WILLIAM CONLEY
University of Wisconsin-Green Bay

Computer Optimization Techniques

Revised Edition

PBI
a petrocelli book
new york / princeton

Copyright © 1984 by Petrocelli Books, Inc.
All rights reserved.

Designed by Joan Greenfield

Printed in the United States of America.
1 2 3 4 5 6 7 8 9 10

Library of Congress Cataloging in Publication Data

Conley, William, 1948–
Computer optimization techniques.

Bibliography: p.
Includes index.
1. Mathematical optimization—Data processing.
2. Integer programming—Data processing. I. Title.
QA402.5.C64 1984 519.7′7 83-26347
ISBN 0-89433-213-9

Computer Optimization Techniques — *Revised Edition*

Dedicated to my Family and Friends

此为试读，需要完整PDF请访问：www.ertongbook.com

INTRODUCTION

Since writing the first edition of *Computer Optimization Techniques*, the powerful multi-stage Monte Carlo optimization technique has been developed. Therefore, this revised edition includes multi-stage Monte Carlo optimization in Chapters 7 and 8. Many more multi-stage Monte Carlo optimization (MSMCO) examples are also referred to in the Suggested Reading section.

The two interesting research areas in MSMCO today are the applications and the answer to the question, "Does the mean, median and/or mode of several MSMCO approximate optimals converge to the true but unknown optimal in very difficult optimization problems?" So let's begin by trying to simplify optimization. Computer science has advanced to the point where it is possible to greatly simplify integer programming. This book is an attempt to do just that.

This simplification takes many forms. It frequently allows us to solve integer, linear, and nonlinear programming problems that were solvable before in a much easier fashion. The technique is easier theoretically, easier on the programmer, and cheaper in actual dollars. The simplification also allows us to solve integer, linear, and nonlinear programming problems that were heretofore unsolvable. These solutions are as easy to obtain as those from theoretically solvable problems. The complexities or nonlinearities of the objective and/or constraint functions will not make any difference to the computer, even though they completely ruin the traditional simplex algorithm approach.

How difficult is this approach and how useful is it? These are valid questions which should be answered. It is necessary to have a knowledge of FORTRAN (we used FORTRAN IV) up to the point where the programmer understands DO-loops. A knowledge of subscripted variables would also be helpful, especially for the optimization problems with hundreds of variables.

No mathematical background is required other than knowing that an equation is something with an equals sign and variables connected in some fashion through addition, subtraction, multiplication, and division signs. It is never necessary to analyze the particular equation to see if it meets certain conditions such as linearity. The system of equations and constraints never needs to be tested for redundancy or cycling as in traditional methods. In fact, the less a person knows theoretically about the system to be optimized, the better. That way, there won't be the temptation to try other methods.

This does not mean to imply that the very considerable, elegant and useful mathematical programming theory developed to date is not useful. On the contrary, this area has been one of the most useful and practically productive areas of mathematics for years. Certainly, anyone who is an expert in these techniques should continue to use and develop them. However, this book can help the expert by providing a method for obtaining a good answer quickly to the numerous theoretically unsolvable problems that arise in applications.

