R. R. Hilleary

THE TANGENT SEARCH METHOD OF CONSTRAINED MINIMIZATION.

TR/RP No. 59

UNITED STATES
NAVAL POSTGRADUATE SCHOOL

THE TANGENT SEARCH METHOD OF CONSTRAINED MINIMIZATION

by

R. R. Hilleary

March 1966

TECHNICAL REPORT/RESEARCH PAPER NO. 59

Distribution of this document is unlimited

TABLE OF CONTENTS

UNITED STATES NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral E. J. O'Donnell, USN, Superintendent

R. F. Rinehart, Academic Dean

ABSTRACT:

This paper describes an adaptation of the Direct Search Method which is designed to find the local minimum of an arbitrary, explicitly-stated function of more than one variable, subject to arbitrary, non-linear constraints. This is sometimes called the general problem of mathematical programming. The algorithm given here usually performs its exploratory procedure in hyperplanes approximately tangent to the constraint hypersurfaces when the base point is in the vicinity of such boundaries. Therefore, the author designates it the "Tangent Search Method".

Calculation of the partial derivatives of a constraint function is necessary when it is violated. However, the method never requires evaluation of any derivative of the objective function. Performance of Tangent Search on various test problems discussed in this paper is generally superior or similar to results with three other recently-published algorithms.

Prepared by: R. R. Hilleary

Mathematician, Computer Facility

Approved by:

D. G. Williams

Head, Computer Facility

Released by:

C. E. Menneken

Dean of

Research Administration

U. S. Naval Postgraduate School Technical Report/Research Paper No. 59

March 1966

UNCLASSIFIED

TANGENT SEARCH METHOD OF CONSTRAINED MINIMIZATION

Introduction

Each of three recent articles (1, 2, 3) has contributed a new method for solving the general problem of minimizing (or optimizing) an arbitrary function subject to arbitrary constraints. The objective functions to be minimized by these methods and the constraint functions considered are real, continuous, and, in general, non-linear. However in practical problems, for example "Problem A" of Box (3), many of the constraints prove to be simple upper or lower bounds on independent variables. All constraints are expressible as inequalities.

The first two of these new methods can be classified as extensions of the Direct Search method of Hooke and Jeeves (4). In these algorithms Direct Search (alias "pattern search"), or a simplification of it, is employed until a constraint is encountered. Then an alternative or adaptive procedure is substituted. This attempts to find a move along the constraint boundary which will continue minimization of the objective function. Trials are made by explorations from the base point which always remains in the feasible region where no constraint is violated. No trial move is accepted as a new base unless it is both feasible and also lowers the value of the objective function from the previous base value.

In Klingman and Himmelblau (1) the alternative exploratory move is called the "multiple-gradient summation technique". Whenever the exploratory move of the Direct Search fails near a constraint boundary this method computes a "new successful direction". This is found as the vector sum of the normalized gradients of the contacted constraint and the objective function. After this adaptive move an attempt is made to return to the Direct Search

[illegible]

algorithm in order to obviate the evaluation steps involved in gradient calculation. The authors of this method do not discuss possible degrading of results in cases when the gradients can only be approximated. It would seem that such would be the more typical situation in actual minimization requirements, such as Problem A of Box (3). However, perhaps the authors had in mind recently developed methods for the automatic calculation of derivatives, such as those reported in Wengert (5) or Smith (6).

A particularly lucid and diagrammatic explanation of the reason Direct Search often either fails or becomes inefficient at a boundary is given in Klingman and Himmelblau (1). Therefore this information will not be included in the present discussion.

In the Sequential Search method of Glass and Cooper (2) the Direct Search rule is slightly modified. To keep computationally-expensive exploratory moves to a minimum their method always continues in an established successful direction until failure occurs. That is, the usual post-pattern exploration of Direct Search is omitted except when the pattern move has failed. Consequently it becomes necessary to build an explicit lengthening process into the pattern move in order to accelerate when an advantageous direction has been found.

When the usual exploratory move fails near a constraint boundary, the method of Glass and Cooper resorts to an alternative routine which in certain situations must, itself, be extended to a further alternative procedure. They show that at such a base point the new required direction can be found as the solution of a linear programming problem. The authors state that this sub-problem can always be solved by the Simplex Method provided a solution exists. However, the constraint hyper-surface is often convex, as defined below. In this case the new direction returns

immediately to the non-feasible region. Therefore an "extension" to the alternative move provides systematic rotation of the indicated direction until it clears the constraint. After an alternative move an attempt is made to return to the modified Direct Search algorithm, since there is obviously considerable expenditure of computational effort in carrying out the procedures outlined above.

Unfortunately only two very simple examples, each with only two independent variables, were included in Cooper and Glass (2). Therefore the present author is not convinced that it offers an efficient method of solving practical problems.

The Complex Method of M. J. Box (3) falls into a different category. It is a novel and elegant adaptation of the Simplex Method of linear programming, and, therefore, is not conceptually allied to the pattern search basis of the two methods outlined above. Starting with any one feasible first point in n-dimensional space a "complex" of 2n vertices is constructed by selecting random points from the feasible region. Then one simple computational loop is employed. These instructions find the current worst vertex, that is, the vertex with largest corresponding value for the objective function, and replace it by its over-reflection through the centroid of all other vertices. If the vertex to be replaced is considered as a vector in n-space, its overflection is opposite in direction, increased in length, and collinear with the old vertex and the centroid of the other vertices. When the overflection is not feasible or remains worst, it is displaced half way toward the centroid. Constraints are classified as explicit or implicit. The former are simple upper or lower bounds on independent variables. The latter are treated as upper or lower bounds on arbitrary functions of the independent variables. This distinction is

used profitably in the Box algorithm.

Eventually all vertices converge to a point which is taken as the solution. Preliminary experience seems to show that the Complex Method is very dependable but not completely infallible. (See problems 3, 5, 7 and 9 in the numerical examples below.)

Basic Concepts of the Tangent Search Method.

The Tangent Search Method which is described in the present paper falls into the pattern search family of suggested solutions to the problem of constrained minimization. The modified pattern search of Glass and Cooper is used except that the test for constraint violations is postponed until the exploratory move is completed for all independent variables. Whenever a trial move passes a boundary the partial derivatives of the violated constraint function are approximated at the current feasible base point. This is performed by the most elementary difference method possible. Every effort is made to eliminate unnecessary use of the auxiliary procedure which evaluates constraint functions.

Then an exploratory move is made in the hyperplane which is approximately tangent to the constraint hyper-surface. (The hyperplane is, in general, slightly removed into the feasible region so that it contains the current base.) This substitute exploratory move seemed to the present author a more obvious adaptation of the Hooke and Jeeves algorithm than either of the two described above. If it is successful it is called a "tangent move". Economy over the method of Klingman and Himmelblau is achieved since evaluation of the gradient of the objective function is completely avoided. The difference technique used to approximate the partial derivatives of the constraint function will be described in a separate section before the numerical examples are considered.

An additional feature of the algorithm here described is the allowance made for small perturbations of the base point when the tangent exploration procedure fails several times consecutively. Under some circumstances it seems better to transfer provisionally the base point to a higher feasible position on the objective function than to continue reducing step sizes. This type of move, called "jump move" below, helps to extract the base point from difficult slots and corners in the feasible region caused by the multiplicity of constraints.

Finally, it is obvious that the directions for exploration in the tangent hyperplace can be chosen in any number of ways. The attempt of this method is to preserve the basic logic of the usual exploratory move, but to allow a variety of possibilities within that outline. Previous success in finding a good "tangent move" for a constraint dictates the mode of the tangential exploration first tried if that constraint is violated again.

Comparisons of the performance of the proposed method and those previously mentioned will be given in the penultimate sections of this paper. The immediately following sections give an informal mathematical definition of the problem of constrained minimization and then a detailed exposition of the Tangent Search Method.

The Problem of Constrained Minimization.

Let $X = (x_1, x_2, \ldots, x_n)$ be a vector in n- dimensional space. Let $f(X)$ and $g_j(X)$, $(j = 1, 2, \ldots, m)$, be continuous functions which can be evaluated for any X within the vector space.

Any X for which

$$g_j(X) \geq 0, \quad (j = 1, 2, \ldots, m),$$

will be called "feasible".

Also, any constraint hyper-surface,

$$g_k(X) = 0,$$

will be called "convex" in a region of interest if every point of every hyperplane tangent to that constraint hyper-surface in the area of interest is not feasible.

The general problem of constrained minimization is to determine the vector, X_s, such that $f(X_s)$ is less than all other $f(X)$, subject to the constraints, $g_j(X_s) \geq 0$, for all $j = 1, 2, \ldots, m$.

If for any point, X,

$$g_j(X) < 0, \text{ then}$$

the jth constraint is "violated" at X.

For the problems of constrained minimization considered here,

$$g_k(X_s) = 0, \text{ where } 1 \leq k \leq m.$$

That is, typically, the solution will lie on one or more constraint boundaries.

An important requirement of the Tangent Search Method is that the first partial derivatives of $g_j(X)$ with respect to each X_i, $(i = 1, 2, \ldots, n)$ should be continuous and computable, at least in the vicinity of any violation. Special provision is made for cases where such partial derivatives are zero.

Details of the Presentation: Chart Symbols and Notes

The discussion below will assume familiarity with the method of Direct Search as given by Hooke and Jeeves (4). It is also outlined in Glass and Cooper (2) and Klingman and Himmelblau (1). The Tangent Search

algorithm will be communicated by means of detailed flow diagrams similar to those of Hooke and Jeeves. Descriptive flow diagrams will be omitted. However each box of the detailed flow diagrams will be numbered within parentheses. Corresponding descriptive titles (underlined) and additional explanatory notes will be given in the text following each chart.

Notation will differ from that of previous papers in order to avoid Greek symbols which are inconvenient on a typewriter. Flow chart conventions will be more standard:

1. The diamond will represent a decision function.
2. The circle will represent a connector. Off-page connector numbers will be preceded by the referenced chart number(s); for example, connector 5 of chart 3 would be denoted "3/5".
3. The square will represent starting/stopping points in the main procedure and entry/exit points for auxiliary procedures.
4. Processing functions are represented by rectangles.
5. Input/output functions are shown as rectangles with rounded corners. Only the most basic of these are given.
6. When an operation involves a matrix, indicated by capitalization, it should be understood that it is carried out on all elements. Otherwise subscripts indicate the elements referenced.
7. The "equals" sign is used as the replacement symbol; the evaluated right hand side replaces the former value (if any) of the variable denoted on the left hand side.

Auxiliary Procedures

Two external auxiliary procedures are used in various parts of the flow diagrams. The actual forms of these vary according to the objective function

to be minimized and the constraints imposed on the solution. The notation and purpose of each are given below:

1. F(X) is a function procedure which, when called, produces a value of the objective function at any point, X, whether or not X is feasible.

2. C(X) is a subroutine which, when called, produces a column matrix of m elements. These are the values at the point X of the constraint functions, g_j, for j = 1, 2, ..., m.

The eight internal auxiliary procedures are not changed from problem to problem. They are separated from the main procedure in the discussion below either because they are called from more than one site or because logical delineation of the function performed may clarify the presentation. The notational representation for the internal auxiliaries is adapted from Hooke and Jeeves (4). A colon separates the auxiliary name-symbol from its parameter list. The list is divided between input and output parameters by a semicolon. As an example, the notation

"Q : x, X; y, Y",

would indicate that the internal auxiliary, Q, using current values of the variables, x (a scalar) and X (a vector or matrix), causes replacement of the values of the variables, y (a scalar) and Y (a vector). If an argument is both an input and output parameter, it will appear only in the latter list.

The details of the internal auxiliary procedures E, E_0, F_0, H, T, and V are given in flow diagrams of later sections.

The internal auxiliary procedure, M, is simply the usual minimum-value function: e.g.,

"M:X; x"

indicates that the algebraically smallest element of X replaces the present value (if any) of x.

The internal auxiliary procedure, S, is the usual sign function; e.g.,

"S : x, y; z"

indicates that the value of z is replaced by the absolute value of x, prefixed with the sign of y.

Matrix Variable Symbols

A Values of the m constraint functions at current best feasible move produced by auxiliary procedure T.

B Values of the m constraint functions at base point.

D Current exploration step sizes for the n variables.

G Values of the m constraint functions at trial move.

I Matrix of indices containing one column of n variable numbers for each of the v constraints presently violated.

J Auxiliary column matrix, defined in the same manner as K.

K Indices of the v currently-violated constraints.

L Lower limits allowed on the corresponding n values of D. When $D_i < L_i$, for all i = 1, 2, ..., n, the search is terminated.

N Table containing the current coupling mode for each of the m constraints.

P Matrix of v columns, each containing n first partial derivatives of a violated constraint function with respect to the independent variables. These are arranged in ascending order of absolute value.

U Upper limits allowed on the corresponding n values of D.

W Base point prior to last jump move, containing n values of the variables.

X Trial move (n elements). Also, the vector of independent variables of n-space in mathematical discussion.

Y Current base point (n elements).

Z In the main procedure, the previous base point. It is also used as a temporary array by auxiliary procedures, H and T.

Scalar Variable Symbols

a Index of primary variable of any component of a tangent exploratory move.

b Index of secondary variable in any component of a tangent exploratory move.

c Failure counter; controls application of jump move option.

d Constant used in step size control.

d_b Decrement of the secondary variable in any component of a tangent exploratory move.

e Temporary indicator, used variously.

f Value of objective function after trial move. Also denotes the objective function in the mathematical discussion.

f_a Value of objective function, corresponding to tentative jump move.

f_w Value of objective function at base prior to last jump move.

f_x Function value used temporarily by conventional exploratory move.

f_y Value of objective function at current base point.

g Coupling constant for active variables in auxiliary procedure T.

g_j The j-th constraint function (in mathematical discussion).

h Coupling constant for passive variables in auxiliary procedure T.

i, j, k Utility indices used variously.

m Number of constraints.

n Number of independent variables.

p_1 First step extension factor.

p_2 Second step extension factor.

p_a Partial derivative of constraint function with respect to primary variable of any component of auxiliary procedure T.

p_b Partial derivative with respect to secondary variable. (See p_a above.)

q Temporary storage and utility index.

r Step reduction factor.

s Utility switch, used variously.

t Feasible move indicator.

u Auxiliary counter, used in the same manner as v.

v Number of constraints currently violated.

w Coupling variable index, used in auxiliary procedure T.

x Variable increment used in auxiliary procedure H.

x_a, x_b Primary and secondary, respectively, trial increments at any component of tangent exploratory move.

x_i The i-th variable in n-space in mathematical discussion.

y_a, y_b Previous values of primary and secondary variables before trial component.

z Variable increment used in auxiliary procedure H.

Details of the Method; Initialization and Search for Feasible Start

The detailed flow diagram for these phases of a solution is given in Chart 1. At times it is a considerable convenience to require that the algorithm, itself, find a feasible initial point before the actual search for constrained minimization is initiated. This is easily accomplished by using a variation of the usual unconstrained Direct Search to minimize the function F_o : X, G; f_x. This internal auxiliary procedure, which is

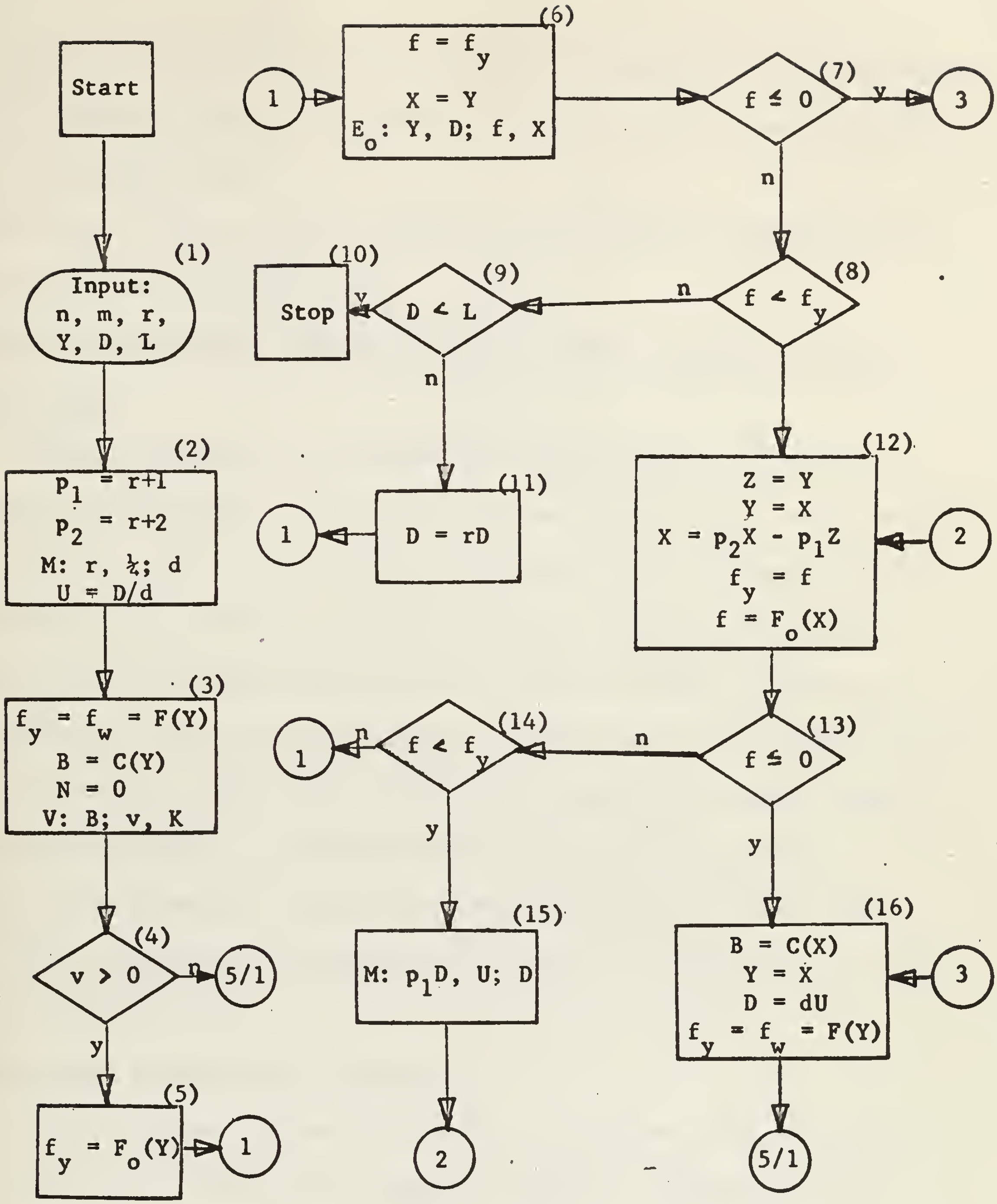


Chart 1: Initialization and Search for a Feasible Start

Main Procedure, Part I

shown in Chart 3, simply calculates the sum of the squares of all constraint functions which are negative at point X. As soon as f_x becomes zero, the actual solution can begin.

