A COMPARATIVE STUDY OF OPTIMIZATION TECHNIQUES APPLIED TO INDUSTRIAL MANAGEMENT SYSTEMS

by 5O3

VIJAY H. MEHTA

B.S.(MECH. ENGG.), Sardar Patel University
Vallabh Vidyanagar, India, 1967

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

1969

Approved by:

Ching-Lai Hwang
Major Professor

TABLE OF CONTENTS

1. INTRODUCTION

There are several optimization techniques available for the various types of optimization problems faced by the management of the modern industries. The search techniques are considered to be efficient procedures among these optimization techniques. The search techniques are contrasted as alternate ways of solving problems to the usual available algorithmic techniques of operations research such as linear programming [6], dynamic programming [1] and the maximum principle [3]. The well known search procedures for multivariables optimization problems are Powell's method [17], gradient methods [19], Fletcher and Powell method [4], Fletcher and Reeves method [5], Hooke and Jeeves pattern search [9] and simplex pattern search [16].

In recent years some of these techniques have been applied in some of the industrial management systems. The effectiveness and behavior of these techniques are entirely depend upon the types and situations of the problems to which they are applied. Each technique claims its superiorarity in certain conditions and in certain situations.

The purpose of this study is to compare the behavior of some of the search techniques for optimization under identical conditions. In this report a comparison of the four well known unconstrained optimization techniques is presented. The four selected techniques are gradient technique, simplex pattern search, Fletcher and Powell method and Fletcher and Reeves method. To see the effect of these techniques on the dimensionality of the optimization problem, each technique is applied to two test problems. One of them is two dimensional problem and another is twenty dimensional problem. Thus it provides the comparison of each technique with other techniques and the effect of each technique on the dimensionality of the problem.

The production and inventory control and the aggregate production and employment scheduling represent the typical problems of the industrial management systems. For this reason they are selected as test problems in this study. The first test problem is a two period production planning problem in which the objective is to determine the optimum production level at each period such that the total operating cost is minimized. The cost is composed principally of the sum of the production cost and inventory cost. This model with 5 stages of planning period was solved by Hwang, et. al. [11] using the discrete maximum principle.

The well known Holt, Modigliani, Muth and Simon [8] paint factory model with planning horizon of ten months is selected as a second test problem. There are two decision variables at each month, namely, production rate and workforce level which are to be determined so as to minimize the total cost. The model was solved by Holt, Modigliani, Muth and Simon [8] using linear decision rule approach. It was also solved by Taubert [21] using Hooke and Jeeves pattern search. A similar model with 5 stages was solved by Hwang, Tillman and Fan using the discrete maximum principle [11] and using the sequential simplex pattern search [3a].

The gradient technique, simplex pattern search, Fletcher and Powell method and Fletcher and Reeves method are described in section 3, 4, 5 and 6 respectively together with the results of test problems. A comparison and discussion of the results obtained by each technique is presented in section 7.

Four different criteria are used to compare the behavior and convergence of these four techniques. They are the optimum function value, the total computation time, number of iterations and required computer memory storage.

2. TEST PROBLEMS

To campare the behavior and effectiveness of these four optimization techniques, namely, gradient technique, simplex pattern search, Fletcher and Powell method and Fletcher and Reeves method, they are applied to two problems of production planning system. It is also desired to study the effect of each technique on the dimensionality of the problem. For this purpose one of the test problems considered is two dimensional production planning problem and another problem is twenty dimensional production and employment scheduling problem.

A. Two dimensional production planning problem.

This problem is a two periods production scheduling problem in which the objective is to minimize the operating cost for the planning periods. The cost is composed principally of the sum of the production cost and inventory cost. Figure 1 represents the schematic diagram of this problem.

θ_1 and θ_2 represent the production rate at each period respectively. Q_1 and Q_2 are the given rate of sales at each period. I_1 and I_2 represent the inventory at the end of each period and I_0 is the given initial inventory level. The recurrence relationship of the inventory is given by

$$I_1 = I_0 + \theta_1 - Q_1$$

and $$I_2 = I_1 + \theta_2 - Q_2$$

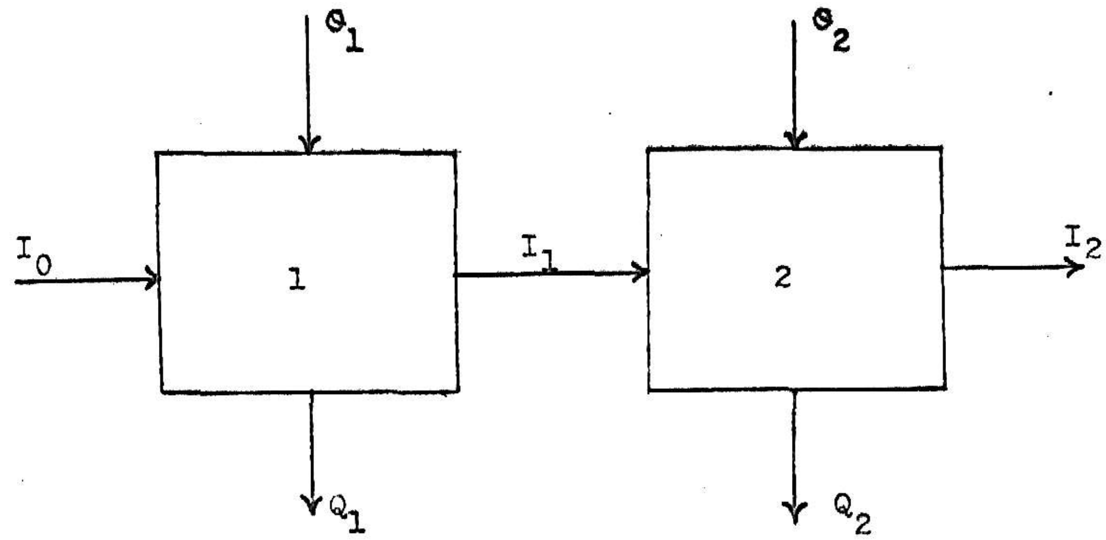


FIG. 1. BLOCK DIAGRAM FOR TWO DIMENSIONAL PROBLEM

The objective function of the problem is assumed to be

$$S = C(\theta_1 - \theta_0)^2 + D(E - I_1)^2 + C(\theta_2 - \theta_1)^2 + D(E - I_2)^2$$

where C, D, and E are given constants.

The problem is to determine the optimal production rate at each period, θ_1 and θ_2, such that the objective function S is minimized. It is obvious that the production rate at each stage should be positive, therefore, $\theta_n \geq 0$, $n = 1, 2$. Further more it is also given that the back log of orders are permitted that is, negative inventory values are allowed in this problem.

Numerical values of the model are given as follows.

Initial inventory level = $I_0 = 12$

Initial production rate = $\theta_0 = 15$

Sales rate at first period = $Q_1 = 30$

Sales rate at second period = $Q_2 = 10$

$C = 100$

$D = 20$

$E = 10$.

B. Twenty dimensional production and employment scheduling problem.

The well Known Holt, Modigliani, Muth and Simon [8] paint facroty model is selected as a second test problem. This model considers the production and inventory system with two independent variables in each planning period. The schematic of the problem is shown in Figure 2.

One pair of the independent variables is used to represent

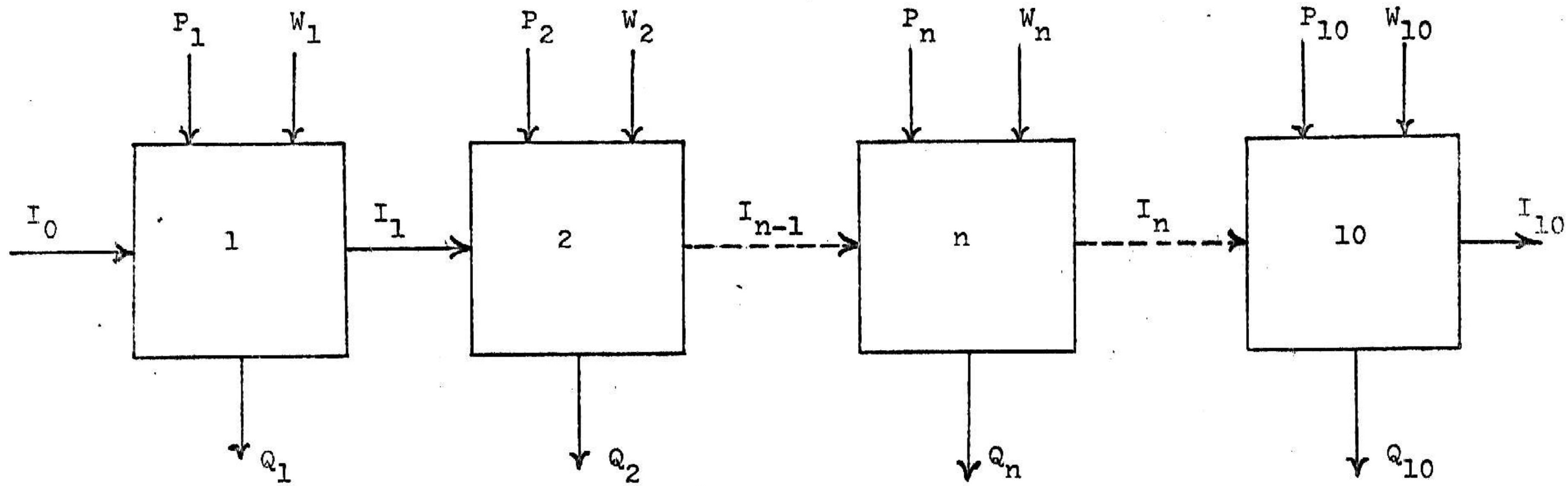


FIG. 2. BLOCK DIAGRAM FOR TWENTY DIMENSIONAL PROBLEM

the production rate and work force level at each month. The problem is to determine the optimal production rate and work force level such that the total operating cost for the 10 months planning horizon is minimized.

Let us define

n = a month in the planning horizon

N = the duration, in months = 10

P_n = production rate at the nth month

W_n = work force level in the nth month

Q_n = sale rate at the nth month

I_n = inventory level at the end of the nth month

Inventory level at the end of each month is computed using the recursive relationship between sales, production and inventory as follows

$$I_n = I_{n-1} + P_n - Q_n, \qquad n = 1, 2, \ldots, N.$$

The model considers that the total operating cost consist of following four cost items.

1. Regular payroll cost, i.e., direct labour cost.
2. Hiring and layoff cost.
3. Overtime cost.
4. Inventory cost.

These individual cost components of this model are given as follows:

1. Regular payroll cost = $340.0\ W_n$

2. Hiring and layoff cost = $64.3\ (W_n - W_{n-1})^2$

3. Overtime cost = $0.2(P_n - 5.67W_n)^2 + 51.2P_n - 281.0W_n$

4. Inventory cost = $0.0825\ (I_n - 320.0)^2$

It is assumed that backlog of orders or negative inventories are permitted.

The decision problem can now be stated as to choose the optimum values for production rate, P_n, and workforce level, W_n, at each month of the planning horizon such as to minimize the total cost S_N which is given by

$$S_N = \sum_{n=1}^{N} S_n$$

where

$$\begin{aligned} S_n = [340.0W_n] &+ [64.3(W_n - W_{n-1})^2] \\ &+ [0.20(P_n - 5.67W_n)^2 + 51.2P_n - 281.0W_n] \\ &+ [0.0825(I_n - 320.0)^2] \end{aligned}$$

Here 10 months planning period has been considered. Therefore, there are ten variables for the production rate and ten for the workforce level. Hence the system which we are considering is a twenty dimensional minimization problem.

The numerical data of the model is given as follows:

$Q_1 = 430$, $Q_6 = 375$,
$Q_2 = 447$, $Q_7 = 292$,
$Q_3 = 440$, $Q_8 = 458$,
$Q_4 = 316$, $Q_9 = 400$,
$Q_5 = 397$, $Q_{10} = 350$,

Initial inventory = I_0 = 263.0

Initial workforce level = W_0 = 81.0

3. GRADIENT TECHNIQUE

The gradient direction is the best searching direction for locating a minimum of a function. The method of steepest descent has been used for many years for finding a minimum value of a function. The main disadvantage with the method of steepest descent is the requirement that each new direction be normal to the old direction. Various modifications have been proposed to improve the original method of steepest descent. Rosenbrock and Storey [19] describes many of these modifications in their book and gradient method is one of these modifications.

To begin the search for a minimum by using the gradient method, the direction of steepest descent which is negative of the gradient direction is determined and then a step of length δ is taken in this direction. The process is continued by again locating the direction of steepest descent and taking a step of certain step size δ in that direction. There are several versions of the gradient method which are different in determining this step size. One of these versions of the gradient methods is presented.

The gradient technique which locates the minimum of a function of several variables is very fast converging method when the trial points are far from the optimum. One of the limitations for this particular method is that it is only useful for unconstrained minimization problems.

Let us consider an optimization problem which is at steady state and represented by the following system of equations.

$$\left.\begin{array}{l} T_1(w_1;\ x_1,\ \dots,\ x_s;\ \theta_1,\ \dots,\ \theta_r) = 0 \\ T_2(w_2;\ x_1,\ \dots,\ x_s;\ \theta_1,\ \dots,\ \theta_r) = 0 \\ \vdots \\ T_s(w_s;\ x_1,\ \dots,\ x_s;\ \theta_1,\ \dots,\ \theta_r) = 0 \end{array}\right\} \qquad (1)$$

or in the vector form

$$T(w;\ x;\ \theta) = 0 \qquad (1a)$$

where w is a given constant, x is a s-dimensional vector representing the state of the system and θ is an r-dimensional vector representing the decision.

