OPTIMIZATION OF INDUSTRIAL MANAGEMENT SYSTEMS

BY THE SEQUENTIAL UNCONSTRAINED
MINIMIZATION TECHNIQUE

by

Keeting Chiecheng Lai

B.S. Chong-Uan College of Science and Technology
Taw-Uan, Formosa, 1965

A Master's Report

Submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering
Kansas State University
Manhattan, Kansas

1970

Approved by

[Signature]
Major Professor
TABLE OF CONTENTS

1. INTRODUCTION 1

2. OPTIMAL RELIABILITY OF A COMPLEX SYSTEM 7
   2.1 INTRODUCTION 7
   2.2 SYSTEM RELIABILITY USING CONDITIONAL PROBABILITIES 8
   2.3 FORMULATION OF AN OPTIMIZATION PROBLEM 11
   2.4 SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE (SUMT) 12
   2.5 A NUMERICAL EXAMPLE 17
   2.6 DISCUSSION 20

3. IMPLEMENTATION OF SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE
   BY HOOKE AND JEEVES PATTERN SEARCH AND HEURISTIC PROGRAMMING 30
   3.1 INTRODUCTION 30
   3.2 SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE (SUMT) 31
   3.3 COMPUTATIONAL PROCEDURE 33
   3.4 PROCEDURE FOR SELECTING A FEASIBLE STARTING POINT FROM THE
      INPUT INFEASIBLE INITIAL POINT 36
   3.5 COMPUTATIONAL PROCEDURE FOR MINIMIZING P(x, r_k) FUNCTION BY
      HOOKE AND JEEVE PATTERN SEARCH 39
   3.6 PROCEDURE FOR MOVING AN INFEASIBLE POINT INTO THE FEASIBLE
      OR NEAR-FEASIBLE REGION BOUNDED BY INEQUALITY CONSTRAINTS 42
   3.7 PROCEDURE FOR MOVING THE NEAR-FEASIBLE k-TH SUB-OPTIMUM INTO
      FEASIBLE REGION 45

4. SUMT IMPLEMENTED BY HOOKE AND JEEVES SEARCH TECHNIQUE APPLIED TO
   PRODUCTION SCHEDULING PROBLEMS 49
   4.1 INTRODUCTION 49
   4.2 A PRODUCTION SCHEDULING AND INVENTORY CONTROL PROBLEM 49
   4.3 A PERSONNEL AND PRODUCTION SCHEDULING PROBLEM 63
   4.4 DISCUSSION AND CONCLUDING REMARKS 68

5. OPTIMIZATION OF A COMPLEX SYSTEM RELIABILITY 72
   5.1 INTRODUCTION 72
5.2 FORMULATIONS OF TWO SYSTEM RELIABILITY PROBLEMS 73
5.3 THE PROBLEM OF MAXIMIZING SYSTEM RELIABILITY 75
5.4 THE COST FUNCTION MINIMIZATION PROBLEM 80
5.5 CONCLUDING REMARKS 86

ACKNOWLEDGEMENT 89
APPENDIX 90
CHAPTER 1

INTRODUCTION

The problems considered in this report are optimization of system reliability of a complex system and optimization of production scheduling and inventory control subject to some linear and/or nonlinear constraints. The optimization method employed is the sequential unconstrained minimization technique (SUMT). This method is considered as one of the simplest and the most efficient methods for solving the constrained nonlinear programming problems.

The purposes of this report are twofold. The first is to present a result of implementing SUMT by a combination of the Hooke and Jeeves pattern search technique [13,14] and a heuristic programming technique [19]. The second is to present results of the optimization study of system reliability of a complex system and production scheduling and inventory control problems by means of the developed technique.

The principle of the sequential unconstrained minimization technique (SUMT) is a transformation of a constrained minimization problem into a sequence of unconstrained minimization problem. This transformation enables us to use well developed unconstrained optimization techniques to solve the constrained problem without inventing a new technique for such a constrained optimization problem. The method was first proposed by Carroll in 1959 [4,5] and further developed by Fiacco and McCormick [8,9,10,11,17]. In 1964, Fiacco and McCormick developed a general algorithm based on SUMT, and in 1965, they proposed a method which is called SUMT without parameters. By using this method, the difficulty of choosing the penalty parameters
can be avoided, although there are still some difficulties exist. There is a general computer program provided by McCormick, Mylander and Fisco called "RAC Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming," (IBM SHARE number 3189) [17]. In this computer program, the unconstrained minimization technique used is the second order gradient method.

Difficulties which arise from use of the second order gradient method as a unconstrained minimization technique in SUMT becomes predominate in a large size and/or very complex nonlinear problem. The difficulties arise particularly in taking correctly the first order and second order partial derivatives of very complex nonlinear functions which most of practical problems have. Therefore, a new algorithm which using a much simpler direct search technique is very desirable.

For the above reason, a new technique of implementing SUMT by Hooke and Jeeves pattern search technique to be its unconstrained minimization process is suggested [6] and is developed. The procedures are presented in Chapter 3 in details. Hooke and Jeeves pattern search technique [13,14] is different from the gradient method by the decision making process to decide the direction of search. The direction of search in the gradient method is in the steepest decent direction while that of the Hooke and Jeeves pattern search technique is determined by direct comparison of the values of the objective function at two points depart from each other for a finite step. For this reason, when the pattern search is getting close to the boundary of some inequality constraints, it shall frequently go out of the feasible region bounded by inequality constraints, and the search might be terminated at some point near the boundary which might not be the
real constrained optimum. A heuristic programming technique was developed by Paviani and Himmelblau [19], which provides a method for applying a sequential simplex pattern search routine [2,3,6a,18] to a constrained problem. The method enables to make turns at the pattern search near the boundary of constraints. This heuristic idea is employed here in order to handle the boundary of inequality constraints [6]. The details of the method are described in Chapter 3 and a general FORTRAN-IV program together with detailed computer diagrams is presented in Appendix.

This newly developed method is utilized to obtain optimum solutions of two examples of production scheduling and inventory control in chapter 4. The first problem is a simple two dimensional problem used for demonstrating the procedure of the algorithm in details and the second problem is a 20-dimensional problem used for demonstrating the capacity and practicability of the technique. Both problems have previously been solved by using the RAC program introduced before [15].

Much has been written about the optimization of the reliability of a system. Usually the increase in the system reliability is due to adding redundancies. Previously, with redundant components in parallel or in series were considered [7, 23, 24, 25, 26]. The problem becomes considerably more difficult when the redundant units of the system cannot be reduced to parallel or series configurations.

In attempting to optimize the reliability of such a complex system a major difficulty is encountered in that the reliability expression is not a separable function and thus cannot be analyzed as a multistage process. Thus another approach is used to solve this type of problem where the reliability is obtained by Bayes' theorem which utilizes conditional
probabilities [1]. With this in mind a mathematical model for the nonlinear system reliability subject to constraints is formulated. The nonlinear programming problem of optimizing the system reliability is then solved by SUMT using RAC computer program [17] in Chapter 2.

The same reliability problem is also solved by the newly developed technique and the results are presented in Chapter 5. Far less preparatory work is required and the partial derivatives of objective function and functions of inequality and equality constraints are not needed. By comparing the results with that obtained in Chapter 2, we can conclude that the newly developed technique is workable and much simpler than the original technique mentioned is. Thus the new technique is capable of solving a wide range of practical optimization problems.
REFERENCES


Search Solution to Optimal Production Planning Problems" Report
No. 18, Institute for Systems Design and Optimization, Kansas
State University, 1969.

15. Hsu, F. T., L. T. Fan and C. L. Hwang, "Sequential Unconstrained
Minimization Technique (SUMT) for Optimal Production Planning",
paper submitted for publication, 1969.


Computer Program Implementing the Sequential Unconstrained Minimization
Technique for Nonlinear Programming," SHARE Number 3189.


19. Paviani, D. A., and D. M. Himmelblau, "Constrained Nonlinear Optimi-
ization by Heuristic Programming", Paper Presented in AIChE meeting,

Function of Several Variables without Calculating Derivatives,"

Problems," T.P. 310, Atomic Energy Research Establishment, Harwell,


with Several Modes of Failures by Integer Programming," IEEE Trans.

Reliability Subject to Multiple Nonlinear Constraints," IEEE Trans.

Reliability of a Complex System," paper submitted for publication,
Institute for Systems Design and Optimization, Kansas State University,
1969.

of Constrained Reliability Problems," Management Science, Vol. 13,
CHAPTER 2

OPTIMAL RELIABILITY OF A COMPLEX SYSTEM

2.1 INTRODUCTION

Much has been written about the optimization of the reliability of a system. Usually these problems are concerned with optimizing some objective function subject to constraints where the increase in the system reliability is due to adding redundancies. In previous work, the systems treated usually have redundant components in parallel or in series (4, 14, 15, 16). The problem becomes considerably more difficult when the redundant units of the system cannot be reduced to parallel and series configurations. One such example is shown in Fig. 1. In the system, unit 1 is backed up by a parallel unit 4. There are two equal paths, where each path has unit 2 in series with the stage formed by unit 1 and unit 4. These two equal paths operate in parallel so that if at least one of them is good the output is assured. However, because unit 2 does not have a high degree of reliability, a third unit, unit 3, is inserted into the circuit. Therefore, the following operations are possible: 2-1, 2-4, 3-1, and 3-4, and each operation has two equal paths.

In attempting to optimize the reliability of such a configuration a major difficulty is encountered in that the reliability expression is not a separable function \(^\dagger\) and thus cannot be analyzed as a multistage process. Thus another approach is used to solve this type of problem where the

\(^\dagger\) A function is separable if \(f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} f(x_i)\).
reliability is obtained by Bayes' theorem which utilizes conditional probabilities (1). With this in mind a mathematical model for the nonlinear system reliability subject to constraints is formulated. The nonlinear programming problem of optimizing the system reliability is then solved by the sequential unconstrained minimization technique (SUMT) (5, 6, 7, 8). This method appears to be one of the more efficient methods of solving constrained nonlinear optimization problems.

2.2 SYSTEM RELIABILITY USING CONDITIONAL PROBABILITIES

In a complex system where the redundant units cannot be reduced to a parallel or series configuration the reliability is obtained by using Bayes' Theorem involving conditional probabilities, Razovsky [1].

In solving this problem, a simplified form of Bayes' probability theorem is used. The theorem says that if A is an event which depends on one of two mutually exclusive events B_1 and B_j of which one must necessarily occur, then the probability of the occurrence of A is given by

\[ P(A) = P(A, \text{ given } B_1) \cdot P(B_1) + P(A, \text{ given } B_j) \cdot P(B_j) \]  

(1)

To put this theorem in the context of a reliability problem, let us denote the event of a system's failure by A and the survival by B_1 and the failure by B_j of a component or unit on whose operation the system reliability depends. The probability of system failure \( P(A) \), then, equals the probability of system failure given that a specified component in the system is good, \( P(A, \text{ given } B_1) \), times the probability that the component is good, \( P(B_1) \), plus the probability of system failure given that the component is bad, \( P(A, \text{ given } B_j) \), times the probability that the component is bad, \( P(B_j) \). Thus if K is a component upon whose state, whether good
or bad, the system reliability depends, we say that the probability of system failure, \( P \) (system failure), is equal to

\[
P(\text{System failure given component } K \text{ is good}) \cdot P(K \text{ is good}) + P(\text{System failure given component } K \text{ is bad}) \cdot P(K \text{ is bad}).
\]  

(2)

Let \( Q_s \) represent the probability of system failure, \( R_k \) the probability that component \( K \) is good, and \( Q_k \) the probability that component \( K \) is bad, then we obtain the usual expression for system unreliability

\[
Q_s = Q_s(\text{given } K \text{ is good}) \cdot R_k + Q_s(\text{given } K \text{ is bad}) \cdot Q_k.
\]  

(3)

The system reliability, \( R_s \), is then

\[
R_s = 1 - Q_s
\]  

(4)

Equation (3) now enables us to calculate the reliability of complex systems. To illustrate we will obtain the reliability of the system presented in Fig. 1. Component 3 for \( K \) is selected for the key component in equation (3), thus we have the expression for system unreliability

\[
Q_s = Q_s(\text{if } 3 \text{ is good}) \cdot R_3 + Q_s(\text{if } 3 \text{ is bad}) \cdot Q_3.
\]  

(5)

If component 3 is good the system can fail if the two paths, which contain unit 2 in series with the stage formed by units 1 and 4 in parallel, fail. With these two paths in parallel, the system's unreliability, given unit 3 is good, is

\[
Q_s(\text{if } 3 \text{ is good}) = [(1-R_1)(1-R_4)]^2.
\]  

(6)

If on the other hand unit 3 is bad the system will fail only if both parallel paths fail, and the system's unreliability, if 3 is bad, is

\[
Q_s(\text{if } 3 \text{ is bad}) = \left[1 - R_2[1 - R_2(1 - (1-R_1)(1-R_4))]\right]^2
\]  

(7)
Fig.1. A schematic diagram of a complex system.
where \( (1 - R_2[1 - (1-R_1)(1-R_4)]) \) is the unreliability of the path which has unit 2 in series with the stage formed by units 1 and 4.

Using equation (5) the unreliability of the system is

\[
Q_s = [(1-R_1)(1-R_4)]^2 \cdot R_3 + [1 - R_2[1 - (1-R_1)(1-R_4)]]^2 \cdot (1-R_3).
\]

The system reliability is given by equation (4).

2.3 FORMULATION OF AN OPTIMIZATION PROBLEM

The problem of maximizing the reliability of the complex system given in Fig. 1 which is subject to a single constraint can be stated as follows:

Maximize

\[
R_s = 1 - Q_s
\]

\[
= 1 - R_3[(1-R_1)(1-R_4)]^2
- (1-R_3)[1 - R_2[1 - (1-R_1)(1-R_4)]]^2
\]

subject to

\[
\sum_i C_i \leq C
\]

where

\[
C_i = K_i R_i^{a_i}
\]

The system reliability, \( R_s \), given by equation (9) can be obtained from equations (4) and (8). The constraint given by equation (10) can be interpreted as follows: \( C_i \) can represent the weight, the cost, or the volume of each unit or component of the system, and the summation of the weight, the cost, or the volume of the system must be less than \( C \). The
weight, cost, or volume of each unit or component of the system is a function of reliability which can be expressed by equation (11), where \( K_i \) is a proportionality constant and \( a_i \), the exponential factor, relates \( C_i \) and the reliability. Usually \( a_i \) is less than one.

The solution of the above constrained nonlinear programming problem can be obtained by the technique which is described in the following section.

2.4 SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE (SUMT)

The general nonlinear programming problem with nonlinear inequality constraints is one where \( x \) is selected to

\[
\text{minimize } f(x) \\
\text{subject to } g_i(x) \geq 0, \quad i = 1, 2, \ldots, m \tag{12}
\]

where \( x \) is an \( n \)-dimensional column vector \( (x_1, x_2, \ldots, x_n)^T \). The superscript \( T \) denotes transposition. If the variables are required to be non-negative, such constraints are included in the \( g_i \)'s. The functions, \( f(x) \) and \( g_i(x), i = 1, 2, \ldots, m \), can take a linear or nonlinear form.

The following algorithm is presented [5, 6, 7, 8] to solve this problem. First define the function (called the \( P \) function)

\[
P(x, r_k) = f(x) + r_k \sum_{i=1}^{m} \frac{1}{g_i(x)} \tag{13}
\]

where \( r_k \) is a positive constant. The subscript \( k \) indicates the number of times the \( P \) function has been solved. The conditions imposed on the \( P \) function are as follows:

1. \( r_k, k = 1, 2, \ldots, \) is a positive real number and \( r_1 > r_2 > \ldots > r_k > \ldots > 0 \). This indicates that \( \{r_k\} \) is a strictly monotonic
decreasing sequence and \( r_k \to 0 \) as \( k \to \infty \).

(2) \( R^0 = \{x \mid g_i(x) > 0, \ i = 1, 2, \ldots, m\} \) is non-empty. This condition indicates that at least one point must exist within the interior of the feasible region.

(3) The functions \( f(x), g_1(x), \ldots, g_m(x) \) are twice continuously differentiable.

(4) The function \( f(x) \) is convex.

(5) The functions \( g_1(x), \ldots, g_m(x) \) are concave.

(6) For every finite \( N, \{x \mid f(x) \leq M; x \in R\} \) is a bounded set, where \( R = \{x \mid g_i(x) \geq 0, i = 1, 2, \ldots, m\} \).

(7) The function \( P(x, r_k) = f(x) + r_k \sum_{i=1}^{\frac{1}{\ell}} \frac{1}{g_i(x)} \) is, for each \( r > 0 \), strictly convex for \( x \in R^0 \). This also indicates that either \( f(x) \) is strictly convex or one of the functions \( g_1, \ldots, g_m \) is strictly concave.

Practical experience indicates that the problems given by equation (1) can be solved even when these conditions are not met. The three conditions which are absolutely required to obtain any useful results are conditions (1), (2), and (6). Condition (1) guarantees that the sequential minimization of the \( P \) function will eventually lead to the solution of minimization of function \( f(x) \). Condition (2) eliminates problems with equality constraints. Condition (6) eliminates problems having local minimum at infinite points.

The characteristics of the \( P \) function are as follows:

(1) \( \lim_{k \to \infty} r_k \sum_{i=1}^{m} \frac{1}{g_i(x)} = 0 \),

(2) \( \lim_{k \to \infty} f[x(r_k)] = u^* \),
(3) \( \lim_{k \to \infty} P(x(r_k), r_k^*) = u^* \),

(4) \( \{ f(x(r_k)) \} \) is a monotonically decreasing sequence,

(5) \( \{ \sum_{i=1}^{m} \frac{1}{g_i(x)} \} \) is a monotonically increasing sequence.

The proofs of these characteristics are presented in detail by Fiacco and McCormick [5, 6, 7, 8].

Intuitive Concept of P Function

The term \( r_k \sum_{i=1}^{m} \frac{1}{g_i(x)} \) in the P function of equation (13) can be considered as a penalty factor attached to the objective function \( f(x) \). By adding the penalty term, the minimization of the P function will assure a minimum to be in the interior of the inequality constrained region by avoiding crossing the boundaries of the feasible region.

Since the feasible boundary is defined by one or more of the \( g_i(x) = 0 \), \( i = 1, \ldots, m \), the value of \( r_k \sum_{i=1}^{m} \frac{1}{g_i(x)} \) will approach infinity as the value of \( x \) approaches one of the boundary lines. Hence the value of \( x \) will tend to remain inside the inequality-constrained region.

The motivation behind this formulation of the P function is the transformation of the original constrained problem into a sequence of unconstrained minimization problems. The desirability of this transformation lies in the fact that numerous methods for minimizing an unconstrained function are known and newer methods are continually being developed [2, 3, 9, 10, 11, 13].
Computational Procedure

The procedure for using SUMT is summarized below [5, 6].

(1) Select the initial value of $r_0$ arbitrarily or use the formula for the selection $r_0$, which is available in reference [6].

(2) Select a feasible starting point $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)$. If the feasible point cannot be easily obtained, select $x^0$ arbitrarily. The computer program [12] will minimize the following $P$ function and obtain a feasible point.

$$P(x, r_k) = -g_s(x) + r_k \sum_{t \in T} \frac{1}{g_t(x)}$$

where $g_s(x^0) \leq 0$ and $T = \{t \mid g_t(x^0) > 0\}$. Note that the constraint function $g_s(x) \geq 0$ is violated.

(3) Minimize the $P$ function for the current value of $r_k$ by using the second-order optimum gradient method.

(4) Check to see if the stopping criterion such as

$$\frac{f[x(r_k)]}{G[x(r_k)]} - 1 < \varepsilon$$

is satisfied. If it is satisfied the solution is optimal; otherwise go to step 5. The dual function, $G[x(r_k)]$, is defined as [5]

$$G[x(r_k)] = f[x(r_k)] - r_k \sum_{i=1}^{m} \frac{1}{g_i[x(r_k)]}$$

(5) Set $k = k+1$ and $r_{k+1} = r_k/C$, where $C > 1$. Repeat the iteration from step 3.

The procedures described above must satisfy two stopping criteria before any meaningful optimal solution can be obtained. The stopping
criterion used for terminating the minimization of the P function [Step 3] may be one of the following

\[ \left| \nabla_p \nabla^T (x, r) \right| \left| \frac{\partial^2 P(x, r)}{\partial x_i \partial x_j} \right|^{-1} \nabla^T P(x, r) < \varepsilon' \quad (16a) \]

or

\[ \left| \nabla_p \nabla^T (x, r) \right| \left| \frac{\partial^2 P(x, r)}{\partial x_i \partial x_j} \right|^{-1} \nabla^T P(x, r) < \frac{P(x, r_{k-1}) - P(x, r_k)}{5} \quad (16b) \]

or

\[ \left| \nabla_p P(x, r) \right| < \varepsilon' \quad (16c) \]

The first stopping criterion was used throughout this study with \( \varepsilon' \) in the range of \( 10^{-3} \) to \( 10^{-5} \). The stopping criterion for terminating the overall minimization of \( f(x(r_k)) \) may take the following form in addition to the form given by equation (14).

\[ r_k \sum_{i=1}^{m} \frac{1}{g_i(x(r_k))} < \varepsilon \quad (17) \]

The first form, equation (14), was used in the numerical examples presented in this work with \( \varepsilon \) generally ranging from \( 10^{-3} \) to \( 10^{-5} \). The procedure should not be terminated until both criteria given by equations (14) and (16) are satisfied. If these stopping criteria are not satisfied within a specified time limit, the iterations should be terminated.

