LINEAR COMPLEMENTARITY, LINEAR AND NONLINEAR PROGRAMMING

Internet Edition

Katta G. Murty
Department of Indutrial and Operations Engineering
University of Michigan, Ann Arbor

The Internet edition of this book has been prepared by

Feng-Tien Yu
Department of Indutrial and Operations Engineering
University of Michigan, Ann Arbor

Partial funding for presenting this book on the web has been provided by the Spring/Summer Research Grants Program Office of the Vice President for Research Horace H. Rackham School of Graduate Studies The University of Michigan, Ann Arbor

Katta G. Murty
Dept. of Industrial and Operations Engineering
The University of Michigan
1205 Beal Avenue
Ann Arbor, MI 48109-2117, U. S. A.
TEL: +1 734 763-3513
E-mail: murty@umich.edu
WWW: http://www-personal.engin.umich.edu/~ murty/

Feng-Tien Yu
Dept. of Industrial and Operations Engineering
The University of Michigan
1205 Beal Avenue
Ann Arbor, MI 48109-2117, U. S. A.
TEL: +1 734 764-8583
E-mail: fengtien@umich.edu
WWW: http://www-personal.umich.edu/~ fengtien/

Other books by Katta G. Murty :

—– **Linear programming, John Wiley & Sons, 1983**

—– **Linear and combinatorial programming, R. E. Krieger, 1985**

—– **Network programming, Prentice Hall, 1992**

—– **Operation research: deterministic optimization models, Prentice Hall, 1995**

PREFACE

INTRODUCTION

I am grateful for the enthusiastic reception given to my book *Linear and Combinatorial Programming* published in 1976. Many readers from all over the world commented that they liked Chapter 16 on the Linear Complementarity Problem (LCP) in this book, but found it too brief, and suggested that a new up-to-date book devoted exclusively to this topic, covering all aspects of linear complementarity would be worthwhile. This book is the result of the encouragement I have received from all these suggestions.

An important class of applications for the LCP stems from the fact that the necessary optimality conditions for a Quadratic Programming Problem (QP) lead to an LCP. Until recently, a practitioner of mathematical programming could have brushed off QP as an academically interesting generalization of linear programming which is not very useful. But the recent development of recursive quadratic programming methods for solving Nonlinear Programming Problems (NLP) has changed all that. These methods solve an NLP through a sequence of quadratic approximations, and have become extremely popular. They have suddenly made QP and thereby LCP an important topic in mathematical programming with a large number of practical applications. Because of this, the study of LCP is attracting a great deal of attention both in academic curricula and in the training of practitioners.

THE OBJECTIVES

1. To provide an in-depth and clear treatment of all the important practical, technical, computational, geometric, and mathematical aspects of the LCP, QP, and their various applications.
2. To discuss clearly the various algorithms for solving the LCP, to present their efficient implementation for the Computer, and to discuss their computational complexity.
3. To present the practical applications of these algorithms and extensions of these algorithms to solve general nonlinear programming problems.
4. To survey new methods for solving linear programs, proposed subsequently to the publication of [2.26].

BACKGROUND NEEDED

The background required to study this book is some familiarity with matrix algebra and linear programming (LP). The basics of LP are reviewed in Chapters 1 and 2.

SUMMARY OF CHAPTER CONTENTS

The book begins with a section titled 'notation' in which all the symbols and several terms are defined. It is strongly recommended that the reader peruse this section first at initial reading, and refer to it whenever there is a question about the meaning of some symbol or term.

Chapter 1 presents a clear geometric interpretation of the LCP through the definition of the system of complementary cones as a generalization of the set of orthants in $\mathbf{R}^n$. Applications to LP, QP, and nonzero sum game problems are discussed. There is a complete discussion of positive definiteness and positive semidefiniteness of square matrices, their relationship to convexity, together with efficient pivotal methods for checking whether these properties hold for a given matrix. Various applications of QP are discussed, as well as the recursive quadratic programming method for solving NLP models.

Chapter 2 presents a complete discussion of the many variants of the complementary pivot method and proofs of its convergence on different classes of LCPs. Section 2.7 contains a very complete, lucid, but elementary treatment of the extensions of the complementary pivot method to simplicial methods for computing fixed points using triangulations of $\mathbf{R}^n$, and various applications of these methods to solve a variety of general NLP models and nonlinear complementarity problems.

Chapter 3 covers most of the theoretical properties of the LCP. There is extensive treatment of the various separation properties in the class of complementary cones, and a complete discussion of principal pivot transforms of matrices. In this chapter we also discuss the various classes of matrices that arise in the study of the LCP. Chapter 4 provides a survey of various principal pivoting methods for solving the LCP. Algorithms for parametric LCP are presented in Chapter 5.

Chapter 6 contains results on the worst case computational complexity of the complementary and the principal pivoting methods for the LCP. Chapter 7 presents a special algorithm for the LCP associated with positive definite symmetric matrices, based on orthogonal projections, which turned out to be very efficient in computational tests. Chapter 8 presents the polynomially bounded ellipsoid methods for solving LCPs associated with positive semidefinite matrices, or equivalently convex QPs.

Chapter 9 presents various iterative methods for LCPs. In Chapter 10 we present an extensive survey of various descent methods for unconstrained and linearly constrained minimization problems; these techniques provide alternative methods for solving quadratic programming problems. In Chapter 11 we discuss some of the newer algorithms proposed for solving linear programming problems and their possible extensions to solve LCPs, and we discuss several unsolved research problems in linear complementarity.

To make the book self-contained, in the appendix we provide a complete treatment of theorems of alternatives for linear systems, properties of convex functions and convex sets, and various optimality conditions for nonlinear programming problems.

EXERCISES

Each chapter contains a wealth of various types of exercises. References are provided for theoretical exercises constructed from published literature. A new sequence of exercise numbers begins with each chapter (e.g. Exercise 3.2 refers to Exercise number 2 of Chapter 3).

HOW TO USE THE BOOK IN A COURSE

This book is ideally suited for first year graduate level courses in Mathematical Programming. For teaching a course in nonlinear programming, the best order for presenting the material may be the following: Section 10.1 (formulation example), 10.2 (types of solutions in NLP), 10.3 (types of nonlinear programs and what can and cannot be done efficiently by existing methods), 10.4 (can we at least compute a local minimum efficiently), 10.5 (precision in computation), 10.6 (rates of convergence), Appendix (theorems of alternatives for linear systems of constraints; convex sets and separating hyperplane theorems; convex, concave functions and their properties; optimality conditions), Chapters 1 to 9 in serial order; remaining portions of Chapter 10; and some supplemental material on algorithms for solving nonlinearly constrained problems like the GRG, penalty and barrier methods, and augmented Lagrangian methods. For teaching a course in linear complementarity using the book, it is best to cover the Appendix first, and then go through Chapters 1 to 10 in serial order.

