Optimization Theory for Large Systems

LEON S. LASDON

Optimization Theory for Large Systems

LEON S. LASDON CASE WESTERN RESERVE UNIVERSITY

THE MACMILLAN COMPANY
COLLIER-MACMILLAN LIMITED, LONDON

First Printing

Library of Congress catalog card number: 78–95301

THE MACMILLAN COMPANY
866 Third Avenue, New York, New York 10022
COLLIER-MACMILLAN CANADA, LTD., TORONTO, ONTARIO

Printed in the United States of America

Preface

Many decision problems are now formulated as mathematical programs, requiring the maximization or minimization of an objective function subject to constraints. Such programs often have special structure. In linear programming, the nonzero elements of the constraint matrix may appear in diagonal blocks, except for relatively few rows or columns. Nonlinear programs may become linear if certain variables are assigned fixed values, or the functions involved may be additively separable. Some definite structure is almost always found in truly large problems, since these commonly arise from a linking of independent subunits in either time or space. By developing specialized solution algorithms to take advantage of this structure, significant gains in computational efficiency and reductions in computer memory requirements may be achieved. Such methods are mandatory for truly large problems, which cannot otherwise be solved because of time and/or storage limitations.

The past decade has seen the identification of many classes of structured problems and the development of a great many algorithms for their solution. This period was initiated by the publication of the Dantzig–Wolfe decomposition principle in 1960. However, there was significant activity even before that. Growth has been explosive, with techniques developed first for linear, then for certain classes of nonlinear programs. The literature is now very large. This forces a beginner exploring the field to search through a maze of articles in many different journals. Even after doing so, the relationships between the various methods are likely to be obscured. Clearly a need to collect together the best of this literature and unify it has existed for some time.

This book is an attempt to fill this need. It discusses some of the most important algorithms for optimizing large systems. The emphasis throughout is on developing the various methods in a straightforward and logical manner from a small set of basic ideas and principles. In so doing, relationships between various procedures are made clear. Application of many methods is illustrated by numerical examples, and there are problems at the end of each chapter. In addition, much of the available computational experience has been included, as well as comments on various computational options. This information should prove useful to those interested in applications, as should those sections dealing with formulation of problems. Chapter 2 deals entirely with how large-scale programs can arise from real-world problems, and other similar examples are found throughout.

Notes for the book have been used in a one-semester graduate course in the Operations Research Department at Case Western Reserve University and in the Faculty of Industrial and Management Engineering at the Technion, Haifa, Israel. The first chapter contains most of the background material on linear and nonlinear programming needed in the rest of the text. The two appendixes deal with convex functions and their conjugates, and with subgradients and directional derivatives of convex functions. These are based on the work of R. T. Rockafellar, are self-contained, and are of much interest in themselves. The ideas in them are used extensively in Chapters 8 and 9 but very little elsewhere.

The greatest single contributor to large-scale mathematical programming is George Dantzig. This is reflected in the contents of this volume. The contributions of Arthur Geoffrion, Ben Rosen, and Philip Wolfe also play a major role. James Schoeffler introduced me to this area. His interest and that of Mihajlo Mesarovic have stimulated mine. My thanks to Allan Waren for aid in preparing Chapter 1. I wish to express appreciation to the School of Management, Case Western Reserve University, and to the Faculty of Industrial and Management Engineering at the Technion for providing secretarial assistance. Thanks also to Miss Christine Yamamoto and Miss Ellon Waters for their excellent work in typing the first draft, and to Mrs. Orah Naor for her fine efforts in typing the manuscript. Finally, special thanks to my wife Louanne, who suffered with me and encouraged me throughout.

L. S. L.

Contents

3. The Dantzig–Wolfe Decomposition Principle 144

4. Solution of Linear Programs with Many Columns by Column-Generation Procedures 207

5. Partitioning and Relaxation Procedures in Linear Programming 267

6. Compact Inverse Methods 304

7. Partitioning Procedures in Nonlinear Programming 358

8. Duality and Decomposition in Mathematical Programming 396

9. Decomposition By Right-Hand-Side Allocation 460

1

Linear and Nonlinear Programming

The problem of mathematical programming is that of maximizing or minimizing an objective function $f(x_1 \cdots x_n)$ by choice of the vector $x = (x_1 \cdots x_n)'$. The variables x_i may be allowed to take on any values whereupon the problem is one of unconstrained minimization or they may be restricted to take on only certain allowable values, whereupon the problem is constrained. Only problems in which (1) the variables x_i can vary continuously within the region of interest and (2) the objective and constraint functions are continuous and differentiable are considered here.

