INTRODUCTION TO OPTIMIZATION

E. M. L. Beale

Scicon Ltd, Milton Keynes, and Imperial College, London

Edited by

Lynne Mackley

Scicon Ltd, Milton Keynes

A Wiley–Interscience Publication

JOHN WILEY & SONS

Chichester · New York · Brisbane · Toronto · Singapore

Library of Congress Cataloging-in-Publication Data

Beale, E. M. L. (Evelyn Martin Landsdowne)
Introduction to optimization.

(Wiley–Interscience series in discrete mathematics and optimization)
'A Wiley–Interscience publication.'
Bibliography: p.
Includes index.
1. Mathematical optimization. I. Mackley, Lynne. II. Title. III. Series.
QA402.5.B42 1988 519 87–29613

ISBN 0 471 91760 5
ISBN 0 471 91761 3 (pbk.)

British Library Cataloguing in Publication Data

Beale, E. M. L.
Introduction to optimization.—(Wiley–interscience series in discrete mathematics and optimization).
1. Mathematical optimization
I. Title II. Mackley, Lynne
515 QA402.5

ISBN 0 471 91760 5
ISBN 0 471 91761 3 Pbk

Phototypesetting by Thomson Press (India) Limited, New Delhi
Printed and bound in Great Britain by Anchor Brendon Ltd., Tiptree, Essex

Preface

This book is based upon a series of lecture notes written by Professor E. M. L. Beale FRS for his undergraduate course 'Introduction to Optimization', given at Imperial College where he was a Visiting Professor in Mathematics from 1967 until his death in December 1985. Where necessary, additional material for the chapters on the dual and primal simplex methods has been taken from Professor Beale's book *Mathematical Programming in Practice* (Beale, 1968). In addition, Sections 6.3, 6.4 and 8.1 are based upon the papers 'The current algorithmic scope of mathematical programming systems' (Beale, 1975), and 'The evolution of mathematical programming systems' (Beale, 1985, reproduced by permission of Pergamon Press), whilst the remainder of Chapters 8 and 9 are based on Professor Beale's unpublished paper 'How to apply mathematical programming', which he wrote and continually revised for training courses given by the company Scicon Ltd, of which he was a founder member and where he was for many years Technical Director.

As the title suggests, this book is intended as an introduction to the many topics covered by the heading Optimization, with special emphasis being placed on applications of optimization in industry. Although the lecture notes upon which the book is based were originally written for third-year mathematics undergraduates, no detailed mathematical knowledge is assumed and the book is equally suitable for engineers and computer scientists who are studying options in Operational Research.

The book is divided into three parts. The first concentrates on Unconstrained Optimization and describes some of the main techniques which have been developed to solve problems of this kind. Chapter 2 discusses methods which can be used to optimize functions of only one variable, whilst Chapter 3 considers multi-variable functions. Emphasis is placed on the practical problems of why and how methods succeed or fail, rather than on rigorous proofs about convergence. The second part of the book, Unconstrained Optimization: Linear Programming, describes the methods used to solve linear programming problems and applications of linear programming in industry. The simplex and dual simplex methods are outlined very simply using numerical examples and ways in which the simplex method has been adapted for use on computers are described. Considerable emphasis is placed on the efficient modelling and systematic

documentation of linear programming problems. The third part is entitled Unconstrained Optimization: Non-Linear and Discrete, and covers non-linear programming, integer programming and dynamic programming, showing how the techniques of linear programming can be extended to handle non-linearities and discrete entities.

I would like to thank all the people who helped me to edit the book. In particular, I am indebted to Steven Vajda for his advice and encouragement and I would like to thank my colleagues Robert Simons and Bob Hattersley for their advice on technical details about linear programming codes. I am also very grateful to Bev Peberdy for her help in typing the manuscript. Finally, I would like to express my thanks to Scicon Ltd for allowing me time to edit this book.

Lynne Mackley (Scicon Ltd)
August 1987

A biographical memoir of E. M. L. Beale, by M. J. D. Powell, was published in Volume 33 (1987) of *Biographical Memoirs of Fellows of the Royal Society*.