Let θ^* be a trial decision vector, then the corresponding state vector x^* can be obtained from equation (1a) such as

$$T(w,\ x^*,\ \theta^*) = 0 \qquad (2)$$

If the decision vector is perturbed arbitrarily but slightly from the trial value (It is desired to insure that perturbations in the control vector are small enough that linearization is valid), that is,

$$\theta = \theta^* + \epsilon\psi \qquad (3)$$

and the resulting perturbation of state vector is

$$x = x^* + \epsilon y \qquad (4)$$

where $\epsilon\psi$ and ϵy represent the small perturbations of the decision vector and the state vector. The θ and x presented by equations (3) and (4) also satisfy equation (1). Then a Taylor series expansion of equation (1) around x^* and θ^* gives (neglecting the second and higher order terms)

$$T(w, x, \theta) = T(w, x^*, \theta^*)$$

$$+ \frac{\partial T(w, x^*, \theta^*)}{\partial x} \epsilon y + \frac{\partial T(w, x^*, \theta^*)}{\partial \theta} \epsilon \psi \tag{5}$$

Therefore we obtain

$$\frac{\partial T(w, x^*, \theta^*)}{\partial x} \epsilon y + \frac{\partial T(w, x^*, \theta^*)}{\partial \theta} \epsilon \psi = 0 \tag{6}$$

or in short

$$\left(\frac{\partial T}{\partial x}\right)^* \epsilon y + \left(\frac{\partial T}{\partial \theta}\right)^* \epsilon \psi = 0 \tag{7}$$

where

$$\left(\frac{\partial T}{\partial x}\right)^*_{sxs} = \begin{bmatrix} \left(\frac{\partial T_1}{\partial x_1}\right)^* & \left(\frac{\partial T_1}{\partial x_2}\right)^* & \cdots & \left(\frac{\partial T_1}{\partial x_s}\right)^* \\ \vdots & & & \\ \left(\frac{\partial T_s}{\partial x_1}\right)^* & \left(\frac{\partial T_s}{\partial x_2}\right)^* & \cdots & \left(\frac{\partial T_s}{\partial x_s}\right)^* \end{bmatrix} \tag{8}$$

$$\left(\frac{\partial T}{\partial \theta}\right)^*_{sxr} = \begin{bmatrix} \left(\frac{\partial T_1}{\partial \theta_1}\right)^* & \left(\frac{\partial T_1}{\partial \theta_2}\right)^* & \cdots & \left(\frac{\partial T_1}{\partial \theta_r}\right)^* \\ \vdots & & & \\ \left(\frac{\partial T_s}{\partial \theta_1}\right)^* & \left(\frac{\partial T_s}{\partial \theta_2}\right)^* & \cdots & \left(\frac{\partial T_s}{\partial \theta_r}\right)^* \end{bmatrix} \tag{8a}$$

In general, the performance index (or the objective function) can be expressed by

$$\phi(x_1, x_2, \ldots, x_s) = \phi(x) \tag{9}$$

In reality the performance index may include the decision vector, however the system can be transformed into the system represented by equation (9) as follows:

$$\left.\begin{array}{l} T_1\ (w_1;\ x_1, \ldots, x_s;\ \theta_1, \ldots, \theta_r) = 0 \\ T_2\ (w_2;\ x_1, \ldots, x_s;\ \theta_1, \ldots, \theta_r) = 0 \\ \vdots \\ T_s\ (w_s;\ x_1, \ldots\ x_s;\ \theta_1, \ldots, \theta_r) = 0 \end{array}\right\} \text{s-system equations}$$

$$\phi(x_1, \ldots, x_s;\ \theta_1, \ldots, \theta_r) = \phi(x, \theta)$$

The above original systems equations are transformed to

$$\left.\begin{array}{l} T_1\ (w_1;\ x_1, \ldots, x_s;\ \theta_1, \ldots, \theta_r) = 0 \\ T_2\ (w_2;\ x_1, \ldots, x_s;\ \theta_1, \ldots, \theta_r) = 0 \\ \vdots \\ T_s\ (w_s;\ x_1, \ldots, x_s;\ \theta_1, \ldots, \theta_r) = 0 \\ T_{s+1}(x_{s+1};\ \theta_1) = x_{s+1} - \theta_1 = 0 \\ \vdots \\ T_{s+r}(x_{s+r};\ \theta_r) = x_{s+r} - \theta_r = 0 \end{array}\right\} \text{(s+r) system equations}$$

$\phi(x_1, \ldots, x_s, x_{s+1}, \ldots, x_{s+r})$ the new performance index

Consider now adjoining the system equation, equation (1) as an equality constraint with the objective function, equation (9). This gives

$$\phi = \phi(x) + \lambda_\phi^T T \tag{10}$$

where λ_ϕ is Lagrangian multiplier and superscript T is the transpose of the column matrix. The problem is transformed from the extremization of equation (9) subject to constraint given by equation (1) to the extremization of equation (10).

Taking the first variations on the objective function, equation (10) gives

$$d\phi = \left[\frac{\partial \phi}{\partial x} + \lambda_\phi^T \frac{\partial T}{\partial x}\right] \in y + \lambda_\phi^T \frac{\partial T}{\partial \theta} \in \psi \tag{11}$$

where

$$\frac{\partial \phi}{\partial x} = \left[\frac{\partial \phi}{\partial x_1}, \frac{\partial \phi}{\partial x_2}, \ldots, \frac{\partial \phi}{\partial x_s}\right]$$

From the trial decision vector, θ^*, and the corresponding state vector, x^*, we can calculate $(\frac{\partial \phi}{\partial x})^*$, $(\frac{\partial T}{\partial x})^*$ and $(\frac{\partial T}{\partial \theta})^*$ in equation (11). The unknown Lagrangian multiplier, λ_ϕ, in equation (11) can be chosen so that

$$\left(\frac{\partial \phi}{\partial x}\right)^* + \lambda_\phi^T \left(\frac{\partial T}{\partial x}\right)^* = 0 \tag{12}$$

therefore, equation (11) becomes

$$d\phi = \lambda_{\phi}^{T} \left(\frac{\partial T}{\partial \theta}\right)^{*} \epsilon\psi \tag{13}$$

At the optimal condition

$$d\phi = 0 \tag{14}$$

however, $d\phi \neq 0$, in general.

The gradient technique is an iterative method which starts from a trial point $(x^{*};\ \theta^{*})$ and decides a proper $\epsilon\psi$ that gives the greatest change in $d\phi$ so that $d\phi \to 0$. However, it is desirable to insure that perturbations in the control vector, $\epsilon\psi$, are small enough that linearization leading to equations (5) and (11) is valid. $\epsilon\psi$ is a step size defined earlier as δ. Let

$$(dp)^{2} = (\epsilon\psi)^{T}\ W\ (\epsilon\psi)$$

or

$$(dp)^{2} = W_{1}(\epsilon\psi_{1})^{2} + \ldots\ldots + W_{r}(\epsilon\psi_{r})^{2} \tag{15}$$

be a positive definite quadratic form with W, a matrix of suitably chosen weighting factors and dp a scalar which is specified to limit the magnitude of the perturbations. W is a (rxr) matrix in general, however, a diagonal matrix is used. Equation (15) is introduced into equation (13) in terms of an undetermined Lagrangian multiplier α as follows:

$$d\phi + \lambda_{\phi}^{T} \left(\frac{\partial T}{\partial \theta}\right)^{*} \epsilon\psi + \alpha \left[(dp)^{2} - (\epsilon\psi)^{T} W (\epsilon\psi)\right] \tag{16}$$

In order to attain the maximum rate of change of $d\phi$ with respect to $\epsilon\psi$, equation (16) is maximized by differentiating with respect to $\epsilon\psi$ and equating the result to zero. This yields

$$\frac{d(d\phi)}{d(\epsilon\psi)} = \lambda_{\phi}^{T} \left(\frac{\partial T}{\partial \theta}\right)^{*} - 2\alpha\, (\epsilon\psi)^{T} W = 0 \tag{17}$$

or

$$\epsilon\psi = \frac{1}{2\alpha} W^{-1} \left(\frac{\partial T}{\partial \theta}\right)^{*T} \lambda_{\phi} \tag{18}$$

substituting equation (18) into equation (15) gives

$$2\alpha = \pm \left[\frac{\lambda_{\phi}^{T} \left(\frac{\partial T}{\partial \theta}\right)^{*} W^{-1} \left(\frac{\partial T}{\partial \theta}\right)^{*T} \lambda_{\phi}}{(dp)^{2}} \right]^{\frac{1}{2}} \tag{19}$$

If dp is given, 2α is obtained from equation (19) and then $\epsilon\psi$ is obtained from equation (18).

Finally, in the iteration procedure, the new trial value becomes

$$\theta^{*}_{new} \oplus \theta^{*}_{old} + \epsilon\psi \tag{20}$$

The determination of the optimal dp for this gradient procedure is a very difficult task. According to Sage [20], there is some merit in adjusting dp, and a practically efficient method consists of using the past value of dp, one-half the past value, and two and ten times the past value of dp in order to determine α in equation (19), which in return determines θ^{*}_{new}. The resulting four values of θ^{*}_{new} are then used to determine x and ϕ, the performance index. The value of dp($\frac{1}{2}dp_{old}$, dp_{old}, $2dp_{old}$, or $10dp_{old}$) which produces the smallest ϕ is then used for the next iteration by the gradient method.

A. Application to two dimensional production scheduling problem.

The function F to be minimized is given by

$$F = C(\theta_1 - \theta_0)^2 + D(E - I_1)^2 + C(\theta_2 - \theta_1)^2 + D(E - I_2)^2$$

The problem here is to find optimal schedule of the production level θ_1 and θ_2 such that the total cost, F, is minimized.

To convert the problem into standard procedure of the gradient technique, we define

$$\begin{aligned}
x_1 &= I_1 = I_0 + \theta_1 - Q_1 \\
x_2 &= I_2 = I_1 + \theta_2 - Q_2 \\
x_3 &= \theta_1 \\
x_4 &= \theta_2
\end{aligned}$$

Hence system equations can be written as follows:

$$\begin{aligned}
T_1 &= x_1 - I_0 - \theta_1 + Q_1 = 0 \\
T_2 &= x_2 - x_1 - \theta_2 + Q_2 = 0 \\
T_3 &= x_3 - \theta_1 = 0 \\
T_4 &= x_4 - \theta_2 = 0
\end{aligned}$$

From the given function F, performance index $\phi(x)$ can be written as

$$\begin{aligned}
\phi(x_1, x_2, x_3, x_4) = {} & C(x_3 - \theta_0)^2 + D(E - x_1)^2 \\
& + C(x_4 - x_3)^2 + D(E - x_2)^2
\end{aligned}$$

From the systems equations and performance index, it is seen that $(\frac{\partial T}{\partial x})^*$ is a 4 x 4 matrix, $(\frac{\partial T}{\partial \theta})^*$ is a 4 x 2 matrix, and $(\frac{\partial \phi}{\partial x})^* = \left[\frac{\partial \phi}{\partial x_1} \quad \frac{\partial \phi}{\partial x_2} \quad \frac{\partial \phi}{\partial x_3} \quad \frac{\partial \phi}{\partial x_4}\right]$, is a 1 x 4 matrix.

This technique is programmed in WATFOR for an IBM 360/50 system. The flowchart and the computer program is given in Appendix I.

Initial starting trail values for θ_1 and θ_2 are assumed to be

$$\theta^* = \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix} = \begin{bmatrix}10 \\ 10\end{bmatrix}$$

In the initial iteration, the trial value of dp = 1 was assumed, which in turn gave a set of four dp values as

0.5, 1, 2, and 10

Stopping criteria for computer program was used as

$$|F_{n+1} - F_n| \leq 0.01$$

After 11 iterations, the optimal answer was obtained upto an accuracy mentioned above. It was seen that near the optimal, convergence became slow.

The optimal answer for this problem is as follows

$\theta_1 = 17.82$

$\theta_2 = 18.22$

minimumF = \$2960.71

This problem consumed 16.10 seconds of computer time on an

IBM 360/50 computer. It required 9816 bytes of computer memory storage.

B. Application to twenty dimensional EMMS paint factory model.

As seen earlier in the section 2, the objective function to be minimized is given by

$$S = \sum_{n=1}^{10} S_n$$

where

$$S_n = 340.0W_n + 64.3(W_n - W_{n-1})^2 + 0.2(P_n - 5.67W_n)^2$$
$$+ 51.2P_n - 281.0W_n + 0.0825(I_n - 320.0)^2$$

To convert the problem into standard procedure of the gradient technique, let

θ_i; $i = 1, 2, \ldots, 10$, represent $P_i (i = 1, 2, \ldots, 10)$, the production rate at the ith stage,

θ_j; $j = 11, 12, \ldots, 20$, represent W_i $(i = 1, 2, \ldots, 10)$, the work force level at the ith stage

Further let us define

$$x_1 = I_1 = I_0 + \theta_1 - Q_1$$
$$x_2 = I_2 = I_1 + \theta_2 - Q_2$$
$$\vdots$$
$$x_{10} = I_{10} = I_9 + \theta_{10} - Q_{10}$$
$$x_{11} = \theta_1$$
$$x_{12} = \theta_2$$
$$\vdots$$

.
.

$x_{30} = \theta_{20}$

System equation for the problem can then be written as

$$T_1 = x_1 - I_0 - \theta_1 + Q_1 = 0$$
$$T_2 = x_2 - x_1 - \theta_2 + Q_2 = 0$$

.
.
.

$$T_{10} = x_{10} - x_9 - \theta_{10} + Q_{10} = 0$$
$$T_{11} = x_{11} - \theta_1 = 0$$
$$T_{12} = x_{12} - \theta_2 = 0$$

.
.
.

$$T_{30} = x_{30} - \theta_{20} = 0$$

From the given objective function, the performance index $\phi(x)$ can be written as

$$\phi_N(x) = \sum_{n=1}^{10} \phi_n(x)$$

$$\begin{aligned}\phi_n(x) = {} & 340.0[x(n+20)] + 64.3[x(n+20) - x(n+19)]^2 \\ & + 0.2[x(n+10) - 5.67x(n+20)]^2 \\ & + 51.2[x(n+10)] - 281.0[x(n+20)] + 0.0825[x(n)-320.0]^2\end{aligned}$$

In this case $(\partial T/\partial x)^*$ is 30 x 30 matrix; $(\partial T/\partial \theta)^*$ is 30 x 20 matrix and $(\partial \phi/\partial x)^*$ is 1 x 30 row matrix. The weighting matrix W is assumed to be an identity matrix of 20 x 20.

Initial trial value for θ^* is assumed as follows

$\theta_i = 300.0, \ i = 1, 2, \ldots, 10$ and

$\theta_j = 50.0, \ \ j = 11, 12, \ldots, 20$

In the first iteration initial trial value for dp was set equal to 1 which in turn gave a set of four values of dp as

0.5, 1, 2 and 10

The stopping criteria for the computer program was used as

$$\left| F_{n+1} - F_n \right| \leq 5.0$$

It took 68 iterations to get an optimal result upto an accuracy mentioned above. As noted earlier, near the optimum convergence became slow and sometimes the fluctuating behavior of the technique was also seen.

This problem consumed 352 seconds of computer time on an IBM 360/50 computer. The problem required 26312 bytes of computer memory storage. The optimum result is shown in Table 1.

Table 1. Results of Twenty Dimensional Problem (Gradient Technique).

n	P_n	W_n	I_n
1	445.23	77.42	278.23
2	432.89	74.22	264.12
3	417.56	71.20	241.68
4	398.71	68.49	324.39
5	386.67	65.95	314.06
6	372.80	63.75	311.86
7	358.06	61.71	377.92
8	349.15	60.06	269.07
9	329.57	58.65	198.64
10	303.52	57.78	152.16

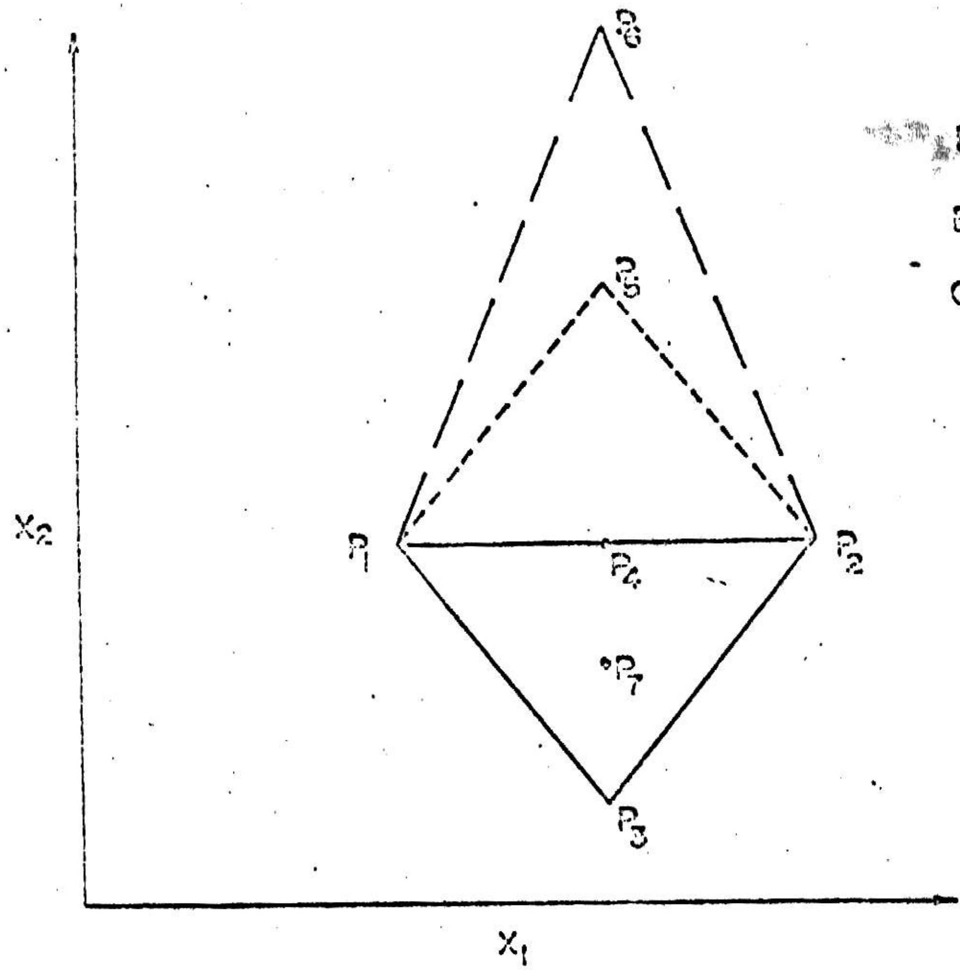
Minimum cost = $242288.70

4. SIMPLEX PATTERN SEARCH

There are number of direct search techniques which have been developed recently for finding the minimum or maximum of a function of several variables. The simplex pattern search is considered to be most efficient and simplest in the direct search procedures. There are number of pattern search techniques available for optimization purposes. The particular method proposed by Nelder and Mead [16] will be presented here.