If the problem is constrained, its difficulty depends critically on the nature of the constraints, i.e., linear, nonlinear, etc. We consider first the unconstrained case, then the more difficult, constrained one. The constrained case will be divided into two parts: linear constraints and linear objective function (linear programming) and at least one nonlinear constraint and/or nonlinear objective (nonlinear programming).

1.1 Unconstrained Minimization

Necessary and Sufficient Conditions for an Unconstrained Minimum. The problem here is to maximize or minimize a function of n variables, $f(x)$, with no restrictions on the variables x. Many real-life problems are of this form, where whatever constraints are present do not restrict the optimum. Also, many problems in which the constraints are binding can be converted to unconstrained problems or sequences of such problems. Since the problem of maximizing $f(x)$ is equivalent to that of minimizing $-f(x)$, only the minimization problem is considered.

A point x^* is said to be a global minimum of $f(x)$ if

$$f(x^*) \leq f(x) \tag{1}$$

for all x. If the strict inequality holds for $x \neq x^*$ the minimum is said to be *unique*. If (1) holds only for all x in some neighborhood of x^*, then x^* is said to be a local or relative minimum of $f(x)$, since x^* is only the best point in the immediate vicinity, not in the whole space.

If $f(x)$ is continuous and has continuous first and second partial derivatives for all x, the necessary conditions for a local minimum are [3]

$$\frac{\partial f(x^*)}{\partial x_i} = 0, \qquad i = 1, 2, \ldots, n \tag{2}$$

and that the matrix of second partial derivatives evaluated at x^* be positive semidefinite. Any point x^* satisfying (2) is called a stationary point of $f(x)$. Sufficient conditions for a relative minimum are that the matrix of second partial derivatives of $f(x)$ evaluated at x^* be positive definite and (2) hold.

Numerical Methods for Finding Unconstrained Minima. The most obvious approach to finding the minimum of $f(x)$ is to solve (2). These are a set of n equations, usually nonlinear, in the n unknowns x_i. Unfortunately the task of solving large sets of nonlinear equations is very difficult. The function $f(x)$ may be so complex that it is difficult even to write out (2) in closed form. Further, even if (2) could be solved, there would be no guarantee that a given solution was not a maximum, saddle point, etc., rather than a minimum. Thus other approaches must be considered.

Gradient. If $f(x)$ is continuous and differentiable, a number of minimization techniques using the gradient of $f(x)$, written $\nabla f(x)$, are available. The gradient is the vector whose ith component is $\partial f/\partial x_i$. It points in the direction of maximum rate of increase of $f(x)$ ($-\nabla f$ points in the direction of greatest decrease). The vector ∇f is, at any point x_0, normal to the contour of constant function value passing through x_0.

Steepest Descent. The method of steepest descent for finding a local minimum of $f(x)$ proceeds as follows. Start at some initial point x_0 and compute $\nabla f(x_0)$. Take a step in the direction of steepest descent, $-\nabla f(x_0)$, using a step length α_0, to obtain a new point x_1. Repeat the procedure until some stop criterion is satisfied. This process is described by the relations

$$\begin{gathered} x_0 \text{ given} \\ x_{i+1} = x_i - \alpha_i \nabla f(x_i) \qquad i = 0, 1, 2, \ldots \end{gathered} \tag{3}$$

where $\alpha_i > 0$. The process will, under very mild restrictions [4] on $f(x)$, converge to at least a local minimum of $f(x)$, if the α_i are chosen so that

$$f(x_{i+1}) < f(x_i) \tag{4}$$

for all i, i.e., if the function is made to decrease at each step. Since the function is initially decreasing in the directions given by $-\nabla f(x_i)$, there always exist $\alpha_i > 0$ such that (4) is satisfied.

Step Length and Optimum Gradients. One way to find α_i satisfying (4) is to choose α_i to minimize the function

$$g(\alpha) = f[x_i - \alpha \nabla f(x_i)] \tag{5}$$

Note that x_i and $\nabla f(x_i)$ are known vectors so that the only variable in (5) is α. The adaptation of the method of steepest descent which uses (5), called the method of optimum gradients [4], is described by

$$x_0 \text{ given}$$
$$s_i = -\nabla f(x_i)$$

Choose $\alpha = \alpha_i$ by minimizing $g(\alpha)$ in (5),

$$x_{i+1} = x_i + \alpha_i s_i \tag{6}$$

Set $i = i + 1$ and repeat.