We used a computer program entitled "RAC Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming" which is available for solving the example problems. Its
SHARE number is 3189 [12]. The program is written in FORTRAN IV and can be used on IBM 360. With minor modifications the program can be run on any sufficiently large computer with a FORTRAN compiler.

2.5 A NUMERICAL EXAMPLE

The nonlinear programming problem formulated in the preceding section is restated again and the objective is to maximize

\[ R_s = 1 - R_3 [(1 - R_1)(1 - R_4)]^2 \]

\[ - (1 - R_3) \left( 1 - R_2 \left[ 1 - (1 - R_1)(1 - R_4) \right] \right)^2 \]  \hspace{1cm} (18)

subject to the constraint

\[ 2K_1 R_1^a + 2K_2 R_2^a + K_3 R_3^a + 2K_4 R_4^a \leq C. \]  \hspace{1cm} (19)

The constants \( K_1, K_2, K_3, \) and \( K_4, \) the constraint, \( C, \) and the exponential constant \( a_i, \) \( i = 1, 2, 3, 4, \) are as follows:

\[ K_1 = 100, \quad K_2 = 100, \quad K_3 = 200, \quad K_4 = 150, \]

\[ C = 800, \quad a_i = 0.6, \quad i = 1, 2, 3, 4. \]

The problem is formulated in SUMT format as follows:

Minimize

\[ f(x) = -R_s \]

\[ = -1 + R_3 [(1 - R_1)(1 - R_4)]^2 + (1 - R_3) \left[ 1 - R_2 \left[ 1 - (1 - R_1)(1 - R_4) \right] \right]^2 \]

subject to the constraints
\[
g_1(x) = C - \left(2K_1R_1^\alpha_1 + 2K_2R_2^\alpha_2 + K_3R_3^\alpha_3 + K_4R_4^\alpha_4 \right) \geq 0
\]

\[
g_{i+1}(x) = 1 - R_i \geq 0, \quad i = 1, 2, 3, 4
\]

The P function of equation (13) is

\[
P(x, r_k) = -1 + R_3[(1-R_1)(1-R_4)]^2 + (1-R_3)(1-R_2)[1-(1-R_1)(1-R_4)]^2
\]

\[
+ r_k \left[ \frac{1}{C - (2K_1R_1^\alpha_1 + 2K_2R_2^\alpha_2 + K_3R_3^\alpha_3 + K_4R_4^\alpha_4)} \sum_{i=1}^{4} \frac{1}{1-R_i} \right]
\]

The optimal solutions which were obtained by starting from two different points, namely, \([R_1, R_2, R_3, R_4] = [0.7, 0.7, 0.7, 0.7]\) and \([R_1, R_2, R_3, R_4] = [0.6, 0.6, 0.6, 0.6]\) are presented in Table 1. The solutions are almost identical, that is, the optimal system reliability \(R_s\) is equal to 0.99996 with the cost of 799.78 for the first starting point and \(R_s\) equal to 0.99995 with the cost of 799.28 for the second starting point. Recall that the constraint on the cost is 800. Note that the optimal components reliabilities are almost the same for both starting points. The stopping criterion for terminating the minimization of the P function at each \(k\) iteration is \(\varepsilon' = 10^{-5}\), and the stopping criterion for terminating the overall minimization of \(f[x(r_k)]\) is \(\varepsilon = 10^{-4}\). For the first starting point, it required 10 iterations for the P functions with a total of 152 functional values calculated, and for the second point, 11 iterations were required for the P functions with a total of 167 functional values calculated.
<table>
<thead>
<tr>
<th>Iteration $k$</th>
<th>Number of functional value calculated</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$R_4$</th>
<th>System Reliability $R_s$</th>
<th>Cost</th>
<th>Stopping criterion $\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.99996</td>
<td>799.78</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>10</td>
<td>152</td>
<td>0.9876</td>
<td>0.9936</td>
<td>0.6972</td>
<td>0.6941</td>
<td>0.99996</td>
<td>799.78</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>0</td>
<td>167</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.99995</td>
<td>799.28</td>
<td>$10^{-5}$</td>
</tr>
</tbody>
</table>
Tables 2a and 2b present some suboptimal solutions according to different stopping criterions. From these tables, we can see that the number of iterations, $k$, is dictated by the final stopping criterion, $\varepsilon$, and that the number of functional values calculated for each iteration is dictated by the stopping criterion for each iteration, $\varepsilon'$. The number of iterations, $k$, increases from 4 for $\varepsilon = 10^{-2}$ to 10 for $\varepsilon = 10^{-4}$, and the number of functional values calculated for each iteration increases from an average of 1 for $\varepsilon' = 10^{-2}$ to an average of 14 for $\varepsilon' = 10^{-4}$.

Although the cost for each suboptimum solution is near the cost constraint of 800, the systems reliability and corresponding set of components reliabilities are different for each combination of $\varepsilon'$ and $\varepsilon$. The highest system reliability is obtained when the stopping criterions are $\varepsilon' = 10^{-5}$ and $\varepsilon = 10^{-4}$.

Results given in Tables 3a and 3b show that the system reliability, $R_s$, is monotonically increasing as the iteration $k$ increases. The value of the $P$ function approaches that of the $f$ function ($= -R_s$) as the iterations proceed. Thus the minimization of the $P$ function will eventually lead to the minimization of $f$ function.

2.6 DISCUSSION

This approach provides a practical method for solving a very complex reliability problem. The system may be one where the redundant components cannot be reduced to a parallel or series configuration. The reliability function is obtained by using Bayes' theorem and a mathematical model is formulated for the constrained nonlinear programming problem. The solution of the problem is obtained by the sequential unconstrained minimization technique (SUMT). As is evident from the results obtained
<table>
<thead>
<tr>
<th>Stopping criterion</th>
<th>Iteration</th>
<th>Number of functional value calculated</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$R_4$</th>
<th>System Reliability $R_s$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-4}$</td>
<td>0</td>
<td></td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.9584</td>
<td>726.61</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>4</td>
<td>27</td>
<td>0.8186</td>
<td>0.9222</td>
<td>0.7690</td>
<td>0.7843</td>
<td>0.9958</td>
<td>798.02</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>6</td>
<td>105</td>
<td>0.9333</td>
<td>0.9722</td>
<td>0.6937</td>
<td>0.7336</td>
<td>0.9992</td>
<td>798.23</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>9</td>
<td>10</td>
<td>0.8156</td>
<td>0.9147</td>
<td>0.7715</td>
<td>0.7991</td>
<td>0.9957</td>
<td>799.95</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>9</td>
<td>79</td>
<td>0.8091</td>
<td>0.9620</td>
<td>0.7512</td>
<td>0.7876</td>
<td>0.9973</td>
<td>799.94</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>10</td>
<td>140</td>
<td>0.9527</td>
<td>0.9926</td>
<td>0.6850</td>
<td>0.7242</td>
<td>0.99975</td>
<td>799.95</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>10</td>
<td>152</td>
<td>0.9876</td>
<td>0.9936</td>
<td>0.6972</td>
<td>0.6941</td>
<td>0.99996</td>
<td>799.78</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>11</td>
<td>12</td>
<td>0.8156</td>
<td>0.9149</td>
<td>0.7715</td>
<td>0.7991</td>
<td>0.9957</td>
<td>799.99</td>
</tr>
<tr>
<td>Stopping criterion</td>
<td>Number of iterations</td>
<td>$P_1$</td>
<td>$P_2$</td>
<td>$P_3$</td>
<td>$P_4$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$ 10^{-2}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$ 10^{-3}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$ 10^{-4}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$ 10^{-5}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td>10^{-3}</td>
<td>10^{-4}</td>
<td>10^{-2}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2b**: Suboptimal solutions according to different stopping criteria.
<table>
<thead>
<tr>
<th>Iteration k</th>
<th>Number of functional value calculated at each iteration</th>
<th>Value of ( r_k )</th>
<th>( R_1 )</th>
<th>( R_2 )</th>
<th>( R_3 )</th>
<th>( R_4 )</th>
<th>-P</th>
<th>(-\frac{\pi}{\pi_s})</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>3000</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.95480</td>
<td>726.61</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.01427</td>
<td>0.6877</td>
<td>0.7504</td>
<td>0.6687</td>
<td>0.6723</td>
<td>0.6850</td>
<td>0.9577</td>
<td>721.61</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.003568</td>
<td>0.7475</td>
<td>0.8223</td>
<td>0.7200</td>
<td>0.7334</td>
<td>0.9019</td>
<td>0.9815</td>
<td>759.10</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.000892</td>
<td>0.8012</td>
<td>0.8764</td>
<td>0.7594</td>
<td>0.7860</td>
<td>0.9682</td>
<td>0.9924</td>
<td>739.08</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.000223</td>
<td>0.8139</td>
<td>0.9095</td>
<td>0.7708</td>
<td>0.7980</td>
<td>0.9883</td>
<td>0.9953</td>
<td>798.77</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.0000553</td>
<td>0.8150</td>
<td>0.9125</td>
<td>0.7713</td>
<td>0.7988</td>
<td>0.9937</td>
<td>0.9956</td>
<td>799.52</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.00001394</td>
<td>0.8153</td>
<td>0.9138</td>
<td>0.7715</td>
<td>0.7990</td>
<td>0.9951</td>
<td>0.99564</td>
<td>799.78</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0.000003485</td>
<td>0.8155</td>
<td>0.9143</td>
<td>0.7715</td>
<td>0.7991</td>
<td>0.9955</td>
<td>0.99567</td>
<td>799.90</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0.0000008712</td>
<td>0.8156</td>
<td>0.9147</td>
<td>0.7715</td>
<td>0.7991</td>
<td>0.9956</td>
<td>0.99569</td>
<td>799.95</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0.000000218</td>
<td>0.8156</td>
<td>0.9148</td>
<td>0.7715</td>
<td>0.7991</td>
<td>0.9957</td>
<td>0.995702</td>
<td>799.98</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.000000054</td>
<td>0.81563</td>
<td>0.9149</td>
<td>0.7715</td>
<td>0.7991</td>
<td>0.9957</td>
<td>0.995706</td>
<td>799.99</td>
</tr>
<tr>
<td>Iteration k</td>
<td>Number of functional value calculated at each iteration</td>
<td>$R_1$</td>
<td>$R_2$</td>
<td>$R_3$</td>
<td>$R_4$</td>
<td>$-P$</td>
<td>$-f$ ($=R_5$)</td>
<td>Cost</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>----------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>3000</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.0</td>
<td>0.8362</td>
<td>786.24</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.066595</td>
<td>0.5869</td>
<td>0.6330</td>
<td>0.5861</td>
<td>0.5828</td>
<td>-0.2231</td>
<td>0.8888</td>
<td>785.95</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.01665</td>
<td>0.6556</td>
<td>0.7357</td>
<td>0.6617</td>
<td>0.6669</td>
<td>0.6406</td>
<td>0.9502</td>
<td>712.98</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.004162</td>
<td>0.72695</td>
<td>0.8136</td>
<td>0.7172</td>
<td>0.7372</td>
<td>0.8886</td>
<td>0.9794</td>
<td>755.56</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.001041</td>
<td>0.7829</td>
<td>0.8706</td>
<td>0.7587</td>
<td>0.7933</td>
<td>0.9640</td>
<td>0.9916</td>
<td>787.26</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.0002601</td>
<td>0.7971</td>
<td>0.9085</td>
<td>0.7722</td>
<td>0.8080</td>
<td>0.9870</td>
<td>0.9952</td>
<td>798.62</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.000065</td>
<td>0.7983</td>
<td>0.9117</td>
<td>0.7729</td>
<td>0.8091</td>
<td>0.9933</td>
<td>0.9954</td>
<td>799.48</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.0000163</td>
<td>0.7987</td>
<td>0.9130</td>
<td>0.7730</td>
<td>0.8094</td>
<td>0.9950</td>
<td>0.9955</td>
<td>799.76</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.000004065</td>
<td>0.7988</td>
<td>0.9136</td>
<td>0.7731</td>
<td>0.8095</td>
<td>0.9954</td>
<td>0.9955</td>
<td>799.89</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.00000102</td>
<td>0.799</td>
<td>0.9139</td>
<td>0.7731</td>
<td>0.8096</td>
<td>0.9955</td>
<td>0.9955</td>
<td>799.95</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.00000025</td>
<td>0.7989</td>
<td>0.9141</td>
<td>0.7731</td>
<td>0.8096</td>
<td>0.9955</td>
<td>0.9955</td>
<td>799.97</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.000000064</td>
<td>0.79895</td>
<td>0.9144</td>
<td>0.7731</td>
<td>0.8096</td>
<td>0.9954</td>
<td>0.9955</td>
<td>799.99</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 3b. Computer results of a suboptimal results for stopping criterion ($\varepsilon' = 10^{-2}$ and $\varepsilon = 10^{-5}$)
in solving the example problem this is an efficient method for solving a difficult problem.

The complex reliability system presented in Fig. 1, can be identified to many practical systems concerning with the space life support systems.

One such example is a communication system of a two man space capsule as shown in Fig. 1. The unit 2 represents each of the two microphones of the headsets of each astronaut in the capsule. Unit 3 is a hand microphone which may be picked up by either astronaut. There are two different type of amplifiers in the system with units 1 and 4 respectively. Such a system is identical to that we have studied in this chapter.

Another example is a high pressure oxygen supply system as shown in Fig. 2. The high pressure oxygen in the cabin is supplied through a system of regulators and valves from a high pressure oxygen storage tank. There are two pairs of the sub-systems of check valves, shut-off valves and non-return automatic shut-off valves in the system. The function of these valves is to stop the reverse flow of air from the cabin to the gas tank in case of pressure drop and to close the line supply if there is same sudden pressure drop in header line or the cabin in order to avoid the wastage of the gas.

Each pair of the valve systems consists of two alternative branches. One consists of a non-return automatic emergency shut off valve, and the other consists of a check valve and a shut off valve in series. Any branch of the two pairs (totally four branches) is capable of supplying sufficient gas to the cabin.

There are three alternative paths between the $O_2$ tank and the pairs of valves. The $O_2$ can pass through either of the two regulator to
Fig. 2. High pressure O$_2$ supply system of a spacecraft life support system.
the pair of valves connected to that regulator then supply to the cabin. It also can pass through a selector valve to either of the two pairs of valves then supply to the cabin.

Suppose the reliability of the high pressure O₂ tank can be considered as 1, and denote the reliability for the regulators (they are the same kind of regulators and have the same reliability) by R₂, the reliability for the selector valve by R₃; the reliability for the non-return automatic emergency valve by R₁; and the reliability for the series of check valve and shut-off valve by R₄. Then the system can be reduced to the system presented in Fig. 1 which has been studied in this chapter.

By grouping all the parallel as well as series parts of a complex reliability system into local sub-systems in the whole system and treating them as single components, the system can often be reduced to such configuration that Bayes’ theorem of conditional probability shall be able to be employed.
REFERENCES


CHAPTER 3

IMPLEMENTATION OF SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE
BY HOOKE AND JEEVES PATTERN SEARCH AND HEURISTIC PROGRAMMING

3.1. INTRODUCTION

The general nonlinear programming problem with nonlinear (and/or linear) inequality and/or equality constraints is to choose \( x \) to

\[
\begin{align*}
\text{minimize } & \ f(x) \\
\text{subject to } & \ g_i(x) \geq 0, \ i = 1, 2, \ldots, m \\
& \ h_j(x) = 0, \ j = 1, 2, \ldots, \ell
\end{align*}
\]

(3.1)

where \( x \) is an \( n \)-dimensional vector \( (x_1, x_2, \ldots, x_n) \). To solve this problem, there are a number of techniques developed recently. Among them, a technique which was originally proposed by Carroll [1,2] and further developed by Fiacco and McCormic [3,4,5,6,7] is introduced here.

This technique, known as the sequential unconstrained minimization technique (SUMT), is considered as one of the simplest and most efficient methods for solving the problem given by equation (3.1). The basic scheme of this technique is that a constrained minimization problem is transformed into a sequence of unconstrained minimization problems which can be optimized by any available techniques for solving unconstrained minimization.

The unconstrained minimization technique which is employed here is the well-known Hooke and Jeeves pattern search technique [3,9]. For increasing the efficiency of the method, some modifications have been made.* Among these modifications, a heuristic programming technique [10] is used to handle the inequality constraints of the problem given by equation (3.1).
The method and its computational procedure is illustrated in details in the following sections of this chapter.*

3.2. SEQUENTIAL UNCONSTRAINED MINIMIZATION TECHNIQUE (SUMT)

The SUMT technique for solving the problem given in equation (3.1) is based on the minimization of a function

\[ P(x, r_k) = f(x) + r_k \sum_{i=1}^{m} \frac{1}{g_i(x)} + r_k \sum_{j=1}^{p} \frac{1}{L_j} h_j^2(x) \]  \hspace{1cm} (3.2)

over a strictly monotonic decreasing sequence \( \{r_k\} \). Under certain restrictions, the sequence of values of the P function, \( P(x, r_k) \), are respectively minimized by a sequence of \( \{x(r_k)\} \) over a strictly monotonic decreasing sequence \( \{r_k\} \), converges to the constrained optimum values of the original objective function, \( f(x) \). The essential requirement is the convexity of the P function.

The intuitive concept of P function is described below:

Since the sequence \( \{r_k\} \) is strictly monotonic decreasing, as \( r_k \to 0 \) the third term of the P function defined in equation (3.2), \( r_k \sum_{j=1}^{p} \frac{1}{L_j} h_j^2(x) \), will approach to \( \infty \) unless \( h_j(x) = 0 \) for \( j = 1, 2, \ldots, p \). While we are minimizing P function, the formulation of P function in equation (3.2) will force all equality constraints to be zero.

For the second term of the P function, \( r_k \sum_{i=1}^{m} \frac{1}{g_i(x)} \), when we start at a point which is inside the feasible region bounded by inequality

* Developments of this modified method and the computer program for implementing SUMT by the Hooke and Jeeves pattern search technique were not financially supported by any source. The possibility of developing the method and computer program was suggested to the author by Professors L. T. Fan and C. L. Hwang (11).
constraints to minimize the $P$ function, $r_k \sum_{i=1}^{m} 1/g_i(x)$ will approach to infinity as the value of $x$ approaches to one of the boundary of the inequality constraints given by equation (3.1), $g_i(x) \geq 0$. Hence, the value of $x$ will tend to remain inside the inequality-constrained feasible region.

The motivation behind this formulation of $P$ function is the transformation of the original constrained problem into a sequence of unconstrained minimization problem, \{\!\{P(x, r_k)\}\!\}. The solution to the problem then is to define the $P$ function as shown in equation (3.2) first. To search for the minimum $P$ function value it is started at an arbitrary point which is inside the feasible region bounded by the inequality constraints. After a minimum $P$ function value is reached, the value of $r_k$ is reduced, and a search is repeated again starting from the previous minimum point of the $P$ function. By employing a strictly monotonic decreasing sequence $\{r_k\}$, a monotonic decreasing sequence \{\!\{P_{\text{min}}(x, r_k)\}\!\} inside the feasible region bounded by the inequality constraints is obtained. The equality constraints, $h_j(x) = 0$ for $j = 1, 2, \ldots, t$, will be satisfied by the nature of the formulation of the $P$ function automatically as $r_k$ tends to zero as explained before.

When $r_k \to 0$, the second term of equation (3.2), $r_k \sum_{i=1}^{m} 1/g_i(x)$ approaches to zero, while the third term, $r_k^{\frac{1}{2}} \sum_{j=1}^{t} h_j^2(x)$, is forced to approach to zero as described before. In other words, as $r_k \to 0$, $P(x, r_k) \to f(x)$, where $x$ is the optimum point which yields the minimum $P(x, r_k)$ and is the optimum point of the problem given in equation (3.1).
Further mathematical proof of the convergence of the method can be
seen in reference [3,4,5,6,7].

3.3. COMPUTATIONAL PROCEDURE

The computational procedure for using SUMT with Hooke and Jeeves
pattern search technique is summarized below (refer to Fig. 1).

(1) Select a starting point \( x^0 = (x^0_1, x^0_2, \ldots, x^0_n) \) and initial
values of the penalty coefficient \( r^0_k \), an initial tolerance limit of the
violation to constraints, \( B^0 \), and the initial step-sizes needed in
search processes, \( d^0 \).

(2) Select a feasible starting point by minimizing the total weight
of violation, if the initial starting point chosen, \( x^0 \), is out of the
feasible region bounded by the inequality constraints. The total weight
of violation, \( TGH \), is defined as [10]

\[
TGH = \left( \sum_{t \in T} g^2_t(x^0) + \sum_{s \in R} h^2_s(x^0) \right)^{1/2}
\]

where \( T = \{ t | g^0_t(x^0) < 0 \} \) and \( R = \{ s | h^0_s(x^0) \neq 0 \} \). Note that \( TGH \) includes
only the violated constraints.

(3) Define \( P \) function as [6,7]

\[
P(x, r_k) = f(x) + r_k \sum_{i} \frac{1}{g^1_i(x)} + r_k \left( \frac{1}{2} \sum_{j} h^2_j(x) \right)
\]

where \( g^1_i(x) > 0, i = 1, 2, \ldots, m \) are inequality constraints, and
\( h^1_j(x) = 0, j = 1, 2, \ldots, \ell \) are equality constraints.

(4) Minimize \( P \) function by Hooke and Jeeves pattern search
technique. After every move during the search, it is checked if the
move goes out of the feasible region or not. If the move is out of the
feasible region, got to step 5; if not, after the optimum x is reached for the current $P(x, r_k)$, go to step 6.