In general to use this method for the minimization of a function of n variables, it is necessary to set up a simplex of (n+1) vertices, that it to select (n+1) trial points in the n dimensional space. The values of the objective function are then calculated at each of these points. By comparing the values of the objective function at these (n+1) points, the vertex or point with the highest value (i.e. the worst point in minimization) is replaced by a point with a lower value of the objective function. A discussion of the operations to select this point will be described in detail. As the objective function approaches the minimum, the point of the simplex with the highest value is discarded and is replaced by a point with a lower value to form a new simplex of (n+1) points. This procedure is repeated until the point corresponding to the minimum value of the objective function is achieved.

The procedure of the technique is described for a two dimensional problem in which objective function $S = f(x_1, x_2)$ is to be minimized. A simplex with $(n+1) = 3$ points is required



Reflection : $P_5 = P_4 + \alpha(P_4 - P_3)$,

Expancion : $P_6 = P_4 + \gamma(P_5 - P_4)$

Contraction : $P_7 = P_4 + \beta(P_3 - P_4)$

$\alpha = 1$, $\beta = \frac{1}{2}$, $\gamma = 2$

Fig. 3 Simplex triangle.

to set up as shown in Figure 3. Let P_1, P_2 and P_3 are the trial points which form the three points in the two dimensional space of x_1 and x_2. The following notations are used to describe the method.

y_n = the value of the objective function at the point, P_n.

P_1 = the vertex or point with the lowest value of the objective function (y_1) in the simplex or set of trial points

P_3 = the vertex or point with the highest value of the objective function (y_3) in the simplex or set of trial points; this point corresponds to P_{n+1} for n=2 variables

P_2 = the vertex or point at which the corresponding value of the objective function (y_2) lies between the values of the objective function (y_1) and (y_3) for points P_1 and P_3.

P_4 = the centroid of the vertices or points, P_1 and P_2, with the value of the objective function (y_4). In general the centroid of a set of n points in a simplex is given by

$$P_c = \sum_{i=1}^{n} P_i/n$$

The three operations through which a new point with a lower value of the objective function is found are known as reflection, expansion and contraction.

The reflection of the highest valued point, P_3 with respect to the centroid, P_4, is denoted by P_5 and its coordinates are defined according to the relation

$$P_5 = P_4 + \alpha\ (P_4 - P_3) \tag{1}$$

where α is a positive constant, the reflection coefficient. Note that P_5 is on the line joining P_3 and P_4, on the far side of P_4 from P_3 with the distance between points P_4 and P_5 denoted by $\overline{P_4P_5}$ which is equal to $\alpha\ \overline{P_3P_4}$.

The reflected point P_5 may be expanded to P_6 according to the relation

$$P_6 = P_4 + \gamma(P_5 - P_4) \tag{2}$$

where γ is the expansion coefficient, which is greater than unity, is the ratio of the distances $\overline{P_6P_4}$ to $\overline{P_5P_4}$.

The contraction of the highest valued point, P_3, with respect to the centroid, P_4, is presented by P_7 and defined by the relation

$$P_7 = P_4 + \beta\ (P_3 - P_4) \tag{3}$$

where β is a positive number between 0 and 1 and is the ratio of the distances $\overline{P_7P_4}$ to $\overline{P_3P_4}$.

The values of the coefficients, α, β and γ, considered best by Nelder and Mead [16] for faster convergance are

$$\alpha = 1, \qquad \beta = 1/2, \qquad \text{and} \qquad \gamma = 2$$

However, the best values for α, β and may be different for different problems and should be determined from experience. The details of the procedure for using the method of simplex pattern search are described as follows:

1. Vertices, P_1, P_2 and P_3 of the initial simplex are located according to the values of the objective function at each point having the relation $y_1 < y_2 < y_3$.
2. P_4, the centroid of P_1 and P_2 is determined.
3. First, P_3, is reflected to P_5 with respect to P_4, and if $y_1 < y_5 \leq y_2$, then P_3 is replaced by P_5 and we start the procedure again with a new simplex, i.e., return to step 1.
4. If $y_5 < y_1$, that is, if the reflection has produced a new minimum, we expand P_5 to P_6. If $y_6 < y_1$, we replace P_3 by P_6 and restart the process by returning to step 1. But if $y_6 > y_1$, we have failed in expansion and must replace P_3 by P_5 before starting again.
5. If after reflection, we find that $y_5 > y_1$ and $y_5 > y_2$, we define a new P_3 to be either the old P_3 or the old P_5, depending on whichever has a lower y_n value and then contract P_3 to P_7. We replace P_3 by P_7 and restart the procedure by returning to step 1, unless $y_1 > y_3$, that is, unless the contracted point has a higher value than P_3. For such a failed contraction, we replace P_2 and P_3 by $(P_2 + P_1)/2$ and $(P_3 + P_1)/2$ respectively and restart the process by returning to step 1.

The procedure used here for the two dimensional problem can easily be extended to the n-dimensional problem. The worst point of a simplex with (n+1) vertices is reflected, expanded or contracted in the same manner with respect to the centroid of the remaining n vertices until the stopping criterion is satisfied. A flow diagram of the method is given in Appendix II.

One stopping criterion is the occurence of five consecutive values of the objective function which are nearly equal in the desired level of accuaracy. Another stopping criterion would be to compare the "standard error" of the y's in the form

$$\left\{\left[\sum_{i=1}^{n+1} (y_i - \bar{y})^2\right]/n\right\}^{\frac{1}{2}}$$

with a prescribed value of desired accuracy and stop the program when it falls below this value.

The initial simplex for the n-dimensional problem is usually set up as follows.

One point which is the centroid of the initial simplex is selected and perturbation size is also specified for each component of the selected point. The (n+1) vertices of the initial simplex then can be formed by (n+1) x (n) matrix which is shown as follows. Let the selected point is

$$\begin{bmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ q_n \end{bmatrix}$$

and the perturbation size is

$$\begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_n \end{bmatrix}$$

The matrix of the vertices of the initial simplex will be

$$\begin{array}{ll} & \\ \text{Point} & 1 \\ & 2 \\ & 3 \\ & 4 \\ & \cdot \\ & \cdot \\ & \cdot \\ & \cdot \\ & n \\ & n+1 \end{array}
\begin{array}{c} \begin{array}{cccccc} \theta_1 & \theta_2 & \theta_3 & \cdots\cdots & \theta_n \end{array} \\
\begin{bmatrix}
q_1 - d_1 & q_2 - d_2 & q_3 - d_3 & \cdots\cdots & q_n - d_n \\
q_1 + d_1 & q_2 - d_2 & q_3 - d_3 & \cdots\cdots & q_n - d_n \\
q_1 & q_2 + 2d_2 & q_3 - d_3 & \cdots\cdots & q_n - d_n \\
q_1 & q_2 & q_3 + 3d_3 & \cdots\cdots & q_n - d_n \\
\cdot & \cdot & q_3 & \cdots\cdots & q_n - d_n \\
\cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & & \cdot \\
q_1 & q_2 & q_3 & \cdots\cdots & q_n - d_n \\
q_1 & q_2 & q_3 & \cdots\cdots & q_n + nd_n
\end{bmatrix} \end{array}$$

Each point of the simplex of (n+1) vertices represents n dimensional vector.

A. Application to two dimensional production scheduling problem.

Here the objective function which is to be minimized is given by

$$S = C(\theta_1 - \theta_0)^2 + D(E - I_1)^2 + C(\theta_2 - \theta_1)^2 + D(E - I_2)^2$$

where,

$$I_1 = 12 + \theta_1 - 30$$
$$I_2 = \theta_1 + \theta_2 - 28$$

The problem is to find optimal values of θ_1 and θ_2 such that S is minimized. Simplex pattern search is programmed in WATFOR for 360/50 computer. The computer program is given in Appendix II.

In this problem the initial simplex is formed by selecting one point as

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} 15.0 \\ 15.0 \end{bmatrix}$$

and the perturbation size as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 5.0 \\ 5.0 \end{bmatrix}$$

Then the initial starting simplex is given by

$$\begin{array}{ll} & \begin{array}{cc} \theta_1 & \theta_2 \end{array} \\ \begin{array}{l} \text{pt. } 1 \\ \text{pt. } 2 \\ \text{pt. } 3 \end{array} & \begin{bmatrix} 10 & 10 \\ 20 & 10 \\ 15 & 25 \end{bmatrix} \end{array}$$

The stopping criteria is to stop when

$$\left[\frac{\sum_{i=1}^{3} (S_i - \bar{S})^2}{2}\right]^{\frac{1}{2}} \leq 0.001$$

where $\bar{S}$ is the mean function value of a simplex of three points. Another stopping criteria is to stop when number of iterations exceeds over one hundred iterations.

The output result of this problem is as follows,

$$\theta_1 = 17.82$$
$$\theta_2 = 18.21$$
$$\text{minimum } S = \$2960.71$$

This problem took 30 iterations to get an optimal solution. The number of objective function evaluated is 53. It consumed 17.33 seconds on IBM 360/50 computer. The problem required 19824 bytes of computer memory storage.

B. Application to twenty dimensional HMMS paint factory model.

The function which is to be minimized is given by

$$S = \sum_{n=1}^{10} S_n$$

where

$$S_n = 340.0W_n + 64.3(W_n - W_{n-1})^2 + 0.2(P_n - 5.67W_n)^2 + 51.2P_n - 281.0W_n + 0.0825(I_n - 320.0)^2$$

Let $\theta_i = P_i, \qquad i = 1, 2, \ldots, 10$

$\theta_j = W_i, \qquad j = 11, 12, \ldots, 20$

and $I_n = I_{n-1} + \theta_n - Q_n$

with initial inventory level, $I_0 = 263.0$

Therefore the objective function now can be written as

$$S = \sum_{n=1}^{10} \Big\{ 340\theta(n+10) + 64.3 \left[\theta(n+10) - \theta(n+9)\right]^2 + 0.2 \left[\theta(n) - 5.67\ \theta(n+10)\right]^2 + 51.2\ \theta(n) - 281.0\ \theta(n+10) + 0.0825 \left[I(n) - 320.0\right]^2 \Big\}$$

The problem is to find $\theta(n)$; $n = 1, \ldots, 20$ such that the objective function S, is minimized.

This problem was solved on an IBM 360/50 computer. The computer program and the flowchart is given in Appendix II.

The point which sets up the initial starting simplex according to the matrix formulation was selected as

$$\theta_i = 400.0, \qquad i = 1, 2, \ldots, 10$$
$$\theta_j = 70.0, \qquad j = 11, 12, \ldots, 20$$

and the perturbation size for each component of the twenty dimensional vector was chosen as follows

$$d_i = 5.0, \qquad i = 1, 2, \ldots, 10$$
$$d_j = 1.0, \qquad j = 11, 12, \ldots, 20$$

The standard deviation, for the stopping criteria used in the computer program, was chosen equal to 10.0.

The optimum result was obtained after 375 iterations on an IBM 360/50 computer upto an accuracy mentioned above. It consumed 612 seconds of computer time. The problem required 20736 bytes of computer memory storage. The output result is shown in Table 2.

Table 2. Results of Twenty Dimensional Problem (Simplex pattern search).

n	P_n	W_n	I_n
1	435.06	77.51	268.06
2	468.26	74.66	289.32
3	428.67	71.24	277.99
4	377.54	68.62	339.53
5	376.02	66.32	318.55
6	377.62	64.19	321.17
7	339.79	62.02	368.96
8	355.70	60.03	266.66
9	326.11	58.28	192.77
10	277.39	56.92	119.16

Minimum cost = $242177.60

5. FLETCHER AND POWELL METHOD

An efficient search technique for finding the minimum of a function of several variables has been developed by Fletcher and Powell [4]. This search method is based on the conjugare gradient method developed by Davidon [2]. The method also utilizes the fact that near the optimum the second order terms in a Taylor series expansion dominate.

The method supposes that the function and its first partial derivatives can be calculated at all points. The application of this method is restricted to only unconstrained minimization problems and thus the method of Fletcher and Powell is useful for finding an unrestricted local optimum.

The conjugate gradient method assumes that in a neighborhood of the minimum the function can be closely approximated by a positive definite quadratic form. From this assumption Fletcher and Powell proved in their paper [4] that their method has quadratic convergence.

The direction for the search are chosen in such a way that conjugate directions are generated; and each direction is a direction of steepest descent. Then the method uses one dimensional searches in these directions. The method is described for a general minimization problem of n variables.

Consider a function to be minimized is

$$S = f(x_1, x_2, \ldots, x_n) \tag{1}$$

The gradient vector for this function is

$$g = [g_1, g_2, \ldots, g_n]$$

where $g_i = \frac{\partial S}{\partial x_i}$, $i = 1, 2, \ldots, n$.

Assuming a general quadratic function in the vector matrix form

$$S = f_0 + \underline{a}^T \underline{x} + \frac{1}{2} \underline{x}^T \underline{G} \underline{x} \tag{2}$$

where $\underline{a}^T$ and $\underline{x}^T$ are row vectors, and G is a matrix of 2^{nd} order partial derivatives. The function S is quadratic if

$$G_{ij} = G_{ji}$$

Further G is a positive definite matrix.

From equation (2), gradient vector which consist of first partial derivatives can be calculated. In vector-matrix form

$$\underline{g} = \underline{a} + \underline{G} \underline{x} \tag{3}$$

Because of the fact that at minimum point, the gradient vanishes; we have

$$\underline{a} + \underline{G} \underline{\bar{x}} = 0 \tag{4}$$

where $\underline{\bar{x}}$ denotes the column vector at the minimum point.

Subtracting equation (4) from equation (3) we obtain

$$\underline{g} = \underline{G}(\underline{x} - \underline{\bar{x}}) \tag{5}$$

The first partial derivatives are known at any point $\underline{x}$. Therefore $\underline{g}$ is known at any point. From equation (5) we obtain

$$\underline{\bar{x}} = \underline{x} - \underline{G}^{-1}\underline{g} \tag{6}$$

where G^{-1} is the inverse of the matrix G.

To solve this equation, the method of Fletcher and Powell utilizes a matrix $\underline{H_0}$ which is an approximation to the matrix G^{-1}. As the optimum is approched the matrix $\underline{H_0}$ converges to G^{-1}. Where G^{-1} is the inverse of G in which the elements of the matrix are second partial derivatives of objective function evaluated at the optimum.

In the first step of iteration it is customary to set $\underline{H_0} = \underline{I}$ where $\underline{I}$ is an identity matrix. Using $\underline{H_0}$ for G^{-1} in equation (6), we obtain a direction vector $\underline{\xi}$

$$\underline{\xi} = - \underline{H_0}\, \underline{g} \tag{7}$$

As said above in the first iteration $\underline{H_0}$ is an identity matrix and $\underline{g}$ is the vector consist of partial derivatives of the objective function at the initial assumed point $\underline{x_0}$.