Contents

Preface

1 Introduction 1
1.1 Introduction to Optimization 1
1.2 Basic Theoretical Notions 3

PART 1 UNCONSTRAINED OPTIMIZATION

2 Introduction to Unconstrained Optimization Techniques 11
2.1 Differential Calculus 11
2.2 Iterative Methods 12

3 One-dimensional Optimization 15
3.1 Finding the Roots of an Equation 15
3.2 Maximization 19

4 Multi-dimensional Optimization 25
4.1 Introduction 25
4.2 Extensions of Newton's Method 26
4.3 Conjugate Gradients 29
4.4 Quasi-Newton Methods 32

PART 2 CONSTRAINED OPTIMIZATION: LINEAR PROGRAMMING

5 The Simplex Method for Linear Programming 39
5.1 Introduction 39
5.2 The Single-mix Blending Problem 40
5.3 The Algebra of the Simplex Method 45

6 Further Details of the Simplex Method 49
6.1 Basic Method of Finding a First Feasible Solution 49
6.2 Practical Method of Finding a First Feasible Solution 52
6.3 Upper and Lower Bounds 55
6.4 Exploiting Sparsity 57

7 Duality and Parametric Programming 61
7.1 Introduction to Duality 61
7.2 Applications of Duality 63
7.3 Dual Simplex Method 65
7.4 Parametric Programming 67

8 How to Apply Linear Programming 70
8.1 Introduction: Mathematical Programming Systems 70
8.2 Applications of Mathematical Programming 71
8.3 A Technique for Model Documentation 72
8.4 Communicating the Formulation to the Computer 76

9 Examples of Linear Programming Problems 78
9.1 Preface to Examples 78
9.2 A Multi-mix Blending Problem 78
9.3 A Processing Problem 80
9.4 A Multi-time-period Production and Storage Problem 82
9.5 A Long-term Investment Planning Problem 83

PART 3 CONSTRAINED OPTIMIZATION: NON-LINEAR AND DISCRETE

10 Non-linear Programming 89
10.1 Introduction 89
10.2 Lagrange Multipliers and Kuhn–Tucker Conditions 89
10.3 Separable Programming 91
10.4 Example of Separable Programming 94
10.5 Reduced Gradient Methods 95

11 Integer Programming 100
11.1 Introduction 100
11.2 Branch and Bound Methods 101
11.3 Formulation of Integer Programming Problems 107

12 Dynamic Programming 109
12.1 Introduction 109
12.2 Shortest-route Problems 109
12.3 An Example Problem 112
12.4 The Allocation of Effort 113
12.5 The Optimum Division of a Line into Segments 114

References 115

Subject Index 119

1

Introduction

1.1 INTRODUCTION TO OPTIMIZATION

Optimization involves finding the best solution to a problem. Mathematically, this means finding the minimum or maximum of a function of n variables, $f(x_1, \ldots, x_n)$, say, where n may be any integer greater than zero. The function may be unconstrained or it may be subject to certain constraints on the variables of the function, say $g_i(x_1, \ldots, x_n) = b_i$ for $i = 1, \ldots, m$. We will see later that the functions $f(\mathbf{x})$ and $g_i(\mathbf{x})$ usually have some real physical meaning (for example, total cost and profit or capacity and demand restrictions).

Although optimization is used in various branches of applied mathematics and statistics it is particularly associated with *operational research*. Before discussing what operational research involves, we will spend a few minutes distinguishing it from statistics.

We can argue that statistics is concerned with trying to understand what is happening (or what might happen) in an uncertain world full of apparently random phenomena, and that operational research is concerned with deciding what to do about it. If we make this distinction we must add that a practical statistician must then spend some of his time practising operational research, and that a practical operational research worker must spend some of his time practising statistics. We may also add that the quality of the solution to any practical problem may be impaired by dividing the problem up in this way. However, such arbitrary divisions of problems into manageable components are often necessary initial steps towards finding any solutions at all. So I believe that this way of expressing the different approaches of these two major branches of applicable mathematics is of some value.

Operational research is therefore concerned with *decision making*. This can be an instinctive, or at least intuitive, process, but this is not always a satisfactory way to make decisions, particularly those made on behalf of other people, for example by a government department or a commercial organization. We may therefore approach the situation more methodically, and list the alternative possible decisions and their respective advantages and disadvantages. We may then go further and quantify them, and this leads us to make what is generally known as a *mathematical model* of the situation requiring a decision.