The material contained in Chapters 12, 14, 15, 16 of [2.26] can be combined with that in Appendices 1, 2, Chapter 9 and Section 11.4 of this book to teach an advanced course in linear programming.

Since the book is so complete and comprehensive, it should prove very useful for researchers in LCP, and practitioners using LCP and nonlinear programming in applied work.

ACKNOWLEDGEMENTS

In preparing this book, I have received encouragement, advice, and suggestions from several people. For this I am indebted to Jeff Alden, Ahmet Bolat, R. Chandrasekaran, Akli Gana, Greg Goulding, Philip Jones, Ikuyo Kaneko, Olvi Mangasarian, Chuck Noon, Steve Pollock, Romesh Saigal, Richard Stone, Robert Smith, Vasant Ubhaya, Gary Waissi, Kai Yang and Bill Ziemba; and to David Gale who first introduced me to linear complementarity. Some of the research appearing in this book was supported partially by National Science Foundation under grants ECS-8401081 and ECS-8521183, this support is gratefully acknowledged. I thank Ivajean Marzano and Tana Beard for typing most of the manuscript. My thanks to Bala S. Guthy for drawing many of the figures with curves in them; and to K. T. Anantha for some of the other figures. Finally, I thank my wife Vijaya Katta, to whom this book is dedicated, for her patience and understanding.

CONTENTS

NOTATION

Superscript T	Denotes transposition. A^T is the transpose of the matrix A. If x is a column vector, x^T is the same vector written as a row vector and vice versa. Column vectors are printed as transposes of row vectors to conserve space in the text.
w, z	$w = (w_1, \ldots, w_n)^T$, $z = (z_1, \ldots, z_n)^T$ are the column vectors of variables in a linear complementarity problem of order n.
(q, M)	A linear complementarity problem in wich the data is the column vector $q = (q_1, \ldots, q_n)^T$, and square matrix $M = (m_{ij})$ of order n.
$\mathbf{R}^n$	Real Euclidean n-dimensional vector space. It is the set of all **ordered** vectors $(x_1, \ldots, x_n)$, where each x_j is a real number, with the usual operations of addition and scalar multiplication defined on it.
$\simeq$	Approximately equal to.
$\lambda \to 0$	λ tends to zero.
$\lambda \to 0^+$	λ tends to zero through positive values.
J, **K**, **H**, **E**, **Z**, **U**, **P**, **A** **Γ**, **I**, **Δ**, **S**, **W**, **D**	These bold face letters usually denote sets that are defined in that section or chapter.
$\sum$	Summation sign.
$\sum(a_j : j \in \mathbf{J})$	Sum of terms a_j over j contained in the set **J**.
$\geqq$, $\geq$, $>$	Given two vectors $x = (x_j)$, $y = (y_j)$ in $\mathbf{R}^n$, $x \geqq y$ means that $x_j \geqq y_j$, that is, $x_j - y_j$ is nonnegative, for all j. $x \geq y$ means that $x \geqq y$ but $x \neq y$, that is, $x_j - y_j$ is nonnegative for all j and strictly positive for at least one j. $x > y$ means that $x_j - y_j > 0$, strictly positive, for all j. The vector x is said to be nonnegative if $x \geqq 0$, semipositive if $x \geq 0$, and positive if $x > 0$.
$A_{i.}$	The ith row vector of the matrix A.

$A_{.j}$ The jth column vector of the matrix A.

Superscripts We use superscripts to enumerate vectors or matrices or elements in any set. When considering a set of vectors, in $\mathbf{R}^n$, x^r may used to denote the rth vector in the set, and it will be the vector $(x_1^r, \ldots, x_n^r)^T$. In a similar manner, while considering a sequence of matrices, the symbol P^r may be used to denote the rth matrix in the sequence. Superscripts should not be confused with exponents and these are distinguished by different type styles.

Exponents In the symbol $\epsilon^{\mathbf{r}}$, r is the exponent. $\epsilon^{\mathbf{r}} = \epsilon \times \epsilon \times \ldots \times \epsilon$, where there are r ϵ's in this product. Notice the difference in type style between superscripts and exponents.

$\log_2 x$ Defined only for positive numbers x. It is the logarithm of the positive real number x, with 2 as the base (or radix).

$||x||$ Euclidean norm of a vector $x \in \mathbf{R}^n$. If $x = (x_1, \ldots, x_n)$, $||x|| = +\sqrt{x_1^{\mathbf{2}} + \ldots + x_n^{\mathbf{2}}}$.

$\lceil \alpha \rceil$ Defined only for real numbers α. It represents the smallest integer that is greater than or equal to α, and is often called the *ceiling* of α. For example $\lceil -4.3 \rceil = -4$, $\lceil 4.3 \rceil = 5$.

$\lfloor \alpha \rfloor$ Defined only for real numbers α. It represents the larg-
est integer less than or equal to α, and is often called the *floor* of α. For example $\lfloor -4.3 \rfloor = -5$, $\lfloor 4.3 \rfloor = 4$.

∞ Infinity.

$\in$ Set inclusion symbol. If $\mathbf{F}$ is a set, "$F_1 \in \mathbf{F}$" means that "F_1 is an element of $\mathbf{F}$". Also "$F_2 \notin \mathbf{F}$" means that "F_2 is not an element of $\mathbf{F}$".

$\subset$ Subset symbol. If $\mathbf{E}$, $\mathbf{\Gamma}$ are two sets, "$\mathbf{E} \subset \mathbf{\Gamma}$" means that "$\mathbf{E}$ is a subset of $\mathbf{\Gamma}$", or that "every element in $\mathbf{E}$ is also an element of $\mathbf{\Gamma}$".

$\cup$ Set union symbol. If **D**, **H** are two sets, $\mathbf{D} \cup \mathbf{H}$ is the set of all elements that are either in **D** or in **H** or in both **D** and **H**.

$\cap$ Set intersection symbol. If **D** and **H** are two sets, $\mathbf{D} \cap \mathbf{H}$ is the set of all elements that are in both **D** and **H**.

$\emptyset$ The empty set. The set containing no elements.

$\setminus$ Set difference symbol. If **D** and **H** are two sets, $\mathbf{D} \setminus \mathbf{H}$ is the set of all elements of **D** that are not in **H**.

$\{ \quad \}$ Set brackets. The notation $\{x$: some property$\}$ represents the set of all elements x, satisfying the property mentioned after the ":".