(5) Move back to the near-feasible region and then return to step 4. The near-feasible region is defined as the region that all the points in that region satisfy the following condition [10].

$$B - TGH > 0$$

where B is the tolerance limit of violation which is sequentially decreased after every violation to the inequality constraints during the search.

(6) Check if the optimum, $\bar{x}$, obtained in step 4 is inside the feasible region or not. If $\bar{x}$ is feasible, go to step 8, and it it is near-feasible or not feasible, go to step 7.

(7) Move the optimum $\bar{x}$ in the infeasible region into the feasible region along the direction toward the last optimum point, then go to step 8.

(8) Check if a stopping criterion such as

$$\left| \frac{\bar{f}(x)}{G(x, r_k)} - 1 \right| < \varepsilon$$

is satisfied. The solution is the optimal one if the criterion is satisfied; otherwise, go to step 9. The dual value $G(x, r_k)$, is defined as [6,7]

$$G(x, r_k) = f(x) - r_k \sum_{i=1}^{m} \frac{1}{g_i(x)} + r_k \sum_{j=1}^{k} \frac{1}{h_j(x)}$$

(9) Set $k = k+1$; $r_{k+1} = r_k / C$, where C is a constant and greater than 1; and $d_{k+1} = d^0/(k+1)$; $d_{k+1}$ to be the starting step-sizes; and go
Fig. 1. Descriptive flow diagram for SUMT with Hooke and Jeeves Pattern Search.
back to step 3.

The following sections present in details procedures of each step described above. The basic Hooke and Jeeves pattern search is presented in Section 3.5.

3.4. PROCEDURE FOR SELECTING A FEASIBLE STARTING POINT FROM THE INFEASIBLE INITIAL POINT

The procedure for selecting a feasible starting point when the initial point is out of the feasible region bounded by inequality constraints, \( g_i(x) \geq 0 \) for \( i = 1, 2, \ldots, m \), is based on Hooke and Jeeves pattern search technique. For increasing the speed and efficiency of the process, some modifications from the basic Hooke and Jeeves pattern search technique have been made.

Note that in above description of the feasible region only the inequality constraints are included. The violation to equality constraints is not considered here but it is taken into account in the SUMT formulation automatically as explained in Section 3.2 [6,7].

The procedure is summarized below (refer to Fig. 2).

(1) Start at the input initial point, \( x_0 \), which is out of the feasible region bounded by the inequality constraints and needs to be moved into the feasible region.

(2) Compute the weight of violation, TGH, at the initial point: [10]

\[
TGH = \left \{ \sum_{t \in T} [g_t(x_0)]^2 + \sum_{s \in R} [h_s(x_0)]^2 \right \}^{1/2}
\]

where \( T = \{t \mid g_t(x_0) < 0\} \) and \( R = \{s \mid h_s(x_0) \neq 0\} \). Note, again, that TGH includes only the violated constraints.
Fig. 2. Descriptive flow diagram for selecting a feasible starting point.
(3) Make an exploratory move to minimize TGH from $x^0$. Note that, the objective function to be minimized in this step is TGH which has been defined in step 2. For increasing the efficiency of the process, two modifications are made here. First, the starting step-sizes used is twice the input initial starting step-sizes, which is used in minimizing the $P(x,r_k)$ function as described in Section 3.5. Second, after every successful move, the feasibility is checked; whenever a move has reached a point which is inside the feasible region bounded by inequality constraints, the process of selecting a feasible starting point is terminated. And the feasible point obtained is used as the desired feasible starting point.

(4) Check if the exploratory move has made any progress; in the other words, it searches a new point which has a less value of TGH than the base point of the exploratory move does. If it does not, cut down the step-sizes and go back to step 2, if it does, go to step 5.

(5) Convert the exploratory move point to be the new base point; let it be $x^0$.

(6) Make a pattern move along the line connecting the two base points to a new pattern move point $x^p$.

(7) Check if $x^p$ has a less value of TGH than $x^0$ does. Return to step 3 if the answer is negative. If $x^p$ does make progress, check if it is in the feasible region bounded by the inequality constraints. Terminate the process of selecting a feasible starting point and use $x^p$ as the feasible starting point if $x^p$ is feasible. Otherwise, set $x^0 = x^p$ and return to step 3.
3.5. **COMPUTATIONAL PROCEDURE FOR MINIMIZING P(x,r_k) FUNCTION BY THE HOOKE AND JEEVES PATTERN SEARCH**

The computational procedure for minimizing the P(x,r_k) function is the basic Hooke and Jeeves pattern search technique [8,9]. The method is a sequential search routine for searching a point \( x = (x_1, x_2, \ldots, x_n) \) which minimize the function, P(x,r_k). A descriptive flow diagram of the method is given in Fig. 3. The procedure consists of two types of moves: **Exploratory** and **Pattern**.

A **move** is defined as the procedure of going from a given point to the following point. A move is a **success** if the value of the P(x,r_k) decreases; otherwise, it is a **failure**. The first type of move is an exploratory move which is designed to explore the local behavior of the function, P(x,r_k). The success or failure of the exploratory move is utilized by combining it into a pattern which indicates a probable direction for a successful move [8,9].

The exploratory move is performed as follows:

1. Introduce a starting point \( x \) with a prescribed step size \( d_i \) in each of the independent variables \( x_i, i = 1, 2, \ldots, n \).
2. Compute the function, P(x,r_k), where \( x = (x_1, x_2, \ldots, x_n) \). Set \( i = 1 \).
3. Compute \( P_i(x,r_k) \) at the trial point
   \[ x = (x_1, x_2, \ldots, x_i + d_i, x_{i+1}, \ldots, x_n) \).
4. Compare \( P_i(x,r_k) \) with P(x,r_k):
   
   (1) If \( P_i(x,r_k) < P(x,r_k) \), set \( P(x,r_k) = P_i(x,r_k) \), \( x = (x_1, x_2, \ldots, x_n) = (x_1, x_2, \ldots, x_i + d_i, \ldots, x_n) \), and \( i = i+1 \).

   Consider this trial point as a starting point, and repeat from step 3.
(ii) If \( P_i(x, r_k) \geq P(x, r_k) \), set \( x = (x_1, x_2, \ldots, x_i - 2d_i, \ldots, x_n) \). Compute \( P_i(x, r_k) \), and see if \( P_i(x, r_k) < P(x, r_k) \). If this move is a success the new trial point is retained. Set \( P(x, r_k) = P_i(x, r_k) \), \( x = (x_1, x_2, \ldots, x_i, \ldots, x_n) = (x_1, x_2, \ldots, x_i - 2d_i, \ldots, x_n) \), and \( i = i + 1 \), and repeat from step 3. If again \( P_i(x, r_k) \geq P(x, r_k) \), then the move is a failure and \( x_i \) remains unchanged, that is, \( x = (x_1, x_2, \ldots, x_i, \ldots, x_n) \).

Set \( i = i + 1 \) and repeat from step 3.

The point \( x_B \) obtained at the end of the exploratory moves, which is reached by repeating step 3 until \( i = n \), is defined as a base point. The starting point introduced in step 1 of the exploratory move is a starting base point or point obtained by the pattern move.

The pattern move is designed to utilize the information acquired in the exploratory move, and executes the actual minimization of the function by moving in the direction of the established pattern. The pattern move is a simple step from the current base to the point

\[
x = x_B + (x_B^* - x_B^*)
\]

\( x_B^* \) is either the starting base point or the preceding base point. Following the pattern move a series of exploratory moves is conducted to further improve the pattern. If the pattern move followed by the exploratory moves brings no improvement, the pattern move is a failure. Then we return to the last base which becomes a starting base and the process is repeated.

If the exploratory moves from any starting base do not yield a point which is better than this base, all the step sizes are reduced and the
Fig. 3. Descriptive flow diagram for Hooke and Jeeves Pattern Search for minimizing \( P(X, r_k) \) function.
moves are repeated. Convergence is assumed when the step sizes, $d_i$, have been reduced below predetermined limits.

The following modifications are made so that the above method originally developed for unconstrained minimization shall be able to handle inequality constraints.

(i) During the exploratory moves, every successful move is checked to see if it goes out of the feasible region bounded by the inequality constraints. If it is so, it will be moved back into feasible or near-feasible region according to the procedure described in Section 3.6, and then continue on the regular search routine.

(ii) After making a pattern move, the function, $P(x, r_k)$, is evaluated. Check if the pattern move makes progress. If the pattern move makes no progress, return to the base point and make an exploratory move from the base point. If the pattern move makes progress, check if the pattern move point is feasible (subject to the inequality constraints only). Move back into the feasible or the near-feasible region bounded by the inequality constraints according to the procedure described in Section 3.6 if the success pattern move is infeasible.

3.6. PROCEDURE FOR MOVING AN INFEASIBLE POINT INTO THE FEASIBLE OR NEAR-FEASIBLE REGION BOUNDED BY INEQUALITY CONSTRAINTS

The procedure for moving an infeasible point into the feasible or the near-feasible region bounded by the inequality constraints is based on a simplified Hooke and Jeeves pattern search. Since the optimum will
be located at somewhere very close to the boundary of the set of constraints for most of the constrained problems, the moving procedure used here consists of small step size exploratory moves only. Pattern moves are not used.

The procedure is summarized below (refer to Fig. 4).

1. Start at the infeasible point, \( x \), which is to be moved into the feasible or the near-feasible region bounded by inequality constraints.

2. Compute the weight of violation, \( T_{GH} \), at \( x \),

\[
T_{GH} = \left\{ \frac{1}{2} \left[ \sum_{t \in T} g_{t}(x)^2 + \sum_{s \in R} h_{s}(x)^2 \right] \right\}^{\frac{1}{2}}
\]

where \( T = \{ t | g_{t}(x) < 0 \} \) and \( R = \{ s | h_{s}(x) \neq 0 \} \).

3. Decide the tolerance limit, \( B \), which is sequentially decreased, for example 3/4 of the preceding value, after each moving back process. The starting tolerance limit, \( B_{0} \), for the \( k \)-th sub-optimum search is defined as [10]

\[
B_{k}^{0} = 0.5 \sum_{i=1}^{n} d_{i}/n
\]

where \( d_{i} \) is the starting step-sizes of the \( i \)-th dimension for the \( k \)-th sub-optimum search; \( n \) is the dimension of the problem. This implies that the starting tolerance limit for the \( k \)-th sub-optimum is set to be a half of the average starting step-sizes. After an infeasible point is moved back to the feasible or near-feasible region bounded by inequality constraints, the size of the tolerance limit is decreased.

4. Check if \( x \) is at least in the near-feasible region. If the answer is positive, go to step 7, otherwise, set \( x \) as the base point
Enter

Start at the infeasible point which need to be pulled back into near-feasible region.

Compute the weight of violation:
\[ TGH = \left( \sum_i \left( g_f(x) \right)^2 + \sum_s \left( h_s(x) \right)^2 \right)^{1/2}, \text{for all } g_f(x) \leq 0, h_s \neq 0 \]

Decide the tolerance limit, B

Is \[ B - TGH \geq 0? \]

YES

check for every move if \[ B - TGH \geq 0? \]

YES

Decrease the tolerance limit, B

NO

Start at base point

Make exploratory move for minimizing TGH, with step-sizes \( \frac{1}{2} \) of the entered step-sizes.

Did exploratory move make progress?

YES

Set new base point

NO

Cut down step sizes

EXIT

Fig. 4. Descriptive flow diagram for moving an infeasible point back into near feasible region.
and go to step 5. The near-feasible region is defined as the point set
\[ A = \{ x | B - TGH > 0 \} \].

(5) Start at the base point and make an exploratory move for
minimizing TGH, with step-sizes one half of the current step-sizes
entered to this routine. Whenever a move is feasible or near-feasible,
go to step 7; otherwise go to step 6.

(6) Check if the exploratory move makes progress. If the answer
is positive, set the exploratory move point to be the new base point
and go to step 5. Otherwise, reduce step-sizes then start at the old
base point, go to step 5.

(7) Reduce the tolerance limit B which will be used as the starting
tolerance limit for next moving back procedure when a preceding move go
out of the feasible region again; set the point which satisfies the
formula
\[ B - TGH > 0 \]
to be \( x \) and terminate the process of moving back procedure.

3.7. PROCEDURE FOR MOVING THE NEAR-FEASIBLE k-TH SUB-OPTIMUM INTO THE
FEASIBLE REGION

After the k-th sub-optimum has been reached, it is desirable to
have the optimum point in the feasible region subject to all the inequality
constraints.

If the optimal point for \( P(x, r_k) \) is in the near-feasible region but
not in the feasible region, it will be moved back into the feasible
region by the following procedure (refer to Fig. 5).
(1) Compute the weight of violation, $TGH$, at the near-feasible $k$-th sub-optimum, $x_k^0$.

$$TGH = \left\{ \left( \sum_{t \in T} [g_t(x_k^0)]^2 + \sum_{s \in R} [h_s(x_k^0)]^2 \right)^{\frac{1}{2}} \right\}$$

where $T = \{ t | g_t(x_k^0) < 0 \}$ and $R = \{ s | h_s(x_k^0) \neq 0 \}$.

(2) Move $x_k^0$ toward $x_{k-1}^0$, the feasible $(k-1)$-th sub-optimum for a small step $\delta$ to obtain a new point $x_k^{0'}$.

(3) Set $x_k^0 = x_k^{0'}$ and check if $x_k^0$ is feasible. If $x_k^0$ is not feasible, go to step 2; if $x_k^0$ is feasible, terminate the process.
Compute the weight of violation

\[ T_{GH} = \left\{ \sum g_i(x_k^o) \right\}^2 + \sum h_s(x_k^o) \right\}^{1/2} \]

for all \( g_i \leq 0, h_s \neq 0 \)

where \( x_k^o \) is the entering \( k \)-th sub-optimum.

Move \( x_k^o \) toward \( x_{k-1}^o \), the feasible \((k-1)\)-th sub-optimum for a step to a new point \( x_k^{o'} \).

Set \( x_k^o = x_k^{o'} \).

Is \( x_k^o \) feasible?

Yes

EXIT

No

Fig. 5. - Descriptive flow diagram for moving the near-feasible \( k \)-th sub-optimum into feasible region.
REFERENCES


CHAPTER 4

SUMT IMPLEMENTED BY HOOK AND JEEVES SEARCH TECHNIQUE
APPLIED TO PRODUCTION SCHEDULING PROBLEMS

4.1 INTRODUCTION

To illustrate the sequential unconstrained minimization technique (SUMT) implemented by the Hooke and Jeeves pattern search technique, two production scheduling problems, a two dimensional production scheduling problem with four inequality constraints [1, 3] and a twenty dimensional personnel and production planning problem with forty inequality constraints [2, 3, 5], are considered here.

The problems and their solutions are described in the following sections of this chapter.

4.2 A PRODUCTION SCHEDULING AND INVENTORY CONTROL PROBLEM

The problem is to minimize the sum of the production cost and inventory cost subject to the constraints of non-negative inventory and the maximum capacity of machine which produces the desired items. The demand of each period is known and must be satisfied.

The cost for changing the production level and for carrying inventory are given by

\[ C(\theta_i - \theta_{i-1})^2 = \text{Cost due to the change in production level from the (i-1)th period to the i-th period,} \]

\[ D(E - I_i)^2 = \text{Inventory cost at the i-th period,} \]

where \(C\), \(D\), and \(E\) are positive constants. \(\theta_i\) and \(I_i\) are the production level and the inventory level at the i-th period respectively.

The problem is to find \(\theta^* = (\theta_1^*, \theta_2^*, \ldots, \theta_n^*)\) which minimizes
\[ f(\theta) = \sum_{i=1}^{n} [C(\theta_i - \theta_{i-1})^2 + D(E - I_i)^2] \tag{4.1} \]

subject to

\[
\begin{align*}
I_i &= I_{i-1} + \theta_i - Q_i \quad 0, \quad i = 1, 2, \ldots, n \\
\text{and} \\
0 &\leq \theta_i \leq M, \quad i = 1, 2, \ldots, n
\end{align*}
\]  \tag{4.2}

where \( M \) is the maximum production capacity. \( Q_i \) represents the sales at the \( i \)-th period. \( \theta_0 \) and \( I_0 \) are the production level and inventory level at the initial period respectively.

**NUMERICAL EXAMPLE 1**

For this example, a two period production and inventory system is presented. The optimal decision variable \( \theta^* = (\theta_1^*, \theta_2^*) \) will be determined by solving the following problem.

Minimize

\[ f(\theta) = C(\theta_1 - \theta_0)^2 + D(E - I_1)^2 + C(\theta_2 - \theta_1)^2 + D(E - I_2)^2 \tag{4.3} \]

subject to

\[
\begin{align*}
g_1(\theta) &= I_1 = I_0 + 1 - Q_1 \geq 0 \\
g_2(\theta) &= I_2 = I_1 + 2 - Q_2 \geq 0 \\
g_3(\theta) &= M - 1 \geq 0 \\
g_4(\theta) &= M - 2 \geq 0
\end{align*}
\]

\[ \tag{4.4} \]

The values of \( C, D, E, M, \theta_0, I_0, \) and \( Q_i, \) \( i = 1, 2, \) are given as

\[
\begin{align*}
C &= 100, & D &= 20, & E &= 10, & M &= 30, \\
\theta_0 &= 15, & 0 &= 12, & Q_1 &= 30, & Q_2 &= 10.
\end{align*}
\]
To illustrate the procedure the contour lines for equal values of total cost, given by equation (4.3), are shown in Fig. 1. The shaded area represents the feasible region bounded by the inequality constraints given by equation (4.4). The global minimum, \( \theta^{**} = (\theta_1^{**}, \theta_2^{**}) = (17.82, 18.21) \), of the original unconstrained problem [1] is apparently located outside the feasible region.

The \( P \) function of this problem is

\[
P(\theta, r_k) = f(\theta) + r_k \left( \sum_{i=1}^{4} \frac{1}{\theta_i(\theta)} \right)
\]

\[
= 100(\theta_1 - 15)^2 + 20(28 - \theta_1)^2 + 100(\theta_2 - 6)^2 + 20(38 - \theta_1 - \theta_2)^2
\]

\[
+ r_k \left( \frac{1}{\theta_1 - 18} + \frac{1}{\theta_1 + \theta_2 - 28} + \frac{1}{30 - \theta_1} + \frac{1}{30 - \theta_2} \right)
\]

The step by step procedure of SUMT implemented by the Hooke and Jeeves pattern search technique is as follows:

1. Let the initial value of \( r \) be \( r_0 = 3000 \). This value of \( r_0 \) has been selected arbitrarily.

2. Let the initial starting point \( \theta^0 = (25, 29) \). Note that \( \theta^0 \) is in the feasible region.

3. Obtain the optimal solution, \( \theta^* = (\theta_1^*, \theta_2^*) = (19.75, 19.00) \), by minimizing the \( P \) function for the current value of \( r \). The minimization technique used is the Hooke and Jeeves pattern search technique (details have been discussed in Chapter 3).

4. Check if the stopping criterion is satisfied. The values of the objective function evaluated at \( \theta^0 \) and \( \theta^* \) are \( f(\theta^0) = 16,900 \) and \( f(\theta^*) = 3,418.75 \) respectively. It indicates the rapid rate of
Fig. 1. Production scheduling problem involving two decision variables; contour lines indicate equal quantities of total cost given by equation (4.3).
convergence at the first iteration. The stopping criterion,
\[ \frac{f(\theta)}{G(\theta)} - 1, \]
has the value 7.71 > 10^{-4}. This indicates that more
iterations are needed. Iteration will be terminated if
\[ |\frac{f(\theta)}{G(\theta)} - 1| < 10^{-4}. \]

(5) Let \( r_1 = r_0/4 = 750 \). Return to step 3.

The computational results of the problem are shown in Tables 4.1,
and 4.2. Table 4.1 shows the results of starting from a feasible point
(25, 29) followed by a series of iterations which converge to the
constrained minimum \((18,000, 18.350)\). Table 4.2 shows the results of
starting from an infeasible point \((5, 10)\) followed by a series of
iterations which also converge to the constrained minimum \((18,000, 18.362)\).
The same problem has been solved by employing SUMT with RAC computer
program which uses a second order gradient method as the minimization
process [4]. The results obtained by these two different programs in
Table 4.3a (for starting at \( \theta_1 = 25, \theta_2 = 29 \)) and in Table 4.3b (for
starting at \( \theta_1 = 5, \theta_2 = 10 \)) [3]. These results are identical. It is
worth noting again that both the computer programs have self-adjusting
procedures to transfer an infeasible starting point to a reasonable
feasible starting point before proceeding to iterations [Step 2]. The
both cases of starting at two different points have required the same
amount of computing time, 1.68 minutes by RAC program and 0.6 minute
by the present program, on IBM 360/50. (Both use the WATFOR processor).