Then new point in the direction ξ is found by

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \lambda \underline{\xi} \tag{8}$$

The one dimensional search in the direction $\underline{\xi}$ is conducted and the value of scalar λ which minimizes the objective function is determined. This value of λ will be denoted by $\bar{\lambda}$.

Now define a new vector $\underline{\sigma}$ as

$$\underline{\sigma} = \bar{\lambda} \underline{\xi} \tag{9}$$

also define a new vector $\underline{y}$ as

$$\underline{y} = \underline{g}_{i+1} - \underline{g}_i \tag{10}$$

The improved matrix $\underline{H}$ is obtained by

$$\underline{H} = \underline{H}_0 + \underline{A} + \underline{B} \qquad (11)$$

where

$$\underline{A} = \frac{\underline{\sigma}\ \underline{\sigma}^T}{\underline{\sigma}^T \underline{y}}$$

and

$$\underline{B} = \frac{-\ \underline{H}_0\ \underline{y}\ \underline{y}^T\ \underline{H}_0}{\underline{y}^T\ \underline{H}_0\ \underline{y}}$$

This new improved matrix $\underline{H}$ is used as the matrix $\underline{H}_0$ in the next iteration to compute a new direction $\underline{\xi}$ and a new gradient vector at the point x^{i+1} which is obtained from equation (8).

The one dimensional cubic interpolation search procedure is usually used in the method of Fletcher and Powell to find the minimum of equation (1) along the line given by equation (8)

The procedure is terminated when each of the correction G_i is less than a prescribed accuracy and when each of the component of direction vector $\underline{\xi}$ is less than a prescribed accuracy $\underline{\epsilon}$; that is we wish to have

$$\underline{\xi} \leq \underline{\epsilon}$$

It is obviously practicable to apply this method to find a local minimum of a general function of a large number of variables whose first derivatives can be evaluated quickly, even if only poor initial approximations to a solution are known.

A. Application to two dimensional production scheduling problem.

The objective function to be minimized is

$$S = C(\theta_1 - \theta_0)^2 + D(E - I_1)^2 + C(\theta_2 - \theta_1)^2 + D(E - I_2)^2$$

Let $x_1 = \theta_1$
$x_2 = \theta_2$

Then the inventories at the first period and that at the second period are

$$I_1 = 12 + x_1 - 30$$
$$I_2 = I_1 + x_2 - 10 = x_1 + x_2 - 28$$

Substituting the values of constants and values for I_1 and I_2 in the objective function, we get

$$S = 100(x_1 - 15)^2 + 20(28 - x_1)^2 + 100(x_2 - x_1)^2 + 20(38 - x_1 - x_2)^2$$

This problem was solved on IBM 360/50 computer using IBM scientific subroutine FMFP [12]. The components of gradient vector provided in the function sub-program are as follows.

$$g_1 = \frac{\partial S}{\partial x_1} = 200(x_1 - 15) - 40(28 - x_1) - 200(x_2 - x_1) - 40(38 - x_1 - x_2)$$

$$g_2 = \frac{\partial S}{\partial x_2} = 200(x_2 - x_1) - 40(38 - x_1 - x_2)$$

The stopping criteria is to stop when

$$|\xi_{i+1} - \xi_1| \leq \epsilon$$

In this problem ϵ is specified as 0.001. The various data which are necessary to provide with the use of subroutine FMFP are provided as follows.

Limit = 10

Estimate = 3000.0

Epsilon = 0.001

where Limit is the upper limit of number of iterations, Estimate is an estimated optimal objective functional value, and Epsilon, ϵ, is the constant used in the stopping criterion. The initial trial value is set at

$x_1 = 10.0$

$x_2 = 10.0$

The output result is found as follows

$x_1 = 19.82$

$x_2 = 18.21$

minimum S = \$2960.71

It took only 3 iterations to obtain the above optimal solution. It consumed 10.31 seconds of computer time on an IBM 360/50 computer. The problem required 8412 bytes of computer memory storage.

B. Application to twenty dimensional HMMS paint factory model.

As seen earlier the function which is to be minimized is given by

$$S = \sum_{n=1}^{10} S_n$$

where

$$S_n = 340.0W_n + 64.3(W_n - W_{n-1})^2 + 0.2(P_n - 5.67W_n)^2$$
$$+ 51.2P_n - 281.0W_n + 0.0825(I_n - 320.0)^2$$

To convert the problem into the standard form of Fletcher and Powell method, let

$$x_i = P_i, \qquad i = 1, 2, \ldots, 10$$
$$x_j = W_i, \qquad j = 11, 12, \ldots, 20 \text{ and } i = 1, 2, \ldots, 10$$

and

$$I_n = I_{n-1} + x_n - Q_n$$

with initial inventory level $I_0 = 263.0$. Therefore, the objective function can be rewritten as

$$S = \sum_{n=1}^{10} \Big[340.0x(n+10) + 64.3\left\{x(n+10) - x(n+9)\right\}^2$$
$$+ 0.2\left\{x(n) - 5.67x(n+10)\right\}^2 + 51.2x(n)$$
$$- 281.0x(n+10) + 0.0825\left\{I(n) - 320.0\right\}^2\Big]$$

where I_n has recurrence relationship shown above.

This problem was also solved by an IBM 360/50 computer using scientific subroutine FMFP [12] together with the function subprogram in the main routine of the computer program.

The components of the twenty dimensional gradient vector were also supplied in the function subprogram. They are as follows

For $n = 1, 2, \ldots, 10$,

$$g(n) = \frac{\partial S}{\partial x(n)} = 0.4\,[x(n) - 5.67x(n+10)] + 51.2$$
$$+ 0.165\,[I(n-1) + x(n) - Q(n) - 320.0]$$

For $n = 11, 12, \ldots, 19$

$$g(n) = \frac{\partial S}{\partial x(n)} = 340.0 + 128.6\,[x(n) - x(n-1)]$$
$$- 2.268\,[x(n-10) - 5.67x(n)] - 281.0$$
$$- 128.6\,[x(n+1) - x(n)]$$

and

$$g(20) = \frac{\partial S}{\partial x(20)} = 340.0 + 128.6\,[x(20) - x(19)]$$
$$- 2.268\,[x(10) - 5.67x(20)] - 281.0$$

The initial starting vector of decision variables was set at

$x_i = 300.0$, $\quad i = 1, 2, \ldots, 10$ and

$x_j = 50.0$, $\quad j = 11, 12, \ldots, 20$

The data for the stopping criteria, limit by number of iterations and estimate of the minimum function value; which are necessary to provide with the use of FMFP subroutine, are as follows.

Epsilon = 0.1

Limit = 100

Estimate = 300000.0

The optimum result was obtained after 19 iterations on an IBM 360/50 computer. This problem consumed 59.90 seconds of computer time to get an optimal answer upto the accuracy mentioned above. The problem required 12572 bytes of computer memory storage. The optimum result is shown in Table 3.

Table 3. Results of Twenty Dimensional Problem. (Fletcher and Powell method)

n	P_n	W_n	I_n
1	470.33	77.66	303.33
2	444.14	74.24	300.47
3	417.09	70.88	277.56
4	381.70	67.71	343.26
5	376.24	65.03	322.50
6	363.99	62.68	311.50
7	348.89	60.64	368.39
8	359.33	58.97	269.73
9	329.08	57.32	198.81
10	272.04	56.05	120.86
		Minimum cost = $241512.10	

6. FLETCHER AND REEVES METHOD.

The method of Fletcher and Reeves [5] is also a quadratically convergent conjugate gradient method for locating an unconstrained local minimum of a function of several variables. It is similar to the method of Fletcher and Powell [4].

The difference in both the methods is only in finding the new direction of search. The method of Fletcher and Powell uses the matrix H for successive improvement in matrix G^{-1}. Hence this method requires larger storage space. Particular advantage of the method of Fletcher and Reeves is its modest demand on storage space as only three vectors being required for storage.

This method also has quadratic convergence, meaning that for quadratic functions it is guranteed that the minimum will be located exactly, apart from rounding errors, within some finite numbers of iterations usually n which is the number of variables. The method also supposes that function and its partial derivatives can be calculated at all points.

The method can be described for a general minimization problem of n variables. Consider a function to be minimized

$$S = f(x_1, x_2, \ldots, x_n)$$

The gradient vector at each point is

$$\underline{g} = [g_1, g_2, \ldots, g_n]$$

where

$$g_i = \frac{\partial S}{\partial x_i}$$

It is seen in the method of Fletcher and Powell that the new direction of search is found by

$$\underline{\xi} = -H\ \underline{g}$$

where H is a matrix. Instead of finding new direction by this way, the method of Fletcher and Reeves finds new direction of search as follows

$$\underline{\xi}_{i+1} = - \underline{g}_{i+1} + \beta_i \underline{\xi}_i \tag{1}$$

where β_i is scalar given by

$$\beta_i = \frac{\underline{g}_{i+1}^T\ \underline{g}_{i+1}}{\underline{g}_i^T\ \underline{g}_i} \tag{2}$$

For the first iteration β_{i-1} will be zero and hence starting direction will be negative of gradient direction that is

$$\underline{\xi}_1 = - \underline{g}_1$$

Then the new point in this direction is found by

$$\underline{x}^{(i+1)} = \underline{x}^{(i)} + \lambda \underline{\xi}_i \tag{3}$$

Then one dimensional linear search in the direction $\underline{\xi}$ is conducted and the value of scalar λ which minimizes the function is determined.

This procedure leads to the following general minimization algorithm.

Initially select an arbitrary point $\underline{x}^{(1)}$ then gradient vector at this point is calculated which is denoted by $\underline{g}_1$. The direction of search at this point will be $\underline{\xi_1} = -\underline{g}_1$. Then new point $x^{(i+1)}$ in this direction is located by equation (3). Then gradient vector at new point is calculated and new direction of search is obtained by equation (1).

As said above this process is guranteed, apart from rounding errors, to locate the minimum of any quadratic function of n variables in the at most n iterations.

The one dimensional cubic interpolation search is usually incorporated in this method to locat the minimum along the direction $\underline{\xi}$ which determines the value of λ.

The procedure is terminated when each of the correction G_i is less than a prescribed accuracy and when each of the component of $\underline{\xi}$ is less than a prescribed value epsilon $\underline{\epsilon}$. Sometimes it might be sufficient to continue the iterations until a complete cycles of (n+1) iterations.

A. Application to two dimensional production scheduling problem.

Here the objective function which is to be minimized is given by

$$S = C(\theta_1 - \theta_0)^2 + D(E - I_1)^2 + C(\theta_2 - \theta_1)^2 + D(E - I_2)^2$$

with notations and values for the constants as described in section 2. The approach of the problem is same as in the method of the Fletcher and Powell.

Let $x_1 = \theta_1$

$x_2 = \theta_2$

Therefore

$$I_1 = 12.0 + x_1 - 30.0$$

and

$$I_2 = x_1 + x_2 - 28.0$$

The objective function becomes,

$$S = 100(x_1 - 15)^2 + 20(28 - x_1)^2 + 100(x_2 - x_1)^2 + 20(38 - x_1 - x_2)^2$$

The problem is to find the optimal values of x_1 and x_2 such that the objective function S is minimized. The components of two dimensional gradient vector can be written as

$$g_1 = \frac{\partial S}{\partial x_1} = 200(x_1 - 15) - 40(28 - x_1) - 200(x_2 - x_1) - 40(38 - x_1 - x_2)$$

$$g_2 = \frac{\partial S}{\partial x_2} = 200(x_2 - x_1) - 40(38 - x_1 - x_2)$$

An IBM scientific subroutine FMCG [12] was incorporated into the main program together with the function subprogram which provides the objective function and components fo the gradient vector as shown above.

The stopping criteria is to stop when $\left|\xi_{n+1} - \xi_n\right| \leq \epsilon$. The data for the limit of iterations, estimate of the minimum function value and epsilon for the above stopping criteria which are necessary to provide with the use of subroutine FMCG are as follows, Epsilon $= 0.001$

Estimate $= 3000.0$

Limit $= 10$

The initial trial value was used as

$$x_1 = 10.0$$
$$x_2 = 10.0$$

The optimum result for the problem is given below.

$$x_1 = 17.82$$
$$x_2 = 18.21$$
$$\text{minimum } S = \$2960.71$$

The method of Fletcher and Reeves took only 3 iterations to get an optimal result. This problem consumed 8.80 seconds of computer time on an IBM 360/50 computer. The problem required 7132 bytes of computer memory storage.

B. Application to twenty dimensional HMMS paint factory model.

The method of Fletcher and Reeves was also applied to 20 dimensional HMMS paint factory model.

As described earlier in the section 2, the objective function of the model is given by

$$S = \sum_{n=1}^{10} S_n$$

where

$$S_n = 340.0W_n + 64.3(W_n - W_{n-1})^2 + 0.2(P_n - 5.67W_n)^2 + 51.2P_n - 281.0W_n + 0.0825(I_n - 320.0)^2$$

with usual notations already described in section 2. To convert the problem into the standard form of the Fletcher and Reeves method, we define

$$x_i = P_i, \qquad i = 1, 2, \ldots, 10$$

and

$$x_j = W_i, \qquad i = 1, 2, \ldots, 10$$
$$j = 11, 12, \ldots, 20$$

also $I_n = I_{n-1} + x_n - Q_n$ with initial given inventory level $I_0 = 263.0$. Now the objective function can be written in the following form.

$$S = \sum_{n=1}^{10} [340.0x(n+10) + 64.3 \{x(n+10) - x(n+9)\}^2 + 0.2 \{x(n) - 5.67x(n+10)\}^2 + 51.2x(n) - 281.0x(n+10) + 0.0825 \{I_n - 320.0\}^2]$$

This problem was solved on an IBM 360 computer using IBM scientific subroutine FMCG [12] together with the function subprogram in the main routine of the computer program.

The components of the twenty dimensional gradient vector supplied in the function subprogram are as follows,

For $n = 1, 2, \ldots, 10$;

$$g(n) = \frac{\partial S}{\partial x(n)} = 0.4 [x(n) - 5.67x(n+10)] + 51.2 + 0.165 [I(n-1) + x(n) - Q(n) - 320.0]$$

For $n = 11, 12, \ldots, 19$;

$$g(n) = \frac{\partial S}{\partial x(n)} = 340.0 + 128.6\,[x(n) - x(n-1)] - 2.268\,[x(n-10) - 5.67x(n)] - 281.0 - 128.6\,[x(n+1) - x(n)]$$

and

$$g(20) = \frac{\partial S}{\partial x(20)} = 340.0 + 128.6\,[x(20) - x(19)] - 2.268\,[x(10) - 5.67x(20)] - 281.0$$

The initial starting vector of decision variables is set at

$$x_i = 300.0\,, \qquad i = 1, 2, \ldots, 10 \qquad \text{and}$$
$$x_j = 50.0\,, \qquad j = 11, 12, \ldots, 20$$

The data for the stopping criteria, limit of maximum number of iterations and estimate of the minimum function value which are necessary to provide in the subroutine FMCG, are as follows.

Epsilon = 0.1

Estimate = 300000.0

Limit = 100

The method of Fletcher and Reeves took 31 iterations to get an optimal result. This problem consumed 49.98 seconds of computer time. This problem required 11292 bytes of computer memory storage. The optimum output result is shown in Table 4.

Table 4. Results of Twenty Dimensional Problem (Fletcher and Reeves Method)

n	P_n	W_n	I_n
1	471.37	77.68	304.37
2	444.64	74.27	302.02
3	416.31	70.90	278.34
4	380.90	67.75	343.24
5	374.88	65.07	321.13
6	363.57	62.72	309.70
7	349.92	60.70	367.62
8	359.52	59.03	269.15
9	329.82	57.40	198.98
10	275.43	56.16	124.41
		Minimum cost = $241517.00	

7. A COMPARISON AND THE DISCUSSION OF RESULTS

The results obtained by these four techniques namely, gradient technique, simplex pattern search, Fletcher and Powell method and Fletcher and Reeves method are compared with respect to following four criteria.