$|\mathbf{F}|$ If **F** is a set, this symbol denotes its **cardinality**, that is, the number of distinct elements in the set **F**.

e The base of the natural logarithms. $\text{e} = 1 + \sum_{n=1}^{\infty} \frac{1}{n!}$, if is approximately equal to 2.7.

e, e_r The symbol e denotes a column vector, all of whose entries are equal to 1. Its dimension is usually understood from the context. When we want to specify the dimension, e_r denotes the column vector in $\mathbf{R}^r$, all of whose entries are equal to 1.

I, I_r The symbol I denotes the unit matrix, its order understood from the context. When we want to specify the order, I_r denotes the unit matrix of order r.

$|\alpha|$ Absolut value of the real number α.

$\square$ This symbol indicates the end of a proof.

y^+ If $y = (y_j) \in \mathbf{R}^n$, let $y_j^+ = \text{Maximum } \{0, y_j\}$, $j = 1$ to n. Then $y^+ = (y_j^+)$.

$\succ$ Lexicographically greater than. Given two vectors $x = (x_j)$, $y = (y_j)$ in $\mathbf{R}^n$, $x \succ y$ means that for the smallest j for which $x_j - y_j \neq 0$, we have $x_j - y_j > 0$.

Pos$\{A_1, \ldots, A_k\}$ If $A_1, \ldots, A_k$ are vectors in $\mathbf{R}^n$ then Pos$\{A_1, \ldots, A_k\} = \{y : y = \alpha_1 A_1 + \ldots + \alpha_k A_k, \alpha_1 \geqq 0, \ldots, \alpha_k \geqq 0\}$. It is the cone in $\mathbf{R}^n$ which is the nonnegative hull of the set of vectors $\{A_1, \ldots, A_k\}$.

Pos(A) If A is a matrix, Pos$(A) = \{x : x = Ay$ for some $y \geqq 0\}$. It is the cone which is the nonnegative hull of the column vectors of the matrix A.

$n!$ n factorial. Defined only for nonnegative integers. $0! = 1$. And $n!$ is the product of all the positive integers from 1 to n, whenever n is a positive integer.

$\binom{n}{r}$ Defined only for positive integers $n \geqq r$. It is the number of distinct subsets of r objects from a set of n distinct objects. It is equal to $\frac{n!}{r!(n-r)!}$.

$\langle v_1, \ldots, v_r \rangle$ When $v_1, \ldots, v_r$ are all column vectors from the space $\mathbf{R}^n$, say, and satisfy the property that the set of column vectors $\left\{ \begin{pmatrix} 1 \\ v_1 \end{pmatrix}, \ldots, \begin{pmatrix} 1 \\ v_r \end{pmatrix} \right\}$ is linearly independent, then $v_1, \ldots, v_r$ are the vertices of an $(r-1)$-dimensional simplex, which is their convex hull, this simplex is denoted by the symbol $\langle v_1, \ldots, v_r \rangle$. See Section 2.7.8.

$\mathcal{C}(M)$ The class of $2^{\mathbf{n}}$ complementary cones associated with the square matrix M of order n.

K(M) The union of all complementary cones in $\mathcal{C}(M)$. It is the set of all vectors q for which the LCP (q, M) has at least one solution.

Z(y), **W**(y) If $y = (y_1, \ldots, y_n)^T$ is a complementary vector for the LCP (q, M) of order n, then **Z**$(y) = \{j : y_j = z_j\}$ and **W**$(y) = \{j : y_j = w_j\}$. See Section 3.1.

Miminum{ } The minimum number among the set of numbers appearing inside the set brackets. Maximum{ } has a similar meaning. If the set is empty we will adopt the convention that the minimum in it is $+\infty$ and the maximum in it is $-\infty$.

Infimum, minimum; Supremum, maximum Let $\boldsymbol{\Gamma}$ be a subset of $\mathbf{R}^n$ and let $f(x)$ be a real valued function defined on $\boldsymbol{\Gamma}$. The **infimum** for $f(x)$ on $\boldsymbol{\Gamma}$ is defined to be the largest number α satisfying: $f(x) \geqq \alpha$ for all $x \in \boldsymbol{\Gamma}$. If α_0 is the infimum for $f(x)$ on $\boldsymbol{\Gamma}$, and there exists an $\bar{x} \in \boldsymbol{\Gamma}$ satisfying $f(\bar{x}) = \alpha_0$, then α_0 is said to be the **minimum value** of $f(x)$ on $\boldsymbol{\Gamma}$ and $\bar{x}$ is the point which attains it. As an example let $\boldsymbol{\Gamma} \subset \mathbf{R}^1$ be the open interval $0 < x < 1$, and let $f(x) = x$. The infimum of $f(x)$ on $\boldsymbol{\Gamma}$ in this example is 0, it is not a minimum since $0 \notin \boldsymbol{\Gamma}$, and there exists no point x in $\boldsymbol{\Gamma}$ where $f(x) = 0$. As another example let $\boldsymbol{\Gamma} \subset \mathbf{R}^1$ be the unbounded set $1 \leqq x < \infty$ and let $f(x) = \frac{1}{x}$. In this example, the infimum of $f(x)$ on $\boldsymbol{\Gamma}$ is 0, and again this is not a minimum. In the same manner, the **supremum** in $\boldsymbol{\Gamma}$ of a real valued function $f(x)$ defined on $\boldsymbol{\Gamma} \subset \mathbf{R}^n$, is the smallest number γ satisfying: $f(x) \leqq \gamma$ for all $x \in \boldsymbol{\Gamma}$. If γ_0 is the supremum of $f(x)$ on $\boldsymbol{\Gamma}$, and there exists an $\hat{x} \in \boldsymbol{\Gamma}$ satisfying $f(\hat{x}) = \gamma_0$, then γ_0 is said to be the **maximum value** of $f(x)$ on $\boldsymbol{\Gamma}$, and $\hat{x}$ is the point which attains it.

Local minimum, global minimum Consider an optimization problem in which an objective function $\theta(x)$, which is a real valued function defined over $\mathbf{R}^n$, is required to be minimized, subject to possibly some constraints on the decision variables x. Let $\mathbf{K} \subset \mathbf{R}^n$ denote the set of feasible solutions for this problem. A point $\hat{x} \in \mathbf{K}$ is said to be a global minimum for this problem if there exists no $x \in \mathbf{K}$ satisfying $\theta(x) < \theta(\hat{x})$. A point $\bar{x} \in \mathbf{K}$ is said to be a local minimum for this problem if there exists an $\varepsilon > 0$ such that the following system has no feasible solution

$$\begin{aligned} &x \in \mathbf{K} \\ &\theta(x) < \theta(\bar{x}) \\ &||x - \bar{x}|| < \varepsilon \end{aligned}$$

that is, $\bar{x}$ is a local minimum for this problem iff $\bar{x}$ is a global minimum for $\theta(x)$ over $\mathbf{K} \cap \{x : ||x - \bar{x}|| < \varepsilon\}$. See Section 10.2.