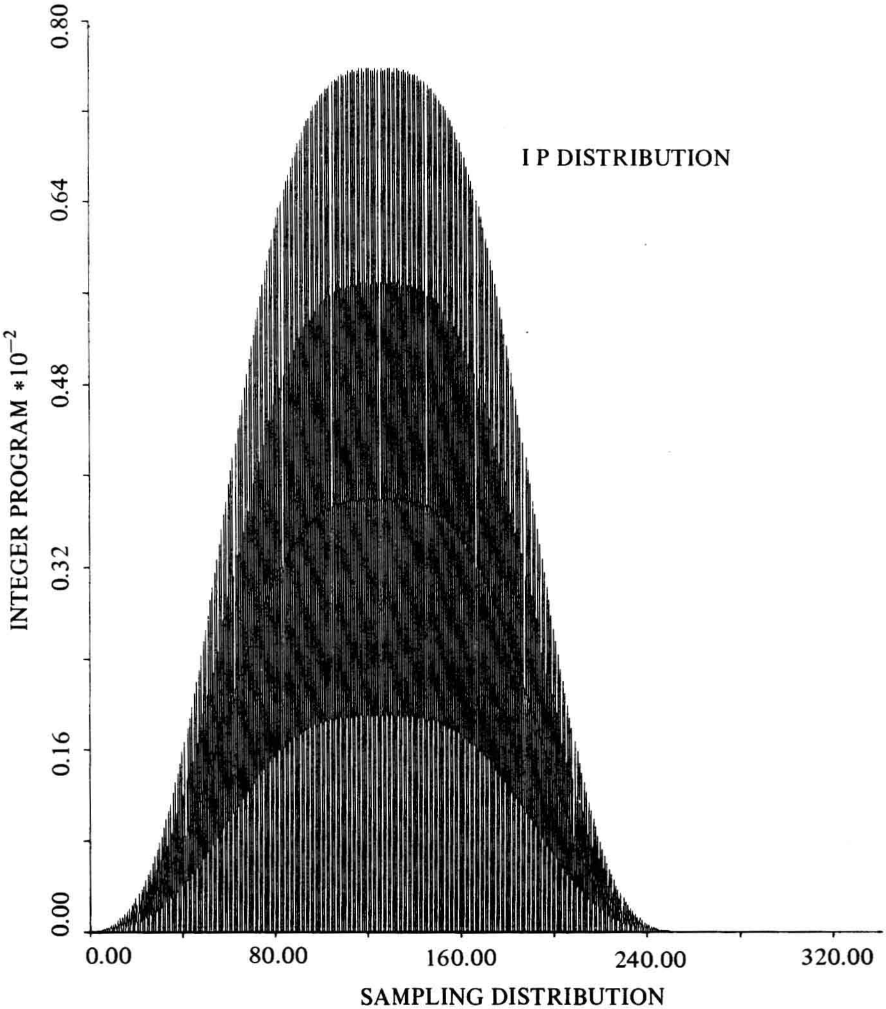
Also, the computer technique that is explained and illustrated in this book should make available optimization solutions to people who have little or no time to develop theoretical expertise in mathematical programming, specifically business managers, beginning business students, advanced business students whose expertise is not in quantitative areas, engineers who do not concentrate on optimization, administrators, accountants, scientists, researchers, small businessmen, decision makers working on a quantitative project, and people who never liked mathematics because it was too difficult.

This book is not an attack on theoretical mathematics but merely the result of a realization that computer technology has made possible, in just the last few years, the simplification and advancement of an extremely complicated and useful area of applied mathematics by taking a different philosophical approach to mathematical programming.

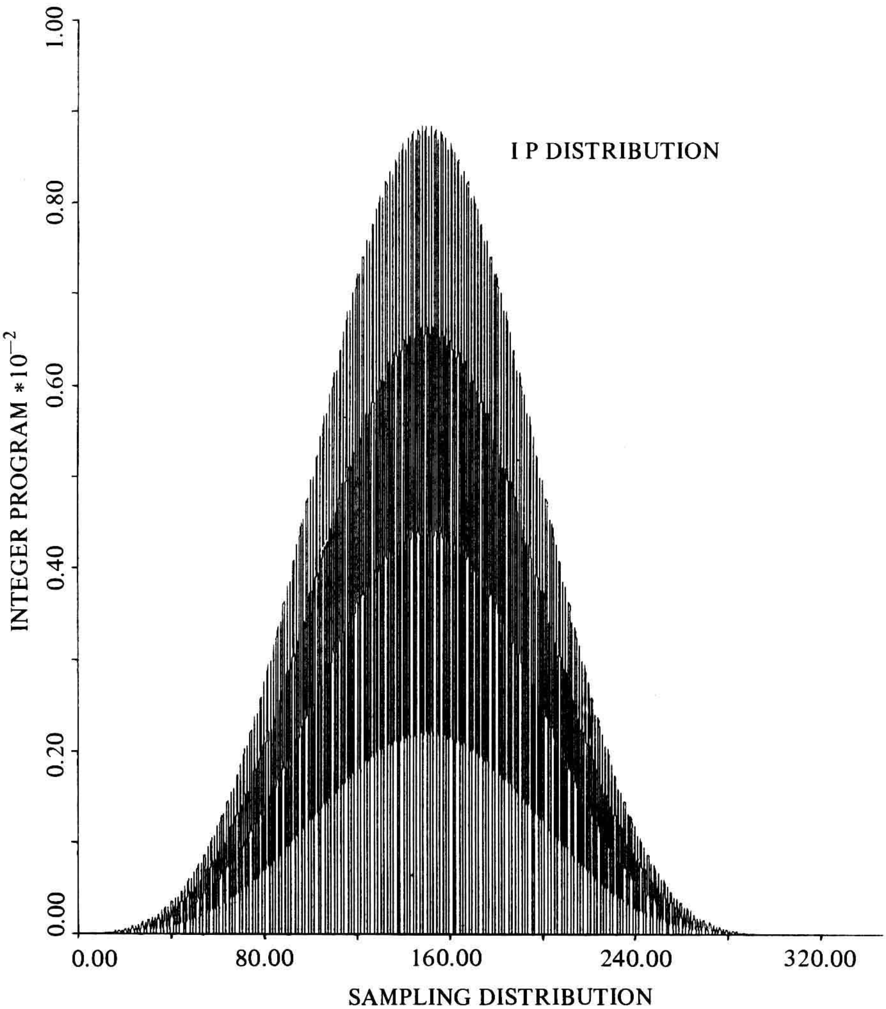
The book is divided into two parts. Conceptually, they are almost the same. Part One allows us to solve problems with a small number of variables, while Part Two allows us to deal with problems that have a great many variables. We will write many programs in Part One to illustrate the technique and obtain the solutions to our stated questions. This will also serve to reinforce the technique and show a variety of applications. However, it is really only necessary to understand one problem somewhere along the way because the technique is the same in each case. We are letting the computer do the difficult work. The same is true in Part Two. One really needs to understand only a few programs (dozens are presented) to understand the technique.

