GEORGIA INSTITUTE OF TECHNOLOGY LIBRARY

Regulations for the Use of Theses

Unpublished theses submitted for the Master's and Doctor's degrees and deposited in the Georgia Institute of Technology Library are open for inspection and consultation, but must be used with due regard for the rights of the authors. Passages may be copied only with permission of the authors, and proper credit must be given in subsequent written or published work. Extensive copying or publication of the thesis in whole or in part requires the consent of the Dean of the Graduate Division of the Georgia Institute of Technology.

This thesis by BRIAN EVERETT ESTERBY has been used by the following persons, whose signatures attest their acceptance of the above restrictions.

A library which borrows this thesis for use by its patrons is expected to secure the signature of each user.

NAME AND ADDRESS OF USER	BORROWING LIBRARY	DATE

In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

_____

7/25/68

MODIFICATION OF ROSENBROCK'S ALGORITHM FOR THE NONLINEAR PROGRAMMING PROBLEM

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Brian Everett Esterby

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in the School of Industrial and Systems Engineering

Georgia Institute of Technology

December, 1970

MODIFICATION OF ROSENBROCK'S ALGORITHM

FOR THE NONLINEAR PROGRAMMING PROBLEM

Approved:

Date approved by Chairman: Dec. 10, 1970

ACKNOWLEDGMENTS

The author wishes to sincerely thank Dr. D. E. Fyffe for his guidance in the preparation of this thesis and for his academic counseling throughout the author's program of study. Dr. C. M. Shetty provided very valuable help in the definition of this research and has profoundly influenced the author's academic development in the classroom. Thanks are also due to Dr. D. C. Montgomery for graciously consenting to serve on the reading committee and for adding his insight to the final result.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

H. H. Rosenbrock's direct search algorithm for unconstrained nonlinear programming problems was examined and modified in an attempt to reduce the number of function evaluations, which is the major portion of the total computer running time, needed to solve certain test problems. The original search logic was compared with a golden section and two Fibonacci schemes, which retained the other basic attributes of the Rosenbrock search method. These new methods did not produce better results for a majority ot the test problems. A method which combined the Rosenbrock search and a Fibonacci style search in an alternating fashion did yield performance results equivalent to the Rosenbrock method alone.

The Rosenbrock algorithm uses a counter on the number of function evaluations and stops when a specified maximum is reached. Two alternate signs of optimality were evaluated as an additional termination criterion, but neither produced consistently reliable results.

The combined Rosenbrock and Fibonacci optimization method was evaluated for compatibility and effectiveness with several approaches to solving constrained nonlinear programming problems. Of these approaches, two proved to be generally successful. One was proposed by Rosenbrock and the other was Carroll's created response surface technique. Both are interior methods, but differ in the manner they treat constraints in the different parts of the feasible region. Rosenbrock's approach requires more manual preparation for its computer implementation.

Neither of these approaches could effectively solve a difficult 24 variable problem with 44 linear and nonlinear, equality and inequality constraints. This failure was attributed to the difficulty of the problem and the limited amount of computer time devoted to its solution.

In the solution of both constrained and unconstrained problems, it was noted that for any individual problem, one method might be far superior to other methods, even though it may not have been generally acceptable for all problems. The relationship between the solution procedure and the problem was not isolated, but this area appears to be suitable for further research into the efficiency of nonlinear optimization techniques.

CHAPTER I

INTRODUCTION

The generalized nonlinear programming problem can be stated as follows:

$$\text{Minimize } F(\bar{x}) \tag{1}$$

$$\text{subject to: } g_i(\bar{x}) \geq 0, \; i=1, \ldots, m \tag{2}$$

$$e_i(\bar{x}) = 0, \; i=m+1, \ldots, p \tag{3}$$

where $\bar{x} = (x_1, x_2, \ldots, x_n)$. Equations (1), (2), and (3) are nominally nonlinear, although special cases exist where some of them are linear. Equations (2) and (3) are called inequality and equality constraints, respectively, and a problem with either or both types is called a constrained nonlinear optimization problem. When equations (2) and (3) are both missing, the problem is said to be unconstrained.

Procedures for solving the nonlinear programming problem fall into two general classes: (a) direct search methods, and (b) gradient methods. Direct search methods employ a logical, iterative approach to solving (1) by choosing a point $\bar{x}$, comparing its function value to that of a previously known point, and using the results of the comparison to choose another point to generate a sequence of $\bar{x}$'s that converge to a solution point. The only computations involved are those that choose the points and evaluate the function. Simplicity in both

theoretical concept and computer implementation is the chief advantage of a direct search, while greater computation time than required by gradient methods, is the main disadvantage. Gradient searches are also iterative in nature, but employ more properties of the function to be optimized. First- and/or second-order partial derivatives are computed at each $\bar{x}$ to indicate the next most profitable $\bar{x}$. Since instructions to solve for these partial derivatives in either analytic or numerical methods form are required, the increase in preparation effort must be weighed against the additional speed of the computer solution.

One of the more widely known direct search methods was introduced by H. H. Rosenbrock in 1960. His method was referenced in many articles throughout the past decade and its performance has been compared to other direct and gradient methods in several books and papers. The method is simple and easy to use. For an n dimensional problem, Rosenbrock uses n orthonormal directions, initially unit vectors along the coordinate axes, and makes a sample function evaluation along each direction in turn, increasing or decreasing the distance and changing the orientation from the previous best point based upon preceding evaluations. When a successful evaluation followed by an unsuccessful evaluation has been made in each direction, new directions are computed from the old and the aggregrate results of each successful evaluation. The first of these new directions is oriented toward the direction of steepest descent. This series of one-dimensional searches is continued until a specified maximum number of function evaluations has been reached.

The step size, which indicates the distance and orientation along each orthonormal direction to the next $\bar{x}$, is modified after each trial. Following a successful evaluation,the step size is multiplied by a positive constant greater than one. After a failure, the current step size is multiplied by a negative fraction between zero and minus one. Thus, the next time this direction is chosen for evaluation, the step size is properly modified to reflect the previous results.

Objectives

Since the directions are changed from one trial to the next, the Rosenbrock method cannot truly be considered a series of one-dimensional searches. However, the process of rewarding success and punishing failure as Rosenbrock does would be an acceptable, though inefficient, one-dimensional search procedure. There are methods for making more exact one-dimensional searches, particularly the golden section and Fibonacci methods. Therefore, during the first phase of this research, alternative methods of point selection using the golden section and Fibonacci techniques will be calculated. Performance of these methods will be compared by the number of function evaluations required to reduce the objective function of several test problems of specified values.

The necessity to predict the required number of function evaluations invites inefficiency from two directions. Since the Rosenbrock method, and presumably any modification to it, will continue to search even at an optimal point, predicting too large an iteration maximum results in wasted function evaluations after the optimum is

reached. Predicting too small a maximum delivers only a reduced function value which is not optimal. The program could be restarted from the last point of the partial solution, but the valuable information in the set of directions at the conclusion of the partial run would normally be lost. Results from a restarted solution would not be equivalent to a single solution from the first initial point. Rosenbrock made some suggestions about signals of either true optimality or problems that are too difficult for his method. By examining these signals for a variety of problems, it should be possible to develop an alternate termination criterion to complement the iteration counter. That development will be secondary goal for this first phase of this research.

Both of the objectives of the first phase involve only the basic operation of Rosenbrock's direct search method. This phase will be characterized by attempted solutions to unconstrained nonlinear programming problems exclusively. Unconstrained problems provide the most efficient means for comparison of solution methods, since computer time need not be devoted to solving constraint equations.

Constrained problems offer the greatest challenge to optimization techniques since they add many conditions for a point to be a solution to (1). A point that satisfies (2) and (3) is said to be feasible. The feasible point which minimizes $F(\bar{x})$ is optimal. Two common approaches to solving constrained problems are: (a) formulate (1), (2), and (3) into a single unconstrained function whose minimum is at the same point which minimizes the generalized nonlinear programming problem, and (b) solve (2) and (3) for feasible points and then minimize $F(\bar{x})$ for $\bar{x}$ feasible. Many approaches to these formulations have been

designed strictly for either (2) or (3), although much of the more recent work has been directed at solution forms for the general non-linear programming problem.

The second phase of this research will be devoted to solving sample constrained problems from the literature of nonlinear programming The goal will be an examination of several problem formulations for their compatibility with the best method found in the first phase and for their effectiveness in solving the test problems. Pre-programming a solution procedure to require a minimum of special instructions for individual problems will provide users with an easy to use optimization tool.

CHAPTER II

LITERATURE SURVEY

Rosenbrock's Method for Unconstrained Problems

In 1960 (14) and again in 1966 with C. Storey (15), H. H. Rosenbrock presented his direct search algorithm for finding the minimum of a nonlinear function. The algorithm makes modified one-dimensional searches along n orthogonal directions, which are periodically changed to provide the most profitable set of directions. Rotation of the directions aligns one along the direction of steepest descent and uses the remaining directions to provide correction for imperfections in the alignment of the first.

To describe the detail of Rosenbrock's method, the logic of its computer implementation will be developed, leading to a flow diagram in Figure 1. The computer program requires a function or subroutine to compute the function value for any $\bar{x}$ vector. A call to this function will be indicated by $F(\bar{x})$. Although the instructions to compute the function value will normally indicate the dimensionality of the problem, certain test problems offer variable dimensionality and make it appropriate to require the value of n to be read in. Along with n, other required inputs are the trial maximum and the n-component vector $\bar{x}$, which is the point where the search is to be started. The method is characterized by trials and stages. A trial is a single function evaluation. A stage is a varying number of trials which meet requirements to be specified below.

Within the program, there are two arrays the same size as $\bar{x}$, which is at least as large as n elements. The first is e, which is an array of step sizes for each direction. The second is d, each element of which records the sum ot the successful steps in the direction it represents. For a minimization problem, a success is defined as a function evaluation less than or equal to the best previous value. There is also a two-dimensional array v, which is at least n by n in size, for the n orthonormal directions each with n components.

After the input variables are read, the directions are set to unit vectors along the coordinate axes and the value of $F(\bar{x})$ is computed and stored in a location called F_{best}. F_{best} always retains the current best function value. All the e_i are set to 0.1, and the d_i are set to zero. Success-failure status indicators for each direction are also initialized. The first trial, not counting the function evaluation during initialization, is computed at $\bar{x} + e_1\bar{v}_1$. The resulting value is compared to F_{best}. If $F(\bar{x} + e_1\bar{v}_1)$ is less than or equal to F_{best}, the value of F_{best} is changed to this value, $\bar{x}$ is set equal to $\bar{x} + e_1\bar{v}_1$, e_1 is added to d_1 and then e_1 is replaced by $3e_1$, and a success is recorded for the first direction. If $F(\bar{x} + e_1\bar{v}_1)$ is greater than F_{best}, $\bar{x}$ and F_{best} retain their original values, d_1 is unchanged, e_1 is replaced by $-0.5e_1$, and a failure is recorded. Rosenbrock chose the multipliers 3 and -0.5 after a comparison of several combinations on a moderately difficult problem. In his evaluation, Rosenbrock chose not to use the optimal combination, 5 and -0.5, because the values 3 and -0.5 offered better performance for general problems. Kowalik and Osborne (7) demonstrated the dependence of these multipliers on particular problems

when they presented results using 5 and -0.5, 3 and -0.5, 1.5 and -0.5, and 2 and -0.3.

The second trial is made at the point $\bar{x} + e_2\bar{v}_2$, where $\bar{x}$ is changed from its initial value if the first trial was a success. The results of the second trial are compared with the current value of F_{best}, and the values of $\bar{x}$, e_2, d_2, and F_{best} are adjusted as above, if necessary, and a success or failure is recorded. The following trials are made in the directions $\bar{v}_3$, $\bar{v}_4$, . . . , $\bar{v}_n$ and then the sequence is started again with $\bar{v}_1$. When at least one success followed by a failure has been made in each direction, the stage counter is incremented and new directions are computed via the Gram-Schmidt orthonormalization process.

The d_i now contain the sum of the progress made in the direction $\bar{v}_i$. Alpha vectors are computed in the following manner:

$$\begin{aligned}
\bar{\alpha}_1 &= d_1\bar{v}_1 + d_2\bar{v}_2 + \ldots + d_n\bar{v}_n \\
\bar{\alpha}_2 &= \quad\; d_2\bar{v}_2 + \ldots + d_n\bar{v}_n \\
&\vdots \\
\bar{\alpha}_n &= \qquad\qquad\quad d_n\bar{v}_n
\end{aligned}$$

The norm of the first alpha vector indicates the total linear progress made in the n-space during the previous stage. γ is defined as the ratio of the norm of the second alpha vector to the norm of the first and is a measure of how closely the initial $\bar{v}_1$ is aligned with the new $\bar{v}_1$. These new $\bar{v}_i$, which are designated $\bar{v}^*_i$, are derived below:

$$\bar{b}_1 = \bar{\alpha}_1$$
$$\bar{v}_1^* = \bar{b}_1 \; / \; \|\bar{b}_1\|$$
$$\bar{b}_2 = \bar{\alpha}_2 - \langle \bar{\alpha}_2, \bar{v}_1^* \rangle \bar{v}_1^*$$
$$\bar{v}_2^* = \bar{b}_2 \; / \; \|\bar{b}_2\|$$
$$\vdots$$
$$\bar{b}_n = \bar{\alpha}_n - \sum_{i=1}^{n-1} \langle \bar{\alpha}_n, \bar{v}_i^* \rangle \bar{v}_i^*$$
$$\bar{v}_n^* = \bar{b}_n \; / \; \|\bar{b}_n\|$$

where $\langle \bar{r}, \bar{s} \rangle = \sum_{i=1}^{n} r_i s_i$. The vector $\bar{v}_1^*$ becomes a unit vector in the direction of the line connecting the first point of the previous stage with the last. Thus, if the program is following a ridge, the first direction is always roughly aligned with the ridge. Each succeeding direction is the best available direction normal to the previous.