Descriptive titles for each numbered block of Chart 1 and additional explanatory notes are given below.

(1) Input starting point, computational constants, initial and minimal step sizes.

Optimal choice of r, the step reduction factor, has not been thoroughly investigated. The example problems discussed in a later section were run with r = .25, .375, and .5. These three values seemed to produce approximately-equivalent satisfactory results for various problems in unconstrained minimization previously solved by the author in trials of the conventional Direct Search algorithm on the Control Data 1604 computer. However, as will be shown in the discussion of numerical examples below, significant differences in performance developed in some problems of constrained minimization. According to Hooke and Jeeves (4) the simple Direct Search algorithm in unconstrained problems is not sensitive to this choice, as long as $r < 1$.

(2) Initialize computational constants.

The pattern and exploration step size extension factors are both $1 + r$; however, the exploration steps are limited to a maximum size of four times the original values.

(3) Evaluate objective function and constraint functions at starting point; check for violations.

The coupling modes are initialized at zero for all constraints. The use of these quantities is explained in a subsequent section. Details of the internal auxiliary procedure V, are given in Chart 2. Briefly stated,

this procedure examines the constraint functions, counts the number of violations, and stores the index number of each violated constraint in K.

(4) Is any constraint violated?

If not, exit to the actual constrained search is made. If so, a feasible starting point must first be found.

(5) Evaluate special function at origin.

Details of this internal function, F_o, are given in Chart 3.

(6) Starting at base point, perform exploratory move, E_o, using the special function, F_o.

A detailed flow diagram of the Hooke's and Jeeves' exploratory procedure, E, is given in Chart 4. The auxiliary, E_o, is identical to E, except that all appearances of "$f_x = F(X)$" are to be replaced by "$f_x = F_o(X)$".

(7) Is trial value of special function zero?

If so, a feasible starting point has been found.

(8) Is trial value of special function below the base value?

If so, a pattern move can be attempted.

(9) Are all step sizes below corresponding minimal criteria?

(10) Stop

If control arrives at Item (10), Direct Search has failed to locate any feasible starting point for the actual search. Perhaps the constraints, as defined in external auxiliary procedure C(X), are inconsistent, or the elements of L are too large.

(11) Reduce exploratory step sizes.

Another exploratory move follows.

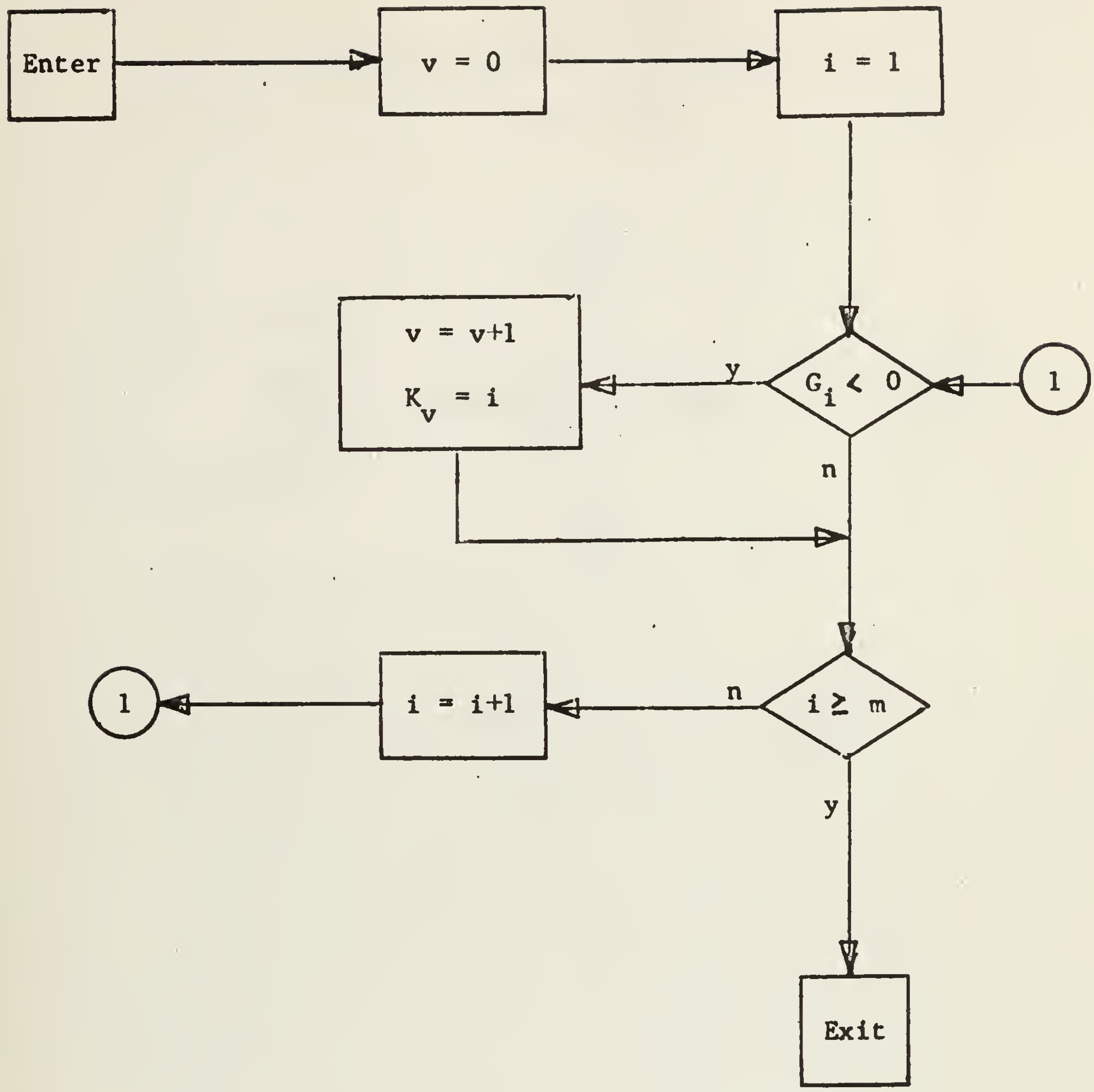


Chart 2: Internal Auxiliary Procedure V: G;v,K.

This procedure places a count of currently-violated constraints at v and tabulates their indices in column vector K.

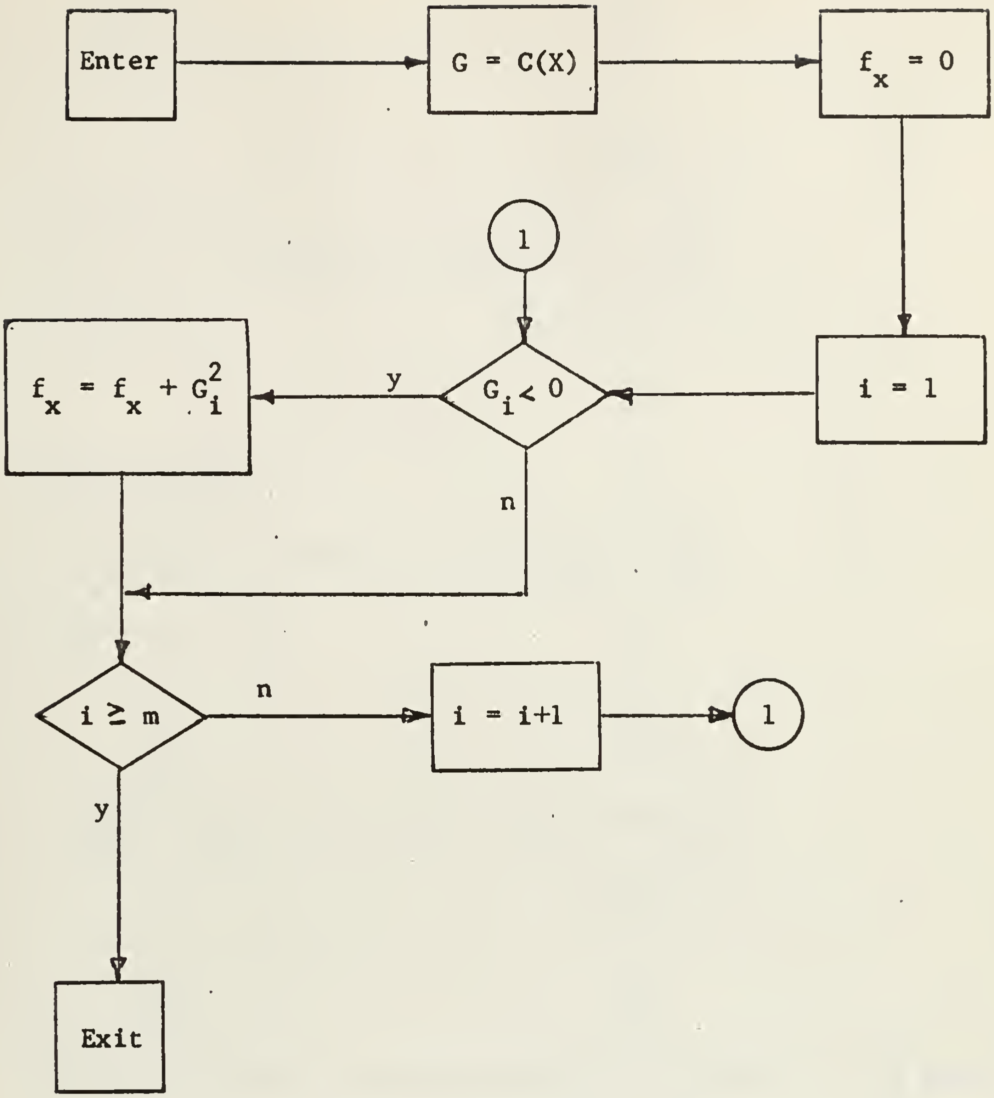


Chart 3: Internal Auxiliary Procedure $\underline{F_o: X,G;f_x}$.
Any values of X for which this function is zero is a feasible point.

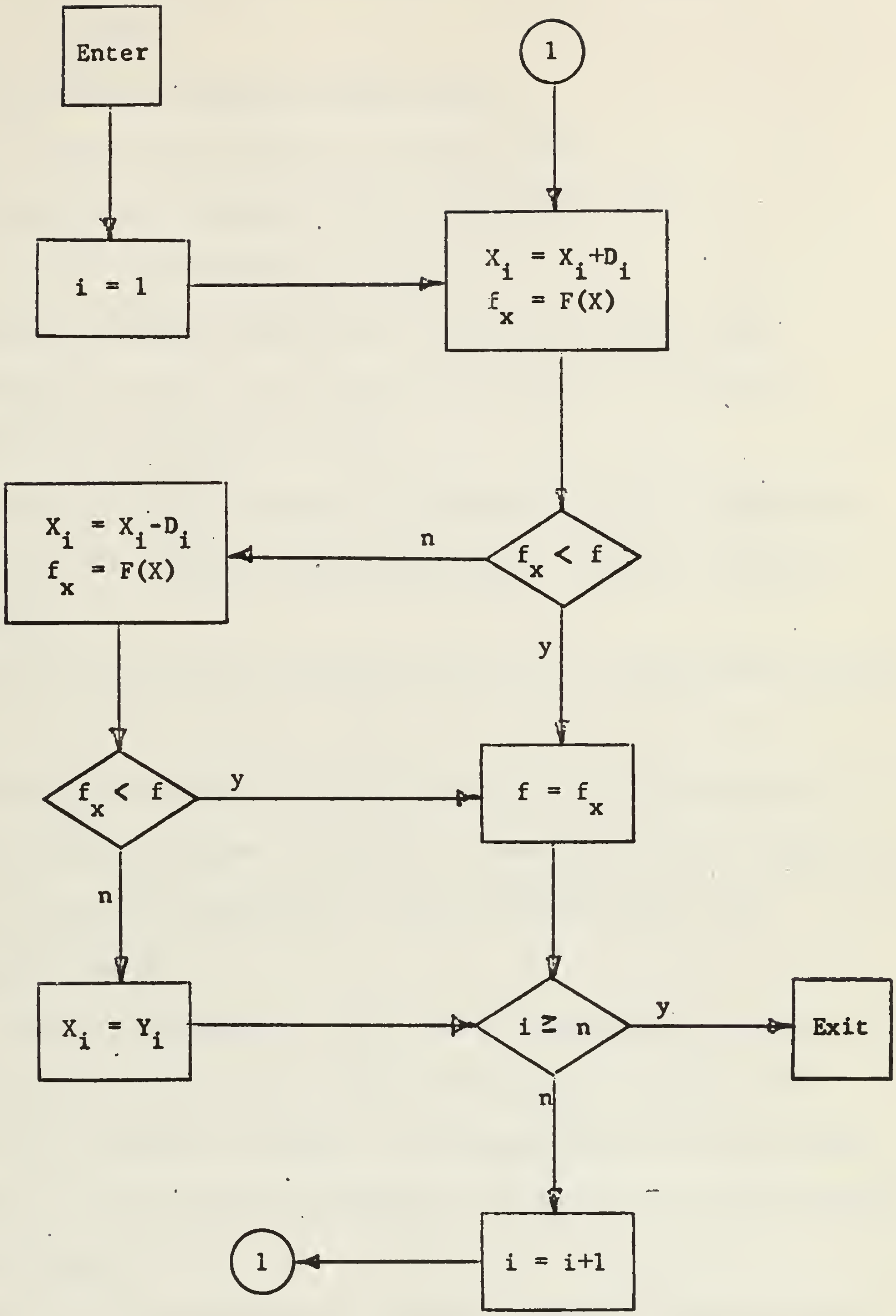


Chart 4: Internal Auxiliary Procedure E: Y,D; f,X.

This is the conventional exploratory algorithm of Hooke and Jeeves.

(12) Set new base point; make extended pattern move; evaluate special function.

(13) Is trial value of special function zero?

If so, pattern move arrived at feasible start.

(14) Is trial value of special function below base value?

If not, an exploratory move follows.

(15) Increase exploratory step sizes, subject to an upper limit.

A successful pattern move causes these steps to be extended. Another extended pattern move follows.

(16) Evaluate constraint functions at feasible start; set this point as first base; restore original step sizes; evaluate objective function.

Initialization for the constrained search is accomplished by these steps.

The preceding three charts show the details of three important internal auxiliary procedures. They do not seem sufficiently complicated to warrant inclusion of descriptive titles and additional notes.

Modified Direct Search

Chart 5 can be conveniently considered as representing the steps of the algorithm which are followed when the base point is not in the vicinity of any constraint boundary. The method shown is substantially Direct Search (4) with some modifications very similar to those proposed by Glass and Cooper (2).

It should be recalled that Direct Search assumes an initial feasible X, and the corresponding f is first computed. In order to explore, each X_i is increased by its D_i and if the corresponding f_x is below f, the trial X_i is adopted. If not, the effect of decreasing X_i by D_i is tried. If both $\pm D_i$ components fail to lower the objective

function, X_i is restored to the base value. The exploratory move tries all variables in this manner, and if the complete move produces an improved f, the vector between the base point and the successful exploratory move provides the direction for a trial pattern move and its reference length. The first pattern step in that direction is (1 + r) times this reference length. Accelerating moves are made in the same direction until some new trial value of f is greater than the base value, f_y.

When a pattern or exploratory move fails, every element of D is reduced by a multiplicity factor, $r < 1$. Eventually either a new successful pattern is found or else the lower limits, L, on all D are passed. When this happens the calculation is terminated.

The method of pattern move acceleration differs from Hooke and Jeeves. The present author believes that this variation usually reduces the required number of exploratory moves which are costly in terms of objective function evaluations. This conclusion was also reached by Glass and Cooper (2). Furthermore, it seems to follow logically from the following observation on page 216 of Hooke and Jeeves (4). "Typically a pattern once established will, through continuous modification, grow until the length of the pattern move is 10 to 100 times the basic step size." Any successful pattern move leads to a longer one in exactly the same direction; it also causes a corresponding increase in every element of D. However, upper limits on the value of D are imposed. These processes are also not part of the original Direct Search. They seem justified on the basis of the favorable results achieved.