Lastly, an attempt is made to explain everything completely, thoroughly, and repeatedly. This is to make the book and the solution of mathematical programming problems accessible to most everyone. Therefore, readers having an extensive background in mathematics and programming can just move at their own pace through the discussion.

In fact, it is quite possible to understand this book completely in a short period of time. In these days of increasing sophistication and complication, maybe this will be helpful. I hope so.



of $P = x_1 + 3x_2 + 4x_3 + 5x_4 + 12x_5$ subject to $0 \leqslant x_i \leqslant 10 \quad i = 1, 5$



of $P = 4x_1 + 5x_2 + 6x_3 + 7x_4 + 8x_5$ subject to $0 \leqslant x_i \leqslant 10 \quad i = 1, 5$

CONTENTS

Introduction ix

Part One

1 Optimization in the Computer Age 3

2 Solving Integer Programming Problems by Looking at All Possibilities 15

3 Optimization Problems of Two through Eight Variables 25

Part Two

4 Monte Carlo Integer Programming 101

5 Integer Programming Problems with a Few Variables 115

6 Integer Programming Problems with Many Variables 129

7 Multi-Stage Monte Carlo Optimization 147

8 A Nine Hundred-Variable Nonlinear Problem 165

Suggested Reading 179

Appendices

A Sampling Distributions of Feasible Solutions of Selected Integer Programming Problems 185

B How to Obtain Sampling Distributions of Feasible Solutions of Integer Programming Problems 199

C How to Solve a System of Equations 207

D Additional Business Examples 213

E The Impact of Computers on the Philosophy of Optimization 241

Index 243

Part One

此为试读,需要完整PDF请访问: www.ertongbook.com

CHAPTER 1

Optimization in the Computer Age

Mathematically, at least in our context, optimization means to find the maximum of a function or process that we want to maximize or to find the minimum of a function or process that we want to minimize. For example, we might wish to maximize a profit function or an output function of a process. Or, we might wish to minimize a cost function. Let's look at a few examples.

Suppose a company manufactures two products, A and B. Let x be the number of units of A produced and y the number of units of B produced. Suppose further that each unit of A returns a profit of two dollars and each unit of B returns a profit of three dollars. Therefore, the profit function would be written

$$P = 2x + 3y$$

where P is the profit in dollars.

Now, the question might naturally arise, how do we maximize this equation? Well, as stated the equation allows any values for x and y, therefore, it is only necessary to produce as much of A and B as possible to maximize P. P becomes infinitely large as either x or y or both go to infinity.