Cardinality Defined only for sets. The cardinality of a set is the number of distinct elements in it.

Principal Submatrix $F_{\mathbf{JJ}}$ of square matrix F Let $F = (f_{ij})$ be a given square matrix of order n. Let $\mathbf{J} \subset \{1, \ldots, n\}$. The **principal submatrix** of F determined by the subset $\mathbf{J}$ is the matrix $F_{\mathbf{JJ}} = (f_{ij} : i \in \mathbf{J}, j \in \mathbf{J})$. See Section 1.3.1. The determinant of $F_{\mathbf{JJ}}$ is known as the principal subdeterminant of F corresponding to the subset $\mathbf{J}$.

BFGS updating formula The Broyden-Fletcher-Goldfarb-Shanno formula for updating a positive definite symmetric approximation to the Hessian (or its inverse) of a twice continuously real valued function $\theta(x)$ defined on $\mathbf{R}^n$, as the algorithm moves from one point to next. See Sections 1.3.6 and 10.8.6.

LCP Linear complementarity problem.

NLCP Nonlinear complementarity problem.

LP Linear program.

BFS Basic feasible solution.

NLP Nonlinear program.

PD Positive definite. A square matrix M of order n is said to be PD if $y^T My > 0$ for all $y \in \mathbf{R}^n$, $y \neq 0$.

PSD Positive semidefinite. A square matrix M of order n is said to be PSD if $y^T My \geqq 0$ for all $y \in \mathbf{R}^n$.

ND Negative definite. A square matrix of order n is said to be ND if $y^T My < 0$ for all $y \in \mathbf{R}^n$, $y \neq 0$.

NSD Negative semidefinite. A square matrix of order n is said to be NSD if $y^T My \leqq 0$ for all $y \in \mathbf{R}^n$.

PPT Principal pivot transform. See Section 3.2.

$(i.j)$ This refer to the jth equation in the ith chapter. Equations are numbered serially in each chapter.

Section $i.j$**;** $i.j.k$ The sections are numbered serially in each chapter. "$i.j$" refers to section j in Chapter i. "$i.j.k$" refers to subsection k in section $i.j$.

Figure $i.j$ The jth figure in Chapter i. The figures are numbered serially in this manner in each chapter.

Reference $[i.j]$ The jth reference in the list of references given at the end of the Chapter i. References given at the end of each chapter are numbered serially.

Exercise $i.j$ The jth exercise in Chapter i. Exercises are numbered serially in each chapter.

Figure i**, Exercise** i**, Theorem** i**, Reference** i**, Example** i In the appendices, figures, examples, exercises, theorems, references, etc. are numbered serially using a single number for each. So any figure, example, exercise, theorem or reference with a single number like this must be in the appendix.

Linear Function, affine function

The real valued function $f(x)$ defined over $x \in \mathbf{R}^n$ is called a **linear function** if $f(x) = c_1x_1 + \ldots + c_nx_n$ where $c_1, \ldots, c_n$ are constants, it satisfies the property: $f(\alpha x^1 + \beta x^2) = \alpha f(x^1) + \beta f(x^2)$ for all $x^1, x^2 \in \mathbf{R}^n$ and for all real numbers α, β. The real valued function $g(x)$ defined over $x \in \mathbf{R}^n$ is said to be an **affine function** if $g(x) = \gamma_0 + \gamma_1x_1 + \ldots + \gamma_nx_n$ where $\gamma_0, \gamma_1, \ldots, \gamma_n$ are constants, it satisfies the property: $g(\alpha x^1 + \beta x^2) = \alpha g(x^1) + \beta g(x^2)$ for all $x^1, x^2 \in \mathbf{R}^n$ and for all real numbers α, β satisfying $\alpha + \beta = 1$. Every affine function defined over $\mathbf{R}^n$ in a linear function plus a constant.

Basis, basic vector, basic solution, basic feasible solution

See Section 2.1.

Bounded set

A subset $\mathbf{S} \subset \mathbf{R}^n$ is bounded if there exists a finite real number α such that $||x|| \leqq \alpha$, for all $x \in \mathbf{S}$.

Proper subset

If $\mathbf{E}$ is a subset of a set $\mathbf{\Gamma}$, $\mathbf{E}$ is said to be a proper subset of $\mathbf{\Gamma}$ if $\mathbf{E} \neq \mathbf{\Gamma}$, that is, if $\mathbf{\Gamma} \setminus \mathbf{E} \neq \emptyset$.

Feasible solution

A numerical vector that satisfies all the constraints and restrictions in the problem.

Optimum solution or Optimum feasible solution

A feasible solution that optimizes (i. e., either maximizes or minimizes as required) the objective value among all feasible solutions.

Algorithm

The word from the last name of the Persian scholar Abu Ja'far Mohammed ibn Mûsâ alkhowârizmî whose textbook on arithmetic (about A.D. 825) had a significant influence on the development of these methods. An **algorithm** is a set of rules for getting a required output from a specific input, in which each step is so precisely defined that it can be translated into computer language and executed by machine.

Size The size of an optimization problem is a parameter that measures how large the problem is. Usually it is the number of digits in the data in the optimization problem, when it is encoded in binary form.

$\mathcal{O}(n^{\mathbf{r}})$ A finitely terminating algorithm for solving an optimization problem is said to be of order $n^{\mathbf{r}}$ or $\mathcal{O}(n^{\mathbf{r}})$, if the computational effort required by the algorithm in the worst case, to solve a version of the problem of size n, grows as $\alpha n^{\mathbf{r}}$, where α, r are numbers that are independent of the size n and the data in the problem.

Polynomially bounded algorithm An algorithm is said to be polynomially bounded if it can be proved that the computational efffort required by it is bounded above by a fixed polynomially in the size of the problem.

The class $\mathcal{P}$ of problems This is the class of all problems for solving which there exists a polynomially bounded algorithm.

$\mathcal{NP}$-complete class of problems

A decision problem is one for which the answer is "yes" or "no". For example, given an integer square matrix D of $\mathbf{R}^n$, the problem "is there an $x \in \mathbf{R}^n$ satisfying $x^T Dx < 0$?" is a decision problem. Also, given a square matrix M of order n and a column vector $q \in \mathbf{R}^n$, the problem "does the LCP (q, M) have a solution?" is a decision problem. Often, optimization problems can be handled by studying decision problem versions of them. For example, consider the problem of minimizing $\theta(x)$ over $x \in \mathbf{K}$, where $\mathbf{K}$ represents the set of feasible solutions of this problem. The decision problem version of this optimization problem is "is there an $x \in \mathbf{K}$ satisfying $\theta(x) \leqq \alpha$?" where α is a specified real number. Clearly, by examining this decision problem with varying values of α, we can narrow down the solution of the optimization problem.