1. The optimum function value obtained upto an accuaracy prescribed.
2. Total computation time in seconds which is considered as execution time plus the compilation time.
3. Number of iteration required to arrive at an optimum solution. An iteration is defined as each successive move from previous point except in simplex method where an iteration means formation of each successful simplex.
4. Computer memory storage required in bytes.

These four criteria give the idea about convergence and effectiveness of each technique under identical conditions, that is, under the same computing system and with the same set of problems. The initial starting point in each problem is kept the same for all techniques except in the simplex pattern search, because simplex pattern search starts its search from initial simplex which consist of (n+1) different points as described in the method.

Table 5 shows a comparison of results of first test problem which is two dimensional production planning problem. It can be seen that each technique produced the same optimum function value upto an accuracy of two decimal points. The

Table 5. A Comparison of Results of Two Dimensional Problem.

Technique	Optimum function value	Computation time in seconds	Number of iterations	Computer memory storage in bytes
Gradient	2960.71	16.10	11	9816
Simplex	2960.71	17.33	30	19824
Fletcher & Powell	2960.71	10.31	3	8412
Fletcher & Reeves	2960.71	8.80	3	7132

computation time and storage requirement for simplex pattern search are highest among all the four techniques. Fletcher and Powell method and Fletcher and Reeves method produced nearly the same results although Fletcher and Reeves method proves the best in this test problem. Gradient technique puts itself in the third place with normal results.

Table 6 shows a comparison of results of second test problem which is twenty dimensional HMMS paint factory model. The optimum function values obtained by all the four techniques are differ from each other by less than 1%. Simplex pattern search took the longest time to arrive at an optimal solution and Fletcher and Reeves method took the minimum time. Gradient technique gave the nominal result with respect to all the four criteria. The computer memory storage required for this problem is largest for gradient technique. In this problem also the method of Fletcher and Powell and method of Fletcher and Reeves produce nearly the same results; though Fletcher and Powell method arrived at an optimum solution in only 19 iterations whereas for the same problem method of Fletcher and Reeves took 31 iterations.

It is seen from the results shown in Table 5 and Table 6 that Fletcher and Powell method and Fletcher and Reeves method gave the highest convergence rate in both the problems. This is expected because they have characteristic of quadratic convergence and the objective functions are in quadratic forms. Fletcher and Powell method requires the storage of matrix H as described in the method while method of Fletcher and Reeves

Table 6. A Comparison of Results of Twenty Dimensional Problem.

Technique	Optimum Function value	Computation time in seconds	Number of iterations	Computer memory storage in bytes
Gradient	242238.70	352	68	26312
Simplex	242177.60	612	375	20736
Fletcher & Powell	241512.10	59.90	19	12572
Fletcher & Reeves	241517.00	49.98	31	11292

requires storage for only three vectors hence the latter took less computation time and less number of memory storage locations in the computer.

Simplex pattern search took the longest time to get an optimal solution in both the test problems. The reason is obvious because simplex pattern search basically searches for all the possible points on the response surface of the objective function and hence it took more number of iterations and computation time compared to other methods.

Gradient technique shows its normal behavior in both the test problems. It is seen that in the second problem it needed the largest number of memory storage as it requires to store three large dimensional metrices as described in the method.

The effect of each method on the dimensionality of the problem can also be compared from these two tables. Fletcher and Powell method and Fletcher and Reeves method have less effect on increasing the dimensionality of the problem with regard to all the four criteria. Both the methods produced optimum results for twenty dimensional problem in less than a minute of computation time. This shows quite encouraging and promising behavior of the optimization techniques based on conjugate gradient method.

In this case also gradient technique has very normal effect on increasing the dimensionality of the problem except that it requires very large number of computer storage locations as the dimension of the optimization problem increases. It produced optimal result for two dimensional problem in only 16 seconds of computer time while for twenty dimensional problem it consumed about 6 minutes of computer time which is considered to be normal effect on the dimensionality of the problem.

It can be stated from the above results that simplex pattern search gets worst as the dimension of the optimization problem increases. The reason for this is that near to optimum; a simplex becomes small and hence it takes more time compared to other optimization techniques. Also it required quite a high number of iterations to arrive at an optimum solution in the second test problem.

The results show that the conjugate gradient method of Fletcher and Powell and method of Fletcher and Reeves present the most consistent behavior among the group of techniques considered here. They can be proved highly efficient for many kinds of unconstrained optimization problems arise in the industrial management systems. The gradient technique is also a fast converging techanique and it is easy to apply and program for the various kinds of optimization problems. Simplex pattern search is also an efficient direct search optimization technique for low dimensional problems as it does not require to calculate the derivatives of the objective function. Therefore this technique is adequate to treat difficult optimization problems where derivatives are difficult to calculate.

REFERENCES

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

2. Davidon, W. C., "Variable Metric Method for Minimization", AEC Research and Development Report, ANL-5990, pp. 21, 1959.

3. Fan, L. T. and Wang, C. S., The Discrete Maximum Principle- A Study of Multistage Systems Optimization, John Wiley and Sons, New York, 1964.

3a. Fan, L. T., Hwang, C. L, and Tillman, F. A., "A Sequential Pattern Search, Solution To Production Planning Problems", AIIE Transaction, In Press, 1969.

4. Fletcher, R., and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization", The Computer Journal, Vol. 6, pp. 163, 1963.

5. Fletcher, R. and Reeves, C. M., "Function Minimization by Conjugate Gradients", The Computer Journal, Vol. 7, pp. 149, 1964.

6. Hadley, G., Linear Programming, Addison-Wesley Publishing Company, Inc., Readin, Mass., 1968.

7. Holt, C. C., Modigliani, F., Muth, J. F. and Simon, H. A., Planning, Production, Inventories and Work Force, Prentice Hall, Englewood Cliffs, N. J., 1960.

8. Holt, C. C., Modigliani, F., Muth, J. F. and Simon, H. A., "A Linear Decision Rule for Production and Employment Scheduling", Management Science, Vol. 2, No. 1, pp. 1, 1955.

9. Hooke, R. and Jeeves, T. A., "Direct Search Solution of Numerical and Statistical Problems", J. Assoc. Comp. Mach., Vol. 8, pp. 212, 1961.

10. Hwang, C. L., Tillman, F. A. and Fan, L. T., "Optimization of Production Scheduling", International Journal of Production Research, Vol. 5, pp. 341, 1967.

11. Hwang, C. L., Fan, L. T., Tillman, F. A. and Sharma R., "Optimal Production Planning and Inventory Control", Submitted to Int. J. of Production Research, Sept. 1968.

12. IBM, "System/360 Scientific Subroutine Package Version III Programmer's Manual", IBM, 1968.

13. Kowallic, J. and Osborne, M., Methods for Unconstrained Minimization Problems, American Elsevier Publishing Co., New York, 1968.

14. Leitmann, G., Optimization Techniques with Applications to Aerospace Systems, Academic Press, New York, 1962.

15. Leon, A., "A Comparison of Eight Known Optimizing Procedures", Recent Advance in Optimization Techniques, Lavi and Voge Editors, John Wiley and Sons, New York, 1966.

16. Nelder, J. A. and Mead, R., "A Simplex Method for Function Minimization", The Computer Journal, Vol. 7, pp. 308, 1965.

17. Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several variables without Calculating Derivatives", The Computer Journal, Vol. 7, pp. 155, 1964.

18. Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or the Least Value of A Function", The Computer Journal, Vol. 3, pp. 175, 1960.

19. Rosenbrock, H. H. and Storey C., Computational Techniques for Chemical Engineers, Pergamon Press, New York, 1966.

20. Sage, A. P., Optimum Systems Control, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1968.

21. Taubert, W. H., "A Search Decision Rule for The Aggregate Scheduling Problem", Management Science Vol. 14, No. 6, pp. B-343, 1968.

22. Wilde, D. J., Optimum Seeking Methods, Prentice Hall, Englewood Cliffs, N. J., 1964.

23. Wilde, D. J. and Beightler, C. S., Foundations of Optimization, Prentice-Hall, Englewood Cliffs, N. J., 1967.

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to his major professor Dr. C. L. Hwang, for his valuable guidance, encouragement and cooperation in the preparation of this report. He sincerely acknowledges the suggestions and encouragement provided by Dr. F. A. Tillman and Dr. L. T. Fan.

APPENDIX I. Computer Program for Gradient Technique.

The computer flow chart which illustrates the computational procedure is presented in Fig.A-1;the program symbols, their explanations and corresponding mathematical notations are summarized in TableA-1. The computer program for twenty dimensional problem follows the symbol table.

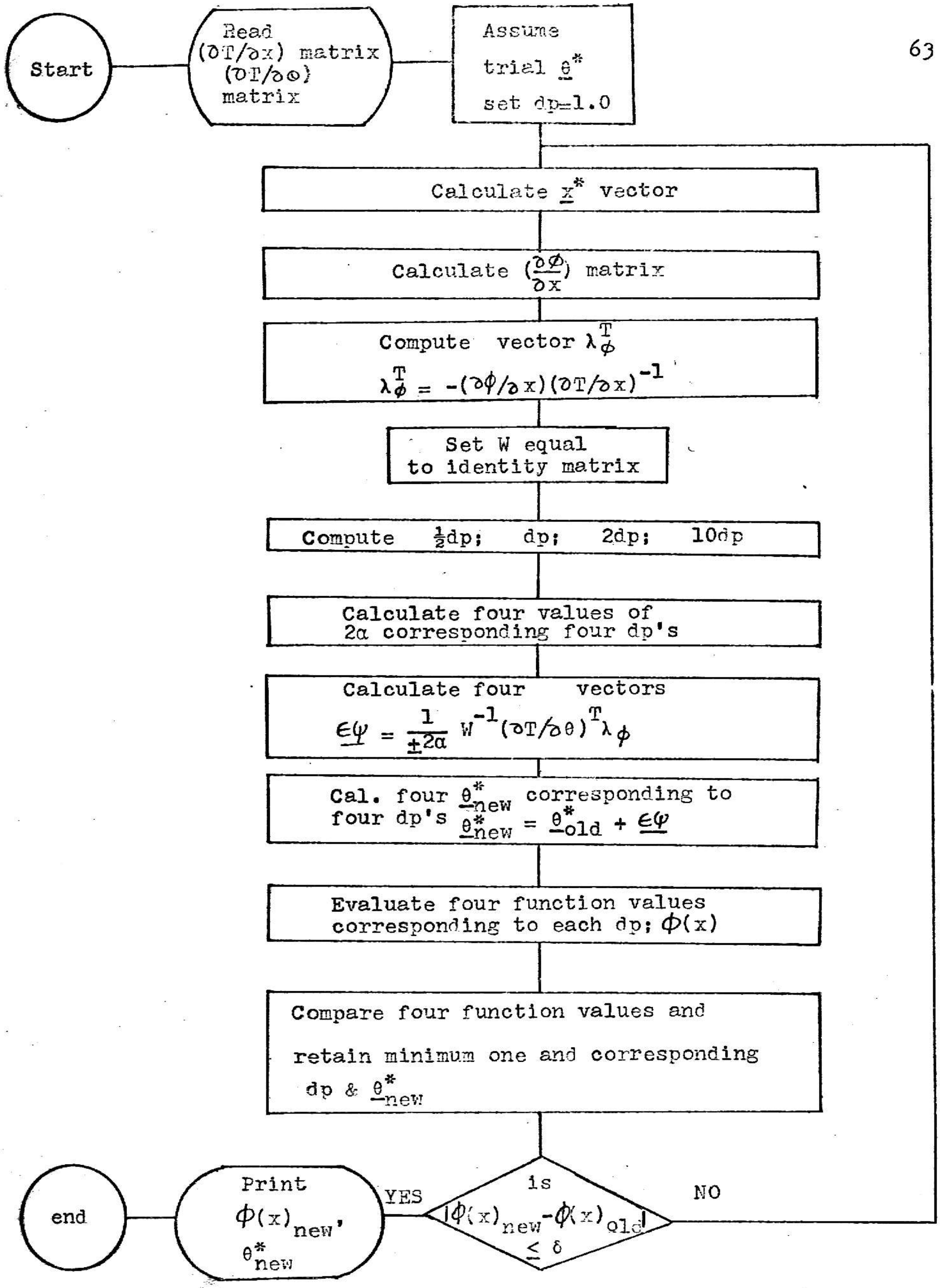


Fig.A-1. Flow diagram for Gradient technique.