However, let's add a few restrictions to the variables x and y. Let's assume that the company's position is such that x must be between 0 and 10 inclusive, and y must be between 0 and 10 inclusive. In symbols this is $0 \leq x \leq 10$ and $0 \leq y \leq 10$. Let's further assume that x and y can only take integer values. This means that each possible x and y value must be a counting number or the negative of a counting number or zero. Equations to be optimized whose solution coordinates are restricted to integers (usually nonnegative integers in practical problems) are called integer programming problems. If we allow solutions that are not integer valued, like $x = .666, y = 7.5$, then we have a linear programming problem or a nonlinear noninteger programming problem. We, of course, can have a nonlinear integer programming problem. This is a problem in which either the function to be optimized and/or the constraints (conditions or restrictions) on the variables are nonlinear (they have squared and cubed terms, etc.). Also, in a nonlinear integer programming problem only integer coordinate solutions are allowed.

This book will deal mainly with integer programming problems (whole number coordinates for the solutions). But, fortunately, most applied problems require integer solutions. These are more difficult and sometimes almost impossible to solve theoretically. Later we hope to present a case for using integer solutions even in most cases where noninteger solutions are acceptable.

Getting back to our function to maximize, let us state the integer programming problem as maximize $P = 2x + 3y$ subject to $0 \leq x \leq 10$, $0 \leq y \leq 10$, and x and y must be integers. Therefore, let's look at the x and y pairs that are possibilities for the optimum. The following points are the only ones that satisfy the constraints:

(0,0) (1,0) (2,0) (3,0) (4,0) (5,0) (6,0) (7,0) (8,0) (9,0) (10,0)

(0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1) (7,1) (8,1) (9,1) (10,1)

(0,2) (1,2) (2,2) (3,2) (4,2) (5,2) (6,2) (7,2) (8,2) (9,2) (10,2)

(0,3) (1,3) (2,3) (3,3) (4,3) (5,3) (6,3) (7,3) (8,3) (9,3) (10,3)

(0,4) (1,4) (2,4) (3,4) (4,4) (5,4) (6,4) (7,4) (8,4) (9,4) (10,4)

(0,5) (1,5) (2,5) (3,5) (4,5) (5,5) (6,5) (7,5) (8,5) (9,5) (10,5)

(0,6) (1,6) (2,6) (3,6) (4,6) (5,6) (6,6) (7,6) (8,6) (9,6) (10,6)

(0,7) (1,7) (2,7) (3,7) (4,7) (5,7) (6,7) (7,7) (8,7) (9,7) (10,7)

(0,8) (1,8) (2,8) (3,8) (4,8) (5,8) (6,8) (7,8) (8,8) (9,8) (10,8)

(0,9) (1,9) (2,9) (3,9) (4,9) (5,9) (6,9) (7,9) (8,9) (9,9) (10,9)

(0,10) (1,10) (2,10) (3,10) (4,10) (5,10) (6,10) (7,10) (8,10) (9,10) (10,10)

Let's solve this problem by listing the 121 possible ordered pairs with their resultant P value in each case and then merely select the one that gives the largest value for P.

Possible points (amounts of A and B to be made) are sometimes called feasible solutions.

Points	$P = 2x + 3y$ value
(0,0)	0
(1,0)	2
(2,0)	4
(3,0)	6
(4,0)	8
(5,0)	10
(6,0)	12
(7,0)	14
(8,0)	16
(9,0)	18
(10,0)	20
(0,1)	3
(1,1)	5
(2,1)	7
(3,1)	9
(4,1)	11
(5,1)	13
(6,1)	15
(7,1)	17
(8,1)	19
(9,1)	21
(10,1)	23
(0,2)	6
(1,2)	8
(2,2)	10
(3,2)	12
(4,2)	14
(5,2)	16
(6,2)	18
(7,2)	20
(8,2)	22
(9,2)	24
(10,2)	26
(0,3)	9
(1,3)	11
(2,3)	13
(3,3)	15
(4,3)	17
(5,3)	19
(6,3)	21
(7,3)	23

Points (continued)	$P = 2x + 3y$ value (continued)
(8,3)	25
(9,3)	27
(10,3)	29
(0,4)	12
(1,4)	14
(2,4)	16
(3,4)	18
(4,4)	20
(5,4)	22
(6,4)	24
(7,4)	26
(8,4)	28
(9,4)	30
(10,4)	32
(0,5)	15
(1,5)	17
(2,5)	19
(3,5)	21
(4,5)	23
(5,5)	25
(6,5)	27
(7,5)	29
(8,5)	31
(9,5)	33
(10,5)	35
(0,6)	18
(1,6)	20
(2,6)	22
(3,6)	24
(4,6)	26
(5,6)	28
(6,6)	30
(7,6)	32
(8,6)	34
(9,6)	36
(10,6)	38
(0,7)	21
(1,7)	23
(2,7)	25
(3,7)	27

