74-6995

LEE, Liang-sun, 1943-
LINEAR APPROXIMATION IN CONJUGATE GRADIENT
METHODS FOR PROCESS OPTIMIZATION.

The University of Oklahoma, Ph.D., 1973
Chemistry, general

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

LINEAR APPROXIMATION IN CONJUGATE GRADIENT METHODS FOR PROCESS OPTIMIZATION

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

LIANG-SUN LEE

Norman, Oklahoma

1973

LINEAR APPROXIMATION IN CONJUGATE GRADIENT

METHODS FOR PROCESS OPTIMIZATION

APPROVED BY

Dr. James H. Christensen (Chairman)

Dr. Kenneth E. Starling

Dr. Carl E. Locke

Dr. William N. Huff

Dr. Henry B. Crichlow

DISSERTATION COMMITTEE

ACKNOWLEDGEMENTS

I wish to express my special sincere thanks and appreciation to Dr. James H. Christensen, one of my best teachers, for his advice, direction and patience through out this work; and to his wife, Mary, for her understanding and offering warm time during my life in Norman.

I would also like to thank Dr. Kenneth E. Starling, Dr. Carl E. Locke, Dr. Henry B. Crichlow and Dr. William N. Huff for their teaching and willingness to serve on my advisory committee.

I would like to give my deepest gratitude to my parents, wife, brothers, and sisters, for their love, encouragement and confidence.

Finally, I also thank The Department of Chemical Engineering and Material Science for part of financial support.

ABSTRACT

The general nonlinear programming problem is a problem which maximizes(or minimizes) an objective function subject to a set of nonlinear equality and inequality constraints. Many methods have been developed to handle certain types of programming problems. A method, based upon Rosen's gradient projection method and conjugate direction of Fletcher and Reeves is presented here for solving maximization problem of nonlinear objective function subject to linear constraints. In some cases, optimal points of problems with nonlinear constraints are also located by this revised algorithm after nonlinear constraints are linearized at each iteration.

The necessary condition for increasing objective function in the search direction is discussed. The most important consideration, the satisfaction of Kuhn-Tucker conditions is also proved in chapter II.

It happens that a feasible point may become infeasible with respect to the new set of relinearized constraints. To avoid the infeasibility, a step back from the infeasible region into the feasible region is necessary. The criterion of step back is given in chapter III.

Different problems which have the optimal point located in the internal region of constraints, at an intersection of constraints and on a single constraint, have been solved successfully.

A chemical process optimization problem is also solved in this dissertation. This process consists of reactor, heat exchanger, decanter, and distillation column. Three irreversible chemical reactions were considered in the reactor. The final solution gives the optimal temperature in the reactor, optimal volume of the reactor, and optimal flow rates of feed and recycle.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LINEAR APPROXIMATION IN CONJUGATE GRADIENT METHODS FOR PROCESS OPTIMIZATION

CHAPTER I

INTRODUCTION

Literature Survey

The theory and application of mathematical programming have drawn deep interest and attention since the simplex method was discovered by Dantzig (13) in 1951. Two years later, the simplex method was revised by Dantzig, Orchard-Hays and others at Rand (14) to overcome the cumbersome operations of the original method. In 1951, an important paper (35) appeared, in which the well-known necessary conditions for a constrained maximum were given by Kuhn and Tucker by relating the non-linear programming problem to an equivalent saddle point problem. In two decades, the field of mathematical programming has been developed very successfully both in the theory and in the solution of practical problems.

Several bibliographies and general methods for solving the general problems as well as its subproblems have been published. All the methods can be sorted into two main categories. The first category contains the methods contributing

to the solving of unconstrained problems. It has two subclasses: direct search methods and large step gradient methods. In the second category are the techniques for solving the general constrained problem. Many techniques in the second category are based upon the modifications of existing methods for unconstrained problems (Rosen,1960;Goldfarb,1968), or transformation of the general constrained problem into an unconstrained problem, which is then solved by the use of existing methods for unconstrained problems.

The steepest ascent(descent) method is the essential technique in the first category. It has the shortcoming that the objective function oscillates when the optimal point is approached. Davidon (15) developed a variable metric method, which forced the point to move in a better direction than steepest ascent and guaranteed finding the maximum of general quadratic problem in a finite number of steps. Fletcher and Powell (21) improved Davidon's method by reforming the variable metric. In 1967, another algorithm, also based upon Davidon's idea, was developed by Broyden (6). It is called a rank one method, because the difference between the (k+1)-th and k-th variable metric is a symmetric matrix of rank one. Some authors used the property of conjugacy, such as Hestenes and Stiefel's method (30) of conjugate gradients for solving linear systems. It is very interesting that the direction vectors generated by Davidon's algorithm and the conjugate gradients algorithm of Hestenes and Stiefel are scalar mul-

tiples of each other, provided the initial step each takes is in the direction of steepest descent. Fletcher and Reeves (22) used a set of directions which are conjugate to the Hessian matrix. The current direction for the movement of point is formed by the linear combination of the current gradient and old directions. Pearson (45) also proposed several ways of computing a variable metric using search directions which are conjugate. Three different variable metrics were obtained and new algorithms were set up. Smith's work (53) belongs to this subclass also. Actually, some methods for solving constrained problems also used the conjugate property, such as, Goldfarb (25) and Zoutendijk (62).

The properties of the Hessian matrix have been used to develop improved algorithms. Greenstadt (26) guaranteed that an estimate of the inverse of the Hessian matrix would be positive definite by a procedure of eigenvalue analysis. He also derived a general relation (27) for differentiating two successive variable metrics by minimizing the norm of the difference and modified the variable metric. Other schemes suggested to maintain a positive definite estimate of the inverse Hessian matrix belong to Marquardt (37) and Zwart (63). In 1970, Fletcher (20) used the relation of variable metric and gradient and the property that the variable metric is an approximation of the Hessian inverse, and that the eigenvalues of both matrices are approximately the same. He derived an algorithm as effective as Fletcher and Powell's.

Another interesting approach (38) was done by Miele and Cantrell. They proposed a method of search in which two parameters were selected to minimize the objective function in each search direction. For a quadratic function this method is the same as Fletcher and Reeves' but takes longer to evaluate since two dimensional search on each stage is involved. Cragg and Levy (11) extended Miele and Cantrell's two parameters method to a greater number of parameters. A case of four parameters has been tested; the result is better than that of two parameters.

So far the methods mentioned above are all developed from classical gradient calculations. Techniques obtained through a different approach are the direct search methods. In this group, not so many methods can be found as those given above. Hooke and Jeeves proposed the pattern method (32) for unconstrained minimization. This method consists of two major searches, an exploratory search around the base point and a pattern search according to an acceleration rule. The directions used in the first phase are the coordinate axes. A full cycle must be performed before the second phase is involved. Powell (46), (47) developed a method from the work of Smith (53) to locate the minimum of a convex quadratic function by successive unidimensional search along a set of conjugate directions. All directions generated are conjugate to the Hessian matrix. In 1960, Rosenbrock (49) used a set of orthonormal directions generated by the Gram-Smith procedure;

his method is also a successive unidimensional search. Actually, if Rosenbrock's method is applied to a quadratic function, it behaves somewhat like a conjugate direction method. Rosenbrock's method was modified by Davis, Swann and Compey (55) by locating the minimum of the objective function in each direction instead of using a step length. This revised method behaves similarly to Fletcher and Powell's search as described by Swann. By using the analytical geometry approach, Spendley, Hext, and Himsworth suggested a simplex method (54) of optimization that sequentially projected the worst point, which is the vertex that gives the worst objective function, in a simplex through the centroid of the remaining points. Nelder and Mead (42) proposed a more efficient simplex method following the idea of Spendley, et al. In their paper the change of the simplex or finding the new vertex can be carried out by reflection, contraction and expansion procedures.

All the methods described above are those for solving unconstrained problems. The techniques belonging to the second category are those implemented to solve constrained problems. The earliest technique for solving the constrained problems was proposed by Frank and Wolfe (23). This is a method of feasible directions and can be applied to solve problems with linear constraints only. Later on, Wolfe (58) and Beale (2) also developed methods for quadratic programming problems. Wolfe's method is similar to Beale's but can not handle non-

concave objective functions. Others who contributed to the research in this class should include Frish (24) and Lemke (36).

As mentioned earlier, some techniques for constrained optimization were developed via the idea of algorithms for solving unconstrained problems. The most well-known are those of Box (5), Rosen (48), Morrison (40) and Goldfarb (25). Rosen's very famous gradient projection method was developed from the idea of a gradient into a linear manifold formed by the intersection of constraints in the basis, which consists of the active constraints encountered by the moving point. Box's complex method is due to the idea of Spendley's simplex method and can be applied to solve problems with inequality constraints. In his method a polyhedron with (n+1) vertices was selected instead of n vertices. Morrison used a least squares technique to solve nonlinear problems with equality constraints. In his method constrained problems are transformed into unconstrained problems by introducing a additional set of parameters. Also Goldfarb developed conjugate gradient method based upon Davidon's variable metric method and provided many of the details of the matrix manipulation. This method can solve problems with linear constraints.

Sometimes, an established method was modified to obtain better efficiency of to extend its application to different types of problems. Stewart and Griffith (28) suggested the MAP algorithm. They extended the use of linear programming to solve nonlinear programming problems by linearizing the

objective function and constraints at the local optimal point of a linear programming subproblem. Yang (60) improved the MAP method by applying GFP technique to linearize constraints. In this improved method the explicit calculation of partial derivatives is not required. The time-consuming procedure and the difficulty for linearizing constraints are avoided during solving problems. Miller (39) also modified and extended the use of linear programming to solve separable programming problems. In 1965, Abadie and Carpentier developed the general reduced gradient method. This method is an extension of Wolfe's reduced gradient method. In this method a set of dependent variables and another set of independent variables are formed. The dependent variables are implicitly determined by independent variables. So far they have not found any problem which the general reduced gradient method was not able to solve. Kowalik, Osborne and Ryan (34) modified Morrison's method by using heavyside functions to change inequality constraints into equality constraints, then solved the problem after it is converted into unconstrained problem. The work of Murtag and Sargent (41) also belongs to this class. Their work, based on Rosen's gradient projection method and Fletcher and Powell's modification of Davidon's method, considered a number of methods which make it possible to update the inverse Hessian for steps of arbitrary length and direction.

Some other developments should be mentioned individually.

Zoutendijk (62) proposed a feasible direction method which can handle linear as well as nonlinear inequality constraints but not equality constraints. A linear programming subproblem has to be solved in order to keep the new point in the feasible region and attain the greatest improvement in the value of the objective function at each iteration. Kelley developed the cutting plane method (33) for solving convex programs, based on the idea that the optimal solution could be represented as the intersection of a set of half-spaces. Actually, the cutting plane method was also developed by Cheney and Goldstein (8) independently. In 1960, Carroll proposed the created response surface technique(CRST), converting a constrained programming problem into a series of nonlinear unconstrained problems. The most important characteristic of this approach is that it automatically avoids constraint violations during the optimization. Fiacco and McCormick (18), (19) developed the SUMT technique based on transforming a given constrained minimization problem into a sequence of unconstrained problems. This method is different from Carroll's CRST in the treatment of constraints. It has been used extensively in solving minimization problems. A different approach, using the property of infeasibility, has been accomplished by Paviani and Himmelblau (44). This is the flexible tolerance method. It improved the value of the objective function by using information provided at feasible points and at near-feasible points. The near-feasibility

limits are gradually made more restricted as the search proceeds toward the solution of the programming problem. In the field of linear programming, Saksena and Cole (50) proposed a method that permits movement in either the feasible or infeasible region of the given problem in the search for the optimal solution. The initial point also can be infeasible. The most recent method was developed by Westerberg and Debrosse (56). In this algorithm the set of inequality constraints is divided into three sets: the set of active constraints, the set of constraints which are active but should be released and the set of nonactive constraints. Certain criteria determine the movement of constraints from one set to another to obtain an improved objective function. They also developed an algorithm (16) for finding an initial feasible solution of the programming problem. The initial feasible point is very important, since most methods for solving constrained optimization problems have to start from a feasible point.

Examples in Process Optimization

The mathematical description of the optimization of a chemical process design is well suited to the mathematical programming formulation, as shown by the following three examples of nonlinear programming problems in chemical engineering. All have nonlinear objective functions, the first is without constraints, while the second has linear con-

straints and the third has nonlinear constraints.

The first example is to find the best fitted analytical equations for thermodynamic properties (43). In this problem, the coefficients of an equation are determined to minimize the deviation of the predicted values from given data. For example, let the predicted enthalpy be

$$H = x_1 + x_2T + x_3T^2 + x_4T^3 + x_5T^4 + x_6T^5$$

where T is the temperature and x_i, i=1,...,6 are derived coefficients. The nonlinear programming problem can be formulated as:

$$\underset{x_1,\ldots,x_6}{\text{Minimize}} \sum_{k=1}^{N} (x_1 + x_2T_k + x_3T_k^2 + x_4T_k^3 + x_5T_k^4 + x_6T_k^5 - H_k^*)^2 \qquad (1\text{-}1)$$

where H_k^* is the experimentally determined at the k-th data point at a temperature of T_k. This is an unconstrained nonlinear programming problem.

The second one is the determination of the equilibrium composition of a mixture of ideal gases at constant temperature and pressure. A solution can be obtained by minimizing the total Gibbs free energy of the System (25). Suppose there are m species and k elements in the system, the problem can be written as

$$\text{Minimize} \quad f(\underline{x}) = \sum_{i=1}^{m} x_i (c_i + \log x_i/x^*) \qquad (1\text{-}2)$$

$$\text{Subject to} \quad \sum_{i=1}^{m} a_{ij} x_i = b_j \qquad j=1,\ldots,k \qquad (1\text{-}3)$$

and

$$x_i \geq 0$$

where

$$x^* = \sum_{i=1}^{m} x_i$$

and

$$c_i = F_i^o/RT + \ln P$$

where x_i is the number of moles of the i-th species, P is the total pressure, T is the temperature, R is the universal gas constant, and F_i^o/RT is the standard molal free energy of the i-th species. In equation (1-3) a_{ij} is the number of atoms of the j-th element in the i-th species and b_j is the number of moles of the j-th element originally present in the mixture. This is a problem of nonlinear objective function with linear constraints.

The third example is the optimization of a chemical process (17). All design variables are determined to maximize the percentage return on investment. This process is shown in the block diagram Fig. 1, it consists of a stirred-tank reactor, a heat exchanger, a decanter and a distillation column. Two pure inputs and a recycle are fed into the reactor to yield a mixture of six components leaving the

reactor. The reactions in the reactor are

$$A + B \xrightarrow{k_1} C$$

$$B + C \xrightarrow{k_2} P + E \qquad (1\text{-}4)$$

$$C + P \xrightarrow{k_3} G$$

where k_i, i=1,2,3 are the reaction coefficients, and can be evaluated by the Arrhenius equations:

$$k_i = A_i \exp(-B_i/T) \qquad (1\text{-}5)$$

The usual assumption of perfect mixing in the reactor is given. The temperature of the reactor is controlled between 580° and 680° Rankine. The effluent of the reactor contains raw materials A and B, an intermediate C an inert E, a residual product G and desired product P. This mixture is cooled down in the heat exchanger and pumped into the decanter in which the residual product G is removed. The desired product P is obtained from the top of the distillation column. The recovery of product is not complete since the mixture at the bottom forms an azeotrope. A portion of the bottoms product of the distillation column is discarded to control the concentration of the inert E, the rest of it is recycled to the reactor.

This plant manufactures 40 millions pounds per year of distillate product P. Dibella and Stevens determined the

optimal values of 12 design variables F_A, F_B, F_D, F_G, F_P, F_{RA}, F_{RB}, F_{RC}, F_{RE}, F_{RP}, V and T, which give the maximal value of the objective function under a set of design constraints. The objective function is considered as the percent return on the investment.

$$f = 100\ \{8400(0.3F_P - 0.0068F_D - 0.02F_A - 0.03F_B - 0.01F_G) - 2.22F_R - 0.124(8400)(0.3F_P + 0.0068 F_D) - 60V\rho\}\ /600 - V\rho \qquad (1\text{-}6)$$

The constraints are equalities and can be set up by using the material balance over individual components and the whole process.

1. Overall material balance

$$h_1 = F_A + F_B - F_G - F_P - F_D = 0 \qquad (1\text{-}7)$$

2. Azeotropic separation in distillation column

$$h_2 = F_{RP} - 0.1F_{RE} - F_P = 0 \qquad (1\text{-}8)$$

3. Material balance over component E

$$h_3 = (M_E/M_B)k_2(F_{RB}F_{RC}/F_R^2)V\rho - F_DF_{RE}/(F_R - F_G - F_D) = 0 \qquad (1\text{-}9)$$

4. Material balance over component P

$$h_4 = \left[k_2F_{RB}F_{RC} - (M_P/M_C)k_3F_{RC}F_{RP}\right](V\rho/F_R^2) -$$

$$F_D(F_{RP} - F_P)/(F_R - F_G - F_P) - F_P = 0 \qquad (1\text{-}10)$$

5. Material balance over component A

$$h_5 = -(k_1F_{RA}F_{RB})(V\rho/F_R^2) - F_DF_{RA}/(F_R - F_G - F_P) + F_A = 0 \qquad (1\text{-}11)$$

6. Material balance over component B

$$h_6 = -(k_1F_{RB}F_{RA} + k_2F_{RB}F_{RC})(V\rho/F_R^2) - F_DF_{RB}/(F_R - F_G - F_P) + F_B = 0 \qquad (1\text{-}12)$$

7. Material balance over component C

$$h_7 = \left[(M_C/M_B) - k_1F_{RA}F_{RB} - (M_E/M_B)k_2F_{RB}F_{RC} - k_3F_{RP}F_{RC}\right](V\rho/F_R^2) - F_DF_{RC}/(F_R - F_G - F_P) = 0 \qquad (1\text{-}13)$$

8. Material balance over component G

$$h_8 = (M_G/M_C)k_3F_{RC}F_{RP}(V\rho/F_R^2) - F_G = 0 \qquad (1\text{-}14)$$

9. Definition of F_R

$$h_9 = F_{RA} + F_{RB} + F_{RC} + F_{RE} + F_{RP} + F_G - F_R = 0 \qquad (1\text{-}15)$$

where V is the volume of the reactor and ρ is the density of reactor solution. All variables must be greater than or equal to zero. The range of temperature is from 580° to 680°

Rankine. The values of all constants in the system are given in table I.