Table A-1. Symbol Table

Program Symbol	Explanation	Mathematical Symbol
A	s x s matrix where s = No. of state variables	$(\partial T/\partial x)$
B	s x r matrix where r = No. of decision variables	$(\partial T/\partial \theta)$
P	A matrix of partial derivatives of a function 1 x s	$(\partial \phi/\partial x)$
H	1 x s row matrix	λ_{ϕ}^{T}
X(I)	A vector of state variables	$\underline{x}^*$
TH(I)	A vector of decision variables at old point	θ^*_{old}
S	Old objective function value	$\phi(x)_{old}$
F1	New objective function value	$\phi(x)_{new}$
D	Numerator in a formula for 2α	$\lambda_{\phi}^{T}(\partial T/\partial \theta)W^{-1}(\partial T/\partial \theta)^{T}\lambda_{\phi}$
DP	A constant	dp
R(I)	A vector to calculate $\underline{\epsilon\varphi}$	$W^{-1}(\partial T/\partial \theta)^{T}\lambda_{\phi}$
ALPHA	A constant to calculate E(I)	2α
E(I)	A vector of change in $\underline{\theta}^*$	$\underline{\epsilon\varphi}$
TH1(I)	A vector of decision variables at new point	$\underline{\theta}^*_{new}$

```
C     COMPUTER PROGRAM FOR GRADIENT TECHNIQUE
C     APPLICATION TO TWENTY DIMENSIONAL PROBLEM
C
      DIMENSION A1(900),B1(600),P(30),L1(30),M1(30),T(30)
      DIMENSION C(20),D(10),E(20),R(20),TH(20),TH1(20,4),DP(4),F(4)
      DIMENSION A(30,30),B(30,20),X(30),H(30),Q(30),F11(4)
      DO 99 I=1,10
   99 TH(I)=300.
      DO 100 J=11,20
  100 TH(J)=50.
      READ 200,(Q(I),I=1,10)
  200 FORMAT(10F5.1)
  203 FORMAT(/30X,6HMIN.F=F10.2)
  302 FORMAT(5X,3HDP=F8.3,10X,2HF=F14.2)
 1000 FORMAT(1H ,'**************************************************')
  151 FORMAT(10X,F10.2)
      DO 16 I=1,30
      DO 16 J=1,30
      IF(I-J)10,11,12
   11 A(I,J)=1.
      GO TO 16
   12 IF(I-J-1)10,13,10
   13 IF(J-9)14,14,10
   14 A(I,J)=-1.
      GO TO 16
   10 A(I,J)=0.
   16 CONTINUE
      DO 102 I=1,30
      DO 102 J=1,30
      K=I+30*(J-1)
  102 A1(K)=A(I,J)
      DO 26 I=1,30
      DO 26 J=1,20
      IF(I-J)20,21,23
   21 IF(I-10)22,22,23
   22 B(I,J)=-1.
      GO TO 26
   23 IF(I-J-10)20,25,20
   25 B(I,J)=-1.
      GO TO 26
   20 B(I,J)=0.
   26 CONTINUE
      DO 103 I=1,30
      DO 103 J=1,20
      K1=I+30*(J-1)
  103 B1(K1)=B(I,J)
      K=0
      DP(2)=1.
      N1=30
      CALL MINV(A1,N1,DET,L1,M1)
  101 X(1)=263.+TH(1)-Q(1)
      DO 1 I=2,10
    1 X(I)=X(I-1)+TH(I)-Q(I)
      DO 2 J=11,30
    2 X(J)=TH(J-10)
      DO 30 I=1,10
   30 P(I)=0.165*(X(I)-320.)
      DO 31 J=11,20
```

```
   31 P(J)=.4*(X(J)-5.67*X(J+10))+51.2
      P(21)=340.+128.6*(X(21)-81.)-2.268*(X(11)-5.67*X(21))-281.-128.6*(
     1X(22)-X(21))
      DO 32 I=22,29
   32 P(I)=340.+128.6*(X(I)-X(I-1))-2.268*(X(I-10)-5.67*X(I))-281.-128.6
     1*(X(I+1)-X(I))
      P(30)=340.+128.6*(X(30)-X(29))-2.268*(X(20)-5.67*X(30))-281.
      IF(K.NE.0)GO TO 601
      S1=340.*X(21)+64.3*(X(21)-81.)**2+.2*(X(11)-5.67*X(21))**2+51.2*X(
     111)-281.*X(21)+.0825*(X(1)-320.)**2
      S=S1
      DO 40 I=2,10
   40 S=S+340.*X(I+20)+64.3*(X(I+20)-X(I+19))**2+.2*(X(I+10)-5.67*X(I+20
     1))**2+51.2*X(I+10)-281.*X(I+20)+.0825*(X(I)-320.)**2
      F1=S
      PRINT 203,F1
      PRINT 1000
  601 DP(1)=DP(2)/2
      DP(3)=2*DP(2)
      DP(4)=10*DP(2)
      S=F1
      CALL GMPRD(P,A1,T,1,30,30)
      DO 104 I=1,30
  104 H(I)=-1*T(I)
      CALL GMPRD(H,B1,C,1,30,20)
      CALL GMTRA(C,R,1,20)
      CALL GMPRD(C,R,D,1,20,1)
      DO 500 J3=1,4
      ALPHA=SQRT(D(1)/(DP(J3))**2)
      DO 105 I=1,20
  105 E(I)=R(I)/(-ALPHA)
      DO 106 I=1,20
  106 TH1(I,J3)=TH(I)+E(I)
      X(1)=263.+TH1(1,J3)-Q(1)
      DO 51 I=2,10
   51 X(I)=X(I-1)+TH1(I,J3)-Q(I)
      DO 52 J=11,30
   52 X(J)=TH1(J-10,J3)
      DO 61 I=1,10
   61 P(I)=0.165*(X(I)-320.)
      DO 62 J=11,20
   62 P(J)=.4*(X(J)-5.67*X(J+10))+51.2
      P(21)=340.+128.6*(X(21)-81.)-2.268*(X(11)-5.67*X(21))-281.-128.6*(
     1X(22)-X(21))
      DO 63 I=22,29
   63 P(I)=340.+128.6*(X(I)-X(I-1))-2.268*(X(I-10)-5.67*X(I))-281.-128.6
     1*(X(I+1)-X(I))
      P(30)=340.+128.6*(X(30)-X(29))-2.268*(X(20)-5.67*X(30))-281.
      F11(J3)=340.*X(21)+64.3*(X(21)-81.)**2+.2*(X(11)-5.67*X(21))**2+51
     1.2*X(11)-281.*X(21)+.0825*(X(1)-320.)**2
      F(J3)=F11(J3)
      DO 111 I=2,10
  111 F(J3)=F(J3)+340.*X(I+20)+64.3*(X(I+20)-X(I+19))**2+.2*(X(I+10)-5.6
     17*X(I+20))**2+51.2*X(I+10)-281.*X(I+20)+.0825*(X(I)-320.)**2
      PRINT 302,DP(J3),F(J3)
  500 CONTINUE
      IF(F(1)-F(2))501,501,502
  501 SMALL=F(1)
      J=1
      GO TO 504
```

```
  502 SMALL=F(2)
      J=2
  504 IF(SMALL-F(3))505,505,506
  506 SMALL=F(3)
      J=3
  505 IF(SMALL-F(4))507,507,508
  508 SMALL=F(4)
      J=4
  507 F1=SMALL
      PRINT 203,F1
      PRINT 1000
      DP(2)=DP(J)
      DO 509 I=1,20
  509 TH(I)=TH1(I,J)
      K=1
      IF(ABS(S-F1)-5.)202,202,101
  202 PRINT 151,(TH(I),I=1,20)
      STOP
      END
```

```
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
```

```
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
      D=D*BIGA
      A(KK)=1.0/BIGA
   80 CONTINUE
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
      SUBROUTINE GMPRD(A,B,R,N,M,L)
      DIMENSION A(1),B(1),R(1)
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

```
      SUBROUTINE GMTRA(A,R,N,M)
      DIMENSION A(1),R(1)
      IR=0
      DO 10 I=1,N
      IJ=I-N
      DO 10 J=1,M
      IJ=IJ+N
      IR=IR+1
   10 R(IR)=A(IJ)
      RETURN
      END
```

APPENDIX II. Computer Program for Simplex Pattern Search

The computer flow chart which illustrates the computational procedure is illustrated in Fig.A-2;the program symbols and their explanation are summarized in Table A-2.The computer program for the solution of twenty dimensional problem follows the symbol table.

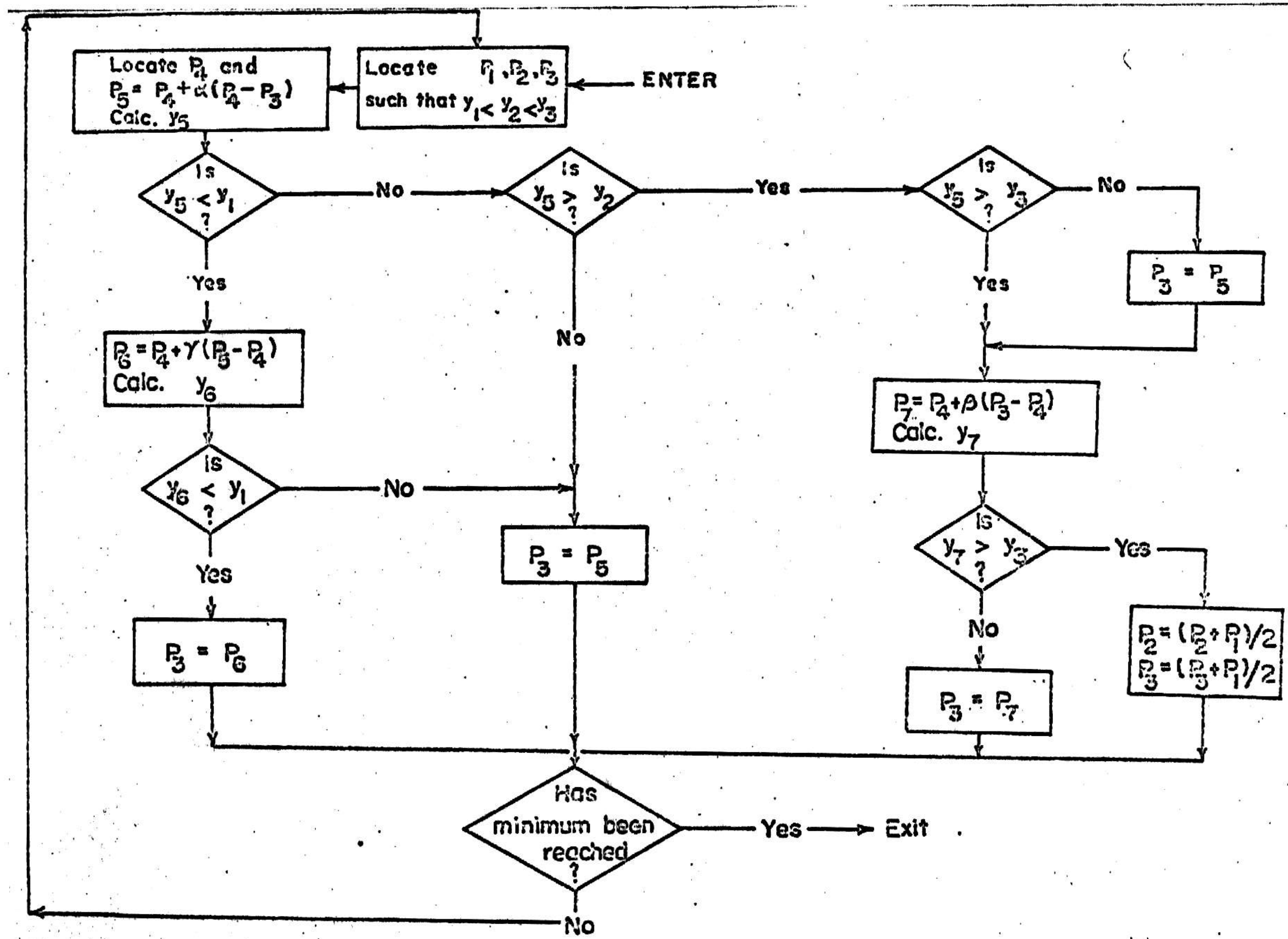


Fig. A-2 Flow diagram for the simplex pattern search method.

Table A-2. Symbol Table

Program Symbol	Explanation
N	Number of decision variables
PCTR(I)	A vector of decision variables
D(I)	A vector of perturbation size for starting initial simplex
ITER	Number of iterations
ITOUT	Interval of output iterations
ITMAX	Maximum number of iterations
DELTA	Accuracy level for stopping criterion
A(I)	Sales rate; I = 1, 2, ..., n
CI(I)	Inventory level; I = 1, 2, ..., n
NOPT	Number of objective function evaluation
NORFT	Number of reflection move
NOEXP	Number of expansion move
NOCNT	Number of contraction move
NOCVGT	Number of convergence in a simplex
SY	Standard deviation in a value of objective functions
Y(I)	Value of objective function at a point P(I) in a simplex I = 1, 2, ..., N+1
YF	Value of objective function
YM	Average function value of a simplex
YMIN	The minimum function value in a simplex
PMIN(J)	A point in a simplex which gives minimum function value

```
C      COMPUTER PROGRAM FOR SIMPLEX PATTERN SEARCH
C      APPLICATION TO TWENTY DIMENSIONAL PROBLEM
C
       DIMENSION P(45,40),Y(45),PCTR(40),D(40),PP(3,40),A(40)
       COMMON A
  1000 FORMAT(4I5<
       READ 1000,N,ITOUT,ITMAX,DELTA
  1100 FORMAT(2F10.2<
       READ 1100,(PCTR(I),D(I),I=1,N)
  1200 FORMAT(5F10.2<
       READ 1200,(A(I),I=1,10)
C      **READ IN ADDITIONAL DATA .
C      CALL READIN*---<
       NOPT=N+1
       MULT=1
       ITER=0
       NORFT=0
       NOEXP=0
       NOCNT=0
       NOCVGT=0
       YM=0.0
       SY=0.0
       FN=N
       NM=N+1
C      **SET UP INITIAL SIMPLEX .
       DO 4 J=1,N
       DO 1 I=1,J
     1 P(I,J)=PCTR(J)-D(J)
       FJ=J
       P(J+1,J)=PCTR(J)+FJ*D(J)
       IF(J-N)2,4,4
     2 JM=J+2
       DO 3 I=JM,NM
     3 P(I,J)=PCTR(J)
     4 CONTINUE
       DO 6 I=1,NM
       DO 5 J=1,N
     5 PCTR(J)=P(I,J)
C      CALL CHECK1*---<
       CALL OBJFN(PCTR,N,YF)
C      CALL CHECK2*---<
     6 Y(I)=YF
C      **REARRANGE ORDER .
       I=1
       NS=N+1
     7 IF(Y(I)-Y(NS))10,8,8
     8 YTEM=Y(NS)
       Y(NS)=Y(I)
       Y(I)=YTEM
       DO 9 J=1,N
       PCTR(J)=P(NS,J)
       P(NS,J)=P(I,J)
     9 P(I,J)=PCTR(J)
    10 IF(NS-I-1)12,12,11
    11 NS=NS-1
       GO TO 7
    12 I=I+1
       IF(I-N-1)13,14,14
```

```
   13 NS=N+1
      GO TO 7
   14 IOPTN=5
      DO 15 J=1,N
   15 PCTR(J)=P(1,J)
      YMIN=Y(1)
      CALL OUTPUT( IOPTN,ITER,NOPT,NOEXP,NOREFT,NOCNT,YM,SY,NOCVGT,PCTR,
     1YMIN,ITOUT,MULT,N)
C     **COMPUTE CENTROID OF THE SIMPLEX .
  160 FN=N
      DO 17 J=1,N
      PXT=P(1,J)
      DO 16 I=2,N
   16 PXT=PXT+P(I,J)
   17 P(N+2,J)=PXT/FN
C     **MAKE REFLECTION MOVE .
      DO 21 J=1,N
      P(N+3,J)=P(N+2,J)+1.0*(P(N+2,J)-P(N+1,J))
   21 PCTR(J)=P(N+3,J)
C     CALL CHECK1%---<
      CALL OBJFN(PCTR,N,YF)
C     CALL CHECK2%---<
      Y(N+2)=YF
      NOPT=NOPT+1
      IF(Y(N+2)-Y(1))30,22,22
   22 IF(Y(N+2)-Y(N))23,40,40
   23 DO 24 I=1,N
   24 P(N+1,I)=P(N+3,I)
      Y(N+1)=Y(N+2)
      ITER=ITER+1
      NOREFT=NOREFT+1
      GO TO 100
C     **MAKE EXPANSION MOVE .
   30 DO 31 J=1,N
      P(N+4,J)=P(N+2,J)+2.0*(P(N+3,J)-P(N+2,J))
   31 PCTR(J)=P(N+4,J)
C     CALL CHECK1%---<
      CALL OBJFN(PCTR,N,YF)
C     CALL CHECK2%---<
      Y(N+3)=YF
      NOPT=NOPT+1
      IF(Y(N+3)-Y(1))32,32,23
   32 DO 33 I=1,N
   33 P(N+1,I)=P(N+4,I)
      Y(N+1)=Y(N+3)
      ITER=ITER+1
      NOEXP=NOEXP+1
      GO TO 100
   40 IF(Y(N+2)-Y(N+1))41,50,50
   41 DO 42 I=1,N
   42 P(N+1,I)=P(N+3,I)
      Y(N+1)=Y(N+2)
      ITER=ITER+1
      NOREFT=NOREFT+1
C     **MAKE CONTRACTION MOVE .
   50 DO 51 J=1,N
      P(N+5,J)=P(N+2,J)+0.5*(P(N+1,J)-P(N+2,J))
   51 PCTR(J)=P(N+5,J)
      CALL OBJFN(PCTR,N,YF)
      Y(N+4)=YF
```

```
      NOPT=NOPT+1
      IF(Y(N+4)-Y(N+1))52,60,60
   52 DO 53 I=1,N
   53 P(N+1,I)=P(N+5,I)
      Y(N+1)=Y(N+4)
      ITER=ITER+1
      NOCNT=NOCNT+1
      NOCVGT=NOCVGT+1
      GO TO 110
C     **CUT DOWN STEP-SIZE .
   60 DO 62 I=2,NN
      DO 61 J=1,N
      P(I,J)=(P(1,J)+P(I,J))/2.0
   61 PCTR(J)=P(I,J)
      CALL OBJFN(PCTR,N,YF)
   62 Y(I)=YF
C     **REARRANGE ORDER .
      I=1
      NS=N+1
   63 IF(Y(I)-Y(NS))66,64,64
   64 YTEM=Y(NS)
      Y(NS)=Y(I)
      Y(I)=YTEM
      DO 65 J=1,N
      PCTR(J)=P(NS,J)
      P(NS,J)=P(I,J)
   65 P(I,J)=PCTR(J)
   66 IF(NS-I-1)68,68,67
   67 NS=NS-1
      GO TO 63
   68 I=I+1
      IF(I-N-1)69,70,70
   69 NS=N+1
      GO TO 63
   70 NOPT=NOPT+N
      NOCVGT=NOCVGT+1
      IOPTM=4
      DO 75 I=1,N
   75 PCTR(I)=P(1,I)
      YMIN=Y(1)
      CALL OUTPUT( IOPTM,ITER,NOPT,NOEXP,NOREF,NOCNT,YM,SY,NOCVGT,PCTR,
     1YMIN,ITOUT,MULT,N)
      GO TO 120
  100 NOCVGT=0
C     **REARRANGE ORDER .
  110 ICR=N
  111 IF(Y(ICR+1)-Y(ICR))112,120,120
  112 YTEM=Y(ICR+1)
      Y(ICR+1)=Y(ICR)
      Y(ICR)=YTEM
      DO 113 J=1,N
      PCTR(J)=P(ICR+1,J)
      P(ICR+1,J)=P(ICR,J)
  113 P(ICR,J)=PCTR(J)
      IF(ICR-1)120,120,114
  114 ICR=ICR-1
      GO TO 111
C     **TEST FOR OPTIMALITY .
  120 YT=Y(1)
      FNN=NN
```

