY .. MINIMUM Y GOT SO FAR .                                 HJS00510
C          FP .. MINIMUM P GOT SO FAR .                                 HJS00520
C          G(J) .. (J)TH INEQUALITY CONSTRAINT VALUE .                  HJS00530
C          H(K) .. (K)TH EQUALITY CONSTRAINT VALUE .                    HJS00540
C          ITER .. WITHIN STAGE ITERATION NO.                           HJS00550
C          NOTE .. CUMULATED ITERATION NO.                              HJS00560
C          NOCUT .. NO. OF CUT DOWN STEP-SIZE OPERATION WITHIN STAGE.HJS00570
C          NOEXP .. NO. OF SUCCESSFUL EXPLORATORY MOVES.                HJS00580
C          NOPAT .. NO. OF SUCCESSFUL PATTERN MOVES.                    HJS00590
C          NOR .. NO. OF TIMES OF PULLING BACK PROCEDURES.              HJS00600
```

```
C
C                    ****************************
C
C      THIS PROGRAM IS FOR OPTIMIZING CONSTRAINED MINIMIZATION PROBLEM
C  BY A COMBINATIONAL USE OF HOOKE AND JEEVES PATTERN SEARCH TECHNIQUE
C  AND SUMT FORMULATION . WHEN THE SEARCH GETS OUT OF THE FEASIBLE
C  REGION , IT WILL BE PULLED BACK BY A HEURISTIC PROGRAMMING TECHNIQUE
C  EXECUTED BY THE SUBROUTINE PACK .
C      THE ORIGINAL IDEALS CAME FROM ..
C           SEARCH TECHNIQUE ... HOOK AND JEEVES .
C           SUMT FORMULATION ... FIACCO AND MCCORMICK .
C           PULL BACK TECHNIQUE ...PAVIANI AND HIMMELBLAU .
C      THE NECESSARY REFERENCE DOCUMENTS CAN BE SEEN IN MY MASTER
C  REPORT .
C                                     K. C. LAI , IE , KSU .
C
C      ***********************************************************
C
C      **INPUT-OUTPUT VARIABLES ...
C           NUPM .. NO. OF SUBPROBLEMS INPUT .
C           NAME1,NAME2,NAME3 .. 3 PARTS OF PROBLEM NAME, USER MAY USE
C                              ANY 6 CHARACTERS TO NAME THE PROBLEM.
C           N .. NO. OF VARIABLES OF THE PROBLEM .
C           MG .. NO. OF INEQUALITY CONSTRAINTS G(J) .GE. 0. .
C           MH .. NO. OF EQUALITY CONSTRAINTS H(K) .EQ. 0. .
C           R .. PENALTY COEFFICIENT FOR SUMT FORMULATION .
C                OPTION --R .LE. 0.0, WILL USE A COMPUTED VALUE .
C           RATIO .. REDUCING RATE FOR R FROM STAGE TO STAGE .
C                OPTION -- RATIO .LE. 0.0,  WILL USE RATIO=4.0 .
C           ITMAX .. INPUT WITHIN-STAGE ITERATION MAXIMUM NO.
C           INCUT .. STOPPING CRITERION FOR STAGE ITERATION, NO. OF
C                CUT-DOWN STEP-SIZE OPERATION .
C           THETA .. FINAL STOPPING CRITERION, SUGESTED VALUE 10**(-4)
C                OR ABOUT .
C           MAXP .. INPUT MAXIMUM NO. OF STAGES , IF EXCEEDED, STOP .
C           X(I) .. (I)TH DIMENSION OF DECISION VARIABLE .
C           D(I) .. (I)TH DIMENSION OF STEP SIZE .
C           OX(I) .. (I)TH DIMENSION OF (ESTIMATED VALUE) OPTIMUM.
C           ISIZE .. OPTION CODE FOR INITIAL STEP-SIZE SET UP ..
C                  0 -- USE INPUT D(I) VALUES.
C                  1 -- USE COMPUTED D(I) =0.02*OX(I).
C           ICUT .. OPTION CODE FOR STAGE STARTING STEP-SIZE
C                SET UP .. 0 -- ALL USE INPUT D(I) VALUE.
C                          1 -- USE INITIAL D(I)/K FOR (K)TH STAGE.
C           P .. P FUNCTION VALUE .
C           Y .. F FUNCTION VALUE .
C           YSTOP .. COMPUTED VALUE OF FINAL-STOPPING DETERMINATOR .
C           IDPM .. SEQUENCE NO. OF SUBPROBLEMS OUTPUT .
C           NOR .. NO. OF STAGES UP TO CURRENT STAGE .
C           B .. TOLERANCE LIMIT FOR VIOLATIONS .
C           FY .. MINIMUM Y GOT SO FAR .
C           FP .. MINIMUM P GOT SO FAR .
C           G(J) .. (J)TH INEQUALITY CONSTRAINT VALUE .
C           H(K) .. (K)TH EQUALITY CONSTRAINT VALUE .
C           ITER .. WITHIN STAGE ITERATION NO.
C           NCIT .. CUMULATED ITERATION NO.
C           NCCUT .. NO. OF CUT DOWN STEP-SIZE OPERATION WITHIN STAGE.
C           NUEXP .. NO. OF SUCCESSFUL EXPLORATORY MOVES.
C           NUPAT .. NO. OF SUCCESSFUL PATTERN MOVES.
C           NOR .. NO. OF TIMES OF PULLING BACK PROCEDURE.
```