Dibella and Stevens (17) solved this process optimization problem by using the MAP method. Yang (60) applied the improved MAP algorithm to obtain an improved solution over that of Dibella and Stevens. In this dissertation, this problem is also solved by the revised gradient projection algorithm. The result is given in chapter IV.

Previous Works and Description of Present Method

As mentioned earlier, techniques for solving constrained problems can be developed from the established methods for unconstrained problems or by modification of existing methods for constrained problems. Yang's improved MAP method (60) is a good example of the latter approach. In his method Yang introduced the GFP technique into conventional MAP method of Griffith and Stewart (28). This improvement saved very much computational effort for the evaluation of the Jacobian matrix at each iteration, especially for large problems and those for which analytical partial differentiation are not easily performed.

The conjugate gradient method of Goldfarb and Lapidus (25) is an extension of Davidon's variable metric method for solving unconstrained optimization problems. It also can be considered as a modification of Rosen's gradient projection method, in which the direction of the movement of the point

was replaced by the direction used in the variable metric method. Some evaluation examples in Goldfarb's paper showed that the conjugate gradient method takes fewer steps to reach optimum than does the gradient projection method, but it requires more matrix manipulations and computer memory.

The algorithm presented in this thesis is also a revised Rosen's gradient projection method or can be considered as an extended Fletcher and Reeves' conjugate direction method (22). The movement of the point is in Fletcher and Reeves' conjugate direction before any of the linear constraints is encountered, or in the direction of the projected conjugate direction when a constraint basis exists. The comparison of the number of steps to obtain optimal point exhibits that this new algorithm gives more rapid convergence than original gradient projection method does, since the conjugate direction does not zigzag as the gradient direction when the optimum is approached. The computational results, given in chapter IV, showed that this revised algorithm is at least as efficient as conjugate gradient method, or even slightly better. The measures of the superiority of the new revised algorithm to the conjugate gradient method are the fewer matrix manipulations and the smaller computer storage.

This revised method is also applicable to solve nonlinear programming problems with convex nonlinear constraints after linearization at each iteration. The GFP technique is used for linearization. Based upon the criterion of the number of

linearizations required to reach the optimum, the revised gradient projection method is far better than Yang's improved MAP. The remarkable evidence is the treatment of interior optimum problems. A large number of iterations is required for the MAP method, but not for the revised gradient projection method. The reason is because either MAP or improved MAP method solved the subproblems of linear programming, the bound range used in the subproblems can not be too large due to the fixed direction of the gradient of the objective function at each iteration. If the selected range is too large, it becomes more difficult to reach the optimum.

The algorithm of this revised method is given in chapter III. The numerical results and comparisons with other methods are given in chapter IV.

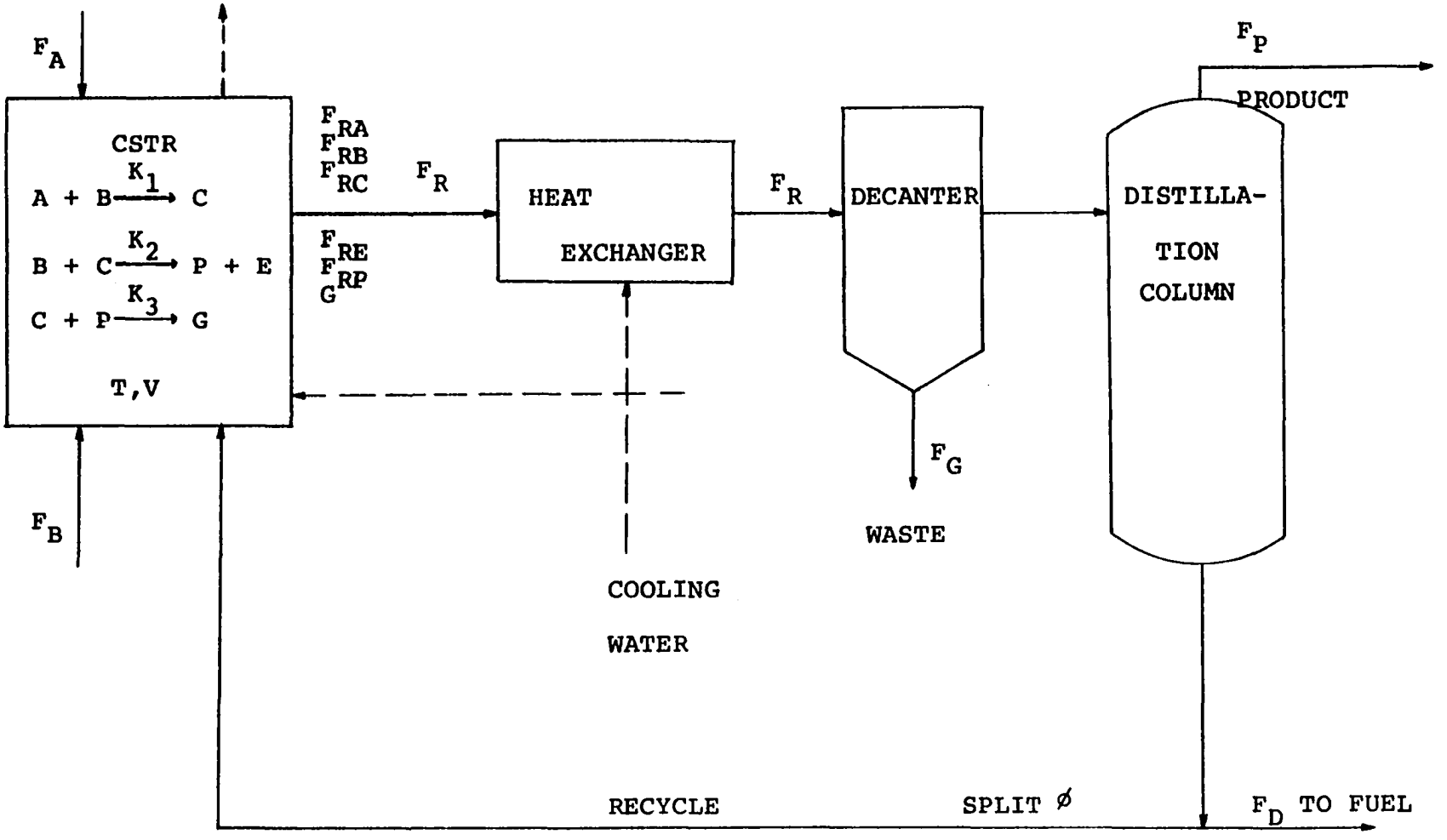


Fig. 1 Flow Sheet of Chemical Process Optimization Example

TABLE I

CONSTANTS IN CHEMICAL PROCESS OPTIMIZATION EXAMPLE

Name	Value	Unit
F_P	4763	lb/hr
A_1	5.9755×10^9	hr^{-1}
A_2	2.5962×10^{12}	hr^{-1}
A_3	9.6283×10^{15}	hr^{-1}
B_1	12000	^oR
B_2	15000	^oR
B_3	20000	^oR
M_B	1000	lb
M_C	200	lb
M_E	200	lb
M_G	300	lb
M_P	100	lb
ρ	50	lb/cu.ft

CHAPTER II

ALGORITHM FOR THE LINEARLY CONSTRAINED PROBLEM

Formulation of NLP problems with Linear Constraints

The standard form of a nonlinear programming problem with linear constraints can be expressed as

$$\text{Maximize} \quad f(\underline{x}) = f(x_1,\ldots\ldots,x_m) \tag{2-1}$$

$$\text{Subject to} \quad \begin{aligned} &\sum_{j=1}^{m} n_{ij}x_j \geq b_i && i=1,\ldots.,k \\ &\sum_{j=1}^{m} n_{ij}x_j = b_i && i=k+1,\ldots,n \end{aligned} \tag{2-2}$$

where the n_{ij} have been normalized such that

$$\sum_{j=1}^{m} n_{ij}^2 = 1 \qquad i=1,\ldots..n \tag{2-3}$$

We may consider the variables x_i, $i=1,\ldots,m$ represent a point in the m-dimensional Euclidean space, E^m, in the finite dimensional geometry. Therefore, a set of variables and its corresponding coefficients of constraints can be written as column vectors. Relations (2-2) becomes

$$\underline{n}_i'\underline{x} \geq b_i \qquad i=1,\ldots,k$$
$$\underline{n}_i'\underline{x} = b_i \qquad i=k+1,\ldots,n \tag{2-4}$$

where

$$\underline{n}_i = (n_{i1},\ldots\ldots,n_{im}) \tag{2-5}$$

A surplus value λ_i is defined for constraint i

$$\underline{n}_i'\underline{x} - b_i = \lambda_i(\underline{x}) \geq 0 \tag{2-6}$$

$\lambda_i=0$, when point $\underline{x}$ lies on the constraint i, that is, when constraint i is in the constraint basis. Point $\underline{x}$ is in the infeasible region with respect to constraint i, if $\lambda_i<0$; and it is a feasible point, if all $\lambda_i \geq 0$.

A projection matrix is a m x m symmetric matrix defined as

$$P_q = I - N_q(N_q'N_q)^{-1}N_q' \tag{2-7}$$

where

$$N_q = \left[\underline{n}_1,\ldots\ldots\ldots,\underline{n}_q\right] \tag{2-8}$$

is the constraint basis, consisting of the constraints which are contained in the basis, i.e., constraints with corresponding $\lambda=0$. N_q is an m x q ordered metric. The sequence 1,2,....,q is determined by the order in which the constraints are encountered by the moving point.

In geometric terms, the linear inequalities and equalities of m variables represent half-spaces and hyperplanes, respectively, in m-dimensional Euclidean space. All these half-spaces and hyperplanes representing constraints will form a polyhedral region R in the Euclidean space, called the feasible region. If the region R is closed, then the programming problem will be bounded, otherwise, the solution of the problem may be unbounded.

Any point $\underline{x}$ inside or on the boundary of region R, i.e., $\underline{x} \varepsilon R$, is called a feasible point or feasible solution. The moving point always stays in the feasible region during the solution of problem with this algorithm. The term "constraint basis" is a set of linearly independent hyperplanes, which are the constraints with equality relations at the current point. A manifold M_q is formed by the intersection of those linearly independent hyperplanes and will restrict the movement of the point. The intersection of any two linearly independent hyperplanes is an (m-2)-dimensional manifold of E^m. Similarly, the manifold M_q formed by the constraint basis is (m-q)-dimensional. A line is an (m-(m-1))-dimensional manifold of E^m and has one degree of freedom. A point is a zero dimensional manifold of E^m and has no degrees of freedom.

Fletcher and Reeves' Direction for Maximization

Assume a function can be expanded in the form (2-9) if

higher terms are neglected.

$$f(\underline{x}) = f(\underline{x}^o) + \underline{g}'(\underline{x}^o)(\underline{x}-\underline{x}^o) + \tfrac{1}{2}(\underline{x}-\underline{x}^o)'H(\underline{x}^o)(\underline{x}-\underline{x}^o) \tag{2-9}$$

where $\underline{g}(\underline{x}^o)$ is gradient of the function at point $\underline{x}^o$,

$$\underline{g}(\underline{x}^o) = (\partial f/\partial \underline{x}^o)_{\underline{x}^o}$$

and matrix H is called Hessian matrix, the element H_{ij} of Hessian matrix is defined as

$$H_{ij} = (\partial^2 f/\partial x_i \partial x_j)_{\underline{x}^o}$$

The Hessian matrix is symmetric. If a function is quadratic, it can be written in the form

$$f(\underline{x}) = \underline{g}'\underline{x} + \tfrac{1}{2}\underline{x}'Q\underline{x} \tag{2-10}$$

Where $\underline{g}$ is a vector, Q is a symmetric matrix. Both $\underline{g}$ and Q are constants. It is obvious that the coefficient vector of the first term in right hand side is the gradient vector of the function, and matrix Q equals to Hessian matrix.

The term conjugate direction means that if a set of directions $\underline{p}_o,\ldots,\underline{p}_n$ are conjugate to Hessian matrix then the following relation is satisfied

$$\underline{p}_i'H\underline{p}_j = 0 \qquad \text{if } i \neq j \tag{2-11}$$

Fletcher and Reeves demonstrated that if any set of H-conjugate directions are used, the method of successive

linear search is quadratically convergent. The minimum is located at the n-th iteration for quadratic functions.

If some modifications are made, Fletcher and Reeves' algorithm is also applicable to maximization problem. The following algorithm is for maximization.

1). Select arbitrary initial point $\underline{x}^o$.

2). Calculate $g(\underline{x}^o)$, and set $\underline{p}_o = \underline{g}(\underline{x}^o)$.

3). If $\underline{x}^{i+1}$ is the maximum of $f(\underline{x})$ in the direction of $\underline{p}_i$, i.e., $f(\underline{x}^{i+1}) = \max_{\eta \geq 0} f(\underline{x}^i + \eta \underline{p}_i)$.

4). Calculate $\underline{g}(\underline{x}^{i+1})$.

5). Let $\beta_i = \underline{g}_{i+1}^2 / \underline{g}_i^2$.

6). $\underline{p}_{i+1} = \underline{g}_{i+1} + \beta_i \underline{p}_i$

In the current algorithm, if the optimal point is in the interior of closed region R and none of the constraints stays in the constraint basis during the searching for the optimum, then the movement of the point will be in Fletcher and Reeves' conjugate direction. This is because the projection matrix is the identity matrix I if none of the constraints is encountered.

Determination of Entering Plane

While the point is moving in a certain direction, a constraint, among one of the nonbasis constraints, may be encountered which stops the moving point from leaving the closed region R and going into infeasible region. This may

be due to the possibility of the best objective function in the moving direction being in the infeasible region or on the constraint. If the movement of the point is not restricted, infeasibility will occur and the algorithm fails to obtain the constrained optimal point. Figure 2 shows the encounter of a constraint.

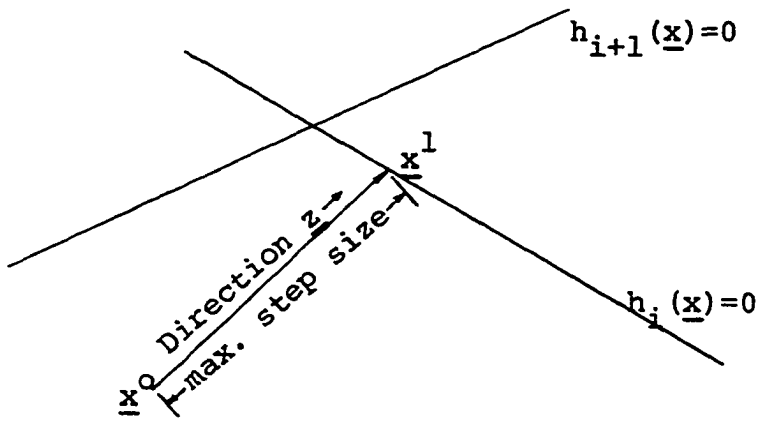


Fig. 2 Determination of Entering Plane

The choice of a constraint coming to the constraint basis is as follow.

$$\underline{x}^1 = \underline{x}^o + \tau \underline{z} \tag{2-12}$$

Where $\underline{x}^1$ is new point, τ is maximum step length and $\underline{z}$ is an unit direction vector. According to (2-6)

$$\lambda_i^o = \underline{n}_i' \underline{x}^o - b_i \tag{2-13}$$

$$\lambda_i^1 = \underline{n}_i' \underline{x}^1 - b_i \tag{2-14}$$

Substract eq.(2-13) from eq. (2-14), get

$$\lambda_i^1 - \lambda_i^o = \underline{n}_i'(\underline{x}^1 - \underline{x}^o) = \tau \underline{n}_i' \underline{z} \tag{2-15}$$

Since $\lambda_i^1 \geq 0$ is necessary condition for feasiblity. Therefore

$$\lambda_i^o + \tau \underline{n}_i' \underline{z} \geq 0$$

and

$$\tau > -\lambda_i^o / \underline{n}_i' \underline{z} \geq 0 \tag{2-16}$$

Since $\lambda_i^o \geq 0$, then $\underline{n}_i' \underline{z}$ must be less than zero. The maximum step length is

$$\tau_{max.} = \min |\lambda_i^o / \underline{n}_i' \underline{z}| \geq 0, \qquad \underline{n}_i' \underline{z} < 0 \tag{2-17}$$

and the coming plane is the constraint which has the smallest value of τ.

During solving a real problem, it is necessary to interpolate between old point and new point with largest step size to find τ_{op}, the best step. The constraint relating to λ_i should come in the constraint basis if $\tau_{op} = \tau_{max}$. Otherwise the addition of the constraint is not required.

Recursion Formula

In the course of Rosen's (48) gradient projection algorithm, the recalculation of the $(N_q' N_q)^{-1}$ matrix is necessary each time a hyperplane is added to or removed from the constraint basis.

Since it is time consuming if matrix $(N_q'N_q)^{-1}$ is calculated directly from N_q at each iteration, it will be very helpful if direct calculation can be avoided.

Rosen used the formula for the inverse of a matrix in terms of the inverses of its partitions and showed how to calculate $(N_{q-1}'N_{q-1})^{-1}$ from $(N_q'N_q)^{-1}$ and $(N_q'N_q)^{-1}$ from $(N_{q-1}'N_{q-1})^{-1}$ with approximately q^2 and $2q^2+mq$ multiplications and divisions.

Suppose the q x q nonsingular inverse matrix $(N_q'N_q)^{-1}$ is known and partitioned as

$$\left[N_q'N_q\right]^{-1} = \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} \tag{2-18}$$

where B_1, B_2, B_3 and B_4 are (q-1 x q-1), (q-1 x 1), (1 x q-1) and (1 x 1) matrices respectively. In particularly, $B_2=B_3'$. The required recursion formula for $(N_{q-1}'N_{q-1})^{-1}$ when constraint q is dropped from the constraint basis is

$$\left[N_{q-1}'N_{q-1}\right]^{-1} = B_1 - B_2B_4^{-1}B_3 \tag{2-19}$$

In case of the plane to be dropped, say H_l, is not the last one, H_q, in the constraint basis, then the l-th and q-th row and column of $(N_q'N_q)^{-1}$ must be interchanged before relation (2-19) is applied. Since $(N_q'N_q)^{-1}$ is a symmetric matrix, the interchange finds a new $(N_q'N_q)^{-1}$ with H_l and H_q in interchanged order in the constraint basis N_q.