```
      DO 121 I=2,NM
  121 YT=YT+Y(I)
      YM=YT/FNM
      SY=(Y(1)-YM)**2
      DO 122 I=2,NM
  122 SY=SY+(Y(I)-YM)**2
      SY=(SY/FN)**0.5
      IF(SY-DELTA)123,123,124
  123 IF(NOCVGT-2)126,125,125
  124 IOPTM=2
      GO TO 130
  125 IOPTM=1
      GO TO 130
  126 IOPTM=3
  130 N=N
      DO 131 I=1,N
  131 PCTR(I)=P(1,I)
      YMIN=Y(1)
      CALL OUTPUT( IOPTM,ITER,NCPT,NCEXP,NORFT,NOCNT,YM,SY,NOCVGT,PCTR,
     1YMIN,ITOUT,MULT,N)
      IF(IOPTM-1)150,150,140
  140 IF(ITMAX-ITER)150,150,160
  150 STOP
      END
```

```
      SUBROUTINE OBJFN(PCTR,N,YF)
      DIMENSION PCTR(40),PR(40),W(40),CI(40),A(40)
      COMMON A
      CI(1)=263.+PCTR(1)-A(1)
      DO 1 I=2,10
    1 CI(I)=CI(I-1)+PCTR(I)-A(I)
      S1=340.*(PCTR(11))+64.3*(PCTR(11)-91.)**2+.2*(PCTR(1)-5.67*(PCTR(1
     11)))**2+51.2*(PCTR(1))-281.*(PCTR(11))+0.0825*(CI(1)-320.)**2
      YF=S1
      DO 2 J=2,10
    2 YF=YF+340.*(PCTR(J+10))+64.3*(PCTR(J+10)-PCTR(J+9))**2+.2*(PCTR(J)
     1-5.67*PCTR(J+10))**2+51.2*(PCTR(J))-281.*(PCTR(J+10))+0.0825*(CI(J
     1)-320.0)**2
      RETURN
      END
```

```
      SUBROUTINE OUTPUT(ICPIN,ITER,NOPT,NOEXP,NORFT,NOCNT,YM,SY,NOCVGT,
     1PMIN,YMIN,ITOUT,MULT,N)
      DIMENSION PMIN(40),PR(40),W(40),CI(40),X(40)
      COMMON A
  100 FORMAT(5I8,2F12.3,I8,3X9H* OPTIMUM<
  200 FORMAT(5I8,2F12.3,I8<
  500 FORMAT(5I8,24%1H-<,I8,3X13H* START POINT<
  700 FORMAT(4X4HNOPT4X4HITER3X5HNORFT3X5HNOEXP3X5HNOCNT7X2HYM10X2HSY5X6
     1HNOCVGT4X6HREMARK<
  710 FORMAT(10X2HX%I3,3H< #F10.3,2H ,<
  720 FORMAT(10X6HYMIN #F15.3,//<
      IF(ICPIN-2)10,20,25
   25 IF(ICPIN-5)29,50,29
   10 PRINT 700
      PRINT 100,NOPT,ITER,NORFT,NOEXP,NOCNT,YM,SY,NOCVGT
      PRINT 710,(J,PMIN(J),J=1,N)
      PRINT 720,YMIN
      RETURN
   20 IF(ITER-ITOUT*MULT)29,21,21
   21 PRINT 700
      PRINT 200,NOPT,ITER,NORFT,NOEXP,NOCNT,YM,SY,NOCVGT
      PRINT 710,(J,PMIN(J),J=1,N)
      PRINT 720,YMIN
      MULT=MULT+1
   29 RETURN
   50 PRINT 700
      PRINT 500,NOPT,ITER,NORFT,NOEXP,NOCNT,NOCVGT
      PRINT 710,(J,PMIN(J),J=1,N)
      PRINT 720,YMIN
      RETURN
      END
```

APPENDIX III. Computer Program for Fletcher and Powell Method

The computer flow chart which illustrates the computational procedure of the method is illustrated in Fig.A-3;the program symbols, their explanations and corresponding mathematical notations are summarized in Table A-3.The computer program for the solution of twenty dimensional problem follows the symbol table.

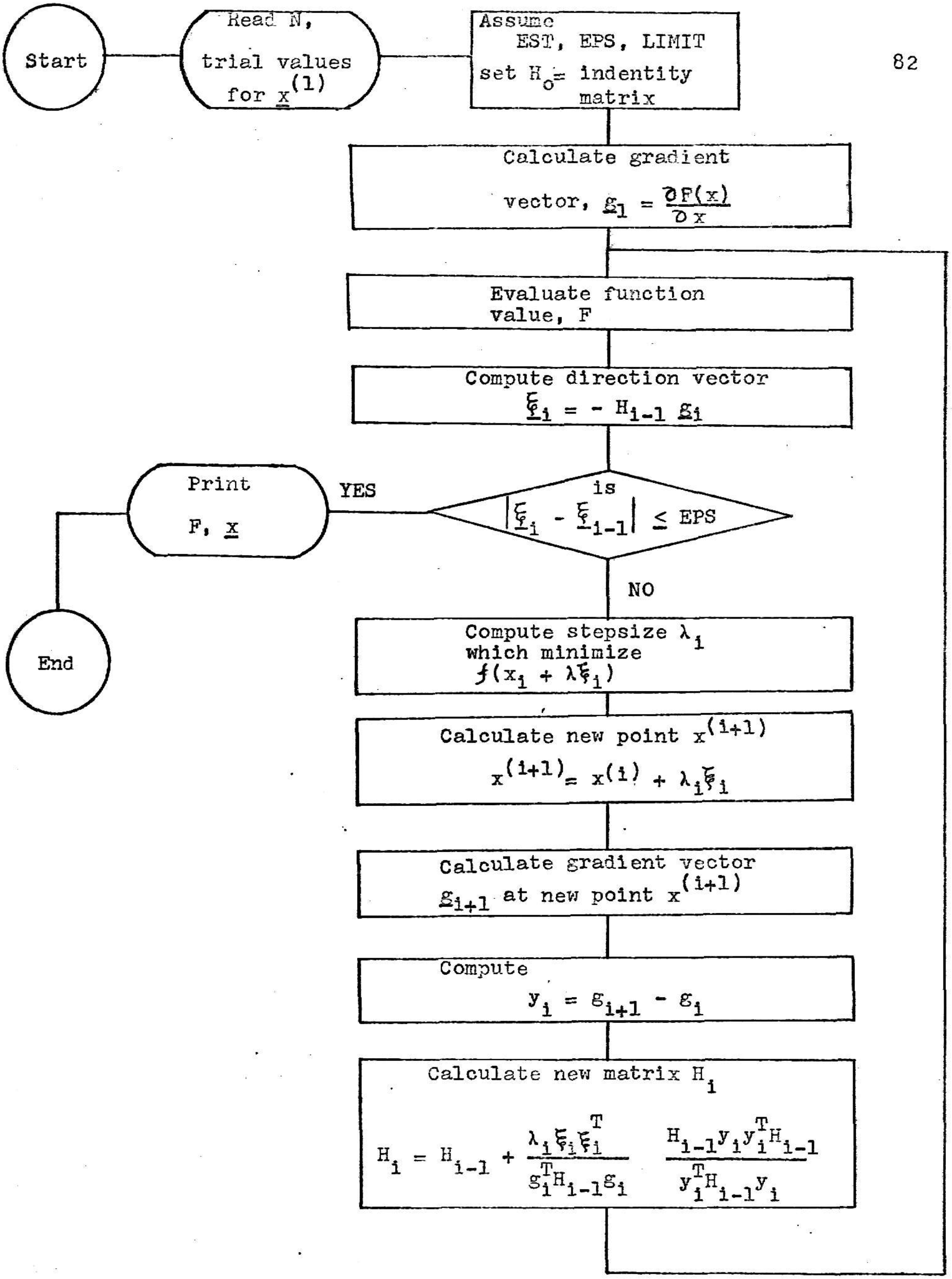


Fig. A-3. Flow diagram for Fletcher and Powell method.

Table A-3. Symbol Table

Program Symbol	Explanation	Mathematical Symbol
N	Number of decision variables	
X(I)	A vector of decision variables	x_n; $n = 1, 2, \ldots, N$
Q(I)	Sales rate; I = 1, 2, ..., m	
CI(I)	Inventory level; I = 1, 2, ..., m, m = No. of periods	I_n; $n = 1, 2, \ldots, m$
EST	Estimate of minimum function value	
EPS	Accuracy level for stopping criterion	ϵ
LIMIT	Maximum No. of iterations	
F	Function value	$F(x)$
G(I)	Gradient vector; I = 1, 2, ..., N	g_n; $n = 1, 2, \ldots, N$
H(I)	Direction vector; I = 1, 2, ..., N	ξ_n; $n = 1, 2, \ldots, N$
AMBDA	Stepsize	λ
H	A matrix to approximate G^{-1}	H

```
C       COMPUTER PROGRAM FOR FLETCHER AND POWELL METHOD
C       APPLICATION TO TWENTY DIMENSIONAL PROBLEM
C
        EXTERNAL FUNCT
        DIMENSION X(30),G(30),C(30),CI(30),H(270)
        COMMON KOUNT,CI,C
        READ 100,(C(I),I=1,10)
   100  FORMAT(10F5.1)
        N=20
        DO 10 I=1,10
    10  X(I)=300.
        DO 20 J=11,20
    20  X(J)=50.
        EPS=0.1
        EST=300000.0
        LIMIT=100
        CALL FMFP(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
   700  FORMAT(1H-,'      FLETCHER AND POWELL METHOD        ')
        PRINT 700
   600  FORMAT(1H-,'           P                 W                 I          ')
        PRINT 600
   200  FORMAT(/3X,F10.3,5X,F10.3,5X,F10.3)
        DO 5 I=1,10
     5  PRINT 200,X(I),X(I+10),CI(I)
    30  FORMAT(//6X9HMINIMUM= F10.3)
        PRINT 30,F
   300  FORMAT(//6X6HKOUNT=I3)
        PRINT 300,KOUNT
        STOP
        END
```

```
      SUBROUTINE FUNCT(N,X,F,G)
      DIMENSION X(30),G(30),CI(30),Q(30),T(30)
      COMMON KOUNT,CI,Q
      CI(1)=263.0+X(1)-Q(1)
      DO 1 I=2,10
    1 CI(I)=CI(I-1)+X(I)-Q(I)
      S1=340.0*X(11)+64.3*(X(11)-81.0)**2+0.2*(X(1)-5.67*X(11))**2+51.2*
     1X(1)-281.0*X(11)+.0825*(CI(1)-320.0)**2
      F=S1
      DO 2 J=2,10
    2 F=F+340.*X(J+10)+64.3*(X(J+10)-X(J+9))**2+.2*(X(J)-5.67*X(J+10))**
     12+51.2*X(J)-281.*X(J+10)+.0825*(CI(J)-320.)**2
      DO 100 I=1,9
  100 T(I)=0.0
      DO 10 I=1,10
   10 T(1)=T(1)+0.165*(CI(I)-320.0)
      DO 20 I=2,10
   20 T(2)=T(2)+0.165*(CI(I)-320.0)
      DO 30 I=3,10
   30 T(3)=T(3)+0.165*(CI(I)-320.0)
      DO 40 I=4,10
   40 T(4)=T(4)+0.165*(CI(I)-320.0)
      DO 50 I=5,10
   50 T(5)=T(5)+0.165*(CI(I)-320.0)
      DO 60 I=6,10
   60 T(6)=T(6)+0.165*(CI(I)-320.0)
      DO 70 I=7,10
   70 T(7)=T(7)+0.165*(CI(I)-320.0)
      DO 80 I=8,10
   80 T(8)=T(8)+0.165*(CI(I)-320.0)
      DO 90 I=9,10
   90 T(9)=T(9)+0.165*(CI(I)-320.0)
      T(10)=0.165*(CI(10)-320.0)
      DO 200 I=1,10
  200 G(I)=.4*(X(I)-5.67*X(I+10))+51.2+T(I)
      G(11)=128.6*(X(11)-81.)+340.-2.268*(X(1)-5.67*X(11))-281.-128.6*(X
     1(12)-X(11))
      DO 4 J=12,19
    4 G(J)=128.6*(X(J)-X(J-1))-2.268*(X(J-10)-5.67*X(J))+340.0-281.-128.
     16*(X(J+1)-X(J))
      G(20)=128.6*(X(20)-X(19))-2.268*(X(10)-5.67*X(20))-281.+340.
      RETURN
      END
```

```
      SUBROUTINE FMFP(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
      DIMENSION H(1),X(1),G(1),CI(30),C(30)
      COMMON KOUNT,CI,C
      CALL FUNCT(N,X,F,G)
      IER=0
      KOUNT=0
      N2=N+N
      N3=N2+N
      N31=N3+1
    1 K=N31
      DO 4 J=1,N
      H(K)=1.
      NJ=N-J
      IF(NJ)5,5,2
    2 DO 3 L=1,NJ
      KL=K+L
    3 H(KL)=0.
    4 K=KL+1
    5 KOUNT=KOUNT +1
      OLDF=F
      DO 9 J=1,N
      K=N+J
      H(K)=G(J)
      K=K+N
      H(K)=X(J)
      K=J+N3
      T=0.
      DO 8 L=1,N
      T=T-G(L)*H(K)
      IF(L-J)6,7,7
    6 K=K+N-L
      GO TO 8
    7 K=K+1
    8 CONTINUE
    9 H(J)=T
      DY=0.
      HNRM=0.
      GNRM=0.
      DO 10 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
   10 DY=DY+H(J)*G(J)
      IF(DY)11,51,51
   11 IF(HNRM/GNRM-EPS)51,51,12
   12 FY=F
      ALFA=2.*(EST-F)/DY
      AMBDA=1.
      IF(ALFA)15,15,13
   13 IF(ALFA-AMBDA)14,15,15
   14 AMBDA=ALFA
   15 ALFA=0.
   16 FX=FY
      DX=DY
      DO 17 I=1,N
   17 X(I)=X(I)+AMBDA*H(I)
      CALL FUNCT(N,X,F,G)
      FY=F
      DY=0.
      DO 18 I=1,N
   18 DY=DY+G(I)*H(I)
```