The fundamental variations from the method of Hooke and Jeeves are, of course, inclusion of tests for constraint violations and the alternate exploratory procedure employed when such occur. Tests for violations are

made after each otherwise-successful exploratory and pattern move. Whenever one or more boundaries have been crossed, exit is made to the tangent exploratory procedure. This will be described in the later sections.

Finally, the jump move is proposed as a wholly new tactic in finding difficult solutions. If, at one or more constraint boundaries, the tangent exploratory procedure repeatedly fails to establish a successful direction (tangent move), a jump move is made to the best feasible trial of all components developed. That is, the base point is shifted to that component of the v′ tangent exploratory moves, just attempted by auxiliary procedure T, which is feasible and has the lowest corresponding value of the objective function.

The solution then proceeds with an initial reapplication of the tangent exploratory procedure. If repeated failures at one or more constraints recur, another jump move is made, provided that the intervening solution steps have produced a base point with function value lower than the value prior to the first jump move. If this has not happened, the subsequent potential jump move is bypassed. Instead, the base is returned to its position before the first jump move and a conventional exploratory move is attempted. More precise information on this feature will be given in the following charts and notes.

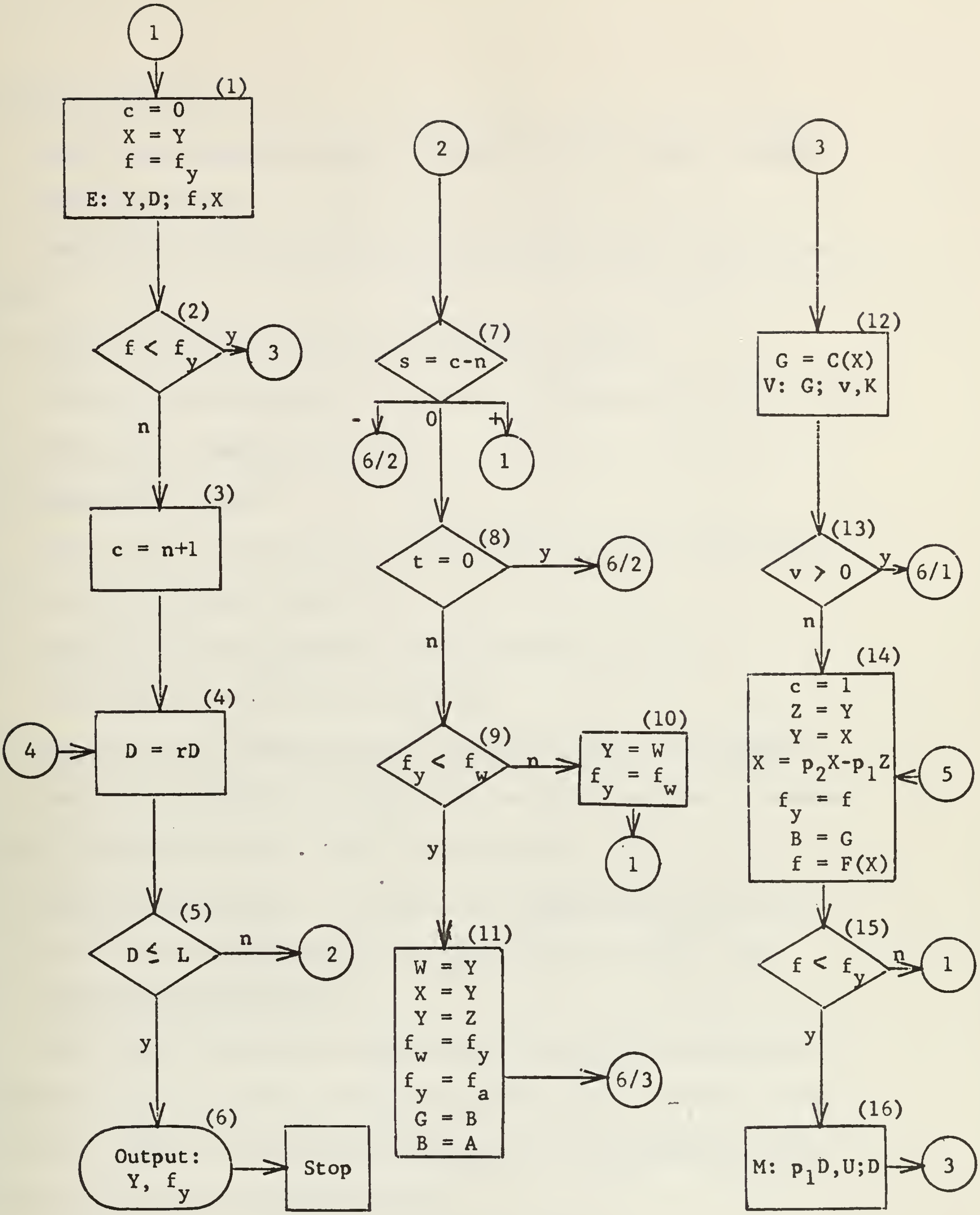


Chart 5: Modified Direct Search with Jump Move.
Main Procedure, Part II

Notes for Chart 5

(1) Set failure counter to zero; starting from base point, perform exploratory move.

The details of this conventional exploration were shown in Chart 4, above.

(2) Is trial objective function below value at base?

If so, constraints must next be checked for any possible violation.

(3) Set failure counter to n + 1.

(4) Reduce step sizes.

This is also the point of re-entry into the main procedure after failure of the tangent explore move.

(5) Are all step sizes less than lowest tolerated limit?

If so, the calculation is terminated.

(6) Output values of variables and objective function at minimum.

The accuracy obtained, provided the method is successful, depends upon the specified values of the L column matrix.

(7) Set and test switch s. If $s < 0$, try tangent exploration; if $s = 0$, test for existence of a feasible move; if $s > 0$, try conventional exploration.

Only after n consecutive applications of the tangent exploration algorithm fail to produce a satisfactory move does a jump move become possible. Switch s will be greater than zero only immediately after failure of a conventional exploration.

(8) Has a feasible jump move been stored by T?

If the previous application of auxiliary procedure T produced no feasible trial, it is re-entered.

(9) <u>Is the objective function at the present base below its value before the most recent jump move?</u>

This is a necessary condition for any subsequent jump move.

(10) <u>If not, restore previous base and try conventional exploratory move.</u>

(11) <u>If so, perform jump move; then go to recompute partial derivatives before tangent exploration.</u>

The present base values are saved, so that the test at Item (9) can be made the next time a jump move may be indicated. The best feasible trial from the most recent application of procedure T becomes the new base, and the present base is treated as a failed trial.

(12) <u>Compute constraint functions and check for violations.</u>

(13) <u>Is there any violation?</u>

If so, control is transferred to the preamble to the tangent exploratory procedure.

(14) <u>Perform extended pattern move.</u>

The failure counter is set to unity as an indicator; present values of the constraint functions are saved for possible use by auxiliary procedure H.

(15) <u>Is the trial value of the objective function below base value?</u>

If not, pattern move has failed and exploration is indicated.

(16) <u>Extend exploration step sizes.</u>

Whenever a pattern move succeeds in lowering the objective function the values of D are increased, subject to a limit.

<u>The Exploratory Move in the Tangent Hyperplane; Basic Concepts.</u>

Whenever at some trial point, X, a conventional exploratory or pattern move finds a value of the objective function lower than the base value, that trial move is immediately checked to determine whether or not it is

feasible. If not, control is transferred to the preamble of the tangent exploratory procedure, shown in Chart 6. This section will provide a general outline of tangent exploration, including ancillary logic such as the preamble and the calculation of partial derivatives of the constraint hyper-surfaces. Later sections will explain each sub-procedure in greater detail.

The internal auxiliary procedure H:v, B, G, K, X, Y; I, P approximates the partial derivative of each constraint function currently violated at the last feasible base, *i.e.*, the quantities;

$$\left| \frac{\partial g_j}{\partial x_i} \right|_Y , \quad (j = 1, 2, \ldots, v), \; (i = 1, 2, \ldots, n).$$

It also generates the matrix I which, after exit from H, contains a column of indices for each violated constraint. Each of these v columns contains n variable numbers arranged in order of ascending absolute value of the corresponding first partial derivatives. If the tangent exploratory move fails, auxiliary H is not re-entered on subsequent attempts to find a tangent move from that base. If, however, after n such failures a jump is made, H is used to re-estimate the appropriate derivatives before procedure T is re-entered.

If Y is the current base point in feasible n-space and the j-th constraint hyper-surface is crossed due to trial move X, the first essential concept of the Tangent Search Method is that the next exploration be made, if possible, in the hyperplane,

$$\sum_{i=1}^{n} \left| \frac{\partial g_j}{\partial x_i} \right|_Y (x_i - Y_i) = 0,$$

where x_i is the i-th independent variable of n-space.

Note that this hyperplane is only approximately tangent to the hypersurface, $g_j(x) = 0$, since it includes the point Y which, in general, is displaced some amount into feasible space. However, the discussion following will usually refer to it as the "tangent hyperplane". It would be so strictly only if the base lay directly upon the constraint boundary.

The second basic idea of the tangent exploratory scheme is that each variable, in turn, be incremented, while holding all others but one constant. That one is adjusted by the amount necessary to confine this trial component of the move to the tangent hyperplane. The a-th variable, incremented by the corresponding D_a, will be called the "primary" variable of a trial component of a tangent exploratory move. The b-th variable, which is that one adjusted to keep the trial in the hyperplane, will be called the "secondary" variable. Each such trial component, therefore, requires the two replacement operations:

$$X_a = X_a + D_a$$

$$X_b = X_b - \left| \frac{\partial g_j}{\partial x_a} \right|_Y \left| \frac{\partial x_b}{\partial g_j} \right|_Y \cdot D_a.$$

The primary index simply cycles through the first (n - 1) elements of the j-th column of the current I matrix, i.e., $a = I_{kj}$, (k=1, 2, ..., n-1). The corresponding b indices are selected by a rule which will be discussed in the section below treating the preamble to tangent exploration.

As in the conventional exploratory move, the objective function is evaluated for any trial component, the k-th, for example. If it is lower than any previous trial of the present move and if no constraint is violated, that component is adopted and the variable indexed by $I_{(k+1)j}$ becomes

primary.

Otherwise the reflection of that trial component is tested, i.e., the component generated by the replacements,

$$X_a = X_a - D_a, \qquad \text{and}$$

$$X_b = X_b + \left|\frac{\partial g_j}{\partial x_a}\right|_Y \left|\frac{\partial x_b}{\partial g_j}\right|_Y D_a.$$

On the first component this reflection is collinear with the original trial and the base point. On subsequent components it is collinear with the previous trial and the best move yet developed during the current exploration. If this trial component is feasible and produces a function value lower than the previous trial, it is adopted. Otherwise both the a-th and b-th variables are restored to their values before the former became primary, and the next primary variable is substituted. This process continues until (n - 1) primary variables have been selected.

Preamble to Tangent Exploratory Move.

Introductory steps before entrance into and testing procedures after exit from procedure T are shown in Chart 6.

If the violation which results in tangent exploration was caused by conventional exploration, step sizes are reduced in the preamble. However, if an attempted pattern move caused the violation, step sizes are unchanged. This policy was chosen solely on empirical grounds. Next, procedure H obtains estimates of the values at the base of the partial derivatives of each violated constraint function with respect to each independent variable. Then a loop is initiated which attempts to obtain one exploratory move for each presently-violated constraint.

It is now necessary to return to the subject of variable coupling during tangent exploration. The manner in which secondary variables are

selected is also partly shown in Chart 6. It should be recalled that the b-th variable is decremented by an approximation of the amount,

$$d_b = \left|\frac{\partial g_i}{\partial x_a}\right|_Y \left|\frac{\partial x_b}{\partial g_i}\right|_Y D_a,$$

when the a-th variable is primary for the exploratory move in a hyperplane approximately tangent to the hyper-surface, $g_j = 0$. The adjustment of X_b by d_b is designed to confine the trial move to that hyperplane when X_a is incremented by D_a.

In order to avoid possible computational difficulties arising from large disparities in magnitude between $\partial g_j / \partial x_a$ and $\partial g_j / \partial x_b$, it was decided simply to choose a and b such that

$$\left| \left| \partial g_j / \partial x_a \right|_Y \right| \geq \left| \left| \partial g_j / \partial x_b \right|_Y \right| .$$

Here the notation refers to the absolute values of the partial derivatives. Both are evaluated at base point Y. To facilitate this rule the matrix I is produced by auxiliary procedure H. It should be noted that the partial derivatives, stored in the matrix P, are correspondingly arranged at exit from H.

Many of the elements of P will be zero for practical problems. In fact, a column of P representing an explicit constraint (upper or lower bound on an independent variable) contains, at every violation, (n - 1) zeros, while the last element is always unity. Implicit and explicit constraints are not treated separately by the algorithm described here. It would probably be advantageous to follow the example of M. J. Box (3) and do so in future work.

When some element, P_{kj}, of matrix P is zero, the corresponding i-th variable, where $i = I_{kj}$, cannot be used either as a primary or secondary

[illegible]

$$[illegible]$$

[illegible]

$$[illegible]$$

[illegible]

[illegible]

[illegible]

variable in the sense previously defined. In that case the tangent hyperplane is parallel to the i-th coordinate axis. Such a variable will be called "passive" in the following discussion. Any variable, the q-th, where $q = I_{kj}$, for which the corresponding P_{kj} is not zero, will be called "active".

Since any increment (or decrement) to X_i cannot remove it from the tangent hyperplane, the trial step for any passive variable can be chosen as the corresponding element of the D vector. Also, if more than one element of P_{kj}, $(k = 1, 2, \ldots, n)$ are zero, any two such passive variables may be simultaneously incremented and/or decremented without departure from the tangent hyperplane. In that sense primary and secondary passive variables may also be chosen arbitrarily.

Therefore, in the selection of the variables for the trial components of the tangent exploratory move at the j-th constraint, the primary index can always become in turn,

$$a = I_{ij}, I_{2j}, \ldots, I_{(n-1)j}.$$

The variable of index I_{nj} is never primary. The corresponding element P_{nj} cannot be zero, since this would imply that all P_{kj}, $(k = 1, 2, \ldots, n)$ are zero, in which case the tangent hyperplane would be parallel to all coordinate axes. Therefore, it cannot be involved in coupling passive variables. Also, it is never used as the primary in coupling active variables, since it will have already been considered at least one time as a secondary variable by the time $a = I_{(n-1)j}$.

The manner in which the secondary variable index, b, is chosen will be termed the "coupling mode". At all times the j-th element of N contains the current coupling mode number to be next applied in calculating a trial move in the tangent hyperplane at the j-th constraint. If the

trial exploration results in a tangent move, the coupling mode is unchanged. Otherwise the present mode number is decreased by one. If all fail in turn, a total of 2(n-1) modes are tried to complete a cycle. If necessary it is then re-initiated.

The coupling constants, g and h, for each constraint are functions of the current corresponding coupling mode number. Constant g is used by procedure T to couple active variables, and h is used to couple passive variables. Further information will be given in the description of that auxiliary procedure. An example of a complete cycle of g and h is given in the notes to Items (8), (9), and (10) of Chart 6.

After each exit from procedure T, the result is tested. The first trial which has been found feasible by T and also produces a value of the objective function below the base value becomes a "tangent move". The base is transferred to this point, and the vector defined by that transfer becomes the information upon which a pattern move is attempted. Exit to that part of the main procedure immediately follows.

However, if after trials for all violations are complete and no tangent move is made, transfer of the base may still be performed by a jump move, provided certain previously-described conditions are met. Such a new base must be feasible, even though it may involve an increase in the objective function over that of the previous base. This transfer of base does not provide a promising vector for a pattern move. Instead, tangent exploration from the new base follows any jump move.

In analyzing Chart 6, it is important to note that every feasible trial component for all violated constraints is tested for corresponding objective function value. The best move of this set is preserved in Z, the function value in f_a, and constraint function values in G. Auxiliary procedure T indicates that at least one feasible trial was found by setting the feasible move indicator, t, to unity.