The procedure to find $(N_q'N_q)^{-1}$ from $(N_{q-1}'N_{q-1})^{-1}$ when

hyperplane H_q is added to the constraint basis is

$$\left[N_q' N_q\right]^{-1} = \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix}$$

where

$$\begin{aligned} B_1 &= \left[N_{q-1} N_{q-1}\right]^{-1} + A_o^{-1} \underline{r}_{q-1} \underline{r}_{q-1}' \\ B_2 &= -A_o^{-1} \underline{r}_{q-1} = B_3' \\ B_4 &= A_o^{-1} \\ A_o &= \underline{n}_q' P_{q-1} \underline{n}_q \\ \underline{r}_{q-1} &= \left[N_{q-1}' N_{q-1}\right]^{-1} (N_{q-1}' \underline{n}_q) \end{aligned} \tag{2-20}$$

In solving a real problem, if the initial feasible point is an interior point of closed convex region R, then the constraint basis is an empty set, there is no equality relation existing, the directions are not projected, and the projection matrix P_o is chosen to be the identity matrix I. When the initial point lies on q linearly independent hyperplanes, then equation (2-20) is used q times to build up the projection matrix P_q.

Determination of the Plane to be Removed

Rosen has shown that if point $\underline{x}$ lies on the manifold M_q of linearly independent hyperplanes H_i, i=1,...q, and satisfies the following condition, then the hyperplane H_q should

be removed from the constraint basis.

$$|P_q \underline{g}| \leq \tfrac{1}{2} r_q b_{qq}^{-\frac{1}{2}} \tag{2-21}$$

where $r_q b_{qq}^{-\frac{1}{2}} \geq r_i b_{ii}^{-\frac{1}{2}}$, $i=1,\ldots.q-1$, and where b_{ii} is the i-th diagonal element of $(N_q' N_q)^{-1}$, and only $b_{ii}>0$ is considered. Where r_q is the q-th element of column vector

$$\underline{R} = \left[N_q' N_q\right]^{-1} N_q' \underline{g} \tag{2-22}$$

According to Rosen's proof, this is because point $\underline{x}$ is on the intersection of q-1 linearly independent hyperplanes, H_i, $i=1,\ldots,q-1$, but not on the manifold M_q of q linearly independent hyperplanes, provided relation (2-21) is met. The hyperplane H_q should be dropped and new projection matrix P_{q-1} must be found. The formulation of matrix P_{q-1} can be established after $(N_{q-1}' N_{q-1})^{-1}$ is obtained by using recursion formulate (2-19).

Remember that if the plane to be dropped H_q is not the last one in constraint basis, the interchange of 1-th and q-th rows and columns of $(N_q' N_q)^{-1}$ must be accomplished before relation (2-19) is applied.

Condition for Increasing Objective Function

It is well known that the gradient, if it is not zero, points in a direction such that a small movement in that direction will increase the value of objective function. But

how does a direction $\underline{p}$ other than gradient increase the value of the objective function is a question. The necessary condition for increasing the value of the objective function in the direction $\underline{p}$ is given by Zangwill (61).

Suppose function $f(\underline{x})$ is differentiable at $\underline{x}$, and there is a direction $\underline{p}$ such that

$$\underline{g}'(\underline{x})\underline{p} > 0$$

Then a $t > 0$, $t \geq \tau > 0$, exists such that

$$f(\underline{x}+\tau\underline{p}) > f(\underline{x})$$

This can be proved in the following way.

If $f(\underline{x})$ is differentiable at point $\underline{x}$, by calculus

$$\frac{d}{d\tau} f(\underline{x}+\tau\underline{p})\Big|_{\tau=0} = \lim_{\tau\to 0}\frac{f(\underline{x}+\tau\underline{p}) - f(\underline{x})}{\tau}\Big|_{\tau=0} = \underline{g}'(\underline{x})\underline{p} \tag{2-23}$$

Via equation (2-23), if

$$\lim_{\tau\to 0} \frac{f(\underline{x}+\tau\underline{p}) - f(\underline{x})}{\tau}\Big|_{\tau=0} > 0$$

Then there must be a $t>0$, such that for all $\tau \neq 0$, and $t \geq \tau > -t$

$$\frac{f(\underline{x}+\tau\underline{p}) - f(\underline{x})}{\tau} > 0$$

If $\tau > 0$ is selected, then we may have $f(\underline{x}+\tau\underline{p}) > f(\underline{x})$.

The above statement shows that any given direction $\underline{p}$ will increase the value of the objective function with a small step in that direction if $\underline{g}'(\underline{x})\underline{p}>0$. The obvious example is that if the gradient is chosen as search direction, i.e., $\underline{p}=\underline{g}$, then $\underline{g}'(\underline{x})\underline{p}=\|\underline{g}(\underline{x})\|^2$, therefore a small movement in the gradient direction will increase the value of the objective function.

When this revised algorithm is applied to solve a real problem, this condition is always held, so that the value of the objective function increases monotonically. This can be checked whenever a new point and new direction are obtained.

Constrained Maximum and Interior Maximum

If the global maximal point exists in the interior of convex closed region R, it is called an interior maximum of the objective function. If the global maximum is found on the boundary of closed region, the global maximal point is called a constrained maximum.

Rosen (48) has proved that the necessary conditions for both a constrained maximum and an interior maximum of a concave objective function are

$$P_q\underline{g}(\underline{x}^*) = 0 \tag{2-24}$$

and

$$\underline{R} = (N_q'N_q)^{-1}N_q'\underline{g}(\underline{x}^*) \leq 0 \tag{2-25}$$

Relation (2-24) can be considered that at global maximum the gradient must be orthogonal to the manifold M_q, i.e., $\underline{g}(\underline{x}^*)=N_q\underline{R}$.

To prove the necessary condition of interior maximum, it can be considered the same as unconstrained global maximum. As well known, the necessary condition that $\underline{x}^*$ is the unconstrained maximum is $\underline{g}(\underline{x}^*)=0$. This requirement is also matched by conditions (2-24) and (2-25). The sufficiency of (2-24) and (2-25) is easily shown for concave objective functions.

Let $\underline{x}$ be any point in the closed region. Using property of concavity of function we obtain

$$f(\underline{x}) < f(\underline{x}^*) + \underline{g}'(\underline{x}^*)(\underline{x} - \underline{x}^*) \tag{2-26}$$

Since $\underline{g}(\underline{x}^*)=0$, then $f(\underline{x})<f(\underline{x}^*)$. Thus $\underline{x}^*$ maximizes $f(\underline{x})$.

To prove the sufficient condition for a constrained maximum, assume there is a point $\underline{x}^1$ in closed region R such that $f(\underline{x}^1)>f(\underline{x}^*)$. Let unit direction be $\underline{z}$ and $\underline{x}^1=\underline{x}^*+\tau\underline{z}$. Since $\underline{x}^1$ is a point in closed region R, the condition that $N_q'\tau\underline{z}\geq 0$ is required, see Rosen (48).

From equation (2-26) and assumption we get

$$f(\underline{x}^1) - f(\underline{x}^*) > \underline{g}(\underline{x}^*)(\underline{x}^1 - \underline{x}^*) = \underline{g}'(\underline{x}^*)\tau\underline{z} \tag{2-27}$$

The relation (2-24) can be rewritten as

$$\left[I - N_q(N_q'N_q)^{-1}N_q'\right]\underline{g}(\underline{x}^*) = 0$$

This can be considered that $\underline{g}(\underline{x}^*)$ is given by the linear com-

bination of constraints in the basis, or in the form

$$\underline{g}(\underline{x}^*) = N_q(N_q'N_q)^{-1}N_q'\underline{g}(\underline{x}^*)$$

or

$$\underline{g}(\underline{x}^*) = N_q\underline{R} = \sum_{i=1}^{q} r_i\underline{n}_i \tag{2-28}$$

Substituting (2-28) into right-hand side of (2-27), we obtain

$$\tau\sum_{i=1}^{q} r_i\underline{z}'\underline{n}_i > 0$$

From the requirement of $N_q'\underline{z}\geq 0$ and $\tau>0$ to gain $f(\underline{x}^1)>$ $f(\underline{x}^*)$, at least one of the r_i must be positive, which contradicts (2-25). Therefore, that $\underline{x}^*$ is the global maximum is proved.

Satisfaction of the Kuhn-Tucker Conditions

The Kuhn-Tucker(K-T) conditions are the necessary, but not sufficient, conditions for a point to be the global maximum of a constrained nonlinear programming problem. Suppose the nonlinear programming problem is in the form

$$\begin{array}{lll} \text{Maximize} & f(\underline{x}) & \\ \text{Subject to} & h_i(\underline{x}) \geq 0 & i=1,\ldots,k \\ & h_i(\underline{x}) = 0 & i=k+1,..,n \end{array} \tag{2-29}$$

and point $\underline{x}^*$ is the maximum. Then the following K-T conditions must be satisfied.

1. $\underline{x}^*$ is feasible

There exist multipliers $\lambda_i \geq 0$, i=1,..,k, and unrestricted multipliers λ_i, i=k+1,...,n, such that

$$2. \quad \lambda_i h_i(\underline{x}^*) = 0 \qquad i=1,\ldots,k \tag{2-30}$$

and

$$3. \quad \underline{g}(\underline{x}^*) + \sum_{i=1}^{m} \lambda_i \underline{\nabla} h_i(\underline{x}^*) = 0$$

To prove the current algorithm satisfies K-T conditions at maximal point, two different cases have to be considered independently. The first case is the interior maximum.

Assume $\underline{x}^*$ is the maximum in the closed region R, then the condition 1 is satisfied naturally. Condition 2 gives $\lambda_i=0$, i=1,...,k, since $h_i(\underline{x}^*)>0$, i=1,...,k. Condition 3 gives

$$\underline{g}(\underline{x}^*) + \sum_{i=1}^{k} \lambda_i \underline{\nabla} h_i(\underline{x}^*) = \underline{g}(\underline{x}^*) \tag{2-31}$$

It is obvious that at interior maximum $\underline{g}(\underline{x}^*)=0$, then (2-31) equal to zero. Thus, the K-T conditions are held.

The second case is the maximum on an intersection of constraints. Assume $\underline{x}^*$ is the maximum on manifold M_q, $\underline{x}^*$ must be a feasible point. To consider the second condition, separate all multipliers λ_i, i=1,...,n into two sets, the first set Z_1 contains multipliers λ_i, i=1,..,q, such that $h_i(\underline{x}^*)=0$. The multipliers in the second set $Z_2=\{\lambda_i, i=q+1,.$

$\ldots,n$, such that $h_i(\underline{x}^*)>0\}$. The multipliers in the former are those corresponding to constraints in the constraint basis.

According to condition 2, the second set $Z_2=\emptyset$, and elements of the first set are greater than or equal to zero, i.e., $\lambda_i \geq 0$, $i=1,\ldots,q$.

Condition 3 gives

$$\underline{g}(\underline{x}^*) + \sum_{i=1}^{n} \lambda_i \underline{\nabla} h_i(\underline{x}^*) = \underline{g}(\underline{x}^*) + \sum_{i=1}^{q} \lambda_i \underline{\nabla} h_i(\underline{x}^*) = 0 \tag{2-32}$$

(2-32) can be rewritten in matrix form

$$\underline{g}(\underline{x}^*) + N_q \underline{\lambda}_q = 0 \tag{2-33}$$

where $\underline{\lambda}_q$ is column vector $(\lambda_1,\lambda_2,\ldots,\lambda_q)$, substituting (2-28) into (2-33), obtain

$$N_q \underline{R} + N_q \underline{\lambda}_q = 0 \tag{2-34}$$

Thus $\underline{\lambda}_q=-\underline{R}$. At constrained global maximum $\underline{R}\leq 0$, therefore $\lambda_i \geq 0$, $i=1,\ldots,q$. We have proved the K-T conditions hold for constrained optimum.

It is very interesting to notice that at a constrained global optimum, the Lagrangian multipliers are equal to r_i but opposite in sign.

<u>Example 1</u>

The following example is a maximization problem with linear constraints. The optimal point is at an intersection of two constraints. The problem is

$$\text{Maximize} \quad f = -(x_1 - 4)^2 - (x_2 - 2)^2$$

$$\text{Subject to} \quad x_1 + x_2 \geq 3$$

$$x_1 + x_2 \leq 1$$

$$x_1 + x_2 \leq 5$$

$$x_1 - x_2 \geq -1$$

The solution is shown as Fig. 3

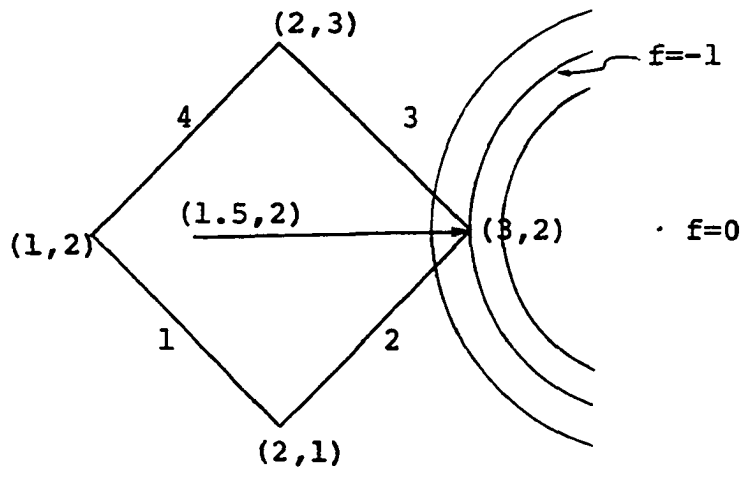


Fig. 3

At the point (1.5,2), $\underline{g}'\underline{p}=5$, therefore the movement in the direction shown in the figure will increase the value of the objective function.

Since constraints 2 and 3 are encountered simultaneously,

the order of constraint basis N_q is immaterial. Point (3,2) is the optimum since $P_q\underline{g}=0$, and vector $\underline{R}=(-1,-1)$. The Lagrangian multipliers of this problem are $\lambda_1=0$, $\lambda_2=1, \lambda_3=1$, $\lambda_4=0$, therefore Kuhn-Tucker conditions are satisfied. The Lagrangian multipliers can be separated into two sets, $Z_2=\emptyset$ and Z_1 containing the multipliers λ_2 and λ_3, $-\underline{\lambda}_q=\underline{R}$ is held at the optimum as proved in the last section.

CHAPTER III

THE NONLINEARLY CONSTRAINED PROBLEM

Linearization of the Nonlinear Programming Problem

The general form of a maximization nonlinear programming problem in m variables, x_i, i=1,....,m, subject to n constraints is

$$\text{Maximize} \quad f(\underline{x}) = f(x_1,....,x_m) \tag{3-1}$$

$$\text{Subject to} \quad h_i(\underline{x}) \geq 0 \qquad i=1,....,k$$

$$h_i(\underline{x}) = 0 \qquad i=k+1,..,n \tag{3-2}$$

where $h_i(\underline{x})$ can be a linear or nonlinear function. In order to fit the algorithm given here, the constraints have to be linearized if they are nonlinear.

A new form of the nonlinear programming problem is obtained, when all nonlinear constraints are linearized and normalized:

$$\text{maximize} \quad f(\underline{x}) = f(x_1,.....,x_m) \tag{3-3}$$

$$\text{subject to} \quad \sum_{j=1}^{m} n_{ij}x_j \geq b_i \qquad i=1,....,k \tag{3-4}$$

$$\sum_{j=1}^{m} n_{ij}x_j = b_i \qquad i=k+1,\ldots,n$$

where

$$\sum_{j=1}^{m} n_{ij}^2 = 1 \qquad i=1,\ldots\ldots n$$

A nonlinear function can be linearized by neglecting the higher order terms of the Taylor's expansion, if the function is differentiable. The calculation of partial derivatives of the function is necessary for the Taylor's expansion. Sometimes the direct evaluation of the partial derivatives is difficult and tedious if functions are complicate or the number of functions is large.

A numerical method suggested by J. H. Christensen and D. M. Clifton (10) is introduced here. In this method, the explicit calculation of partial derivatives is not necessary. Instead, it is completed by introducing a set of auxiliary points of which the number is equal to that of the independent variables in the functions. This method is very available for computer manipulation.

Suppose a set of functions, $f_1, \ldots\ldots f_m$, of n independent variables are given and the linearization of this set of functions at point $\underline{x}$ is to be calculated. If n arbitrary points $\underline{x}^1,\ldots\ldots\underline{x}^n$ are chosen for the set of functions, the Taylor's expansion at point $\underline{x}^o$ gives

$$\begin{aligned}
\underline{f}(\underline{x}^1)-\underline{f}(\underline{x}^o) &\simeq (\partial\underline{f}/\partial x_1)\big|_{\underline{x}^o}(x_1^1-x_1^o)+\ldots+(\partial\underline{f}/\partial x_n)\big|_{\underline{x}^o}(x_n^1-x_n^o)\\
&\vdots\\
\underline{f}(\underline{x}^n)-\underline{f}(x^o) &\simeq (\partial\underline{f}/\partial x_1)\big|_{\underline{x}^o}(x_1^n-x_1^o)+\ldots+(\partial\underline{f}/\partial x_n)\big|_{\underline{x}^o}(x_n^n-x_n^o)
\end{aligned} \tag{3-5}$$

Let $\underline{f}^1=\underline{f}(\underline{x}^1),\ldots,\ \underline{f}^n=\underline{f}(\underline{x}^n)$ and a m x n matrix ΔF be

$$\Delta F = \left[\underline{f}^1-\underline{f}^o,\ldots\ldots\ldots\ldots,\underline{f}^n-\underline{f}^o\right]$$

then the set of Taylor's expansion (5-5) can be rewritten in matrix form

$$\Delta F \simeq J\Delta X \tag{3-6}$$

where

$$J = \left[\partial\underline{f}/\partial x_1,\ldots\ldots\ldots\ldots,\partial\underline{f}/\partial x_n\right]_{\underline{x}^o}$$

is a m x n matrix, and

$$\Delta X = \begin{bmatrix} x_1^1-x_1^o,\ldots\ldots\ldots,x_1^n-x_1^o \\ \\ x_n^1-x_n^o,\ldots\ldots\ldots,x_n^n-x_n^o \end{bmatrix} \tag{3-7}$$

is an n x n matrix.