Note that in Table 4.2 the iteration makes practically no moves
since \( k = 13 \). The final stopping criterion is not satisfied at the
Table 4.1 Computer Results of Production And Inventory Problem  
[Feasible Starting Point (25,29)]

<table>
<thead>
<tr>
<th>Number of Iteration $k$</th>
<th>Value of $r$</th>
<th>Value of $\theta_1$</th>
<th>Value of $\theta_2$</th>
<th>Value of $f$</th>
<th>Value of $P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3000</td>
<td>25</td>
<td>29</td>
<td>16900</td>
<td>21040</td>
</tr>
<tr>
<td>1</td>
<td>3000</td>
<td>19.7500</td>
<td>19.0000</td>
<td>3685.00</td>
<td>6243.76</td>
</tr>
<tr>
<td>2</td>
<td>750</td>
<td>19.1875</td>
<td>19.5000</td>
<td>3325.94</td>
<td>4168.48</td>
</tr>
<tr>
<td>3</td>
<td>187.5</td>
<td>18.6875</td>
<td>18.8333</td>
<td>3100.94</td>
<td>3426.72</td>
</tr>
<tr>
<td>4</td>
<td>46.88</td>
<td>18.4063</td>
<td>18.5833</td>
<td>3024.61</td>
<td>3153.36</td>
</tr>
<tr>
<td>5</td>
<td>11.72</td>
<td>18.0239</td>
<td>18.3313</td>
<td>2968.75</td>
<td>3387.05</td>
</tr>
<tr>
<td>6</td>
<td>2.930</td>
<td>18.1475</td>
<td>18.4222</td>
<td>2980.58</td>
<td>3001.27</td>
</tr>
<tr>
<td>7</td>
<td>0.7324</td>
<td>18.0239</td>
<td>18.3414</td>
<td>2968.37</td>
<td>2999.25</td>
</tr>
<tr>
<td>8</td>
<td>0.1831</td>
<td>18.0024</td>
<td>18.3261</td>
<td>2966.84</td>
<td>3043.32</td>
</tr>
<tr>
<td>9</td>
<td>0.04578</td>
<td>18.0003</td>
<td>18.3261</td>
<td>2966.69</td>
<td>3133.37</td>
</tr>
<tr>
<td>10</td>
<td>0.01144</td>
<td>18.0127</td>
<td>18.3594</td>
<td>2967.58</td>
<td>2968.48</td>
</tr>
<tr>
<td>11</td>
<td>0.002861</td>
<td>18.9971</td>
<td>18.3551</td>
<td>2967.19</td>
<td>2967.59</td>
</tr>
<tr>
<td>12</td>
<td>0.0007153</td>
<td>18.0019</td>
<td>18.3512</td>
<td>2966.83</td>
<td>2967.20</td>
</tr>
<tr>
<td>13</td>
<td>0.0001788</td>
<td>18.0004</td>
<td>18.3500</td>
<td>2966.73</td>
<td>2967.18</td>
</tr>
<tr>
<td>14</td>
<td>0.0000447</td>
<td>18.0002</td>
<td>18.3499</td>
<td>2966.71</td>
<td>2966.90</td>
</tr>
</tbody>
</table>
Table 4.2 Computer Results of Production and Inventory Problem
[Infeasible Starting Point (5, 10)]

<table>
<thead>
<tr>
<th>Number of Iteration $k$</th>
<th>Value of $r$</th>
<th>Value of $\theta_1$</th>
<th>Value of $\theta_2$</th>
<th>Value of $f$</th>
<th>Value of $P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3000</td>
<td>5.000</td>
<td>10.000</td>
<td>33660</td>
<td>34390</td>
</tr>
<tr>
<td>1</td>
<td>3000</td>
<td>23.000</td>
<td>18.000</td>
<td>9580</td>
<td>11090</td>
</tr>
<tr>
<td>2</td>
<td>750</td>
<td>19.250</td>
<td>19.000</td>
<td>3345</td>
<td>4156.12</td>
</tr>
<tr>
<td>3</td>
<td>187.5</td>
<td>18.750</td>
<td>19.000</td>
<td>3125</td>
<td>3427.94</td>
</tr>
<tr>
<td>4</td>
<td>46.88</td>
<td>18.469</td>
<td>18.500</td>
<td>3041.5</td>
<td>3154.85</td>
</tr>
<tr>
<td>5</td>
<td>11.72</td>
<td>18.2437</td>
<td>18.5000</td>
<td>2994.0</td>
<td>3045.44</td>
</tr>
<tr>
<td>6</td>
<td>2.93</td>
<td>18.0058</td>
<td>18.3317</td>
<td>2967.06</td>
<td>3477.19</td>
</tr>
<tr>
<td>7</td>
<td>0.732</td>
<td>18.0839</td>
<td>18.4150</td>
<td>2973.65</td>
<td>2982.59</td>
</tr>
<tr>
<td>8</td>
<td>0.183</td>
<td>18.0189</td>
<td>18.3669</td>
<td>2968.05</td>
<td>2977.78</td>
</tr>
<tr>
<td>9</td>
<td>0.0458</td>
<td>18.0122</td>
<td>18.3669</td>
<td>2967.59</td>
<td>2971.35</td>
</tr>
<tr>
<td>10</td>
<td>0.01145</td>
<td>18.0065</td>
<td>18.3628</td>
<td>2967.18</td>
<td>2968.94</td>
</tr>
<tr>
<td>11</td>
<td>0.002862</td>
<td>18.0021</td>
<td>18.3628</td>
<td>2966.90</td>
<td>2968.29</td>
</tr>
<tr>
<td>12</td>
<td>0.000713</td>
<td>18.0005</td>
<td>18.3615</td>
<td>2966.79</td>
<td>2968.35</td>
</tr>
<tr>
<td>13</td>
<td>0.000178</td>
<td>18.0000</td>
<td>18.3615</td>
<td>2966.76</td>
<td>2970.67</td>
</tr>
<tr>
<td>14</td>
<td>0.0000445</td>
<td>18.0000</td>
<td>18.3615</td>
<td>2966.76</td>
<td>2968.23</td>
</tr>
<tr>
<td>15</td>
<td>0.00001112</td>
<td>18.0000</td>
<td>18.3615</td>
<td>2966.76</td>
<td>2967.49</td>
</tr>
<tr>
<td>16</td>
<td>0.00000278</td>
<td>18.0000</td>
<td>18.3615</td>
<td>2966.76</td>
<td>$\infty^*$</td>
</tr>
<tr>
<td>17</td>
<td>0.000000642</td>
<td>18.0000</td>
<td>18.3615</td>
<td>2966.76</td>
<td>2966.76</td>
</tr>
</tbody>
</table>

* On the boundary.
Table 4.3a. Comparison of the Optimal Solutions of the Production Scheduling and Inventory Control Problem [Feasible starting point (25, 29)]

<table>
<thead>
<tr>
<th>Program</th>
<th>Number of $k$ iterated</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>Cost</th>
<th>Stopping criteria</th>
<th>Computing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAC</td>
<td>0</td>
<td>25</td>
<td>29</td>
<td>16900</td>
<td>$\epsilon = 10^{-4}$</td>
<td>1.68 min.</td>
</tr>
<tr>
<td>Program</td>
<td>12</td>
<td>18.003</td>
<td>18.335</td>
<td>2966.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New</td>
<td>0</td>
<td>25</td>
<td>29</td>
<td>16900</td>
<td>INCUT = 3</td>
<td>0.6 min.</td>
</tr>
<tr>
<td>Program</td>
<td>14</td>
<td>18.0002</td>
<td>18.3499</td>
<td>2966.71</td>
<td>$\epsilon = 10^{-4}$</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.3b. Comparison of the Optimal Solution of the Production Scheduling and Inventory Control Problem

<table>
<thead>
<tr>
<th>Program</th>
<th>Number of (k) iterated</th>
<th>(\theta_1)</th>
<th>(\theta_2)</th>
<th>Cost</th>
<th>Stopping criteria for each (k)</th>
<th>Stopping criteria for final (\epsilon)</th>
<th>Computing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAC</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>33660</td>
<td>(\epsilon = 10^{-4})</td>
<td></td>
<td>1.68 min.</td>
</tr>
<tr>
<td>(feasible starting point)</td>
<td>20.587</td>
<td>15.732</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>12</td>
<td>18.003</td>
<td>18.335</td>
<td>2966.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>33660</td>
<td>(\epsilon = 10^{-4})</td>
<td>INCUT = 3</td>
<td>0.6 min.</td>
</tr>
<tr>
<td>(feasible starting point)</td>
<td>23</td>
<td>18</td>
<td>9580</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>17</td>
<td>18.0000</td>
<td>18.3615</td>
<td>2966.76</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
end of the 13th iteration. The value of $r_k$ is reduced and the iteration
goes to $k = 14$. Because the value of $r_k$ is decreasing as $k$ increasing,
the dual comparison term used in the final stopping criterion is de-
creasing. The final stopping criterion is finally satisfied at $k = 17$.

Note that at $k = 16$, the value of $P$ function is $+\infty$ (for avoiding
this critical situation which essentially will cause overflow in
computation, a large finite number (,10^{49},) is used to replace $+\infty$).
The reason for this is that the sub-optimum point is right on the bounding
of an inequality constraint; thus the value of the $P$ function becomes
infinity. Recall that the $P$ function is defined as

$$P(x, r_k) = f(x) + r_k \sum_i \frac{1}{g_i(x)}$$

Figure 1 shows the locus of convergence for both the case for
feasible starting point and the case for feasible starting point and the
case for infeasible starting point.

**NUMERICAL EXAMPLE 2**

For demonstrating the solution to a problem involving the equality
constraints, the above numerical example is modified by adding an equality
constraint, namely,

$$h(\theta) = \theta_1 - \theta_2 - 5 = 0$$

This implies that the production level in the first period is five unit
larger than that in the second period.

The problem is restated as follows:
Minimize

\[ f(\theta) = C(\theta_1 - \theta_0)^2 + D(E - I_1)^2 + C(\theta_2 - \theta_1)^2 + D(E - I_2)^2 \]

subject to

\[ g_1(\theta) = I_0 + \theta_1 - Q_1 \geq 0 \]
\[ g_2(\theta) = I_1 + \theta_2 - Q_2 \geq 0 \]
\[ g_3(\theta) = M - \theta_1 \geq 0 \]
\[ g_4(\theta) = M - \theta_2 \geq 0 \]
\[ h(\theta) = \theta_1 - \theta_2 - 5 = 0 \]

With the same numerical values given in numerical example 1, the solutions obtained are presented in Tables 4.3c and 4.3d.

During the early iterations, say, from \( k = 1 \) to \( k = 3 \) or 4, the equality constraint does not play any significant effect to the searches. However, as \( k \) increased, the value of \( r_k \) approaches to a small numbers, the penalty of violation to the equality constraint becomes significant. The search after \( k = 4 \) or 5 in both Table 4.3c and Table 4.3d, as one can see that, the equality constraint is forced to approach to zero. Recall that the formulation of the P-function with equality constraints is defined as

\[ P(x, r_k) = f(x) + r_k \sum \frac{1}{g_i(x)} + r_k \sum \frac{1}{2} h_j^2(x) \]

As \( r_k \to 0 \), the penalty to equality constraints \( h_j \)'s, \( r_k \sum \frac{1}{2} h_j^2(x) \), becomes very large. When minimizing the P-function, all the \( h_j \)'s will be forced to approach to zero.
Table 4.3c  Computer Result of the 2-Dimensional Problem with a Equality Constraint [Start at (25, 29)].

<table>
<thead>
<tr>
<th>Number of Iteration $k$</th>
<th>Value of $\tau_k$</th>
<th>Value of $\theta_1$</th>
<th>Value of $\theta_2$</th>
<th>Value of $f(\theta)$</th>
<th>Value of $P(\theta)$</th>
<th>Value of $g_1(\theta)$</th>
<th>Value of $g_2(\theta)$</th>
<th>Value of $g_3(\theta)$</th>
<th>Value of $g_4(\theta)$</th>
<th>Value of $h(\theta)$</th>
<th>Value of $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>51.29</td>
<td>25</td>
<td>29</td>
<td>16,900</td>
<td>16,980</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>51.29</td>
<td>18.4375</td>
<td>18.5586</td>
<td>3,032.09</td>
<td>3,167.60</td>
<td>0.4375</td>
<td>8.9961</td>
<td>11.5625</td>
<td>11.4414</td>
<td>-5.121</td>
<td>0.04414</td>
</tr>
<tr>
<td>2</td>
<td>12.82</td>
<td>18.2969</td>
<td>18.5586</td>
<td>3,003.00</td>
<td>3,057.58</td>
<td>0.2969</td>
<td>8.8554</td>
<td>11.7031</td>
<td>11.4414</td>
<td>-5.262</td>
<td>0.01320</td>
</tr>
<tr>
<td>3</td>
<td>3.205</td>
<td>18.1621</td>
<td>18.4140</td>
<td>2,982.47</td>
<td>3,018.57</td>
<td>0.1621</td>
<td>8.5762</td>
<td>11.838</td>
<td>11.586</td>
<td>-5.252</td>
<td>0.00178</td>
</tr>
<tr>
<td>4</td>
<td>0.8013</td>
<td>18.0918</td>
<td>18.3437</td>
<td>2,974.67</td>
<td>3,014.44</td>
<td>0.0918</td>
<td>8.436</td>
<td>11.908</td>
<td>11.656</td>
<td>-5.252</td>
<td>0.00729</td>
</tr>
<tr>
<td>5</td>
<td>0.2003</td>
<td>18.0503</td>
<td>18.0503</td>
<td>2,971.60</td>
<td>3,036.53</td>
<td>0.0503</td>
<td>8.3211</td>
<td>11.950</td>
<td>11.729</td>
<td>-5.220</td>
<td>0.0188</td>
</tr>
<tr>
<td>6</td>
<td>0.05008</td>
<td>18.0503</td>
<td>18.2708</td>
<td>2,974.82</td>
<td>3,093.27</td>
<td>0.0503</td>
<td>8.227</td>
<td>11.950</td>
<td>11.823</td>
<td>-5.127</td>
<td>0.03767</td>
</tr>
<tr>
<td>7</td>
<td>0.00626</td>
<td>18.0503</td>
<td>18.2708</td>
<td>2,987.98</td>
<td>3,295.88</td>
<td>0.0503</td>
<td>8.035</td>
<td>11.950</td>
<td>12.015</td>
<td>-4.935</td>
<td>0.09335</td>
</tr>
<tr>
<td>8</td>
<td>0.001138</td>
<td>18.0472</td>
<td>17.3114</td>
<td>3,103.38</td>
<td>3,642.37</td>
<td>0.0472</td>
<td>7.359</td>
<td>11.953</td>
<td>12.689</td>
<td>-4.264</td>
<td>0.1480</td>
</tr>
<tr>
<td>9</td>
<td>0.000207</td>
<td>18.235</td>
<td>16.561</td>
<td>3,438.83</td>
<td>4,208.13</td>
<td>0.2347</td>
<td>6.796</td>
<td>11.765</td>
<td>13.439</td>
<td>-3.327</td>
<td>0.1828</td>
</tr>
<tr>
<td>10</td>
<td>0.000038</td>
<td>18.462</td>
<td>15.66</td>
<td>4,103.98</td>
<td>4,891.48</td>
<td>0.4621</td>
<td>6.122</td>
<td>11.538</td>
<td>14.340</td>
<td>-2.198</td>
<td>0.1610</td>
</tr>
<tr>
<td>11</td>
<td>0.0000068</td>
<td>18.658</td>
<td>14.879</td>
<td>4,910.26</td>
<td>5,479.97</td>
<td>0.65797</td>
<td>5.537</td>
<td>11.342</td>
<td>15.121</td>
<td>-1.221</td>
<td>0.1040</td>
</tr>
<tr>
<td>12</td>
<td>0.0000012</td>
<td>18.787</td>
<td>14.387</td>
<td>5,533.93</td>
<td>5,856.30</td>
<td>0.7869</td>
<td>5.173</td>
<td>11.213</td>
<td>15.614</td>
<td>-0.5996</td>
<td>0.05505</td>
</tr>
<tr>
<td>13</td>
<td>0.00000023</td>
<td>18.8695</td>
<td>14.143</td>
<td>5,896.71</td>
<td>6,053.39</td>
<td>0.8695</td>
<td>5.012</td>
<td>11.131</td>
<td>15.858</td>
<td>-0.2730</td>
<td>0.0259</td>
</tr>
<tr>
<td>14</td>
<td>0.00000004</td>
<td>18.990</td>
<td>14.089</td>
<td>6,081.04</td>
<td>6,148.27</td>
<td>0.990</td>
<td>5.097</td>
<td>11.010</td>
<td>15.893</td>
<td>-0.117</td>
<td>0.0109</td>
</tr>
<tr>
<td>15</td>
<td>0.000000008</td>
<td>19.004</td>
<td>14.055</td>
<td>6,159.13</td>
<td>6,189.37</td>
<td>1.004</td>
<td>5.0595</td>
<td>10.996</td>
<td>15.945</td>
<td>-0.05114</td>
<td>0.00208</td>
</tr>
<tr>
<td>16</td>
<td>$1.4 \times 10^{-9}$</td>
<td>19.028</td>
<td>14.048</td>
<td>6,195.14</td>
<td>6,208.08</td>
<td>1.028</td>
<td>5.077</td>
<td>10.972</td>
<td>15.951</td>
<td>-0.0218</td>
<td>0.00208</td>
</tr>
<tr>
<td>17</td>
<td>$2.5 \times 10^{-10}$</td>
<td>19.033</td>
<td>14.043</td>
<td>6,210.13</td>
<td>6,215.95</td>
<td>1.033</td>
<td>5.076</td>
<td>10.967</td>
<td>15.957</td>
<td>-0.0095</td>
<td>0.000932</td>
</tr>
</tbody>
</table>
Table 4.3d  Computer Result of the 2-Dimensional Problem with a Equality Constraint [Start at (5, 10)].

<table>
<thead>
<tr>
<th>Number of Iteration $k$</th>
<th>Value of $r_k$</th>
<th>Value of $\theta_1$</th>
<th>Value of $\theta_2$</th>
<th>Value of $f(\theta)$</th>
<th>Value of $P(\theta)$</th>
<th>Value of $g_1(\theta)$</th>
<th>Value of $g_2(\theta)$</th>
<th>Value of $g_3(\theta)$</th>
<th>Value of $g_4(\theta)$</th>
<th>Value of $h(\theta)$</th>
<th>Computed $\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>83.95</td>
<td>5</td>
<td>10</td>
<td>33660</td>
<td>33690</td>
<td>0.5000</td>
<td>9.0625</td>
<td>11.5000</td>
<td>11.4375</td>
<td>-5.0625</td>
<td>0.0661</td>
</tr>
<tr>
<td>1</td>
<td>83.95</td>
<td>23</td>
<td>10</td>
<td>24300</td>
<td>24360</td>
<td>0.3224</td>
<td>8.8867</td>
<td>11.6758</td>
<td>11.4375</td>
<td>-5.2383</td>
<td>0.02199</td>
</tr>
<tr>
<td>2</td>
<td>20.99</td>
<td>18.324</td>
<td>18.563</td>
<td>3007.92</td>
<td>3084.64</td>
<td>0.1895</td>
<td>8.6426</td>
<td>11.8105</td>
<td>11.5469</td>
<td>-5.264</td>
<td>0.00576</td>
</tr>
<tr>
<td>3</td>
<td>5.247</td>
<td>18.190</td>
<td>18.453</td>
<td>2980.44</td>
<td>3027.30</td>
<td>0.1104</td>
<td>8.4936</td>
<td>11.8896</td>
<td>11.6172</td>
<td>-5.272</td>
<td>0.00402</td>
</tr>
<tr>
<td>4</td>
<td>1.312</td>
<td>18.110</td>
<td>18.383</td>
<td>2976.37</td>
<td>3012.90</td>
<td>0.0630</td>
<td>8.361</td>
<td>11.937</td>
<td>11.702</td>
<td>-5.235</td>
<td>0.0141</td>
</tr>
<tr>
<td>5</td>
<td>0.3279</td>
<td>18.063</td>
<td>18.298</td>
<td>2972.32</td>
<td>3025.48</td>
<td>0.0347</td>
<td>8.241</td>
<td>11.965</td>
<td>11.793</td>
<td>-5.172</td>
<td>0.00797</td>
</tr>
<tr>
<td>6</td>
<td>0.08198</td>
<td>18.035</td>
<td>18.207</td>
<td>2971.90</td>
<td>3067.71</td>
<td>0.0346</td>
<td>8.081</td>
<td>11.965</td>
<td>11.954</td>
<td>-5.011</td>
<td>0.07675</td>
</tr>
<tr>
<td>7</td>
<td>0.0103</td>
<td>18.035</td>
<td>18.207</td>
<td>2980.76</td>
<td>3229.15</td>
<td>0.0193</td>
<td>7.504</td>
<td>11.981</td>
<td>12.516</td>
<td>-4.465</td>
<td>0.1312</td>
</tr>
<tr>
<td>8</td>
<td>0.00186</td>
<td>18.019</td>
<td>17.485</td>
<td>5057.14</td>
<td>3519.13</td>
<td>0.1734</td>
<td>6.9773</td>
<td>11.827</td>
<td>13.196</td>
<td>-3.631</td>
<td>0.1779</td>
</tr>
<tr>
<td>9</td>
<td>0.000339</td>
<td>18.173</td>
<td>16.804</td>
<td>3308.58</td>
<td>4024.71</td>
<td>0.39365</td>
<td>6.30998</td>
<td>11.606</td>
<td>14.084</td>
<td>-2.523</td>
<td>0.17243</td>
</tr>
<tr>
<td>10</td>
<td>0.0000616</td>
<td>18.394</td>
<td>15.916</td>
<td>3883.35</td>
<td>4694.25</td>
<td>0.6045</td>
<td>5.6799</td>
<td>11.3955</td>
<td>14.9246</td>
<td>-1.4709</td>
<td>0.1213</td>
</tr>
<tr>
<td>11</td>
<td>0.0000112</td>
<td>18.605</td>
<td>15.075</td>
<td>4683.45</td>
<td>5329.99</td>
<td>0.7490</td>
<td>5.2411</td>
<td>11.2510</td>
<td>15.5079</td>
<td>-0.7431</td>
<td>0.0671</td>
</tr>
<tr>
<td>12</td>
<td>0.00000204</td>
<td>18.749</td>
<td>14.492</td>
<td>5382.23</td>
<td>5769.16</td>
<td>0.8998</td>
<td>5.1485</td>
<td>11.000</td>
<td>15.851</td>
<td>-0.149</td>
<td>0.0139</td>
</tr>
<tr>
<td>13</td>
<td>0.00000037</td>
<td>18.839</td>
<td>14.184</td>
<td>5814.27</td>
<td>6010.20</td>
<td>0.9998</td>
<td>5.0235</td>
<td>11.1609</td>
<td>15.8156</td>
<td>-0.3453</td>
<td>0.0326</td>
</tr>
<tr>
<td>14</td>
<td>0.000000067</td>
<td>18.9998</td>
<td>14.131</td>
<td>6043.99</td>
<td>6129.43</td>
<td>1.0498</td>
<td>5.1639</td>
<td>10.950</td>
<td>15.886</td>
<td>-0.0644</td>
<td>0.00966</td>
</tr>
<tr>
<td>15</td>
<td>0.000000012</td>
<td>19.0498</td>
<td>14.106</td>
<td>6145.96</td>
<td>6183.45</td>
<td>1.059</td>
<td>5.145</td>
<td>10.942</td>
<td>15.914</td>
<td>-0.0277</td>
<td>0.00263</td>
</tr>
<tr>
<td>16</td>
<td>0.0000000022</td>
<td>19.0585</td>
<td>14.082</td>
<td>6189.96</td>
<td>6206.27</td>
<td>1.055</td>
<td>5.122</td>
<td>10.945</td>
<td>15.933</td>
<td>-0.01196</td>
<td>0.00114</td>
</tr>
<tr>
<td>17</td>
<td>0.0000000004</td>
<td>19.0450</td>
<td>14.067</td>
<td>6208.52</td>
<td>6215.63</td>
<td>1.059</td>
<td>5.125</td>
<td>10.940</td>
<td>15.935</td>
<td>-0.00514</td>
<td>0.000496</td>
</tr>
</tbody>
</table>
Fig. 4-1a. Production scheduling problem involving two decision variables; contour lines indicate equal quantities of total cost given by equation (4.3).
The optimum of this numerical example is \((\theta_1, \theta_2) = (19.03, 14.04)\).
The route of the iterations start from two different starting points, 
\((25, 29)\) and \((5, 10)\) is shown in Fig. 4-1a. Note that the selected 
feasible starting point searched from the infeasible initial point, 
\((5, 10)\), are different for these two numerical examples, one with ine-
quality constraints only, and the other, with inequality constraints 
and equality constraints.