Mathematical modelling is at the heart of operational research. This really just means analysing some logical structure that is as simple as possible while still representing the essence of the problem faced by the decision maker.

Let us take a trivial example. Suppose that I take seriously the problem of whether or not to take a raincoat when I leave home in the morning to go to work. Let us suppose that it is not raining, but that I do not want to get wet coming home in the evening. On the other hand, I dislike spending time finding and carrying my raincoat if it is not going to rain. Then I could represent all possible events in the coming day by just two possibilities: wet in the evening or dry in the evening. Then I could try to assess the probabilities of these two alternatives, and I could also try to assess the relative inconveniences of not having a raincoat if it is wet and of carrying one if it is dry. By multiplying the relative inconveniences by the corresponding probabilities, I can decide which action to take.

This is an example of a mathematical model. Like many such models, it does not involve any deep mathematical techniques; but it seems fair to call it mathematical, since it is concerned with expressing the logic of the situation in a formal way.

There are three things that are worth noting about this model, because they are typical of real operational research models. The first is that we do *not* try to make the model as *realistic as possible*. I could easily make the model more realistic. For example, I could consider a range of possible intensities of rainfall, which would affect the amount of inconvenience in not having my raincoat, and I could consider a range of possible temperatures, which would affect the amount of inconvenience in having my raincoat if it is fine. I could also enlarge on the set of possible decisions. For example, if it is raining when I leave work I could run to the bus shelter and plan to stay there until it stops; or I could consider listening to the weather forecast before making my original decision.

Now any or all of these extensions of the model may turn it into a more effective aid to decision making. However, they might equally lead me into a worse state of confusion than my original simple model, perhaps because I have no confidence in any possible way of assessing the probabilities of different temperatures or amounts of rainfall. Just as the problem for which the model is developed is one of finding the best compromise between partially conflicting objectives, so the art of model building itself is one of finding the best compromise between realistically representing the situation and being able to collect data easily and draw conclusions from the results.

The next thing to note is that the use of the model involves *optimization*. I want to know what is best for me, so I choose my decision to minimize the expected inconvenience. In this case the optimization problem is mathematically rather trivial: there are only two possible decisions, so I can compute the expected inconvenience from each, and choose the smaller. However, other problems involve quantitative variables, such as how much of some material to make, or for how long to operate a machine. These problems may require numerical techniques for finding maxima or minima, as well as skill in model building. In this book we will concentrate on these numerical techniques, while giving some

thought to model building. It is important for model builders to know a fair amount about optimization techniques, even if they can use existing computer programs to implement them. This is because of the need to compromise in model building between realism and ease of use. We can only do this if we have a fair idea of the numerical problems involved in solving any model.

The third point to be made about this and many other models is that their value is not so much that they give the best answer to the problem—which, of course, is only true to the extent that the model is valid—it is much more that the model provides a convenient *framework* for constructive thought about the problem. In the simplest form of the raincoat model we see two ingredients: the probabilities of certain events and the value of different outcomes in each circumstance. This may lead us on to more elaborate versions of the model if we are dissatisfied with the alternatives offered to us by the simple model.

In practice we often need to see the numerical solution to the model to help us to realize that the data are incomplete or incorrect. This makes the techniques for computing the solutions very important: to solve a real problem we may need to compute the answers to a number of alternative mathematical models, in which case we cannot afford to take too long solving any of them.

1.2 BASIC THEORETICAL NOTIONS

Before we begin studying optimization techniques it is worth spending some time defining the mathematical concepts which are used in the course of this book. In most cases detailed knowledge of these concepts will not be needed and an understanding of the definitions given below will suffice.

The point $\mathbf{x}^*$ in the region R is said to be a *local maximizer* of the function $f(\mathbf{x})$, subject to $\mathbf{x} \in R$, if there exists a small positive number ε such that

$$f(\mathbf{x}^*) \geqslant f(\mathbf{x})$$

for all $\mathbf{x} \in R$ which satisfy $\|\mathbf{x}^* - \mathbf{x}\| < \varepsilon$. The value of $f(\mathbf{x}^*)$ is then the corresponding *local maximum*. The symbol $\in$ is standard mathematical notation, meaning 'is a member of' or 'belongs to'. The norm, or distance measure, is not particularly important. We may use the Euclidean norm, where

$$\|\mathbf{x}\| = \left(\sum_j x_j^2 \right)^{1/2}$$

The point $\mathbf{x}^*$ is a *global maximizer* of the function $f(\mathbf{x})$, where $\mathbf{x} \in R$, if

$$f(\mathbf{x}^*) \geqslant f(\mathbf{x})$$

for *all* $\mathbf{x} \in R$. The value of $f(\mathbf{x}^*)$ is then the *global maximum* of the function $f(\mathbf{x})$ in the region R.