Equation (3-6) can be rewritten as

$$J = \Delta F(\Delta X)^{-1} \tag{3-8}$$

Let $\Delta X=X-\underline{x}^o\underline{1}$, where X is an n x n matrix, $\underline{1}$ is an 1 x n sum vector and $\underline{x}^o$ is n x 1 column vector, or they can be

expressed in the following mathematical form

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^n \\ \vdots & & \vdots \\ x_n^1 & \cdots & x_n^n \end{bmatrix} \qquad \underline{x}^o = \begin{bmatrix} x_1^o \\ \vdots \\ x_n^o \end{bmatrix}$$

$$\underline{1} = (1,1,\ldots,1)$$

Premultiplying and postmultiplying equation $\Delta X = X - \underline{x}^o \underline{1}$ by X^{-1} and ΔX^{-1}, we obtain

$$\Delta X^{-1} = X^{-1} + X^{-1}\underline{x}^o\underline{1}\Delta X^{-1} \tag{3-9}$$

Premultiplying equation (3-9) by $\underline{1}$

$$\underline{1}\Delta X^{-1} = \underline{1}X^{-1} + \underline{1}X^{-1}\underline{x}^o\underline{1}\Delta X^{-1} \tag{3-10}$$

Let $\alpha = \underline{1}X^{-1}\underline{x}^o$, equation (3-10) becomes

$$\underline{1}\Delta X^{-1} = \underline{1}X^{-1}/(1-\alpha) \tag{3-11}$$

Substituting (3-11) into (3-9), we get

$$\Delta X^{-1} = \left[I + (1/(1-\alpha))X^{-1}\underline{x}^o\underline{1}\right] X^{-1} \tag{3-12}$$

Substituting (3-12) into (3-8), the Jacobian matrix becomes

$$J = \Delta F\left[I + (1/(1-\alpha))X^{-1}\underline{x}^o\underline{1}\right] X^{-1} \tag{3-13}$$

Let

$$\Delta F = F - \underline{f}^o\underline{1} \tag{3-14}$$

where

$$F = \left[f^1, f^2, \ldots\ldots f^n\right]$$

and

$$\underline{f}'^{o} = (f_1^o, \ldots\ldots\ldots, f_m^o)$$

Substituting (3-14) into (3-13) we obtain the final Jacobian matrix

$$J = \left[F + (1/(1-\alpha))(Fy - \underline{f}^o)\underline{1}\right]X^{-1} \qquad (3\text{-}15)$$

The set of linearized functions can be performed after the manipulation of Jacobian matrix has accomplished.

If only one of the auxiliary points is replaced by a new point, the new inverse of matrix X, say $\hat{X}^{-1}$, can be updated by the following formula. The direct calculation of $\hat{X}^{-1}$ is avoided.

$$\begin{aligned} \hat{X}_{ij}^{-1} &= X_{ij}^{-1} - y_i/y_r \cdot X_{rj}^{-1} && j=1,\ldots,n,\ i\neq r \\ \hat{X}_{ij}^{-1} &= X_{ij}^{-1}/y_r && j=1,\ldots,n,\ i=r \end{aligned} \qquad (3\text{-}16)$$

where y_r is the r-th element of relation $\underline{y}=X^{-1}\underline{x}^o$.

This method exploits the linear approximation of functions without direct calculating the partial derivatives of functions. Therefore, it is very useful when the partial derivatives of functions are difficult to evaluate or the number of functions is large. Equation (3-16) for updating matrix $\hat{X}^{-1}$ when one of the auxiliary points is changed is a time-saving procedure. The conventional MAP method used

explicit calculation to obtain partial derivatives. It required very much computation effort at each iteration.

Formulation of Linearized Constraints

The procedure for calculating the Jacobian matrix was given in the last section. The application of the Jacobian matrix to obtain a set of linearized constraints will be discussed here.

Suppose a set of nonlinear constraints

$$\underline{h}(\underline{x}) \geq 0 \tag{3-17}$$

will be linearized at point $\underline{x}^{o}$.

The auxiliary points are selected and Jacobian matrix is evaluated according to equation (3-15). The set of constraints can be written as

$$\underline{h}(\underline{x}) = \underline{h}(\underline{x}^{o}) + J(\underline{x}-\underline{x}^{o}) \geq 0$$

or

$$J\underline{x} \geq J\underline{x}^{o} - \underline{h}(\underline{x}^{o})$$

Then the set of linearized constraints is

$$J\underline{x} - \underline{b} \geq 0 \tag{3-18}$$

Where $\underline{b}=J\underline{x}^{o}-\underline{h}(\underline{x}^{o})$.

Equation (3-18) gives the hint that the coefficients of each linearized constraints are the elements of the corresponding row vector in the Jacobian matrix. This set of

linearized constraints must be normalized to obtain the constraints similar to relations (3-3) and (3-4).

The selection of the set of auxiliary points is very important to the linearization of functions. In general, the auxiliary points should be chosen closed enough to the point at which functions are linearized, and the existence of collinear auxiliary points should be avoided. Sometimes, the random selection of auxiliary points does not give accurate approximation. The auxiliary points which are very far away linearizarion point always give inaccurate approximations. Some trials are always needed to decide the best selection of the auxiliary points.

Step Back Criterion

During the solution of a programming problem with non-linear constraints, it is possible that a feasible point becomes infeasible with respect to the new set of relinearized constraints. In order to obtain a feasible point, it is necessary to step back into the feasible region. There are many acceptable ways to choose a new feasible point, such as, using the first initial point or the point in between the last two optimal points.

The following criterion gives a reasonable way to obtain a new feasible point. The new point will stay on one or more of the violated constraints. Let

$\underline{x}^{i-1}$: The optimal point of the (i-1)-th subproblem.

$\underline{x}^{i}$: The nonlinearly infeasible solution to the i-th linearized subproblem.

$\underline{x}$: The desired feasible point.

$\underline{z}$: Unit direction vector from point $\underline{x}^{i}$ to $\underline{x}^{i-1}$.

At point $\underline{x}$ and point $\underline{x}^{i}$ the constraints can be written as

$$\underline{n}_i'\underline{x} - \underline{b}_i = \lambda_i \tag{3-19}$$

$$\underline{n}_i'\underline{x}^{i} - \underline{b}_i = \lambda_i^i \tag{3-20}$$

Substracting equation (3-20) from equation (3-19) we get

$$\underline{n}_i'(\underline{x} - \underline{x}^{i}) = \lambda_i - \lambda_i^i \tag{3-21}$$

Since $\underline{x}$ is chosen on the line connecting point $\underline{x}^{i-1}$ and $\underline{x}^{i}$ therefore, $\underline{x}=\underline{x}^{i}+\tau\underline{z}$. Equation (3-21) becomes

$$\tau\underline{n}_i'\underline{z} = \lambda_i - \lambda_i^i$$

Because the new feasible point will stay on at least one of the violated constraints, λ_i must be equal to zero. The permissive step length from point $\underline{x}^{i}$ toward point $\underline{x}^{i-1}$ is

$$\tau = -\lambda_i^i/\underline{n}_i'\underline{z} > 0$$

Where $\lambda_i^i<0$, since $\underline{x}^{i}$is the infeasible point with respect to constraint i. In order to maintain τ be positive, only the constraints with $\underline{n}_i'\underline{z}>0$ are considered. The maximum required

step length will be

$$\tau_{max} = \min |\lambda_i^i / \underline{n}_i' \underline{z}| > 0 \qquad \underline{n}_i' \underline{z} > 0 \tag{3-22}$$

Via the procedure of deriving the criterion it is obvious that the new point will be feasible and stay on one or more constraints which are violated by the optimal point of previous linear constraint subproblem. The new feasible point is a good starting point for the next iteration. The constraints on which the new feasible point stays will come in to constraint basis immediately after the iteration begins, and the criteria in the chapter II will continue the process of solving the problems.

The following example shows the application of this criterion.

<u>Example 2</u>

$$\begin{aligned} &\text{Maximize} && f = 2x_1 + x_2 \\ &\text{Subject to} && x_1^2 + x_2^2 \le 25 \\ & && x_1^2 - x_2^2 \le 7 \end{aligned} \tag{3-23}$$

This example is a maximization problem with optimum located at the intersection of two constraints. The movement of the point is given in Table II and also shown in Fig. 7.

Table II showed that the moving point stayed in the feasible region and the value of the objective function increased monotonically until point (4.159,3) is reached.

This point is feasible with respect to the current set of linearized constraints, but infeasible relative to the relinearized constraints. The step back procedure drew the point back to point (3.987, 3.001), which is at the intersection of constraints and also the optimal point of the new subproblem. The infeasibility occured again when constraints were linearized at point (4.003, 3). The new point after step back procedure is (4, 3) which is the global optimum of the problem.

Algorithm and Procedure of Calculation

The proposed method in this dissertation satisfies the Kuhn-Tucker conditions at the optimal point and the necessary conditions for increasing the value of the objective function during the process of solving problems. These satisfactions had been proved in the last chapter. In this section the algorithm is given for the solution of nonlinear programming problems with convex nonlinear constraints. The entire algorithm should be considered to consist of two major parts: the routine for solving linearly constrained subproblems and the manipulation of the linearization and step back processes. Some minor modifications should be made, such as the deletion of the procedure of linearization of constraints and step back of infeasible point, if the given problem is linearly constrained.

The description of the algorithm is

1. Linearize the constraints at point $\underline{x}$.
2. Normalize the constraints.
3. If $\underline{x}$ is feasible with respect to the set of normalized constraints proceed to step 5. Otherwise, go to step 4.
4. Step back into feasible region, go to next step.
5. If $\underline{x}$ lies on manifold M_q, build up N_q and P_q. Otherwise, set $P_q=I$.
6. Compute $P_q\underline{g}$ and $\underline{R}$ according to equation (2-22). If $P_q\underline{g}=0$ and $\underline{R}\leq 0$, $\underline{x}$ is a stationary point of the subproblem. Set $\underline{x}^i=\underline{x}$, proceed to step 7. Otherwise, go to step 8.
7. Compare current stationary point $\underline{x}^i$ with last stationary point $\underline{x}^{i-1}$. If $|\underline{x}-\underline{x}^{i-1}|\leq\varepsilon$, where ε is the tolerance, the global optimum is reached. If not, return to step 1.
8. If $|P_q\underline{g}|\leq\frac{1}{2}r_q b_{qq}^{-\frac{1}{2}}$, where $r_q b_{qq}^{-\frac{1}{2}}\geq r_i b_{ii}^{-\frac{1}{2}}$, $i=1,\ldots.q-1$, and where b_{ii} is the i-th diagonal element of $(N_q N_q)-1$, drop the q hyperplane from the constraint basis and obtain N_{q-1} and P_{q-1} by the recursion formula (2-19), evaluate unit direction vector $\underline{z}=P_{q-1}\underline{p}/|P_{q-1}\underline{p}|$. Then go to step 10.
9. If the case in step 8 does not occur, compute unit direction vector $\underline{z}=P_q\underline{p}/|P_q\underline{p}|$.
10. Evaluate the maximal step length τ_m according relation (2-17) in the direction of $\underline{z}$.
11. Obtain y, $0\leq y\leq\tau_m$, which maximize the objective function in the direction $\underline{z}$.
12. Set $\underline{x}=\underline{x}+y\underline{z}$.

13. If $y=\tau_m$, add hyperplane H_q to the constrain basis and form new N_{q+1}, P_{q+1} by recursion formula (2-20), then gc to step 14.
14. Find new gradient $\underline{g}(\underline{x})$ and direction $\underline{p}$.
15. Return to step 6.

The procedure of the algorithm is also described by the flow chart shown in figure 4.

TABLE II

THE MOVEMENT OF THE POINT OF EXAMPLE 2

Point	Constraint Basis	Violated Constraints	Objective Function
(1.000,1.000)	0	0	3.000
(3.000,3.000)	0	0	9.000
(4.159,3.000)	1,2	1,2	11.318
(3.978,3.001)	1,2	0	10.957
(4.003,3.000)	1,2	1,2	11.012
(4.000,3.000)	1,2	0	11.000

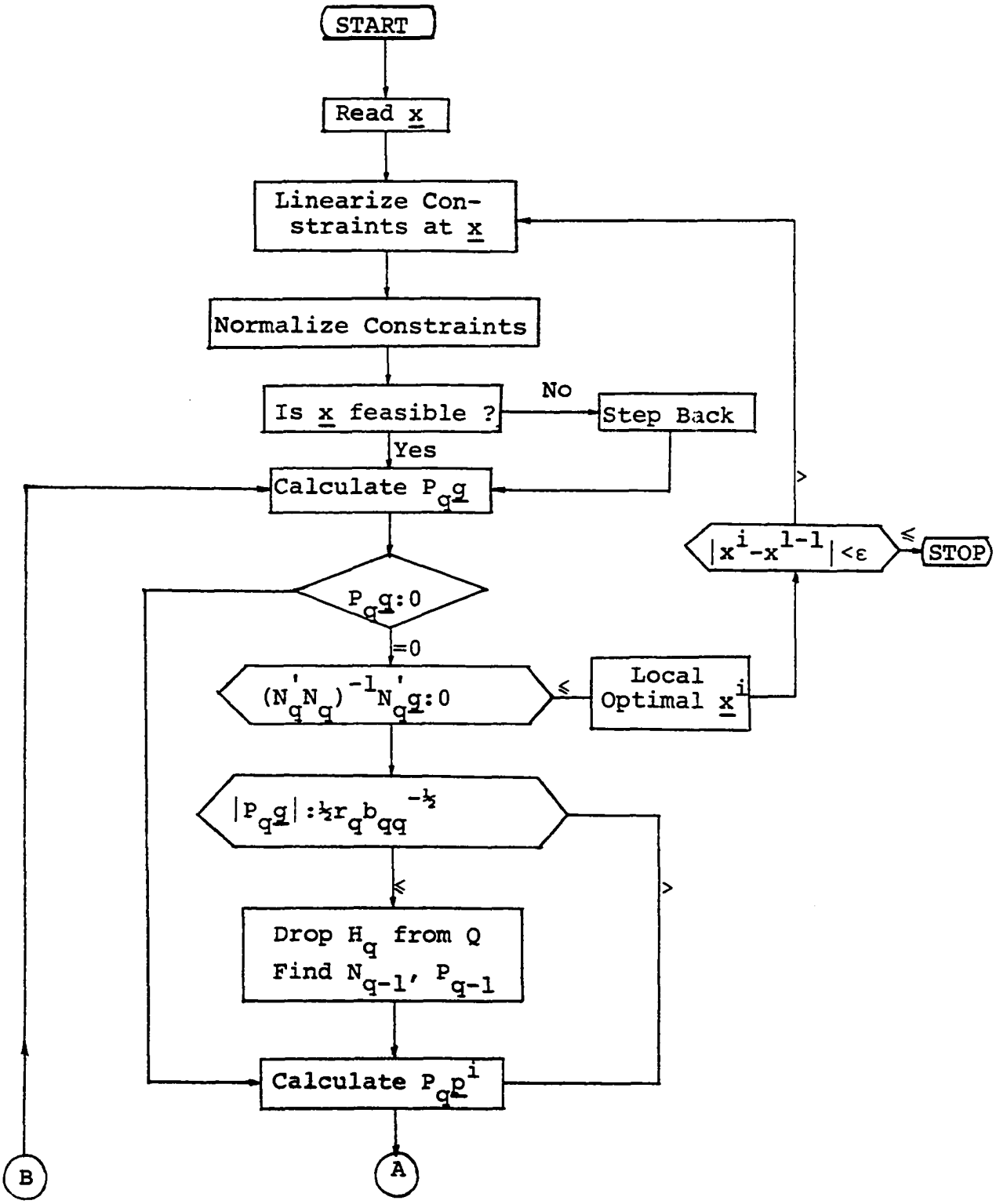


Fig. 4(a) Flow Chart of Revised Algorithm

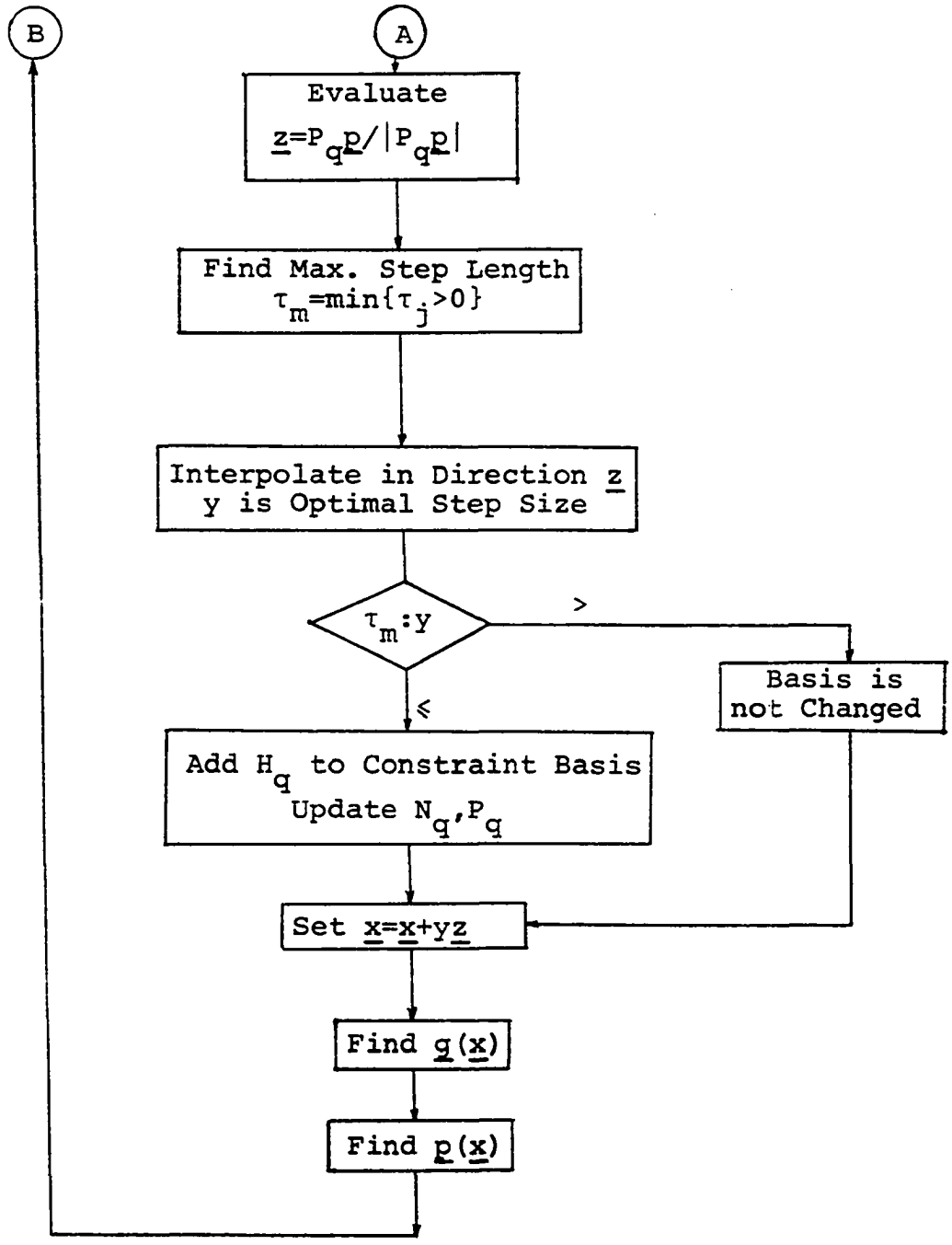


Fig. 4(b) Flow Chart (Cont'd)

CHAPTER IV

NUMERICAL RESULTS AND CONCLUSIONS

Problem of Cubic Objective Function

The first example is a minimization of a convex cubic function of five variables subject to 15 linear inequality constraints. This problem had been solved by Goldfarb and Lapidus (25) using the conjugate gradient method.

$$\text{Minimize} \quad \sum_{j=1}^{5} e_j x_j + \sum_{j=1}^{5} \sum_{i=1}^{5} c_{ij} x_i x_j + \sum_{j=1}^{5} d_j x_j^3 \tag{4-1}$$

$$\text{Subject to} \quad \sum_{j=1}^{5} a_{ij} x_j \geq b_i \qquad i=1,\ldots.10$$

$$x_j \geq 0 \tag{4-2}$$

where the coefficients e_j, c_{ij}, d_j and b_i are given in Table III.