Geometrically, α_i is chosen by minimizing $f(x)$ along the direction s_i starting from x_i. At a minimum,

$$\left.\frac{dg}{d\alpha}\right|_{\alpha=\alpha_i} = s_i' \nabla f(x_i + \alpha_i s_i) = 0 \tag{7}$$

so the vector $x_i + \alpha s_i$ must be tangent to a contour at $\alpha = \alpha_i$, for dg is then zero for small changes $d\alpha$. Since $\nabla f(x_{i+1})$ is normal to the same contour, successive steps are at right angles to one another. Practical methods for carrying out the one-dimensional minimization are discussed later in this section.

Stop Criteria. Some possible stop criteria are as follows:

1. Since, at a minimum $\partial f/\partial x_i = 0$, stop when

(a) $$\left|\frac{\partial f}{\partial x_i}\right| < \epsilon \qquad i = 1, 2, \ldots, n$$

or

(b) $$\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i}\right)^2 < \epsilon$$

2. Stop when the change in function is less than some limit η, i.e.,

$$|f(x_{i+1}) - f(x_i)| < \eta$$

Others are possible. Criterion 2 appears to be the more dependable of the two, provided it is satisfied for several successive values of i.

Local versus Global Minima. The most that can be guaranteed of this or any other iterative minimization technique is that it will find a local minimum, in general the one "nearest" the starting point x_i. To attempt to find all local minima (and thus the global minimum), the method most used is to repeat the minimization from many different initial points.

Numerical Difficulties. The fact that successive steps of the optimum gradient method are orthogonal leads to very slow convergence for some functions. If the function contours are hyperspheres (circles in two dimensions), the method finds the minimum in one step. However, if the contours are in any way eccentric, an inefficient zigzag behavior results, as shown in Figure 1-1. This occurs because, for eccentric contours, the gradient direction is generally quite different from the direction to the minimum. Many, if not most, of the functions occurring in practical applications are ill-behaved in

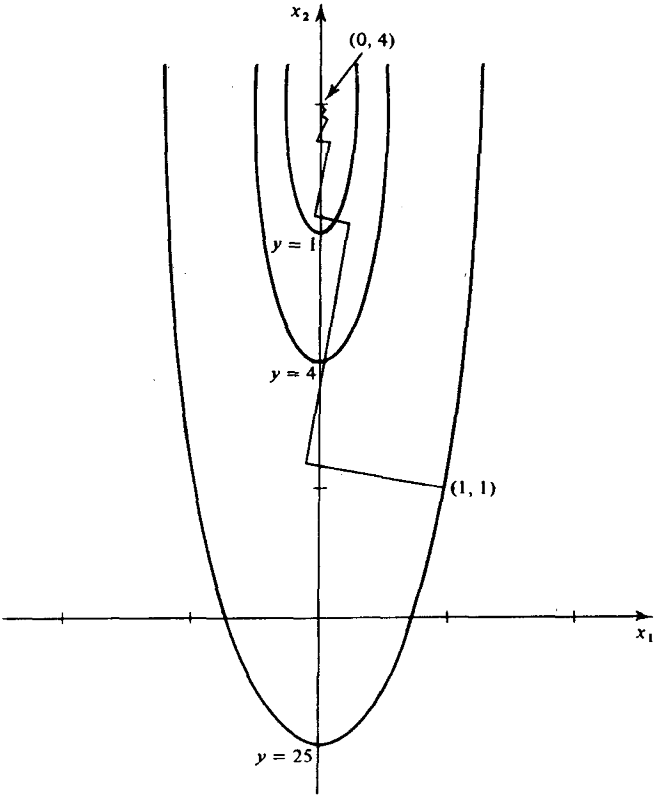

FIGURE 1-1 Equivalue lines of $y = 16x_1^2 + (x_2 - 4)^2$, normal steepest-descent method.

that their contours are eccentric or nonspherical. Thus more efficient schemes are desirable.

"Second-Order" Gradient Methods. Recently, a number of minimization techniques have been developed which substantially overcome the above difficulties. What appear to be the best of these will be described in detail. First, however, the logic behind these methods will be explained.

Since the first partial derivatives of a function vanish at the minimum, a Taylor-series expansion about the minimum x^* yields

$$f(x) \doteq f(x^*) + \tfrac{1}{2}(x - x^*)' H_f(x^*)(x - x^*) \tag{8}$$

where $H_f(x^*)$, the matrix of second partials of f evaluated at x^*, is positive definite. Thus the function behaves like a pure quadratic in the vicinity of x^*. It follows that the only methods which will minimize a general function quickly and efficiently are those which (1) work well on a quadratic and (2) are guaranteed to converge eventually for a general function. All others will be slow, at least in the vicinity of the minimum (see Figure 1-1), and often elsewhere.

Conjugate Directions. General minimization procedures can be designed which will minimize a quadratic function of n variables in n steps [5–7]. Most, if not all, are based on the ideas of conjugate directions.