4.3 A PERSONNEL AND PRODUCTION SCHEDULING PROBLEM

To demonstrate the capability and practical nature of the method, it
is employed to obtain the solution of a problem based on the well known
model of Holt, Modigliani, Muth and Simon [2]. The problem is to find
the optimal operation cost in a paint factory by considering the monthly
production and work force level as decision variables in four different
sub-costs, namely, the cost of regular payroll, the cost of hiring and
firing, the cost of overtime, and the inventory cost. The schematic
diagram of the system is shown in Fig. 2. The problem is to minimize the
sum of all four different costs over a planning period subject to the
constraints of non-negative inventory and non-negative overtime cost.
(The main reasons of considering non-negative overtime cost will be dis-
cussed later.) The demand of each period is known in advance and must
be satisfied.

Let

\[ n = \text{a month in the planning horizon} \]
\[ N = \text{the duration, in months} \]
\[ P_n = \text{production rate at the } n\text{-th month} \]
\[ W_n = \text{work force level in the } n\text{-th month} \]
Fig. 2. Block diagram for personnel and production scheduling.
Q_n = demand at the n-th month

L_n = inventory level at the end of the n-th month

Inventory level at the end of each month is represented by the recursive relationship among current demand, current production and inventory level of the preceding month.

\[ I_n = I_{n-1} + P_n - Q_n, \quad n = 1, 2, ..., N \]

The model considers the following monthly operation cost.

1. Regular payroll cost = 340.0 \( W_n \)
2. Hiring and lay off cost = 64.3 \( (W_n - W_{n-1})^2 \)
3. Overtime cost = 0.2 \( (P_n - 5.67 W_n)^2 + 51.2 P_n - 281.0 W_n \)
4. Inventory cost = 0.0825 \( (I_n - 320.0)^2 \)

The system can then be represented by the following mathematical model.

Minimize

\[ f(P_1, P_2, \ldots, P_N; W_1, W_2, \ldots, W_N) = \sum_{n=1}^{N} S_n \]

subject to

\[ I_n = I_{n-1} + P_n - Q_n \geq 0, \quad n = 1, 2, \ldots, N-1 \]

\[ I_N = I_{N-1} + P_N - Q_N \geq I_f \]

and

\[ 0.2(P_n - 5.67 W_n)^2 + 51.2 P_n - 281.0 W_n \geq 0, \quad n = 1, 2, \ldots, N \]

where

\[ S_n = [340.0 W_n^1 + [64.3(W_n - W_{n-1})^2] \]

\[ + [0.2(P_n - 5.67 W_n)^2 + 51.2 P_n - 281.0 W_n] \]

\[ + [0.0825(I_n - 320.0)^2] \]
The reason of considering the non-negative overtime cost is due to the characteristics of its mathematical formula. Taubert [5] found that minimizing the total costs over the planning period by selecting a certain \( W_n \) and \( P_n \) combination contributed a negative overtime cost to the objective function. Since the negative cost is illogical in the context of the original paint factory example, a careful examination has been carried out by investigating the overtime cost equation.

\[
\text{Overtime cost} = 0.2 (P_n - 5.67W_n)^2 + 51.2P_n - 281.0W_n
\]

The equation is in quadratic form and can have negative cost contours as plotted in Fig. 3 [3, 5]. Therefore, the constraint of the non-negative overtime cost should be imposed.

**A NUMERICAL EXAMPLE**

To demonstrate the technique, a numerical example of the model with ten stages is studied.

The numerical data used are as follows:

Demand:

\[
\begin{align*}
Q_1 &= 430, & Q_6 &= 375, \\
Q_2 &= 447, & Q_7 &= 292, \\
Q_3 &= 440, & Q_8 &= 458, \\
Q_4 &= 316, & Q_9 &= 400, \\
Q_5 &= 397, & Q_{10} &= 350.
\end{align*}
\]

The initial inventory, \( I_0 = 263 \), the inventory for the last month, \( I_{-1} = 263 \) and the initial work force level \( W_0 = 81 \).

The starting point is chosen arbitrarily at \( x^0 = (P_1^0, W_1^0, P_2^0, W_2^0, P_3^0, W_3^0, P_4^0, W_4^0, P_5^0, W_5^0, P_6^0, W_6^0, P_7^0, W_7^0, P_8^0, W_8^0, P_9^0, W_9^0, P_{10}^0, W_{10}^0) = (500, 90, 500, 90, 500, 90, 500, 90, 500, 90, 500, 90, 500, 90, 500, 90) \). The final result is obtained in 9 iteration (\( k = 9 \)) when the stopping
criterion is $c = 10^{-5}$ and the starting penalty coefficient is $r_0 = 3.352 \times 10^6$. The value of $r_0$ is computed by the formula

$$r_0 = \left(\frac{1}{4}\right) \cdot \left\{ \frac{f(x^0)}{\sum \frac{1}{g_i(x^0)}} \right\}.$$  

The results are presented in Table 4.4.

Table 4.5 lists the optimal results obtained by employing the RAC computer program and the present program for comparison. The results obtained by both methods are almost identical.

The computing time is 15.12 minutes for the RAC program and is 8 minutes for the present program; on IBM 360/50 (use the WATFOR processor).

4.4 DISCUSSION AND CONCLUDING REMARKS

The developed technique is a workable technique; and because of its simplicity it can be applied to a wide range of practical problems. The important advantages of this technique over the original available RAC technique are that the new technique does not need to evaluate any derivatives, and requires less computing time.

There is a disadvantage that exploratory moves with small step sizes in the Hooke and Jeaves pattern search may produce the values of $P$-functional identical in all significant digits for a large numerical value problem. A double precision specification which specifies more significant digits in a computer may be able to overcome this disadvantage.
| Number of Iteration $k$ | Value of $r$ | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | $p_{10}$ | $f(p_n;w_n)$ | $P(p_n;w_n)$ |
|-------------------------|-------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|----------------|----------------|
| 0                       |             | 500   | 500   | 500   | 500   | 500   | 500   | 500   | 500   | 500   | 500   | 613,200        | 766.500        |
| 1                       | 3.352x10^6  | 467.0 | 415.0 | 415.0 | 415.0 | 415.0 | 415.0 | 415.0 | 471.0 | 527.0 | 527.0 | 303,020        | 439,884        |
| 2                       | 8.381x10^5  | 487.3 | 451.0 | 420.5 | 392.0 | 381.0 | 372.5 | 367.5 | 383.0 | 383.0 | 378.5 | 252,845        | 285,509        |
| 3                       | 2.095x10^5  | 473.5 | 443.5 | 416.0 | 389.0 | 379.5 | 371.0 | 365.5 | 381.5 | 378.5 | 368.5 | 247,745        | 258,539        |
| 4                       | 5.238x10^4  | 469.0 | 442.3 | 415.3 | 387.5 | 378.5 | 370.0 | 362.8 | 379.8 | 373.0 | 359.5 | 245,882        | 249,436        |
| 5                       | 1.31x10^4   | 468.8 | 442.3 | 415.4 | 385.3 | 377.6 | 368.4 | 361.3 | 380.5 | 370.0 | 352.8 | 245,076        | 246,364        |
| 6                       | 3.274x10^3  | 472.3 | 441.6 | 414.6 | 381.9 | 376.6 | 367.1 | 359.8 | 389.5 | 369.0 | 350.0 | 244,716        | 245,218        |
| 7                       | 8.185x10^2  | 471.5 | 443.9 | 417.1 | 383.1 | 375.0 | 366.4 | 357.8 | 379.0 | 368.3 | 347.6 | 244,525        | 244,747        |
| 8                       | 2.046x10^2  | 469.4 | 444.1 | 417.2 | 383.3 | 376.1 | 367.0 | 357.6 | 378.3 | 367.8 | 346.6 | 244,312        | 244,532        |
| 9                       | 51.15        | 468.8 | 443.8 | 416.9 | 383.1 | 376.1 | 367.0 | 357.6 | 378.3 | 367.8 | 346.6 | 244,375        | 244,438        |
Table 4.5. Comparison of the Optimal Solution of the Personnel and Production Planning Problem

<table>
<thead>
<tr>
<th>Month</th>
<th>RAC Program</th>
<th>New Program</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P_n$</td>
<td>$W_n$</td>
</tr>
<tr>
<td>0</td>
<td>81.0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>468.6</td>
<td>77.8</td>
</tr>
<tr>
<td>2</td>
<td>443.0</td>
<td>74.7</td>
</tr>
<tr>
<td>3</td>
<td>416.4</td>
<td>71.6</td>
</tr>
<tr>
<td>4</td>
<td>382.2</td>
<td>68.8</td>
</tr>
<tr>
<td>5</td>
<td>377.7</td>
<td>66.6</td>
</tr>
<tr>
<td>6</td>
<td>368.3</td>
<td>64.9</td>
</tr>
<tr>
<td>7</td>
<td>358.8</td>
<td>63.6</td>
</tr>
<tr>
<td>8</td>
<td>379.4</td>
<td>62.9</td>
</tr>
<tr>
<td>9</td>
<td>368.0</td>
<td>62.1</td>
</tr>
<tr>
<td>10</td>
<td>344.8</td>
<td>61.6</td>
</tr>
</tbody>
</table>

Total cost = 244,336  
Total cost = 244,375
REFERENCE


CHAPTER 5

OPTIMIZATION OF A COMPLEX SYSTEM RELIABILITY

5.1 INTRODUCTION

In this chapter the reliability of a complex system studied in Chapter 2 is again investigated. Two optimization problems associated with this system are considered and the results are compared to the results obtained in Chapter 2.

The first problem is the maximization of system reliability which is identical to the problem studied in Chapter 2. The problem is to find the optimal component reliabilities for the components of the complex system shown in Fig. 1 in Chapter 2. The system reliability is maximized subject to a nonlinear cost function. The second problem is to minimize the cost of the system. In this problem, constraints of a minimal required system reliability and a minimum component reliability for each component must be satisfied.

The method used to solve the above two problems in this chapter is the method developed in Chapter 3.

The purposes of this chapter are: (1) to demonstrate the usefulness of the method developed in Chapter 3, particularly to the system reliability optimization problems, and (2) to compare the capacity and efficiency of this method with that of the method used in the RAC program.

The optimal solution of the cost minimization problem cannot be obtained by the RAC program, however, an optimal solution can be obtained by employing the new method. The reasons of the particular difficulty in system reliability optimization problem is explained in section 5.5. For the problem of maximizing the system reliability subject to cost constraint,
the same results are obtained by the RAC program and by the new method. The computer time requirement, however, has substantial difference. The RAC program requires over 20 minutes on an IBM 360/50 computer and the new method it requires less than one minute (55 seconds) on the same computer.

5.2 FORMULATIONS OF TWO SYSTEM RELIABILITY PROBLEMS

For the convenience the complex system reliability problem in chapter 2 is briefly summarized below. The diagram which shows the configuration of this system is presented in Fig 1 in chapter 2. In the system, unit 1 is backed up in a parallel by unit 4. There are two equal paths, where each path has unit 2 in series with the stage formed by units 1 and 4. These two equal paths operate in parallel so that if at least one of them is good the output is assured. However, because unit 2 does not have a high degree of reliability, a third unit, unit 3, is inserted into the circuit. Therefore, the following operations are possible: 2-1, 2-4, 3-1, and 3-4, and each operation has two equal paths.

By applying Bayes' theorem of conditional probability, the following expression of the reliability of this system has been derived (see chapter 2).

\[
R_s = 1 - Q_s
\]

where

\[
Q_s = \left[ (1-R_1)(1-R_4) \right]^2 R_3 + \{1-R_2[1-(1-R_1)(1-R_4)]\}^2 (1-R_3)
\]

(5.1)

The two optimization problems studied with this system can be summarized as follows:
The first problem is to find the optimal component reliability, \( R_i \), which maximize the system reliability, \( R_s \), subject to a maximal cost functional constraint. It can be restated as:

Maximize

\[
R_s = 1 - Q_s
= 1 - R_3 \left( 1 - R_1 \right) \left( 1 - R_4 \right)^2
- \left( 1 - R_3 \right) \left( 1 - R_2 \right) \left( 1 - (1 - R_1) (1 - R_4) \right)^2
\]

subject to

\[
\sum_{i} C_i \leq C
\]

where

\[
C_i = K_i R_i^{\alpha_i}, \text{ } K_i \text{ and } \alpha_i \text{ are constants}
\]

The second problem is to find the optimal component reliability, \( R_i \), which minimize the cost function, \( C \). It can be restated as:

Minimize

\[
C = \sum_{i} C_i
\]

where

\[
C_i = K_i R_i^{\alpha_i}, \text{ } K_i \text{ and } \alpha_i \text{ are constants}
\]

subject to a minimal system reliability constraint

\[
R_s \geq R_{\min}
\]

where \( R_{\min} \) is a constant minimal system reliability required; and the system reliability, \( R_s \), is given by Equation (5.1).

The cost function \( C_i \) in equations (5.2) and (5.3) can represent the weight, cost, or volume of each component of the system, and the
summation of $C_i$ then represent the total weight, the total cost, or the total volume of the system. The weight, cost, or volume of each unit or component of the system is a function of reliability which can be expressed by equations (5.2) and (5.3), where $K_i$ is a proportionality constant and $\alpha_i$, the exponential factor, relates $C_i$ and the reliability. Usually $\alpha_i$ is less than one.

5.3. THE PROBLEM OF MAXIMIZING SYSTEM RELIABILITY

The numerical example solved in chapter 2 is resolved by the new developed technique. The problem is to find the optimal $R_1$ which maximize

$$R_s = 1 - R_3[(1-R_1)(1-R_4)]^2$$

$$- (1-R_3)[1 - R_2[1 - (1-R_1)(1-R_4)]]^2$$

subject to the constraint

$$2K_1^{\alpha_1} + 2K_2^{\alpha_2} + K_3^{\alpha_3} + 2K_4^{\alpha_4} \leq C.$$  

(5.5)

The constants $K_1$, $K_2$, $K_3$, and $K_4$, the constraint, $C$, and the exponential constant $\alpha_i$, $i = 1, 2, 3, 4$, are follows:

$$K_1 = 100, \quad K_2 = 100, \quad K_3 = 200, \quad K_4 = 150,$$

$$C = 800, \quad \alpha_i = 0.6, \quad i = 1, 2, 3, 4.$$

The problem is formulated in SUMT format as follows:

Minimize

$$f(x) = -R_s$$

$$= -1 + R_3[(1-R_1)(1-R_4)]^2 + (1-R_3)[1 - R_2[1 - (1-R_1)(1-R_4)]]^2$$

subject to the constraints

$$g_1(x) = C - (2K_1^{\alpha_1} + 2K_2^{\alpha_2} + K_3^{\alpha_3} + K_4^{\alpha_4}) \geq 0$$

(5.8)
\[ g_{i+1}(x) = 1 - R_i \geq 0, \quad i = 1, 2, 3, 4 \]  

(5.9)

The P function for this problem is

\[
P(x, r_k) = f(x) + r_k \sum_{i} 1/g_i(x)
\]

\[
= -1 + R_3[(1-R_1)(1-R_4)]^2 + (1-R_3)(1 - R_2[1 - (1-R_1)(1-R_4)])^2
\]

\[
+ r_k \left( \frac{1}{c - (2K_1R_1 \alpha_1 + 2K_2R_2 \alpha_2 + K_3R_3 \alpha_3 + K_4R_4 \alpha_4)} + \sum_{i=1}^{4} \frac{1}{(1-R_i)} \right)
\]

(5.10)

where \( x \) is the row vector of \((R_1, R_2, R_3, R_4)\).

The optimal solutions obtained from two sets of different starting components reliabilities, namely, \([R_1, R_2, R_3, R_4] = [0.7, 0.7, 0.7, 0.7]\) and \([R_1, R_2, R_3, R_4] = [0.6, 0.6, 0.6, 0.6]\), are presented in Table 5.1 together with the corresponding results obtained in Chapter 2. The solutions are almost identical, that is, the optimal system reliability, \( R_s \), of 0.999998 with the cost of 799.733 for the first set of starting components reliabilities, and the optimal system reliability, \( R_s \), of 0.999997 with the cost of 799.908 for the second set of starting components reliabilities are obtained. Recall that the constraint on the cost is 800. The optimal components reliabilities are almost the same for the both starting sets of the starting points. The stopping criterion for terminating the minimization of the P function at each \( k \) iteration is that terminating when the number of cut-down step-size operations in the Hooke and Jeeves pattern search is 3, and the final stopping criterion for terminating the problem is \( \varepsilon = 10^{-4} \). For the
THE FOLLOWING DOCUMENTS ARE BEING FILMED IN SECTIONS.