Starting from an initial feasible (0,0,0,0,1), the same starting point for the conjugate gradient method and the gradient projection method, the revised gradient projection method took seven steps to reach the constrained minimum and

had four hyperplanes in the constraint basis. This is the same as the other two methods. The execution time for solving this problem on an IBM 1130 was 4.93 seconds. A comparison of execution time is not significant since the computers are not the same model, but the comparison of the number of steps taken for each method showed the revised gradient projection method is the best. Other advantages of the revised gradient projection method are the fewer matrix manipulations and less computer storage than those required for the conjugate gradient method. The step by step progress of the three methods for this problem is presented in Table IV.

Comparison with Improved MAP Method

Three different characteristic problems, which have the optimal point located in the internal regions of constraints, at an intersection of constraints and on a single constraint are presented in this section. The solutions of the problems at each iteration are shown on Figs 5, 6, and 8. These three problems were also solved by Yang's improved MAP method (60) and the solutions are given in Figs 5, 7, and 9. In the figures, the solid lines represent the moving points obtained by the revised gradient projection method, and the broken lines represent those obtained by the improved MAP method. The first problem is an interior optimum. The current method took only one iteration to reach the optimal

point, it is much more efficient than the improved MAP method which took 12 iterations to obtain the optimum.

When both methods are used to solve problems, a step size restriction on the movement of the variables is required. The purpose of the additional restriction is to keep the solution closed to the feasible region, since the linearized constraints might be very far away from the real positions. This has been shown in Yang's work. If the restriction is not added, the solution will be away from the feasible region. From another aspect, since the improved MAP method solved the subproblems of linear programming, the gradient direction is fixed at each iteration. The restriction can not be too large, otherwise, it becomes difficult to reach the optimal point. This consideration is not necessary for the revised gradient projection method. Therefore, the restriction for MAP method is smaller than that for the current method. It is obvious that the former method takes more iterations to obtain the optimum than does the latter.

The second problem is a linear objective function with circular constraints, and the optimum located at an intersection of the constraints. It took 8 iterations to reach the optimum for the improved MAP method and 4 iterations for the revised gradient projection method. The third problem is an elliptical objective function with optimum located on a constraint. In this problem the iteration number of Yang's work is 7 and 6 for the current method.

The linearized constraints do not converge very fast, thus it took almost the same number of iterations to get the optimum.

All these examples show that the revised gradient projection method is much more efficient than the improved MAP method in handle nonlinear problems.

The Chemical Process Problem

The mathematical description and system model of this chemical process are given in chapter one. Dibella and Stevens solved this problem by conventional MAP method, in which the simplex technique was used to solve the subproblem. There were 45 variables and 21 constraints when Dibella and Stevens solved the original problem. It needed very much running time and computer storage for this cumbersome problem.

J. H. Christensen (9) simplified the problem by introducing 5 additional equations and variables. He then used the structure of the system of equations to convert the original problem into one of 4 variables and 7 constraints. This simplification not only saved tremendous computer storage and computer execution time but also made it easier to carry out the procedure of optimization.

The objective function and constraints are rewritten in terms of four variables F_{RE}, F_{RC}, $\emptyset$, and T.

$$\text{Maximize} \quad f = (368F_P + 8.4F_D - 28F_A - 42F_B - 14F_G -$$

$$0.37F_R)/V\rho - 10 \qquad (4\text{-}3)$$

Subject to

$$F_{RE} \geq 0$$
$$\emptyset \geq 0$$
$$1 - \emptyset \geq 0$$
$$R_3 \geq 0$$
$$F_{RC} \geq 0$$
$$T - 580 \geq 0$$
$$680 - T \geq 0$$

where

$$F_{RP} = F_P + 0.1\ F_{RE}$$
$$R_2 = (M_B/M_E)F_{RE}\emptyset$$
$$R_3 = (M_C/M_P)(R_2 - F_P - F_{RP}\emptyset + F_P\emptyset)$$
$$R_1 = (M_B/M_C)(R_3 + F_{RC}\emptyset) + R_2$$
$$t = R_3/(k_3F_{RP}F_{RC})$$
$$F_{RB} = R_2/(k_2F_{RC}t)$$
$$F_B = R_1 + R_2 + F_{RB}\emptyset$$
$$F_{RA} = R_1/(k_1F_{RB}t)$$
$$F_G = (M_G/M_C)R_3$$
$$F_R = F_{RA} + F_{RB} + F_{RC} + F_{RE} + F_{RP} + F_G$$
$$F_D = \emptyset(F_R - F_G - F_P)$$
$$F_A = R_1 + F_{RA}\emptyset$$
$$V = F_R^2t/\rho$$

Dibella and Stevens (17) solved the problem on IBM 709 computer. They stopped searching at 600 iterations and supposed it was the optimal solution of the problem. Yang

found that the value of the objective function he obtained after 140 iterations on IBM 1130 computer was much better than Dibella's. The reason Dibella and Stevens gave up when the value of the objective function reached 72.5%, might be that they thought the further searching is not significant, since their problem contained too many variables which complicated the problem and converged very slowly. Yang also found that the objective function has the tendency to increase without limit. He obtained 99.25% for the objective function at the 140-th iteration and still could improve it much better than this. The unlimited increase of the objective function is not realistic, because the percent return will not be higher than 50% in general. It is then obvious that Dibella's original problem should be modified. Yang suggested an additional constraint to restrict the irrational increase of the objective function. The new constraint restricts the flow rate of the effluent from the reactor can not exceed thirty times the flow rate of product P.

$$h_8 = 30F_P - F_R \geq 0$$

The original problem became one of 4 variables with 8 inequality constraints.

The optimal solutions of the temperature in the reactor, volume of the reactor and flow rate of feed and recycle were obtained by the revised gradient projection method and given in Table V. The percent return evaluated by Yang and the

current method were 65.098 and 64.9635. The slight difference might be due to the linearized constraints, which were not exactly the same in two methods, and the sensibility of the variables to the objective function. The variation of the objective function with the number of iterations were given in Fig. 10. The solid line represents the value of objective function obtained by the revised gradient projection method and the broken line is for improved MAP technique. It is remarkable that the former method took only 4 linearizations to reach the optimum while the latter needed 36 linearizations. The value of the objective function increased very fast for the revised gradient projection method, but not for the improved MAP method after the 25-th iteration. This fact shows the superiority of the current method to the improved MAP technique.

Alkylation Process Problem

This problem represents as alkylation process, which consists of a reactor and a fractionator. Sauer, Coville and Burwick (51) had described this process. Westerberg and Debrosse (56) solved this problem by using their algorithm developed for nonlinear programming problems.

The mathematical model of this process problem is

$$\text{Minimize} \quad f = -0.063x_4x_7 + 5.04x_1 + 0.035x_2 + 10x_3 + 3.36x_5 \tag{4-4}$$

$$\text{Subject to} \quad x_1 \geq 0 \qquad x_1 \leq 2000$$

$$
\begin{aligned}
&x_2 \geq 0 && x_2 \leq 16000 \\
&x_3 \geq 0 && x_3 \leq 120 \\
&x_4 \geq 0 && x_4 \leq 5000 \\
&x_5 \geq 0 && x_5 \leq 2000 \\
&x_6 \geq 85 && x_6 \leq 93 \\
&x_7 \geq 90 && x_7 \leq 95 \\
&x_8 \geq 3 && x_8 \leq 12 \\
&x_9 \geq 1.2 && x_9 \leq 4 \\
&x_{10} \geq 145 && x_{10} \leq 162 \\
&x_1(1.12+0.13167x_8-0.00667x_8^2)-0.99x_4 \geq 0 \\
&x_1(1.12+0.13167x_8-0.00667x_8^2)-1.01x_4 \leq 0 \\
&86.35+1.098x_8-0.038x_8^2+0.325(x_6-89)-0.99x_7 \geq 0 \\
&86.35+1.098x_8-0.038x_8^2+0.325(x_6-89)-1.01x_7 \leq 0 \\
&35.82-0.222x_{10}-0.99x_9 \geq 0 \\
&35.82-0.222x_{10}-1.01x_9 \leq 0 \\
&-133+3x_7-0.99x_{10} \geq 0 \\
&-133+3x_7-1.01x_{10} \leq 0 \\
&(x_2+x_5)/x_1 - x_8=0 \\
&98000x_3/(x_4x_9+1000x_3) - x_6 = 0 \\
&1.22x_4-x_1-x_5 = 0
\end{aligned}
\tag{4-7}
$$

This is a nonlinear programming problem of 10 variables with 31 nonlinear constraints. The last three equality constraints can be used to solve for 3 independent variables. After some mathema[illegible] manipulations, the original problem can be simplified to a new one of 7 variables with 28

inequality constraints.

Starting at a slightly different point from Westerberg and Debrosse's, the problem was solved by using the revised gradient projection method. The value of the objective function increased very fast from 700.562 to 1310.238 in two iterations. It is surprising to find that infeasibility occurred and the objective function oscillated in the following iterations. The global optimum could not be reached, at which the value of the objective function is 1714.93 obtained by Westerberg and Debrosse. A reasonable explanation for the failure of the solution of this problem is the nonconvexity of the region formed by the constraints. Suppose an optimal point of the subproblem was obtained at n-th iteration, the movement of the point in the next iteration is in the direction of increasing the value of objective function, but the point in the (n+1)-th iteration may be in the infeasible region of the actual constraints if the nonconvexity exists. The final optimal point of the (n+1)-th subproblem might be infeasible to the real constraints, although it is feasible to the current linearized constraints. Therefore, the infeasibility will occur repeatedly, since the region is nonconvex. The global optimum can not be reached no matter how much computer time is used. The point obtained at each iteration is a local optimum.

The local optimum due to nonconvex constraints is shown in Fig. 11, in which the global optimum is at point b. The

solution stops at point a, since the point moves in the direction of increasing objective function will be in the infeasible region.

Conclusion

The proposed revised gradient projection method is an efficient method for nonlinear programming problems with convex linear constraints. It is more efficient than the original gradient projection method, since the direction of the movement of the point in the proposed method is better than the steep ascent(or descent) direction near the optimum.

The advantages of the proposed method over Goldfarb's conjugate gradient method are that it requires fewer matrix manipulations and less computer storage. Since the proposed method is at least as efficient as Goldfarb's, or even slightly better, and requires fewer matrix manipulations, the computer time for the former should be shorter than that of the latter.

The numerical approximation of the Jacobian matrix is used to handle the nonlinear constraints problems in this method. The tedious and time-consuming task of deriving the first partial derivatives is avoided.

The proposed method is much more efficient than Yang's improved MAP method in the treatment of both linearly and nonlinearly constrained problems. Solving the nonlinearly constrained problems the improved MAP technique requires a

smaller maximum step size which requires more iterations than the present method to reach the optimum. The example problems in this chapter show the superiority of the proposed method.

This revised gradient projection method has many advantages over Rosen's gradient projection, Goldfarb's conjugate gradient and Yang's improved MAP methods. It can be applied to solve nonlinear programming problems with convex linear or nonlinear constraints, but special problems with nonconvex constraints can not be solved by this method as shown in the last example.

TABLE III

DATA FOR FIVE VARIABLES CUBIC FUNCTION

	i	1	2	3	4	5	
	1	30	-20	-10	32	-10	
	2	-20	39	-6	-31	32	
c_{ij}	3	-10	-6	10	-6	-10	
	4	32	-31	6	39	-20	
	5	-10	32	-10	-20	30	
d_j		4	8	10	6	2	
e_j		-15	-27	-36	-18	-12	
							b_i
a_{ij}	1	-16	2	0	1	0	-40
	2	0	-2	0	0.4	2	-2
	3	-3.5	0	2	0	0	-0.25
	4	0	-2	0	-4	-1	-4
	5	0	-9	-2	1	-2.8	-4
	6	2	0	-4	0	0	-1
	7	-1	-1	-1	-1	-1	-40
	8	-1	-2	-3	-2	-1	-60
	9	1	2	3	4	5	5
	10	1	1	1	1	1	1

TABLE IV

MINIMIZATION OF CUBIC FUNCTION OF FIVE VARIABLES

	Revised Gradient Projection		Conjugate Gradient		Gradient Projection	
Step	$-f(\underline{x})$	No. of Constraints in Basis	$-f(\underline{x})$	No. of Constraints in Basis	$-f(\underline{x})$	No. of Constraints in Basis
0	-20	0	-20	0	-20	0
1	23.8967	1	23.8967	1	23.8967	1
2	24.8052	2	25.1972	2	25.1972	2
3	25.2201	3	25.2605	3	25.2605	3
4	30.4768	2	28.5235	2	25.5748	2
5	31.5119	3	29.6326	3	31.4719	3
6	31.97897	3	32.0165	4	32.1252	3
7	32.34870	4	32.1134	3	32.2955	3
8			32.3353	3	32.34865	3
9			32.34868	4	32.34867	4
10					32.34870	4
11					32.34870	4

Fig. 5 Interior Optimum

Minimize $(x_1 - 2)^2 + (x_2 - 3)^2$

Subject to $25 - x_1^2 - x_2^2 \geq 0$

$7 - x_1^2 + x_2^2 \geq 0$

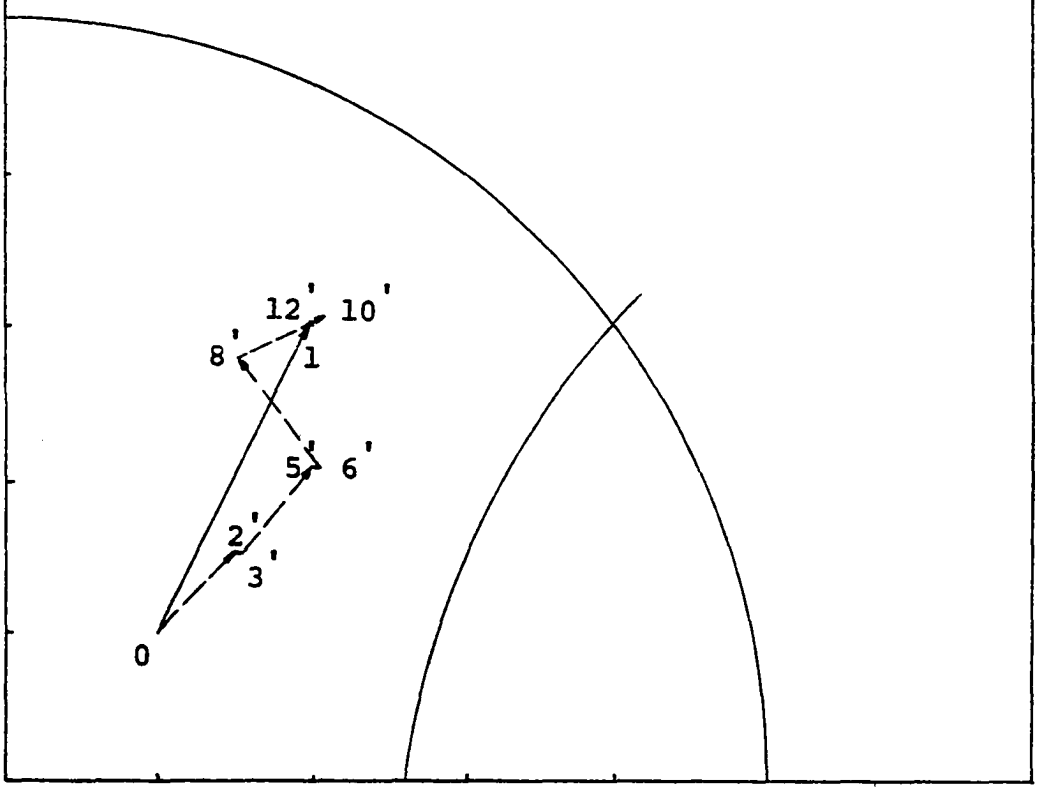


Fig. 6 Optimum at an intersection (Lee's)