```
C           NOP .. NO. OF SUCCESSFUL MOVES INSIDE FEASIBLE REGION.
C           NOITB .. NO. OF SUCCESSFUL MOVES OUT OF FEASIBLE REGION.
C
C     ******************************************************************
C
C     **SEQUENCE OF INPUT DECK ...
C           (1) PROBLEM ID CARD .. ONE CARD, FORMAT 1000 .
C                  PARAMETERS -- NOPM,NAME( COMPOSED BY 3 PARTS ),
C                                N,NG AND MH .
C           (2) PROBLEM ADDITIONAL DATA CARDS .. SPECIFIED IN THE
C                  SUBROUTINE READIN BY USER HIMSELF, ( OPTIONAL ).
C           (3) SUBPROBLEM 1 INITIAL DATA CARDS ..
C                  FIRST  -- ONE CARD, FORMAT 1002 .
C                            PARAMETERS - R,RATIO,ITMAX,INCUT,THETA
C                                         MAXP,ISIZE, AND ICUT.
C                  SECOND -- N CARDS, FORMAT 1004 .
C                            PARAMETERS - J,X(I),D(I),AND DX(I).
C                             *NOTE -- 1. J IS ONLY FOR USER TO
C                                         CHECK THE SEQUENCE OF CARDS.
C                                      2. CARDS SHOULD BE IN ORDER
C                                         ( SEQUENCE OF DIMENSION )
C                                      3. D(I) MAY BE ANY VALUE
C                                      3. D(I) MAY BE USE ANY VALUE
C                                         WHEN ISIZE USE 1 .
C                                      4. DX(I) MAY USE ANY VALUE
C                                         WHEN ISIZE USE 0 .
C           (4) SUBPROBLEM 2 INITIAL DATA CARDS .
C                             .
C                             .
C                             .
C           ( ... UP TO THE LAST SUBPROBLEM INITIAL DATA CARDS ...)
C
C     ******************************************************************
C
C     **SUBROUTINES NEEDED ...
C           BACK     -- USED TO PULL BACK INFEASIBLE POINT .
C           PENAT    -- USED TO COMPUTE PENALTY TERMS .
C           WEIGH    -- USED TO COMPUTE VIOLATION WEIGHT .
C           READIN   -- A USER SUPPLIED SUBROUTINE, USED TO READ IN
C                       ADDITIONAL DATA NEEDED .
C           OBRES    -- A USER SUPPLIED SUBROUTINE, USED TO COMPUTE
C                       THE OBJECTIVE AND CONSTRAINTS .
C           OUTPUT   -- A USER SUPPLIED SUBROUTINE, USED TO OUTPUT
C                       ADDITIONAL INFORMATION DESIRED .
C
C     ******************************************************************
C
C     **DIMENSIONS ...
C       THIS PROGRAM IS DESIGNED FOR N,MH.LE.20 AND MG.LE.50.
C           THE DIMENSIONS ARE ONLY DEFINED IN MAIN PROGRAM, WHEN N,
C  OR MH.GT.20 AND/OR MG.GT.50,MAKE PROPER CHANGES. THE KEY OF
C     CHANGES ..
C           X,FX,PX,BX,DX,D,PD -- N DIMENSIONS
C           G,FG               -- MG DIMENSIONS
C           H,FH               -- MH DIMENSIONS .
C
C     ******************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(20),FX(20),BX(20),PX(20),DX(20),PD(20),D(20),G(50),
```

```
     1FG(50),H(20),FH(20)
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12
      COMMON /BLCGY/ N,MG,MH,ITER,ITMAX,ICHECK,IB,LOST
      COMMON /BLOGP/ NOITP,NOITB,B,D,ISKIP
C    **D(10) ARE NOT NEEDED FOR RUNNING THIS PROGRAM, USER MAY TAKE
C     THEM AWAY.
      COMMON /BLOGR/ D(10)
C    **FG(20) IN BLCGO ARE USED FOR OUTPUT ADDITIONAL DATA CONCERN
C     FG(20) AT SUB-OPTIMUM. USER MAY TAKE THEM AWAY.
      COMMON /BLOGO/ FG
 1000 FORMAT(I5,5X,A2,A2,A2,3I5)
 1001 FORMAT(31X,1H*,A2,A2,A2,10H* PROBLEMS/
     130X,20(1H*)///25X,'NO. OF X(I) ...',I4/25X,
     2'NO. OF G(J) ...',I4/25X,'NO. OF H(K) ...',
     3I4,////,' NO. OF PROBLEMS ...',I4)
 1002 FORMAT(2D15.4,2I5,D15.4,3I5)
 1003 FORMAT(1H1,5X,7HPROBLEM,I4//////)
 1004 FORMAT(I5,3D15.4)
 1005 FORMAT(20X,13HINITIAL POINT/5X,4HY = ,D11.4,7H,  P = ,D11.4,
     17H,  R = ,D11.4,11H,  RATIO = ,D11.4,2H, /5X,4HB = ,D11.4,
     211H,  INCUT = ,I4,  11H,  THETA = ,D11.4,2H .)
 1006 FORMAT(10X,2HX(,I3,4H) = ,D14.6,7H,    D(,I3,4H) = ,D14.6,2H .)
 1007 FORMAT(3X,75(1H*))
 1008 FORMAT(3X,15H**P OPTIMUM.. (,I4,1H)          /5X,5HFY = ,D13.6,6
     1H,FP = ,D13.6,7H,  R = ,D11.4,10H,  ITER = ,I5,1H,/5X,7HNOIT = ,I5
     2,9H,  NOB = ,I4,9H,  NOP = ,I4,10H,  NOBP = ,I4/5X,8HNOEXP = ,I4
     3,11H,  NOPAT = ,I4,11H,  NOCUT = ,I4,2H ./5X,8HYSTOP = ,D13.6,1H.)
 1011 FORMAT(5X/5X,16H**CONSTRAINTS ..)
 1012 FORMAT(10X,2HG(,I3,4H) = ,D14.6,2H ,)
 1013 FORMAT(10X,2HH(,I3,4H) = ,D14.6,2H ,)
 1015 FORMAT(3X,46H*****THE ABOVE RESULTS ARE THE FINAL OPTIMUM .)
 1016 FORMAT(3X,28H**NO. OF P OPTIMUM EXCEEDED ,I5,2H .)
 1020 FORMAT(5X//5X,47H**SELECTED FEASIBLE STARTING POINT ..          )
 1021 FORMAT(   ////////////////////////5X)
 1022 FORMAT(1H 5X,44H**THE PROBLEM MIGHT BE TOO FLAT, CHECK TIMES,I4,
     127H, R AND RATIO BE ADJUSTED, /7X,43HPROBABLY A DOUBLE PRECISION W
     2ILL BE NEEDED.)
C
C    **READ IN PROBLEM NUMBER, PROBLEM NAME, AND DIMENSIONS .
      READ(5,1000) NOPM,NAME1,NAME2,NAME3,N,MG,MH
      WRITE(6,1021)
      WRITE(6,1001) NAME1,NAME2,NAME3,N,MG,MH,NOPM
      IOPM=1
C    **READ IN ADDITIONAL DATA ( USED FOR ALL SUB-PROBLEMS ).
      CALL READIN(N,MG,MH)
C
C    **READ IN INITIAL PARAMETERS AND STOPPING CRITERIA .
    1 READ(5,1002) R,RATIO,ITMAX,INCUT,THETA,MAXP,ISIZE,ICUT
      WRITE(6,1003) IOPM
      MP=1
      MULT=1
      NOEXP=0
      NOPAT=0
      NOCUT=0
      NOR=1
      FNOR=NOR
      NOBP=0
      NOITP=0
      NOITB=0
      ITER=0
```