```
      IF(DY)19,36,22
   19 IF(FY-FX)20,22,22
   20 AMBDA=AMBDA+ALFA
      ALFA=AMBDA
      IF(HNRM*AMBDA-1.E10)16,16,21
   21 IER=2
      RETURN
   22 T=0.
   23 IF(AMBDA)24,36,24
   24 Z=3.*(FX-FY)/AMBDA+DX+DY
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF(DALFA)51,25,25
   25 W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 250,251,250
  250 ALFA=(DY-Z+W)/ALFA
      GO TO 252
  251 ALFA=(Z+DY-W)/(Z+DX+Z+DY)
  252 ALFA=ALFA*AMBDA
      DO 26 I=1,N
   26 X(I)=X(I)+(T-ALFA)*H(I)
      CALL FUNCT(N,X,F,G)
      IF(F-FX)27,27,28
   27 IF(F-FY)36,36,28
   28 DALFA=0.
      DO 29 I=1,N
   29 DALFA=DALFA+G(I)*H(I)
      IF(DALFA)30,33,33
   30 IF(F-FX)32,31,33
   31 IF(DX-DALFA)32,36,32
   32 FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 23
   33 IF(FY-F)35,34,35
   34 IF(DY-DALFA)35,36,35
   35 FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 22
   36 IF(OLDF-F+EPS) 51,38,38
   38 DO 37 J=1,N
      K=N+J
      H(K)=G(J)-H(K)
      K=N+K
   37 H(K)=X(J)-H(K)
      IER=0
      IF(KOUNT-N)42,39,39
   39 T=0.
      Z=0.
      DO 40 J=1,N
      K=N+J
      W=H(K)
      K=K+N
      T=T+ABS(H(K))
   40 Z=Z+W*H(K)
      IF(HNRM-EPS)41,41,42
```

```
41 IF(T-EPS)56,56,42
42 IF(KOUNT-LIMIT)43,50,50
43 ALFA=0.
   DO 47 J=1,N
   K=J+N3
   W=0.
   DO 46 L=1,N
   KL=N+L
   W=W+H(KL)*H(K)
   IF(L-J)44,45,45
44 K=K+N-L
   GO TO 46
45 K=K+1
46 CONTINUE
   K=N+J
   ALFA=ALFA+W*H(K)
47 H(J)=W
   IF(Z*ALFA)48,1,48
48 K=N31
   DO 49 L=1,N
   KL=N2+L
   DO 49 J=L,N
   NJ=N2+J
   H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
49 K=K+1
   GO TO 5
50 IER=1
   RETURN
51 DO 52 J=1,N
   K=N2+J
52 X(J)=H(K)
   CALL FUNCT(N,X,F,G)
   IF(GNRM-EPS)55,55,53
53 IF(IER)56,54,54
54 IER=-1
   GOTO 1
55 IER=0
56 RETURN
   END
```

APPENDIX IV. Computer Program for Fletcher and Reeves Method

The computer flow chart which illustrates the computational procedure of the method is illustrated in Fig.A-4;the program symbols, their explanations and corresponding mathematical notations are summarized in Table A-4.The computer program for the solution of twenty dimensional problem follows the symbol table.

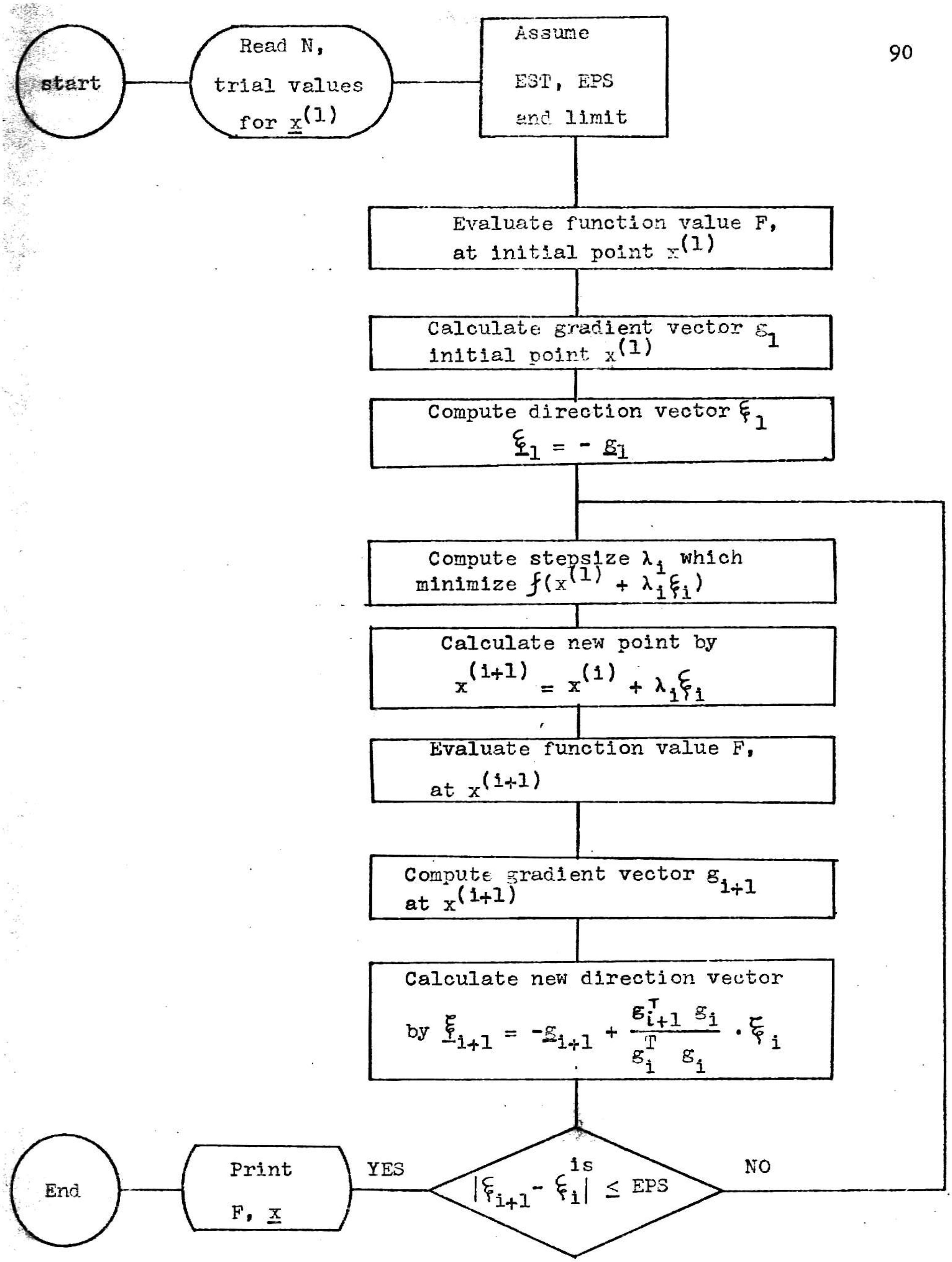


Fig. A-4. Flow diagram for Fletcher and Reeves method.

Table A-4. Symbol Table

Program Symbol	Explanation	Mathematical Symbol
N	Number of decision variables	
X(I)	A vector of decision variables	x_n; $n = 1, 2, \ldots, N$
Q(I)	Sales rate; I = 1, 2, ..., m	
CI(I)	Inventory level; I = 1, 2, ..., m m = No. of periods	I_n; $n = 1, 2, \ldots, m$
EST	Estimate of minimum function value	
EPS	Accuracy level for stopping criterion	ϵ
LIMIT	Maximum No. of iterations	
F	Function value	$F(x)$
G(I)	Gradient vector; I = 1, 2, ..., N	g_n; $n = 1, 2, \ldots, N$
H(I)	Direction vector; I = 1, 2, ..., N	ξ_n; $n = 1, 2, \ldots, N$
AMBDA	Stepsize	λ

```
C      COMPUTER PROGRAM FOR FLETCHER AND REEVES METHOD
C      APPLICATION TO TWENTY DIMENSIONAL PROBLEM
C
       EXTERNAL FUNCT
       DIMENSION X(30),G(30),Q(30),CI(30),H(270)
       COMMON KOUNT,CI,Q
       READ 100,(Q(I),I=1,10)
   100 FORMAT(10F5.1)
       N=20
       DO 10 I=1,10
    10 X(I)=300.
       DO 20 J=11,20
    20 X(J)=50.
       EPS=0.1
       EST=300000.0
       LIMIT=100
       CALL FMCG(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
   700 FORMAT(1H-,'      FLETCHER AND REEVES METHOD        ')
       PRINT 700
   600 FORMAT(1H-,'         P                  W                 I            ')
       PRINT 600
   200 FORMAT(/3X,F10.3,5X,F10.3,5X,F10.3)
       DO 5 I=1,10
     5 PRINT 200,X(I),X(I+10),CI(I)
    30 FORMAT(//6X9HMINIMUM= F10.3)
       PRINT 30,F
   300 FORMAT(//6X6HKOUNT=I3)
       PRINT 300,KOUNT
       STOP
       END
```

```
      SUBROUTINE FUNCT(N,X,F,G)
      DIMENSION X(30),G(30),CI(30),Q(30),T(30)
      COMMON KOUNT,CI,Q
      CI(1)=263.0+X(1)-Q(1)
      DO 1 I=2,10
    1 CI(I)=CI(I-1)+X(I)-Q(I)
      S1=340.0*X(11)+64.3*(X(11)-81.0)**2+0.2*(X(1)-5.67*X(11))**2+51.2*
     1X(1)-281.0*X(11)+.0825*(CI(1)-320.0)**2
      F=S1
      DO 2 J=2,10
    2 F=F+340.*X(J+10)+64.3*(X(J+10)-X(J+9))**2+.2*(X(J)-5.67*X(J+10))**
     12+51.2*X(J)-281.*X(J+10)+.0825*(CI(J)-320.)**2
      DO 100 I=1,9
  100 T(I)=0.0
      DO 10 I=1,10
   10 T(1)=T(1)+0.165*(CI(I)-320.0)
      DO 20 I=2,10
   20 T(2)=T(2)+0.165*(CI(I)-320.0)
      DO 30 I=3,10
   30 T(3)=T(3)+0.165*(CI(I)-320.0)
      DO 40 I=4,10
   40 T(4)=T(4)+0.165*(CI(I)-320.0)
      DO 50 I=5,10
   50 T(5)=T(5)+0.165*(CI(I)-320.0)
      DO 60 I=6,10
   60 T(6)=T(6)+0.165*(CI(I)-320.0)
      DO 70 I=7,10
   70 T(7)=T(7)+0.165*(CI(I)-320.0)
      DO 80 I=8,10
   80 T(8)=T(8)+0.165*(CI(I)-320.0)
      DO 90 I=9,10
   90 T(9)=T(9)+0.165*(CI(I)-320.0)
      T(10)=0.165*(CI(10)-320.0)
      DO 200 I=1,10
  200 G(I)=.4*(X(I)-5.67*X(I+10))+51.2+T(I)
      G(11)=128.6*(X(11)-81.)+340.-2.268*(X(1)-5.67*X(11))-281.-128.6*(X
     1(12)-X(11))
      DO 4 J=12,19
    4 G(J)=128.6*(X(J)-X(J-1))-2.268*(X(J-10)-5.67*X(J))+340.0-281.-128.
     16*(X(J+1)-X(J))
      G(20)=128.6*(X(20)-X(19))-2.268*(X(10)-5.67*X(20))-281.+340.
      RETURN
      END
```

```
      SUBROUTINE FMCG(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
      DIMENSION X(1),G(1),H(1),CI(30),C(30)
      COMMON KOUNT,CI,C
      CALL FUNCT(N,X,F,G)
      KOUNT=0
      IER=0
      N1=N+1
    1 DO 43 II=1,N1
      KOUNT=KOUNT+1
      OLDF=F
      GNRM=0.
      DO 2 J=1,N
    2 GNRM=GNRM+G(J)*G(J)
      IF(GNRM)46,46,3
    3 IF(II-1)4,4,6
    4 DO 5 J=1,N
    5 H(J)=-G(J)
      GO TO 8
    6 AMBDA=GNRM/OLDG
      DO 7 J=1,N
    7 H(J)=AMBDA*H(J)-G(J)
    8 DY=0.
      HNRM=0.
      DO 9 J=1,N
      K=J+N
      H(K)=X(J)
      HNRM=HNRM+ABS(H(J))
    9 DY=DY+H(J)*G(J)
      IF(DY)10,42,42
   10 SNRM=1./HNRM
      FY=F
      ALFA=2.*(EST-F)/DY
      AMBDA=SNRM
      IF(ALFA)13,13,11
   11 IF(ALFA-AMBDA)12,13,13
   12 AMBDA=ALFA
   13 ALFA=0.
   14 FX=FY
      DX=DY
      DO 15 I=1,N
   15 X(I)=X(I)+AMBDA*H(I)
      CALL FUNCT(N,X,F,G)
      FY=F
      DY=0.
      DO 16 I=1,N
   16 DY=DY+G(I)*H(I)
      IF(DY)17,38,20
   17 IF(FY-FX)18,20,20
   18 AMBDA=AMBDA+ALFA
      ALFA=AMBDA
      IF(HNRM*AMBDA-1.E10)14,14,19
   19 IER=2
      F=OLDF
      DO 100 J=1,N
      G(J)=H(J)
      K=N+J
  100 X(J)=H(K)
      RETURN
   20 T=0.
   21 IF(AMBDA)22,38,22
```

```
   22 Z=3.*(FX-FY)/AMBDA+DX+DY
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF(DALFA)23,27,27
   23 DO 24 J=1,N
      K=N+J
   24 X(J)=H(K)
      CALL FUNCT(N,X,F,G)
   25 IF(IER)47,26,47
   26 IER=-1
      GOTO 1
   27 W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 270,271,270
  270 ALFA=(DY-Z+W)/ALFA
      GO TO 272
  271 ALFA=(Z+DY-W)/(Z+DX+Z+DY)
  272 ALFA=ALFA*AMBDA
      DO 28 I=1,N
   28 X(I)=X(I)+(T-ALFA)*H(I)
      CALL FUNCT(N,X,F,G)
      IF(F-FX)29,29,30
   29 IF(F-FY)38,38,30
   30 DALFA=0.
      DO 31 I=1,N
   31 DALFA=DALFA+G(I)*H(I)
      IF(DALFA)32,35,35
   32 IF(F-FX)34,33,35
   33 IF(DX-DALFA)34,38,34
   34 FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 21
   35 IF(FY-F)37,36,37
   36 IF(DY-DALFA)37,38,37
   37 FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 20
   38 IF(OLDF-F+EPS) 19,25,39
   39 OLDG=GNRM
      T=0.
      DO 40 J=1,N
      K=J+N
      H(K)=X(J)-H(K)
   40 T=T+ABS(H(K))
      IF (KOUNT-N1) 42,41,41
   41 IF (T-EPS) 45,45,42
   42 IF (KOUNT-LIMIT) 43,44,44
   43 IER=0
      GO TO 1
   44 IER=1
      IF(GNRM-EPS)46,46,47
   45 IF((GNRM-EPS).LE.0.) GO TO 46
      IF(IER.NE.0) GO TO 47
      IER=-1
      GO TO 1
   46 IER=0
```

```
   47 RETURN
      END
```

A COMPARATIVE STUDY OF OPTIMIZATION TECHNIQUES APPLIED TO INDUSTRIAL MANAGEMENT SYSTEMS

by

VIJAY H. MEHTA

B.S.(MECH. ENGG.), Sardar Patel University,
Vallabh Vidyanagar, India, 1967

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

1969

In this report a comparison of the four well known unconstrained optimization techniques is presented. The four selected techniques are gradient technique, simplex pattern search, Fletcher and Powell method and Fletcher and Reeves method. The production planning problem and the production and employment scheduling problem represent the typical problems of the industrial management systems. For this reason they are selected as test problems in this study. To see the effect of these four techniques on the dimensionality of the optimization problem, one of the test problem considered is two dimensional problem and another is twenty dimensional problem. The second test problem is the well known Holt, Modigliani, Muth and Simon paint factory model.

The basic theory and procedure of each technique is described together with the results of both the test problems. The four different criteria, namely, the optimal objective function value, the total computation time, number of iterations and required computer memory storage are used to compare the behavior and effectiveness of these techniques.

The results show that Fletcher and Powell method and Fletcher and Reeves method gave the highest convergence rate among the four techniques in both the test problems. It is seen that they have the least effect on increasing the dimensionality of the problem. The gradient technique proves itself third best with regard to all the four criteria and also in the effect of increasing the dimensionality of the problem. Although the simplex pattern search is an efficient search technique for low dimensional problems, it seems that the technique is inadequate for large dimensional optimization problems.