Following the computation of the new directions, the program returns to the initialization of the e_i, d_i, and the success-failure status indicators. The process continues until the iteration maximum is reached and the program is terminated. A computer flow diagram of the Rosenbrock algorithm is presented in Figure 1. Write statements have been omitted, but it is normally convenient to print the values of F_{best}, $\bar{x}$, $\|\bar{\alpha}_1\|$ and $\aleph$ at each stage along with the trial and stage counters.

Palmer Modification

In 1969, J. R. Palmer (9) presented an alternative procedure for calculating new directions. The Gram-Schmidt procedure requires that all of the d_i be nonzero. This is no problem with Rosenbrock's method, but

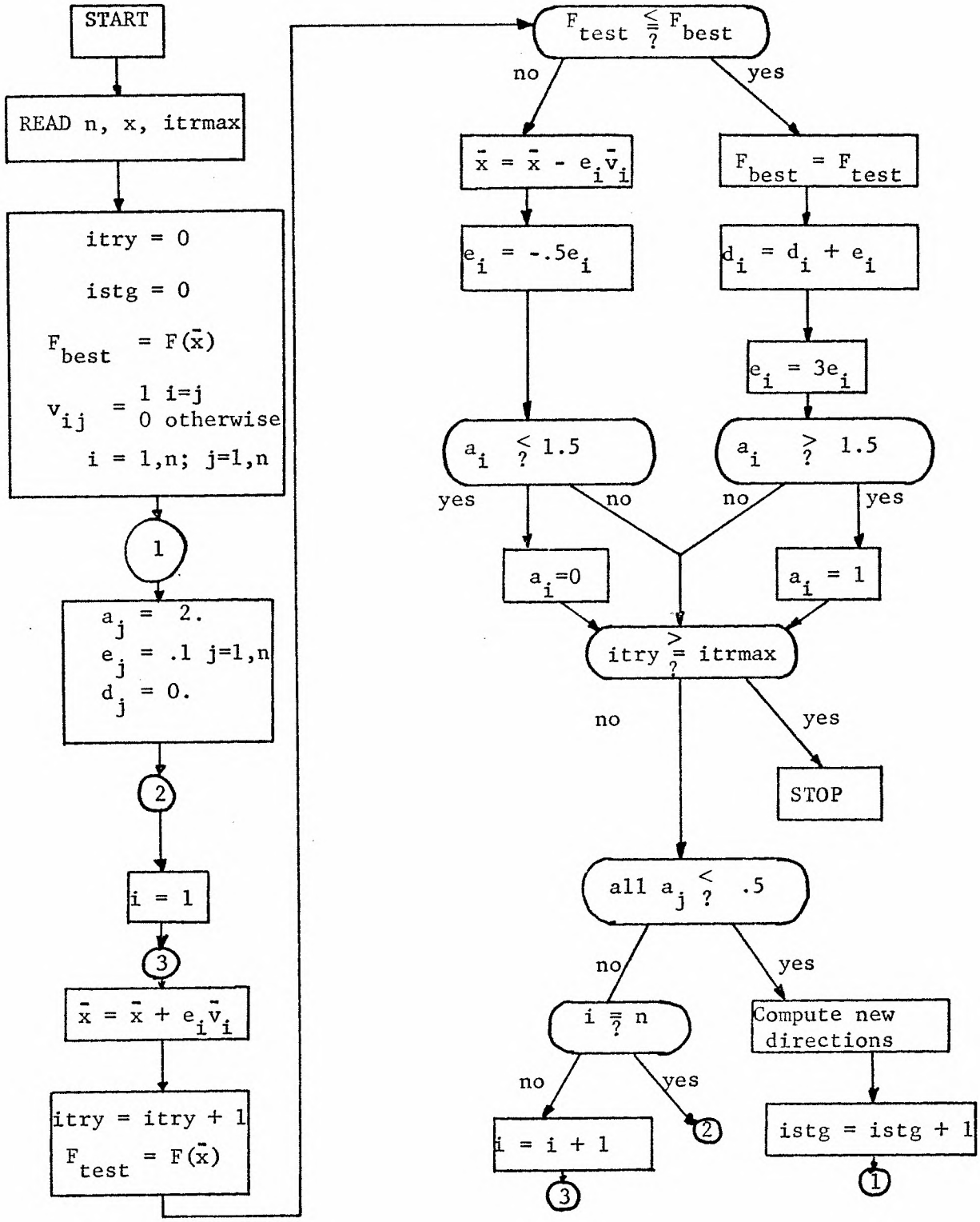

Figure 1. Rosenbrock's Algorithm

certain modifications to the basic method may generate d_i that are exactly zero. Palmer's derivation is too lengthy to reproduce here, but it has been tested and works satisfactorily. The modification also has an efficiency advantage over the Gram-Schmidt procedure. Palmer's method reduces the number of additions, subtractions, replacements, and storage requirements by a factor of n, and reduces the number of multiplications by approximately n/2.

One-Dimensional Searches

The logic for determining the value of each e_i in the basic Rosenbrock method brings to mind one-dimensional search procedures. The optimal one-dimensional search is a Fibonacci search, which reduces the final interval of uncertainty (the interval in which the minimum is known to exist) to a minimum for a previously specified number of function evaluations (16). If it is of no value to specify the number of function evaluations in advance, or the experimenter wishes to specify the final interval of uncertainty, the golden section search approaches the performance of the Fibonacci search. The disadvantages of these methods are that they do not easily extend to n dimensions (12) and they require a finite interval to be searched. Rosenbrock's search is open-ended for each direction and could detect unbounded solutions during any stage. The Fibonacci and golden section searches could not detect unbounded solutions because the independent variable must be bounded.

Constrained Problems

Practical applications of nonlinear programming force the consideration of constrained functions. The techniques for solving constrained

problems are varied, but most either attempt to combine the constraints with the objective function so that the solution to the combined function will be the optimal feasible solution to the original function or solve for feasible points and then minimize $F(\bar{x})$ for these points.

Rosenbrock's Approach

Rosenbrock proposed a method for solving an inequality constrained problem which is compatible with his method (14)(15). The method is an interior type (that is, it considers only feasible points). If an initial feasible point is not available, an auxiliary function based on the constraints can be solved to generate such a point. The objective function for Rosenbrock's approach is not modified when the point $\bar{x}$ is not near an active constraint. As an active constraint is approached and a narrow boundary near the constraint is entered, the objective function is penalized by a nonlinear function of the distance to the constraint. No function evaluations are allowed outside the feasible region. The program logic is modified to transfer control directly to the failure path of the flow diagram when a constraint violation is detected.

Step Function Approach

A unit step function approach to constrained optimization was first proposed in 1943 (7). One of these methods uses the Heaviside unit step function:

$$H(z) = \begin{cases} 1 \text{ for } z < 0 \\ 0 \text{ for } z \geq 0 \end{cases}$$

and solves a series of problems of the form:

$$T_j(\bar{x}) = (F(\bar{x})-f_j)^2 \, H(f_j-F(\bar{x})) + \sum_i g_i^2(\bar{x}) \, H(g_i(\bar{x})) + \sum_i e_i^2(\bar{x})$$

where f_1 is chosen less than the initial F_{best}. The f_j are a monotonically decreasing sequence. When the solution to any $T_j(\bar{x})$ is zero, $F(\bar{x})$ is known to be less than f_j and the constraints are satisfied. When the minimum $T_j(\bar{x})$ is greater than zero, the optimal value of $F(\bar{x})$ is bounded by f_{j-1} and f_j. Further investigations could be made in that interval for more accuracy.

Created Response Surface Technique

C.W. Carroll presented the created response surface technique in 1961 (3). His techniques were later validated and extended by Fiacco and McCormick (4)(5). This technique is also an interior method which solves the transformation function:

$$T(\bar{x},r_j) = F(\bar{x}) + r_j \sum_i g_i^{-1}(\bar{x}) + r_j^{-\frac{1}{2}} \sum_i e_i^2 (\bar{x})$$

where $\lim_{j\to\infty} r_j = 0$. As the solution progresses, the values of r_j become smaller, which permits the point $\bar{x}$ to approach inequality constraints with a small penalty, but provides a large penalty for loosely satisfied equality constraints. In the limit, $T(\bar{x},r_j) = F(\bar{x})$ if a solution exists. Otherwise, $T(\bar{x},r_j)$ is unbounded.

Penalty Functions

W. I. Zangwill presents a discussion of a general penalty function approach, which is an exterior method (18). Since it is an exterior method, penalty functions allow evaluations at infeasible points, but attach a penalty for doing so. A sequence of functions, in the above manner, is formulated so that the penalty becomes increasingly larger and forces the solution to the optimal feasible point. The transformation function has a form similar to:

$$T(\bar{x},r_j) = F(\bar{x}) + r_j^{-1} \sum_i [\min(g_i(\bar{x}),0)]^{1+z}$$

where $\lim_{j\to\infty} r_j = 0$ and $z > 0$. Equality constraints could be added in a form similar to that for the created response surface technique. In the limit, the summation in the transformation function must be zero to cancel the effect of the r_j factor, and allow $T(\bar{x}, r_j) = F(\bar{x})$. There are no restrictions on the selection of an initial point.

Paviani and Himmelblau Approach

The last approach for constrained problem formulation which will be presented here is due to Paviani and Himmelblau (10). Their paper presents a solution method for constrained problems based on the sequential simplex algorithm. The objective function is unchanged, but a criterion function is evaluated at each trial. If the criterion is satisfied, the minimization is allowed to proceed. If it is violated, the minimization is interrupted to return the selected point to the near feasible region. A point in the near feasible region satisfies the criterion function, but may or may not satisfy the constraint equations. The criterion function is:

$$C_j - [\sum_i [\min(g_i(\bar{x}),0)]^2 + \sum_i e_i^2(\bar{x})]^{\frac{1}{2}} \geq 0$$

where C_j is positive and non-increasing. Paviani and Himmelblau chose to make C_j a function of the simplex points, which is not applicable to solution with the Rosenbrock algorithm. This method is not strictly an interior method, but approximates one to the extent that the criterion function must be satisfied.

Comparative Performance Results

Rosenbrock, 1960 and 1966

Several authors, including Rosenbrock, have presented comparisons of the performance of Rosenbrock's algorithm with others. In his original paper, Rosenbrock concluded that his method was only slightly inferior in comparison to other methods on simple problems, but that it was probably superior on difficult problems (14). He developed a difficult problem to support the last claim. In 1966, Rosenbrock presented more detail in the comparison of his method with others. He reached essentially the same conclusions, adding that his method was "relatively difficult to defeat" (15).

Wilde, 1964, and Wilde and Beightler, 1967

D. J. Wilde, in 1964, presented results favorable to Rosenbrock's method on the Rosenbrock problem when compared to several other methods (16). In 1967, he presented the same results, but added another method using gradients which solved the problem with one-tenth the effort required by Rosenbrock's method (17).

Fletcher, 1965

In 1965, Fletcher reported a comparison of three algorithms for several problems (6). None of the three was Rosenbrock's, which was mentioned in Fletcher's introduction as being one of the more efficient early algorithms. Fletcher did compare the performance of a method by Davies, Swann, and Campey, which is considered to be an improvement of Rosenbrock's method. Their technique involved linear minimization along each of the search directions. The linear minimization was accomplished by approximating the function with a quadratic along each direction over

an interval that bracketed the minimum value, and then solving for the minimum of the quadratic. When no progress was made in a given direction, that direction was eliminated from the next orthonormalization process to insure future orthogonality. Fletcher concluded that another method, due to Powell, was more efficient than the Davies, Swann, and Campey method.

Box, 1965

Also in 1965, M. J. Box compared the performance of Rosenbrock's constrained formulation to a new method he developed (1). Box's study began when he found that Rosenbrock's method did not effectively solve an applied problem after attaining 98 per cent of the true optimal value. In addition to his new method, Box discovered that if inequality constraints are in the proper form, some improvement can be made in the performance of Rosenbrock's method by setting components of $\bar{x}$ to their bounding value when they enter the boundary zone. In doing this to his applied problem, Box eventually set all of the component values but one and then found what a comparatively poor one-dimensional search procedure Rosenbrock's method is.

Box, 1966

In 1966, Box compared the performance of eight optimization procedures, including both the Rosenbrock algorithm and the Davies, Swann, and Campey modification (2). These results show that there are more powerful methods than Rosenbrock's, but that his does have the reliability he claimed. Several of the methods, including Davies, Swann, and Campey's, failed to solve some of the problems, but Rosenbrock's method always achieved the desired accuracy, usually in a few more trials than the others. For constrained problems, Rosenbrock's formulation did

not compare favorably, but Box's modification to it did very well.

Kowalik and Osborne, 1968

Kowalik and Osborne discovered a combination of a problem and success-failure multipliers for which Rosenbrock's method failed (7). Box, however, reported a solution to the identical problem, probably with different multipliers (2). Other Kowalik and Osborne data compared Rosenbrock's method to other direct search methods for unconstrained problems. Their results indicate that Rosenbrock's procedure is better than Hooke and Jeeves pattern search, but not as effective as the Nelder and Mead version of the sequential simplex. No results for solution to constrained problems were presented for direct search methods.

CHAPTER III

FIRST PHASE - UNCONSTRAINED PROBLEMS

Objectives

The main objective of the First Phase was to refine the Rosenbrock algorithm within its basic nature to produce a reduction in the total number of function evaluations to solve standard test problems. The term "basic nature" is intended to mean a search procedure using n orthonormal directions, characterized by trials and stages. The primary difference between Rosenbrock's basic method and the proposed method was the manner in which the points were selected for evaluation.

An intuitive disadvantage to Rosenbrock's constant multipliers for point selection is that more precision can be made. To get this precision, more instructions are needed and bounds must be added to the region of search. The benefit of added precision was not known at the beginning of the study, but it was felt that it would hasten convergence in the later stages of a solution. Precision during the initial stages seemed unnecessary, since only a general indication of the direction toward optimality is desirable.