THE FOLLOWING IMAGES WILL BE TAKEN FROM LEFT TO RIGHT, TOP TO BOTTOM. SEE EXAMPLE BELOW:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Program</td>
<td>Number of $k$ iterated</td>
<td>$R_1$</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>RAC</td>
<td>0</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.9876</td>
</tr>
<tr>
<td>Program</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>0.9869</td>
</tr>
<tr>
<td>New</td>
<td>0</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.997626</td>
</tr>
<tr>
<td>Program</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>0.997409</td>
</tr>
</tbody>
</table>

Table 5.1. Comparison of the Optimal Solutions
of the System Reliability Maximization Problem

<table>
<thead>
<tr>
<th>System Reliability $R$</th>
<th>Cost</th>
<th>Stopping criteria for each $k$</th>
<th>Stopping criteria for final $\varepsilon$</th>
<th>Computing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99996</td>
<td>799.78</td>
<td>$\varepsilon' = 10^{-5}$</td>
<td>$\varepsilon = 10^{-4}$</td>
<td>exceeds 20 min.</td>
</tr>
<tr>
<td>0.99995</td>
<td>799.28</td>
<td>$\varepsilon' = 10^{-5}$</td>
<td>$\varepsilon = 10^{-4}$</td>
<td>exceeds 20 min.</td>
</tr>
<tr>
<td>0.99998</td>
<td>799.733</td>
<td>INCUT = 3</td>
<td>$\varepsilon = 10^{-5}$</td>
<td>(both problems together)</td>
</tr>
<tr>
<td>0.99997</td>
<td>799.908</td>
<td>INCUT = 3</td>
<td>$\varepsilon = 10^{-5}$</td>
<td>90.4 sec.</td>
</tr>
<tr>
<td>Iteration</td>
<td>Times of $f$-value calculated at each iteration</td>
<td>Value of $r_k$</td>
<td>$R_1$</td>
<td>$R_2$</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------</td>
<td>----------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>$2.214 \times 10^{-2}$</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>1</td>
<td>70</td>
<td>$2.214 \times 10^{-2}$</td>
<td>0.6200</td>
<td>0.7150</td>
</tr>
<tr>
<td>2</td>
<td>68</td>
<td>$5.535 \times 10^{-3}$</td>
<td>0.7900</td>
<td>0.7900</td>
</tr>
<tr>
<td>3</td>
<td>59</td>
<td>$1.384 \times 10^{-3}$</td>
<td>0.8700</td>
<td>0.8700</td>
</tr>
<tr>
<td>4</td>
<td>89</td>
<td>$3.459 \times 10^{-4}$</td>
<td>0.872499</td>
<td>0.91125</td>
</tr>
<tr>
<td>5</td>
<td>72</td>
<td>$8.648 \times 10^{-5}$</td>
<td>0.964999</td>
<td>0.94274</td>
</tr>
<tr>
<td>6</td>
<td>174</td>
<td>$2.162 \times 10^{-5}$</td>
<td>0.944907</td>
<td>0.96815</td>
</tr>
<tr>
<td>7</td>
<td>202</td>
<td>$5.405 \times 10^{-6}$</td>
<td>0.973031</td>
<td>0.95076</td>
</tr>
<tr>
<td>8</td>
<td>129</td>
<td>$1.351 \times 10^{-6}$</td>
<td>0.983415</td>
<td>0.98817</td>
</tr>
<tr>
<td>9</td>
<td>110</td>
<td>$3.378 \times 10^{-7}$</td>
<td>0.989665</td>
<td>0.99262</td>
</tr>
<tr>
<td>10</td>
<td>85</td>
<td>$8.446 \times 10^{-8}$</td>
<td>0.993554</td>
<td>0.99550</td>
</tr>
<tr>
<td>11</td>
<td>76</td>
<td>$2.111 \times 10^{-8}$</td>
<td>0.996045</td>
<td>0.99720</td>
</tr>
<tr>
<td>12</td>
<td>60</td>
<td>$5.279 \times 10^{-9}$</td>
<td>0.997409</td>
<td>0.99811</td>
</tr>
</tbody>
</table>

Table 5.2a. Computer Results of the
[Start at $R_1 = 0$.]
<table>
<thead>
<tr>
<th>$R_3$</th>
<th>$R_4$</th>
<th>$-P$</th>
<th>$-f$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.6</td>
<td>0.6647</td>
<td>0.8862</td>
<td>662.4</td>
</tr>
<tr>
<td>0.5850</td>
<td>0.6175</td>
<td>0.677501</td>
<td>0.924867</td>
<td>683.298</td>
</tr>
<tr>
<td>0.6600</td>
<td>0.6750</td>
<td>0.88815</td>
<td>0.970493</td>
<td>753.431</td>
</tr>
<tr>
<td>0.7400</td>
<td>0.70833</td>
<td>0.991246</td>
<td>0.991240</td>
<td>776.258</td>
</tr>
<tr>
<td>0.791250</td>
<td>0.736458</td>
<td>0.986439</td>
<td>0.996132</td>
<td>796.919</td>
</tr>
<tr>
<td>0.767749</td>
<td>0.722707</td>
<td>0.994712</td>
<td>0.997902</td>
<td>798.981</td>
</tr>
<tr>
<td>0.730748</td>
<td>0.707252</td>
<td>0.997960</td>
<td>0.999207</td>
<td>798.889</td>
</tr>
<tr>
<td>0.715331</td>
<td>0.693293</td>
<td>0.999216</td>
<td>0.999740</td>
<td>798.831</td>
</tr>
<tr>
<td>0.710138</td>
<td>0.688581</td>
<td>0.999691</td>
<td>0.999898</td>
<td>799.273</td>
</tr>
<tr>
<td>0.704067</td>
<td>0.687510</td>
<td>0.999878</td>
<td>0.999960</td>
<td>799.504</td>
</tr>
<tr>
<td>0.703067</td>
<td>0.685176</td>
<td>0.999952</td>
<td>0.999984</td>
<td>799.680</td>
</tr>
<tr>
<td>0.702590</td>
<td>0.683271</td>
<td>0.999981</td>
<td>0.999994</td>
<td>799.730</td>
</tr>
<tr>
<td>0.702590</td>
<td>0.682817</td>
<td>0.999992</td>
<td>0.999997</td>
<td>799.938</td>
</tr>
</tbody>
</table>
Table 5.2b. Computer Results of the System

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Tires of f-value calculated at each iteration</th>
<th>Value of $r_k$</th>
<th>$R_1$</th>
<th>$R_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68</td>
<td>$1.788 \times 10^{-2}$</td>
<td>0.640000</td>
<td>0.730000</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>$4.471 \times 10^{-3}$</td>
<td>0.726250</td>
<td>0.816250</td>
</tr>
<tr>
<td>3</td>
<td>64</td>
<td>$1.118 \times 10^{-3}$</td>
<td>0.816250</td>
<td>0.876250</td>
</tr>
<tr>
<td>4</td>
<td>149</td>
<td>$2.794 \times 10^{-4}$</td>
<td>0.878124</td>
<td>0.92124</td>
</tr>
<tr>
<td>5</td>
<td>38</td>
<td>$6.986 \times 10^{-5}$</td>
<td>0.881124</td>
<td>0.92724</td>
</tr>
<tr>
<td>6</td>
<td>126</td>
<td>$1.747 \times 10^{-5}$</td>
<td>0.911037</td>
<td>0.94932</td>
</tr>
<tr>
<td>7</td>
<td>232</td>
<td>$4.366 \times 10^{-6}$</td>
<td>0.969679</td>
<td>0.98001</td>
</tr>
<tr>
<td>8</td>
<td>115</td>
<td>$1.092 \times 10^{-6}$</td>
<td>0.985835</td>
<td>0.98912</td>
</tr>
<tr>
<td>9</td>
<td>94</td>
<td>$2.729 \times 10^{-7}$</td>
<td>0.990263</td>
<td>0.99327</td>
</tr>
<tr>
<td>10</td>
<td>68</td>
<td>$6.822 \times 10^{-8}$</td>
<td>0.993763</td>
<td>0.99577</td>
</tr>
<tr>
<td>11</td>
<td>69</td>
<td>$1.706 \times 10^{-8}$</td>
<td>0.996263</td>
<td>0.99726</td>
</tr>
<tr>
<td>12</td>
<td>69</td>
<td>$4.264 \times 10^{-9}$</td>
<td>0.997626</td>
<td>0.99836</td>
</tr>
<tr>
<td>$R_3$</td>
<td>$R_4$</td>
<td>$-P$</td>
<td>$-f$ ((= R_5))</td>
<td>Cost</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>----------------</td>
<td>-------</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>0.7161</td>
<td>0.9548</td>
<td>726.6</td>
</tr>
<tr>
<td>0.610000</td>
<td>0.632500</td>
<td>0.726307</td>
<td>0.936015</td>
<td>695.180</td>
</tr>
<tr>
<td>0.711250</td>
<td>0.783750</td>
<td>0.906109</td>
<td>0.992678</td>
<td>753.153</td>
</tr>
<tr>
<td>0.791250</td>
<td>0.783750</td>
<td>0.957770</td>
<td>0.992678</td>
<td>790.870</td>
</tr>
<tr>
<td>0.766874</td>
<td>0.744843</td>
<td>0.988329</td>
<td>0.996569</td>
<td>797.343</td>
</tr>
<tr>
<td>0.763874</td>
<td>0.750843</td>
<td>0.994753</td>
<td>0.996927</td>
<td>798.672</td>
</tr>
<tr>
<td>0.740436</td>
<td>0.736226</td>
<td>0.997489</td>
<td>0.998211</td>
<td>799.632</td>
</tr>
<tr>
<td>0.692296</td>
<td>0.715007</td>
<td>0.999295</td>
<td>0.999699</td>
<td>799.637</td>
</tr>
<tr>
<td>0.687223</td>
<td>0.702855</td>
<td>0.999731</td>
<td>0.999911</td>
<td>799.305</td>
</tr>
<tr>
<td>0.685080</td>
<td>0.698569</td>
<td>0.999894</td>
<td>0.999965</td>
<td>799.322</td>
</tr>
<tr>
<td>0.684080</td>
<td>0.697069</td>
<td>0.999958</td>
<td>0.999986</td>
<td>799.593</td>
</tr>
<tr>
<td>0.682652</td>
<td>0.695640</td>
<td>0.999983</td>
<td>0.999994</td>
<td>799.568</td>
</tr>
<tr>
<td>0.682652</td>
<td>0.694958</td>
<td>0.999993</td>
<td>0.999998</td>
<td>799.733</td>
</tr>
</tbody>
</table>
first set of starting points, it takes 12 iterations for \( P \) functions, \( k = 12 \), with totally 1192 \( f \)-functional values evaluated. And for the second set, 12 iterations for \( P \) functions, \( k = 12 \), with totally 1194 \( f \)-functional values evaluated.

Tables 5-2a and 5-2b present the iteration results converging to the optimal solution. Results given in these tables show that the system reliability, \( R_s \), is monotonically increasing as iteration \( k \) increases. The value of \( P \) function approaches to that of \( f \) function \( (= -R_s) \) as the iteration proceeds. Thus the minimization of \( P \) function will eventually lead us to the minimization of \( f \) function.

The values of \( r_0 \) used in Tables 5.2a and 5.2b are determined by

\[
 f(x_0) = r_0 \sum \frac{1}{g_i(x_0)} \tag{5.11}
\]

where \( x_0 \) is the initial point. The basis for of this selection procedure is to render the value of the penalty of the constraints to be approximately the same order of magnitude as the value of the \( f \)-function at the starting point in the \( P \)-function formulation

\[
 P(x_0, r_0) = f(x_0) + r_0 \sum \frac{1}{g_i(x_0)}
\]

The computer time consumed to obtain each set of the solutions presented in Tables 5.2a and 5.2b is 45 seconds respectively on an IBM 360/50 computer by using the Watfor processor. Recall that the same problems solved by the RAC program, as presented in Chapter 2, consumes over 20 minutes on the same computer.

5.4. THE COST FUNCTION MINIMIZATION PROBLEM

The numerical example of this problem studied is restated below. The objective is to find the optimal \( R_i \)'s which minimize
\[ C = 2K_1R_1^{\alpha_1} + 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4} \] 

(5.11)

subject to the constraints

\[ R_{\min} \leq 1 - R_3[(1-R_1)(1-R_4)]^2 - (1-R_3)(1-R_2[1 - (1-R_1)(1-R_4)])^2 \] 

(5.12)

\[ R_i \geq R_{i,\min} \] 

(5.13)

The numerical values of parameters are

\[ K_1 = 100, \quad K_2 = 100, \quad K_3 = 200, \quad K_4 = 150 \]

\[ C = 800, \quad \alpha_i = 0.6, \quad i = 1, 2, 3, 4. \]

\[ R_{\min} = 0.9, \quad R_{i,\min} = 0.5, \quad i = 1, 2, 3, 4. \]

The problem is formulated in SUMT format as follows:

Minimize

\[ f(x) = C \]

\[ = 2K_1R_1^{\alpha_1} + 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4} \] 

(5.14)

subject to the constraints

\[ g_1(x) = 1 - R_3[(1-R_1)(1-R_4)]^2 - (1-R_3)(1 - R_2[1 - (1-R_1)(1-R_4)])^2 - R_{\min} \geq 0 \] 

(5.15)

\[ g_{i+1}(x) = R_i - R_{i,\min} \geq 0, \quad i = 1, 2, 3, 4. \] 

(5.16)

The \( P \) function for this problem is

\[ P(x, r_k) = f(x) + r_k \sum_{i} 1/g_i(x) \]

\[ = 2K_1R_1^{\alpha_1} + 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4} \]
\[ + r_k \left\{ \frac{1}{k(1 - R_3((1-R_1)(1-R_4))^2 - (1-R_3) 1-R_2(1 - (1-R_1)(1-R_4)^2 - R_{\text{min}} ) \right}\right. \\
= + \left. \frac{4}{i=1} \left( \frac{1}{R_i - R_{i,\text{min}}} \right) \right\} \]

(5.17)

where \( x \) is the row vector of \( (R_1, R_2, R_3, R_4) \).

For this problem, the RAC program fails to satisfy the special requirement that the violable non-negativity constraints should never be violated during the search. The results obtained by applying the new developed program is presented in Tables 5.3, 5.4a and 5.4b.

The optimal solutions obtained from two sets of different starting components reliabilities, namely, \([R_1, R_2, R_3, R_4] = [0.6, 0.6, 0.6, 0.6]\) and \([R_1, R_2, R_3, R_4] = [0.7, 0.7, 0.7, 0.7]\) are presented in Table 5.3. The solutions are almost identical, that is, the optimal minimum cost, \( C \), of 642.249 with the system reliability, \( R_s \), of 0.900159 for the first set of starting components reliabilities, and the optimal minimum cost, \( C \), of 642.426 with the system reliability, \( R_s \), of 0.900021 for the second set of starting components reliabilities are obtained. Recall that the constraint on the system reliability is 0.9. The optimal components reliabilities are almost the same for both starting sets. The stopping criterion for terminating minimization of the \( P \) function at each iteration is that terminating when the number of cut-down step-size operations is 3. And the final stopping criterion for terminating the problem is \( \epsilon = 10^{-4} \). For the first set of starting points, it takes 12 iterations for \( P \) functions, \( k = 12 \), with totally 1896 \( f \)-functional values calculated. And for the second set, 14 iterations for \( P \) functions,
<table>
<thead>
<tr>
<th>Iteration of $r_k$</th>
<th>Iteration of $f$</th>
<th>Values of Component Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td>$n$</td>
<td>$R_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>0.502711</td>
</tr>
<tr>
<td>0</td>
<td>14</td>
<td>0.7</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>0.512435</td>
</tr>
</tbody>
</table>
Solution of the Cost Minimization Problem

<table>
<thead>
<tr>
<th>$R_4$</th>
<th>$R_s$</th>
<th>Cost</th>
<th>Stopping criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.8992</td>
<td>662.4</td>
<td></td>
</tr>
<tr>
<td>0.50202%</td>
<td>0.900159</td>
<td>642.249</td>
<td>3, $10^{-5}$</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9548</td>
<td>726.6</td>
<td></td>
</tr>
<tr>
<td>0.501431</td>
<td>0.900021</td>
<td>642.428</td>
<td>3, $10^{-5}$</td>
</tr>
</tbody>
</table>
Table 5.4a. Computer results of the cos
[Start at \( R_1 = 0.6 \),

<table>
<thead>
<tr>
<th>Iteration k</th>
<th>Times of f-value calculated at each iteration</th>
<th>Value of ( r_k )</th>
<th>( R_1 )</th>
<th>( R_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0.6</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>70</td>
<td>1.470</td>
<td>0.605000</td>
<td>0.645000</td>
</tr>
<tr>
<td>2</td>
<td>87</td>
<td>0.3675</td>
<td>0.554999</td>
<td>0.795000</td>
</tr>
<tr>
<td>3</td>
<td>67</td>
<td>0.09189</td>
<td>0.541666</td>
<td>0.821666</td>
</tr>
<tr>
<td>4</td>
<td>44</td>
<td>0.02297</td>
<td>0.531666</td>
<td>0.806666</td>
</tr>
<tr>
<td>5</td>
<td>107</td>
<td>0.005743</td>
<td>0.513666</td>
<td>0.828666</td>
</tr>
<tr>
<td>6</td>
<td>101</td>
<td>0.001436</td>
<td>0.507302</td>
<td>0.833211</td>
</tr>
<tr>
<td>7</td>
<td>68</td>
<td>0.0003589</td>
<td>0.503969</td>
<td>0.835711</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>0.00008973</td>
<td>0.503199</td>
<td>0.834942</td>
</tr>
<tr>
<td>9</td>
<td>189</td>
<td>0.00002243</td>
<td>0.502712</td>
<td>0.834632</td>
</tr>
<tr>
<td>10</td>
<td>51</td>
<td>0.000005608</td>
<td>0.502712</td>
<td>0.834632</td>
</tr>
<tr>
<td>11</td>
<td>1009</td>
<td>0.000001402</td>
<td>0.502711</td>
<td>0.834631</td>
</tr>
<tr>
<td>12</td>
<td>51</td>
<td>0.0000003505</td>
<td>0.502711</td>
<td>0.834631</td>
</tr>
</tbody>
</table>
for all $i$]

<table>
<thead>
<tr>
<th>$R_3$</th>
<th>$R_4$</th>
<th>$P$</th>
<th>$f$ (= Cst)</th>
<th>$R_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.6</td>
<td>828.0</td>
<td>662.4</td>
<td>0.8892</td>
</tr>
<tr>
<td>0.60000</td>
<td>0.60000</td>
<td>981.8</td>
<td>676.9</td>
<td></td>
</tr>
<tr>
<td>0.60000</td>
<td>0.64000</td>
<td>848.140</td>
<td>664.796</td>
<td>0.912690</td>
</tr>
<tr>
<td>0.559999</td>
<td>0.580000</td>
<td>704.935</td>
<td>673.866</td>
<td>0.927331</td>
</tr>
<tr>
<td>0.519999</td>
<td>0.533333</td>
<td>672.318</td>
<td>658.077</td>
<td>0.916563</td>
</tr>
<tr>
<td>0.509999</td>
<td>0.518332</td>
<td>656.939</td>
<td>650.045</td>
<td>0.906742</td>
</tr>
<tr>
<td>0.506999</td>
<td>0.507332</td>
<td>649.202</td>
<td>645.493</td>
<td>0.903444</td>
</tr>
<tr>
<td>0.503362</td>
<td>0.503696</td>
<td>645.501</td>
<td>643.648</td>
<td>0.901716</td>
</tr>
<tr>
<td>0.501696</td>
<td>0.501821</td>
<td>643.660</td>
<td>642.738</td>
<td>0.900850</td>
</tr>
<tr>
<td>0.500926</td>
<td>0.501821</td>
<td>642.817</td>
<td>642.395</td>
<td>0.900363</td>
</tr>
<tr>
<td>0.500439</td>
<td>0.501494</td>
<td>644.016</td>
<td>642.124</td>
<td>0.900012</td>
</tr>
<tr>
<td>0.500439</td>
<td>0.501452</td>
<td>649.370</td>
<td>642.114</td>
<td>0.900001</td>
</tr>
<tr>
<td>0.500438</td>
<td>0.501451</td>
<td>665.640</td>
<td>642.113</td>
<td>0.900000 **</td>
</tr>
<tr>
<td>0.500438</td>
<td>0.502024</td>
<td>642.252</td>
<td>642.249</td>
<td>0.900159</td>
</tr>
</tbody>
</table>
Table 5.4b. Computer results of the computations [Start at $R_1 = 0.7$]

<table>
<thead>
<tr>
<th>Iteration $k$</th>
<th>Times of $f$-value calculated at each iteration</th>
<th>Value of $r_k$</th>
<th>$R_1$</th>
<th>$R_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81</td>
<td>4.749</td>
<td>0.745000</td>
<td>0.842500</td>
</tr>
<tr>
<td>2</td>
<td>102</td>
<td>1.187</td>
<td>0.043749</td>
<td>0.793750</td>
</tr>
<tr>
<td>3</td>
<td>99</td>
<td>0.2968</td>
<td>0.581249</td>
<td>0.798749</td>
</tr>
<tr>
<td>4</td>
<td>84</td>
<td>0.07421</td>
<td>0.549374</td>
<td>0.806249</td>
</tr>
<tr>
<td>5</td>
<td>76</td>
<td>0.01855</td>
<td>0.526874</td>
<td>0.818249</td>
</tr>
<tr>
<td>6</td>
<td>91</td>
<td>0.004638</td>
<td>0.515965</td>
<td>0.825749</td>
</tr>
<tr>
<td>7</td>
<td>58</td>
<td>0.001159</td>
<td>0.514089</td>
<td>0.825749</td>
</tr>
<tr>
<td>8</td>
<td>68</td>
<td>0.0002899</td>
<td>0.513224</td>
<td>0.826325</td>
</tr>
<tr>
<td>9</td>
<td>497</td>
<td>0.00007247</td>
<td>0.512688</td>
<td>0.825790</td>
</tr>
<tr>
<td>10</td>
<td>358</td>
<td>0.00001812</td>
<td>0.512412</td>
<td>0.825519</td>
</tr>
<tr>
<td>11</td>
<td>51</td>
<td>0.000004529</td>
<td>0.512412</td>
<td>0.825519</td>
</tr>
<tr>
<td>12</td>
<td>288</td>
<td>0.000001132</td>
<td>0.512433</td>
<td>0.825194</td>
</tr>
<tr>
<td>13</td>
<td>1006</td>
<td>0.0000002831</td>
<td>0.512435</td>
<td>0.825132</td>
</tr>
<tr>
<td>14</td>
<td>59</td>
<td>0.00000007977</td>
<td>0.512435</td>
<td>0.825132</td>
</tr>
</tbody>
</table>
The function minimization problem.

for all $i$)

<table>
<thead>
<tr>
<th>$R_3$</th>
<th>$R_4$</th>
<th>$P$</th>
<th>$\xi$ (= Cost)</th>
<th>$R_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>908.3</td>
<td>726.6</td>
<td>0.9548</td>
</tr>
<tr>
<td>0.715000</td>
<td>0.670000</td>
<td>890.124</td>
<td>747.532</td>
<td>0.980070</td>
</tr>
<tr>
<td>0.610000</td>
<td>0.589374</td>
<td>756.752</td>
<td>694.793</td>
<td>0.946418</td>
</tr>
<tr>
<td>0.552499</td>
<td>0.546874</td>
<td>696.952</td>
<td>668.156</td>
<td>0.924405</td>
</tr>
<tr>
<td>0.526249</td>
<td>0.524999</td>
<td>668.814</td>
<td>655.247</td>
<td>0.912314</td>
</tr>
<tr>
<td>0.512749</td>
<td>0.512999</td>
<td>655.177</td>
<td>648.439</td>
<td>0.905970</td>
</tr>
<tr>
<td>0.505931</td>
<td>0.506692</td>
<td>648.484</td>
<td>645.156</td>
<td>0.902995</td>
</tr>
<tr>
<td>0.502806</td>
<td>0.503723</td>
<td>645.279</td>
<td>643.667</td>
<td>0.901447</td>
</tr>
<tr>
<td>0.501652</td>
<td>0.501704</td>
<td>643.706</td>
<td>642.946</td>
<td>0.900739</td>
</tr>
<tr>
<td>0.501116</td>
<td>0.501194</td>
<td>642.998</td>
<td>642.586</td>
<td>0.900259</td>
</tr>
<tr>
<td>0.500879</td>
<td>0.501078</td>
<td>642.807</td>
<td>642.443</td>
<td>0.900056</td>
</tr>
<tr>
<td>0.500879</td>
<td>0.501978</td>
<td>642.679</td>
<td>642.656</td>
<td>0.900298</td>
</tr>
<tr>
<td>0.500560</td>
<td>0.501531</td>
<td>642.481</td>
<td>642.461</td>
<td>0.900065</td>
</tr>
<tr>
<td>0.500549</td>
<td>0.501470</td>
<td>642.448</td>
<td>642.438</td>
<td>0.900031</td>
</tr>
<tr>
<td>0.500549</td>
<td>0.501431</td>
<td>642.432</td>
<td>642.428</td>
<td>0.900021</td>
</tr>
</tbody>
</table>
END

OF

OVERSIZED

DOCUMENTS
k = 14, with totally 2918 f-functional values calculated.