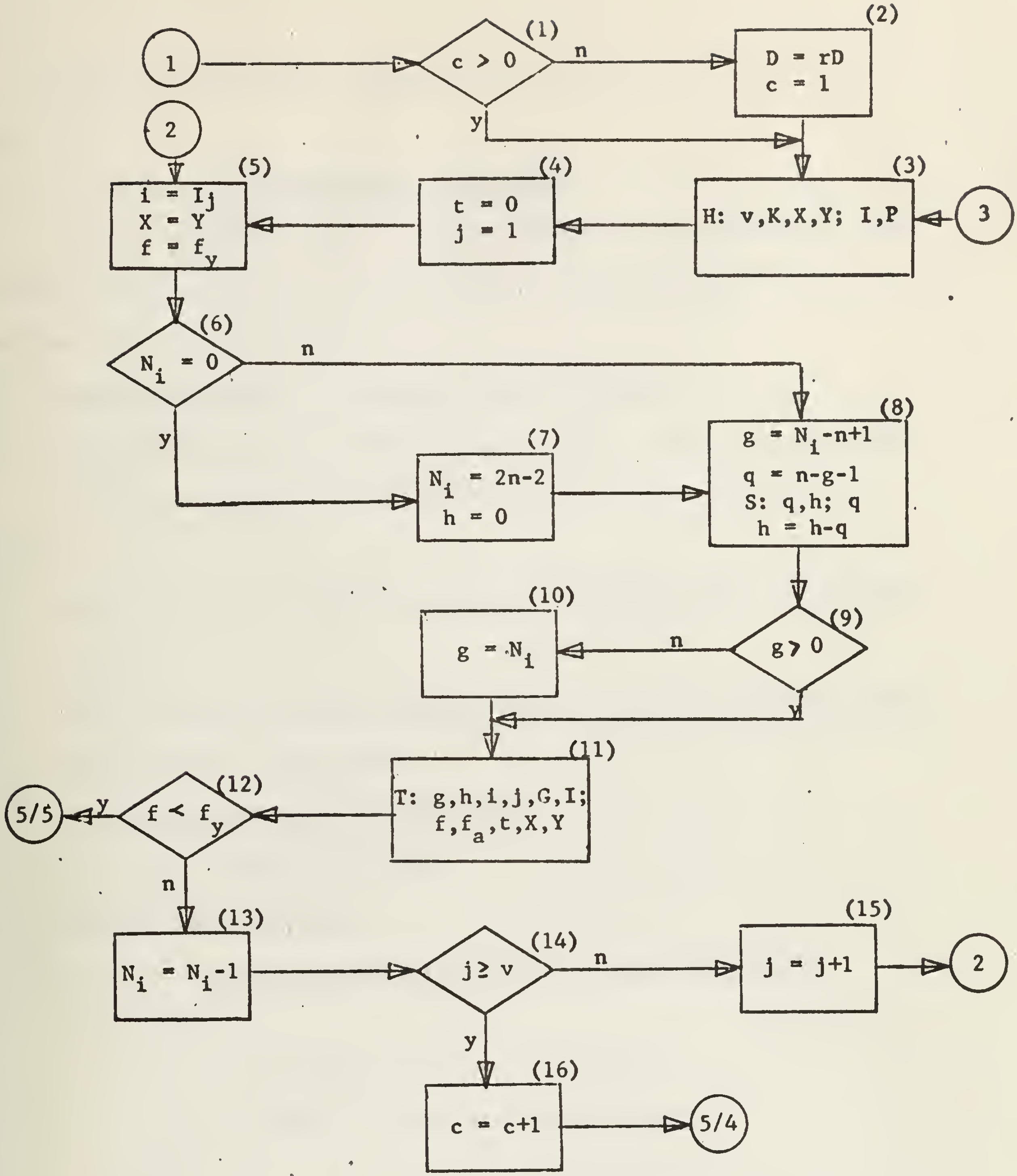
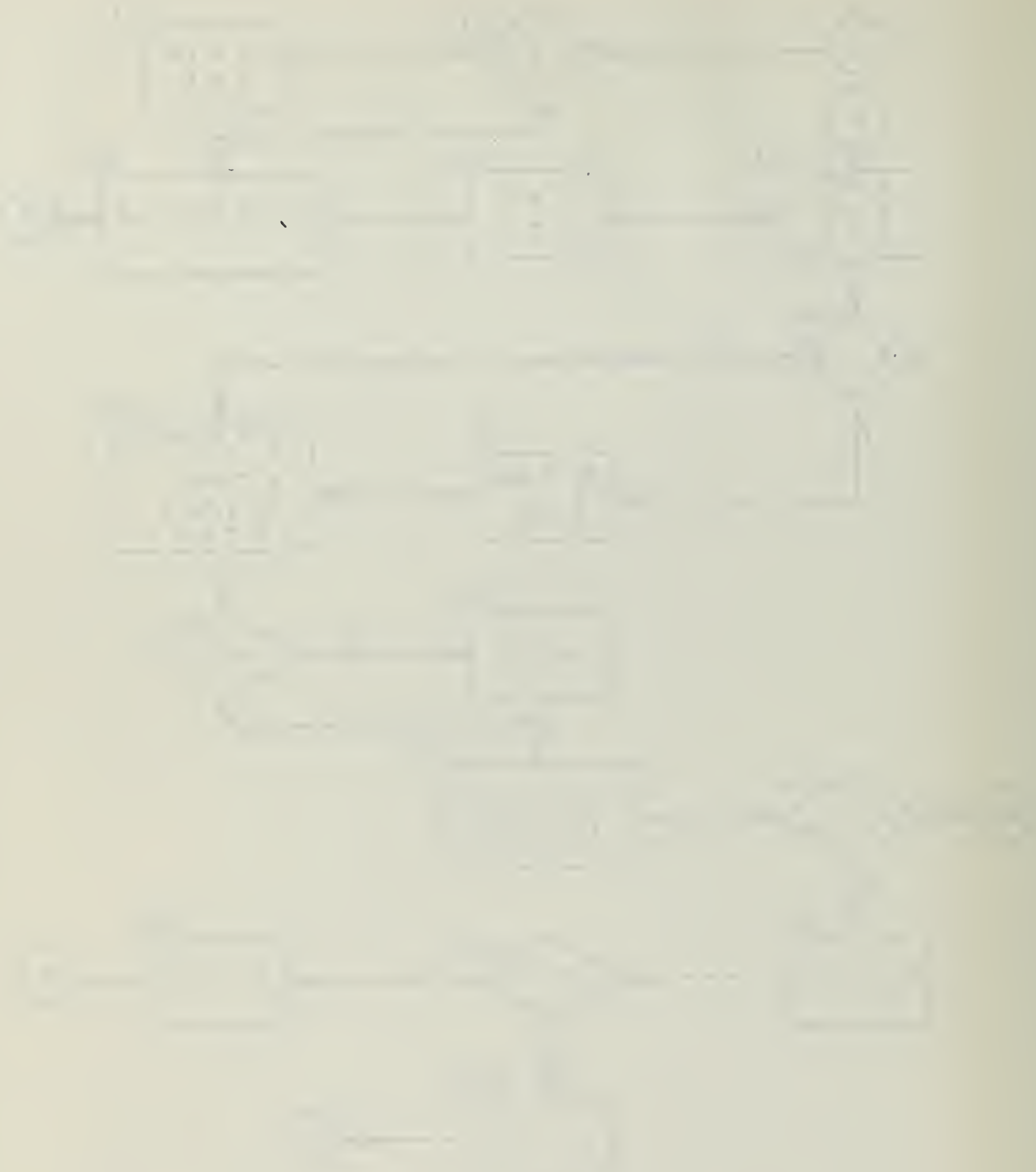


Chart 6: Preamble to a Series of Tangent Exploratory Moves and Tests of Results.

Main Procedure, Part III



Descriptive titles and further notes for the items of Chart 6 are given below.

(1) Is the failure counter greater than zero?

Here the counter is used to indicate whether the preamble has been entered after failure of an exploratory or a pattern move. If the former, explore step sizes are reduced.

(2) Reduce step sizes; set failure counter to unity.

(3) Internal auxiliary procedure H:v, B, G, K, X, Y; I, P is performed.

The detailed steps of this auxiliary are shown in Charts 9 and 10 below.

(4) Set feasible move indicator to zero; initialize loop to be indexed $j = 1\ (1)\ v$.

(5) Obtain index of the j-th violated constraint; reset trial variables and function to base values.

(6) Is present coupling mode number zero?

If so, cycle must be restarted.

(7) Initiate coupling cycle.

(8), (9), (10) Calculate active and passive coupling constants.

In all cases,

$g = N_i - n + 1$ or N_i, whichever is positive.

$h = h_o - q$, where h_o is the previous value of h, and

$|q| = n - g - 1$, to be prefixed by the sign of h_o.

Therefore, when n = 4, a complete cycle is:

Order	N_i	g	h
1st	6	3	0
2nd	5	2	-1
3rd	4	1	+1
4th	3	3	-2
5th	2	2	+2
6th	1	1	-3

The manner in which g and h are used to select secondary variables is described in the discussion of auxiliary procedure T below.

(11) Exploratory Move in Tangent Hyperplane.

The details of this internal auxiliary procedure are shown in Charts 7 and 8.

(12) Is trial function lower than value at base?

If so, an immediate pattern move is made and, in this process, the base is transferred to the tangent move just found.

(13) Decrease coupling mode number.

Since the mode used by procedure T at the i-th constraint did not produce a tangent move, the next attempt at that constraint will try the next coupling mode.

(14) Have explorations been tried in all hyperplanes tangent to presently-violated boundaries?

(15) Increment index and return to consider next constraint.

(16) Increment failure counter.

No tangent move has been found. Therefore, return to the main procedure is made to decrease step sizes, test for end of search and,

possibly, a jump move.

Special Expedient for Convex Constraint Hyper-surfaces.

It should be noted that convex curvature of a violated constraint hyper-surface is inherently troublesome for the algorithm described here; any move which were strictly confined to a hyperplane actually tangent to a convex constraint hyper-surface would cause immediate re-violation of that constraint.

Therefore, a special expedient is employed whenever any component of a trial exploratory move produces re-violation of the constraint currently considered. This open sub-procedure is located within the auxiliary procedure T. This problem was dealt with by Glass and Cooper (2) by their "extension" to the alternate move.

The method proposed here takes advantage of the fact that a measure of the gravity of a subsequent re-violation is provided at the same time it is detected. When, for the secondary variable, indexed by b,

$$p_b \approx \left| \partial g_j / \partial x_i \right|_Y \quad \text{and}$$

$G_j < 0$, after application of the external auxiliary C(X), the following simple replacement is used to adjust the trial move:

$$X_b = X_b - 2G_j / p_b.$$

A factor of 2 is used to over-compensate for the estimated convexity. If re-violation follows, this adjustment is again employed. However, if a third attempt fails, the present trial component is abandoned. Either the next primary variable is selected or the reflection is then tried.

The steps outlined above are appropriate only when the variables involved are active. If the primary variable is passive and not presently coupled, the variable of largest $\left| |\partial g_j / \partial x_i|_Y \right|$, (i = 1, 2, ..., n), is selected to serve as the secondary variable for this special expedient.

The index of that variable is the element I_{nj}.

Finally, if two passive variables are coupled and re-violation of the currently-considered constraint occurs on any trial component, the secondary variable of that component becomes the k-th where

$$k = I_{nj}.$$

Then all calculations for the component are restarted. Detailed steps of this expedient are shown in Charts 7 and 8 below.

Tangent Exploratory Auxiliary Procedure.

Input parameters of this procedure, T, include g and h, the coupling mode constants of the current trial move; i and j, indices of the constraint hyper-surface to be considered; the column I_{kj}, $(k = 1, 2, \ldots, n)$, which holds indices of variables ordered by magnitude of the corresponding P_{kj}, $(k = 1, \ldots, n)$; and the column matrix G, containing the constraint function values at X, where one or more constraints are violated.

This internal auxiliary procedure usually generates values for t, f_a, A, and Z, all of which are utilized only if no tangent move is found for any one of the currently-violated constraints. Values of f and X, produced by T, constitute a tangent move if and only if f lies below f_y, the value of the objective function at the base point, Y.

An introduction to the steps of this procedure has been given in various previous sections. However, it is still necessary to state the rules by which g and h are used to select secondary variables. If g is the coupling mode constant for active variables and $a = I_{kj}$, then b is chosen as the $(k + g)$th index of the j-th column of I, except that the index at I_{nj} pre-empts in all cases when $k + g > n$.

Similarly, if h is the coupling mode constant for passive variables and $a = I_{kj}$, then b is chosen as the w-th index of the j-th column

whenever $w = k + |h| \leq n-1$. If $w > n-1$ or if $P_{wj} \neq 0$, no coupling is effected. When the increment/decrement of the b-th variable is actually used, the sign of this operations is dictated by the sign of h.

As an example, we shall assume that $n = 4$, $m = 3$, $v = 2$, $K_1 = 2$, $K_2 = 3$, and

$$I = \begin{vmatrix} 3 & 1 \\ 1 & 2 \\ 4 & 4 \\ 2 & 3 \end{vmatrix} , \qquad P = \begin{vmatrix} .1 & 0 \\ -.5 & 0 \\ 7.1 & 0 \\ -9.8 & 1 \end{vmatrix} .$$

Then the variables involved in the trial components of the tangent exploratory move near the hyper-surface, $g_2(X) = 0$, would be, for the first three coupling modes:

N_i	g	a	b
6	3	3	2
		1	2
		4	2
5	2	3	4
		1	2
		4	2
4	1	3	1
		1	4
		4	2

The last three modes of the active coupling cycle repeat the above table. Since no element of the first column of P is zero, only active coupling is used at this constraint. The variable coupling used in trial components near $g_3(X) = 0$ is shown in the table below. This constraint boundary is actually a hyperplane, perpendicular to the 3rd coordinate axis. Therefore, all coupling is passive.

N_i	h	a	b	Remark
6	0	No coupling		a = 1, 2, 4
5	-1	1 2 4	2 4 None	Increments/Decrements opposite in sign
4	+1	1 2 4	2 4 None	Increments/Decrements the same in sign
3	-2	1 2 4	4 None None	Increments/Decrements opposite in sign
2	+2	1 2 4	4 None None	Increments/Decrements same in sign
1	-3	No coupling		a = 1, 2, 4

Detailed steps of procedure T are shown in Charts 7 and 8. Descriptive titles for the items given and additional explanatory notes follow the charts.

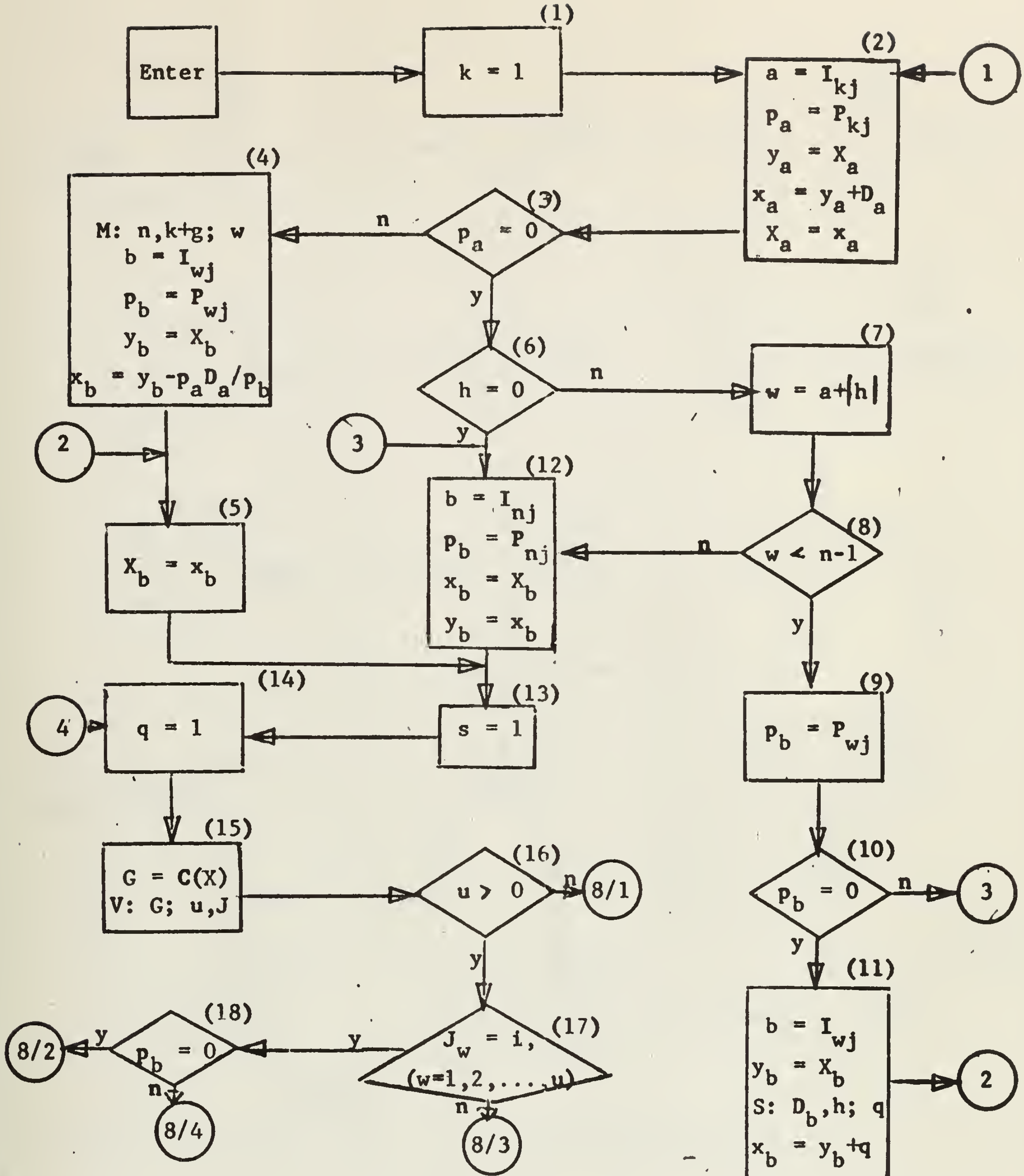


Chart 7: Internal Auxiliary Procedure T: g,h,i,j,G,I; f,f_a,t,A,X,Z.
The Tangent Exploratory Move, Part I

