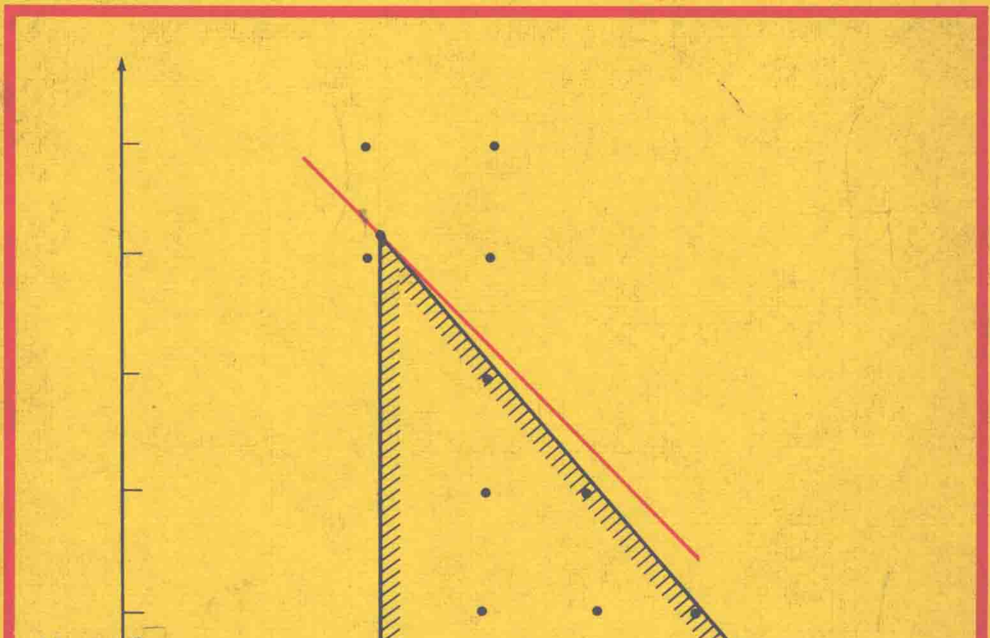
Undergraduate Texts in Mathematics

L. R. Foulds

Combinatorial Optimization for Undergraduates



Springer-Verlag New York • Berlin • Heidelberg • Tokyo

L. R. Foulds

Combinatorial Optimization for Undergraduates

With 56 Illustrations

Springer-Verlag
New York Berlin Heidelberg Tokyo

L. R. Foulds
Department of Industrial and Systems Engineering
University of Florida
Gainesville, FL 32611
U.S.A.

AMS Classifications: 05-01, 49-01

Library of Congress Cataloging in Publication Data
Foulds, L. R.
Combinatorial optimization for undergraduates.
(Undergraduate texts in mathematics)
Bibliography: p.
Includes index.
1. Combinatorial optimization. I. Title.
II. Series.
QA164.F68 1984 519 84-5381

Typeset by Science Typographers, Inc., Medford, New York.
Printed and bound by R. R. Donnelley & Sons, Harrisonburg, Virginia.
Printed in the United States of America.