Maximize $2x_1 + x_2$

Subject to $25 - x_1^2 - x_2^2 \geq 0$

$7 - x_1^2 + x_2^2 \geq 0$

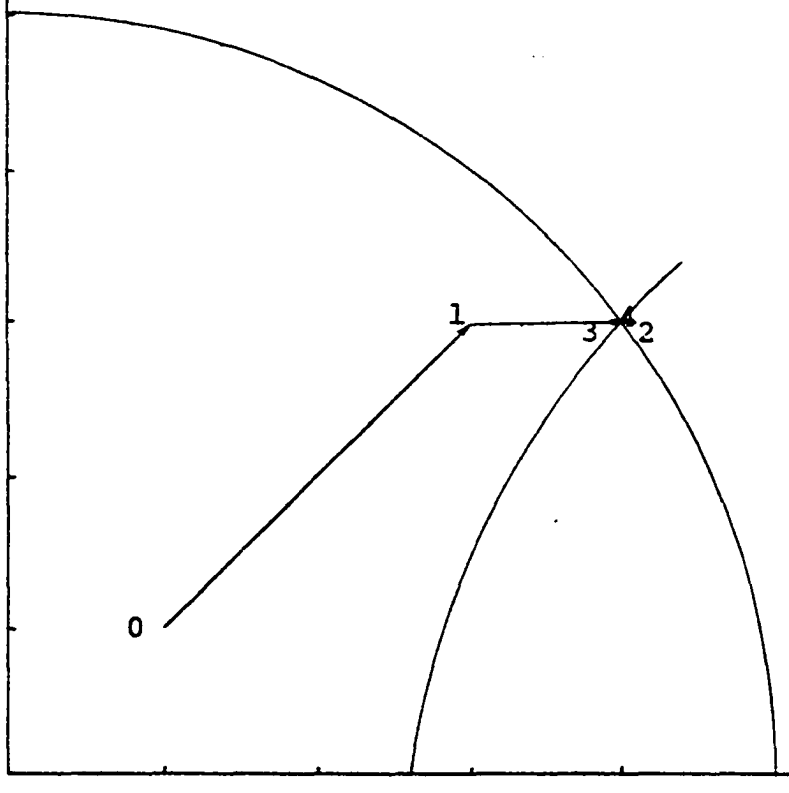


Fig. 7 Optimum at an Intersection (Yang's)

Maximize $2x_1 + x_2$

Subject to $25 - x_1^2 - x_2^2 \geq 0$

$7 - x_1^2 + x_2^2 \geq 0$

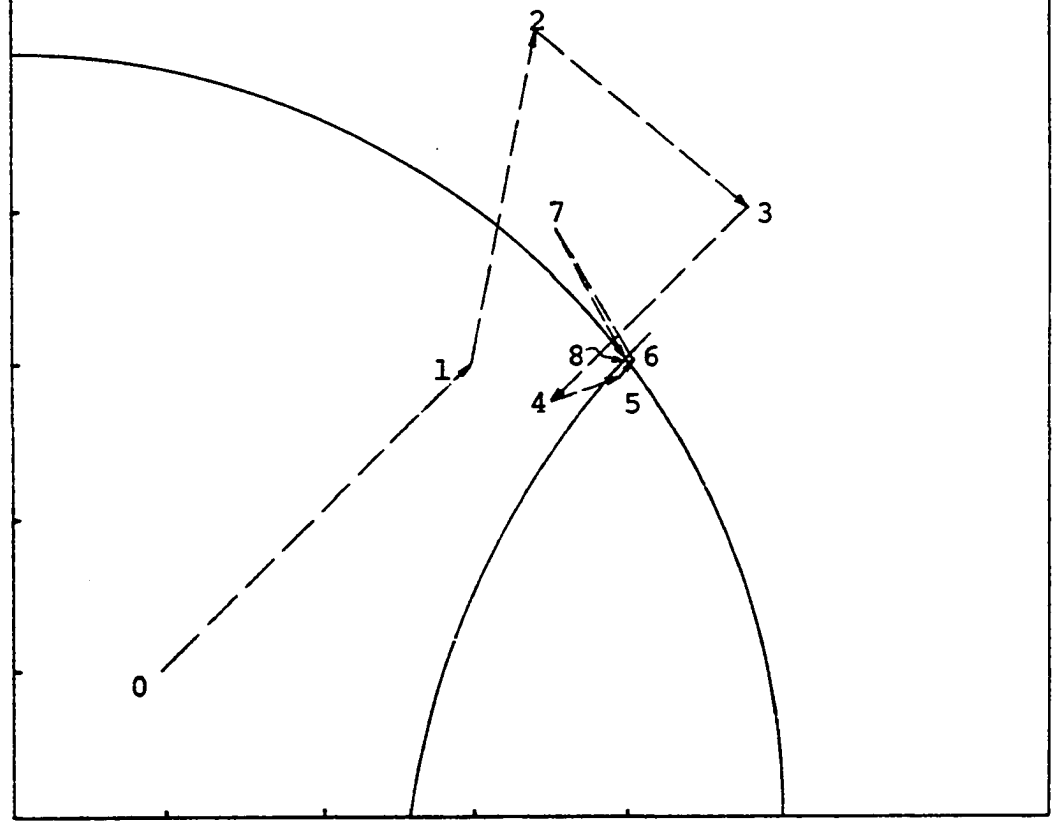


Fig. 8 Optimum on a Constraint (Lee's)

Maximize $4x_1^2 + 9x_2^2$

Subject to $25 - x_1^2 - x_2^2 \geq 0$

$7 - x_1^2 + x_2^2 \geq 0$

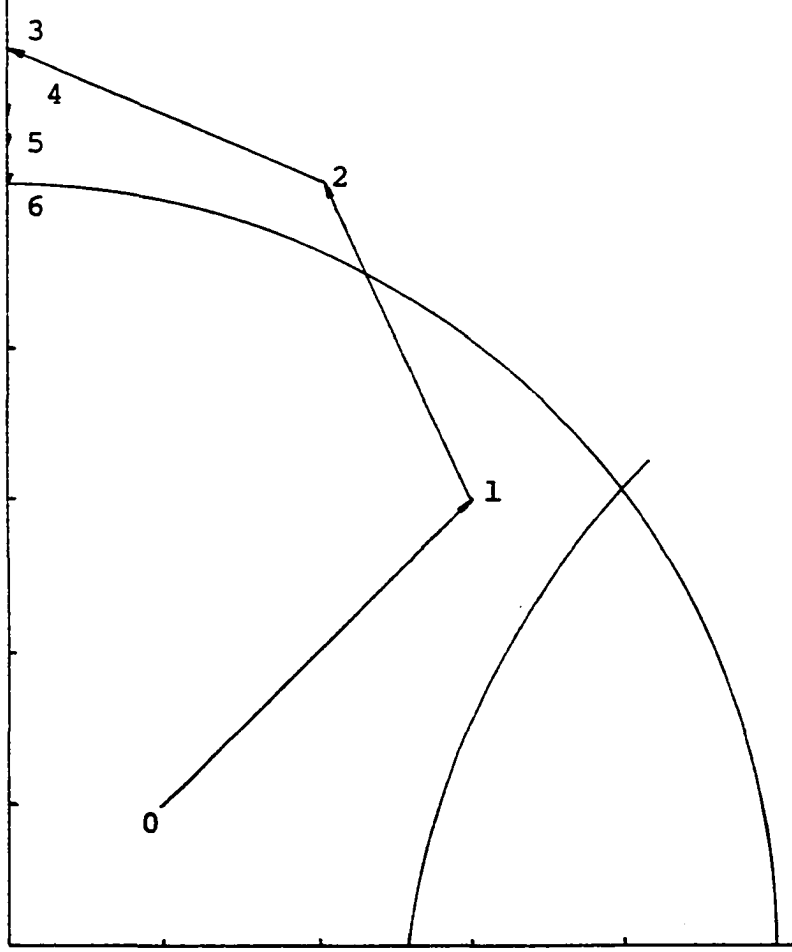


Fig. 9 Optimum on a Constraint (Yang's)

Maximize $4x_1^2 + 9x_2^2$

Subject to $25 - x_1^2 - x_2^2 \geq 0$

$7 - x_1^2 - x_2^2 \geq 0$

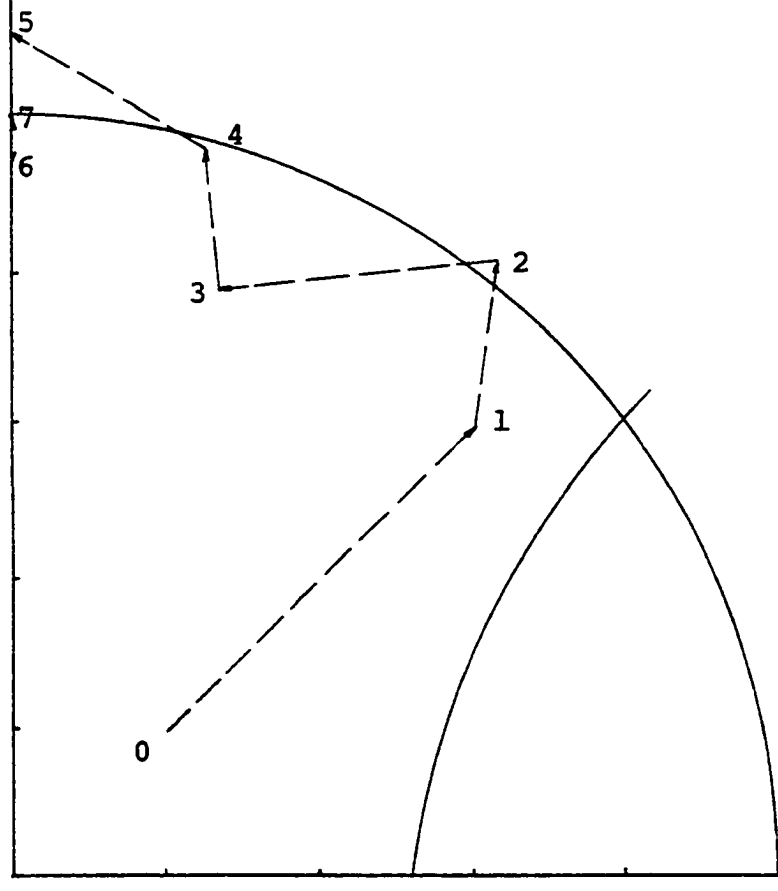


TABLE V

COMPARISON OF THE OPTIMAL SOLUTION OBTAINED BY PRESENT METHOD AND YANG'S IMPROVED MAP METHOD

Variable	Initial Guess	Optimal Solution Yang's	Optimal Solution Lee's	Unit
F_{RE}	43119.5	52972.465	52884.710	lb/hr
Ø	0.333	0.278	0.278	
F_{RC}	3120.00	3258.539	3282.963	lb/hr
T	644.75	650.667	650.137	^{o}R
F_{RP}	9074.95	10060.25	10051.47	lb/hr
R_2	7179.40	7360.524	7350.97	lb/hr
R_3	1961.04	2250.84	2235.54	lb/hr
R_1	8679.40	8938.72	8925.07	lb/hr
K_1	49.33	58.46	58.90	hr^{-1}
K_2	204.29	252.60	246.70	hr^{-1}
K_3	324.65	430.87	423.10	hr^{-1}
F_{RB}	52798.89	56115.43	56288.232	lb/hr
F_B	33440.82	31893.72	31896.24	lb/hr
F_{RA}	15620.67	17099.05	16677.43	lb/hr
F_G	2941.55	3376.26	3353.31	lb/hr
F_R	126675.53	142881.98	142538.11	lb/hr
F_D	39617.34	37445.01	37356.294	lb/hr
F_A	13881.08	13690.54	13558.06	lb/hr
Return	49.69	65.098	64.967	%
h_1	1961.03	2250.83	467.00	
h_2	16214.46	0.148	0.002	
Iteration Number	0	36	4	

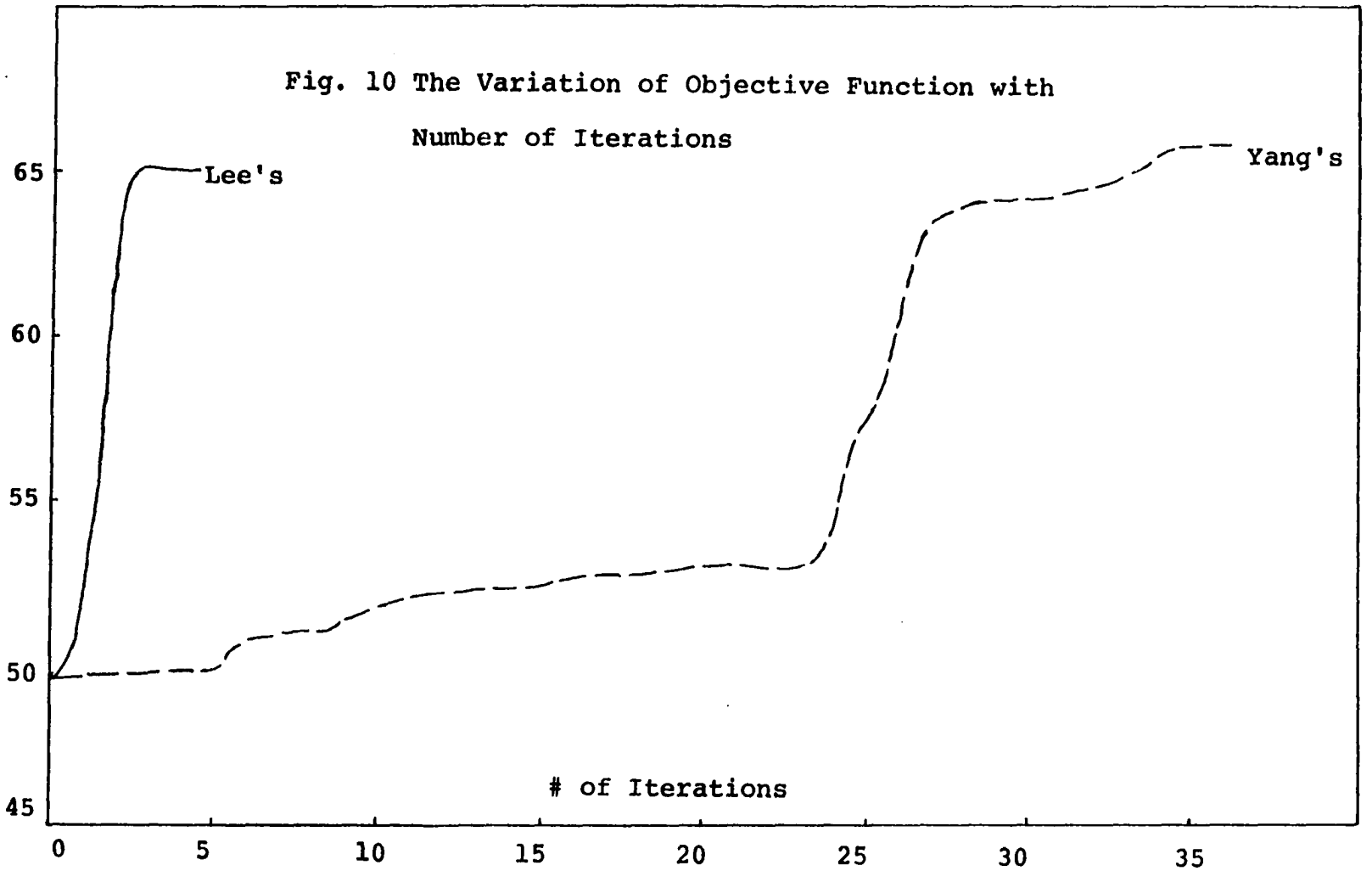
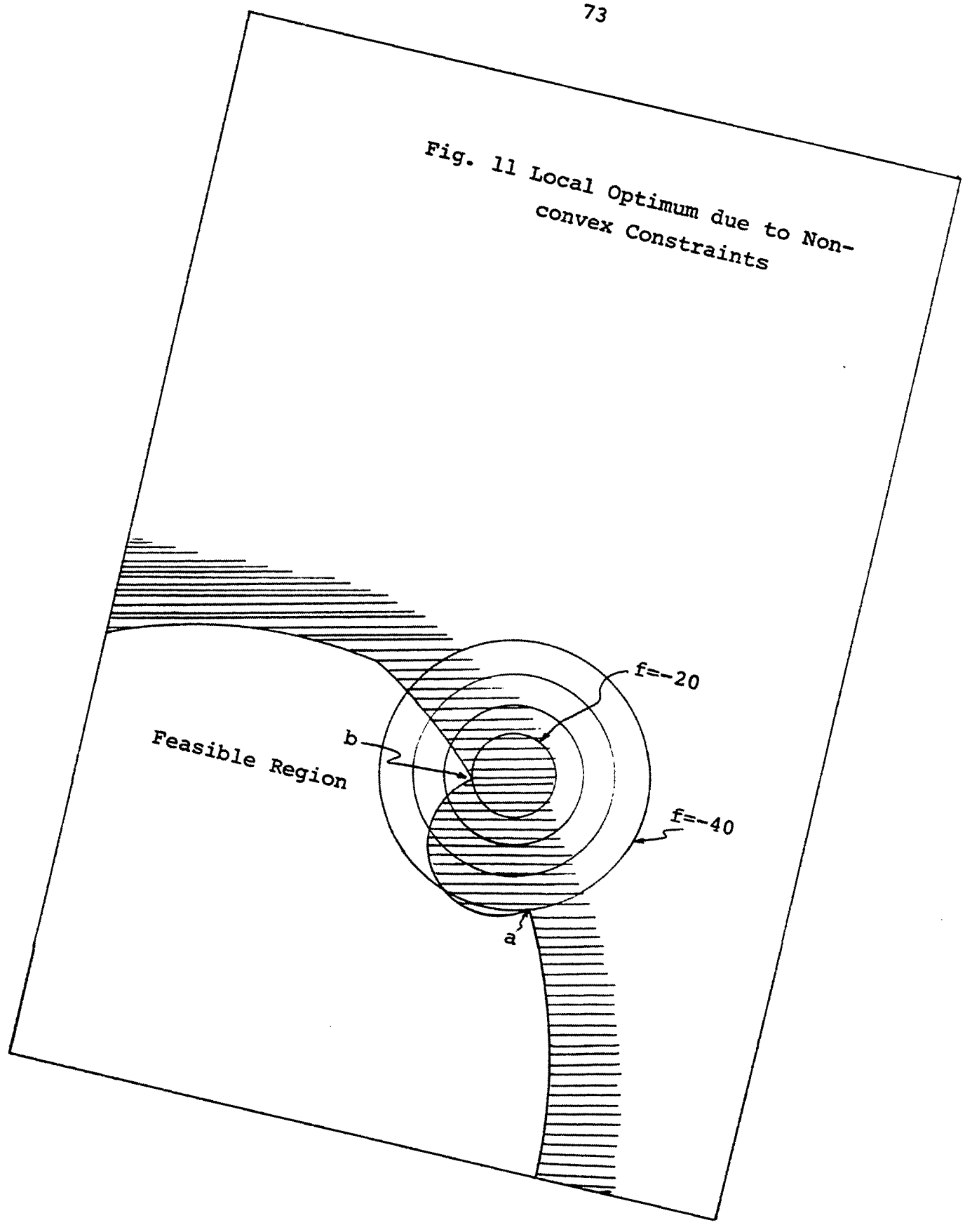


Fig. 10 The Variation of Objective Function with Number of Iterations

Fig. 11 Local Optimum due to Non-convex Constraints



LIST OF REFERENCES

1. Abadie J., and J. Carpentier, in chapter 8, Integer and Nonlinear Programming, edited by J. Abadie, North-Holland, Amsterdam, (1970).