```
      NOIT=0
      LOST=0
      LLOST=0
      IB=0
      ICHECK=0
      B=0.0D0
      FN=N
C
C    **READ IN INITIAL POINT, INITIAL STEP-SIZES AND ESTIMATED OPTIMUM.
      DO 4 I=1,N
      READ(5,1004) J,X(I),D(I),OX(I)
C    **VARIABLE (J) IS USED FOR CHECKING THE SEQUENCE OF CARDS BY THE
C      USER HIMSELF, AND HAS NO INFERENCE TO THE PROGRAM ( USER MAY
C      USE ANY INTEGER NUMBER FOR (J) ).
      IF(ISIZE) 3,3,2
    2 D(I)=OX(I)*0.02
    3 BX(I)=X(I)
      FX(I)=X(I)
      PD(I)=D(I)
      OX(I)=X(I)
    4 B=B+0.5D0*D(I)
C    **DECIDE THE STARTING VALUE OF TOLERENCE LIMIT FOR G(J) .LT. 0. .
      B=B/FN
      B=2.0D0*B
      PB=B
      CALL OBRES(FX,FY,FG,FH)
      CALL WEIGH(STGH,MG,FG,MH,FH)
      ITER=0
   11 CALL PENAT(FG,FH,PENA1,PENA2)
C    **COMPUTE AN INITIAL VALUE OF R WHEN INPUT R VALUE IS .LE. 0. .
      IF(R) 12,12,13
   12 R=DABS(FY/(PENA1+PENA2))
      R=R/4.0D0
C    **USE RATIO=4.0 WHEN INPUT RATIO VALUE IS .LE. 0. .
   13 IF(RATIO)14,14,15
   14 RATIO=4.0
   15 FP=FY+R*PENA1+R**(-0.5)*PENA2
      WRITE(6,1005) FY,FP,R,RATIO,B,INCUT,THETA
      WRITE(6,1006) (I,FX(I),I,D(I),I=1,N)
      WRITE(6,1007)
      IF(LOST-2) 50,16,16
C    **SELECT AFEASIBLE STARTING POINT WHEN INPUT INITIAL POINT IS
C      NOT FEASIBLE SUBJECT TO INEQUALITY CONSTRAINTS .
C
C    **MAKE EXPLORATORY MOVE FOR SELECTING A FEASIBLE STARTING POINT .
   16 NUF=0
      DO 28 I=1,N
      FX(I)=X(I)+2.0D0*D(I)
      CALL OBRES(FX,FY,FG,FH)
      CALL WEIGH(TGH,MG,FG,MH,FH)
      IF(LOST-2) 44,18,18
   18 IF(STGH-TGH) 20,20,26
   20 FX(I)=FX(I)-4.0D0*D(I)
      CALL OBRES(FX,FY,FG,FH)
      CALL WEIGH(TGH,MG,FG,MH,FH)
      IF(LOST-2) 44,22,22
   22 IF(STGH-TGH) 24,24,26
   24 FX(I)=FX(I)+2.0D0*D(I)
      NUF=NUF+1
      GO TO 28
```

```
   26 STGH=TGH
      X(I)=FX(I)
   28 CONTINUE
C
      IF(NOF-N) 34,30,30
C     **CUT STEP-SIZES FOR SELECTING A FEASIBLE STARTING POINT .
   30 DO 32 I=1,N
   32 D(I)=D(I)*0.500
      GO TO 16
C     **MAKE PATTERN MOVE FOR SELECTING A FEASIBLE STARTING POINT .
   34 DO 36 I=1,N
   36 PX(I)=FX(I)+(FX(I)-BX(I))
      CALL OBRES(PX,PY,PG,PH)
      CALL WEIGH(YGH,MG,PG,MH,PH)
      IF(STGH-YGH) 16,16,40
   40 DO 42 I=1,N
      X(I)=PX(I)
   42 FX(I)=PX(I)
      IF(LOST-2) 44,43,43
   43 STGH=YGH
      GO TO 16
   44 DO 46 I=1,N
      D(I)=PD(I)
      OX(I)=FX(I)
   46 BX(I)=FX(I)
      LOST=0
C     **OUTPUT THE MESSAGE OF THE SELECTED FEASIBLE STARTING POINT .
      WRITE(6,1020)
      GO TO 11
   48 DO 49 I=1,N
   49 X(I)=FX(I)
      LLOST=LOST
C     **START TO MINIMIZE THE CURRENT P-FUNCTION .
C
C     **MAKE EXPLORATORY MOVE FOR MINIMIZING THE P-FUNCTION .
   50 IDIFF=0
      MCUT=1
   51 NOF=0
      GO TO (52,102,52), MCUT
   52 IDIFF=IDIFF+1
      DO 101 I=1,N
      X(I)=FX(I)+D(I)
      LOST=0
      CALL OBRES(X,Y,G,H)
      IF(LOST-1) 62,62,53
   53 IF(Y-FY) 55,55,68
   55 CALL BACK(X,X,Y,G,H)
      NOITB=NOITB+1
      NOBP=NOBP+1
C     **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST=1 MEANS THE
C       RETURNED POINT IS INFEASIBLE )
      IF(LOST-1) 56,150,56
   56 LOST=0
C     **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST .NE. 1 MEANS
C       THE ENTERED POINT IS NEAR-FEASIBLE )
   62 IF(ICHECK-1) 64,140,140
   64 CALL PENAT(G,H,PENA1,PENA2)
      P=Y+R*PENA1+R**(-0.5)*PENA2
      IF(P-FP) 88,68,68
   68 X(I)=FX(I)-D(I)
```