9 8 7 6 5 4 3 2 1

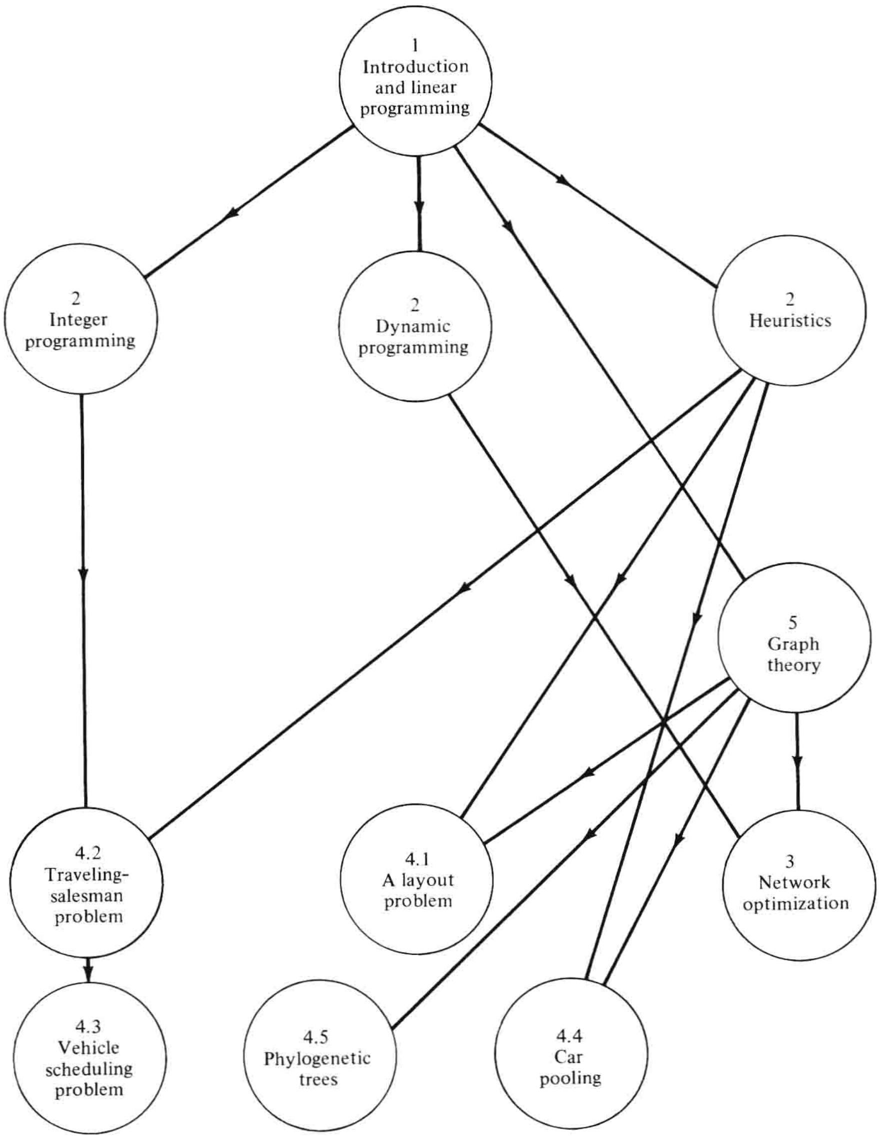
ISBN 0-387-90977-X Springer-Verlag New York Berlin Heidelberg Tokyo
ISBN 3-540-90977-X Springer-Verlag Berlin Heidelberg New York Tokyo

Preface

The major purpose of this book is to introduce the main concepts of discrete optimization problems which have a finite number of feasible solutions. Following common practice, we term this topic *combinatorial optimization*. There are now a number of excellent graduate-level textbooks on combinatorial optimization. However, there does not seem to exist an undergraduate text in this area. This book is designed to fill this need.

The book is intended for undergraduates in mathematics, engineering, business, or the physical or social sciences. It may also be useful as a reference text for practising engineers and scientists. The writing of this book was inspired through the experience of the author in teaching the material to undergraduate students in operations research, engineering, business, and mathematics at the University of Canterbury, New Zealand. This experience has confirmed the suspicion that it is often wise to adopt the following approach when teaching material of the nature contained in this book. When introducing a new topic, begin with a numerical problem which the students can readily understand; develop a solution technique by using it on this problem; then go on to general problems. This philosophy has been adopted throughout the book. The emphasis is on plausibility and clarity rather than rigor, although rigorous arguments have been used when they contribute to the understanding of the mechanics of an algorithm. An example of this is furnished by the construction of the labeling method for the maximal-network-flow problem from the proof of the max-flow, min-cut theorem.

The book comprises two parts—Part I: Techniques and Part II: Applications. Part I begins with a motivational chapter which includes a description of the general combinatorial optimization problem, important current problems, a description of the fundamental algorithm, a discussion of the



Plan of the Book.

need for efficient algorithms, and the effect of the advent of the digital computer. This is followed by a chapter on linear programming and its extensions. Chapter 2 describes the basic procedures of three of the most important combinatorial optimization techniques—integer programming, dynamic programming, and heuristic methods. Chapter 3 is concerned with optimization on graphs and networks.

Part II poses a variety of problems from many different disciplines—the traveling-salesman problem, the vehicle scheduling problem, car pooling, evolutionary tree construction, and the facilities layout problem. Each problem is analyzed and solution procedures are then presented. Some of these procedures have never appeared before in book form.