Definitions of *local* and *global minima* follow in the same way, replacing $\geqslant$ by $\leqslant$ in the obvious places. Otherwise we can say that $f(\mathbf{x})$ has a local or global minimum if and only if $-f(\mathbf{x})$ has a local or global maximum.

Certain optimization techniques—generally known as *hill-climbing* techniques—start with an estimate of the global maximum of the function and repeatedly try to improve upon it by finding other points which have a greater function value than the current estimate. The existence of local maxima that are not also global maxima is clearly an undesirable hazard for these optimization techniques, since it is possible that the methods will converge to a local maximizer instead of a global maximizer. Therefore we will consider some important circumstances in which local maxima must also be global maxima.

The most usual such circumstances are connected with the notions of *convexity* and *concavity*. A region R is defined to be a *convex region* if the point

$$(1-\theta)\mathbf{x}_1 + \theta\mathbf{x}_2 \qquad (0<\theta<1)$$

is always in the region, providing the points $\mathbf{x}_1$ and $\mathbf{x}_2$ also belong to it. A *convex function* is one that is never underestimated by linear interpolation, i.e. if

$$\mathbf{x} = (1-\theta)\mathbf{x}_1 + \theta\mathbf{x}_2 \qquad (0<\theta<1)$$

then

$$f(\mathbf{x}) \leqslant (1-\theta)f(\mathbf{x}_1) + \theta f(\mathbf{x}_2)$$

If this inequality holds with strict inequality, i.e. $<$ rather than $\leqslant$, the function is said to be *strictly convex*. A function $f(\mathbf{x})$ is *concave* if and only if $-f(\mathbf{x})$ is *convex*. Similarly, $f(\mathbf{x})$ is *strictly concave* if and only if $-f(\mathbf{x})$ is *strictly convex*.

Note that a linear function is both convex and concave, and also that a twice-differentiable function $f(x)$ of a single variable is convex if and only if

$$f''(x) \geqslant 0$$

everywhere, where $f''(x)$ is the second derivative of $f(x)$. Similarly, a twice-differentiable function of n variables $f(x_1, \ldots, x_n)$ is convex if its matrix of second partial derivatives is *positive semi-definite* everywhere. In other words,

$$\mathbf{s}^T\mathbf{A}\mathbf{s} \geqslant 0 \qquad \text{(for all vectors } \mathbf{s} \neq 0) \tag{1.2.1}$$

where

$$\mathbf{A} = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \dfrac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

This matrix of second partial derivatives is commonly known as the *Hessian* matrix. A matrix $\mathbf{A}$ is said to be *positive definite* if equation (1.2.1) holds with strict inequality.

The main importance of convexity comes from the following proposition. If the region R is convex and $f(\mathbf{x})$ is a convex function in R, then any local minimizer $\mathbf{x}^*$ of $f(\mathbf{x})$ in R is also a global minimizer.

To prove this, observe that if it were not so, there must exist a point $\mathbf{x}_G$ in R such that

$$f(\mathbf{x}_G) < f(\mathbf{x}^*)$$

So this means that, since R is a convex region, the points given by

$$\mathbf{x} = (1 - \theta)\mathbf{x}^* + \theta\mathbf{x}_G \qquad (0 < \theta < 1)$$

must also belong to R. Also since $f(\mathbf{x})$ is a convex function we know that

$$f(\mathbf{x}) \leqslant (1 - \theta)f(\mathbf{x}^*) + \theta f(\mathbf{x}_G)$$

which in turn implies that $f(\mathbf{x}) < f(\mathbf{x}^*)$ for all θ. However, as θ tends towards zero, so the point $\mathbf{x}$ tends towards $\mathbf{x}^*$, which contradicts the hypothesis that there must exist a positive ε such that $f(\mathbf{x}) \geqslant f(\mathbf{x}^*)$ for all $\mathbf{x}$ such that $\| \mathbf{x} - \mathbf{x}^* \| < \varepsilon$. Hence it must be true that a local minimizer of a convex function in a convex region is also a global minimizer.