```
      LOST=0                                                            HJS02990
      CALL OBRES(X,Y,G,H)                                               HJS03000
      IF(LOST-1) 80,80,70                                               HJS03010
   70 IF(Y-FY) 73,73,86                                                 HJS03020
   73 CALL BACK(X,X,Y,G,H)                                              HJS03030
      NOITB=NOITB+1                                                     HJS03040
      NOBP=NOBP+1                                                       HJS03050
C   **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST=1 MEANS THE    HJS03060
C     RETURNED POINT IS INFEASIBLE )                                    HJS03070
      IF(LOST-1) 74,150,74                                              HJS03080
   74 LOST=0                                                            HJS03090
C   **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST .NE. 1 MEANS   HJS03100
C     THE ENTERED POINT IS NEAR-FEASIBLE )                              HJS03110
   80 IF(ICHECK-1) 82,140,140                                           HJS03120
   82 CALL PENAT(G,H,PENA1,PENA2)                                       HJS03130
      P=Y+R*PENA1+R**(-0.5)*PENA2                                       HJS03140
      IF(P-FP) 88,86,86                                                 HJS03150
   86 X(I)=FX(I)                                                        HJS03160
      NOF=NOF+1                                                         HJS03170
      GO TO 99                                                          HJS03180
   88 FY=Y                                                              HJS03190
      FP=P                                                              HJS03200
      NOITP=NOITP+1                                                     HJS03210
      FX(I)=X(I)                                                        HJS03220
      LLOST=LOST                                                        HJS03230
      IF(MG) 94,94,90                                                   HJS03240
   90 DO 92 JJ=1,MG                                                     HJS03250
   92 FG(JJ)=G(JJ)                                                      HJS03260
   94 IF(MH) 99,99,96                                                   HJS03270
   96 DO 98 KK=1,MH                                                     HJS03280
   98 FH(KK)=H(KK)                                                      HJS03290
C                                                                       HJS03300
C   **CHECK THE STAGE STOPPING CRITERION IS SATISFIED OR NOT .          HJS03310
   99 IF(NOCUT-INCUT) 100,150,150                                       HJS03320
  100 IF(ICHECK-1) 101,150,101                                          HJS03330
  101 CONTINUE                                                          HJS03340
      IF(NOF-N) 111,104,104                                             HJS03350
  102 DO 103 I=1,N                                                      HJS03360
  103 X(I)=FX(I)+D(I)                                                   HJS03370
      CALL OBRES(X,Y,G,H)                                               HJS03380
      IF(LOST-1) 1107,1107,1104                                         HJS03390
 1104 IF(Y-FY) 1105,1105,1108                                           HJS03400
 1105 CALL BACK(X,X,Y,G,H)                                              HJS03410
      NOITB=NOITB+1                                                     HJS03420
      NOBP=NOBP+1                                                       HJS03430
      IF(LOST-1) 1106,150,1106                                          HJS03440
 1106 LOST=0                                                            HJS03450
      IF(ICHECK-1) 1107,140,140                                         HJS03460
 1107 CALL PENAT(G,H,PENA1,PENA2)                                       HJS03470
      P=Y+R*PENA1+R**(-0.5)*PENA2                                       HJS03480
      IF(P-FP) 1115,1108,1108                                           HJS03490
 1108 DO 1109 I=1,N                                                     HJS03500
 1109 X(I)=FX(I)-D(I)                                                   HJS03510
      CALL OBRES(X,Y,G,H)                                               HJS03520
      IF(LOST-1) 1113,1113,1110                                         HJS03530
 1110 IF(Y-FY) 1111,1111,1114                                           HJS03540
 1111 CALL BACK(X,X,Y,G,H)                                              HJS03550
      NOITB=NOITB+1                                                     HJS03560
      NOBP=NOBP+1                                                       HJS03570
      IF(LOST-1) 1112,150,1112                                          HJS03580
```

```
 1112 LOST=0
      IF(ICHECK-1) 1113,140,140
 1113 CALL PENAT(G,H,PENA1,PENA2)
      P=Y+R*PENA1+R**(-0.5)*PENA2
      IF(P-FP) 1115,1114,1114
 1114 MCUT=3
      GO TO 51
 1115 FP=P
      FY=Y
      MCUT=1
      DO 1116 I=1,N
 1116 FX(I)=X(I)
      IF(MG) 1119,1119,1117
 1117 DO 1118 J=1,MG
 1118 FG(J)=G(J)
 1119 IF(MH) 50,50,1120
 1120 DO 1121 K=1,MH
 1121 FH(K)=H(K)
      GO TO 50
C
C    **CUT STEP-SIZES FOR MINIMIZING THE P-FUNCTION .
  104 DO 105 I=1,N
  105 D(I)=0.5D0*D(I)
      NOCUT=NOCUT+1
      IF(IDIFF-INCUT) 51,106,106
  106 IF(MCUT-1) 107,107,110
  107 MCUT=2
  108 R=R/2.0D0
      CALL PENAT(FG,FH,PENA1,PENA2)
      FP=FY+R*PENA1+R**(-0.5)*PENA2
      INCUT=INCUT+1
      NOCUT=0
      DO 109 I=1,N
      PD(I)=PD(I)*4.0D0
  109 D(I)=PD(I)
      WRITE(6,1022) MCUT
      IF(ISIZE) 2109,2109,51
 2109 DO 2110 I=1,N
 2110 D(I)=D(I)/FNOR
      GO TO 51
  110 IF(NOCUT-INCUT) 1114,150,150
  111 NOEXP=NOEXP+1
      MCUT=3
C
C    **MAKE PATTERN MOVE FOR MINIMIZING THE P-FUNCTION .
      DO 112 I=1,N
      PX(I)=FX(I)+(FX(I)-BX(I))
  112 BX(I)=FX(I)
      LOST=0
      CALL OBJES(PX,Y,G,H)
      IF(LOST-1) 124,124,113
  113 IF(Y-FY) 114,114,51
  114 CALL BACK(PX,X,Y,G,H)
      NOITB=NOITB+1
      NOBP=NOBP+1
C    **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST=1 MEANS THE
C      RETURNED POINT IS INFEASIBLE )
      IF(LOST-1) 115,150,115
C
  115 LOST=0
```

```
C     **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST .NE. 1 MEANS     HJS04160
C       THE ENTERED POINT IS NEAR-FEASIBLE )                                HJS04170
  122 IF(ICHECK-1) 123,140,140                                              HJS04180
  123 IF(ISKIP-1) 124,48,48                                                 HJS04185
  124 CALL PENAT(G,H,PENA1,PENA2)                                           HJS04190
      P=Y+R*PENA1+R**(-0.5)*PENA2                                           HJS04200
      IF(P-FP) 128,48,48                                                    HJS04210
  128 NOPAT=NOPAT+1                                                         HJS04220
      NOITP=NOITP+1                                                         HJS04230
      DO 129 II=1,N                                                         HJS04240
  129 FX(II)=PX(II)                                                         HJS04250
      LLOST=LOST                                                            HJS04260
  130 IF(MG) 133,133,131                                                    HJS04270
  131 DO 132 J=1,MG                                                         HJS04280
  132 FG(J)=G(J)                                                            HJS042 0
  133 IF(MH) 136,136,134                                                    HJS04300
  134 DO 135 K=1,MH                                                         HJS04310
  135 FH(K)=H(K)                                                            HJS04320
  136 FY=Y                                                                  HJS04330
      FP=P                                                                  HJS04340
C                                                                           HJS04350
C     **CHECK THE STAGE STOPPING CRITERION IS SATISFIED OR NOT .            HJS04360
      IF(NOCUT-INCUT) 138,150,150                                           HJS04370
  138 IF(ICHECK-1) 50,150,150                                               HJS04380
C                                                                           HJS04390
C     **CHECK THE ITMAX EXCEEDED POINT( WHEN IT IS RETURNED FROM BACK )     HJS04400
C       IS BETTER OR NOT AND SET PROPER STAGE-OPTIMUM .                     HJS04410
  140 CALL OBRES(X,Y,G,H)                                                   HJS04420
      CALL PENAT(G,H,PENA1,PENA2)                                           HJS04430
      P=Y+R*PENA1+R**(-0.5)*PENA2                                           HJS04440
      IF(P-FP) 142,150,150                                                  HJS04450
  142 DO 144 I=1,N                                                          HJS04460
  144 FX(I)=X(I)                                                            HJS04470
      LLOST=LOST                                                            HJS04480
      GO TO 130                                                             HJS044 0
C     **SET THE SUB-OPTIMUM GOT BEFORE ENTERED TO BACK BE THE               HJS04500
C       STAGE-OPTIMUM .                                                     HJS04510
  150 NOPULL=0                                                              HJS04520
      PULL=0.63D0                                                           HJS04530
      IF(MG) 15,15,151                                                      HJS04540
  151 DO 152 J=1,MG                                                         HJS04550
      IF(FG(J)) 162,162,152                                                 HJS04560
  152 CONTINUE                                                              HJS04570
C                                                                           HJS04580
C     **CHECK THE STAGE OPTIMUM IS FEASIBLE OR NOT .                        HJS04590
  160 IF(LLOST-1) 170,162,162                                               HJS04600
C     **PULL BACK THE INFEASIBLE STAGE-OPTIMUM INTO THE FEASIBLE REGION     HJS04610
  162 DO 163 I=1,N                                                          HJS04620
  163 FX(I)=PULL*(FX(I)-OX(I))+OX(I)                                        HJS04630
      NOPULL=NOPULL+1                                                       HJS04640
      CALL OBRES(FX,FY,FG,FH)                                               HJS04650
      LLOST=LOST                                                            HJS04660
      NOITB=NOITB+1                                                         HJS04670
      IF(NOPULL-5) 160,164,164                                              HJS04680
  164 NOPULL=0                                                              HJS046 Z
  165 DO 166 I=1,N                                                          HJS04700
  166 FX(I)=OX(I)                                                           HJS04710
      CALL OBRES(FX,FY,FG,FH)                                               HJS04720
  170 LOST=0                                                                HJS04730
      CALL PENAT(FG,FH,PENA1,PENA2)                                         HJS04740
```