A secondary objective for the First Phase was the development of an automatic stopping criterion suitable for either Rosenbrock's algorithm or the modification to it. The trial counter maximum was undesirable for reasons specified in Chapter I. In the author's previous work, a stage counter with its own maximum was used, but this proved

useful only if the purpose of the computer run was to demonstrate the program's operation for a few stages. Predicting the required number of stages is no less difficult than predicting the number of trials.

Rosenbrock developed two parameters to aid the termination process. One is $\|\bar{\alpha}_1\|$ which measures the total progress in n-space during the previous stage. Clearly, a series of small $\|\bar{\alpha}_1\|$ indicates that little progress is being made and that optimality may be near. The second parameter is γ, which is a measure of how closely two consecutive $\bar{v}_1$'s are aligned with each other. Both of these parameters are computed when new directions are developed at the end of a stage. Rosenbrock suggested that when six consecutive stages have small $\|\bar{\alpha}_1\|$ and have γ nearly equal to one, "then the [optimum] has been reached or the problem is so difficult that it has defeated the method" (15).

Test Problems

The First Phase was best accomplished using only unconstrained minimization problems, since constraints only add computation time and do not affect the basic nature of the method. Several standard test problems were chosen from the literature. Initially eight problems were selected, but this number was later reduced to four representative problems.

Problem 1

$$F(\bar{x}) = (x_1 - x_2)^2 + \left(\frac{x_1 + x_2 - 10}{3}\right)^2$$

Initial Point: $\bar{x} = (0,-1)$ $F(\bar{x}) = 14.4$

Solution Point: $\bar{x} = (5,5)$ $F(\bar{x}) = 0$

Problem 1 was a two-dimensional simple elliptical function from Rosenbrock and Storey (15). Rosenbrock used this problem to demonstrate the ease with which certain gradient methods solve well-shaped functions, while his method experienced comparatively more difficulty. A steepest descent method solved the problem in 25 trials (where a trial was a function evaluation and two partial derivative computations; this "trial" is comparable to three function evaluations). Rosenbrock reported that his method took 94 trials to reach the same solution, with relatively slow progress during the last 60 trials.

Problem 2

$$F(\bar{x}) = 100\ (x_1^2 - x_2)^2 + (1 - x_1)^2$$

Initial Point:	$\bar{x} = (-1.2,1)$	$F(\bar{x}) = 24.2$
Solution Point:	$\bar{x} = (1,1)$	$F(\bar{x}) = 0$

Problem 2 was constructed by Rosenbrock to confound many of the then available optimization techniques, and it is not coincidental that his method solves the problem effectively (14)(15). The function represents a highly asymmetric parabolic valley in which the direction of steepest descent changes constantly for small changes in the point $\bar{x}$. Rosenbrock reported a solution in 248 trials with his method, while other methods did not achieve similar accuracy in 2800, 1800, or 1400 equivalent trials.

Problem 3

$$x_1 = \sin\ (x_1 + x_2)$$
$$x_2 = \cos\ (x_1 - x_2)$$

Initial Point: $\bar{x} = (.5, .5)$

Desired Solution Point: $\bar{x} = (.93508, .99802)$

Problem 4

$$x_1^4 + x_2^4 = 67$$

$$x_1^3 - 3 x_1 x_2^2 = -35$$

Initial Point: $\bar{x} = (2,3)$

Solution Point: $\bar{x} = (1.8836, 2.7159)$

Problem 5

$$x_1 = \sin(x_1) \cosh(x_2)$$

$$x_2 = \cos(x_1) \sinh(x_2)$$

Initial Point: $\bar{x} = (7,3)$

Desired Solution Point: $\bar{x} = (7.4977, 2.7687)$

In his book with C. Storey, which is directed at computational applications for chemical engineers, Rosenbrock devotes some attention to the use of nonlinear programming algorithms for solving systems of nonlinear equations (15). Problems 3,4, and 5 are from Schaum's Outline on Numerical Analysis by Francis Scheid, and are the only test problems which do not appear in the literature of nonlinear programming. They are solved by forming the sum of the square of the residual for each equation and minimizing this sum. Problems 3 and 5 challenge the solution techniques to determine the nearest local optimum when several exist due to the nature of the trigonometric functions.

Problem 6

$$F(\bar{x}) = \sum_{i=1}^{10} \left[\left(e^{-ix_1/10} - e^{-ix_2/10} \right) - \left(e^{-i/10} - e^{-i} \right) \right]^2$$

6a. Initial Point: $\bar{x} = (0,0)$ $F(\bar{x}) = 3.064$

6b. Initial Point: $\bar{x} = (0,20)$ $F(\bar{x}) = 2.087$

6c. Initial Point: $\bar{x} = (5,0)$ $F(\bar{x}) = 19.59$

6d. Initial Point: $\bar{x} = (5,20)$ $F(\bar{x}) = 1.808$

6e. Initial Point: $\bar{x} = (2.5,10)$ $F(\bar{x}) = .8081$

Solution Point: $\bar{x} = (1,10)$ $F(\bar{x}) = 0$

M. J. Box presented Problem 6 in 1966 (2). It arises from chemical engineering problems concerning reaction rate estimation. This also is a highly asymmetric curved valley, similar to Rosenbrock's Problem 2. Box compared eight optimization methods on this problem, and although Rosenbrock's method did not rate highly, it never experienced a failure to converge as some other methods did.

Problem 7

$$F(\bar{x}) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

Initial Point: $\bar{x} = (3,-1,0,1)$ $F(\bar{x}) = 215$

Solution Point: $\bar{x} = (0,0,0,0)$ $F(\bar{x}) = 0$

Problem 7 was introduced by Powell in 1962 when he presented his procedure for finding the optimum of a function (11). This problem was reported by Fletcher as being a particularly stringent test for optimization algorithms (6). Note that as the components of $\bar{x}$ become small, the fourth order terms become less significant and a result of the form $x_1 = -10x_2$ and $x_3 = x_4$ can be expected.

Problem 8

Chebyquad Function

Initial Point: $\bar{x} = (\frac{1}{n+1}, \frac{2}{n+1}, \ldots, \frac{n}{n+1})$, $n=2,3,\ldots$

Solution Point: Varies with n

$F(\bar{x}) = 0$ for n = 2-7,9

The Chebyquad function was introduced by R. Fletcher (who credits it to Dr. C. M. Reeves) as a problem which is typical of the type that might be encountered in normal applications (6). A desirable attribute of the problem is that the dimensionality is variable. It was formed by making a Chebyshev n-point quadrature over the interval (0,1). The problem is suitable for minimization for all n, but results in $F(\bar{x}) = 0$ only for n = 2,3,4,5,6,7, and 9 (8). The solution points are the roots of n degree polynomials. The existence of imaginary roots causes the final result to be greater than 0. During various stages of the current work, Chebyquad was solved for n = 2, 4, 6, 7, 8, 9,and 10. For the purposes of evaluation, only n = 9 was used since the increased number of variables provided the major difference from the previous problems.

Test Procedures

During the first stages of the search for a modified procedure, all of the test problems were used to compare the proposed methods. It is significant that the fastest method for solving Problems 1 and 3, which were comparatively easy, had to be discarded because of its failure to solve the more difficult problems. Once trends were established, it became necessary to make subtle, more minute changes in the methods. This greatly increased the number of computer runs which dictated a reduction in the number of problems for evaluation. Problems 2, 6d, 7, and 8(n=9) were chosen. These were more difficult and best demonstrated differences in problem solving ability.

The methods for solving these problems were evaluated in two ways. First, it was important to consider whether or not a method could obtain a solution to all of the attempted problems. Second, the number of function evaluations to reach specified values of the function was compared for each problem. Any acceptable solution method first had to solve all the problems that Rosenbrock's method could solve, and then had to do it in as many or fewer function evaluations. The rate of progress toward the solution was also important. If two methods obtained roughly the same solution in a thousand trials, but one obtained a function value of 10^{-9} in 300 trials while the other was closer to 10^{-3} at 300 trials, the first might be more advantageous for getting near-optimality in a small number of trials.

During the First Phase, all eight digits of the solution point and the function value were printed and compared. This is definitely not typical of the literature comparing these methods, but was done to provide the greatest amount of information and to demonstrate whether or not true minimums, within the accuracy to which a computer can represent numbers in single precision format, could be obtained. To show this significance, Rosenbrock's method solved Problem 1 in 22 stages (584 trials) yielding $\bar{x}$ = (5.0000000, 5.0000000) and $F(\bar{x})$ = .00000000 (a true computer zero). The values at the twenty-first stage (537 trials) were $\bar{x}$ = (4.9999999, 4.9999999) and $F(\bar{x}) = 1.5789839 \times 10^{-15}$. In his book, Rosenbrock reported only four place data, which, in this research, was obtained after ten stages (134 trials) with $\bar{x}$ = (5.0003321, 5.0001953) and $F(\bar{x}) = 4.9615750 \times 10^{-8}$. Within a few trials, this is equivalent to the performance reported by Rosenbrock. The differences are attributed to the use of different computers.

Rosenbrock's Method

The first task of this phase was to establish baseline performance data. This involved programming Rosenbrock's algorithm and running it with the test problems. A subroutine approach was taken. All of the actual search logic was written into a special subroutine. The main program contained the read statements, initialization instructions, the direction rotation commands, printing statements, and tests for termination. A separate function, which contained all of the test functions and logic to indicate which one was being used, was subject to call by the main program during initialization and by the search subroutine during the minimization process. This approach permitted replacement of only the search subroutine by the other methods and reduced the probability that errors in the programming process confused the actual results. Later it was necessary to re-write the part of the main program for direction rotation to incorporate Palmer's method.

The basic Rosenbrock method solved all of the test problems successfully. True minimums were obtained for Problems 1, 2, 3, 6a, 6b, and 6c. Final function values for the remaining problems ranged between 3.64×10^{-12} and 1.39×10^{-17} when the desired value was 0. This resulted in 6-place or greater accuracy for the components of $\bar{x}$, except for Powell's function of four variables (Problem 7). The best solution to Problem 7 came after 80 stages (6430 trials) and was $F(\bar{x}) = 4.8405 \times 10^{-14}$ where $\bar{x} = (.0003782, -.0000378, .0002034, .0002034)$. This is exactly the form predicted for $\bar{x}$. This relationship became evident after 300 trials. More than 9000 trials were attempted, but no significant improvement was made after the 6430th.

Palmer Modification

Two other methods involving the basic Rosenbrock search subroutine were tested. The first was using the Palmer method of changing directions. This method changed the results slightly, sometimes arriving at the solution more quickly while at other times not getting the same degree of accuracy or being slower. With this set of problems, it appears that the Gram-Schmidt orthonormalization process is preferable to the Palmer method, even though the latter was more than adequate in comparison for all but Problems 6c and 7.

Gradient Approximation

A second modification to the main program computed a gradient approximation at the initial point to determine the first set of directions to be searched, instead of using the coordinate axes. It was felt that these directions would provide more progress in the first stage, thus reducing the total number of trials. Recall that the Rosenbrock method tries to align the first direction, $\bar{v}_1$, along the direction of greatest progress during the previous stage. Thus, for a cost of n trials instead of a minimum of 2n and probably more, an approximation to the direction of steepest descent could be obtained. This was done with only the addition of a few instructions and did not add any new variables. The results of this effort are mixed, as could be expected from the variety and difficulty of the problems. Greater initial progress was not a general trait.

A tabulation of the results for the basic Rosenbrock method, the Palmer modification, and the Rosenbrock search with initial gradient directions is presented in Table 1. The trial numbers were chosen to

Table 1. Function Value After i Trials

Legend: R = Rosenbrock's Method
RP = R with Palmer's Method for Computing Directions
GR = R with Initial Gradient Approximations

Notation: 3-07 = $3. \times 10^{-7}$

Problem 1

i	R	RP	GR
0	14.4	14.4	14.4
50	.006	.001	2-03
100	2-06	9-08	1-06
150	5-08	6-09	3-08
200	2-09	2-10	4-09
300	1-11	4-13	5-13
400	2-13	2-14	6-15
500	6-15	2-15	0
600	0	2-15	

Problem 2

i	R	RP	GR
0	24.2	24.2	24.2
50	3.24	3.16	4.38
100	.878	1.16	1.79
200	5-04	.078	.006
300	5-07	3-06	1-06
400	5-10	8-11	1-06
600	8-11	4-13	7-11
800	3-13	8-14	2-15
900	0	8-14	2-15

Problem 3

i	R	RP	GR
0	.367	.367	.367
10	.002	.002	.072
20	3-05	3-05	.002
50	7-06	7-06	7-07
100	3-08	3-08	3-09
200	3-10	1-13	2-12
300	2-13	4-15	2-14
400	5-16	6-17	0
450	0	6-17	

Problem 4

i	R	RP	GR
0	1021	1021	1021
25	.593	1.88	12.2
50	.542	1.69	.474
100	.178	.013	.012
200	2-05	5-05	1-06
300	5-10	1-05	8-08
400	9-12	8-06	2-10
500	5-12	3-06	2-11
800	4-12	9-11	2-11

Problem 5

i	R	RP	GR
0	20.9	20.9	20.9
25	7-04	7-04	.118
50	4-05	4-05	.001
75	2-08	2-08	1-05
100	4-09	7-09	1-05
150	7-11	2-09	3-09
200	1-12	2-11	2-10
250	2-13	4-14	5-13
325	4-14		

Problem 6a

i	R	RP	GR
0	3.06	3.06	3.06
50	.122	.004	.054
100	2-06	6-05	1-04
200	1-08	2-10	2-05
300	2-10	8-13	8-11
400	6-12	3-14	2-11
500	4-13	5-16	3-12
600	4-14	1-16	2-12
700	0	0	2-12