The $\mathcal{NP}$-complete class is a class of decision problems in discrete optimization, satisfying the property that if a polynomially bound algorithm exists for any one problem in the classs, then polynomially bounded algorithms exist for every problem in the class. So far no polynomially bounded algorithm is known for any problem in the $\mathcal{NP}$-complete class, and it is believed that all these problems are hard problems (in the worst case, the computational effort required for solving an instance of any problem in the class by any known algorithm, grows asymptotically, faster than any polynomial in the size of the problem). See reference [8.12] for a complete discussion of $\mathcal{NP}$-completeness.

Necessary conditions, sufficient conditions, necessary and sufficient conditions

When studying a property of a system, a condition is said to be a **necessary condition** for that property if that condition is satisfied whenever the property holds. A condition is said to be a **sufficient condition** for the property if the property holds whenever the condition is satisfied. A **necessary and sufficient condition** for the property is a condition that is both necessary condition and a sufficient condition for that property.

Active or tight constraint An inequality constraint $g_p(x) \geqq 0$ is said to be **active** or **tight**, at a point $\bar{x}$ satisfying it, if $g_p(\bar{x}) = 0$. The equality constraint $h_i(x) = 0$ is always an active constraint at any point $\bar{x}$ satifying it.

Infeasible system A system of constraints in the variables $x = (x_j)$ is said to be infeasible, if there exists no vector x satisfying all the constraints.

Complementary pair A pair of variables in an LCP, at least one of which is required to be zero. Each variable in a complementary pair is said to be the complement of the other. A pair of column vectors corresponding to a complementary pair of variables in an LCP is a complementary pair of column vectors. Each column vector in a complementary pair is the complement of the other. In an LCP of order n, there are n complementary pairs, numbered 1 to n.

Complementary set of vectors A vector of n variables in an LCP of order n is a complementary vector if the jth variable in the vector is from the jth complementary pair of variables, for each j. A complementary set of column vectors is an ordered set in which the jth vector is from the jth complementary pair for each j.

Complementary matrix In an LCP of order n, this is a square matrix of order n whose jth column vector is from the jth complementary pair, for each j.

Complementary cone In an LCP of order n, this is $\text{Pos}(A)$ where A is a complementary matrix of this problem.

Complemetary basis It is a complementary matrix which is nonsingular.

Complementary basic vector It is a complementary vector of variables associated with a complementary basis.

Complementary feasible basis It is a complementary basis which is a feasible basis for the problem.

Complementary feasible basic vector — It is a complementary basic vector which is feasible to the problem.

$\bar{z}$ leads to a solution of the LCP (q, M) — We say that the vector $\bar{z}$ leads to a solution of the LCP (q, M) if $(\overline{w} = M\bar{z} + q, \bar{z})$ is a solution of the LCP (q, M).

To process an LCP — When an algorithm for solving LCPs is applied on an LCP, it may either obtain a solution of the LCP, or terminate without obtaining a solution. It is possible that some algorithms may terminate without a solution even though the LCP may have a solution. An algorithm for solving LCPs is said **to process a specified class of LCPs** if, when the algorithm is applied on any LCP from this class and it terminates without obtaining a solution, we can prove that the LCP in fact has no solution. In other words, an algorithm is said to process a class of LCPs iff for every LCP in this class, the algorithm either produces a solution or conclusively establishes that the LCP cannot have a solution.

Secondary ray or terminal ray — This is the half-line or ray obtained at the end of executing the complementary pivot algorithm on an LCP, if the algorithm terminates in ray termination. This secondary ray, if it is obtained, is distinct from the initial ray with which the algorithm is initiated. See Section 2.2.6.

Subcomplementary set, vector — It is a complementary set or vector with one element missing.

Almost complementary vector — It is a vector that is complementary except for one violation which is set up appropriately. See Sections 2.2.4, 2.4.

Copositive matrix — A square matrix M of order n is said to be copositive if $y^T My \geqq 0$ for all $y \geqq 0$ in $\mathbf{R}^n$.

Strictly copositive matrix — A square matrix M of order n is said to be strictly copositive if $y^T My > 0$ for all $y \geq 0$ in $\mathbf{R}^n$.

Copositive plus matrix A square matrix M of order n is said to be copositive plus if it is copositive, and for $y \geqq 0$ in $\mathbf{R}^n$ if $y^T My = 0$ then $(M + M^T)y = 0$.

P_0-matrix A square matrix is a P_0-matrix if all its principal subdeterminants are $\geqq 0$.

P-matrix A square matrix is said to be a P-matrix if all its principal subdeterminants are strictly positive.

Q-matrix A square matrix M of order n is said to be a Q-matrix if the LCP (q, M) has a solution for all $q \in \mathbf{R}^n$.

Z-matrix A square matrix $M = (m_{ij})$ is a Z-matrix if $m_{ij} \leqq 0$ for all $i \neq j$.

Q_0-matrix The square matrix M is said to be a Q_0-matrix if $\mathbf{K}(M)$ is a convex cone.

$\bar{Q}$-matrix, or Completely Q-matrix A square matrix M such that M and all its principal submatrices are Q-matrices.

$\bar{Q}_0$-matrix, or Completely Q_0-matrix A square matrix M such that M and all its principal submatrices are Q_0-matrices.

Faces, Facets

Let $\mathbf{K} \subset \mathbf{R}^n$ be a convex polyhedron. $\mathbf{H} = \{x : ax = a_0\}$ where $a \neq 0$ is a given row vector in $\mathbf{R}^n$. $\mathbf{H}$ is a hyperplane in $\mathbf{R}^n$. $\mathbf{H}$ is said to have $\mathbf{K}$ on one of its sides if either $ax \geqq a_0$ for all $x \in \mathbf{K}$, or $ax \leqq a_0$ for all $x \in \mathbf{K}$. If $\mathbf{H}$ has $\mathbf{K}$ on one of its sides and $\mathbf{H} \cap \mathbf{K} \neq \emptyset$, $\mathbf{H}$ is said to be a **supporting hyperplane** for $\mathbf{K}$. A **face** of $\mathbf{K}$ is either the empty set $\emptyset$, or the set $\mathbf{K}$ itself, or $\mathbf{H} \cap \mathbf{K}$ for some supporting hyperplane $\mathbf{H}$ for $\mathbf{K}$. See reference [2.26]. For example, extreme points of $\mathbf{K}$ are its faces of dimension zero. Edges of $\mathbf{K}$ are its faces of dimension 1, etc.