```
      FP=FY+R*PENA1+R**(-0.5)*PENA2
  203 NOIT=NOIT+ITER
      YSTOP=DABS(FY/(FY-R*PENA1+R**(-0.5)*PENA2))
      YSTOP=DABS(YSTOP-1.0)
      CALL OBRES(FX,FY,FG,FH)
      WRITE(6,1005) NOR,FY,FP,R,ITER,NOIT,NOBP,NOITP,NOITB,NOEXP,
     1NOPAT,NOCUT,YSTOP
      WRITE(6,1006) (I,FX(I),I,D(I),I=1,N)
      WRITE(6,1011)
      IF(MG) 216,216,215
  215 WRITE(6,1012) (J,FG(J),J=1,MG)
  216 IF(MH) 218,218,217
  217 WRITE(6,1013) (K,FH(K),K=1,MH)
C    **OUTPUT ADDITIONAL INFORMATION.
  218 CALL OUTPUT(N,MG,MH)
      WRITE(6,1007)
C
C    **CHECK THE FINAL STOPPING CRITERION IS SATISFIED OR NOT.
      IF(YSTOP-THETA) 230,230,220
C    **CHECK THE MAXP IS EXCEEDED OR NOT .
  220 IF(NOR-MAXP) 221,232,232
C    **STORE LAST SUB-OPTIMUM POINT .
  221 DO 222 I=1,N
      D(I)=PD(I)
  222 OX(I)=FX(I)
C    **SHIFT TO THE NEXT STAGE SEARCH .
      R=R/RATIO
      FP=FY+R*PENA1+R**(-0.5)*PENA2
      NOR=NOR+1
      IF(NOR-5*MP) 224,224,223
  223 INCUT=INCUT+1
      MP=MP+1
  224 IF(NOBP) 226,226,225
  225 INCUT=INCUT+1
  226 NOBP=0
      MULT=1
      NOITB=0
      NOITP=0
      ICHECK=0
      NOEXP=0
      NOPAT=0
      NOCUT=0
      ITER=0
      IR=0
      FNOR=NOR
      B=0.0D0
      MCUT=1
      IDIFF=0
C
C    **DECIDE THE INITIAL STEP-SIZES AND TOLERENCE LIMIT.
      IF(ICUT) 229,229,227
  227 DO 228 I=1,N
      D(I)=PD(I)/FNOR
  228 B=B+0.5D0*D(I)
      B=B/FN
      GO TO 50
  229 B=PB
      GO TO 50
  230 WRITE(6,1015)
      GO TO 234
```

```
  232 WRITE(6,1016) MAXP                                                  HJS05340
  234 IOPM=IOPM+1                                                         HJS05350
      IF(IOPM-NOPM) 1,1,236                                               HJS05360
  236 STOP                                                                HJS05370
      END                                                                 HJS05380
      SUBROUTINE BACK(XB,X,Y,G,H)                                         HJS05390
C                                                                         HJS05400
C        THIS SUBROUTINE PULLS INFEASIBLE POINTS BACK INTO THE            HJS05410
C     FEASIBLE OR NEAR-FEASIBLE REGION .                                  HJS05420
C                                                                         HJS05430
C        **DEFINITION ..                                                  HJS05440
C             FEASIBLE .. ALL G(I) .GE. O. ,                              HJS05450
C             NEAR-FEASIBLE.. (B-TGH) .GE. O. .                           HJS05460
C                                                                         HJS05470
      IMPLICIT REAL*8(A-H,O-Z)                                            HJS05480
      DIMENSION XB(20),X(20),G(50),H(20),D(20)                            HJS05490
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12    HJS05495
      COMMON /BLOGY/ N,MG,MH,ITER,ITMAX,ICHECK,IB,LOST                    HJS05500
      COMMON /BLOGB/ NOITP,NOITB,B,D,ISKIP                                HJS05510
      ITERB=ITER                                                          HJS05520
      ISKIP=0                                                             HJS05525
      FRAC=0.5                                                            HJS05530
      CALL WEIGH(TGH,MG,G,MH,H)                                           HJS05540
      IF(TGH) 8,8,4                                                       HJS05550
C    **DECREASE THE VALUE OF B IN RETURN .                                HJS05560
    4 IF(B-TGH) 12,12,6                                                   HJS05570
    6 IF(0.70D0*B-TGH) 10,8,8                                             HJS05580
    8 B=0.75D0*B                                                          HJS05590
   10 LOST=0                                                              HJS05600
      RETURN                                                              HJS05610
   12 FTGH=TGH                                                            HJS05620
C                                                                         HJS05630
C    **MAKE EXPLORATORY MOVE FOR MINIMIZING TGH .                         HJS05640
   22 NOF=0                                                               HJS05650
      DO 38 NB=1,N                                                        HJS05660
      XB(NB)=XB(NB)-FRAC*D(NB)                                            HJS05670
      CALL OBRES(XB,Y,G,H)                                                HJS05680
      CALL WEIGH(TGH,MG,G,MH,H)                                           HJS05690
      IF(LOST-2) 24,26,26                                                 HJS05700
   24 NOITP=NOITP+1                                                       HJS05710
   25 LOST=0                                                              HJS05720
      GO TO 46                                                            HJS05730
   26 NOITB=NOITB+1                                                       HJS05740
      IF(ICHECK-1) 27,45,45                                               HJS05750
   27 IF(TGH-FTGH) 28,32,32                                               HJS05760
   28 FTGH=TGH                                                            HJS05770
      IF(B-TGH) 38,38,25                                                  HJS05780
C                                                                         HJS05790
   32 XB(NB)=XB(NB)+D(NB)*2.0*FRAC                                        HJS05800
      CALL OBRES(XB,Y,G,H)                                                HJS05810
      CALL WEIGH(TGH,MG,G,MH,H)                                           HJS05820
      IF(LOST-2) 24,34,34                                                 HJS05830
   34 NOITB=NOITB+1                                                       HJS05840
      IF(ICHECK-1) 35,45,45                                               HJS05850
   35 IF(TGH-FTGH) 28,36,36                                               HJS05860
   36 XB(NB)=XB(NB)-FRAC*D(NB)                                            HJS05870
      NOF=NOF+1                                                           HJS05880
   38 CONTINUE                                                            HJS05890
      IF(NOF-N) 22,42,42                                                  HJS05900
C                                                                         HJS05910
```