Results given in Tables 5.4a and 5.4b show that the cost of the system, C, is monotonically decreasing as iteration k increases. The value of the P function approaches to that of the f function (=C) as the iteration proceeds. Thus the minimization of the P function will eventually lead us to the minimization of f function.

Again, the values of \( r_0 \) are determined from Equation (5.11) as explained in Section 5.3.

The computer time consumed to obtain both sets of the results presented in Tables 5.4a and 5.4b is 75 seconds on an IBM 360/50 computer by using the Watfor processor.

It is worth noting that the starting point \( R^0 = (R_1, R_2, R_3, R_4) = (0.6, 0.6, 0.6, 0.6) \) in Table 5.4a is in infeasible region. The system reliability given by \( R^0 \) is 0.8892 which is less than \( R_{s,\text{min}} \) of 0.9. Therefore, before the P-function minimization routine is started, a new feasible point is searched first. The point \( (0.64, 0.64, 0.6, 0.6) \) in the second row of Table 5.4a is thus selected and is used as the feasible starting point to start the minimization procedure. The method used to search this new feasible starting point has been discussed in Chapter 3.

5.5 CONCLUDING REMARKS

From the results presented in this chapter and those in Chapter 4, several conclusions can be drawn.

(1) The procedure of selecting the initial value of penalty coefficient, \( r_0 \), is valid and convenient. In this procedure the value of the sum of the penalty terms is made approximately the same order of magnitude of the f-function at the initial point, \( x_0 \), that is,
\[ f(x_0) = r_0 \sum_i \frac{1}{g_i(x_0)} + r_0 \frac{1}{2} \sum_j h_j^2(x_0) \]  

Equation (5.18) is solved for \( r_0 \) and this value is used as the starting \( r \).

(2) The modified Hooke and Jeeves pattern search technique has been proven to be a successful one in the solution of the numerical examples studied in this chapter. As shown in Table 1, the optimal solutions obtained by employing the RAC program and that by the new program developed in the present work are almost identical.

(3) The number of functional values evaluated by applying the new computer program is large. This can be a significant disadvantage, especially when the \( f \)-function and/or constraint functions cannot be evaluated in a straighforward manner, for example, the functions are nonlinear differential equations.

(4) The computing time compared in Table 5.1 shows the big difference on the time consumptions by the two different programs for the same system reliability maximization problem. Only 90.4 seconds are needed for obtaining the two solutions by the new computer program developed in this work. While either problem needs to consume over 20 minutes by the RAC program.

(5) The optimal solution for the cost minimization problem can be obtained by the new program while the RAC program fails to give an solution.

(6) There is a difficulty in the optimization of the cost minimization problem mentioned. The feasible region bounded by the given constraints is very narrow and so the constraints is violated frequently. Usually,
in most techniques such as the RAC program and the method developed in
Chapter 3, there provided some modification to move a point in the in-
feasible region back to the feasible region. In maximizing the system
reliability problem both programs does not give rise to much difficulty.
However, in the minimizing cost problem for this particular character
of system reliability optimization problems the RAC program fails to
solve the problem. Because the cost function to be minimized is

\[ C = 2K_1 R_1^{0.6} + 2K_2 R_2^{0.6} + K_3 R_3^{0.6} + 2K_4 R_4^{0.6} \]

where \( K_1, K_2, K_3 \) and \( K_4 \) are constants and \( R_i \) is the component reliability
for the \( i \)th component which involve \( R_i^{0.6} \) terms. When \( C \) is minimized,
\( R_i \)'s, essentially, decrease. The non-negativity constraints over
component reliabilities are violable. When the non-negativity con-
straints are violated, the respective \( R_i^{0.6} \) is mathematically undefined
and so is the cost function.
ACKNOWLEDGEMENTS

The author thanks Dr. C. L. Hwang, and Dr. L. T. Fan for their valuable guidance, encouragement and cooperation in the preparation of this report. He also sincerely acknowledges the support and encouragement provided by Dr. F. A. Tillman and Mr. F. T. Hsu.

This study was partly supported by NASA Grant No. NGR-17-001-034, and by the Air Force Office of Scientific Research, Office of Aerospace Research, United States Air Force, Under Contract F44620-68-0020 (Themis Project).
APPENDIX

COMPUTER PROGRAM FOR IMPLEMENTING SUMT BY
HOOKE AND JEEVES PATTERN SEARCH TECHNIQUE

The computer flow chart which illustrates the computational procedure is presented in Fig. 1, 2, 3, 4 and 5; the FORTRAN program symbols, their explanations and corresponding mathematical notations are summarized in Table 1. The computer program listing follows the symbol table.
Fig. I. Descriptive flow diagram for SUMT with Hooke and Jeeves Pattern Search.
Fig. 2. Descriptive flow diagram for selecting a feasible starting point.
Enter

Compute the $P(X, r_k)$ at starting point for k-th suboptimun search.

Start at base point

Make exploratory move

Did exploratory move make progress?

YES

Set new base point

Make pattern move

Did pattern move make progress?

YES

Is pattern move point feasible?

YES

Set new base point

NO

If a move go out of the feasible region, move back according to procedure in Fig. 4.

Is step size small enough?

YES

EXIT

NO

Cut down step sizes

Move back into feasible region (or near-feasible region) according to Fig. 4.

Fig. 3. Descriptive flow diagram for Hooke and Jeeves Pattern Search for minimizing $P(X, r_k)$ function.
Fig. 4. Descriptive flow diagram for moving an infeasible point back into near feasible region.
Compute the weight of violation

\[ TGH = \left\{ \sum g_i(x_k^o)^2 + \sum h_s(x_k^o)^2 \right\}^{1/2} \text{ for all } g_i \leq 0, h_s \neq 0 \]

where \( x_k^o \) is the entering \( k \)-th sub-optimum.

Move \( x_k^o \) toward \( x_{k-1}^o \), the feasible \((k-1)\)-th sub-optimum for a step to a new point \( x_k^{o'} \).

Set \( x_k^o = x_k^{o'} \).

Is \( x_k^o \) feasible?

Yes

EXIT

No

Fig. 5. Descriptive flow diagram for moving the near-feasible \( k \)-th sub-optimum into feasible region.
### Table 1. Program Symbols and Explanation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Explanation</th>
<th>Mathematical Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>tolerance limits for constraint violation</td>
<td></td>
</tr>
<tr>
<td>BX(I)</td>
<td>base point in Hooke and Jeeves pattern search</td>
<td></td>
</tr>
<tr>
<td>D(I)</td>
<td>step-size in Hooke and Jeeves pattern search</td>
<td>( d_i )</td>
</tr>
<tr>
<td>FG(J)</td>
<td>((j)th inequality constraint value at point FX(I))</td>
<td>( g_j )</td>
</tr>
<tr>
<td>FH(K)</td>
<td>((k)th equality constraint value at point FX(I))</td>
<td>( h_k )</td>
</tr>
<tr>
<td>FP</td>
<td>P-function value at point FX(I)</td>
<td>( P )</td>
</tr>
<tr>
<td>FRAC</td>
<td>the fraction of step-sizes used in pulling back infeasible point to the feasible region</td>
<td></td>
</tr>
<tr>
<td>FX(1)</td>
<td>the intermediate suboptimum point during search</td>
<td></td>
</tr>
<tr>
<td>FY</td>
<td>(f)-function value at point FX(I)</td>
<td>( f )</td>
</tr>
<tr>
<td>FTGH</td>
<td>the intermediate least value of TGH during pulling-back procedure</td>
<td></td>
</tr>
<tr>
<td>G(J)</td>
<td>((j)th inequality constraint value at point X(I))</td>
<td>( g_j )</td>
</tr>
<tr>
<td>H(K)</td>
<td>((k)th equality constraint value at point X(I))</td>
<td>( h_k )</td>
</tr>
<tr>
<td>IB</td>
<td>program control code, IB = 1 means that the point is on the boundary</td>
<td></td>
</tr>
<tr>
<td>ICHECK</td>
<td>program control code, ICHECK = 1 means that ITMAX is exceeded</td>
<td></td>
</tr>
<tr>
<td>ICUT</td>
<td>input option code for initial step-sizes set-up</td>
<td></td>
</tr>
<tr>
<td>IDPM</td>
<td>problem number</td>
<td></td>
</tr>
<tr>
<td>INCUT</td>
<td>stopping criterion for stopping each k-iteration</td>
<td></td>
</tr>
<tr>
<td>ISIZE</td>
<td>input option code for initial step-sizes set-up</td>
<td></td>
</tr>
<tr>
<td>ITER</td>
<td>number of times of calculating f-functional values within a k-iteration</td>
<td></td>
</tr>
<tr>
<td>ITMAX</td>
<td>specified maximum number of calculating f-functional values within each k-iteration</td>
<td></td>
</tr>
<tr>
<td>LOST</td>
<td>program control code, LOST ≠ 0 means that some ( g_j &lt; 0 )</td>
<td></td>
</tr>
<tr>
<td>MG</td>
<td>total number of inequality constraints</td>
<td>( m )</td>
</tr>
<tr>
<td>Program Symbols</td>
<td>Explanation</td>
<td>Mathematical Symbols</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>MH</td>
<td>total number of equality constraints</td>
<td>$\ell$</td>
</tr>
<tr>
<td>MAXP</td>
<td>specified maximum number of k-iterations</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>total number of decision variables</td>
<td>$n$</td>
</tr>
<tr>
<td>NAME1</td>
<td>three parts of the name of the input problem (6 characters)</td>
<td></td>
</tr>
<tr>
<td>NAME2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAME3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOBP</td>
<td>number of moves go out of the feasible region</td>
<td></td>
</tr>
<tr>
<td>NOSCUT</td>
<td>number of cut-down step-size operations</td>
<td></td>
</tr>
<tr>
<td>NOEXP</td>
<td>number of exploratory moves</td>
<td></td>
</tr>
<tr>
<td>NOIT</td>
<td>total number of times of calculating f-functional values from the very beginning</td>
<td></td>
</tr>
<tr>
<td>NOITB</td>
<td>number of moves iterated in the infeasible region</td>
<td></td>
</tr>
<tr>
<td>NOITP</td>
<td>number of moves iterated in the feasible region</td>
<td></td>
</tr>
<tr>
<td>NOPAT</td>
<td>number of pattern moves</td>
<td></td>
</tr>
<tr>
<td>NOPM</td>
<td>number of input problem sets</td>
<td></td>
</tr>
<tr>
<td>NOPULL</td>
<td>number of times of the operations pulling back suboptimum to the feasible region</td>
<td></td>
</tr>
<tr>
<td>NOR</td>
<td>number of k</td>
<td>$k$</td>
</tr>
<tr>
<td>OX(1)</td>
<td>suboptimum point</td>
<td>$x^0$</td>
</tr>
<tr>
<td>P</td>
<td>P-functional value at point X(1)</td>
<td>$P$</td>
</tr>
<tr>
<td>PB</td>
<td>initial tolerance limit of constraint violation</td>
<td></td>
</tr>
<tr>
<td>PD(1)</td>
<td>initial step-size</td>
<td>$d^0_i$</td>
</tr>
<tr>
<td>PENAL</td>
<td>penalty value to inequality constraints</td>
<td>$r_k \sum_{j=1}^{\ell} \frac{1}{s_j}$</td>
</tr>
<tr>
<td>PENAL2</td>
<td>penalty value to equality constraints</td>
<td>$r_k \sum_{j} h_j^2$</td>
</tr>
<tr>
<td>PX(1)</td>
<td>pattern move point in Hooke and Jeeves pattern search</td>
<td></td>
</tr>
</tbody>
</table>
Table 1. Program Symbols and Explanation (continued)

<table>
<thead>
<tr>
<th>Program Symbols</th>
<th>Explanation</th>
<th>Mathematical Symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>PULL</td>
<td>a fraction used to pull back suboptimum to the feasible region</td>
<td>$r_k$</td>
</tr>
<tr>
<td>R</td>
<td>penalty coefficient</td>
<td>$C$</td>
</tr>
<tr>
<td>RATIO</td>
<td>reducing rate for reducing R</td>
<td></td>
</tr>
<tr>
<td>STGH</td>
<td>least value of TGH during searching a feasible starting point procedure</td>
<td></td>
</tr>
<tr>
<td>TGH</td>
<td>weight of violation to constraints</td>
<td>$(\sum_k g_k^2 + \sum_s h_s^2)^{1/2}$</td>
</tr>
<tr>
<td>THETA</td>
<td>final stopping criterion</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>X(I)</td>
<td>a point</td>
<td>$x_i$</td>
</tr>
<tr>
<td>XB(NB)</td>
<td>a point in dulling-back processes</td>
<td>$x_i$</td>
</tr>
<tr>
<td>Y</td>
<td>f-functional value at point X(I)</td>
<td>$f$</td>
</tr>
<tr>
<td>YSTOP</td>
<td>computed value of $\varepsilon$</td>
<td>$\left</td>
</tr>
</tbody>
</table>
THIS PROGRAM IS FOR OPTIMIZING CONSTRAINED MINIMIZATION PROBLEMS
BY A COMBINATIONAL USE OF HOOK AND JEEVES PATTERN SEARCH TECHNIQUE
AND SUMT FORMULATION, WHEN THE SEARCH GETS OUT OF THE FEASIBLE
REGION, IT WILL BE PULLED BACK BY A HEURISTIC PROGRAMMING TECHNIQUE,
EXECUTED BY THE SUBROUTINE CALL.
THE ORIGINAL IDEAS CAME FROM
SEARCH TECHNIQUE, HOOK AND JEEVES
SUMT FORMULATION, FIACCO AND MCCORMICK
PULL BACK TECHNIQUE, AVIANI AND Himmelblau.
THE NECESSARY REFERENCE DOCUMENTS CAN BE SEEN IN MY MASTER
REPORT.

**INPUT-OUTPUT VARIABLES**

**INPUT**
NORM, NO. OF SUBPROBLEMS INPUT.
NAME1, NAME2, NAME3, 3 PARTS OF PROBLEM NAME, USER MAY USE
ANY 6 CHARACTERS TO NAME THE PROBLEM.
N, NO. OF VARIABLES OF THE PROBLEM.
M, NO. OF INEQUALITY CONSTRAINTS GI(j), GE, 0.
M1, NO. OF EQUALITY CONSTRAINTS HI(k), EQ, 0.
Q, PENALTY COEFFICIENT FOR SUMT FORMULATION.
OPTION -= R, LE, 0, Q, WILL USE A COMPUTED VALUE.
RATIO, REDUCING RATE FOR R FROM STAGE TO STAGE.
OPTION -= RATIO, LE, 0, Q, WILL USE RATIO=4, Q.
ITMAX, INPUT WITHIN-STAGE ITERATION MAXIMUM NO.
INPUT, STOPPING CRITERION FOR STAGE ITERATION, NO. OF
CUT-DOWN STEP-SIZE OPERATION, USE 2, 3 OR 4.
THETA, FINAL STOPPING CRITERION, USE ABOUT 10**(-4).
MAXP, INPUT MAXIMUM NO. OF STAGES, IF EXCEEDED, STOP.
X(I), (I)TH DIMENSION OF DECISION VARIABLE.
Y(I), (I)TH DIMENSION OF STEP SIZE.
P, P FUNCTION VALUE.
Y, F FUNCTION VALUE.
YSTOP, COMPUTED VALUE OF FINAL-STOPPING DETERMINATOR.
IDPM, SEQUENCE NO. OF SUBPROBLEMS OUTPUT.
NOR, NO. OF STAGES UP TO CURRENT STAGE.
B  .. TOLERANCE LIMIT FOR VIOLATIONS.
EY  .. MINIMUM Y GOT SO FAR.
EP  .. MINIMUM P GOT SO FAR.
S(J)  .. JTH INEQUALITY CONSTRAINT VALUE.
H(K)  .. KTH EQUALITY CONSTRAINT VALUE.
ITER  .. WITHIN STAGE ITERATION NO.
NOIT  .. CUMULATED ITERATION NO.
NOGUT  .. NO. OF CUT DOWN STEP-SIZE OPERATION WITHIN STAGE.
NOCXP  .. NO. OF SUCCESSFUL EXPLORATORY MOVES.
NOPAT  .. NO. OF SUCCESSFUL PATTER MOVES.
NOH  .. NO. OF TIMES OF PULLING BACK PROCEDURE.
NOP  .. NO. OF SUCCESSFUL MOVES INSIDE FEASIBLE REGION.
NOR  .. NO. OF SUCCESSFUL MOVES INSIDE FEASIBLE REGION.
NOITH  .. NO.
NOITHB  .. NO. OF ITERATIONS OUT OF FEASIBLE REGION.

******************************************************************************

**SEQUENCE OF INPUT DECK **

(1) PROBLEM ID CARD  .. ONE CARD, FORMAT 1000.
   PARAMETERS -- N,M,NAM=(COMPOSED BY 3 PARTS),
   N,MG AND MH.

(2) PROBLEM ADDITIONAL DATA CARDS  .. SPECIFIED IN THE
   SUBROUTINE READING BY USER HIMSELF, (OPTIONAL).

(3) SUBPROBLEM 1 INITIAL DATA CARDS  ..
   FIRST  -- ONE CARD, FORMAT 1002.
   PARAMETERS - R, RATIO, ITMAX, INCUT, THETA
   AND MAXP.
   SECOND -- N CARDS, FORMAT 1004.
   PARAMETERS - J, X(I), AND D(I).
   **NOTE -- 1, J IS ONLY FOR USER TO
   CHECK.
   2, CARDS SHOULD BE IN ORDER
   (SEQUENCE OF DIMENSION.)

(4) SUBPROBLEM 2 INITIAL DATA CARDS  ..
   ..
   ..
   ( ... UP TO THE LAST SUBPROBLEM INITIAL DATA CARDS ...)
**READ IN PROBLEM NUMBER, PROBLEM NAME, AND DIMENSIONS.**

READ(1,1000) N,DPM,NAME1,NAME2,NAME3,N,M,G,MH
WRITE(3,1021)
WRITE(3,1001) NAME1,NAME2,NAME3,N,M,G,MH,DPM
DPM=1

**READ IN ADDITIONAL DATA (USED FOR ALL SUB-PROBLEMS).**

CALL READIN(N,M,G,MH)

**READ IN INITIAL PARAMETERS AND STOPPING CRITERIA.**

READ(1,1002) R, RATIO, ITMAX, INCUT, THETA, MXP
WRITE(3,1003) IDPM
MP=1
MULT=1
NOEXP=0
NOPAT=0
NOCUT=0
NOR=1
NOB=0
NOITP=C
NOITP=0
ITER=0
NIIT=0
LST=0
IB=0
**SUBROUTINES NEEDED...**

BACK -- USED TO PULL BACK INFEASIBLE POINT

PENAT -- USED TO COMPUTE PENALTY TERMS

WEIG -- USED TO COMPUTE VIOLATION WEIGHT

READTN -- A USER SUPPLIED SUBROUTINE, USED TO READ IN ADDITIONAL DATA NEEDED

OPRNS -- A USER SUPPLIED SUBROUTINE, USED TO COMPUTE THE OBJECTIVE AND CONSTRAINTS

OUTPUT -- A USER SUPPLIED SUBROUTINE, USED TO OUTPUT ADDITIONAL INFORMATION DESIRED

**DIMENSIONS...**

THIS PROGRAM IS DESIGNED FOR N,MG,MH.LE. 20, WHEN THE DIMENSIONS OF N,MG, AND/OR MH EXCEED 20, MAKE PROPUR CHANGES, THE KEY TO CHANGES ARE...

K,FK,FY,BY,OY,DY -- N DIMENSIONS

G,FG -- MG DIMENSIONS

H,FH -- MH DIMENSIONS

****

DIMENSION X(20),FX(20),DX(20),PX(20),NOX(20),PD(20),D(20),G(20),
1FG(20),H(20),FH(20)

COMMON /BLOGY/ N,MG,MH,ITER,ITMAX,ICHECK,IP,LUST

COMMON /BLOGY/ WOIT,WOITG,BD

COMMON /BLOGY/ O110

COMMON /BLOGY/ FG

1000 FORMAT(15,5X2,42,23,15)
1001 FORMAT(1H30X1H*42,23,20,16H* PROBLEMS/30X20(1H*//25X14HNO. OF X
111) .14,2H .25X14HNO. OF G(I) .14,2H .25X14HNO. OF H(K) .14,2
2H .18X17HNO. OF PROBLEMS .14,2H .16X58HPARAMETERS IN OUTPUT,
3. SEE COMMENT CARDS IN THE PROGRAM.