2. Beale, E. M. L., Naval Res. Logist. Quart. 6, 227 (1959).

3. Beveridge, G. S. G., and R. S. Schechter, Optimization: Theory and Practice, McGraw-Hill, Inc., N. Y., (1970).

4. Box, N. J., Computer J., 9, 67 (1966).

5. Box, M. J., Computer J., 8, 42 (1965).

6. Broyden, C. G., Math. of Computation, 21, 368 (1967).

7. Carroll, C. W., Operation Res., 9, 169 (1961).

8. Cheney, E. W., and A. A. Goldstein, Numer. Math., 4, 253 (1959).

9. Christensen , J. H., Doctoral Dissertation, University of Wisconsin, (1967).

10. Clifton, D. M., General Examination, University of Oklahoma, Norman, Okla., (1969).

11. Cragg, E. E., and A. V. Levy, Rice Univ. Aero-Astronautic Rept., 58, Houston, Texas, (1969).

12. Dantzig, G. B., RAND Rept., p-1486, Santa Monica, (1959).

13. Dantzig, G. B., Linear Programming and Extension, Princeton Univ. Press, Princeton, N. J., (1962).

14. Dantzig, G. B., and W. Orchard-Hays, RM-1268, The RAND Corp., (1953).

15. Davidon, W. C., Atomic Energy Commission Res. Devel. Rept., ANL-5990, (1959).

16. Debrosse, C. J., and A. W. Westerberg, AIChE J., 19, 251 (1973).

17. Dibella, C. W., and W. F. Stevens, I&EC Pros. Design and Devel., 4, 17 (1965).

18. Fiacco, A. V., and G. P. McCormick, Management Sci., 10, 360 (1964).

19. Fiacco, A. V., and G. P. McCormick, Management Sci., 10, 601 (1964).

20. Fletcher, R., Computer J., 13, 317 (1970).

21. Fletcher, R., and M. J. D. Powell, Computer J., 6, 163 (1963).

22. Fletcher, R., and C. M. Reeves, Computer J., 7, 149 (1964).

23. Frank, M., and P. Wolfe, Naval Res. Logist. Quart., 3, 95 (1956).

24. Frish R., Memorandumn, Universetets Socialokonomisk Institutt, Oslo, (1958).

25. Goldfarb, D., and L. Lapidus, I&EC Fundamentals, 7, 142 (1968).

26. Greenstadt, J., Math. of Computation, 21, 360 (1967).

27. Greenstadt, J., Meth. of Computation, 24, 1 (1970).

28. Griffith, R. E., and R. A. Stewart, Management Sci., 7, 379 (1961).

29. Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley Pub. Co., Mass., (1964).

30. Hestenes, M. R., and E. Stiefel, J. Res. Nat. Burea of Standards, 49, 409 (1952).

31. Himmelblau, D. M., Applied Nonlinear Programming, McGraw-Hill, Inc., New York, (1972).

32. Hooke, R., and T. A. Jeeves, J. Asso. Computer Machinery, 8, 212 (1961).

33. Kelley, J. E. Jr., J. Soc. Indust. Appl. Math., 8, 703 (1960).

34. Kowalik, J., Osborne, M. R., and D. M. Ryan, Operation Res., 71, 973 (1968).

35. Kuhn H. W., and A. W. Tucker, Proceedings of second Berkeley Symp. on Math. Statistics and Prob., Univ. of California Press, Berkeley, 481 (1951).

36. Lemke, C. E., J. Soc. Indust. Appl. Math., 9, 1 (1961).

37. Marquardt, W., J. Soc. Indust. Appl. Math., 11, 431 (1963).

38. Miele, A., and J. W. Cantrell, Rice Univ. Aero-Astronautic Rept., 56, Houston, Texas, (1969).

39. Miller, C. E., in Recent Advances in Mathematical Programming, p.89 , edited by R. L. Groves and P. Wolfe, McGraw-Hill, Inc., N.Y. (1963).

40. Morrison, D. D., J. Soc. Indust. Appl. Math., 5, 83 (1968).

41. Martagh, B. A. and R. W. H. Sargent, Paper presented at Inst. of Math. and Its Applications and Br. Computer Soc. Jt. Conf. on Optimization, Univ. Keele, (1968).

42. Nelder, J. A., and R. Mead, Computer J., 7, 308 (1965).

43. Passut, C. A., and R. P. Danner, I&EC Proc. Des. Develop. 11, 543 (1972).

44. Paviani, D. A., and D. M. Himmelblau, Operation Res., 17, 872 (1969).

45. Pearson, J. D., Computer J., 12, 171 (1969).

46. Powell, M. J. D., Computer J., 5, 147 (1962).

47. Powell, M. J. D., Computer J., 7, 155 (1964).

48. Rosen, J. B., J. Soc. Indust. Appl. MAth., 8, 181 (1960).

49. Rosenbrock, H. H., Computer J., 3, 175 (1960).

50. Saksena, C. P., and A. J. Cole, Operation Res., 19, 1 (1971).

51. Sauer, R. N., A. R. Colville, Jr., and C. W. Burwick, Hydro-Carbon Pro. Petrol. Ref., 43, 84 (1964).

52. Shah, B. V., R. J. Buehler, and O. Kempthorne, Naval Res. Rept., NR-042-207, No. 2, (1961).

53. Smith, C. S., N.C.B. Scientic Dept., Rept. No. S.C. 846/MR/40, (1962).

54. Spendley, W.,G. R. Hext, and F. R. Himsworth, Technometrics, 4, 441 (1962).

55. Swann, W. H., Central Instr. Lab. Res., Imperial Chem. Indus., Ltd., Note 64/3, (1964).

56. Westerberg, A. W., and C. J. Debrosse, AIChE J., 19, 335 (1973).

57. Wilde, D. J., Optimization Seeking Methods, Prentice-Hall, Inc., Englewood Cliffs, N. J., (1964)

58. Wolfe, P., Econometrica, 27, 382 (1959).

59. Wolfe, P., J. Soc. Indust. Appl. Math., 9, 481 (1961).

60. Yang, S. F., Master Thesis, Univ. of Oklahoma, Norman, Oklahoma, (1971).

61. Zangwill, W. D., Nonlinear Programming: A Unified Approach, Prentice-Hall, Inc., Englewood Cliffs, N. J., (1969).

NOMENCLATURE

a_{ij} coefficient of x_j of constraint i before normalization

A chemical reactant of chemical process problem

A_i rate constant of Arrhenius equation of reaction i

A_o constant defined in connection with Eq. (2-20)

b_i constant of constraint i

B chemical reactant of chemical process problem

B_i rate constant of Arrhenius equation of reaction i

B_1 a partition matrix of $(N_q'N_q)^{-1}$

B_2 a partition matrix of $(N_q'N_q)^{-1}$

B_3 a partition matrix of $(N_q'N_q)^{-1}$

B_4 a partition matrix of $(N_q'N_q)^{-1}$

c_i constant of F_i^o/RT

c_{ij} corresponding coefficient of $x_i x_j$

C intermediate of chemical reaction of chemical process problem

d_j corresponding coefficient of x_j^3

e_j corresponding coefficient of x_j

f objective function

F_i^o activation energy, Btu/lb

F_A flow rate of reactor input A, lb/hr

F_B flow rate of reactor input B, lb/hr

F_D flow rate of portion of column bottoms to plant fuel

F_G flow rate of G from decanter, lb/hr

F_P flow rate of product P, 40 million lb/yr=4763 lb/hr

F_{RA} flow rate of A from reactor, lb/hr

F_{RB} flow rate of B from reactor, lb/hr

F_{RC} flow rate of C from reactor, lb/hr

F_{RE} flow rate of E from reactor, lb/hr

F_R total flow rate from reactor, lb/hr

F_{RP} flow rate of P from reactor, lb/hr

$\underline{f}$ vector formed by a set of functions

$\underline{f}^i$ vector of $\underline{f}$ at point $\underline{x}^i$

ΔF matrix in connection with Eq. (3-14)

F matrix formed by vectors $\underline{f}^i$

$\underline{g}$ gradient vector of a function

G residual product in reactor

h_i constraint i

H enthalpy, Btu/lb

H_l hyperplane l

H_q hyperplane q

H(x) Hessian matrix at point $\underline{x}$

I identity matrix

J Jacobian matrix

k number of inequality constraints

k_i reaction coefficient of reaction i

m number of variables or functions

M_q manifold formed by the intersection of q hyperplanes

$\underline{n}_i$ normalized vector of constraint i

n_{ij} coefficient of variable j of constraint i after normalization

N_q constraint basis containing q hyperplanes

$\underline{p}$ direction vector

P pressure in reactor or desired product of chemical process

P_q projection matrix

q number of constraints in the constraint basis

r_{q-1} vector of $\left[N'_{q-1}N_{q-1}\right]^{-1}(N'_{q-1}n_q)$

r_i i-th element of vector $\underline{R}$

R universal gas constant

$\underline{R}$ constant vector in connection with Eq. (2-22)

T temperature in reactor

V volume of reactor

x_i independent variable i

$\underline{x}^*$ stationary point

X an n x n matrix of the auxiliary points

X new matrix of the auxiliary points

ΔX matrix in connection with Eq. (3-7)