```
C     **ADD STEP-SIZES FOR MINIMIZING TGH .                             HJS05920
   42 IF(ITER-IT(FR-4-N) 44,43,59                                       HJS05930
   43 FRAC=FRAC*5.0D0                                                   HJS05940
      GO TO 22                                                          HJS05950
   44 FRAC=FRAC*1.5                                                     HJS05960
      GO TO 22                                                          HJS05970
   45 LOST=1                                                            HJS05980
C                                                                       HJS05990
C     **SET BASE POINT TO RETURN .                                      HJS06000
   46 DO 50 NB=1,N                                                      HJS06010
      D(NB)=D(NB)*0.55D0                                                HJS06020
   50 X(NB)=XB(NB)                                                      HJS06030
C     **DECREASE THE VALUE OF B IN RETURN .                             HJS06040
      IF(0.7D0*B-TGH) 60,58,58                                          HJS06050
   58 B=0.75D0*B                                                        HJS06060
      RETURN                                                            HJS06062
   59 LOST=0                                                            HJS06064
      ISKIP=1                                                           HJS06066
   60 RETURN                                                            HJS06070
      END                                                               HJS06080
      SUBROUTINE PENAT(G,H,PENA1,PENA2)                                 HJS06090
C                                                                       HJS06100
C     THIS SUBROUTINE COMPUTES THE PENALTY TERMS FOR SUMT FORMULATION . HJS06110
C        PENA1 FOR INEQUALITY CONSTRAINTS .                             HJS06120
C        PENA2 FOR EQUALITY CONSTRAINTS .                               HJS06130
C                                                                       HJS06140
      IMPLICIT REAL*8(A-H,O-Z)                                          HJS06150
      DIMENSION G(50),H(20)                                             HJS06160
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12  HJS06165
      COMMON /BLOGY/ N,MG,MH,ITER,ITMAX,ICHECK,IB,LOST                  HJS06170
      PENA1=0.D0                                                        HJS06180
      PENA2=0.D0                                                        HJS06190
      IF(MG) 5,5,1                                                      HJS06200
    1 DO 4 I=1,MG                                                       HJS06210
      IF(G(I)) 4,2,4                                                    HJS06220
C     **SET G(I)=0.1D-48  WHEN G(I)=0. ( ON THE BOUNDARY )              HJS06230
    2 G(I)=0.1D-48                                                      HJS06240
    4 PENA1=PENA1+DABS(1.0D0/G(I))                                      HJS06250
    5 IF(MH) 10,10,6                                                    HJS06260
    6 DO 9 K=1,MH                                                       HJS06270
    8 PENA2=PENA2+H(K)**2                                               HJS06280
    9 CONTINUE                                                          HJS06290
   10 RETURN                                                            HJS06300
      END                                                               HJS06310
      SUBROUTINE WEIGH(TGH,MG,G,MH,H)                                   HJS06320
C                                                                       HJS06330
C     THIS SUBROUTINE COMPUTES THE TOTAL WEIGHT OF VIOLATION            HJS06340
C     TO THE INEQUALITY CONSTRAINTS .                                   HJS06350
C                                                                       HJS06360
      IMPLICIT REAL*8(A-H,O-Z)                                          HJS06370
      DIMENSION G(50),H(20)                                             HJS06380
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12  HJS06385
      TGH=0.                                                            HJS06390
      IF(MG) 4,4,1                                                      HJS06400
    1 DO 3 IR=1,MG                                                      HJS06410
      IF(G(IR)) 2,3,3                                                   HJS06420
    2 TGH=TGH+G(IR)**2                                                  HJS06430
    3 CONTINUE                                                          HJS06440
    4 IF(MH) 8,8,5                                                      HJS06450
    5 DO 7 IR=1,MH                                                      HJS06460
```