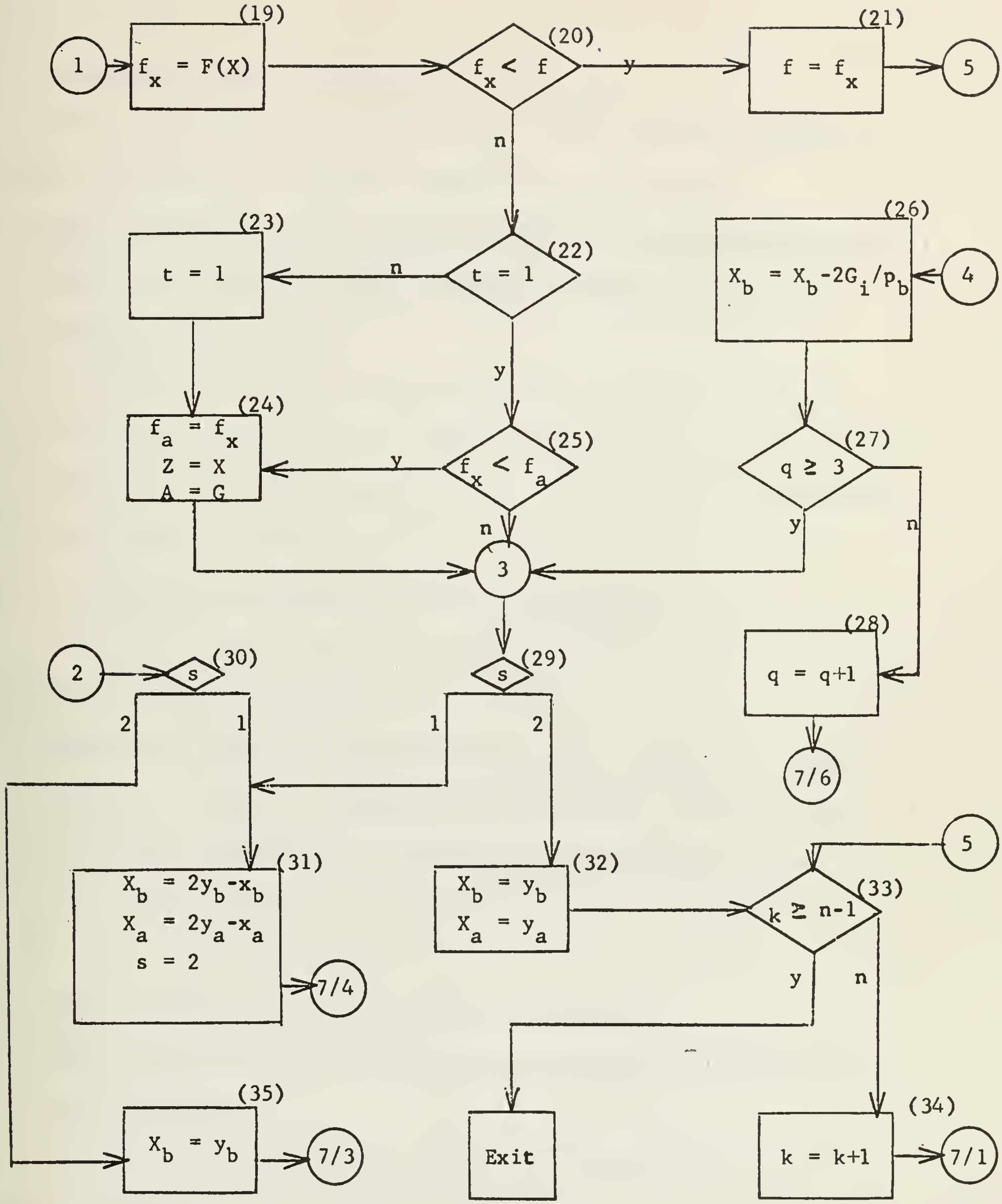


Chart 8: Internal Auxiliary Procedure T: g,h,i,j,G,I; f,f_a,t,A,X,Z.
The Tangent Exploratory Move, Part II

Charts 7 and 8

(1) Initialize index of outer loop, $k = 1(1)n-1$.

Index k will pick up, in order, the variable numbers of the n-1 primary variables of the present tangent exploratory move.

(2) Select index of primary variable; select the corresponding partial derivative; save old value of primary; increment and save new primary variable value.

The values of p_a, y_a, and x_a are preserved separately from the matrices in which they are found. This policy reduces indexing steps in an object program. It also simplifies the calculation of the component reflection, if this becomes necessary.

(3) Is the primary partial derivative, $\partial g_j / \partial x_a$ zero?

If so, any change in the a-th variable would be parallel to the tangent hyperplane and, therefore, it is passive.

(4) Select the index, b, for the secondary active variable by using constant g; select secondary partial derivative; save old value of the active secondary; save new decremented variable value.

Values of p_b, y_b, and x_b are also preserved for later use in calculating a trial reflection.

(5) Change effective value of secondary variable, X_b.

(6) Does coupling constant indicate that the primary passive variable is to be coupled?

If so, it will be necessary to select the appropriate index of a passive variable.

(7) Compute tentative location of the secondary index.

(8) Is location within bounds?

If so, it is still necessary to discover whether or not the tentatively-selected variable is also passive.

(9) Save the partial derivative of tentative secondary passive variable.

(10) Is the partial derivative zero?

If not, coupling is not allowed since any change in one active variable would cause departure from hyperplane.

(11) Select secondary passive index, b, from I matrix; save old value of X_b; compute incremented or decremented secondary variable.

The direction of that adjustment depends upon the present sign of the coupling constant.

(12) Select variable with largest absolute partial derivative as secondary; save value of derivative and old value of variable; maintain this value for trial.

These steps allow the reflection/reset logic to handle non-coupled components without special tests. They also provide a secondary variable for possible use in the special expedient for convex surfaces.

(13) Set reflection/reset switch to unity.

This switch is tested whenever a trial component is non-feasible or fails to improve the objective function value. When s is unity, the reflection of the present trial component is next tested; if s = 2, the values of the variables which existed prior to the failed component are reset.

(14) Initialize loop to count possible applications of the special expedient at a convex hyper-surface.

This sub-procedure is tried a maximum of three times consecutively before a trial component is abandoned.

(15) Compute constraint functions and test for violations.

(16) Is any constraint violated?

If not, exit from the special expedient loop is made. In that case the next step is to test whether or not this feasible trial component brings an improvement in the objective function value.

(17) Is the presently-considered constraint re-violated by this trial component?

If not, the special expedient is useless and this trial component has failed.

(18) Is the secondary variable passive?

If so, it cannot be used in the logic of the special expedient. Therefore, exit is made in order to try the reflection or to select a different secondary variable.

If not, the secondary variable is adjusted by approximately twice the minimum amount necessary to produce a feasible trial. These steps are discussed at Item (26), etc., below.

(19) Calculate the value of the objective function corresponding to the present trial component.

(20) and (21) Is this value an improvement over the best previous result at this constraint?

If so, it is saved, the variables remain undisturbed, and the next primary variable is selected.

(22) Has any feasible move been previously found at any presently-violated constraint?

If not, this component will be preserved.

(23) Set feasible move indicator to unity.

(24) Save present values of objective function, independent variables, and constraint functions as best feasible move so far.

(25) Test objective function.

The function value at the present trial component is compared with the value for the best feasible component so far at all constraint surfaces currently violated.

If it is below the previous best value the appropriate quantities are preserved for a possible jump move. If not, the reflection/reset switch is tested to determine the manner of finding the next trial component. See Item (29), below.

(26) Correct secondary variable.

(27) and (28) Have three attempts to apply the special expedient been made?

(29) and (30) Test reflection/reset switch.

If a trial component fails, its reflection in the hyperplane about the previous best reference is tried. If a reflection fails, variables are reset to the previous reference.

The switch test at Item (30) is made after failure of the special expedient.

(31) Compute reflection of trial component in hyperplane about previous best point; set reflection/reset switch to 2.

It should be noted that the reflection of a trial which is modified by the special expedient is not used. Control next returns to the feasibility test.

(32) Reset primary and secondary variables to previous best values.

Both the original and reflection components for this pair of variables have failed.

(33) and (34) Have (n-1) primary variables been tried?

If so, this application of T is complete. If not, k is

incremented and the next primary variable is selected.

(35) Reset secondary variable.

If both components fail with passive coupling due to convexity of the constraint hyper-surface, a new secondary variable, which is necessarily active, is selected.

Calculation of Partial Derivatives of the Constraint Functions.

In the material above discussion of the problem of evaluating the partial derivatives,

$$\left|\frac{\partial g_j}{\partial x_i}\right|_Y, \quad (j = 1, 2, \ldots, v), \ (i = 1, 2, \ldots, n),$$

has been bypassed. In Chart 6 this process is shown as an internal auxiliary procedure

H: v, B, G, K, X, Y; I, P ,

which is used as part of the preamble to a series of tangent exploratory moves. The details of one formulation of that auxiliary are shown in Charts 9 and 10. This is the algorithm used in the test calculations to be discussed in the following sections.

However, there are many computational alternatives for accomplishing the same purposes, one or more of which may be more elegant and economical than that given in Charts 9 and 10. The last section of this paper will return to this point.

Five major presuppositions led to the particular version of procedure H which is described here.

1. Only one external auxiliary procedure to define all constraint functions should be allowed. All such functions should be evaluated each time this auxiliary, C(X), is called.

The author has visualized C(X) as a subroutine to be supplied by

the user of a standard library routine which contains all steps except F(X) and C(X). For this reason complete simplicity of logic, e.g., no conditional branching, within both of these external auxiliaries has been an overriding desideratum.

2. No special external auxiliaries to calculate the derivatives, themselves, either by difference methods or by the evaluation of formulas obtained by analytic differentiation, should be allowed.

The present author felt that the entire task of obtaining derivatives should be shouldered by any proposed minimization method. This is particularly important for any method seriously suggested for the solution of practical problems.

3. Special software or subroutines to obtain derivatives by the methods of Wengert (5) or Smith (6) should not be assumed.

4. Implicit and explicit constraints would not be separately treated.

5. The minimum number of evaluations of F(X) and, expecially, C(X) in obtaining satisfactory solutions would be a primary goal.

In the context of these presuppositions it was decided to approximate the desired partial derivatives by simple first-order differences. In other words the secant is actually substituted for the tangent.

When procedure H is called, one or more tangent exploratory moves are about to be tried. The actual number is, at most, v, the number of constraints violated by the last trial move. The point Y is the present base, to which corresponds the column matrix of constraint function values, B. All elements of B are zero or positive. The point X is the non-feasible trial move, to which corresponds the column matrix, G, v elements of which are negative. Additionally, K is a column matrix of v elements, each of which is the index number of a violated constraint.

In the following discussion attention will be restricted to the approximation of

$$\frac{\partial g_k}{\partial X_i}, \quad (i = 1, 2, \ldots, n),$$

where k is any one of the indices stored in K, and g_k is the k-th constraint function. It should be understood that, each time it is called, H produces all elements of the matrix, P, which contains n rows and v columns.

The total differential of g_k can be expressed,

$$dg_k = \sum_{i=1}^{n} \left(\frac{\partial g_k}{\partial x_i} \right) dx_i,$$

where the x_i, $(i = 1, 2, \ldots, n)$ are general independent variables. Evaluation of the partial derivatives is desired either at the base point or nearby, for example, at some point between the current base, Y, and the non-feasible trial, X. The differentials above are approximated by simple differences,

$$dx_i \approx X_i - Y_i, \qquad (i = 1, 2, \ldots, n),$$

and

$$dg_k \approx G_k - B_k.$$

It is, in general, necessary to evaluate only (n-1) of the partial derivatives, $\partial g_k / \partial x_i$, $(i = 1, 2, \ldots, n)$, since the n-th can be found from,

$$\frac{\partial g_k}{\partial x_n} = dg_k - \left[\sum_{i=1}^{n-1} \left(\frac{\partial g_k}{\partial x_i} \; dx_i \right) \right] \Big/ dx_n.$$

[illegible]

$$[illegible]$$

[illegible]

$$[illegible]$$

[illegible]

$$[illegible]$$

[illegible]

$$[illegible]$$

One call of the external auxiliary procedure C(X) is necessary to calculate each $\partial g_k/\partial x_i$ for i = 1(1)n-1. These values are then approximated by the simple differences,

$$\partial g_k/\partial x_i \approx (A_k - B_k) / (X_i - Y_i),$$

where A_k is g_k (W).

The point W is defined,

$$W_q = Y_q, \text{ for all } q \neq i,$$

and

$$W_i = X_i.$$

The method described above would fail whenever any $X_i = Y_i$ When this happens W_i is taken as

$$W_i = X_i + D_i,$$

in the evaluation of A_k.

In the notation of the flow charts

$$\left| \partial g_k/\partial x_i \right|_Y \approx P_{ik}.$$

Auxiliary procedure H generates matrix P containing n rows and v columns of such approximate partial derivatives corresponding to the n variables and v violated constraints. At exit from H each column of P is arranged in ascending order of absolute value. The procedure also generates matrix I which contains the variable numbers corresponding to the arrangement of P. The sorting which is necessary to order P and I is accomplished by the simple exchange method.

The steps which calculate the elements of matrix P are given in Chart 9. The rearrangement of P and generation of I are shown in Chart 10. The author has omitted the usual listing of descriptive titles for these charts, since there are no difficult points requiring more detailed explanation. It should be mentioned that the control, e, and switch s are used for

branching when any increment, $Z_i = (X_i - Y_i)$, is near zero and to eliminate the unnecessary use of C(X), as explained above.

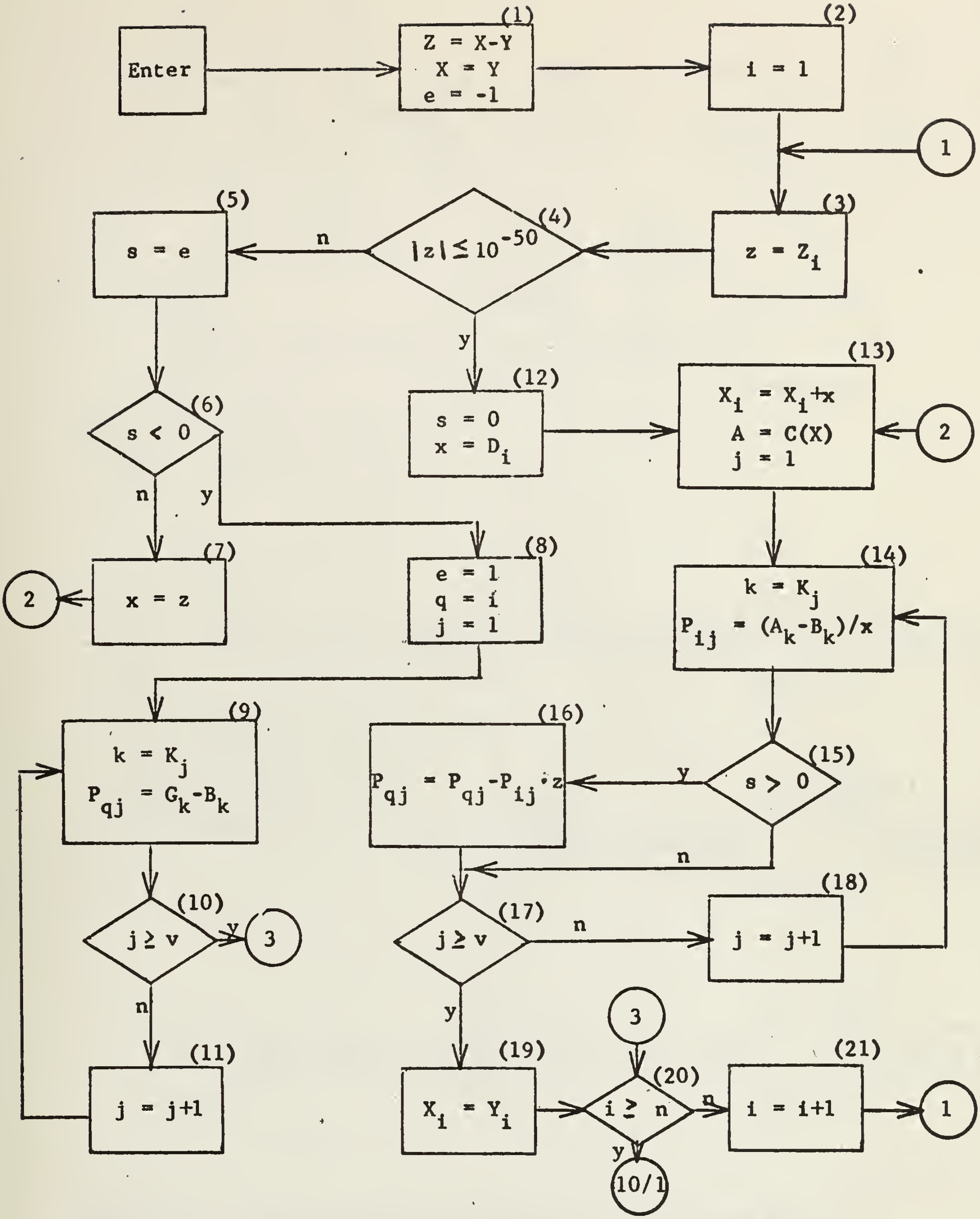


Chart 9: Internal Auxiliary Procedure H: v,B,G,K,X,Y; I,P.