Table 1 - continued

Problem 6b

i	R	RP	GR
0	2.09	2.09	2.09
50	.015	.014	.036
100	5-05	2-05	2-05
200	4-10	1-08	1-08
300	2-11	1-09	4-12
400	6-12	9-11	8-13
500	4-12	2-11	2-14
600	9-15	4-13	5-16
700	0	6-16	2-16

Problem 6c

i	R	RP	GR
0	19.6	19.6	19.6
50	.145	.145	.127
100	.015	.144	1-04
150	2-08	4-04	1-06
200	2-09	9-05	2-07
250	9-11	2-06	2-07
300	2-13	2-06	6-08
350	2-16	1-06	8-10
400	0	1-07	2-10

Problem 6d

i	R	RP	GR
0	1.81	1.81	1.81
50	.106	.090	.020
100	3-03	8-06	7-04
150	8-09	8-06	2-07
200	3-12	5-09	2-07
250	2-13	5-10	5-08
300	4-14	2-10	8-09
350	3-14	1-10	6-13
400		4-13	6-13

Problem 6e

i	R	RP	GR
0	.808	.808	.808
50	.050	2-05	2-04
100	1-05	2-07	2-06
150	7-09	9-11	7-07
200	6-11	1-12	1-11
250	2-11	4-13	7-12
300	4-13	6-14	1-15
350	8-14	2-14	1-15
400	7-15	2-14	1-15

Problem 7

i	R	RP	GR
0	215.	215.	215.
100	.871	.474	.837
200	.003	1-05	.001
500	7-07	3-07	2-05
800	1-07	2-07	1-08
1500	4-09	1-10	2-09
3000	6-10	1-10	4-13
5000	5-10	1-10	1-17
6400	5-14	1-10	1-17

Problem 8 (n=9)

i	R	RP	GR
0	.029	.029	.029
100	.010	.010	.011
200	.006	.010	.009
500	2-04	9-05	1-04
800	9-06	2-06	1-05
1500	1-07	3-07	2-06
2500	1-09	9-10	2-07
3500	1-11	1-13	1-08
5000	1-13		3-12

best depict the progress of the Rosenbrock method and therefore might tend to favor that method. As a general summary of the data in Table 1, it can be concluded that the Rosenbrock-Palmer method most frequently had the minimum function value for the first half of the trials, while the Gradient-Rosenbrock approach performed as well as or better than the others most frequently at intermediate trials, contrary to the purpose of attempting the gradient approximation. The basic Rosenbrock method was equivalent to the others at intermediate trials and was superior most frequently at the end of the trials shown.

The Rosenbrock-Palmer method did solve problem 6c to a function value of 1.67×10^{-16} in 976 trials, compared with less than half as many for the basic Rosenbrock method. The same method converged very slowly toward a solution to Problem 7, making reductions in the seventh significant digit of the function value from trial 1993 to 9999. The final function value was $1.2536764 \times 10^{-10}$, not significantly different from the four values reported in Table 1, with $\bar{x}$ = (-.0027, 00027, -.0014, -.0014). The Gradient-Rosenbrock method solved the same problem to an $\bar{x}$ vector of (.000046, -.0000046, .000012, .000012) in 3908 trials, which was the best solution to this problem.

Approximate computation times for solving the problems in Table 1 were 33, 40, and 35 seconds for the Rosenbrock, Rosenbrock-Palmer, and Gradient-Rosenbrock methods, respectively. The number of trials for each method varied, depending on the type of termination, with the Rosenbrock-Palmer method generally attempting more trials. These times were taken from "real time clock interrogations" on the Univac 1108 computer, and are not as reliable as the charge time reported on the

summary at the end of a run. Because of the multi-processing capability of the computer, real time can be interrupted during a run. Problems 1-5 can be solved in less than a second each.

Secondary Objective

During this initial part of the First Phase, it became apparent that there would be no effective way to develop an alternate stopping criteria. Solving these problems to eight digit accuracy necessarily forced $\|\bar{\alpha}_1\|$ to a small value, but the reinforcement from the γ parameter did not appear by the time an optimum was reached. Listed below are the final six stages to the Rosenbrock solution of Problem 2:

Stage	Trial	$F(\bar{x})$	$\|\bar{\alpha}_1\|$	γ
35	666	1.58×10^{-11}	6.1×10^{-6}	.031
36	712	1.27×10^{-12}	7.2×10^{-6}	.007
37	760	3.01×10^{-13}	1.8×10^{-6}	.013
38	808	1.35×10^{-14}	7.6×10^{-7}	.031
39	858	2.00×10^{-15}	9.6×10^{-8}	.124
40	908	0	9.6×10^{-8}	.124

Since these methods will continue to function although optimality has been reached, the proper correlation could eventually appear. In Problem 8(n=10), the Gradient-Rosenbrock method produced six consecutive stages with no change in $F(\bar{x})$, all $\|\bar{\alpha}_1\|$ less than 10^{-7}, and γ's .95, .97, .90, .99, .70, and .90, but these stages consumed 1636 trials. It is not worthwhile to permit these useless trials for an automatic stopping criterion that may or may not develop and will always require time consuming tests at each stage.

Successive stages with identical function values do not provide an adequate criterion, either. This case appeared infrequently, but it

can happen that no progress will be made during one or more stages followed by progress toward a solution during later trials.

The only useful approach to the automatic termination problem turned out to be a test on the function value itself. This requires previous knowledge of the solution, which is available in an experiment involving several runs of the same problems, but certainly is not normally known. By inserting this test and increasing the trial maximum, the program stopped when a true and/or acceptable minimum was reached, but did not stop prematurely on a trial maximum developed from the basic Rosenbrock data. In one special class of problems, solving a system of nonlinear equations, the minimum is known (since the function value is the sum of the squares of the residuals), and termination by this test is appropriate. It was a simple matter to negate the purpose of this test if the minimum were not specified by the user.

Alternate Methods

Three separate methods for changing the Rosenbrock search logic within its basic nature were attempted. The first was called the Sequential Golden Section. For convenience, the subroutine deck was labelled SG1, and reference to that label will be used here. The second approach was called Sequential Fibonacci Search, labelled SF1. The last was the Simultaneous Fibonacci Search, or SF2.

Sequential Golden Section Search

The SG1 method conducted a golden section search along a predefined portion of the line through $\bar{x}$ in the $\bar{v}_1$ direction. When a minimum along $\bar{v}_1$ was obtained, it was defined as the new $\bar{x}$ and a search was made along $\bar{v}_2$. After a single search had been made along

each direction, new directions were computed from the old and factors for each direction representing the net progress made in that direction.

The interval of search was defined to be $\bar{x} \pm a\bar{v}_i$ for direction i, where $a = \max(\|\bar{\alpha}_1\|, .001)$. $\|\bar{\alpha}_1\|$ was set to 10 for the first stage. Choosing this initial value allowed the search to expand rapidly should the initial value of $\bar{x}$ be a poor approximation of the optimum. The .001 minimum value kept the length of the line to be searched long enough as not to be wasteful. Had this approach been more successful, these parameters could have been further studied and possibly improved.

The tolerance criterion for the golden section search (the final length of the interval of uncertainty) was made a decreasing function of increasing stage numbers. For the first stage, the golden section searches stopped when the length of uncertainty reached one-tenth. At subsequent stages, the tolerance value became 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, and finally 10^{-6} for all further stages. This last value would have been inadequate for eight digit accuracy for most of the test problems, but proved too restrictive for continued progress on some of the problems.

The SG1 subroutine required 436 memory locations compared to 218, exactly half as many, for the basic Rosenbrock subroutine. The arrays in each program allowed only ten components in $\bar{x}$ and each $\bar{v}_i$. The Rosenbrock method requires two arrays not in common with the rest of the program. SG1 was programmed with five such arrays. Both the number of required locations and possibly the number of arrays in SG1 could be reduced, but the relative size comparison remains valid.

The main difficulty with SG1 arose when it failed to continue progressing toward optimality after some initial success. In an

attempt to solve Problem 2, it reached a function value of .005 in 85 trials (compared with .878 for the Rosenbrock method in 100 trials), but made only minute progress through trial 909 (the iteration maximum) with no staging after trial 132. The choice of $\|\bar{\alpha}_1\| = 10$ for the first stage influenced the rapid initial progress since Problem 2 is bimodal along the x_1 coordinate axis and the program chose the more fortunate minima. Solutions to other problems produced a similar lack of intermediate and final progress.

Sequential Fibonacci Search

The second alternative to the basic Rosenbrock method was SF1, the sequential Fibonacci search. The method was identical to SG1, except that the points along the orthonormal directions were selected in the manner of a Fibonacci one-dimensional search. One cycle through the directions constituted a stage.

The length of the line to be searched was also $2(\|\bar{\alpha}_1\|)$, but in SF1 the point $\bar{x}$ was selected to be one of the initial points for function evaluation, not the midpoint of the line. This saves one function evaluation for each direction per stage, since $F(\bar{x})$ is F_{best} at the beginning of each search. The penalty for this approach is biasing the region of search in the positive direction of each of the orthonormal directions. To illustrate, assume that $\bar{x} = \bar{v}_1 = (1,0)$, $\|\bar{\alpha}_1\| = 1$, and the Fibonacci constant is 144 (the eleventh Fibonacci number). The first search would be along the x_1 axis from 17/72 (.236) to 161/72 (2.236). The first experiment would be placed at (106/72 ,0) ((1.47((. At the conclusion of this search, the interval of uncertainty would be

$$\frac{2(\|\bar{\alpha}_1\|)}{F_{11}} = \frac{2(1)}{144} = .0139$$

As the $\|\bar{\alpha}_1\|$ decreases, as it will as optimality is approached, the interval of uncertainty will decrease.

There is obviously a second parameter, besides the initial and minimum values allowed for $\|\bar{\alpha}_1\|$ in regard to defining the search, for investigation in SF1. It is the choice of the Fibonacci constant. As stated above, the eleventh Fibonacci number will reduce the interval of uncertainty to 1/144 of the original. The sixth through the 16th Fibonacci numbers (13, 21, 34, 55, 89, 144, 233, 377, 610, 987, and 1597) were attempted on Problem 1. The ninth (55) in the sequence produced the best result ($F(\bar{x}) = 4.46$ in 300 trials, compared with $F(\bar{x}) = 2 \times 10^{-11}$ for Rosenbrock's method). The next best solution came from the 13th Fibonacci number (377) followed by 15, 16, 10, 14, 12, 11, 8, 7, and 6.

With the SF1 method, the number of trials per stage is fixed at n times the ordinality of the Fibonacci number used. This may have a bearing on the results presented in the preceding paragraph, since smaller Fibonacci numbers will be tested with more sets of directions. The three smallest numbers produced the poorest solutions, although the fourth was the best. The lack of ordering above was not investigated further with SF1.

SF1 had a computer storage advantage over SG1, but did not require as few locations as the basic Rosenbrock method. The subroutine needed 313 locations, some of which were instructions to compute the numbers of the Fibonacci sequence, which were variable for testing

purposes. SF1 required one more array than the basic Rosenbrock routine.

SF1 suffered from the same problem as SG1. The method made progress toward the solution points, but it was very slow and was never allowed to consume the excessive number of trials to reach desired solution points. The problem of slow progress perpetuated itself when $\| \bar{\alpha}_1 \|$ became small and tended to remain small.

Simultaneous Fibonacci Search

SF2 was more successful than either SG1 or SF1 during its first tests and for that reason received much more attention, examination, and modification. SF2 is more closely related to the logic of Rosenbrock's method, but copied the Fibonacci search scheme of SF1. SF2 was called a simultaneous Fibonacci search because the direction of search was changed after each trial. The first two experiments were placed exactly the same as in SF1, using the given $\bar{x}$ as one of the points. Then, taking the better of these two points as the new $\bar{x}$, the second trial of the stage was placed along direction $\bar{v}_2$ the same distance from the current $\bar{x}$ as the first was from the original $\bar{x}$. All of the regions of search contained the same positive bias that appeared in SF1. This bias is logical in the long run, since the first of the directions tended to align itself along the direction of steepest descent.