```
      IF(H(IR)) 6,7,6                                                   HJS06470
    6 TGH=TGH+H(IR)**2                                                  HJS06480
    7 CONTINUE                                                          HJS06490
    8 TGH=TGH**0.500                                                    HJS06500
      RETURN                                                            HJS06510
      END                                                               HJS06520
      SUBROUTINE READIN(N,MG,MH)                                        HJS06530
C     THIS SUBROUTINE IS FOR READ IN ADDITIONAL DATA .                  HJS06540
C     USER SUPPLIES HIS OWN READ STATEMENT AND FORMAT .                 HJS06550
C     ARGUMENTS N,MG,MH ARE NUMBERS OF VARIABLES,OF INEQUALITY CONSTRAINHJS06560
C                     AND OF EQUALITY CONSTRAINTS .                     HJS06570
C     COMMON /BLOGE/ ...... STATEMENT IS FOR TRANSFER DATA USE .        HJS06580
C                                                                       HJS06590
      IMPLICIT REAL*8(A-H,O-Z)                                          HJS06600
      COMMON /BLOGE/ Q(10)                                              HJS06610
      RETURN                                                            HJS06620
      END                                                               HJS06630
      SUBROUTINE OUTPUT(N,MG,MH)                                        HJS06640
C        THIS SUBROUTINE IS FOR USER TO PRINT OUT ADDITIONAL INFORMATIONHJS06650
C     WANTED. ARGUMENTS N,MG,MH ARE NUMBERS OF VARIABLES,OF INEQUALITY  HJS06660
C     CONSTRAINTS,AND OF EQUALITY CONSTRAINTS .                         HJS06670
C        THE NEEDED DATA INFORMATION                                    HJS06680
C        COMMON /BLOGO/..... IS FOR TRANSFER NEEDED DATA IN MAIN TO     HJS06690
C     THE SUBROUTINE OUTPUT .                                           HJS06700
C        USER SUPLLIES ALL NECESSARY FORMATS .                          HJS06710
C                                                                       HJS06720
      IMPLICIT REAL*8(A-H,O-Z)                                          HJS06730
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12  HJS06735
      COMMON /BLOGO/ G(50)                                              HJS06740
      WRITE(6,9020)PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9
 9020 FORMAT(' ','THE COST COMPONENTS OF THE SUBSYSTEMS ARE'/(' ',
     13D15.6))
      WRITE(6,9021) PD10,PD11,PD12
 9021 FORMAT(' ','COST=',3D15.6/)
      RETURN                                                            HJS06750
      END                                                               HJS06760
      SUBROUTINE OBRES(X,Y,G,H)                                         HJS06770
C                                                                       HJS06780
C     THIS SUBROUTINE COMPUTES OBJ. AND CONSTRAINT VALUES .             HJS06790
C     USER SHOULD SUPPLY ALL NECESSARY STATEMENTS IN THE FORM ..        HJS06800
C        Y=......, FUNCTION OF X(I) , FOR OBJECTIVE FUNCTION .          HJS06810
C        G(J)=......, J FROM 1 TO MG , FOR CONSTRAINTS G(J) .GT. 0.0  . HJS06820
C        H(K)=......, K FROM 1 TO MH , FOR CONSTRAINTS H(K) .EQ. 0.0  . HJS06830
C     INSERT THESE STATEMENTS IN THE BLOCK BELOW LINED BY ********** . HJS06840
C                                                                       HJS06850
      IMPLICIT REAL*8(A-H,O-Z)                                          HJS06860
      DIMENSION X(20),G(50),H(20),Q(10)                                 HJS06870
      DIMENSION CC(6,3),COMP(3,3)
      COMMON /BLOGY/ N,MG,MH,ITER,ITMAX,ICHECK,IB,LOST                  HJS06880
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12  HJS06885
      COMMON /BLOGE/ Q                                                  HJS06890
  100 FORMAT(3X,25H**THE ITERATION EXCEEDED ,I5,1H.)           .        HJS06900
C                                                                       HJS06910
C  ************************************************************        HJS06920
C   **NOTE.. STATEMENT NUMBERS 1,2,3,4,5,6,7,8,100 HAVE BEEN USED.      HJS06930
      CC(1,1)=.6
      CC(1,2)=.5
      CC(1,3)=.8
      CC(2,1)=400.
      CC(2,2)=500.
```

```
      CC(2,3)=600.
      CC(3,1)=5.
      CC(3,2)=5.
      CC(3,3)=5.
      CC(4,1)=1.8
      CC(4,2)=2.
      CC(4,3)=1.7
      CC(5,1)=20.
      CC(5,2)=15.
      CC(5,3)=50.
      CC(6,1)=3.
      CC(6,2)=4.
      CC(6,3)=2.
      ST=1500.
      Y1=0.
      Y2=0.
      Y3=0.
      VAVO=1.
      DO 50 I=1,3
      IM=I+3
      IC=IM+3
      IE=IC+3
      IG=IE+3
      IH=IG+3
      IJ=IH+3
      IK=IJ+3
      UNREL=(1.-DEXP(-X(I)*X(IE)))**2
      REL=1.-UNREL
      CALL INTEG(X,I,IE,RMTBM)
      UMTBM=RMTBM/UNREL
      SMTBM=RMTBM/REL
      CTM=1./X(IM)
      PTM=X(IC)
      RTM=CTM*UNREL+PTM*REL
      COMP(1,I)=CC(1,I)*RMTBM+CC(2,I)/RTM-CC(3,I)
      COMP(2,I)=(ST/UMTBM)*(CC(4,I)*CTM)**2
      COMP(3,I)=(ST/SMTBM)*(CC(5,I)*PTM-CC(6,I))
      AV=RMTBM/(RMTBM+RTM)
      Y1=Y1+COMP(1,I)
      Y2=Y2+COMP(2,I)
      Y3=Y3+COMP(3,I)
      VAVO=VAVO*AV
      G(I)=X(I)-.001
      G(IM)=.02-X(I)
      G(IC)=X(IM)-.02
      G(IE)=.6667-X(IM)
      G(IG)=X(IC)-.5
      G(IH)=25.-X(IC)
      G(IJ)=X(IE)-100.
      G(IK)=800.-X(IE)
   50 CONTINUE
      G(25)=VAVO-.97
      G(26)=1.-VAVO
      Y=Y1+Y2+Y3
      PD1=COMP(1,1)
      PD2=COMP(2,1)
      PD3=COMP(3,1)
      PD4=COMP(1,2)
      PD5=COMP(2,2)
      PD6=COMP(3,2)
```

```
      PD7=COMP(1,3)
      PD8=COMP(2,3)
      PD9=COMP(3,3)
      PD10=Y1
      PD11=Y2
      PD12=Y3
C                                                                       HJS06940
C  ***********************************************************************HJS06950
      LOST=0                                                            HJS06960
      ITER=ITER+1                                                       HJS06970
      IF(ITER-ITMAX) 3,1,2                                              HJS06980
C    **OUTPUT THE MESSAGE OF ITMAX EXCEEDED.                            HJS06990
    1 WRITE(6,100) ITMAX                                                HJS07000
    2 ICHECK=1                                                          HJS07010
C    **CHECK FOR THE VIOLATION TO INEQUALITY CONSTRAINTS.               HJS07020
    3 IB=0                                                              HJS07030
      IF(MG) 8,8,4                                                      HJS07040
    4 DO 7 I=1,MG                                                       HJS07050
      IF(G(I)) 5,6,7                                                    HJS07060
    5 LOST=2                                                            HJS07070
      GO TO 7                                                           HJS07080
    6 IB=1                                                              HJS07090
    7 CONTINUE                                                          HJS07100
    8 RETURN                                                            HJS07110
      END                                                               HJS07120
      SUBROUTINE INTEG(XA,J,JE,FSUB)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XA(20)
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12
      ZERO=0.
      RI=.5
      DINTVL=(XA(JE)-ZERO)/100.
   10 ZERO=ZERO+DINTVL
      RM=1.-(1.-DEXP(-XA(J)*ZERO))**2
      RI=RI+RM
      IF(ZERO.LT.(XA(JE)-DINTVL)) GO TO 10
      RF=1.-(1.-DEXP(-XA(J)*XA(JE)))**2
      FSUB=DINTVL*(RI+RF/2.)
      RETURN
      END
```

A2.4 SUMT : USER SUPPLIED SUBROUTINES FOR EXAMPLE 2

The following listed subroutines may be inserted in place of corresponding subroutines listed in Appendix A2.3.