A face of $\mathbf{K}$ is said to be a **facet** if its dimension is one less than the dimension of $\mathbf{K}$.

For some special convex polyhedra, simplicial cones or simplexes, it is possible to characterize all faces easily. If $\{B_{.1}, \ldots, B_{.n}\}$ is a linearly independent set of column vectors in $\mathbf{R}^n$, then, for the simplicial cone $\text{Pos}\{B_{.1}, \ldots, B_{.n}\}$, the cone $\text{Pos}\{B_{.1}, \ldots, B_{.j-1}, B_{.j+1}, \ldots, B_{.n}\}$ is a facet for any j, and the cone $\text{Pos}\{B_{.j} : j \in \mathbf{J}\}$ is a face for any subset $\mathbf{J} \subset \{1, \ldots, n\}$ (this face is defined to be $\{0\}$, if $\mathbf{J} = \emptyset$). If $\{v_0, \ldots, v_n\}$ are the set of vertices of an n-dimensonal simplex in $\mathbf{R}^n$, the convex hull of $\{v_0, \ldots, v_{j-1}, v_{j+1}, \ldots, v_n\}$ is a facet of this simplex for all j, and the convex hull of $\{v_j : j \in \mathbf{J}\}$ is a face of this simplex for all subsets $\mathbf{J} \subset \{1, \ldots, n\}$ (this face is defined to be the empty set if $\mathbf{J} = \emptyset$).

Principally degenerate, principally nondegenerate, matrices

A square matrix A is said to be **principally nondegenerate** if all its principal subdeterminantes are nonzero; **principally degenerate** if at least one of its principal subdeterminantes has value zero. In this book we are usually concerned only with principal degeneracy or nondegeneracy of square matrices, and hence we usually omit the adjective "principally" and refer to the matrices as being degenerate or nondegenerate.

Degenerate or nondegenerate complementary cone

A complementary cone is nondegenerate if its interior is nonempty, degenerate otherwise.

Strongly degenerate or weakly degenerate complementary cone

A degenerate complementary cone $\text{Pos}(A_{.1}, \ldots, A_{.n})$ is said to be **strongly degenerate** if there exists $(\alpha_1, \ldots, \alpha_n) \geq 0$ such that $0 = \alpha_1 A_{.1} + \ldots + \alpha_n A_{.n}$, that is, if the zero vector can be expressed as a semipositive linear combination of the complementary set of column vectors $\{A_{.1}, \ldots, A_{.n}\}$; **weakly degenerate** otherwise.

Degenerate or nondegenerate basic solutions, vectors, systems of linear equations

Consider the system of linear constraints "$Ax = b$" where A is a matrix of order $m \times n$ and rank m. A basic solution $\bar{x}$ for this system is said to be **nondegenerate** if the number of nonzero variables in $\bar{x}$ is m, **degenerate** if this number is $< m$. The right hand side constants vector b in the system is said to be degenerate if the system has at least one degenerate basic solution, b is said to be nondegenerate if the system has no degenerate basic solution. Thus b is degenerate in the system if it can expressed as a linar combination of $m - 1$ or less column vectors of A, nondegenerate otherwise. The system of constraints is itself said to be degenerate or nondegenerate depending on whether b is degenerate or nondegenerate.

Lipschitz continuous

Let $f(x)$ be a continuous real valued function defined on $\mathbf{K} \subset \mathbf{R}^n$. It is said to be Lipschitz continuous (or Lipschitzian) on $\mathbf{K}$ if there exists a nonnegative number α such that $|f(x) - f(y)| \leqq \alpha ||x - y||$ for all $x, y \in \mathbf{K}$. The number α is known as the Lipschitz constant for this function.

Principal subproblem

Consider the LCP (q, M) with variables $(w_1, \ldots, w_n)^T$, $(z_1, \ldots, z_n)^T$. Let $\mathbf{J} \subset \{1, \ldots, n\}$, $\mathbf{J} \neq \emptyset$. Let $q_{\mathbf{J}} = (q_i : i \in \mathbf{J})^T$, $M_{\mathbf{JJ}} = (m_{ij} : i \in \mathbf{J}, j \in \mathbf{J})$. The LCP $(q_{\mathbf{J}}, M_{\mathbf{JJ}})$ in variables $w_{\mathbf{J}}$, $z_{\mathbf{J}}$ is called the **principal subproblem** of the LCP (q, M) corresponding to the subset $\mathbf{J}$.

Simplex

See Section 2.7.

$\nabla\theta(\bar{x})$

The row vector of partial derivatives $\left(\frac{\partial\theta(x)}{\partial x_1}, \ldots, \frac{\partial\theta(x)}{\partial x_n}\right)$, **gradient vector**, evaluated at $x = \bar{x}$.

$\partial f(x)$ — The subdifferential set of the function $f(x)$ at the point x. See Appendix 3 and Section 2.7.1.

Differentiable function — A real valued function $\theta(x)$ defined on an open subset $\mathbf{\Gamma} \subset \mathbf{R}^n$ is said to be **differentiable** at a point $\bar{x} \in \mathbf{\Gamma}$, if all the partial derivatives $\frac{\partial \theta(\bar{x})}{\partial x_j}$, $j = 1$ to n exist, and for any $y \in \mathbf{R}^n$, $[\theta(\bar{x} + \alpha y) - \theta(\bar{x}) - \alpha \nabla \theta(\bar{x}) y]/\alpha$ tends to zero as α tends to zero. If it is differentiable at every point $\bar{x} \in \mathbf{\Gamma}$, it is said to be differentiable in $\mathbf{\Gamma}$.

Continuously differentiable function — A real-valued function $\theta(x)$ defined on an open subset $\mathbf{\Gamma} \in \mathbf{R}^n$ is said to be **continuously differentiable** at a point $\bar{x} \in \mathbf{\Gamma}$ if it is differentiable at $\mathbf{\Gamma}$ and $\nabla \theta(x)$ is contiuous at $\bar{x}$. If it is continuously differentiable at every point $\bar{x} \in \mathbf{\Gamma}$, it is said to be continuoulsy differentiable in $\mathbf{\Gamma}$.

$H(\theta(\bar{x}))$ — The **Hessian matrix** of $\theta(x)$ at $\bar{x}$. It is the square matrix of second partial derivatives $\left(\frac{\partial^2 \theta(\bar{x})}{\partial x_i \partial x_j}\right)$ evaluated at $\bar{x}$.