The book contains a number of exercises which the reader is strongly encouraged to try. Mathematics is not a spectator sport! These exercises range from routine numerical drill-type exercises to open questions from the research literature. The more challenging problems have an asterisk preceding them. The author is grateful for this opportunity to express his thanks for the support he received from the University of Canterbury while writing this book, and to his doctoral student John Giffin, who contributed to Section 4.1. He is also extremely thankful to his wife Maureen, who not only provided enthusiastic encouragement, but also typed the complete manuscript. Finally, the author pays a hearty tribute to the staff at Springer-Verlag New York for their patience, skill, and cooperation during the preparation of this book.

Gainesville, Florida L. R. FOULDS

Contents

PART I

TECHNIQUES

CHAPTER 0

Introduction to the Techniques of Combinatorial Optimization

0.1. The General Problem

Optimization is concerned with finding the best (or optimal) solution to a problem. In this book we are concerned with problems that can be stated in an unambiguous way, usually in terms of mathematical notation and terminology. It is also assumed that the value of any solution to a given problem can be measured in a quantifiable way and this value can be compared with that of any other solution to the problem. Problems of this nature have been posed since the beginning of mankind. One of the earliest is recorded by Virgil in his *Aeneid* where he relates the dilemma of Queen Dido, who was to be given all the land she could enclose in the hide of a bull. She cut the hide into thin strips and joining these together formed a semicircle within which she enclosed a sizeable portion of land with the Mediterranean coast as the diameter. Later Archimedes conjectured that her mathematical solution was optimal; that is, a semicircle is the curve of fixed length which will, together with a straight line, enclose the largest possible area. This conjecture can be proved using a branch of optimization called the calculus of variations.

The problem just described has an infinite number of solutions as there is an infinite number of possible curves of any given length. However, there is an important class of optimization problems which have only a finite number of solutions. The body of knowledge concerned with the theory and techniques for these problems is called "combinatorial optimization" and it is with this class that our book deals. Let S be the finite set of solutions to a problem and assume each solution $x \in S$, can be evaluated and assigned a real number $f(x)$ indicating its worth. This assignment may be in terms of some benefit, such as profit, which is to be maximized, or some detriment, such as cost, which is to be minimized.

We now formally introduce the general problem of combinatorial optimization. Let

$$f: D \to \mathbb{R}$$

be a real-valued function with domain D. Let

$$S \subseteq D.$$

Definition 0.1. $x^* \in S$ is a *global maximum of* f if

$$f(x^*) \geq f(x) \qquad \text{for all } x \in S.$$

The definition of a global minimum is analogous.

Definition 0.2. $x^* \in S$ is a *global extremum of* f if x^* is either a global maximum or global minimum of f.

The "general maximization problem of combinatorial optimization" is to find x^* such that x^* is a global maximum of f; that is, to identify x^* such that

$$f(x^*) = \text{Max}(f(x)) \qquad x \in S.$$

The definition of the "general minimization problem of combinatorial optimization" is analogous.

S is called the set of *feasible* solutions and if

$$x \in S,$$

x is called a *feasible* solution or is termed *feasible*. If $\bar{x} \in D$ and

$$f(\bar{x}) \geq f(x) \qquad \text{for all } x \in S,$$

$\bar{x}$ is termed an *upper bound for f on S*. If

$$f(\bar{x}) \leq f(x) \qquad \text{for all } x \in S,$$

$\bar{x}$ is termed a *lower bound for f on S*. If $\bar{x}$ is an upper bound for f on S and

$$f(\bar{x}) \leq f(x)$$

for all upper bounds x for f on S, then $\bar{x}$ is termed a *least upper bound for f on S*. If $\bar{x}$ is a lower bound for f on S and

$$f(\bar{x}) \geq f(x)$$

for all lower bounds x for f on S, then $\bar{x}$ is termed a *greatest lower bound for f on S*.

Note that $\bar{x}$ may or may not be a member of S. Of course if

$$\bar{x} \in S,$$

$\bar{x}$ is a global extremum of f.