Evaluation of Partial Derivatives of Constraint Functions, Part I

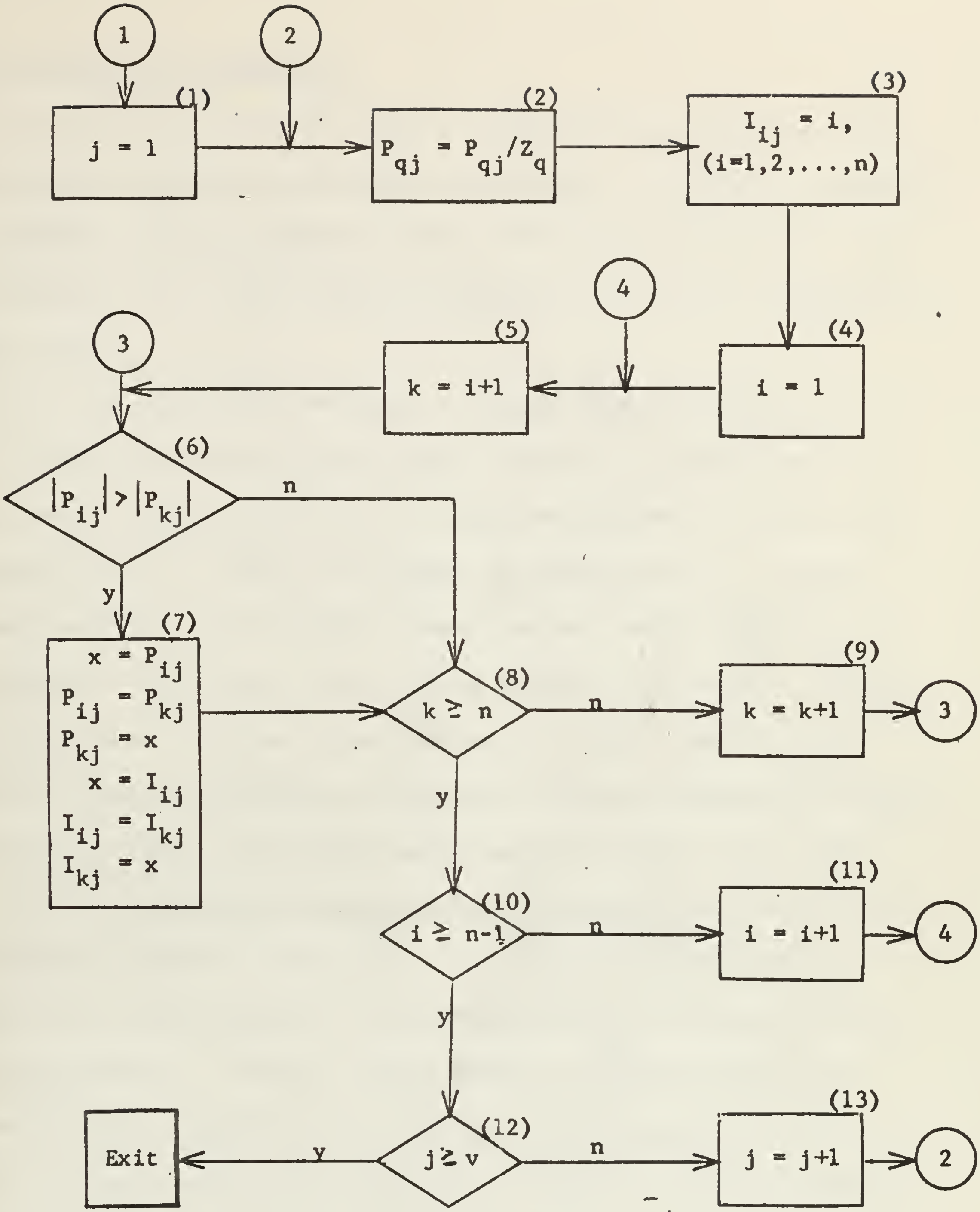


Chart 10: Internal Auxiliary Procedure H: v,B,G,K,X,Y; I,P.
Evaluation of Partial Derivatives of Constraint Functions, Part II

Numerical Results and Comparisons

The new calculations reported here were performed on the Control Data Corporation 1604 Computer of the Computer Facility, U. S. Naval Postgraduate School, Monterey. Both the Tangent Search Method described above and the Complex Method of M. J. Box (3) were programmed by the author in FORTRAN 60 for these trials.

Comparisons between other methods and Tangent Search are made on the basis of criteria used by previous authors. Counts of required moves are given for comparison with the results of Glass and Cooper (2), and for the one problem reported in detail by Klingman and Himmelblau (1). However, most comparisons are between Tangent Search and the Complex Method. All eleven examples were run with both of these methods by the author. These comparisons are made in terms of the numbers of function and constraint evaluations, tabulated separately, required to attain a constrained minimum of a defined accuracy. The present author wants to concur in the remark of M. J. Box (3) concerning the superiority of this measure. However for Tangent Searches counts of base points, conventional exploratory moves, and tangent exploratory moves are also tabulated below for possible help to other investigators. For the Complex Method, total and permissible trials are counted in addition to evaluations of the external auxiliary procedures. These statistics are used by M. J. Box. It is the interpretation of the present author that a trial vertex is "permissible" only if it both is feasible and also produces a value of the objective function which is no longer worst.

Each trial problem will be defined below and results tabulated. A summary of the results is given subsequently.

PROBLEM 1

This example is taken from the paper of Glass and Cooper (2). Their Table 1 shows the solution as developed by Sequential Search.

Required is the minimum of

$$F(x) = -\sqrt{21 - (x_1 - 5)^2 - (x_2 - 5)^2},$$

subject to the constraints,

$$G_1(x) = x_1^2 - 4x_2 \geq 0, \qquad \text{and}$$

$$G_2(x) = (x_2 - 6)^2 - 4(x_1 - 3) \geq 0 .$$

All trial runs were calculated from the initial point,

$x_1 = 7.000$

$x_2 = 1.000$

$F = -2.236$, with

initial step sizes,

$D_1 = .0666667$ and

$D_2 = .1333333$.

According to the article the actual solution to the problem is

$x_{1s} = 4.$

$x_{2s} = 4.$

$F_s = -4.7958 \ldots$

Sequential Search attained a result within a circle of radius .0015 about the true solution as its 23rd base point. Tangent Search solutions were also obtained to the same accuracy for three choices of step reduction factor, r. Comparison of all these results is given in Table Ia.

TABLE Ia

End criterion: $\sqrt{(x_1 - 4)^2 + (x_2 - 4)^2} \leq .0015$

Result	Algorithm			
	Seq. Search	Tangent Search		
		r =		
		.25	.375	.500
x_1 x_2 F	4.000 3.999 -4.796	4.0005 3.9994 -4.7958	3.9994 3.9988 -4.7955	3.9992 4.0008 -4.7958
Base Points	23	21	31	32
Explore Moves	12	5	7	11
Alternate Moves*	5	6	12	19

* For the purposes of Tangent Search this counter was incremented before any entry to Item 5 of Chart 6.

Results with Sequential or Tangent Search seem substantially the same for this problem. The best run is for Tangent Search with r = .25 . However with larger factors this method attains the required minimum less efficiently than Sequential Search.

Problem 1 was also solved using the Complex Method of M. J. Box (3) . Initial conditions were, as before, $x_1 = 7.0$ and $x_2 = 1.0$. Two runs were made in order to discover, to a slight extent, the effect of the random choices of vertices made at the beginning of each calculation. The sequence of pseudo-random numbers, used in setting up the first complex, varies between the two solutions. These are denoted "a" and "b". To compare

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

[illegible]

performance with Tangent Search the number of functions and constraint evaluations were counted. For the Complex Method the centroid of the vertices is the reference point for determining the progress of the solution. Comparison is made for the same degree of accuracy described previously; results are given in Table Ib.

TABLE Ib

End Criterion: $\sqrt{(x_1 - 4)^2 + (x_2 - 4)^2} \leq .0015$

Algorithm	Number of Evaluations	
Complex Method	Constraints	Obj. Function
a	191	109
b	128	64
Tangent Search		
r = .125	42	38
r = .375	63	72
r = .500	81	85

These results seem to indicate that Tangent Search is superior to the Complex Method for this problem.

PROBLEM 2

This is the second illustration of Glass and Cooper. The problem is to minimize the objective function of Problem 1, above, but now subject to the single constraint,

$$G(x) = 32-4x_1 - x_2^2 \geq 0.$$

The initial point and step sizes are, as before,

$x_1 = 7.0$. $\qquad D_1 = .0666667,$

$x_2 = 1.0$, $\qquad D_2 = .1333333.$

The solution (to four decimal places) is said to be:

$$x_{1s} = 4.3741$$

$$x_{2s} = 3.8083$$

$$F_s = -4.8154 .$$

Table 2 of that reference shows the steps generated by Sequential Search until a base approximately .004 units from the true result is found. Previous base points were actually closer than this. However none of these was chosen as final since subsequent results diverged. Comparisons between that result and corresponding tests using Tangential Search are given in Table IIa.

TABLE IIa

End Criterion: $\sqrt{(x_1 - x_{1s})^2 + (x_2 - x_{2s})^2} \leq .004$

Result	Algorithm			
	Seq. Search	Tangent Search		
		r = .25	.375	.500
x_1 x_2 F	4.3764 3.8071 -4.8154	4.3741 3.8081 -4.8154	4.3741 3.8083 -4.8154	4.3741 3.8082 -4.8154
Base Points	47	36	32	36
Explore Moves	20	10	13	15
Alternate Moves	15	14	15	21

Since two runs confirm the conclusion, it seems that Tangent Search is better than Sequential Search for Problem 2. Either of the two smaller values of r seems to be the proper choice in this case.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Trial calculations were also made, as on Problem 1, with the Complex Method. These results are shown in Table IIb.

TABLE IIb

End Criterion: $\sqrt{(x_1 - x_{1s})^2 + (x_2 - x_{2s})^2} \leq .004$

Algorithm	Number of Evaluations	
Complex Method	Constraints	Obj. Function
a	132	68
b	122	65
Tangent Search		
r = .250	74	81
r = .375	72	84
r = .500	91	101

The author feels this table shows substantially similar performance for the two methods.

PROBLEM 3

This example is taken from the paper of Klingman and Himmelblau (1); however, it originated in another article by R. A. Mugele where it demonstrated the IBM Probe Method of Optimization. A detailed narrative of the progress of the solution by the Multiple-Gradient Summation Technique is provided by the Appendix to the former paper. It is required to locate the optimum of

$$F(x) = 1/ \left((x_1 + 1)^2 + x_2^2\right),$$

under the two constraints,

$$G_1 (x) = x_1^2 + x_2^2 - 4 \geq 0$$

$$G_2 (x) = 16 - x_1^2 - x_2^2 \geq 0.$$

To prevent possible confusion, this problem will be immediately re-cast to require the minimum of

$$F(x) = -1/ ((x_1 + 1)^2 + x_2^2),$$

subject to the given constraints.

Initial conditions for trial runs are,

$x_1 = 5,$ $x_2 = 4$, and

$D_1 = 1.5,$ $D_2 = 1.2$.

Since $G_2(x) = -25$ for this origin, the solution must begin by searching for some feasible point. Both the Multiple Gradient Summation and Tangent Search Methods include a preliminary procedure for this purpose. A comparison of results is provided by Table IIIa. Moves required to find an initial feasible point and subsequent performance criteria are tabulated. In order accurately to compare results the constrained minimization phases of all three Tangent Search Calculations were started at the point (2.8, 2.0) at which all counts were reset to zero. Final tabulations are made for the point in each calculation which corresponds in accuracy to the last point reported in the Appendix of (1). Specifically this is the point at which the base enters a circle of radius less than .52 about the true solution,

$$x_{1s} = -2.0$$

$$x_{1s} = 0$$

$$F_s = -1.0$$

Evaluation of analytic first partial derivatives are also counted for Multiple-Gradient Summation. Other criteria are as described previously.

TABLE IIIa

End Criterion: $\sqrt{(x_1 + 2)^2 + x_2^2} \leq .52$

Criterion		Algorithm			
Initial Feasible Pt.		M-G Summation	Tangent Search r = .250	Tangent Search r = .375	Tangent Search r = .500
	x_1	2.8	1.625	3.500	-2.125
	x_2	2.0	1.300	1.150	-1.700
	F	-.0594	-.1165	-.0464	-.2406
	Base Points	1	2	3	3
	Explore Moves	1	1	2	1
	Constraint Evaluations	6	7	15	14
Final Point	x_1	-2.49	-2.118	-2.104	-2.131
	x_2	-.182	.4699	-.3347	.4960
	F	-.4005	-.6799	-.7513	-.6554
Moves	Base	7	15	8	8
	Explore	10	5	4	6
	Alt.	3	5	4	6
Evaluations	Constraints	43	31	20	28
	Obj. Funct.	33	43	29	43
	Part. Deriv.	6	---	---	---

Inspection of Table IIIa does not seem to establish any superiority between the methods. Among step reduction factors for Tangent Search, r = .375 seems best. With that choice the final result is better than that of Multiple-Gradient Summation. That method seems to do better in the initial search for a feasible origin; however the author feels this is due to a particularly fortuitous initial pattern move.

Comparisons between Tangent Search and the Complex Method are given in Table IIIb. The end criterion is smaller for these calculations.

TABLE IIIb

End Criterion: $\sqrt{(x_1 + 2)^2 + x_2^2} \leq .005$

Algorithm		Evaluations		Trials/Moves		
Complex Meth.		Constraints	Obj. Funct.	Total	Permissible	
	a	295	223	287	180	
	b	-	-	failed	-	
Tangent Search				Base	Explore	Alt.
	r = .250	68	87	33	11	13
	r = .375	71	86	30	10	16
	r = .500	75	101	25	15	19

Using the sequence of pseudo-random numbers designated "b", the Complex algorithm failed to develop a feasible fourth vertex at the start. This was due to the fact that both the random trial vertex and the centroid of the three previously-chosen vertices were not feasible. Both lay within the inner boundary,

$$G_1(x) = x_1^2 + x_2^2 - 4 = 0.$$

This difficulty may indicate an undesirable restriction on the use of that method in some practical problems. The author feels that this failure, with the poor result for run "a", indicates the superiority of a pattern search for Problem 3. Finally, the results given for Tangent Search in Table IIIb are better for the two smaller step reduction factors considered.

PROBLEM 4

This example is based upon the "banana-shaped valley" proposed originally by Rosenbrock (7). A modification of this function was used in the calculations reported by Krolak and Cooper (8). Problem 4 calls for minimizing this function under the constraints proposed for the second problem

of Table 1 in the article of Klingman and Himmelblau (1).

Specifically, the minimum of

$$F(x,y) = (y - x^2)^2 + (1 - x)^2$$

is required, assuming the constraints,

$$G_1 = x - .2 \geq 0$$

$$G_2 = 2 - x \geq 0$$

$$G_3 = y -.2 \geq 0$$

$$G_4 = 2 - y \geq 0$$

$$G_5 = 1 - x^2 - y^2 \geq 0.$$

It will not be possible to compare trial calculation with the Multiple-Gradient Summation Technique since it failed on this problem. From evidence in the text it is assumed that Problem 4, here defined, is the same as the second problem of Klingman and Himmelblau, and that the function actually shown in their table contains a typographical error. Exact comparisons are impossible for another reason; the starting points of all of the tests reported in their Table 1 are omitted.

Tangent Search Calculations were at first started with the non-feasible point, (5, -5) and with both step sizes unity. Table IVa displays totals of moves and evaluations required to find feasible origins, which are also given.

TABLE IVa, Tangent Search

Reduction Factor	Feasible Origin			Moves		Constraint Evaluations
	x	y	F	Base	Expl.	
.250	.333	.766	.874	6	3	17
.375	.734	.447	.079	6	5	30
.500	.250	.875	1.223	4	3	13

[illegible]

[illegible]

[illegible]

[illegible]

In order validly to compare step reduction factors in Tangent Search and that algorithm with the Complex Method, all of the results in Table IVb were obtained for calculations initiated at the feasible point, (.25, .875). Beginning step sizes for Tangent Search runs were unity in all cases. The correct solution to Problem 4 is (to six significant figures in each independent variable),

$$x_s = .808169$$

$$y_s = .588951$$

$$F_s = .0409190.$$

TABLE IVb

End Criteria: $|x - x_s| \leq .000005$ and $|y - y_s| \leq .000005$

Algorithm	Evaluations		Trials/Moves		
Complex Meth.	Const.	Obj. Funct.	Totals		Permissible
a	179	124	169		90
b	196	131	186		96
Tangent Search			Base	Explore	Alt.
r = .250	64	80	27	14	15
r = .375	96	109	37	18	24
r = .500	98	118	33	21	26

All three Tangent Search results are superior to both Complex Method solutions. Additionally, a reduction factor of .25 is best for this problem.

PROBLEM 5

This trial problem is also adapted from Krolak and Cooper (8) with constraints proposed by Klingman and Himmelblau (1). It is the minimization problem corresponding to the eighth example of their table 1.