Twice differentiable function — A real valued function $\theta(x)$ defined over an open set $\mathbf{\Gamma} \in \mathbf{R}^n$ is said to be **twice differentiable** at $\bar{x} \in \mathbf{\Gamma}$ if $\nabla \theta(\bar{x})$ and $H(\theta(\bar{x}))$ exist, and for all $y \in \mathbf{R}^n$, $[\theta(\bar{x}+\alpha y) - \theta(\bar{x}) - \alpha(\nabla \theta(\bar{x}))y - \frac{\alpha^2}{2} y^T H(\theta(\bar{x}))y]/\alpha^2$ tends to zero as α tends to zero. $\theta(x)$ is said to be twice differentiable in $\mathbf{\Gamma}$ if it is twice differentiable at every point in $\mathbf{\Gamma}$.

Twice continuously differentiable function — A real valued function $\theta(x)$ defined over an open set $\mathbf{\Gamma} \in \mathbf{R}^n$ is said to be **twice continuously differentiable** at $\bar{x} \in \mathbf{\Gamma}$ if it is twice differentiable at $\bar{x}$ and $H(\theta(x))$ is continuous at $\bar{x}$. It is twice continuously differentiable in $\mathbf{\Gamma}$ if it is twice continuously differentiable at every point in $\mathbf{\Gamma}$.

Smooth function

Mathematically, a real valued function defined on $\mathbf{R}^n$ is said to be a smooth function if it has derivatives of all orders. Many of the algorithms discussed in this book use only derivatives of the first or at most second orders. So, for our purpose, we will consider a smooth function to be one which is continuously differentiable, or twice continuously differentiable if the method under consideration uses second order derivatives.

Optimization problems in minimization form

Whenever a function $f(x)$ has to be maximized subject to some conditions, we can look at the equivalent problem of minimizing $-f(x)$ subject to the same conditions. Both problems have the same set of optimum solutions and the maximum value of $f(x) = -$minimum value of $(-f(x))$. Because of this, we discuss only minimization problems.

$\nabla h(x)$ when $h(x) = (h_1(x), \ldots, h_m(x))^T$

Let $h(x)$ denote the column vector of m differentiable functions $h_i(x)$, $i = 1$ to m, defined over $\mathbf{R}^n$. Then $\nabla h(x) = \left(\frac{\partial h_i(x)}{\partial x_j} : i = 1 \text{ to } m,\ j = 1 \text{ to } n\right)$ is the **Jacobian matrix** in which the ith row vector is the gradient vector of $h_i(x)$ written as a row vector.

Nonlinear programming problem

This refers to an optimization problem of the following general form :

$$\begin{array}{lll} \text{minimize} & \theta(x) & \\ \text{subject to} & h_i(x) = 0, & i = 1 \text{ to } m \\ & g_p(x) \geqq 0, & p = 1 \text{ to } t \end{array}$$

where all the functions $\theta(x)$, $h_i(x)$, $g_p(x)$ are real valued continuous functions of $x = (x_1, \ldots, x_n)^T \in \mathbf{R}^n$. The problem is said to be a **smooth nonlinear program** if all the functions are in fact continuously differentiable functions. In this book we only consider smooth nonlinear programs. See Chapter 10.

Quadratic forms in matrix notations

Consider the quadratic form in n variables $x = (x_1, \ldots, x_n)^T$, $f(x) = \sum_{i=1}^{n} g_{ii}x_i^2 + \sum_{i=1}^{n}\sum_{j=i+1}^{n} g_{ij}x_ix_j$. An example for $n = 3$ is $h(x) = 81x_1^2 - 7x_2^2 + 5x_1x_2 - 6x_1x_3 + 18x_2x_3$. Let $F = (f_{ij})$ be a square matrix of order n satisfying

$$f_{ii} = g_{ii}, \quad i = 1 \text{ to } n$$
$$f_{ij} + f_{ji} = g_{ij}, \quad \text{for } i \neq j \text{ and } j > i.$$

Then it can be verified that $f(x) = x^T F x$. In particular, if we define the symmetric matrix $D = (d_{ij})$ of order n, where

$$d_{ii} = g_{ii}, \quad i = 1 \text{ to } n$$
$$d_{ij} = d_{ji} = \frac{1}{2}g_{ij}, \quad \text{for } i \neq j \text{ and } j > i$$

then $f(x) = x^T D x$. For the quadratic form $h(x)$ in 3 variables, $x = (x_1, x_2, x_3)^T$, given above, the matrix D turns out to be

$$D = \begin{pmatrix} 81 & \frac{5}{2} & -3 \\ \frac{5}{2} & -7 & 9 \\ -3 & 9 & 0 \end{pmatrix}$$

and $h(x) = x^T D x$.

Quadratic programming problem; convex or nonconvex quadratic programs

An optimization problem in which a quadratic function of $x = (x_1, \ldots, x_n)^T \in \mathbf{R}^n$ is to be optimized subject to linear constraints on the variables, is called a quadratic programming problem. Its general form is:

$$\begin{array}{ll} \text{minimize} & Q(x) = cx + \frac{1}{2}x^T Dx \\ \text{subject to} & Ax \geqq b \\ & Ex = d \end{array}$$

where D is a square symmetric matrix of order n. The inequality constraints here include any nonnegativity restrictions or the lower or upper bound restrictions on the variables.
This problem is called a **convex quadratic program** if D is a PSD matrix (in this case the objective function to be minimized, $Q(x)$, is convex); a **nonconvex quadratic program** otherwise.

QP Quadratic Programming Problem.

Complemetary basis It is a complementary matrix which is nonsingular.

$\nabla_x(f(\overline{x}, \overline{\mu}))$, $H_x(f(\overline{x}, \overline{\mu}))$ These are respectively the row vector of the partial deri-vates, and the square matrix of the second order partial derivates, of the function $f(x, \mu)$, with respect to the variables in the vector x, at $(\overline{x}, \overline{\mu})$.

Karush-Kuhn-Tucker (or KKT) necessary optimality conditions

Let $\theta(x)$, $h_i(x)$, $g_p(x)$, be real valued continuously differentiable functions defined on $\mathbf{R}^n$ for all i, p. Consider the following mathematical program:

$$\begin{array}{lll} \text{minimize} & \theta(x) & \\ \text{subject to} & h_i(x) = 0, & i = 1 \text{ to } m \\ & g_p(x) \geqq 0, & p = 1 \text{ to } t \end{array}$$