1002 FORMAT(2515,4,215,015,4,15)
1003 FORMAT(10X7HPROBLEM14//)
1004 FORMAT(15,2215,4)
1005 FORMAT(20X13HINITIAL POINT/5X4LH = E11.4,7H, P = E11.4,7H, R = E
IDR=0
ICHECK=0
B=0.
FN=N

C
C **READ IN INITIAL POINT AND STARTING STEP-SIZES**
C DO 2 I=1,N
C READ(1,1004) J,X(I),D(I)
C
C **VARIABLE (I) IS USED FOR CHECKING THE SEQUENCE OF CARDS BY THE
C USER HIMSELF, AND HAS NO INFLUENCE TO THE PROGRAM (USER MAY
C USE ANY INTEGER NUMBER FOR J).
B*X(I)=X(I)
F*X(I)=X(I)
P*C(I)=C(I)
D*X(I)=X(I)
2 B=0.5*P(I)

C **DECIDE THE STARTING VALUE OF TOLERANCE LIMIT FOR G(J) .LT. 0.**
B=1/FN
CALL OBRES(FX,FX,FG,FH)
CALL WEIGH(STGH,MX,FG)
ITER=0
11 CALL PENAT(FG,FH,PENAI,PENA2)

C **COMPUTE AN INITIAL VALUE OF R WHEN INPUT R VALUE IS .LE. 0.**
IF(R) 12,12,13
12 R=ABS(FX/(PENA1+PENA2))
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(3,1005) FY,FP,R,RATIO,B,INCUT,THETA
WRITE(3,1006) (I,FX(I),I,D(I),I=1,N)
WRITE(3,1007)
16 IF(LOST-2) 50,16,16
C **SELECT INFEASIBLE 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.**
NOF=0
DO 28 I=1,N
28 FX(I)=X(I)+2.0*D(I)
CALL OBRES(FX,FY,FG,FH)
CALL WEIGHT(TGK,MG,FG)
IF(TGK) 44,44,13
10 IF(STGH-TGH) 20,20,26
20 FX(I)=FX(I)-O*D(I)
CALL OBRES(FX,FY,FG,FH)
CALL WEIGHT(TGK,MG,FG)
IF(TGK) 44,44,22
22 IF(STGH-TGH) 24,24,26
24 FX(I)=FX(I)+2,O*D(I)
NOF=NOF+1
GO TO 20
26 STGH=TGH
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)*0.5
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)-X(I))
CALL OBRES(PX,FY,FG,FH)
CALL WEIGHT(TGK,MG,FG)
IF(STGH-TGH) 16,16,40
40 DO 42 I=1,N
42 X(I)=PX(I)
44 IF(TGK) 44,44,43
43 STGH=TGH
46 GO TO 16
100 DO 46 I=1,N
46 D(I)=PD(I)
48 PX(I)=FX(I)
GO TO 16
C **OUTPUT THE MESSAGE OF THE SELECTED FEASIBLE STARTING POINT.
WRITE(3,1020)
104 GO TO 11
105
48. DO 49 I=1,N
49. X(I)=FX(I)
   **START TO MINIMIZE THE CURRENT P-FUNCTION.**
   **MAKE EXPLORATORY MOVE FOR MINIMIZING THE P-FUNCTION.**
50. NDF=C
51. 101 I=1,N
52. X(I)=FX(I)+D(I)
53. LOST=0
54. CALL OBRES(X,Y,G,H)
55. IF(LOST-1) 62,52,52
56. IF(Y-FY) 55,49,49
57. CALL BACK(X,X,Y,G,H)
58. NOITH=NOITH+1
59. NOBP=NOBP+1
   **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK).** LOST=1 MEANS THE
   RETURNED POINT IS INFEASIBLE.
60. IF(LOST-1) 56,150,56
61. IF(NOCUT-5*MULT) 54,54,54
62. NOCUT=NOCUT+1
63. MULT=MULT+1
64. LOST=0
   **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK).** LOST=NE. 0 MEANS
   THE ENTERED POINT IS NEAR-FEASIBLE.
65. IF(ICHFCK-1) 64,140,140
66. CALL PENAT(G,H,PENA1,PENA2)
67. P=Y+**PENA1+3**(-0.5)*PENA2
68. IF(P-FP) 89,68,68
69. X(I)=FX(I)-D(I)
70. LOST=0
71. CALL OBRES(X,Y,G,H)
72. IF(LOST-1) 80,80,70
73. IF(Y-FY) 73,69,69
74. CALL BACK(X,X,Y,G,H)
75. NOITH=NOITH+1
76. NOBP=NOBP+1
   **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK).** LOST=1 MEANS THE
C RETURNED POINT IS INFEASIBLE
    IF (LOST-1) 74,150,74
C **ADD NOCUT 1 CREDIT TO STOP THE CURRENT STAGE FASTER AFTER
C EVERY 5 VIOLATIONS MADE WITHIN THE STAGE ,
137  74 IF (NDEC=5*MULT) 73,75,75
138  75 NOCUT=NOCUT+1
139  76 MULT=MULT+1
140  79 LOST=0
C **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK) ( LOST .NE. 0 MEANS
C THE ENTERED POINT IS NEAR-FEASIBLE )
141  83 IF (ICHECK-1) 82,140,140
142  82 CALL PENAT(G,H,PENA1,PENA2)
143  83 P=Y+R*PENA1+R**(-0.5)*PENA2
144  84 IF (P-FP) 88,96,96
145  86 X(I)=FX(I)
146  87 NOF=NOF+1
147  89 GO TO 99
148  83 FX=V
149  84 FP=P
150  85 NOITP=NOITP+1
151  86 FX(I)=X(I)
152  90 IF (MG) 94,94,90
153  94 DD 92 JJ=1,MG
154  92 FG(JJ)=G(JJ)
155  94 IF (MH) 99,99,96
156  99 DD 98 KK=1,MK
157  98 FHI(KK)=H(KK)
C **CHECK THE STAGE STOPPING CRITERION IS SATISFIED OR NOT .
158  99 IF (INOCUT-INCUT) 100,150,150
159  100 IF (ICHECK-1) 101,150,150
160  101 CONTINUE
161  102 IF (NOF-M) 103,104,104
C **CUT STEP-SIZES FOR MINIMIZING THE P-FUNCTION .
162  103 DD 106 I=1,N
163  106 D(I)=0.5*V(I)
164  107 NOCUT=NOCUT+1
165  108 [DIFF=IDIFF+1]
166  109 IF (IDIFF-INCUT) 50,107,107
167  107 INCUT=INCUT*2
168   GO TO 50
169  108 NOEXP=NOEXP+1

C **MAKE PATTERN MOVE FOR MINIMIZING THE P-FUNCTION**
170   DO 110 I=1,N
171    PX(I)=FX(I)+(FX(I)-BX(I))
172   110 BX(I)=FX(I)
173    LOST=0
174   CALL ORRES(PX,Y,G,H)
175    IF(LOST) 124,124,112
176    112 IF(Y-FY) 114,50,50
177   114 CALL BACK(PX,X,Y,G,H)
178    NOITR=NOITR+1
179    NOBP=NOBP+1

C **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST=1 MEANS THE
C RETURNED POINT IS INFEASIBLE )
180   IF(LOST) 115,150,115

C **ADD NOCUT 1 CREDIT TO STOP THE CURRENT STAGE FASTER AFTER
C EVERY 5 VIOLATIONS MADE WITHIN THE STAGE .
181   115 IF(NOBP-5>MULT) 120,116,116
182   116 NOCUT=NOCUT+1
183   117 MULT=MULT+1
184   120 LOST=0

C **CHECK THE ITMAX IS EXCEEDED OR NOT IN (BACK)( LOST NE. 0 MEANS
C THE ENTERED POINT IS NEAR-FEASIBLE )
185   122 IF(INCHECK) 124,140,140
186   124 CALL PENA(G,H,PENA1,PENA2)
187    P=Y+R*PENA1+R**(-0.5)*PENA2
188    IF(P<FP) 128,49,49
189   128 NOPAT=NOPAT+1
190   129 NOITP=NOITP+1
191   129 DO 129 II=1,N
192   129 FX(II)=PX(II)
193   130 IF(MG) 133,133,131
194   131 DO 132 J=1,MG
195   132 FG(J)=G(J)
196   133 IF(MH) 136,136,134
197   134 DO 135 K=1,MH
135 FH(K) = H(K)
136 FY = Y
137 FP = P

C

C **CHECK THE STAGE STOPPING CRITERION IS SATISFYED OR NOT.
C IF(NOcut-INCUT) 138,150,150
C IF(CHECK-1) 138 50,150,150

C

C **CHECK THE ITMAX EXCEEDED POINT ( WHEN IT IS returned FROM BACK )
C IS BETTER OR NOT AND SET PROPER STAGE-OPTIMUM.
C
C
C CALL ORES(X,Y,G,H)
C CALL PENAT(G,H,PENA1,PENA2)
C P = Y + R * PENA1 + R ** (-0.5) * PENA2
C IF(P-PF) 142,150,150
C DO 144 I=1,N
C 144 FX(I) = X(I)
C GO TO 137

C

C **SET THE SUB-OPTIMUM GOT BEFORE ENTERED TO BACK BE THE
C STAGE-OPTIMUM.

C

150 NOPULL = 0
151 IPULLS = 0
152 PULL = 3.63
153 CALL WEIGH(TGH, MG, FG)
154 IF(TGH) 170, 170, 162
155 160 CALL WEIGH(TGH, MG, FG)

C

C **CHECK THE STAGE OPTIMUM IS FEASIBLE OR NOT.
C IF(TGH) 170, 170, 162

C

C **PULL BACK THE INFEASIBLE STAGE-OPTIMUM INTO THE FEASIBLE REGION
C DO 163 I = 1, N
C 163 FX(I) = PULL * (FX(I) - OX(I)) + OX(I)
C NOPULL = NOPULL + 1
C CALL ORES(FX, FY, FG, FH)
C NOITB = NOITB + 1
C IF(NOPULL-5) 168, 164, 164
C 164 NOPULL = 0
C IPULLS = IPULLS + 1
C PULL = PULL / 2.0
C IF(IPULLS-4) 160, 160, 165
C 165 DO 166 I = 1, N
166 FX(I)=OX(I)
C **CHECK THE POINT IS ON THE BOUNDARY OR NOT.
229 168 IF(IR-1)=160,162,162
230    170 LOST=0
231 CALL PENAT(FG,FH,PENA1,PENA2)
232    FP=FY+R PENA1+R**(-0.5)*PENA2
233    200 NOIT=NOIT+ITER
234 YSTOP=ABS( FY/(FY-R PENA1+R**(-0.5)*PENA2))
235 YSTOP=ABS( YSTOP-1.0)
236 WRITE(3,1008) NOR, FY, FP, R, ITER, NOIT, NOBP, NOITP, NOITB, NOEXP,
1NOPAT, NOCUT, YSTOP
237 WRITE(3,1006) (I, FX(I), I, D(I), I=1,N)
238 WRITE(3,1011)
239 215 WRITE(3,1012) (J, FG(J), J=1,MG)
240 216 IF(MH) 218,218,217
241 217 WRITE(3,1013) (K, FH(K), K=1,MH)
242 218 CALL OUTPUT(N,MG,MH)
243 WRITE(3,1007)
244 IF(YSTOP-TETA) 226,226,220
C **CHECK THE MAXP IS EXCEEDED OR NOT.
245 220 IF(NOR-MAXP) 221,227,227
C **STORE LAST SUB-OPTIMUM POINT.
246 221 DO 222 I=1,N
247 222 OX(I)=FX(I)
C **SHIFT TO THE NEXT STAGE SEARCH.
248 R=R/RATIO
249 FP=FY+R PENA1+R**(-0.5)*PENA2
250 NOR=NOR+1
251 IF(NOR-5*MP) 224,224,223
252 223 INCUT=INCUT+1
253 MP=MP+1
254 IDP=IDP+5
255 224 NOBP=0
256 MULT=1
257 NOITB=0
258 NOITP=0
259 ISK=0
260 NOEXP=0
261 NOPAT=0
262 NOCUT=0
ITER=0
IB=0
FNOR=NCR+IDR

C
R=0
DO 225 I=1,N
D(I)=PD(I)/FNOR
225 R=R+C(I)
R=R/FN
GO TO 50
226 WRITE(2,1015)
GO TO 228
227 WRITE(3,1016) MAXP
228 IDPM=IDP+1
229 IF(IDPM-NORM) 1,1,230
230 STOP
END
SUBROUTINE BACK (XB, X, Y, G, H)
C THIS SUBROUTINE PULLS INFEASIBLE POINTS BACK INTO THE
C FEASIBLE OR NEAR-FEASIBLE REGION.
C
C **DEFINITION**
C FEASIBLE . . . ALL G(I) .GE. 0.
C NEAR-FEASIBLE . . (B-TGH) .GE. 0.
C
C DIMENSION XB(20), X(20), G(20), H(20), D(20)
C COMMON /BLOGY/ N, MG, MH, ITER, ITMAX, ICHECK, IB, LOST
C COMMON /BLOGB/ NOITP, NOITB, B, D
C CALL WEIGHT(TGH, MG, G)
C IF(TGH) 8, 8, 4
C
C **DECREASE THE VALUE OF B IN RETURN**
C 4 IF(B-TGH) 12, 12, 6
C 6 IF(0.7*B-TGH) 10, 8, 8
C 8 B =0.75*B
C 10 LOST=0
C 28 RETURN
C 12 FTGH=TGH
C
C **MAKE EXPLORATORY MOVE FOR MINIMIZING TGH**
C 22 NOF=0
C 24 DO 38 ND=1, N
C 26 XB(NB)=XB(NB)-0.5*D(NB)
C 28 CALL OBRES(XB, Y, G, H)
C 30 CALL WEIGHT(TGH, MG, G)
C 32 IF(TGH) 24, 24, 26
C 24 NOITP=NOITP+1
C 25 LOST=0
C 29 GO TO 46
C 30 NOITB=NOITB+1
C 31 IF(ICHECK-1) 27, 45, 45
C 32 IF(TGH-FTGH) 29, 32, 32
C 33 FTGH=TGH
C 34 IF(B-TGH) 38, 38, 25
C
C 35 XB(NP)=XB(NP)+D(NP)
C 36 CALL OBRES(XB, Y, G, H)
CALL WEIGH(TGH, MG, G)
IF(TGH) 24, 24, 34
309  34 NOITH=NOITH+1 -
310  IF(ICHECK=1) 35, 45, 45
311  35 IF(TGH=FTGH) 28, 36, 36
312  36 XB(NB)=XB(NB)-0.5*D(INB)
313  NOF=NOF+1
314  38 CONTINUE
315  IF(NOF-N) 22, 42, 42

C
C  **CUT STEP-SIZES FOR MINIMIZING TGH .
316  42 DO 44 IR=1,N
317  44 D(IR)=0.5*U(IR)
318  44 CONTINUE
319  GO TO 22
320  45 LOST=1

C
C  **SET BASE POINT TO RETURN .
321  46 DO 50 NR=1,N
322  50 D(NR)=D(NR)*0.5
323  52 X(NR)=XB(NR)
C  **DECREASE THE VALUE OF B IN RETURN .
324  IF(0.7*B-TGH) 60, 58, 58
325  58 R=0.75*R
326  60 RETURN
327  END
SUBROUTINE PENAT(G,H,PENAL,PENAO)
C  THIS SUBROUTINE COMPUTES THE PENALTY TERMS FOR SUMT FORMULATION.
C     PENAL1 FOR INEQUALITY CONSTRAINTS
C     PENAO2 FOR EQUALITY CONSTRAINTS
C
DIMENSION G(2N),H(2N)
COMMON /BLOGY/ N,NG,MH,ITER,ITMAX,ICHECK,IB,LOST

PENAL1=0,
PENAO2=0,

IF(MG) 5,5,1
1  DO 4 I=1,NG
2   IF(G(I)) 4,7,4
3
C     **SET G(I)=0,1E-48 WHEN G(I)=0. ( ON THE BOUNDARY )
4  G(I)=0,1E-48
5  PENAL1=PENAL1+ABS(1.00/G(I))
6  IF(MH) 10,10,6
7  DO 9 K=1,MH
8   IF(H(K)) 9,8,9
9
C     **SET H(K)=0,1E-48 WHEN H(K)=0.
8  H(K)=0,1E-48
9  PENAO2=PENAO2+H(K)**2
10  CONTINUE
10  RETURN
END
SUBROUTINE WEIGH(TGH, MG, G)

C THIS SUBROUTINE COMPUTES THE TOTAL WEIGHT OF VIOLATION
C TO THE INEQUALITY CONSTRAINTS.

C

346 DIMENSION G(20)
347 TGH=0.
348 IF(MG) 4,4,1
349 1 DD 3 IR=1, MG
350 IF(G(IR)) 2,3,3
351 2 TGH=TGH+G(IR)**2
352 3 CONTINUE
353 4 TGH=TGH**0.5
354 RETURN
355 END
SUBROUTINE READIN(N, MG, MH)
C THIS SUBROUTINE IS FOR READ IN ADDITIONAL DATA.
C USER SUPPLIES HIS OWN READ STATEMENT AND FORMAT.
C ARGUMENTS N,MG,MH ARE NUMBERS OF VARIABLES, OF INEQUALITY CONSTRAINTS
C AND OF EQUALITY CONSTRAINTS.
C COMMON /DLOGR/ ....... STATEMENT IS FOR TRANSFER DATA USE.
C
357 COMMON /DLOGR/ Q(10)
358 RETURN
359 END
SUBROUTINE OUTPUT(N,MG,MH)
C THIS SUBROUTINE IS FOR USER TO PRINT OUT ADDITIONAL INFORMATION
C WANTED. ARGUMENTS N,MG,MH ARE NUMBERS OF VARIABLES,OF INEQUALITY
C CONSTRAINTS, AND OF EQUALITY CONSTRAINTS.
C COMMON /BLOGO/ IS FOR TRANSFER NEED DATA IN MAIN TO
C THE SUBROUTINE OUTPUT.
C USER SUPPLIES ALL NECESSARY FORMATS.
C
361 DIMENSION G(20)
362 COMMON /BLOGO/ G
363 RETURN
364 END
SUBROUTINE ORES(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.
G(J) = ******** J FROM 1 TO MG, FOR CONSTRAINTS G(J) GT 0.0.
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
DIMENSION X(20),G(20),H(20),Q(10)
COMMON /BLOGY/ N, MG, MH, ITER, ITMAX, ICHECK, IP, LOST
COMMON /DPLOGR/ 0
100 FORMAT(3X25H**THE ITERATION EXCEEDED 15,14,1)
C
********* ********** STATEMENT NUMBERS 1,2,3,4,5,6,7,8,100 HAVE BEEN USED.
C
C
C
************ ********** ********** ********** ********** ********** ********** **********
ITER=ITER+1
IF(ITER-ITMAX) 3,1,2
C
**OUTPUT THE MESSAGE OF ITMAX EXCEEDED.
1 PUNCH 100, ITMAX
2 ICHECK=1
C
**CHECK FOR THE VIOLATION TO INEQUALITY CONSTRAINTS.
3 IB=0
4 IF(MG) 8,8,4
5 DO 7 I=1, MG
7 CONTINUE
6 IB=1
10 CONTINUE
9 RETURN
8 END

OPTIMIZATION OF INDUSTRIAL MANAGEMENT SYSTEMS
BY THE SEQUENTIAL UNCONSTRAINED
MINIMIZATION TECHNIQUE

by

KEETING CHIECHENG LAI

B.S. Chong-Uan College of Science and Technology
Taw-Uan, Formosa, 1965

AN ABSTRACT OF A MASTER'S REPORT

Submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering
Kansas State University
Manhattan, Kansas

1970
The problems considered in this report are optimization of system reliability of a complex system and optimization of production scheduling and inventory control subject to some linear and/or nonlinear constraints. The optimization method employed is the sequential unconstrained minimization technique (SUMT).

The purposes of this report are twofold. The first is to present a result of implementing SUMT by a combination of the Hooke and Jeeves pattern search technique and a heuristic programming technique. The second is to present results of employing the developed technique to the optimization of system reliability of a complex system and production scheduling and inventory control problems.

There is a general computer program available entitled "RAC Computer program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming". In this computer program, the unconstrained minimization technique used is the second order gradient method. Difficulties which arise from use of the second order gradient method as a unconstrained minimization technique in SUMT becomes predominate in a large size and/or very complex nonlinear problem. The difficulties arise particularly in taking correctly the first order and second order partial derivatives of complex nonlinear functions which most of practical problems have. Therefore, a new algorithm which using a much simpler direct search technique is very desirable. For this reason, a new technique of implementing SUMT by the Hooke and Jeeves pattern search technique to be its unconstrained minimization has been developed.

This newly developed method is utilized to obtain the optimum solutions of two examples of production scheduling and inventory control problems. The first problem is a simple two dimensional problem used for
demonstrating the procedure of the algorithm in details and the second problem is a 20-dimensional problem used for demonstrating the capacity and practicability of the technique.

The problem of optimizing a system reliability becomes considerably more difficult when the redundant units of the system cannot be reduced to pure parallel or series configurations. In such a complex system the system reliability is obtained by Bayes' theorem which utilizes conditional probabilities. A mathematical model for the nonlinear system reliability subject to constraints is formulated. The nonlinear programming problem of optimizing the system reliability is then solved by SUNT using RAC computer program and by the newly developed technique and the results are compared.