y_r the r-th element of $X^{-1}\underline{x}^o$

$\underline{y}$ vector of $X^{-1}\underline{x}^0$

$\underline{z}$ unit direction vector

Z_1 set of nonzero Lagrangian multipliers

Z_2 set of zero Lagrangian multipliers

ρ density of the reactor solution lb/cu.ft

λ_i slack value of constraint i or the Lagrangian multiplier i

β_i value of $\underline{g}_{i+1}^2/\underline{g}_i^2$

η a positive number

τ_m maximum step length in certain direction

τ a positive number $0<\tau\leq\tau_m$

∇ gradient of a function

$\underline{\lambda}_q$ vector consisting of Lagrangian multipliers

$\underline{1}$ sum vector

α a constant of $\underline{1}X^{-1}\underline{x}^o$

$\emptyset$ new independent variable of chemical process problem

LIST OF COMPUTER PROGRAMMING

```
$JOB              00119656,TIME=50
C*****CONJUGATE GRADIENT METHOD WITH REEVE S DIRECTION
      DIMENSION POP(5),QNL(5,5),B(5),A(10),XF(5),ADML(10)
      DIMENSION ALF(5),Z(5),GL(5),PF(5),PI(5),ZP(5),ZO(5)
      DIMENSION X(5),G(9),PN(5),P(5),D(5)
      DIMENSION XB(5),GB(9),GH(11),XP(5)
      DIMENSION XHA(5),XHB(5)
      COMMON M,N,ML,MM,MA,I2,IQ,NX(5),BIGN(6,10),QN(5,5),XA(5,5)
      COMMON MAA,XS(4)
      COMMON COBJ(11,4)
      MAA=0
      MA=1
      M=4
      N=10
      MM=N+1
      ML=M+1
      READ (5,510) (X(I),I=1,M)
  510 FORMAT (5F8.5
  515 MAX=0
      JK=1
      JM=1
      IQ=0
      DO 516 I=1,M
      XP(I)=X(I)
  516 XS(I)=X(I)
      CALL LICM (X,COBJ,OBJ)
      MAA=MAA+1
C*****NORMALIZATION OF CONTRAINS
      DO 521 J=1,N
      F=0.
      DO 522 I=1,M
  522 F=F+BIGN(I,J)**2
      F=F**0.5
      DO 525 I=1,ML
  525 BIGN(I,J)=BIGN(I,J)/F
  521 CONTINUE
      DO 524 I=1,M
      NX(I)=0
      DO 524 J=1,M
  524 QN(I,J)=0.
  523 WRITE (6,526) (X(I),I=1,M)
  526 FORMAT (/,  INITIAL FEASIBLE POINT X   ,/,5E16.4
      CALL GRAFN (X,G)
      DO 535 I=1,M
      ALF(I)=0.
  535 P(I)=G(I)
C*****DETERMINATION OF SLACK VALUES
  545 DO 530 J=1,N
      F=0.
      DO 531 I=1,M
  531 F=F+X(I)*BIGN(I,J)
  530 ADML(J)=F-BIGN(ML,J)
      WRITE (6,533) (ADML(I),I=1,N)
  533 FORMAT(/,  VECTOR LMDA   ,/,15F8.3
      AZ=-1.E+03
      J=1
  413 IF (ADML(J)) 415,414,414
  415 IF (ABS(ADML(J))-0.0001) 414,414,401
  414 J=J+1
      IF (J-N) 413,413,536
```

```
  401 IF (ADML(J)) 403,408,408
  403 F=0.
      IF (AB-0.) 422,422,423
  422 DO 425 K=1,M
  425 F=F+ZP(K)*BIGN(K,J)
      GO TO 426
  423 DO 404 K=1,M
  404 F=F-ZP(K)*BIGN(K,J)
  426 IF (F) 408,408,405
  405 ETA=-ADML(J)/F
      IF (ETA-AZ) 408,408,406
  406 AZ=ETA
  408 J=J+1
      IF (J-N) 401,401,407
  407 IF (AB-0.) 402,402,409
  402 DO 411 I=1,M
  411 X(I)=X(I)+AZ*ZP(I)
      GO TO 523
  409 DO 410 I=1,M
  410 X(I)=X(I)-AZ*ZP(I)
      GO TO 523
  536 IF (IQ) 534,534,532
  532 WRITE (6,547) (NX(I),I=1,IQ)
  547 FORMAT(/,  ACTIVE CONSTRAINTS IN THE BASIS  ,/,5I4
  534 DC=0.
      DB=0.
      DO 546 I=1,M
      GL(I)=G(I)
  546 PI(I)=P(I)
C*****MAX IS THE NUMBER OF ITERATIONS
      MAX=MAX+1
      CALL MLQN(G,ALF,D)
C*****DETERMINE OPTIMAL POINT REACHES
      DO 548 I=1,M
  548 DB=DB+D(I)**2
      IF (IQ) 551,551,549
  549 DO 550 I=1,IQ
  550 DC=DC+ALF(I)**2
      WRITE(6,552) (ALF(I),I=1,IQ),DB
  552 FORMAT (/,  VECTOR ALF  ,/,5E12.4,  LETHPQG  ,1E12.4
  551 IF(DB-5.E-01) 553,553,555
  553 I=1
  554 IF (ABS(ALF(I))-0.001) 556,556,557
  557 IF (ALF(I)-0.) 556,556,555
  556 I=I+1
      IF(I-IQ) 554,554,1000
C*****FIND MAXIMUM DIAGONAL ELEMENT OF NQ NQ-1
  555 GAV=0.
      I=1
  584 IF (GAV-QN(I,I)) 587,587,590
  587 GAV=QN(I,I)
  590 I=I+1
      IF(I-IQ) 584,584,600
  600 ALFQ=-1.E+06
      I=1
  301 JB=NX(I)
      IF (JB) 620,620,305
C*****TEST ACTIVE CONSTRAINTS EXIST OR NOT
C*****FIND MAX. VALUE OF ALF WHICH ARE IN THE BASIS
  305 IF (ALFQ-ALF(I)) 306,306,309
```

```
  306 ALFQ=ALF(I)
      I3=I
C*****PLANE NX I3  SHUULD BE DROPPED
  309 I=I+1
      IF(I-IQ) 301,301,320
  320 F=0.5*ALFQ*GAV**(-0.5)
      WRITE (6,307) GAV,ALFQ,DB
  307 FORMAT (/,  GAV    ,1E12.4,  ALFQ    ,1E12.4,  LETHPQG    ,1E12.4
C*****DETERMINE PLANE NX I3  SHOULD BE DROPPED OR NOT
      JB=NX(I3)
      IF (DB-F) 617,617,620
  617 JM=JM+1
      IF (JM-2) 618,618,1000
  618 IF (I3-IQ) 625,628,625
C*****INTERCHANGE COLUMNS AND ROWS BEFORE DROPPING PLANE I3
  625 NX(I3)=NX(IQ)
      DO 631 K=1,IQ
      B(K)=QN(IQ,K)
      A(K)=QN(I3,K)
      QN(IQ,K)=A(K)
  631 QN(I3,K)=B(K)
      DO 634 K=1,IQ
      B(K)=QN(K,IQ)
      A(K)=QN(K,I3)
      QN(K,IQ)=A(K)
  634 QN(K,I3)=B(K)
C*****NEW QN MATRIX AFTER DROPPING PLANE Q
  628 B4I=1./QN(IQ,IQ)
      LN=IQ-1
      DO 202 I=1,LN
  202 B(I)=QN(I,IQ)*B4I
      DO 204 I=1,LN
      DO 204 J=1,LN
  204 QNL(I,J)=QN(I,IQ)*B(J)
      DO 209 I=1,LN
      DO 209 J=1,LN
  209 QN(I,J)=QN(I,J)-QNL(I,J)
      IQ=IQ-1
      WRITE (6,212) JB,I3
  212 FORMAT(/,  OUT GOING PLANE IN BIGN IS  ,1I4,  IN NQ IS  ,1I4,  M
     1  ATRIX QN IS
      DO 214 I=1,IQ
  214 WRITE (6,216) (QN(I,J),J=1,IQ)
  216 FORMAT (5E12.4
  620 CALL MLQN(P,PN,PQP)
      DC=0.
      DO 640 I=1,M
  640 DC=DC+PQP(I)**2
      DC=DC**0.5
      DO 643 I=1,M
  643 Z(I)=PQP(I)/DC
      WRITE (6,648) (Z(I),I=1,M)
  648 FORMAT (/,  UNIT DIRECTION VECTOR Z  ,/,5E12.4
C*****CHOOSE NONACTIVE CONSTRAINTS
      Y=1.E+05
      I2=0
      DO 652 I=1,N
      IF (IQ) 658,658,655
  655 DO 656 J=1,IQ
      IF (I-NX(J)) 656,652,656
```

```
  656 CONTINUE
  658 IF (ABS(ADML(I))-5.E-03) 657,657,654
  654 F=0.
      DO 653 K=1,M
  653 F=F+Z(K)*BIGN(K,I)
      IF (F) 649,652,652
  649 AA=ABS(ADML(I)/F)
      IF (Y-AA) 652,647,647
  657 Y=0.
      GO TO 659
  647 Y=AA
  659 I2=1
  652 CONTINUE
      BPS=Y
C*****I2 IS THE PLANE COMING IN
C*****Y IS MIN. OF ABS LMDA/N Z  FOR WHICH N Z ARE LESS THAN ZERO
C*****BPS OS THE MAX. POSSIBLE STEP LENGTH
C*****XB IS THE POINT WITH POSSIBLE MAX. STEP
C*****IF Y 0. INTERPOLATION IS NOT NECESSARY
 1200 DO 1210 I=1,M
 1210 XB(I)=X(I)+BPS*Z(I)
      WRITE (6,1220) Y,(XB(I),I=1,M)
 1220 FORMAT(/,  POINT XB WITH POSSIBLE MAX. STEP LENGTH  ,1F10.3,
     1/,5F12.7
C*******FIABONACCI UNTERPOLATION
      MM=1
      IS=1
      DO 1221 I=1,M
      XF(I)=X(I)
      XHB(I)=X(I)+0.618*(XB(I)-X(I))
 1221 XHA(I)=XB(I)-0.618*(XB(I)-X(I))
      CALL FUNEV (XHA,XS,GH,FA)
      CALL FUNEV (XHB,XS,GH,FBB)
 1224 IF (FA-FBB) 1225,1225,1228
 1225 DO 1226 I=1,M
      XF(I)=XHA(I)
      XHA(I)=XHB(I)
      XHB(I)=XF(I)+0.618*(XB(I)-XF(I))
 1226 XP(I)=XHA(I)
      FA=FBB
      CALL FUNEV (XHB,XS,GH,FBB)
      GO TO 1229
 1228 DO 1230 I=1,M
      XB(I)=XHB(I)
      XHB(I)=XHA(I)
      XHA(I)=XB(I)-0.618*(XB(I)-XF(I))
 1230 XP(I)=XHB(I)
      FBB=FA
      CALL FUNEV (XHA,XS,GH,FA)
 1229 IS=IS+1
      IF (IS-14) 1224,1224,1235
 1235 DO 1236 I=1,M
 1236 XF(I)=XP(I)
 1255 CALL FUNEV (XF,XS,GH,FF)
      WRITE (6,1240) (XF(I),I=1,M)
 1240 FORMAT (/,  INTERPOLATION POINT  ,5F12.7
      WRITE (6,1256) FF
 1256 FORMAT (  OBJ. FUNCTION OF INTERPOLATION POINT  ,/,1F12.5
      DS=0.
      DO 1257 I=1,M
```

```
 1257 DS=DS+(XF(I)-X(I))**2
      DS=DS**0.5
      IF (ABS(DS-Y)-0.0005) 1242,1242,1258
 1258 Y=0.5*BPS
 1242 CALL GRAFN (XF,G)
      MM=N+1
      MAA=MAA+1
  683 D1=0.
      F=0.
      DO 678 I=1,M
      D1=D1+G(I)**2
  678 F=F+GL(I)**2
      BET=D1/F
      DO 682 I=1,M
  682 PF(I)=G(I)+BET*PI(I)
C IF THE FINAL STEP LENGTH IS GREATER THAN OR EQUAL TO  BPS ADD PLANE C
      IF (BPS-Y) 694,694,690
  694 DO 698 I=1,M
      X(I)=XF(I)
      ZP(I)=X(I)-XS(I)
  698 P(I)=PF(I)
      WRITE (6,700) I2
  700 FORMAT(/,  THE COMING PLANE IS  ,I3
      WRITE (6,707) (P(I),I=1,M)
  707 FORMAT (/,  DIRECTION P  ,/,5E12.4
      CALL ADCOJ
      IF (MAX-20) 545,545,1050
C*****IF THE FINAL STEP LENGTH IS LESS THAN BPS, BASIS IS NOT CHANGED
  690 WRITE (6,723)
  723 FORMAT(/,  THE FINAL STEP LENGTH IS LESS THAN BPS BASIS IS
     1  NOT CHANGED
      DO 724 I=1,M
      X(I)=XF(I)
      ZP(I)=X(I)-XS(I)
  724 P(I)=PF(I)
      IF(MOD(MAX,M+1)) 727,725,727
  725 DO 726 I=1,M
  726 P(I)=G(I)
  727 IF (MAX-20) 545,545,1050
 1000 WRITE (6,728) MAX
  728 FORMAT (  OPTIMAL POINT REACHES IN ITERATIONS  ,I3
      WRITE (6,800) (X(I),I=1,M)
  800 FORMAT (//,  THE OPTIMAL POINT IS  ,/,5F12.7
      WRITE (6,803) (P(I),I=1,M)
  803 FORMAT (//,  DIRECTION P  ,/,5E12.4
      ZX=0.
      EE=0.
      BB=0.
      DO 804 I=1,M
      ZX=ZX+ZP(I)**2
      EE=EE+X(I)**2
  804 BB=BB+XS(I)**2
      ZX=ZX**0.5
      AB=EE-BB
      AC=ABS(EE-BB)
  808 IF (ABS(AC)-1.E-06) 1111,1111,805
  805 DO 807 I=1,M
  807 ZP(I)=ZP(I)/ZX
      MA=MA+1
      IF (MA-16) 515,1111,1111
```

```
 1050 WRITE (6,806) MAX
  806 FORMAT(  DOES NOT CONVERGE IN  ,I12,  ITERATION
 1111 I1=1
      STOP
      END

      SUBROUTINE LICM (XH,FAU,OBJ)
C*****LINEARIZATION OF CONTRAINTS
      DIMENSION R8(11),XAI(5,5),YV(7),FO(11),FAU(11,4),XH(5)
      COMMON M,N,ML,MM,MA,I2,IQ,NX(5),BIGN(6,10),QN(5,5),XA(5,5)
      CCMMCN MAA,XS(4)
      COMMCN COBJ(11,4)
      DO 120 I=1,ML
      DO 120 J=1,N
  120 BIGN(I,J)=0.
      DO 101 I=1,11
      DO 101 J=1,4
  101 COBJ(I,J)=0.
      CALL FUNEV(XH,XS,FO,OBJ)
      CALL GRAFN (XH,YV)
      DO 102 I=3,6
  102 COBJ(I,I-2)=1.
      COBJ(7,3)=-1.
      COBJ(8,4)=-1.
      COBJ(10,1)=-1.
      COBJ(11,2)=-1.
      DO 345 I=1,N
      DO 345 J=1,M
  345 BIGN(J,I)=COBJ(I+I,J)
      DO 350 I=1,N
      DO 349 J=1,M
  349 BIGN(ML,I)=BIGN(ML,I)+COBJ(I+1,J)*XH(J)
  350 BIGN(ML,I)=BIGN(ML,I)-FO(I+1)
      DO 355 I=2,MM
  355 WRITE (6,356) (COBJ(I,J),J=1,M)
  356 FORMAT(5E12.4
      DO 357 J=1,N
  357 WRITE (6,358) (BIGN(I,J),I=1,ML)
  358 FORMAT(5E12.4
      RETURN
      END

      SUBROUTINE GRAFN (H,G)
      DIMENSION X(5),G(5),H(5)
      COMMON M,N,ML,MM,MA,I2,IQ,NX(5),BIGN(6,10),QN(5,5),XA(5,5)
      COMMON MAA,XS(4)
      CCMMON COBJ(11,4)
      X(1)=H(1)*1.E+04
      X(2)=H(2)*1.E+03
      X(3)=H(3)*0.1
      X(4)=H(4)*10.
      RCA=5.9755E+09*EXP(-12000./(X(4)+580.))
      RCB=2.5962E+12*EXP(-15000./(X(4)+580.))
      RCC=9.6283E+15*EXP(-20000./(X(4)+580.))
      FRP=4763.+0.1*X(1)
      R2=0.5*X(1)*X(3)
      R3=2.*(R2-4763.-FRP*X(3)+4763.*X(3))
      R1=0.5*(R3+X(2)*X(3))+R2
      T=R3/(RCC*FRP*X(2))
      FRB=R2/(RCB*X(2)*T)
```

```
      FB=R1+R2+FRB*X(3)
      FRA=R1/(RCA*FRB*T)
      FG=1.5*R3
      FR=FRA+FRB+X(1)+X(2)+FRP+FG
      FD=X(3)*(FR-FG-4763.)
      FA=R1+FRA*X(3)
      VOL=FR**2*T/50.
      OF=368.*4763.+8.4*FD-28.*FA-42.*FB-14.*FG-0.37*FR
      RPRE=0.1
      R2RE=0.5*X(3)
      R3RE=2.*(R2RE-X(3)*0.1)
      R1RE=0.5*R3RE+R2RE
      TRE=R3RE/(RCC*FRP*X(2))-R3*RPRE/(RCC*X(2)*FRP**2)
      RBRE=R2RE/(RCB*X(2)*T)-R2*TRE/(RCB*X(2)*T**2)
      BRE=R1RE+R2RE+X(3)*RBRE
      RARE=(R1RE-R1*RBRE/FRB-R1*TRE/T)/(RCA*FRB*T)
      GRE=1.5*R3RE
      RRE=RARE+RBRE+1.+RPRE+GRE
      DRE=X(3)*(RRE-GRE)
      ARE=R1RE+X(3)*RARE
      VRE=2.*FR*T*RRE/50.+FR**2*TRE/50.
      R1RC=0.5*X(3)
      TRC=-R3/(RCC*FRP*X(2)**2)
      RBRC=-R2/(RCB*T*X(2)**2)-R2*TRC/(RCB*X(2)*T**2)
      BRC=X(3)*RBRC+R1RC
      RARC=(R1RC-RBRC*R1/FRB-TRC*R1/T)/(RCA*FRB*T)
      RRC=RARC+RBRC+1.
      DRC=X(3)*RRC
      ARC=X(3)*RARC+R1RC
      VRC=2.*FR*T*RRC/50.+FR**2*TRC/50.
      R2P=0.5*X(1)
      R3P=2.*(R2P-FRP+4763.)
      R1P=0.5*(R3P+X(2))+R2P
      TP=R3P/(RCC*FRP*X(2))
      RBP=R2P/(RCB*X(2)*T)-R2*TP/(RCB*X(2)*T**2)
      BP=R1P+R2P+FRB+X(3)*RBP
      RAP=(R1P-R1*RBP/FRB-TP*R1/T)/(RCA*FRB*T)
      GP=1.5*R3P
      RP=RAP+RBP+GP
      DP=FR-FG-4763.+X(3)*(RP-GP)
      AP=R1P+FRA+X(3)*RAP
      VP=2.*FR*T*RP/50.+FR**2*TP/50.
      RCAT=RCA*12000./((X(4)+580.)**2)
      RCBT=RCB*15000./((X(4)+580.)**2)
      RCCT=RCC*20000./((X(4)+580.)**2)
      TT=-R3*RCCT/(FRP*X(2)*RCC**2)
      RBT=-R2*RCBT/(X(2)*T*RCB**2)-R2*TT/(RCB*X(2)*T**2)
      BT=X(3)*RBT
      RAT=-FRA*(RCAT/RCA+RBT/FRB+TT/T)
      RT=RBT+RAT
      DT=RT*X(3)
      AT=RAT*X(3)
      VT=2.*FR*T*RT/50.+FR**2*TT/50.
      VOD=50.*VOL**2
      COBJ(9,1)=R3RE*10.
      COBJ(9,3)=R3P*1.E-04
      COBJ(9,1)=-RRE
      COBJ(9,2)=-RRC*1.E-01
      COBJ(9,3)=-RP*1.E-05
      COBJ(9,4)=-RT*1.E-03
```

```
      G(1)=(8.4*DRE-28.*ARE-42.*BRE-14.*GRE-0.37*RRE)/(VOL*50.)-OF*VRE/
     1VOD
      G(2)=(8.4*DRC-28.*ARC-42.*BRC-0.37 *RRC)/(VOL*50.)-OF*VRC/VOD
      G(3)=(8.4*DP-28.*AP-42.*BP-14.*GP-0.37*RP)/(VOL*50.)-OF*VP/VOD
      G(4)=(8.4*DT-28.*AT-42.*BT-0.37*RT)/(VOL*50.)-OF*VT/VOD
      G(1)=G(1)*1.E+04
      G(2)=G(2)*1.E+03
      G(3)=G(3)*0.1
      G(4)=G(4)*10.
      WRITE (6,500) (G(I),I=1,M)
  500 FORMAT (/,  GRADIENT IS  ,/,5E13.5
      RETURN
      END

      SUBROUTINE FUNEV (HH,XT,FO,OBJ)
      DIMENSION  FO(11),FAU(11,4),XX(5),HH(5),XT(4)
      DATA NEU/0/
      COMMON M,N,ML,MM,MA,I2,IQ,NX(5),BIGN(6,10),QN(5,5),XA(5,5)
      COMMON MAA,XS(4)
      COMMON COBJ(11,4)
      XX(1)=HH(1)*1.E+04
      XX(2)=HH(2)*1.E+03
      XX(3)=HH(3)*0.1
      XX(4)=HH(4)*10.
      DES=50.
      GC=1.5
      BE=0.5
      CP=2.
      BC=0.5
      FP=4763.
      RCA=5.9755E+09*EXP(-12000./(XX(4)+580.))
      RCB=2.5962E+12*EXP(-15000./(XX(4)+580.))
      RCC=9.6283E+15*EXP(-20000./(XX(4)+580.))
      FRP=FP+0.1*XX(1)
      R2=BE*XX(1)*XX(3)
      R3=CP*(R2+FP-FRP*XX(3)+FP*XX(3))
      R1=BC*(R3+XX(2)*XX(3))+R2
      T=R3/(RCC*FRP*XX(2))
      FRB=R2/(RCB*XX(2)*T)
      FB=R1+R2+FRB*XX(3)
      FRA=R1/(RCA*FRB*T)
      FG=GC*R3
      FR=FRA+FRB+XX(1)+XX(2)+FRP+FG
      FO(9)=(30.*FP-FR)*1.E-04
      IF (MM-1) 405,400,405
  400 FD=XX(3)*(FR-FG-FP)
      FA=R1+FRA*XX(3)
      VOL=FR**2*T/DES
      FO(1)=(368.*FP+8.4*FD-28.*FA-42.*FB-14.*FG-0.37*FR)/(VOL*DES)-10.
      OBJ=FO(1)
      NEU=NEU+1
      RETURN
  405 FO(2)=2.*(0.4*HH(1)*HH(3)-4.763)
      FO(3)=HH(1)
      FO(4)=HH(2)
      FO(5)=HH(3)
      FO(6)=HH(4)
      FO(7)=10.-HH(3)
      FO(8)=10.-HH(4)
      FO(10)=1.15*XS(1)-HH(1)
```

```
      FO(11)=1.15*XS(2)-HH(2)
      RETURN
      END

      SUBROUTINE MLQN(P,PN,D)
      DIMENSION P(5),D(5),PN(5)
      COMMON M,N,ML,MM,MA,I2,IQ,NX(5),BIGN(6,10),QN(5,5),XA(5,5)
      COMMON MAA,XS(4)
      COMMON COBJ(11,4)
      IF (IQ) 55,55,5
    5 DO 10 I=1,IQ
   10 PN(I)=0.
      DO 30 I1=1,IQ
      JB=NX(I1)
      PQ=0.
      DO 20 J=1,M
   20 PQ=PQ+BIGN(J,JB)*P(J)
      DO 30 J=1,IQ
   30 PN(J)=PN(J)+PQ*QN(J,I1)
      DO 40 I=1,M
      DIF=P(I)
      DO 50 J1=1,IQ
      J=NX(J1)
   50 DIF=DIF-BIGN(I,J)*PN(J1)
   40 D(I)=DIF
      RETURN
   55 DO 60 I=1,M
      D(I)=P(I)
   60 PN(I)=0.
      RETURN
      END

       SUBROUTINE ADCOJ
C*****SUBROUTINE FOR UPDATING QN MATRIX
      DIMENSION PN(5),D(5)
      COMMON M,N,ML,MM,MA,I2,IQ,NX(5),BIGN(6,10),QN(5,5),XA(5,5)
      COMMON MAA,XS(4)
      COMMON COBJ(11,4)
C*****TEST THE EXISTENCE OF ACTIVE CONSTRAINTS IN THE BASIS
      IF (IQ) 200,200,151
  151 CALL MLQN(BIGN(1,I2),PN,D)
      SUM=0.
C*****CALC. NQ  I-NQ NQ NQ NQ  NQ
      DO 170 J=1,M
  170 SUM=SUM+D(J)**2
      B4=1./SUM
      DO 175 I=1,IQ
      B2I=-B4*PN(I)
C*****FORMING THE NEW QN MATRIX
      DO 180 I1=1,IQ
  180 QN(I,I1)=QN(I,I1)-B2I*PN(I1)
      QN(I,IQ+1)=B2I
  175 QN(IQ+1,I)=B2I
  184 QN(IQ+1,IQ+1)=B4
      IQ=IQ+1
      NX(IQ)=I2
      WRITE (6,185)
  185 FORMAT (/,  MATRIX QN
      DO 186 I=1,IQ
  186 WRITE (6,187) (QN(I,J),J=1,IQ)
  187 FORMAT (5E12.4
      RETURN
  200 B4=0.
      DO 189 I=1,M
  189 B4=B4+BIGN(I,I2)**2
      B4=1./B4
      GO TO 184
      END

$EXEC
```