Figure 2 is a functional flow diagram of SF2. Some of the details are omitted (such as saving the cumulative progress in each direction) to aid clarity. Note that the T array contains all the information necessary to keep track of the point selection in each

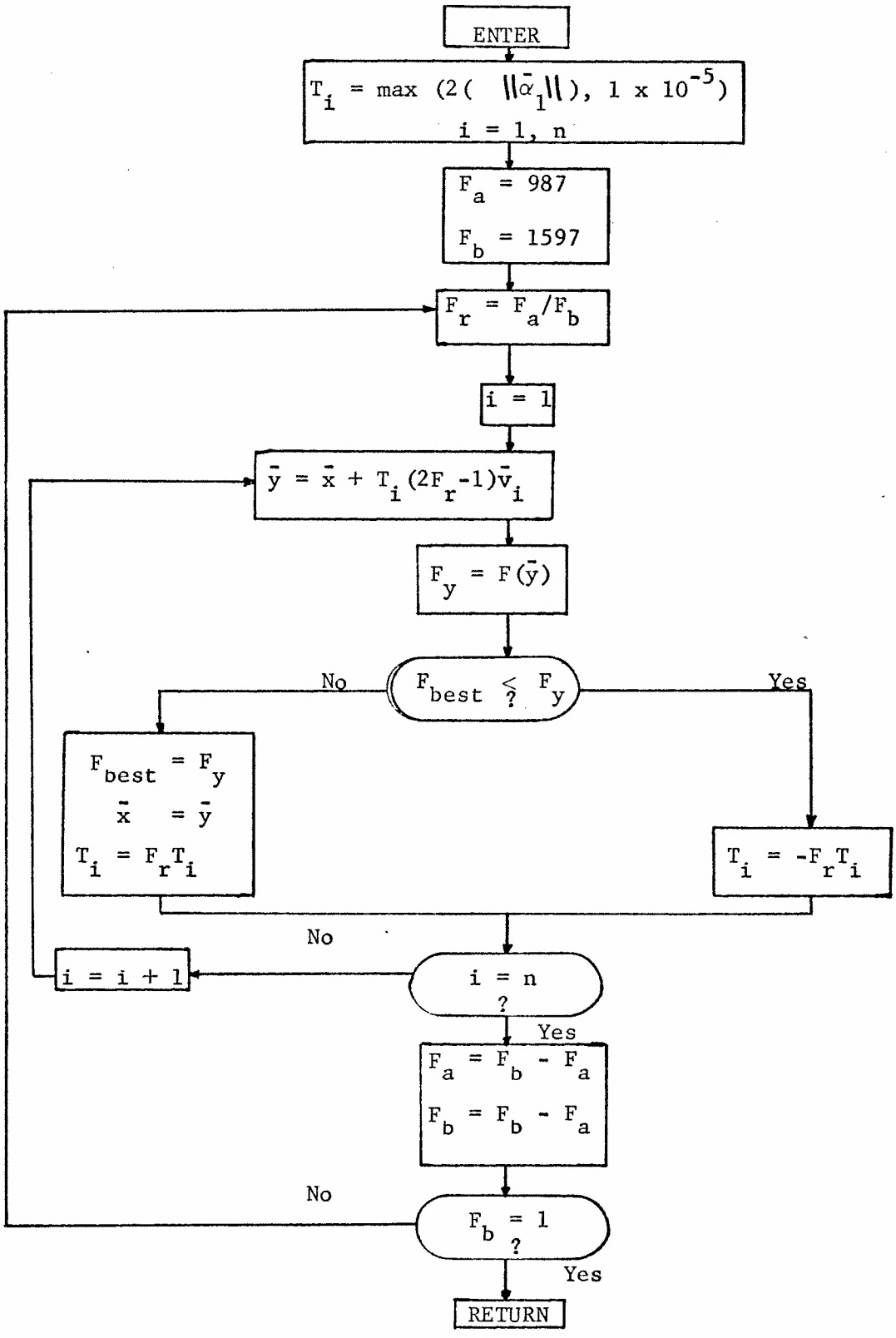

Figure 2. SF2 Flow Diagram

direction. It is properly modified after each trial to reject that portion of the region of search known not to contain the optimum of the (assumed) unimodal function. The figure assumes that the 16th Fibonacci number was selected and the length of the line was defined to be $2(\|\bar{\alpha}_1\|)$

The SF2 subroutine had the same variables as SF1. The particular Fibonacci number could be varied. The initial value of $\|\bar{\alpha}_1\|$ affected the progress made during the first stage. A minimum value for the length of the search region had to be specified to insure that the region was not too small. It was also discovered that the Fibonacci number could be changed from stage to stage, making possible the evaluation of sequences of Fibonacci numbers. Although many sequences were tried, they generally began with smaller numbers and increased with each stage to a maximum value.

The variety of conditions for evaluating SF2 forced the consideration of only four problems to keep the total computer time within the limits for research of this type. These problems were 2, 6d, 7 and 8(n=9). The difficulty of these problems dictated that prime consideration be given to whether or not the methods could effectively solve them within a prescribed trial maximum, not comparing intermediate progress. The trials were limited to 1500, 500, 4200, and 3000, respectively. These limitations did not bias the results toward any method since they were determined after testing several methods with more liberal maximums.

In this search philosophy, it frequently occurs that no progress is made in one or more of the n directions. This situation creates havoc with the Gram-Schmidt orthonormalization process. The result

is directions which are not orthogonal and have zero components for all successive stages. Division by zero or errors in the square root routine may also occur. A solution to this problem is the Palmer method of direction rotation, which can accept some zero progress multipliers, causing only a slight loss of orthogonality, but without permanent repercussions. However, the sum of these multipliers squared must be nonzero, which cannot always be guaranteed. This forced the programmer to set these zero multipliers to small, non-zero values. This process destroys the relationship of the information intended for orthonormalization, but no other solution was found. The extent of the harm could not be measured, but the successful solution of the problems indicates that it was not severe.

One hundred and nine distinct versions of SF2 were tested on these four problems. Another was tested on three of them and abandoned. All but four of the 109 were designated PSF2, implying that the Palmer coordinate rotation was used. Fibonacci numbers from 13 through 28,657 (the sixth through the 22nd) and 121,393 (the 25th) were used. Initial values of $\|\bar{\alpha}_1\|$ were chosen to be 2.5, 1.25, 1., .5, .1, and .05. The minimum length of the search region was restricted to 10^{-3}, 10^{-4}, 10^{-5}, and 10^{-6}. Multiplication will reveal that all of the possibilities were not tried, but it is believed that enough were attempted to bracket the most successful.

Presentation of all the results would require several pages of tables, which will be omitted. To summarize, the best method tried was PSF2 with the eleventh Fibonacci number (144), $\|\bar{\alpha}_1\|$ initially equal to 1, and the minimum search length greater than 10^{-5}, but this

combination was not better than any of the three versions of the basic Rosenbrock method. At termination, the combination reached function values less than 10^{-14}, 10^{-10}, 10^{-7}, and 10^{-12}, respectively, compared with 0, 10^{-13}, 10^{-9}, and 10^{-9} for Rosenbrock's method. The conclusion that one method was better than another was reached by adding the exponents of the final function values (considering zero to be 10^{-16}) and comparing the sum.

It was interesting to watch the performance of the different Fibonacci numbers with the other variables held constant. Considering the above combination, the 13th, 14th, and 21st Fibonacci numbers (377, 610, 17,711) almost solved the problems as well as the eleventh. Problem 6d proved to be the strictest test. The 21st solved it to only $F(\bar{x}) = .0538$ after 500 trials, although it solved the remaining problems better than the eleventh. For other variable values, the performance of the Fibonacci numbers was not identical, but the eleventh was most frequently the best.

Twenty-four of the problems involved sequences of Fibonacci numbers. In some cases, they were repeating sequences of two, four, or eight numbers. In other cases, the first eight stages were different and the eighth was retained for the remaining stages. The better performing sequences, in terms of Fibonacci ordinality, were (10, 11, 12, 13, 14, 15, 16, 17, 10, 11, 12, . . .), (21, 17, 17, 14, 14, 13, 13, 11, 11, 11, 11, . . .), and (10, 11, 12, 13, 14, 15, 16, 17, 17, 17, . . .). These did nearly as well as the eleventh, above. One of the sequences was tried with initial gradient directions, but the results were far less fortunate than those obtained with Rosenbrock's basic method.

One of the sequences, (7, 8, 9, 10, 12, 14, 16, 18, 18, . . .), solved Problems 1 and 3 to $F(\bar{x}) = 0$ in 223 and 134 trials. While this performance was rather remarkable, the same method did very poorly on Problems 2, 6d, and 6e. Performance like this helped indicate which four problems should be chosen for the final evaluation. Only one sequence found an undesirable local optimum in Problem 3, when it began with an initial $\|\bar{\alpha}_1\| = 2.5$, apparently too large a value when used in combination with that sequence. This is a characteristic of the problem as much as the method, and even Rosenbrock's initial step of .1 could be too large for certain problems.

The storage requirement for SF2 was about 275 locations, 57 more than the Rosenbrock routine. The saving by using Palmer's method is not counted since it is not included in the search subroutine.

Rosenbrock-Fibonacci-Palmer Combination

Within the previously established framework of this study, there was only one additional method to attempt. That was to combine the Rosenbrock and Fibonacci (SF2) methods with the Palmer method of coordinate rotation, allowing the Rosenbrock and SF2 methods to alternate stages. There is one significant advantage to this approach. If the Rosenbrock method is used for the first stage, an initial value for $\|\bar{\alpha}_1\|$ need not be specified. In fact, the first value for SF2 will be computed from the first Rosenbrock stage and will be indicative of the amount of progress that can be made with the given objective function. Each subsequent value will continue to be indicative of the progress made during the previous Rosenbrock stage.

Combining the two methods is not as dreadful as might be imagined.

By sharing certain array locations, since SF2 requires no more arrays than the basic Rosenbrock method, and other programming tricks, the storage requirement for both instructions and variable storage was held to 352 memory locations, just over half again as large as Rosenbrock's requirement. Further reduction is possible by removing diagnostic statements included for testing purposes. This requirement will not expand any faster for the combination than it would for Rosenbrock's method when solving larger problems.

The combination experienced much better success than expected. It was attempted with the eleventh and 13th through 18th Fibonacci numbers. Fourteen, 15, 16, and 17 solved the four problems to a slightly better combined degree of accuracy than the basic Rosenbrock methods, with 16 being the best of the group. Table 2 lists the results for RFP16, the best method, for all the problems presented in Table 1. At nearly half the comparison points in the table, RFP16 performed as well as or better than the Rosenbrock method. In the final four problems, the Rosenbrock method did better on Problems 2 and 6d, while RFP16 held a larger comparative lead on Problems 7 and 8. Not shown in the table are the results at the maximum number of trials allowed, 1500, 500, 4200, and 3000, respectively. It was at these points where the final comparison was made and reported above. The approximate time for RFP16 to solve these problems was 29 seconds, which is about the same as for the basic Rosenbrock method. The difference can be attributed to improved termination criteria and quick solutions to four of the problems.

Table 2. Function Values for RFP16

Notation: 3-07 = 3×10^{-7}

* Performed better than Rosenbrock's basic method.

Problem 1		Problem 2		Problem 3		Problem 4	
Trial	$F(\bar{x})$	Trial	$F(\bar{x})$	Trial	$F(\bar{x})$	Trial	$F(\bar{x})$
0	14.4	0	24.2	0	.367	0	1021
50	.136	50	4.16	10	.015	25	.009*
100	1-06*	100	3.45	20	1-04	50	3-05
150	1-08*	200	.867	50	3-08*	100	1-08*
200	9-13*	300	.076	100	2-14*	200	9-09*
300	2-14*	400	.034	200	0 *	300	8-10
400	0 *	600	8-09	300	- *	400	4-12*
500	- *	800	2-16*	400	- *	500	- *
600	-	900	2-16	450	-	800	-

Problem 5		Problem 6a		Problem 6b		Problem 6c	
Trial	$F(\bar{x})$	Trial	$F(\bar{x})$	Trial	$F(\bar{x})$	Trial	$F(\bar{x})$
0	20.9	0	3.06	0	2.09	0	19.6
25	8-04	50	.957	50	.014*	50	.149
50	1-08*	100	.007	100	.002	100	.149
75	3-10*	200	4-09*	200	1-04	150	.149
100	1-13*	300	3-13*	300	1-05	200	.149
150	8-14*	400	2-14*	400	8-12	250	4-05
200	4-14*	500	2-14*	500	2-12*	300	3-07
250	- *	600	2-14*	600	0 *	350	1-12
325	-	700	2-14	700	-	400	4-13

Problem 6d		Problem 6e		Problem 7		Problem 8 (n=9)	
Trial	$F(\bar{x})$	Trial	$F(\bar{x})$	Trial	$F(\bar{x})$	Trial	$F(\bar{x})$
0	1.81	0	.808	0	215.	0	.029
50	.108	50	.054	100	9.53	100	.009*
100	.006	100	.038	200	.076	200	.009
150	1-05	150	9-04	500	2-06	500	7-05*
200	7-08	200	6-06	800	9-07	800	3-05
250	2-09	250	9-11	1500	1-07	1500	4-10*
300	2-10	300	6-12	3000	1-09	2500	4-12*
350	2-10	350	7-14*	5000	9-14*	3500	3-13*
400	2-10	400	7-14	6400	9-14	5000	-

Summary

During the First Phase, two modifications were tried with the basic Rosenbrock search logic and three alternate searches were examined. The two modifications, Palmer's method of orthogonalization and initial gradient approximations, did not provide any increased capability for general problem solving, although each did experience better success with some individual problems. The three alternate methods, the Sequential Golden Section, the Sequential Fibonacci Search, and the Simultaneous Fibonacci Search, did not produce an acceptable problem solving method. The Simultaneous Fibonacci Search, with its parameters set to optimal values, did exhibit an ability to solve some problems much faster than the Rosenbrock method. No way was found to predict this performance and capitalize on its benefits.

The combination method, RFP16, which involved alternating stages of the Rosenbrock search and the Simultaneous Fibonacci Search with Palmer's orthonormalization process, proved to be equally as effective as Rosenbrock's method alone at solving these test problems. Each of these two methods solves some problems more accurately and/or quickly than the other. Neither was stopped by any of these problems. The Rosenbrock method has an efficiency advantage if extra variables (not arrays) and instructions are considered, but with the computer capability of today, these few extra storage requirements will never hinder the program. Since RFP16 was the creation of this research, it will be used as the basic optimization technique for the next phase.

CHAPTER IV

SECOND PHASE - CONSTRAINED PROBLEMS

Objectives

The objective of the second phase was to take the resulting method from the first phase, which was the Rosenbrock-Fibonacci-Palmer combination, and to examine it for compatibility and effectiveness at solving constrained nonlinear programming problems. The final goal was a pre-programmed solution procedure to provide users with an effective optimization tool.

The five techniques which were presented in Chapter II formed the basis of the present examination. The Rosenbrock approach required that a method for considering equality constraints be developed, while the other methods were sufficiently general to permit both types of constraints. The greatest challenge to making these approaches compatible with the combined unconstrained program was the method of introducing the iterative nature of the constraint techniques, except for Rosenbrock's, into the trials and stages of the solution method. Success or failure in meeting this challenge largely determined the effectiveness of the method for dealing with constraints.

Test Problems

There are not as many constrained problems in the literature as there are unconstrained problems. This is especially true of two-,three-, and four-dimensional problems. Since constrained problems involve many

more calculations than unconstrained problems, three relatively small constrained problems were chosen for initial comparisons and a fourth problem with 24 components of $\bar{x}$ was chosen for comparison of those methods that successfully conquered the first three.