0.1.1. An Illustrative Example of a Combinatorial Optimization Problem: The Shortest Hamiltonian Path

This section is based upon an article by D. F. Robinson appearing in the Proceedings of the First Australian Conference on Combinatorics, Newcastle, Australia, August 1972.

Of course the nature of S can vary considerably from one problem to another. Even though S is finite it may be extremely large and further it may not be an easy task to identify its elements. We now present an illustrative example of a combinatorial optimization problem that is simple in concept, in order to give the reader some idea of what is to come.

Let $V=\{v_1, v_2, \ldots, v_n\}$ be a set of n cities where $n>1$. Consider the problem of finding a shortest itinerary which passes through all the cities of V. Let d_{ij} represent the distance from v_i to v_j, $1 \le i \le n$, $1 \le j \le n$. The distance matrix

$$\mathbf{D}=\left[d_{ij}\right]_{n \times n}$$

is assumed to be symmetric in the sense that

$$d_{ij}=d_{ji} \qquad 1 \le i \le n, 1 \le j \le n.$$

This problem is similar to one in the literature known as the traveling-salesman problem which is the subject of Section 4.2.

As V has n members there are $n!$ paths. We express a typical path as

$$x=\left\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)}\right\rangle,$$

where $\{v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)}\}=\{v_1, v_2, \ldots, v_n\}$, and x is the path which begins at $v_{\alpha(1)}$ and then visits $v_{\alpha(2)}$, $v_{\alpha(3)}$, and so on, ending at $v_{\alpha(n)}$. The set S of solutions to this problem is

$$S=\left\{\left\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)}\right\rangle : \{\alpha(1), \alpha(2), \ldots, \alpha(n)\}=\{1,2, \ldots, n\}\right\}.$$

If $x=\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)}\rangle \in S$, then the value of x, $f(x)$ is the length of x,

$$f(x)=\sum_{i=1}^{n-1} d_{\alpha(i), \alpha(i+1)}.$$

Then the problem is find

$$f(x^*)=\min_{x \in S}\{f(x)\}.$$

To each path $x=\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)}\rangle$ there corresponds a reverse path, $x^R=\langle v_{\alpha(n)}, v_{\alpha(n-1)}, \ldots, v_{\alpha(1)}\rangle$. Because d is symmetric, $f(x)=f(x^R)$ for all $x \in S$. Hence the minimum path will not be unique.

We define the following elementary operations on a path:

(i) Break the path $\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)} \rangle$ after some point $v_{\alpha(m-1)}$ ($2 \leq m \leq n$) and join $v_{\alpha(n)}$ to $v_{\alpha(1)}$. The new path is

$$\langle v_{\alpha(m)}, v_{\alpha(m+1)}, \ldots, v_{\alpha(n)}, v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(m-1)} \rangle.$$

This new path will be shorter than the original one if and only if

$$d_{\alpha(n),\,\alpha(1)} < d_{\alpha(m-1),\,\alpha(m)}.$$

(ii) Break the path $\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)} \rangle$ after some point $v_{\alpha(m-1)}$ ($2 \leq m \leq n-1$) and then reverse the direction of the second part to yield

$$\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(m-1)}, v_{\alpha(n)}, v_{\alpha(n-1)}, \ldots, v_{\alpha(m)} \rangle.$$

This new path will be shorter than the original one if and only if

$$d_{\alpha(m-1),\,\alpha(n)} < d_{\alpha(m-1),\,\alpha(m)}.$$

(iii) The reverse of operation (ii). Break the path $\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)} \rangle$ after $v_{\alpha(m)}$ ($2 \leq m \leq n-1$) and reverse the first half to give the path

$$\langle v_{\alpha(m)}, v_{\alpha(m-1)}, \ldots, v_{\alpha(1)}, v_{\alpha(m+1)}, v_{\alpha(m+2)}, \ldots, v_{\alpha(n)} \rangle.$$

This new path is shorter than the original if

$$d_{\alpha(1),\,\alpha(m+1)} < d_{\alpha(m),\,\alpha(m+1)}.$$

(iv) Take any pair of adjacent points $v_{\alpha(m)}, v_{\alpha(m+1)}$ in the path

$$\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(n)} \rangle$$

and reverse their order to obtain

$$\langle v_{\alpha(1)}, v_{\alpha(2)}, \ldots, v_{\alpha(m-1)}, v_{\alpha(m+1)}, v_{\alpha(m)}, v_{\alpha(m+2)}, \ldots, v_{\alpha(n)} \rangle.$$