Specifically, it is desired to find the minimum of

$$F(x) = -\left[(x_4 - 1)^{\sin x_2} + (x_1 - x_3)^2\right],$$

subject to the constraints:

$$0 \leq x_1 \leq 1$$

$$0 \leq x_2 \leq 2$$

$$-1 \leq x_3 \leq 1$$

$$1.05 \leq x_4 \leq 2$$

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 \leq 3$$

$$x_2 + x_3 + x_4 + x_2 x_3 + x_1 x_4 + x_3 x_4 \leq 2$$

$$x_2^2 - x_2 x_3 + x_3^2 + x_1^2 x_4 - x_1 x_3^2 \leq 6.$$

For the purposes of solution by Tangent Search the inequalities given above were modified and expressed as eleven constraint functions of the following form:

$$G_1(x) = x_1 \geq 0$$

$$G_2(x) = 1 - x_1 \geq 0$$

$$G_3(x) = x_2 \geq 0$$

$$G_4(x) = 2 - x_2 \geq 0$$

$$\ldots$$

$$G_{11}(x) = 6 - x_2^2 + x_2 x_3 - x_3^2 - x_1^2 x_4 + x_1 x_3^2 \geq 0.$$

Klingman and Himmelblau list the following solution obtained by their method:

$$x_1 = .995$$

$$x_2 = 4.04 \times 10^{-5}$$

$$x_3 = -.953$$

$$x_4 = 1.050$$

$$F = -4.793 \ .$$

Examination of this result shows that it lies very near the intersection of the 3rd, 7th, 9th, and 10th constraint boundaries. If it is assumed that the exact solution is at that intersection, the correct values can be immediately calculated:

$$x_{1s} = + \sqrt{.94875} = .974037987 \ldots$$

$$x_{2s} = 0$$

$$x_{3s} = -.974037987\ldots$$

$$x_{4s} = 1.05$$

$$F_s = -4.795 .$$

Calculations with the Complex Method for two initial random complexes and with Tangent Search for three values of r were performed for Problem 5, as for all problems reported in this section. When supplied with a non-feasible origin the first phase of Tangent Search always correctly located an initial feasible base for the constrained minimization phase. This feature of the algorithm was tested successfully on this and various other problems; it will not be discussed further.

All calculations reported in Tables Va, Vb, and Vc began from the feasible origin,

$$x_1 = .75$$

$$x_2 = .75$$

$$x_3 = -.375$$

$$x_4 = 1.3125$$

$$F = -1.718179 .$$

Initial step sizes for Tangent Search runs were .5 for all variables.

TABLE Va

End Criterion: F < -4.7

Algorithm		Evaluations		Trials/Moves		
Complex Meth.		Const.	Obj. Funct.	Total	Permissible	
	a	14719	10208	14669	10200	
	b	-	-	failed	-	
Tangent Search				Base	Explore	Alt.
	r = .250	157	81	13	1	15
	r = .375	279	125	15	1	21
	r = .500	259	85	8	1	18

Table Vb provides terminal counts for the same runs. The last two lines give totals for that point in each solution at which the base entered a hypersphere of radius .001 about the true solution. Both Complex Method runs and the Tangent Search calculation with r = .25 failed to reach results of comparable accuracy.

TABLE Vb, Terminal Results

Algorithm		Evaluations		Trials/Moves			Note/
Complex Meth.		Const.	Obj.Funct.	Total	Permissible		Case
	a	14879	10258	14827	10250		(1)
	b	10038	6942	9999	6934		(2)
Tangent Search				Base	Explore	Alt.	
	r = .250	664	303	67	1	65	(3)
	r = .375	830	369	88	1	76	(4)
	r = .500	930	382	73	4	84	(4)

N. B. For final results on failed cases, see Table Vc below.

(1) Failed; terminated due to excessive time.

(2) Failed; terminated when trials reached 9999.

(3) Failed: terminated when all D less than 10^{-10}.

(4) Succeeded; results given for the base at which $\sqrt{\sum_{i=1}^{n} (x_i - x_{is})^2} < .001$.

TABLE Vc

Best Point Achieved for Failed Runs. (See Table Vb.)

Final Value	Case		
	(1)	(2)	(3)
x_1	1.000	.840	.998
x_2	-.2E-7	.403	1.1E-10
x_3	-.947	-.666	-.949
x_4	1.050	1.300	1.050
F	-4.7922	-2.8903	-4.7926

For this problem Tangent Search seems superior to both Multiple-Gradient Summation and to the Complex Method. Also, r = .375 seems the best choice of reduction factor among the three considered.

PROBLEM 6

This illustration is essentially the same as the fifth problem of Klingman and Himmelblau. The objective function, once again, originated with Krolak and Cooper (8). Here the minimum of

$$F(w,x,y,z) = -\left(\sin x + z^{(w+y)}\right)$$

is desired, subject to constraints,

$$0 < w < 1$$

$$0 < x < 2$$

$$0 < y < 1$$

$$0 < z < 2$$

$$x^2 + y^2 + z^2 + w^2 < 1 .$$

As before, for the Tangent Search solution the constraints are re-expressed as simple inequalities for which the boundaries are feasible, i.e.,

$$G_1 = w \geq 0$$

$$G_2 = 1 - w \geq 0$$

$$\ldots$$

$$G_9 = 1 - x^3 - y^2 - z^2 - w^2 \geq 0.$$

Klingman and Himmelblau list the following result, which is erroneous:

$$w = 4.89 \times 10^{-6}$$

$$x = .559$$

$$y = 4.89 \times 10^{-6}$$

$$z = .829$$

$$F = -1.530 \ .$$

Examination of this problem reveals that the constrained minimum actually lies at the intersection of the 1st, 5th, 7th, and 9th constraint boundaries. That is, the correct solution is:

$$x_s = 1$$

$$w_s = y_s = z_s = 0$$

$$F_s = -1.841471 \ldots$$

All calculations with both Tangent Search and Complex Methods correctly solved this problem. Results are given in Tables VIa and VIb.

TABLE VIa

End Criterion: $F < -1.84$

<table>
<tr><th colspan="2">Algorithm</th><th colspan="2">Evaluations</th><th colspan="4">Trials/Moves</th></tr>
<tr><td colspan="2">Complex Meth.</td><td>Constr.</td><td>Obj. Funct.</td><td colspan="2">Total</td><td colspan="2">Permissible</td></tr>
<tr><td></td><td>a</td><td>292</td><td>199</td><td colspan="2">272</td><td colspan="2">180</td></tr>
<tr><td></td><td>b</td><td>279</td><td>172</td><td colspan="2">258</td><td colspan="2">160</td></tr>
<tr><td colspan="2">Tangent Search</td><td></td><td></td><td>Base</td><td colspan="2">Explore</td><td>Alt.</td></tr>
<tr><td></td><td>r = .250</td><td>431</td><td>222</td><td>43</td><td colspan="2">3</td><td>37</td></tr>
<tr><td></td><td>r = .375</td><td>558</td><td>252</td><td>52</td><td colspan="2">3</td><td>48</td></tr>
<tr><td></td><td>r = .500</td><td>564</td><td>321</td><td>46</td><td colspan="2">10</td><td>48</td></tr>
</table>

TABLE VIb, Terminal Results

Final Result Item	Algorithm				
	Complex		Tangent Search, r =		
	a	b	.250	.375	.500
w	.98E-9	1.0E-9	1.0E-10	.41E-7	.67E-8
x	1.0000	.99952	1.0000	1.0000	1.0000
y	.10E-8	.98E-9	1.2E-10	1.9E-7	6.9E-8
z	.783E-4	3.11E-2	1.18E-3	1.02E-3	1.15E-3
F	-1.841471	-1.841210	-1.841471	-1.841467	-1.841470
Evaluations					
Constraints	640	749	1422	2163	2016
Obj.Funct.	440	421	657	1060	1191
Trials/Moves					
Total	620	729	-	-	-
Permiss.	395	407	-	-	-
Base	-	-	136	190	171
Explore	-	-	5	22	39
Alternate	-	-	119	160	148
Note No.	(1)	(2)	(3)	(4)	(4)

(1) No further significant changes in variables could be found after this point.

(2) Method could not find a feasible replacement to the worst vertex at this point.

(3) All step sizes become less than 10^{-10} at this point.

(4) Solution terminated due to excessive computing time at this point; final step sizes were approximately 4×10^{-5}.

The evidence of these tables seems to show that the Complex Method is better than Tangent Search for solving this problem. The best result for the latter method was with r = .25 . Finally, both of these algorithms produce satisfactory solutions, even though Multiple-Gradient Summation failed.

PROBLEM 7

This is the practical problem which stimulated development of the Complex Method by M. J. Box (3). He was unable to solve it either by the method of Rosenbrock (7) or by certain other techniques.

The problem, as re-expressed here, is to minimize the following function of five independent variables, x_i, i = 1, 2,...,5:

$$F(x) = -\Big\{\Big[a_2y_1 + a_3y_2 + a_4y_3 + a_5y_4 + c_1 - c_2 - c_3(x_2 + .01x_3) + k_{31} + k_{32}x_2 + k_{33}x_3 + k_{34}x_4 + k_{35}x_5\Big] x_1 - 24345 + a_1x_6\Big\},$$

where,

$$x_6 = (k_1 + k_2x_2 + k_3x_3 + k_4x_4 + k_5x_5)x_1,$$

$$y_1 = k_6 + k_7x_2 + k_8x_3 + k_9x_4 + k_{10}x_5 ,$$

$$y_2 = k_{11} + k_{12}x_2 + k_{13}x_3 + k_{14}x_4 + k_{15}x_5 ,$$

$$y_3 = k_{16} + k_{17}x_2 + k_{18}x_3 + k_{19}x_4 + k_{20}x_5 ,$$

$$y_4 = k_{21} + k_{22}x_2 + k_{23}x_3 + k_{24}x_4 + k_{25}x_5 ,$$

$$x_7 = (y_1 + y_2 + y_3)\, x_1 ,$$

$$x_8 = (k_{26} + k_{27}x_2 + k_{28}x_3 + k_{29}x_4 + k_{30}x_5)\, x_1 + x_6 + x_7 ;$$

subject to the constraints:

$$0 \le x_1$$

$$1.2 \le x_2 \le 2.4$$

$$20 \le x_3 \le 60$$

$$9.0 \le x_4 \le 9.3$$

$$6.5 \le x_5 \le 7.0$$

$$0 \le x_6 \le 294000$$

$$0 \le x_7 \le 294000$$

$$0 \le x_8 \le 277200 .$$

The numerical values of a_i, $(i=0, 1,\ldots,7)$ and k_j, $(j=1,\cdots,35)$ are to be found in the Appendix of the paper of M. J. Box (3). In the function definition above, take

$$c_1 = 7840\ a_6$$

$$c_2 = 100{,}000\ a_0$$

$$c_3 = 50800\ a_7\ .$$

The initial point, specified by M. J. Box, is

$$x_1 = 2.52$$

$$x_2 = 2.0$$

$$x_3 = 37.5$$

$$x_4 = 9.25$$

$$x_5 = 6.8$$

$$F = -2{,}351{,}243.5,$$

which is feasible. According to Box the correct solution is:

$$x_{1s} = 4.53743$$

$$x_{2s} = 2.4$$

$$x_{3s} = 60$$

$$x_{4s} = 9.3$$

$$x_{5s} = 7.0$$

$$\cdots$$

$$x_{8s} = 277{,}200$$

$$F_s = -5{,}280{,}334$$

The paper of Box (3) gives results achieved by him in two trial runs. These figures will be labelled "B_1", and "B_2" in the tables below. For this problem three initial random complexes were employed by the present author in trial runs with the Complex Method. These will be labelled "a", "b", and "c".

A comparison of results among these and three Tangent Search calculations is given in Table VIIa for points comparatively early in each run. Step sizes for the latter method began at .1 for all variables.

TABLE VIIa

Criterion: Approximately 520 Constraint Evaluations

Algorithm	Evaluations		Obj. Function
Complex Method	Const.	Obj.Funct.	
B_1	527*	*	-5,236,850
a	538	425	-5,280,169
b	527	402	-5,261,088
c	521	402	-5,269,406
Tangent Search			
r = .250	524	261	-5,273,234
r = .375	534	248	-5,276,780
r = .500	518	247	-5,253,474

* At 300 permissible trials and 517 total trials.

Here it seems that Tangent Search achieves roughly comparable results early in the calculation with considerably fewer objective function evaluations. Comparisons at another convenient point, available for various runs, are provided in Table VIIb.

TABLE VIIb

Criterion: Approximately 1200 Constraint Evaluations

Algorithm	Evaluations		Obj.Funct. ($x10^3$)	Note
Complex Meth.	Const.	Obj.Funct.		
B_2	1206	?	-5,280.3	(1)
a	821	645	-5,280.3	(2)
b	1184	922	-5,278.9	(3)
c	1032	745	-5,271.8	(3)
Tangent Search				
r = .250	1186	587	-5,274.2	(4)
r = .375	1201	538	-5,280.2	(5)
r = .500	1201	565	-5,265.4	(6)

[illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]

(1) Reported by Box as 620 permissible moves out of 1196 trial moves.

(2) Calculation stopped after only 811 trials, 406 of which were permissible, with correct constrained minimum!

(3) These calculations terminated with the results shown because the algorithm was unable to generate a feasible vertex to replace the current worst vertex. In both cases the complex seemed to be contracted almost to the limits of significance of floating-point numbers on the CDC 1604. This may indicate that larger over-reflection factor would be helpful; 1.3 was used, as recommended.

(4) This is the 225th base point, found after 2 conventional and 83 tangent explore moves.

(5) This is the 192nd base point, found after 2 conventional and 90 tangent explore moves.

(6) This is the 145th base point, found after 7 conventional and 90 tangent explore moves.

Table VIIc gives terminal results for the Tangent Search calculations. Included are the results of a run with r = .375 but using a coupling cycle slightly different from that described in earlier parts of this paper. This column is labelled "Special". All of these began with initial step sizes at .1 for all variables. They were terminated normally.

TABLE VIIc, Tangent Search

End Criteria: $D_i < 10^{-6}$ for all i = 1, 2,..., n.

Final Result	Normal Algorithm r = .250	Normal Algorithm r = .375	Normal Algorithm r = .500	Special r = .375
x_1	4.61804	4.53810	4.54438	4.53744
x_2	2.25576	2.39875	2.38736	2.39997
x_3	59.9997	59.9916	59.9999	59.9991
x_4	9.30000	9.30000	9.30000	9.30000
x_5	6.99999	7.00000	7.00000	7.00000
F	-5274279	-5280269	-5279811	-5280332
Moves				
Base	298	235	418	511
Explore	3	6	27	14
Alt.	107	124	336	244
Evaluations				
Constr.	1539	1779	4884	3423
Obj. Funct.	778	805	2092	1616

The special run with r = .375 gives the best Tangent Search result for this example. It is almost exact. However the coupling cycle it uses was much less satisfactory on other problems than the one described here. The intermediate step reduction size is best with either coupling cycle. Satisfactory performance of Tangent Search seems marginal on this problem; it is highly dependent on coupling modes and choice of r. The Complex Method appears to be somewhat more dependable; however, it did not work perfectly for all initial random complexes.

PROBLEM 8

This example was constructed by the author to test Tangent Search's special expedient for dealing with convex constraint boundaries. The objective surface is the bottom part of the interior of a hemisphere. A convex constraint boundary lies between the initial point and the lowest point of the surface. The constraint is chosen in such a way that the base must always move along it in order to reach the correct solution.

Specifically, it is desired to minimize

$$F(x,y) = \sqrt{(x-5)^2 + (y-10)^2}$$

subject to the constraint,

$$G(x,y) = 144 - (x-5)^2 - (y+10)^2 \geq 0.$$

The initial point is

$$x = 11.5, \; y = 0, \quad F = 11.9269 \ldots$$

and the solution is

$$x_s = 5, \; y_s = 2, \quad F_s = 8.$$

Initial step sizes for Tangent Search calculations were .1 for both variables. Results for the usual Tangent Search and Complex method runs are given in Table VIII.

TABLE VIII

End Criterion: $\sqrt{(x-x_s)^2 + (y-y_s)^2} \leq .0001$

Algorithm		Evaluations		Trials/Moves		
Complex Meth.		Const.	Obj.Funct.	Total	Permiss.	
	a	138	67	131	60	
	b	142	66	137	60	
Tangent Search				Base	Expl.	Alt.
	r = .250	161	147	62	12	31
	r = .375	149	151	52	16	31
	r = .500	133	138	41	17	29

The Complex Method seems more efficient for this example. With Tangent Search the reduction factor, r =.5, appears best.

PROBLEM 9

The Multiple-Gradient Summation Method failed on this example, which corresponds to number seven of Klingman and Himmelblau (1). The objective function is the same as that of Problem 5 above, i.e.,

$$F(w,x,y,z) = -((z-1)^{\sin x} + (w-y)^2).$$

However, the constraints of this problem are

$$0 \le w \le 1$$

$$0 \le x \le 2$$

$$-1 \le y \le 1$$

$$1.05 \le z \le 2$$

$$x^2 + y^2 + z^2 + w^2 \le 2.$$

As before, these must be re-expressed in the following manner for Tangent Search:

$$G_1 = w \ge 0$$

$$G_2 = 1 - w \ge 0$$

$$\ldots$$

$$G_9 = 2-x^2 - y^2 - z^2 - w^2 \ge 0.$$

Trial calculations with Tangent Search and the Complex Methods were initiated at the feasible origin,

$$w = x = y = .001$$

$$z = 1.051$$

$$F = -.99703 \ldots$$

Initial step size for the former algorithm was .0625 for all variables. Examination of the problem shows that the constrained minimum lies at the intersection of the 3rd, 7th, and 9th constraint boundaries; specifically, this solution is:

$$w_s = .66988805\ldots$$

$$x_s = 0$$

$$y_s = -.66988805\ldots$$

$$z_s = 1.05$$

$$F_s = -2.795.$$

The Complex Method run beginning with random vertices chosen by sequence "a" failed to develop a satisfactory solution. It stopped after the 469th trial (262 permissible) with the following centroid:

$w = 1.0 \times 10^{-9}$

$x = 2.1 \times 10^{-9}$

$y = .9473648$

$z = 1.050000$

$F = -1.8975 \ldots$

No significant changes in the variables had taken place during the last eight permissible trials; this constitutes the usual end criterion for the Complex Method.