```
      SUBROUTINE OBJES(X,Y,G,H)
C
C     THIS SUBROUTINE COMPUTES OBJ. AND CONSTRAINT VALUES .
C     USER SHOULD SUPPLY ALL NECESSARY STATEMENTS IN THE FORM ..
C        Y=....., FUNCTION OF X(I) , FOR OBJECTIVE FUNCTION .
C        G(J)=....., J FROM 1 TO MG , FOR CONSTRAINTS G(J) .GT. 0.0  .
C        H(K)=....., K FROM 1 TO MH , FOR CONSTRAINTS H(K) .EQ. 0.0  .
C     INSERT THESE STATEMENTS IN THE BLOCK BELOW LINED BY *********** .
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(20),G(50),H(20),Q(10)
      DIMENSION CC(6,3),COMP(3,3)
      COMMON /BLOGY/ N,MG,MH,ITER,ITMAX,ICHECK,IB,LOST
      COMMON /CHAY/ PD1,PD2,PD3,PD4,PD5,PD6,PD7,PD8,PD9,PD10,PD11,PD12
      COMMON /BLOGR/ Q
  100 FORMAT(5X,25H**THE ITERATION EXCEEDED ,I5,1H.)
C
C  ****************************************************************
C   **NOTE.. STATEMENT NUMBERS 1,2,3,4,5,6,7,8,100 HAVE BEEN USED.
      CC(1,1)=1.8
      CC(1,2)=1.5
      CC(1,3)=2.0
      CC(2,1)=200.
      CC(2,2)=170.
      CC(2,3)=250.
      CC(3,1)=5.
      CC(3,2)=5.
      CC(3,3)=5.
      CC(4,1)=2.
      CC(4,2)=2.5
      CC(4,3)=3.
      CC(5,1)=40.
      CC(5,2)=100.
      CC(5,3)=5 .
      CC(6,1)=3.
      CC(6,2)=4.
      CC(6,3)=2.
      ST=1500.
      Y1=0.
      Y2=0.
      Y3=0.
      VAVG=1.
      DO 50 I=1,3
      IM=I+3
      IC=IM+3
      IL=IC+3
      IG=IL+3
      IH=IG+3
      IJ=IH+3
      IK=IJ+3
      UNREL=(1.-DEXP(-X(I)*X(IT)**2))**2
      REL=1.-UNREL
      CALL INT S(X,1,IM,RNT(M))
      UNTOM=RNTOM/UNREL
      SMTOM=RNTOM/REL
      CTM=X(IM)
```

```
      PTM=X(IC)
      RTM=CTM*INRIL+RTM*RIL
      COMP(1,I)=CC(1,I)*RNTEM+CC(2,I)/RTM-CC(3,I)
      COMP(2,I)=(ST/RMTEM)*(CC(4,I)*CTM)**2
      COMP(3,I)=(ST/SMTEM)*(CC(5,I)*RTM-CC(6,I))
      AV=RMTEM/(SMTEM+RTM)
      Y1=Y1+COMP(1,I)
      Y2=Y2+COMP(2,I)
      Y3=Y3+COMP(3,I)
      VAVO=VAVO*AV
      G(I)=X(I)-.0001
      G(IM)=.0007-X(I)
      G(IL)=X(IM)-.5
      G(IN)=10.-X(IM)
      G(IG)=X(IC)-.1
      G(IH)=10.-X(IC)
      G(IJ)=X(IL)-50.
      G(IK)=150.-X(IL)
   50 CONTINUE
      G(25)=VAVO-.93
      G(26)=1.-VAVO
      Y=Y1+Y2+Y3
      PD1=COMP(1,1)
      PD2=COMP(2,1)
      PD3=COMP(3,1)
      PD4=COMP(1,2)
      PD5=COMP(2,2)
      PD6=COMP(3,2)
      PD7=COMP(1,3)
      PD8=COMP(2,3)
      PD9=COMP(3,3)
      PD10=Y1
      PD11=Y2
      PD12=Y3
C                                                                       HJS06940
C     ****************************************************************  HJS06950
      LOST=0                                                            HJS06960
      ITER=ITER+1                                                       HJS06970
      IF(ITER-ITMAX) 3,1,2                                              HJS06980
C     **OUTPUT THE MESSAGE OF ITMAX EXCEEDED.                           HJS06990
    1 WRITE(6,100) ITMAX                                                HJS07000
    2 ICHECK=1                                                          HJS07010
C     **CHECK FOR THE VIOLATION TO INEQUALITY CONSTRAINTS.              HJS07020
    3 IB=0                                                              HJS07030
      IF(MG) 8,8,4                                                      HJS07040
    4 DO 7 I=1,MG                                                       HJS07050
      IF(G(I)) 5,6,7                                                    HJS07060
    5 LOST=2                                                            HJS07070
      GO TO 7                                                           HJS07080
    6 IB=1                                                              HJS07090
    7 CONTINUE                                                          HJS07100
    8 RETURN                                                            HJS07110
      END                                                               HJS07120
```

```
      SUBROUTINE INTEG(XA,J,JL,FSUM)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XA(20)
      COMMON /CHAY/ PP1,P02,PP3,P04,P05,P06,P07,P08,P09,P10,P11,P12
      ZERO= .
      RI=.5
      DINTVL=(XA(JL)-ZERO)/100.
   10 ZERO=ZERO+DINTVL
      RM=1.-(1.-DEXP(-XA(J)*ZERO**2))**2
      RI=RI+RM
      IF(ZERO.LT.(XA(JL)-DINTVL)) GO TO 10
      RF=1.-(1.-DEXP(-XA(J)*XA(JL)**2))**2
      FSUM=DINTVL*(RI+RF/2.)
      RETURN
      END
```

OPTIMAL AVAILABILITY ALLOCATION IN
SERIES-PARALLEL MAINTAINED SYSTEMS

by

Chang Hoon Lie

B.S. (Nuclear Engineering), Seoul National University

Seoul, Korea, 1970

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

Kansas State University

Manhattan, Kansas

1974

In designing maintained systems, availability is used as a single measure for the system effectiveness. The series-parallel system which has subsystems in series, where each subsystem has identical units in parallel, is considered. Considering both corrective and preventive maintenance, availability models for the series-parallel systems are developed under the assumption of various probability density functions for failure and repair times of each unit. The cost of the system consists of three cost components : the cost for designing mean time between maintenance and mean corrective and preventive maintenance time, the cost for corrective maintenance, and the cost for preventive maintenance.

The optimal availability allocation problem, then, is to determine individual units' detailed availability specification that will allow a system availability requirement to be met with a minimum cost for the system. Both the generalized reduced gradient (GRG) method and sequential unconstrained minimization technique (SUMT) are employed to solve this problem. The results obtained from these two different optimization methods are compared. This availability allocation technique is applicable in the early stages of maintained system design as well as in the latter stages of system design when modifications and improvements for the initial specifications are required.