The general quadratic function can be written

$$q(x) = a + b'x + \tfrac{1}{2}x'Ax \tag{9}$$

where A is positive definite and symmetric. Let x^* minimize $q(x)$. Then

$$\nabla q(x^*) = b + Ax^* = 0 \tag{10}$$

Given a point x_0 and a set of linearly independent directions $\{s_0, s_1, \ldots, s_{n-1}\}$, constants β_i can be found such that

$$x^* = x_0 + \sum_{i=0}^{n-1} \beta_i s_i \tag{11}$$

If the directions s_i are A-conjugate, i.e., satisfy

$$s_i'As_j = 0, \qquad i \neq j, \quad i, j = 0, 1, \ldots, n - 1 \tag{12}$$

and none are zero, then the s_i are easily shown to be linearly independent and the β_i can be determined from (11) as follows:

$$s_j'Ax^* = s_j'Ax_0 + \sum_{i=0}^{n-1} \beta_i s_j'As_i \tag{13}$$

Using (12),

$$s_j'Ax^* = s_j'Ax_0 + \beta_j s_j'As_j \tag{14}$$

and, using (10),

$$\beta_j = -(b + Ax_0)' \frac{s_j}{s_j'As_j} \tag{15}$$

Now consider an iterative minimization procedure, starting at x_0 and successively minimizing $q(x)$ down the directions $s_0, s_1, s_2, \ldots, s_{n-1}$, where these directions satisfy (12). Successive points are then determined by the relations

$$x_{i+1} = x_i + \alpha_i s_i, \qquad i = 0, 1, \ldots, n - 1 \tag{16}$$

where α_i is determined by minimizing $f(x_i + \alpha s_i)$, as in the optimum gradient method, so that

$$s_i' \nabla q(x_{i+1}) = 0 \tag{17}$$

Using (10) in (17) gives

$$s_i'(b + A(x_i + \alpha_i s_i)) = 0 \tag{18}$$

or

$$\alpha_i = -(b + Ax_i)' \frac{s_i}{s_i'As_i} \tag{19}$$

From (16),

$$x_i = x_0 + \sum_{j=0}^{i-1} \alpha_j s_j \tag{20}$$

so that

$$x_i'As_i = x_0'As_i + \sum_{j=0}^{i-1} \alpha_j s_j'As_i = x_0'As_i \tag{21}$$

Thus (19) becomes

$$\alpha_i = -(b + Ax_0)' \frac{s_i}{s_i'As_i} \tag{22}$$

which is identical to (15). Hence this sequential process leads, in n steps, to the minimum x^*.

Method of Fletcher and Powell. A method recently presented by Fletcher and Powell [5] is probably the most powerful general procedure now known [8] for finding a local minimum of a general function, $f(x)$. It is designed so that, when applied to a quadratic, it minimizes in n iterations. It does this by generating conjugate directions.

Central to the method is a symmetric positive definite matrix H_i which is updated at each iteration, and which supplies the current direction of motion, s_i, by multiplying the current gradient vector. An iteration is described by the following:

$$\begin{aligned} H_0 &= \text{any positive definite matrix} \\ s_i &= -H_i \, \nabla f(x_i) \end{aligned}$$

Choose $\alpha = \alpha_i$ by minimizing $f(x_i + \alpha s_i)$,

$$\begin{aligned} \sigma_i &= \alpha_i s_i \\ x_{i+1} &= x_i + \sigma_i \\ H_{i+1} &= H_i + A_i + B_i \end{aligned} \tag{23}$$

where the matrices A_i and B_i are defined by

$$\begin{aligned} A_i &= \frac{\sigma_i \sigma_i'}{\sigma_i' y_i}, \qquad y_i = \nabla f(x_{i+1}) - \nabla f(x_i) \\ B_i &= \frac{-H_i y_i y_i' H_i}{y_i' H_i y_i} \end{aligned} \tag{24}$$

Note that the numerators of A_i and B_i are both matrices, while the denominators are scalars. Fletcher and Powell prove the following:

1. The matrix H_i is positive definite for all i. As a consequence of this, the method will usually converge, since

$$\frac{d}{d\alpha} f(x_i + \alpha s_i)\bigg|_{\alpha = 0} = -\nabla f'(x_i) H_i \, \nabla f(x_i) < 0 \tag{25}$$

   i.e., the function f is initially decreasing along the direction s_i, so that the function can be decreased at each iteration by minimizing down s_i.

2. When the method is applied to the quadratic (9), then
   (a) The directions s_i (or equivalently σ_i) are A-conjugate, thus leading to a minimum in n steps.