The cases $m = 1$ or $m = n - 1$ have been dealt with in (ii) and (iii). Otherwise the new path is shorter than the original if

$$d_{\alpha(m-1),\,\alpha(m+1)} + d_{\alpha(m),\,\alpha(m+2)} < d_{\alpha(m+2),\,\alpha(m-1)} + d_{\alpha(m),\,\alpha(m+1)}.$$

We now note some properties of these operations:

(a) If it is possible to obtain path y by an elementary operation on a path x, then x can be obtained from y by an elementary operation.
(b) If a path y can be obtained from a path x by an elementary operation, then path y^R, the reverse of y, can be obtained from x^R by an elementary operation.
(c) Each path can be considered a permutation on $\{v_1, v_2, \ldots, v_n\}$. Type (iv) operations are in effect permutation transpositions.

It can be shown that any permutation can be expressed as a product of transpositions. Hence any path can be transformed into any other path by a finite sequence of type (iv) operations. Hence any pair of paths can be transformed, one into the other, by a finite sequence of operations of types (i)–(iv).

Table 0.1. A City-to-City Distance Table

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_2	201									
v_3	428	227								
v_4	207	156	348							
v_5	232	159	351	25						
v_6	564	363	136	448	423					
v_7	73	274	501	186	211	634				
v_8	19	220	447	226	251	583	72			
v_9	508	307	176	340	314	118	526	527		
v_{10}	302	101	126	222	225	262	375	321	217	
v_{11}	165	210	437	65	90	513	144	184	405	311

Let us now search for a path of minimum length. The ideas will be made more clear by examining a numerical example. Table 0.1 gives the distances between a set of 11 cities.

In Table 0.2 we set out the successive shorter paths, represented by the cities in order and the distances between them. The starting path A_0 consists of the cities in order of increasing subscript. The greatest distance between successive cities is 634 (from v_6 to v_7). This is greater than $d_{1,11}$ $(=165)$. We therefore split the path A_0 between v_6 and v_7 and join v_1 to v_{11} by a type (i) operation. We denote this break by the symbol $\wedge$ in the appropriate place. The new path is denoted by A_1. We can never use type (i) operations twice successively to any advantage. We now turn to type (ii) operations and compare the distances between cities with the distances from the right-hand side of a pair to v_7. We find that v_7 is closer to v_{11} than v_{10} is. So we form path A_2 by reversing the section of the path from v_7 to v_{10}. This is denoted by the symbols $<$ and $>$. In A_2 the cities v_9 and v_8 are 527 apart. This exceeds the distance between v_{10} and v_6. So a type (i) operation will reduce this length. We continue using type (i) and (ii) operations until we reach A_{13}. No further type (i) or (ii) operation will reduce the length of this path. A type (iii) operation, reversing the order of the last three cities, will reduce the length of A_{13}. This produces path A_{14}. No operation of any type will reduce the length of A_{14}, which is 965.

Note that the distance from v_6 to v_7 is 634. Hence it is evident that any shorter path must have one end "close" to v_6 and the other "close" to v_7. It is then a simple matter to prove that A_{14} (with its reverse) is a global minimum.

Definition 0.3. If, for all such x_i

$$f(x_0) \le f(x_i),$$

x_0 is said to be a *local minimum of* f.