Results of the "b" calculation and the usual three Tangent Search runs are given in TAble IXa for a point closer to the true solution.

TABLE IXa

End Criterion: $\sqrt{(w-w_s)^2 + (x-x_s)^2 + (y-y_s)^2 + (z - z_s)^2} < .0075$

Algorithm	Evaluations		Trials/Moves		
Complex Meth.	Constr.	Obj.Funct.	Total		Permiss.
a	Failed	-	-		-
b	773	428	751		420
Tangent Search			Base	Exp.	Alt.
r = .250	503	263	100	1	44
r = .375	485	217	57	1	40
r = .500	601	255	49	2	58

The Complex Method, run "b", did not proceed to a more accurate solution after the point tabulated above. Best results achieved for the three Tangent Search runs are given in Table IXb.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

TABLE IXb, Tangent Search

End Criteria: $D_i < 10^{-10}$ for all i = 1, 2, ..., n.

Final Result	Step Reduction Factor .250	.375	.500
w	.672964	.670272	.669901
x	5.6E-11	4.0E-11	1.1E-11
y	-.666798	-.669504	-.669875
z	1.05000	1.05000	1.05000
F	-2.79496	-2.79500	-2.79500
Moves			
Base	169	170	103
Explore	1	2	6
Alternate	98	135	141
Evaluations			
Constr.	1049	1499	1507
Obj.Funct.	512	645	617

These results for Problem 9 are clearly favorable to the Tangent Search Method over both the Multiple-Gradient Summation and Complex Methods. An additional observation is that a less accurate solution (Table IXa) is achieved most easily with r = .375. However the most accurate final answer is found (with fewer evaluations) by r = .500.

PROBLEM 10

This illustration is taken from the article of M. J. Box (3) where it appears as "Problem B". As re-expressed here, the minimum of

$$F(x) = -\left\{ x_2^3 \left[9 - (x_1 - 3)^2 \right] / 27\sqrt{3} \right.$$

is required, under the constraints,

$$G_1(x) = x_1 \geq 0$$

$$G_2(x) = x_2 \geq 0$$

$$G_3(x) = x_1/\sqrt{3} - x_2 \geq 0$$

$$G_4(x) = x_3 \geq 0$$

$$G_5(x) = 6 - x_3 \geq 0, \qquad \text{where} \quad x_3 = x_1 + \sqrt{3}\, x_2 .$$

The initial point, as specified by Box, is

$$x_1 = 1, \quad x_2 = .5, \qquad F = -.01336 .$$

Initial step sizes for Tangent Search calculations were .1 for both variables. The correct constrained minimum is

$$x_{1s} = 3, \quad x_{2s} = \sqrt{3}, \quad F_s = -1.$$

Each independent variable must have explicitly-designated upper and lower bounds for the purposes of solution by the Complex Method. Since the upper bound of x_1 is not stated by Box (3), the present author chose 6.0 for the value.

Results of the usual two Complex Method and three Tangent Search calculations are given in Table X.

TABLE X

End Criterion: $F < -.99995$

Algorithm	Evaluations		Trials/Moves		
Complex Meth.	Constr.	Obj.Funct.	Total	Permiss.	
a	133	64	122	60	
b	122	59	111	55	
Tangent Search			Base	Expl.	Alt.
r = .250	121	91	55	4	26
r = .375	153	110	51	7	36
r = .500	159	107	38	10	36

The Complex Method seems to solve this problem more economically. With Tangent Search the smallest step reduction factor seems the best choice.

PROBLEM 11

This is the Post Office Parcel Problem, originally proposed by Rosenbrock (7), but here subject to the constraints imposed by Box(3).

Expressed as a minimization problem, the objective function is

$$F(x) = -x_1 \, x_2 \, x_3 \; ,$$

and the constraints to be observed are,

$$0 \leq x_1 \leq 20$$

$$0 \leq x_2 \leq 11$$

$$0 \leq x_3 \leq 42$$

$$x_1 + 2(x_2 + x_3) \leq 72 \ .$$

The designated origin for all computations is

$$x_1 = 18, \quad x_2 = 10, \quad x_3 = 16, \quad F = -2880 \ .$$

Initial step size on Tangent Searches was unity for all variables.

The correct answer for Problem 11 is

$$x_{1s} = 20, \quad x_{2s} = 11, \quad x_{3s} = 15, \quad F_s = -3300.$$

The usual trial runs with Complex and Tangent Search algorithms were made. A comparison of results is provided by Table XI. As stated previously, the centroid (or corresponding objective function value) is the reference compared with the end criterion for Complex Method calculations. The base point is similarly the reference for Tangent Search.

TABLE XI

End Criterion: $F < -3299.9$

Algorithm	Evaluations		Trials/Moves		
Complex Meth.	Constr.	Obj.Funct.	Total	Permiss.	
a	289	209	271	150	
b	368	258	359	190	
Tangent Search			Base	Expl.	Alt.
r = .250	195	107	38	1	30
r = .375	186	97	25	1	28
r = .500	268	131	27	3	40

The evidence of Table XI seems favorable to Tangent Search, and with that method the intermediate step reduction factor produces the best result.

Conclusions

The following general summary of the results of trial computations tabulated above is based to some degree upon qualitative judgments. Results with different methods for a particular problem are called "similar" if the total number of function evaluations required to reach a valid solution of pre-defined accuracy are approximately the same. This total for each run includes both objective and constraint function calls.

Otherwise, a method is termed "superior" to another for any particular problem if a valid solution is attained with substantially fewer evaluations. Also, of course, a method is superior if it achieves a valid constrained minimum while the compared method fails to do so. Finally, Tangent Search is never rated superior to another if a better result is achieved with only one of the step reduction factors considered. The converse concepts define when one algorithm is considered inferior to another. Table XII provides a summary of these judgments applied to the tables of the previous section.

TABLE XII

Number of Examples for Which Tangent Search Was Similar, Superior, or Inferior to Other Methods

Qualitative Comparison	Multiple Gradient Summation	Sequential Search	Complex Method
Similar	1	1	1
Superior	4	1	6
Inferior	0	0	4
Total Cases	5	2	11

It appears from this summary that the Tangent Search Method can be considered a reasonably-efficient means of locating a constrained minimum, provided the other methods are so classified. However, it must be remarked that all but one of the examples considered are text-book or artificial problems. Problem 7 is the only practical example and for it the performance of the Complex Method was somewhat superior.

Another general conclusion would be that with Tangent Search a step reduction factor of .25 or .375 seems better as a generally-applicable choice than .5 . On only one problem, the 8th, was the largest factor superior in performance to both of the others. It also appears that calculations with r = .25 converged toward the solution more rapidly than the others at the beginning of problems in 4 or 5 dimensions. However, more accurate final results were often obtained for runs with r = .375 or .500 (at the cost of extra evaluations). In particular, the best final values for Problems 5 and 7 were achieved with r = .375, and best final values for Problems 6 and 9 were for r = .500 . These examples of higher dimensionality were the most troublesome considered. The step reduction factor r = .25 seemed to work well for problems of dimensionality 3 or less.

As a final conclusion, based upon intuition alone, the author wants to endorse an often-stated opinion of most other investigators in this area; viz., the final word on the best method of obtaining constrained minima/optima remains to be said. The method of Davidon (9), the foundation of the method of Fletcher and Powell (10), has emerged as a tentative "optimum optimizer" for the unconstrained problem. However, no such leading candidate for solving the general problem of non-linear mathematical programming has yet been heralded.

Future Work

Several avenues for further investigations have been opened in the previous discussion. The present author hopes to investigate some of these:

(1) The step reduction factor in Tangent Search probably should be subject to modification by the algorithm, itself. Probably it should start at the smaller end of some range and become larger in the vicinity of a solution.

(2) Preliminary results of computational experiments now in progress show that partial derivatives should be calculated by the method of Wengert(5) or, perhaps, that of Smith (6). A package of subroutines allowing users to utilize the former method conveniently has been developed at the Computer Facility, U. S. Naval Postgraduate School. All necessary jumps and calling lists are generated by the Fortran-63 Compiler; therefore, the user is not inconvenienced by the fact that the derivatives are actually developed in subroutines. Of course the constraint functions must be capable of explicit statement if such a method is to be used.

(3) Explicit constraints should be treated separately to avoid useless evaluation of partial derivatives which are always either zero or unity. For the same reason the procedure which calculates derivatives of the implicit constraints could utilize a simple switch so that only those presently needed would be evaluated.

(4) More computational comparisons between Sequential Search and Tangent Search will be necessary to establish whether or not one method is generally superior.

(5) The author feels that more investigation of the parameters α and k is necessary in establishing their best general values in the Complex Method.

[illegible]

[illegible]

[illegible]

[illegible]

M. J. Box chose $\alpha = 1.3$ and $k = 2n$ on the basis of results after only 200 trials on only two examples. However, in several of the calculations reported above a complex became totally contracted before the correct solution was attained. This might be remedied by using a larger value for the over-reflection factor or more vertices.

REFERENCES

1. KLINGMAN, W. R. AND HIMMELBLAU, D. M. Nonlinear programming with the aid of a multiple-gradient summation technique. J. ACM 11 (Oct. 1964), 400-415.

2. GLASS, H. AND COOPER, L. Sequential search: A method for solving constrained optimization problems. J. ACM 12 (Jan. 1965), 71-82.

3. BOX, M. J. A new method of constrained optimization and a comparison with other methods. Computer Journal 8 (Apr. 1965), 42-52.

4. HOOKE, ROBERT AND JEEVES, T. A. "Direct search" solution of numerical and statistical problems. J. ACM 8 (Apr. 1961), 212-229.

5. WENGERT, R. E. A simple automatic derivative evaluation program. C. ACM 7 (Aug. 1964), 463-464.

6. SMITH, PETER J. Symbolical derivatives without list processing, subroutines, or recursion. C. ACM 8 (Aug. 1965), 494-496.

7. ROSENBROCK, H. H. An automatic method for finding the greatest or least value of a function. Computer Journal 3 (Oct. 1960), 175-184.

8. KROLAK, P. AND COOPER, L. An extension of Fibonaccian search to several variables. C. ACM 6 (Oct. 1963), 639-642.

9. DAVIDON, W. C. Variable metric method for minimization. A.E.C. Research and Development Report, ANL-5990, (Nov. 1959).

10. FLETCHER, R. AND POWELL, M.J.D. A rapidly convergent descent method for minimization. Computer Journal 6 (Jul. 1963), 163-168.

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* U. S. Naval Postgraduate School Monterey, California	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

TANGENT SEARCH METHOD OF CONSTRAINED MINIMIZATION

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(Last name, first name, initial)*

Hilleary, Roger R.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
17 March 1966	88	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	TR/RP No. 59
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES

Unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

The abstract is as given on the title page.

DD FORM 1 JAN 64 1473

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Constraint						
Direct						
Minimum						
Optimization						
Pattern						
Search						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization (*corporate author*) issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

DD FORM 1 JAN 64 1473 (BACK)

DISTRIBUTION LIST

Documents Department
General Library
University of California
Berkeley, California 94720

Lockheed-California Company
Central Library
Dept. 77-14, Bldg. 170, Plt. B-1
Burbank, California 91503

Naval Ordnance Test Station
China Lake, California
Attn: Technical Library

Serials Dept., Library
University of California, San Diego
La Jolla, California 92038

Aircraft Division
Douglas Aircraft Company, Inc.
3855 Lakewood Boulevard
Long Beach, California 90801
Attn: Technical Library

Librarian
Government Publications Room
University of California
Los Angeles 24, California

Librarian
Numerical Analysis Research
University of California
405 Hilgard Avenue
Los Angeles 24, California

Chief Scientist
Office of Naval Research
Branch Office
1030 East Green Street
Pasadena, California 91101

Commanding Officer and Director
U. S. Navy Electronics Lab. (Library)
San Diego 52, California

General Dynamics/Convair
P. O. Box 1950
San Diego, California 92112
Attn: Engineering Library
Mail Zone 6-157

Ryan Aeronautical Company
Attn: Technical Information Services
Lindbergh Field
San Diego, California 92112

General Electric Company
Technical Information Center
P. O. Drawer QQ
Santa Barbara, California 93102

Library
Boulder Laboratories
National Bureau of Standards
Boulder, Colorado

Government Documents Division
University of Colorado Libraries
Boulder, Colorado 80304

The Library
United Aircraft Corporation
400 Main Street
East Hartford, Connecticut 06108

Documents Division
Yale University Library
New Haven, Connecticut

Librarian
Bureau of Naval Weapons
Washington 25, D. C.

George Washington University Library
2023 G Street, N. W.
Washington, D. C. 20006

National Bureau of Standards Library
Room 301, Northwest Building
Washington, D. C. 20234

Director
Naval Research Laboratory
Washington, D. C. 20390
Attn: Code 2027

University of Chicago Library
Serial Records Department
Chicago, Illinois 60637

Documents Department
Northwestern University Library
Evanston, Illinois

The Technological Institute, Library
Northwestern University
Evanston, Illinois

Librarian
Purdue University
Lafayette, Indiana

Johns Hopkins University Library
Baltimore
Maryland 21218

Martin Company
Science-Technology Library
Mail 398
Baltimore, Maryland 21203

Scientific and Technical Information Facility
Attn: NASA Representative
P. O. Box 5700
Bethesda, Maryland 20014

Documents Office
University of Maryland Library
College Park, Maryland 20742

The Johns Hopkins University
Applied Physics Laboratory
Silver Spring, Maryland
Attn: Document Librarian

Librarian
Technical Library, Code 245L
Building 39/3
Boston Naval Shipyard
Boston 29, Massachusetts

Massachusetts Institute of Technology
Serials and Documents
Hayden Library
Cambridge 39, Massachusetts

Technical Report Collection
303A, Pierce Hall
Harvard University
Cambridge 38, Massachusetts
Attn: Mr. John A. Harrison, Librarian

Alumni Memorial Library
Lowell Technological Institute
Lowell, Massachusetts

Librarian
University of Michigan
Ann Arbor, Michigan

Gifts and Exchange Division
Walter Library
University of Minnesota
Minneapolis, Minnesota 55455

Reference Department
John M. Olin Library
Washington University
6600 Millbrook Boulevard
St. Louis, Missouri 63130

Librarian
Forrestal Research Center
Princeton University
Princeton, New Jersey 08540

U. S. Naval Air Turbine Test Station
Attn: Foundational Research Coordinator
Trenton, New Jersey

Engineering Library
Plant 25
Grumman Aircraft Engineering Corp.
Bethpage, L. I., New York 11714

Librarian
Fordham University
Bronx 58, New York

U. S. Naval Applied Science Laboratory
Technical Library
Building 291, Code 9832
Naval Base
Brooklyn, New York 11251

Librarian
Cornell Aeronautical Laboratory
4455 Genesee Street
Buffalo 21, New York

Central Serial Record Dept.
Cornell University Library
Ithaca, New York 14850

Columbia University Libraries
Documents Acquisitions
535 W. 114 Street
New York, New York 10027

Engineering Societies Library
345 East 47th Street
New York, New York 10017

Library-Serials Department
Rensselaer Polytechnic Institute
Troy, New York 12181

Librarian
Documents Division
Duke University
Durham, North Carolina

Ohio State University Libraries
Serial Division
1858 Neil Avenue
Columbus 10, Ohio

Commander
Philadelphia Naval Shipyard
Philadelphia, Pennsylvania 19112
Attn: Librarian, Code 249c

Steam Engineering Library
Westinghouse Electric Corporation
Lester Branch Post Office
Philadelphia, Pennsylvania 19113

Hunt Library
Carnegie Institute of Technology
Pittsburgh 13, Pennsylvania

Documents Division
Brown University Library
Providence, Rhode Island 02912

Central Research Library
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee

Documents Division
The Library
Texas A & M University
College Station, Texas 77843

Librarian
LTV Vought Aeronautics Division
P. O. Box 5907
Dallas, Texas 75222

Gifts and Exchange Section
Periodicals Department
University of Utah Libraries
Salt Lake City, Utah 84112

Defense Documentation Center (DDC)
Cameron Station
Alexandria, Virginia 22314
Attn: IRS (20 copies)

FOREIGN COUNTRIES

Exchange Section
National Lending Library for
Science and Technology
Boston Spa
Yorkshire, England

The Librarian
Patent Office Library
25 Southampton Buildings
Chancery Lane
London W. C. 2., England

Librarian
National Inst. of Oceanography
Wormley, Godalming
Surrey, England

Dr. H. Tigerschiöld, Director
Library
Chalmers University of Technology
Gibraltargatan 5
Gothenburg S, Sweden

International Computation Centre
Palazzo degli Uffici-Zona dell' E.U.R.
Rome, Italy

TA7
.U62
no.59
85251
Hilleary
The tangent search method of constrained minimization.