An immediate corollary of this is that a local maximizer of a concave function in a convex region must also be a global maximizer. By definition, the actual value of the global maximum must be unique. However, it does not follow that the global maximizer must be unique; multiple global maximizers are possible.

Turning now to more general functions, the *Taylor series* for a continuous function $f(x)$ of a single variable x, with continuous derivatives $f'(x)$, $f''(x)\ldots$ in a given interval $a \leqslant x \leqslant b$, is defined to be

$$f(x) = f(a) + \frac{(x-a)}{1!}f'(a) + \frac{(x-a)^2}{2!}f''(a) + \cdots$$

or, equivalently,

$$f(a+x) = f(a) + \frac{x}{1!}f'(a) + \frac{x^2}{2!}f''(a) + \cdots$$

If $f(\mathbf{x})$ is a function of n variables then the Taylor series, expanded around a point $\mathbf{x}_0$, written in matrix notation becomes

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \mathbf{c}^T(\mathbf{x} - \mathbf{x}_0) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T\mathbf{A}(\mathbf{x} - \mathbf{x}_0) + \cdots$$

where the elements of the vector $\mathbf{c}$ are the first-order partial derivatives $\partial f/\partial x_j$ evaluated at the point $\mathbf{x}_0$, and where the matrix $\mathbf{A}$ is the Hessian matrix, also evaluated at the point $\mathbf{x}_0$.

Note that the Hessian matrix is a *square* matrix because it has the same number of rows and columns. It is also *symmetric*. Writing a_{ij} as the ijth element of $\mathbf{A}$, this means that

$$a_{ij} = a_{ji} \qquad \text{(for all } i, j\text{)}$$

or, alternatively,

$$\mathbf{A} = \mathbf{A}^T$$

A *diagonal* matrix is a square matrix which has zero elements everywhere except for the leading diagonal, i.e. $a_{ij} = 0$ for $i \neq j$. A special example of a diagonal matrix is the *unit* matrix, which has all its diagonal elements equal to

one. The unit matrix is usually denoted by the symbol **I**. For example,

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

If the matrix is a square one such that all the elements above the leading diagonal are zero, it is known as a *lower triangular* matrix. Similarly, an *upper triangular* matrix is one where all the elements below the leading diagonal are equal to zero.

The following useful inequality, known as *Cauchy's* inequality, is referred to in Section 4.4 on quasi-Newton methods:

$$\sum_r (x_r^2) \sum_r (y_r^2) \geqslant \left(\sum_r x_r y_r \right)^2$$

or equivalently in matrix notation:

$$\mathbf{x}^T\mathbf{x}\mathbf{y}^T\mathbf{y} \geqslant (\mathbf{x}^T\mathbf{y})^T(\mathbf{x}^T\mathbf{y})$$

The number of ways of choosing r items from a group of n, where order is not important, is called the number of *combinations* of n items taken r at a time and is given by

$$\frac{n!}{(n-r)!r!}$$

The number of possible combinations of n items is 2^n. For example, the possible combinations of the three letters A, B, C are

None at all, A, B, C, AB, AC, BC, ABC

The number of *permutations* or arrangements of n items taken r at a time, where order matters, is given by

$$\frac{n!}{(n-r)!}$$

Finally, computer *rounding-off* errors are referred to throughout this book. These errors occur because most real numbers cannot be represented exactly on a

Table 1.2.1

	Fraction	Exact decimal	Binary (to 8 binary places)	Decimal representation of binary
A	2/5	0.4	0.01100110	0.39843750
B	3/5	0.6	0.10011001	0.59765625
A + B	1	1.0	0.11111111	0.99609375

computer. Computers store numbers in base 2 (binary) rather than base 10 (decimal). Each number is represented as a binary fraction, called the mantissa, multiplied by 2 raised to the power of some number, the exponent. The mantissa is a string of binary digits (zero or one) which is truncated after a certain number of places (typically 26). Table 1.2.1 shows an example of the large rounding-off errors that arise when the two fractions 2/5 and 3/5 are stored as binary numbers, truncated after only 8 places.

PART 1

UNCONSTRAINED OPTIMIZATION