Problem C-1

$$\text{Maximize } F(\bar{x}) = x_1 x_2 x_3$$

$$\text{subject to: } 0 \leqq x_i \leqq 42, \quad i=1,2,3$$

$$x_1 + 2x_2 + 2x_3 = 72$$

Initial Point: $\bar{x} = (6,6,6)$ $\qquad F(\bar{x}) = 216$

Solution Point: $\bar{x} = (24,12,12)$ $\qquad F(\bar{x}) = 3456$

This is the Post Office Parcel problem presented by Rosenbrock in 1960 (14) and used by Box (1)(2). The object is to maximize the volume of a rectangular carton subject to length and girth restrictions. The indicated initial point is slightly different from those suggested by Rosenbrock, but Box reported little or no initial point dependence for Rosenbrock's procedure or his modification to it. At the optimal point, only the last constraint is active.

Problem C-2

$$\text{Minimize } F(\bar{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 + 2x_3^2$$

subject to:

$$-x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geqq 0$$

$$-x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geqq 0$$

$$-2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geqq 0$$

Initial Point: $\bar{x} = (0,0,0,0)$ $\quad F(\bar{x}) = 0$

Solution Point: $\bar{x} = (0,1,2,-1)$ $\quad F(\bar{x}) = -44$

Problem C-2 was introduced by Rosen and Suzuki (13), although this particular version was taken from Kowalik and Osborne (7). At the solution point, the last constraint is active.

Problem C-3

$$\text{Minimize } F(\bar{x}) = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + x_3^2 + 2x_1x_2 + 2x_1x_3 + 2x_2^2$$

$$\text{subject to: } x_i \geq 0, \quad i=1,2,3$$

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

Initial Point: $\bar{x} = (.1, .1, .1)$ $\quad F(\bar{x}) = 7.29$

Solution Point: $\bar{x} = (4/3, 7/9, 4/9)$ $\quad F(\bar{x}) = 1/9$

Problem C-3 is due to E. M. L. Beale, but was also taken from Kowalik and Osborne (7). As in the previous problems, only the final constraint is active.

Problem C-4

$$\text{Minimize } F(\bar{x}) = \sum_{j=1}^{24} c_j x_j$$

subject to:

$$(x_{i+12}/M_{i+12})/S_v - (V_i/P)(x_i/M_i)/S_L = 0, \quad i=1,\ldots,12$$

$$x_1 + x_2 + \ldots + x_{24} = 1$$

$$\sum_{i=1}^{12} \frac{x_i}{A_i} + [(.7302)(14.7)T/P]S_v - 1.671 = 0$$

$$x_i \geq 0, \quad i=1,\ldots,24$$

$$.1 - (x_1 + x_{13})/S_x \geq 0$$

$$.3 - (x_2 + x_{14})/S_x \geq 0$$

$$.4 - (x_3 + x_{15})/S_x \geq 0$$

$$.3 - (x_7 + x_{19})/S_x \geq 0$$

$$.6 - (x_8 + x_{20})/S_x \geqq 0$$

$$.3 - (x_9 + x_{21})/S_x \geqq 0$$

where $S_x = \sum_{i=1}^{24} x_i$, $S_L = \sum_{j=1}^{12} x_j/M_j$, $S_V = \sum_{j=13}^{24} x_j/M_j$, and the values of the c, M, A, and V arrays and the constants T and P are specified.

Initial Point: $\bar{x} = (.04, .04, .04, \ldots, .04)$ $\quad F(\bar{x}) = .14696$

Solution Point: $\bar{x} = (.013, .091, .115, 10^{-7}, 10^{-7}, 10^{-7}, .069, 10^{-4},$
$10^{-5}, 10^{-8}, .017, .029, .079, .146, .266, 10^{-7},$
$10^{-7}, 10^{-8}, .173, 10^{-4}, 10^{-6}, 10^{-6}, .001, .001)$

$F(\bar{x}) = .058106$

The objective function to this problem from Paviani and Himmelblau is linear, which is an allowable diversion from true nonlinearity (10). Due to its 24 variables and 44 constraints, the problem is described by Paviani and Himmelblau as being difficult enough to challenge any non-linear programming solution procedure. The solution point given above was obtained from a modification to the sequential simplex algorithm for constrained problems in 13.2 minutes on a CDC 6600 computer. This amount of time is a luxury not available to this research. Therefore, Problem C-4 was chosen only to be an additional test for the most successful approaches, and even then, time in excess of one minute could not be allowed. The results indicate that this time was sufficient to signify only if the problem was too difficult to prohibit progress.

Test Procedures

Each of the transformations for solving constrained problems was entirely programmed within the function computing $F(\bar{x})$, except for a small number of instructions in the main program and search subroutine

which required modification due to the nature of the transformation. For example, Rosenbrock's approach required a special entry into the function for the initialization of F_{best}, and provision for direct return to the failure path of the flow diagram if the point $\bar{x}$ were infeasible. Once the peculiarities for each approach were identified, these instructions were inserted and required no further attention.

The components of $\bar{x}$ and the function value were only printed to seven digits to allow the insertion of a new parameter in the same line of print. This new parameter was the value of the transformation function, which is the value to be minimized. Printing the true function value and the transformed value adds clarity to a maximization problem, since the negative need not be the only value to appear, and allows comparison of these two values to determine the effect of the constraints at any given stage.

The five transformations (Rosenbrock's approach, the Heaviside unit step function, the created response surface technique, penalty functions, and Paviani and Himmelblau's criterion function) were each tried with the three smaller problems. The more successful of these methods attempted to solve the Paviani and Himmelblau problem (C-4). Effective solution, rather than efficient, was the most important consideration, since the solution itself was far more important than differences of a few trials.

Results

Rosenbrock's Approach

The Rosenbrock formulation was the most difficult to program and would require the most preparation for solving an individual problem.

The approach, as described in Chapter II, is an interior method which leaves the function value unchanged until a boundary zone near constraining values is entered. Inside the zone, the objective function is penalized by a cubic function of the distance into the zone. The constraints are put in the form

$$L_i(\bar{x}) \leq g_i(\bar{x}) \leq U_i(\bar{x})$$

where the lower and upper limits may be functions of $\bar{x}$, but normally are constant. The boundary zone width is defined by

$$b_i = [U_i(\bar{x}) - L_i(\bar{x})]\ 10^{-4}$$

for the initial values of $\bar{x}$, and because of this computation, both the upper and lower limits must be specified, even though one may not be meaningful. For a standard inequality constraint, the upper limit must be intelligently chosen to prevent the boundary width from being too excessive. This was not a great problem for any of the test problems, but it did increase the manual preparation time.

The 10^{-4} multiplier in the boundary width equation was the optimal value for eight digit computers, according to Rosenbrock (15). There may be a considerable amount of dependence on the particular problem because several values were tested on Problem C-1, with 10^{-8} being superior to 10^{-6}, 10^{-5}, 10^{-4}, and 10^{-3}. The ranges for the constraints, $U_i - L_i$, for Problem C-1 are 42 and 72. Problems with ranges either much larger or much smaller probably would require a different multiplier.

Rosenbrock took the first n constraints to be restrictions on the value of each component of $\bar{x}$. For most constrained problems, this is not

wasteful and was appropriate for three of the four test problems. However, for a general problem solving technique, which might be called on to solve unconstrained problems, this is a nuisance and should be avoided.

Rosenbrock's approach was programmed to require a second set of input data for each problem. This data was for the upper and lower limits on each constraint, an integer specifying the total number of constraints, and the multiplier for the boundary zone width. This information could have been programmed into the function, but with several problems in the test function, this approach simplifies the amount of manual work.

With the border zone multiplier of 10^{-8}, the Rosenbrock approach with the Rosenbrock-Fibonacci-Palmer optimization method reached a function value of 3455.984 in four stages (133 trials) for Problem C-1. This value improved to 3455.999 in 16 more stages (up to 1000 trials). The final $\bar{x}$ vector was (24.00515, 12.00512, 11.99231), which is very close to the true value. Problem C-2 was solved to a function value of -43.95000 in 2000 trials and was still making slow progress. The $\bar{x}$ vector was (.008250994, .7631167, 2.030796, -1.072993), which doesn't yet have one digit accuracy even though the function value is close to optimal. In 1000 trials, the Rosenbrock approach solved Problem C-3 to $F(\bar{x}) = .1111860$, $\bar{x} = (1.330870, .7730718, .4480286)$. Comparing the above results to presentations in the literature, it appears that the solutions to Problems C-1 and C-3 are quite satisfactory and that the solution to Problem C-2 might eventually become acceptable after an excessive number of trials.

There were two reasons for attempting to solve Problem C-4 with Rosenbrock's approach. First, it solved the first three problems to an

apparently satisfactory degree. Second, since Rosenbrock did not discuss equality constraints, the method had to be examined for compatibility with equality constraints.

There seemed to be two logical approaches to adapting Rosenbrock's formulation for equality constraints. The first choice was to artifically set upper and lower bounds ($0 \pm r$, where r is set to a prescribed value and then made to decrease as a function of stage number). The other choice was to add the sum of the squares of the equality constraint equations, multiplied by an increasing function of the stage counter, to the objective function. Both approaches were tried, but due to the difficulty of Problem C-4, they can only be evaluated in terms of their effect on the equality constraints. For the first approach, changing the value of r at any stage makes it possible for a previously acceptable point to become unacceptably infeasible. This then requires that a direction exist to point back to the feasible region and that the initial step size be long enough to enter the region. The chance of this occurrence are very slim, so that the first method reached a point of infeasibility that could not be escaped. When this occurs, an auxiliary problem which minimizes the sum of the squares of the violated constraints can be solved for an acceptably feasible point. Then the original problem can be re-started from this new point. The best $F(\bar{x})$ produced by this method was .07115096 after 14,400 trials of finding a good feasible point with the auxiliary problem and then 2400 trials of Problem C-4 (only two stages were accomplished because the value of r became too restrictive). The $\bar{x}$ vector did not bear any relationship to the desired result.

The second approach, combining the sum of the squares of the equality constraints with the objective function, had no better success at solving Problem C-4, but did reduce the absolute value of the equality constraints equations to smaller values. The actual form of the new objective function was

$$F^*(\bar{x}) = F(\bar{x}) + 2^{istg} \sum_i e_i^2(\bar{x})$$

and since this is a function of the stage counter, F_{best} had to be re-initialized each time the stage counter was incremented. This re-initialization resulted in a loss of monotonicity in the sequence of function values. After 18,400 trials, 3600 of which were used to solve the auxiliary feasibility problem to generate a better point, $F(\bar{x})$ was reduced to .07049412. The components of $\bar{x}$ still did not bear any resemblance to the desired solution. These 18,400 trials consumed about 50 seconds of computer time.

Heaviside Unit Step Function

The Heaviside unit step function was judged to be not very compatible with the Rosenbrock-Fibonacci-Palmer combination method because of its dependence on the function value. The logic required to check $T_j(\bar{x}) = 0$ was not implemented, but j and f_j were changed every two stages, which made it very difficult to bracket the minimum with two successive f_j values. Although it may be possible to arrange the necessary compatibility between this approach and the optimization method, the interruption requires major revision of the basic search routine and the main program. This destruction of the combination method was considered undesirable in the present effort. Choosing a

sequence of f_j in an optimal manner also appeared to be more challenging than could be attempted within the scope of this study.

Created Response Surface Technique

The created response surface technique solved Problems C-1, C-2, and C-3 as effectively as Rosenbrock's approach, solving Problem C-3 to much better accuracy. The particular form of the transformed function was:

$$T(\bar{x},r_j) = F(\bar{x}) + r_j \sum_i g_i^{-1}(\bar{x}) + 2^{istg} \sum_i e_i^2(\bar{x})$$

where r was a previously defined sequence of values and $g_i^{-1}(\bar{x}) = 10^{16}$ when $g_i(\bar{x}) = 0$. The best general sequence of r's was 10^0, 10^{-2}, 10^{-4}, 10^{-6}, . . . , 10^{-22}, 0, in which the value of r was changed before every stage solved by Rosenbrock's method. This particular combination solved Problem C-1 to $\bar{x}$ = (24.07326, 11.97545, 11.98792), $F(\bar{x})$ = 3455.975 in 464 trials (11 stages), making no further progress through 2000 trials (38 stages). Problem C-2 was solved to only $\bar{x}$ = (-.02875676, .9022216, 2.044782, -.9629524), $F(\bar{x})$ = -43.94405 in 3343 trials (46 stages). At 2000 trials, the number reported for Rosenbrock's approach, $F(\bar{x})$ was-043.93735. The created response surface technique continued making slow progress through 4000 trials. Problem C-3 was solved to $\bar{x}$ = (1.333542, .7780167, .4442205) and $F(\bar{x})$ = .1111112 in 695 trials (17 stages), which was the most accurate solution to this problem.

The same sequence of r values was changed every fourth stage, yielding $F(\bar{x})$ values of 3455.940 and .1111119 for Problemc C-1 and C-3, respectively. Changing the value of r at each stage ($r_{istg} = 10^{-istg}$) produced an improved value of $F(\bar{x})$ for Problem C-1 (3455.993) in 508 trials, but the $F(\bar{x})$ for Problem C-3 was .1117549. In both of these

modifications, no results were obtained for Problem C-2 because of an error in the function computation instructions.

The created response surface technique was tried on Problem C-4, with r changed after every other stage. The minimum true function value obtained was .1122559 (about twice the desired value) after seven stages (1983 trials) in a run of 17,500 trials. Toward the end of the run, the transformed function value was on the order of 10^9, indicating that the equality constraints were poorly satisfied. The components of the $\bar{x}$ vector did not resemble the desired values.

Penalty Functions

For the penalty function solution procedure, the following transformation was used:

$$T(\bar{x},r_j) = F(\bar{x}) + r_j^{-1} \sum_i [\min(g_i(\bar{x}),0)]^2 + 2^{istg} \sum_i e_i^2(\bar{x})$$

The value of r was incremented at every other stage from the following sequence: 10^0, 10^{-2}, 10^{-4}, . . ., 10^{-22}, 10^{-30}.