Table 0.2. A Succession of Shorter Paths

A_0	v_1	201	v_2	227	v_3	348	v_4	25	v_5	423	v_6	634 $\wedge$	v_7	72	v_8	527	v_9	217	v_{10}	311	v_{11}
A_1	v_7 $<$	72	v_8	527	v_9	217	v_{10}	311 $>$	v_{11}	165	v_1	201	v_2	227	v_3	348	v_4	25	v_5	423	v_6
A_2	v_{10}	217	v_9	527 $\wedge$	v_8	72	v_7	144	v_{11}	165	v_1	201	v_2	227	v_3	348	v_4	25	v_5	423	v_6
A_3	v_8 $<$	72	v_7	144	v_{11}	165 $>$	v_1	201	v_2	227	v_3	348	v_4	25	v_5	423	v_6	262	v_{10}	217	v_9
A_4	v_{10}	144	v_7	72	v_8	19	v_1	201	v_2	227	v_3	348	v_4	25	v_5	423 $\wedge$	v_6	262	v_{10}	217	v_9
A_5	v_6 $<$	262	v_{10}	217 $>$	v_9	405	v_{11}	144	v_7	72	v_8	19	v_1	201	v_2	227	v_3	348	v_4	25	v_5
A_6	v_{10}	262	v_6	118	v_9	405 $\wedge$	v_{11}	144	v_7	72	v_8	19	v_1	201	v_2	227	v_3	348	v_4	25	v_5
A_7	v_{11} $<$	144	v_7	72	v_8	19	v_1	201	v_2	227	v_3	348 $>$	v_4	25	v_5	225	v_{10}	262	v_6	118	v_9
A_8	v_3	227	v_2	201	v_1	19	v_8	72	v_7	144	v_{11}	65	v_4	25	v_5	225	v_{10}	262 $\wedge$	v_6	118	v_9
A_9	v_6 $<$	118	v_9	176 $>$	v_3	227	v_2	201	v_1	19	v_8	72	v_7	144	v_{11}	65	v_4	25	v_5	225	v_{10}
A_{10}	v_9	118	v_6	136	v_3	227 $\wedge$	v_2	201	v_1	19	v_8	72	v_7	144	v_{11}	65	v_4	25	v_5	225	v_{10}
A_{11}	v_2 $<$	201	v_1	19	v_8	72	v_7	144	v_{11}	65	v_4	25	v_5	225 $>$	v_{10}	217	v_9	118	v_6	136	v_3
A_{12}	v_5 $<$	25	v_4	65	v_{11}	144	v_7	72	v_8	19	v_1	201 $>$	v_2	101	v_{10}	217	v_9	118	v_6	136	v_3
A_{13}	v_1	19	v_8	72	v_7	144	v_{11}	65	v_4	25	v_5	159	v_2	101	v_{10}	217 $<$	v_9	118	v_6	136	v_3 $>$
A_{14}	v_1	19	v_8	72	v_7	144	v_{11}	65	v_4	25	v_5	159	v_2	101	v_{10}	126	v_3	136	v_6	118	v_9

Definition 0.4. If, for all such x_i

$$f(x_0) \geq f(x_i),$$

x_0 is said to be a *local maximum of* f.

Definition 0.5. If x_0 is either a local minimum of f or a local maximum of f then x_0 is said to be a *local extremum of* f.

We may generalize the above approach. Consider the general minimization problem of combinatorial optimization as defined in Section 0.1. We can define on S a collection of elementary operations with the following properties:

(a) If $y \in S$ can be obtained from $x \in S$ by an elementary operation, then x can be obtained from y by an elementary operation.
(b) Given any two $x, y \in S$ there is a finite sequence of elementary operations which convert x into y.

The elementary operations thus define a connected graph, G (see Section 5.2), whose vertices are the members of S and whose edges join members of S linked by an elementary operation. A solution process can be constructed as follows. Begin at an arbitrary vertex $x_0 \in S$ and evaluate $f(x_0)$. We then evaluate $f(x_i)$ for each x_i adjacent to x_0 in G.

If no such local minimum is detected, choose an $x_j \in S$ adjacent to x_0 for which

$$f(x_0) > f(x_j) \tag{0.1}$$

and repeat the above process with x_0 replaced by x_j. One method of selecting x_j at each stage is to choose the first member of S adjacent to x_0 for which it is discovered that (0.1) holds.

The above process must terminate in the identification of a local minimum in a finite number of steps as S is finite. A possible minor complication may arise in that a "plateau" x_p may be detected where

$$f(x_p) = f(x_j)$$

for some x_j adjacent to x_p, but with no adjacent vertex x_k such that

$$f(x_p) > f(x_k).$$

If this situation arises, the set S' of all such vertices x_j is progressively examined in the hope that a vertex x_s may be found which is adjacent to a vertex $x_j \in S'$ and

$$f(x_p) = f(x_j) > f(x_s).$$

Then the process is repeated with x_0 replaced by x_s.

Some exercise of judgment is needed in the application of this process. If the number of vertices adjacent to any given vertex is usually relatively