Points (continued)	$P = 2x + 3y$ value (continued)	Points (continued	$P = 2x + 3y$ value (continued)
(4,7)	29	(2,9)	31
(5,7)	31	(3,9)	33
(6,7)	33	(4,9)	35
(7,7)	35	(5,9)	37
(8,7)	37	(6,9)	39
(9,7)	39	(7,9)	41
(10,7)	41	(8,9)	43
(0,8)	24	(9,9)	45
(1,8)	26	(10,9)	47
(2,8)	28	(0,10)	30
(3,8)	30	(1,10)	32
(4,8)	32	(2,10)	34
(5,8)	34	(3,10)	36
(6,8)	36	(4,10)	38
(7,8)	38	(5,10)	40
(8,8)	40	(6,10)	42
(9,8)	42	(7,10)	44
(10,8)	44	(8,10)	46
(0,9)	27	(9,10)	48
(1,9)	29	(10,10)	50

We can see that, as expected, the optimum solution (the one that maximizes the profit is $x = 10$ units of A and $y = 10$ units of B.

This may seem like a lot of work to obtain this rather obvious result. However, it should be noted that conceptually it is an easy approach, namely, just examine all possible points. Also, it will always lead to the correct answer. This will be especially useful when the function to be maximized or minimized and/or the constraints are sufficiently complicated so that the solution is difficult to obtain either by inspection or through mathematical theory. This is frequently the case in applications.

Of course, the approach we take, namely, listing all possible solutions, is extremely tedious for people even though it is straightforward. However, a computer just loves repetitive, tedious work and will produce the answer in seconds. And as the speed and capacity of computers increase this technique will become more and more practical.

Let's look at another example. Try to minimize the cost equation $C = 2x^2 - y^2 + xy$ where x can take the values between 0 and 5 and y can take the values between 0 and 5, and x and y must be integers. The possible points meeting the constraints are as follows:

(0,0) (1,0) (2,0) (3,0) (4,0) (5,0)

(0,1) (1,1) (2,1) (3,1) (4,1) (5,1)

(0,2) (1,2) (2,2) (3,2) (4,2) (5,2)

(0,3) (1,3) (2,3) (3,3) (4,3) (5,3)

(0,4) (1,4) (2,4) (3,4) (4,4) (5,4)

(0,5) (1,5) (2,5) (3,5) (4,5) (5,5)

Let's list the possible points (combinations of x and y) along with the corresponding $C = 2x^2 - y^2 + xy$ value and take the points which produce the minimum. There are 36 possibilities:

Points	$C = 2x^2 - y^2 + xy$ value	Points (continued)	$C = 2x^2 - y^2 + xy$ value (continued)
(0,0)	0		
(1,0)	2	(1,3)	−4
(2,0)	8	(2,3)	5
(3,0)	18	(3,3)	18
(4,0)	32	(4,3)	35
(5,0)	50	(5,3)	56
(0,1)	−1	(0,4)	−16
(1,1)	2	(1,4)	−10
(2,1)	9	(2,4)	0
(3,1)	20	(3,4)	14
(4,1)	35	(4,4)	32
(5,1)	54	(5,4)	54
(0,2)	−4	(0,5)	−25
(1,2)	0	(1,5)	−18
(2,2)	8	(2,5)	−7
(3,2)	20	(3,5)	8
(4,2)	36	(4,5)	27
(5,2)	56	(5,5)	50
(0,3)	−9		

We can see that the optimum solution (the one that minimizes the cost) is $x = 0$ and $y = 5$. This yields a C value of -25.

Now, let's try to maximize $P = 3x^2 - 2y$ where x and y must be nonnegative integers and, further, they must satisfy $y \le -.5x + 5$ and $y \le -2x + 10$. A graph of the related equalities is given in Figure 1.1. The shaded region shows the area that satisfies the inequalities. Generally speaking, with an inequality of the form $y \le mx + b$ the solution is the half plane below the line $y = mx + b$. This is the case here. Let's now list the integer combinations that satisfy the constraints along with their corresponding function values and take the coordinates that give us a maximum P under the constraints:

此为试读，需要完整PDF请访问：www.ertongbook.com