The penalty function was not severe enough for Problem C-1. Both the true function value and the transformation value blew up to 10^{38} (the largest number that can be represented in the computer in single precision format) during the first stage and remained at that value. There may be a sufficiently strict sequence of r values, but no attempt was made to find one because problem dependent solution techniques were not desirable for this research. This approach did solve Problem C-2 to $F(\bar{x}) = -43.93340$ and Problem C-3 to $F(\bar{x}) = .1111211$, the latter of which is very good. Because of the performance on Problem C-1, penalty functions were ruled to be not acceptable.

Paviani and Himmelblau Approach

For the Paviani and Himmelblau approach, the value of C_j was chosen to be $2^{-istg}C_0$, where $C_0 = \max(\sum_i [\min(g_i(\bar{x}),0)]^2, 1)$ for the initial $\bar{x}$. This approach was not strictly followed concerning attempted function evaluations outside the near feasible region. When such a trial was attempted, it was aborted back through the failure path of the flow diagram, letting the search procedure find a subsequent near feasible point. Relocating this point in the Paviani and Himmelblau fashion would destroy the logic of the search routine. The method did not work well. It started each problem in an acceptable manner, but quit making progress after two stages because the criterion became too strict. The reason for this failure is also the compatibility of the iterative nature of the solution procedure and the iterative nature of the approach for solving constrained problems.

Summary

Of the five approaches to solving the constrained nonlinear programming problems presented in this chapter, only two exhibited any general capability for the three easier problems. These were Rosenbrock's approach and the created response surface technique, both of which displayed natural compatibility with the Rosenbrock-Fibonacci-Palmer combination method. Of the two, the Rosenbrock approach provided slightly better results on Problems C-1 and C-2, at the expense of preparing more instructions for the solution of a given problem and a loss of generality when called upon to solve an unconstrained problem. The created response surface technique solved Problem C-3 to the finest degree of accuracy and seems the easier to implement.

Neither of the two approaches could obtain the exacting accuracy presented for unconstrained problems in Chapter III. This is because of the special behavior of these techniques as inequality constraints approach equality. Since both of these methods treat the equalities as barriers, evaluations exactly on the boundary, where the optimum must occur if at least one of the constraints is meaningful, cannot be permitted during the early stages, and never without penalty in the Rosenbrock approach. The accuracy presented in this chapter is probably the best that can be expected, and certainly indicates what the true values might be.

Choosing between Rosenbrock's approach and the created response surface technique is largely a matter of personal choice. Both are acceptably efficient and one cannot predict which might be better for a given problem. The failure of both methods to solve Problem C-4 should be attributed more to the difficulty of the problem than to deficiencies in the methods. Since Problem C-4 was solved by its authors in over 13 minutes, and neither of these methods was allowed more than one minute, there is room for modification of the attack on this problem that might eventually lead to a solution. For demonstration purposes, the Rosenbrock approach is presented in the appendix, along with instructions for making both the Rosenbrock and the Rosenbrock-Fibonacci-Palmer combination searches on unconstrained or constrained problems.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

During the First Phase of this research, alternative methods for making improvements in the Rosenbrock direct search algorithm were investigated. None of the alternatives was found to be more generally effective than Rosenbrock's method, although one of the approaches frequently solved some individual problems faster than Rosenbrock's method. A combination method making alternating stages of Rosenbrock's search and the simultaneous Fibonacci search performed about equally as well as the Rosenbrock method alone. This method was chosen as the basic optimization method for the Second Phase.

Also during the First Phase, an investigation of alternate termination criteria produced no reliable improvements to the inefficient trial counter. False signs of optimality appeared frequently, while the true signs did not appear rapidly enough. Only when the optimal function value is known can an alternate termination criterion be implemented.

Five approaches to solving constrained nonlinear programming problems were evaluated with the Rosenbrock-Fibonacci-Palmer combination method in the Second Phase. Two of these approaches, one suggested by Rosenbrock and the created response surface technique, had the necessary compatibility with the combination method and displayed nearly equivalent performance on three relatively easy constrained problems. Both of the successful methods failed to make significant progress within one minute

of computing time on a 24-dimensional problem with 44 constraints. The other methods were rejected mainly for incompatibility with the iterative nature of the combination solution procedure.

There is only one recommendation to be made based upon the results of this study. During both phases, different solution methods solved some problems much more quickly than other methods. In many cases, the best solution method for an individual problem was not the best general method, since it may not have been able to solve other problems. If this relationship between the problem and the variables of a solution technique like the simultaneous Fibonacci search could be identified, users of direct search algorithms could capitalize on that relationship to decrease their computing time and increase their efficiency. The investigation of this relationship is appropriate for follow-on research.

APPENDIX

The following is a listing of the Rosenbrock-Fibonacci-Palmer computer program, written in Fortran V for the Univac 1108 at the Rich Electronic Computer Center, with appropriate comments and data cards for an unconstrained and a constrained problem. After the listing, modifications for running a strictly Rosenbrock search are discussed. No attempt has been made to provide an exhaustive diagnostic routine. The program requires valid input and can only produce garbage, which may appear meaningful, from invalid input.

```
@ FØR, IS MAIN                                                          @M01
C   RØSENBRØCK-FIBØNACCI-PALMER ØPTIMIZATIØN METHØD                     @M02
C                                                                       @M03
C   LIST ØF VARIABLES - MAIN PRØGRAM                                    @M04
C      ALPHA1 = THE NØRM ØF THE FIRST ALPHA VECTØR.  INDICATES          @M05
C               TØTAL PRØGRESS ØF THE PREVIØUS STAGE IN THE SPACE.      @M06
C      BLANK  = A BLANK FØR ØUTPUT PURPØSES.                            @M07
C      D      = AN ARRAY FØR STØRING PRØGRESS SUMS FØR EACH             @M08
C               DIRECTIØN.                                              @M09
C      FBEST  = THE BEST VALUE ØF THE ØBJECTIVE FUNCTIØN.               @M10
C      FEND   = TERMINATIØN VALUE FØR THE ØBJECTIVE FUNCTIØN.           @M11
C               REQUIRES MINIMIZATIØN PRØBLEM.                          @M12
C      FØNLY  = RUNCTIØN ENTRY TØ CØMPUTE INITIAL F(X).                 @M13
C      FØFX   = NØRMAL FUNCTIØN ENTRY FØR SUBSEQUENT F(X).              @M14
C      FTRUE  = TRUE FUNCTIØN VALUE, NØT MØDIFIED FØR MAXIMIZA-         @M15
C               TIØN ØR CØNSTRAINTS.                                    @M16
C      GAMMA  = PARAMETER MEASURING RELATIVE ALIGNMENT ØF               @M17
C               SUCCESSIVE DIRECTIØNS.                                  @M18
C      IPRØB  = DESIGNATES PRØBLEM BEING SØLVED.                        @M19
C      ISTG   = STAGE CØUNTER.                                          @M20
C      ITRMAX = TRIAL MAXIMUM.  HAS NØ NØMINAL VALUE.                   @M21
C      ITRY   = TRIAL CØUNTER.                                          @M22
C      MSG    = EIGHT WØRD ARRAY FØR ØUTPUT MESSAGE.                    @M23
C      N      = DIMENSIØNALITY ØF THE PRØBLEM.                          @M24
C      NML    = NAMELIST FØR FIRST DATA CARD PER PRØBLEM.               @M25
C      V      = DIRECTIØN VECTØRS.                                      @M26
C      X      = PØINT VECTØR.                                           @M27
C                                                                       @M28
```

```
C   MAIN PRØGRAM INITIALIZES, READS, WRITES, AND DRIVES                @M29
C      REMAINDER ØF THE PRØGRAM                                        @M30
C                                                                      @M31
C      CØMMØN /ØNE/ FTRUE,IPRØB,ISTG,N                                 @M32
       CØMMØN /TWØ/ ALPHA1,D(25),FBEST,GAMMA,ITRMAX,                   @M33
      1 ITRY,V(25,25),X(25)                                            @M34
       DIMENSIØN MSG(8)                                                @M35
       DATA BLANK/6H      /,MSG/24HRØSENBRØCK-FIBØNACCI-PAL,           @M36
      1 24HMER ØPTIMIZATIØN METHØD  /                                  @M37
       NAMELIST /NML/ IPRØB,N,ITRMAX,FEND,X                            @M38
       ITRMAX=1                                                        @M39
     1 ITRY=0                                                          @M40
       ISTG=0                                                          @M41
       FEND=-1.E+35                                                    @M42
       READ (5,NML,END=6)                                              @M43
       FBEST=FØNLY(X,$1,2.E+35)                                        @M44
       WRITE (6,8) IPRØB,MSG,(BLANK,I,I=1,N)                           @M45
       WRITE (6,10) ITRY,FTRUE,FBEST,(X(I),I=1,N)                      @M46
       DØ 3 I=1,N                                                      @M47
       DØ 2 J=1,N                                                      @M48
     2 V(I,J)=0.                                                       @M49
     3 V(I,I)=1.                                                       @M50
     4 CALL SEARCH                                                     @M51
       IF (ITRY .GE. ITRMAX)                                           @M52
       ISTG=ISTG+1                                                     @M53
       CALL RØTATE                                                     @M54
       WRITE (6,9) ISTG,ITRY,FTRUE,FBEST,ALPHA1,GAMMA,(X(I),           @M55
      1 I=1,N)                                                         @M56
       IF (FTRUE .LE. FEND) GØ TØ 5                                    @M57
       FBEST=FØFX(X,$1,2.E+35)                                         @M58
       ITRY=ITRY+1                                                     @M59
       WRITE (6,9) ISTG,ITRY,FTRUE,FBEST                               @M60
       GØ TØ 4                                                         @M61
     5 WRITE (6,10) ITRY,FTRUE,FBEST,(X(I),I=1,N)                      @M62
       GØ TØ 1                                                         @M63
     6 WRITE (6,7) MSG                                                 @M64
     7 FØRMAT (1H1, 8A6/////)                                          @M65
     8 FØRMAT (1H1,6(5H*****),15H PRØBLEM NUMBER,I3,3H,  ,8A6,         @M66
      1 6(5H*****)//10H STG  TRY,7X,4HF(X),8X,8HMØDF(X),6X             @M67
      2 6HALPHA1,5X,5HGAMMA,5(8X,A1,2HX(,I2,1H))/(57X,                 @M68
      3 5(8X,A1,2HX(,I2,1H)))                                          @M69
     9 FØRMAT (I4,I6,2G14.7,2G11.4,5G14.7/(60X,5G14.7))                @M70
    10 FØRMAT (1H0,I9,2G14.7,22X,5G14.7/(60X,5G14.7))                  @M71
       END                                                             @M72
```

```
@FØR,IS RØTATE                                                          @R01
      SUBRØUTINE RØTATE                                                 @R02
C                                                                       @R03
C  PALMER METHØD FØR GENERATING NEW DIRECTIØNS.                         @R04
C                                                                       @R05
C  LIST ØF VARIABLES - SUBRØUTINE RØTATE                                @R06
      CØMMØN /ØNE/ FTRUE,IPTRØB,ISTG,N                                  @R07
      CØMMØN /TWØ/ ALPHA1,D(25),FBEST,GAMMA,ITRMAX,                     @R08
     1 ITRY,V(25,25),X(25)                                              @R09
C     ALPHA  = INTERMEDIATE DIRECTIØN ARRAY.                            @R10
C     DUM    = DUMMEY VARIABLE FØR INTERMEDIATE VALUES.                 @R11
C     T      = ARRAY FØR DEVELØPING NEW DIRECTIØNS.                     @R12
C                                                                       @R13
      DIMENSIØN ALPHA(25,25),T(25)                                      @R14
      DØ 1 J=1,N                                                        @R15
      DØ 1 K=1,N                                                        @R16
    1 ALPHA(J,K)=0.                                                     @R17
      DØ 2 J=1,N                                                        @R18
      DØ 2 K=1,N                                                        @R19
      DØ 2 I=J,N                                                        @R20
    2 ALPHA(J,K)=ALPHA(J,K)+D(I)*V(I,K)                                 @R21
      DUM=ALPHA(1,1)*ALPHA(1,1)                                         @R22
      GAMMA=0.                                                          @R23
      DØ 3 I-2,N                                                        @R24
      DUM=DUM+ALPHA(1,I)*ALPHA(1,I)                                     @R25
    3 GAMMA=GAMMA+ALPHA(2,I)*ALPHA(2,I)                                 @R26
      ALPHA1=SQRT(DUM)                                                  @R27
      GAMMA=SQRT(GAMMA)/ALPHA1                                          @R28
      T(N)=D(N)*D(N)                                                    @R29
      I=N-1                                                             @R30
      DØ 4 K=I,1,-1                                                     @R31
    4 T(K)=T(K+1)+D(K)*D(K)                                             @R32
      DØ 6 K=N,2,-1                                                     @R33
      DUM=SQRT(T(K-1)*T(K))                                             @R34
      IF (DUM .LT. L.E-16) GØ TØ 6                                      @R35
      DØ 5 I=1,N                                                        @R36
    5 V(K,I)=(D(K-1)*ALPHA(K,I)-V(K-1,I)*T(K))/DUM                      @R37
    6 CØNTINUE                                                          @R38
      DUM=SQRT(T(1))                                                    @R39
      DØ 7 I=1,N                                                        @R40
    7 V(1,I)=ALPHA(1,I)/DUM                                             @R41
      RETURN                                                            @R42
      END                                                               @R43


@FØR,IS SEARCH                                                          @S01
      SUBRØUTINE SEARCH                                                 @S02
C                                                                       @S03
C  RØSENBRØCK AND FIBØNACCI CØMBINATIØN SEARCH                          @S04
C                                                                       @S05
```

```
C  LIST OF VARIABLES - SUBRØUTINE SEARCH                                @S06
      CØMMØN /ØNE/ FTRUE,IPRØB,ISTG,N                                   @S07
      CØMMØN /TWØ/ ALPHA1,D(25),FBEST,GAMMA,ITRMAX,                     @S08
     1 ITRY,V(25,25),X(25)                                              @S09
C     A       = SUCCESS-FAILURE STATUS INDICATØR FØR RØSEN-             @S10
C               BRØCK SEARCH.                                           @S11
C     ALPHA2  = LENGTH ØF SEARCH REGIØN FØR FIBØNACCI SEARCH.           @S12
C     E       = STEP SIZE FØR RØSENBRØCK SEARCH.                        @S13
C     FNA     = SMALLER ØF THE TWØ FIBØNACCI NUMBERS.                   @S14
C     FNB     = LARGER ØF THE TWØ FIBØNACCI NUMBERS.                    @S15
C     FNR     = RATIØ ØF THE TWØ FIBØNACCI NUMBERS.                     @S16
C     FTEMP   = TEMPØRARY LØCATIØN FØR THE VALUE ØF FTRUE.              @S17
C     FTEST   = FUNCTIØN VALUE FØR A NEW PØINT.                         @S18
C     FY      = FUNCTIØN VALUE FØR A CØMPARATIVE PØINT.                 @S19
C     TA      = ARRAY ØF STEP SIZES FØR FIBØNACCI SEARCH.               @S20
C     TFNRMØ  = TWICE FNR MINUS ØNE.                                    @S21
C     WHICH   = LØGICAL VARIABLE INDICATING WHICH SEARCH TØ MAKE.       @S22
C     Y       = THE PØINT FØR CØMPARATIVE EVALUATIØN IN F. SCH.         @S23
C                                                                       @S24
      DIMENSIØN A(25),E(25),TA(25),Y(25)                                @S25
      EQUIVALENCE (A(1),TA(1)),(E(1),Y(1)),(FTEST,FY),                  @S26
     1 (ALPHA2,TFNRMØ)                                                  @S27
      LØGICAL WHICH                                                     @S28
      IF (ISTG .NE. 0) GØ TØ 1                                          @S29
      WHICH=.TRUE.                                                      @S30
    1 WHICH=.NØT.WHICH                                                  @S31
      IF (WHICH) GØ TØ 13                                               @S32
C  RØSENBRØCK SEARCH                                                    @S33
      DØ 2 J=1,N                                                        @S34
      A(J)=2.                                                           @S35
      D(J)=0.                                                           @S36
    2 E(J)=.1                                                           @S37
    3 I=1                                                               @S38
    4 DØ 5 J=1,N                                                        @S39
    5 X(J)=X(J)+E(I)*V(I,J)                                             @S40
      FTEMP=FTRUE                                                       @S41
      ITRY=ITRY+1                                                       @S42
      FTEST=FØFX(X,$6,FBEST)                                            @S43
      IF (FTEST .LE. FBEST) GØ TØ 8                                     @S44
    6 DØ 7 J=1,N                                                        @S45
    7 X(J)=X(J)-E(I)*V(I,J)                                             @S46
      FTRUE=FTEMP                                                       @S47
      E(I)=-.5*E(I)                                                     @S48
      IF (A(I) .LT. 1.5) A(I)=0.                                        @S49
      GØ TØ 9                                                           @S50
    8 D(I)=D(I)+E(I)                                                    @S51
      E(I)=3.*E(I)                                                      @S52
      FBEST=FTEST                                                       @S53
      IF (A(I) .GT. 1.5) A(I)=1.                                        @S54
    9 IF (ITRY .GE. ITRMAX) RETURN                                      @S55
```

```
      DØ 10 J=1,N                                                    @S56
      IF (A(J) ,GT. .5) GØ TØ 11                                     @S57
   10 CØNTINUE                                                       @S58
      RETURN                                                         @S59
   11 IF (I-N) 12,3,3                                                @S60
   12 I=I+1                                                          @S61
      GØ TØ 4                                                        @S62
C  SIMULTANEOUS FIBØNACCI SEARCH                                     @S63
   13 FNA=987.                                                       @S64
      FNB=1597.                                                      @S65
      ALPHA=2.*AMAX1(ALPHA1,5.E-6)                                   @S66
      DØ 14 I=1,N                                                    @S67
      D(I)=0.                                                        @S68
   14 TA(I)=ALPHA2                                                   @S69
      DØ 21 M-1,16                                                   @S70
      FNR=FNA/FNB                                                    @S71
      TFNRMØ=2.*FNR-1.                                               @S72
      DØ 20 I=1,N                                                    @S73
      IF (M ,NE. 15) GØ TØ 15                                        @S74
      TA(I)=FNR*TA(I)                                                @S75
      GØ TØ 20                                                       @S76
      DØ 16 J-1,N                                                    @S77
   16 Y(J)=X(J)+TA(I)TFNRMØ*V(I,J)                                   @S78
      FTEMP=FTRUE                                                    @S79
      ITRY=ITRY+1                                                    @S80
      FY=FØFX(Y,$18,FBEST)                                           @S81
      IF (FBEST .LT. FY) GØ TØ 18                                    @S82
      FBEST=FY                                                       @S83
      DØ 17 J=1,N                                                    @S84
   17 X(J)=Y(J)                                                      @S85
      D(I)=D(I)+TA(I)*TFNRMØ                                         @S86
      TA(I)=(FNR*TA(I)                                               @S87
      GØ TØ 19                                                       @S88
   18 TA (I)=-FNR*TA(I)                                              @S89
      FTRUE=FTEMP                                                    @S90
   19 IF (ITRY .GE. ITRMAX) RETURN                                   @S91
   20 CONTINUE                                                       @S92
      FNA=FNB=FNA                                                    @S93
   21 FNB=FNB-FNA                                                    @S94
      DØ 24 I=1,N                                                    @S95
      IF (D(I)) 23,22,23                                             @S96
   22 D(I)=1.E=7                                                     @S97
      GØ TØ 24                                                       @S98
   23 IF (ABS(D(I) .GT. 1.E-7) GØ TØ 24                              @S99
      D(I)=D(I)+SIGN(1.E-6,D(I))                                     @SA0
   24 CONTINUE                                                       @SA1
      RETURN                                                         @SA2
      END                                                            @SA3
```

```
@FØR,IS FØFX                                                          @F01
      FUNCTIØN FØFX(X,$,FBEST)                                        @F02
C                                                                     @F03
C  CØMPUTES THE TRUE AND TRANSFØRMED FUNCTIØN VALUES.                 @F04
C  RØSENBRØCK FØRMULATIØN FØR CØNSTRAINED PRØBLEM.                    @F05
C                                                                     @F06
C  LIST ØF VARIABLES - FUNCTIØN FØFX                                  @F07
      CØMMØN /ØNE/ FTRUE,IPRØB,ISTG,N                                 @F08
C     AL     = WIDTH ØF THE BØUNDARY ZØNE.                            @F09
C     BØRDER = BØUNDARY ZØNE MULTIPLIER.                              @F10
C     CØN    = NAMELIST INPUT FØR CØNSTRAINT VALUES.  SØME            @F11
C              INPUT REQUIRED FØR UNCØNSTRAINED PRØBLEMS.             @F12
C     FMULT  = -1 FØR MAXIMIZATIØN, +1 ØTHERWISE.                     @F13
C     G      = ARRAY FØR CØNSTRAINT LØWER BØUNDS.                     @F14
C     H      = ARRAY FØR CØNSTRAINT UPPER BØUNDS.                     @F15
C     HJ     = BEST PREVIØUS FEASIBLE FUNCTIØN VALUE FØR              @F16
C              CØNSTRAINT J.                                          @F17
C     NCØN   = TØTAL NUMBER ØF CØNSTRAINTS.                           @F18
C     PHI    = PENALTY FUNCTIØN IN THE BØUNDARY ZØNE.                 @F19
C     RNU    = DISTANCE INTØ THE BØUNDARY ZØNE.                       @F20
C     X      = INPUT ARRAY FRØM CALLING RØUTINE.                      @F21
C     XJ     = CØNSTRAINT VALUES, FIRST N ARE X(I)                    @F22
C                                                                     @F23
      DIMENSIØN AL(50),G(50),H(50),HJ(50),X(25),XJ(50)                @F24
      NAMELIST /CØN/ NCØN,G,H,BØRDER                                  @F25
      DATA BØRDER,FMULT/1.E-8,1./                                     @F26
      IF (NCØN .EQ. 0) GØ TØ 2                                        @F27
      DØ 1 J-1,N                                                      @F28
    1 XJ(J)=X(J)                                                      @F29
    2 GØ TØ (1000,2000),IPRØB                                         @F30
 1000 FTRUE=100.*(X(1)**2-X(2))**2+(1.-X(1))**2                       @F31
C  UNCØNSTRAINED PRØBLEM 2                                           @F32
      FØFX=FTRUE                                                      @F33
      RETURN                                                          @F34
 2000 FTRUE=X(1)*X(2)*X(3)                                            @F35
C  CØNSTRAINED PRØBLEM C-1                                           @F36
      XJ(4)=X(1)+2.*X(2)+@.*X(3)                                      @F37
      F=FMULT*FTRUE                                                   @F38
      IF (F .GT. FBEST) RETURN 2                                      @F39
      DØ 5 J-1,NCØN                                                   @F40
      IF (XJ(J) .LT. G(J)) RETURN 2                                   @F41
      IF (XJ(J) .GT. H(J)) RETURN 2                                   @F42
      IF (XJ(J) .GE. G(J)+AL(J)) GØ TØ 3                              @F43
      RNU=(G(J)+AL(J)-XJ(J))/AL(J)                                    @F44
      PHI=1.-3.*RNU+4.*RNU*RNU-2.*RNU**3                              @F45
      F=HJ(J)+(F-HJ(J))*PHI                                           @F46
      GØ TØ 5                                                         @F47
    3 IF (XJ(J) .LE. H(J)-AL(J)) GØ TØ 4                              @F48
      RNU=(XJ(J)-H(J)+AL(J))/AL(J)                                    @F49
      PHI=1.-3.*RNU+4.*RNU*RNU-2.*RNU**3                              @F50
```

```
      F=HJ(J)+(F-HJ(J))*PHI                                    @F51
      GØ TØ 5                                                  @F52
    4 HJ(J)=FBEST                                              @F53
    5 CØNTINUE                                                 @F54
      FØFX=F                                                   @F55
      RETURN                                                   @F56
      ENTRY FØNLY(X,$,FBEST)                                   @F57
      READ (5,CØN)                                             @F58
      IF (IPRØB .EQ. 1) GØ TØ 1000                             @F59
      DØ 6 J=1,NCØN                                            @F60
    6 AL(J)=(H(J)-G(J))*BØRDER                                 @F61
      FMULT=-1                                                 @F62
      FTRUE=X(1)*X(2)*X(3)                                     @F63
      F=FMULT*FTRUE                                            @F64
      DØ 7 J=1,NCØN                                            @F65
    7 HJ(J)=F                                                  @F66
      FØFX=F                                                   @F67
      RETURN                                                   @F68
      END                                                      @F69

@MAP,S
@XQT
 $NML IPRØB=1, N-2, ITRMAX=1000, FEND=0., X=-1.2,1 $END
 $CØN NCØN=0 $END
 $NML IPRØB=2, N=3, ITRMAX=1000, X=3*6. $END
 $CØN NCØN=4, G=4*0., H=3*42.,72., BORDER=1.E-8 $END
@ FIN
```

To make a run with only the Rosenbrock search, the simplest change is to insert a card with the instruction WHICH=.TRUE. between cards @S28 and @S29. Many other changes are possible, but some require complete revision of the subroutine, while others may produce bothersome diagnostic messages.

BIBLIOGRAPHY

1. Box, M.J., "A Method of Constrained Optimization and a Comparison with Other Methods," Computer Journal, Vol. 8, pp. 42-52, 1965.

2. _____, "A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems," Computer Journal, Vol. 9, pp. 67-77, 1966.

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

4. Fiacco, Anthony V., and Garth P. McCormick, "The Sequential Unconstrained Minimization Technique for Nonlinear Programming, A Primal-Dual Method," Management Science, Vol. 10, pp. 360-366,1964.

5. _____, and _____, "Extensions of SUMT for Nonlinear Programming: Equality Constraints and Extrapolation," Management Science, Vol. 12, pp. 816-828, 1966.

6. Fletcher, R., "Function Minimization Without Calculating Derivatives - A Review," Computer Journal, Vol. 8, pp. 33-41, 1965.

7. Kowalik, J., and M.R. Osborne, Methods for Unconstrained Optimization Problems, American Elsevier Publishing Company, Inc., New York, 1968.

8. Krylov, Vladimir Ivanovich, translated by Arthur H. Stroud, Approximate Calculation of Integrals, MacMillan Company, New York, 1962.

9. Palmer, J.R., "An Improved Procedure for Orthogonalising the Search Vectors in Rosenbrock's and Swann's Direct Search Optimisation Methods," Computer Journal, Vol. 12, pp. 69-71, 1969.

10. Paviani, D.A., and D.M. Himmelblau, "Constrained Nonlinear Optimization by Heuristic Programming," Operations Research, Vol. 17, pp. 872-882, 1969.

11. Powell, M.J.D., "An Iterative Method for Finding Stationary Values of a Function of Several Variables," Computer Journal, Vol. 5, pp. 147-151, 1962.

12. Pun, Lucas, Introduction to Optimization Practice, John Wiley and Sons, Inc., New York, 1969.

13. Rosen, J.B., and S. Suzuki, "Construction of Nonlinear Programming Test Problems," Communications of the ACM, Vol. 8, p. 113, 1965.

14. Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or Least Value of a Function," Computer Journal, Vol. 3, pp. 175-184, 1960.

15. Rosenbrock, H. H., and C. Story, Computational Techniques for Chemical Engineers, Pergamon Press, Oxford, 1966.

16. Wilde, Douglass J., Optimum Seeking Methods, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1964.

17. Wilde, Douglass J., and Charles S. Beightler, Foundations of Optimization, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967.

18. Zangwill, W. I., Nonlinear Programming: A Unified Approach, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1969.