The Karush-Kuhn-Tucker (KKT) Lagrangian for this problem is: $L(x, \mu, \pi) = \theta(x) - \sum_{i=1}^{m} \mu_i h_i(x) - \sum_{p=1}^{t} \pi_p g_p(x)$ where μ_i, π_p are the Lagrange multipliers associated with the constraints. The Karush-Kuhn-Tucker (KKT) necessary optimality condition for this problem are :

$$\frac{\partial}{\partial x} L(x, \mu, \pi) = \nabla\theta(x) - \sum_{i=1}^{m} \mu_i \nabla h_i(x) - \\ - \sum_{p=1}^{t} \pi_p \nabla g_p(x) = 0$$

$$\begin{aligned} h_i(x) &= 0, \quad i = 1 \text{ to } m \\ g_p(x) &\geqq 0, \quad p = 1 \text{ to } t \\ \pi_p &\geqq 0, \quad p = 1 \text{ to } t \\ \pi_p g_p(x) &= 0, \quad p = 1 \text{ to } t \end{aligned}$$

where $\nabla\theta(x)$ etc. are the vectors of partial derivatives. If $\bar{x}$ is a local minimum for this problem, under fairly general conditions (see Appendix 4) it can be shown that there exist multiplier vectors $\bar{\mu}$, $\bar{\pi}$ such that $\bar{x}$, $\bar{\mu}$, $\bar{\pi}$ together satisfy these KKT conditions. In the literature these conditions are usually called **first-order necessary optimality conditions** or Kuhn-Tucker conditions. But it has been found recently that Karush was the first to discuss them. Hence, nowadays, the name Karush-Kuhn-Tucker necessary optimality conditions is coming into Vogue.

A feasible solution $\bar{x}$ satisfying the property that there exist Lagrange multiplier vectors $\bar{\pi}$, $\bar{\mu}$ such that $\bar{x}$, $\bar{\mu}$, $\bar{\pi}$ together satisfy the KKT conditions, is called a KKT point for the problem.

Stationary point for an NLP

Given an NLP, a stationary point for it usually refers to any feasible solution satisfying a necessary optimality condition for it. Every optimum solution is a stationary point, but, in general, there may be stationary points which are not even locally optimal to the problem.

Direction, half-line

Any point $y \in \mathbf{R}^n$, $y \neq 0$ defines a **direction** in $\mathbf{R}^n$. Given $\bar{x} \in \mathbf{R}^n$, points $\bar{x} + \alpha y$, $\alpha \geqq 0$ are obtained when you move from $\bar{x}$ in the direction y. The set of all these points $\{x : x = \bar{x} + \alpha y, \alpha \geqq 0\}$ is the **half-line** or **ray** through $\bar{x}$ in the direction of y. See Section 1.1.1.

Step length

Given $\bar{x} \in \mathbf{R}^n$, $y \in \mathbf{R}^n$, $y \neq 0$; for $\alpha > 0$, the point $\bar{x}+\alpha y$ is obtained by taking a step of length α from $\bar{x}$ in the direction of y. In this process α is the **step length**.

Feasible direction

Given a set $\mathbf{\Gamma} \subset \mathbf{R}^n$, and a point $\bar{x} \in \mathbf{\Gamma}$; the direction $y \in \mathbf{R}^n$, $y \neq 0$, is called a **feasible direction** at $\bar{x}$ for $\mathbf{\Gamma}$ if there exists a positive number $\bar{\alpha}$ such that $\bar{x} + \alpha y \in \mathbf{\Gamma}$ for all $0 \leqq \alpha \leqq \bar{\alpha}$. Thus the direction y is a feasible direction at $\bar{x}$ for $\mathbf{\Gamma}$ iff an initial segment of positive length on the half-line through $\bar{x}$ in the direction y is contained in $\mathbf{\Gamma}$.

Given an optimization problem, and a feasible solution $\overline{x}$ for it, the direction y (in the x-space) is said to be a feasible direction at $\overline{x}$ for this optimization problem if there exists an $\overline{\alpha} > 0$ such that $\overline{x}+\alpha y$ is a feasible solution to the problem for all $0 \leqq \alpha \leqq \overline{\alpha}$.

Descent direction

Let $\theta(x)$ be a real valued function defined over $x \in \mathbf{R}^n$. The direction $y \in \mathbf{R}^n$, $y \neq 0$, is said to be a **descent direction** for $\theta(x)$ at $\bar{x}$ if $\theta(\bar{x}+\alpha y) < \theta(\bar{x})$ whenever α is positive and sufficiently small. So by moving from $\bar{x}$ a small but positive step length in a descent direction, $\theta(x)$ is guaranteed to strictly decrease in value.

A descent direction for a minimization problem at a feasible solution $\overline{x}$, is a feasible direction for the problem at $\overline{x}$, which is a descent direction at $\overline{x}$ for the objective function being minimized.

Line search problem, line search method

Let $\theta(x)$ be a real valued function defined on $\mathbf{R}^n$. Let $\bar{x} \in \mathbf{R}^n$ be a given point and $y \in \mathbf{R}^n$, $y \neq 0$ a given direction. The problem of minimizing $\theta(\bar{x} + \alpha y)$ over $a \leqq \alpha \leqq b$ where a, b are given bounds on α, is called a **line search problem** or a **line minimization problem**; and any method for solving such a problem is called a **line search method**. Since $\bar{x}$, y are given, $\theta(\bar{x} + \alpha y)$ is purely a function of the single variable α, if we denote $\theta(\bar{x} + \alpha y) = f(\alpha)$, the line search problem is the one dimensional minimization problem of finding the minimum of $f(\alpha)$ over $a \leqq \alpha \leqq b$. Typically, in most line search problems encountered in applications, we will have $a = 0$ and b is either a finite positive number, or $+\infty$. When b is finite, the problem is often called a **constrained line search problem**. Several line search methods are discussed in Section 10.7. Many nonlinear programming algorithms use line search methods repeatedly in combination with special subroutines for generating feasible descent directions.

Hereditary symmetry, hereditary PD

Many algorithms for nonlinear programming (for example those discussed in Section 1.3.6 or Chapter 10) are iterative methods which maintain a square matrix B of order n and update it in each step. Let B_t denote this matrix in the tth step. The updating formula in this method provides B_{t+1} as a function of B_t and other quantities which are computed in the tth step or earlier. This updating procedure is said to possess the **hereditary symmetry property** if for any t, the fact that B_t is symmetric implies that B_{t+1} is also symmetric. Similarly, the updating procedure possesses the **hereditary PD property** if for any t the fact that B_t is PD implies that B_{t+1} is also PD. Thus, if the updating procedure has the hereditary symmetry and PD properties, and the initial matrix B used in the method is both symmetric and PD, the matrices B_t obtained in all the steps of the method will also be symmetric and PD.

Active set method

Any method for solving an NLP which partitions the set of inequality constraints into two groups — the active set consisting of those inequalities which are to be treated as active, that is, as equality constraints; and the inactive set. Inequality constraints in the inactive set are presumed to hold as strict inequalities at the optimum solution and are essentially ignored. The remaining problem is solved (treating all the constraints as equality constraints) by any method for solving equality constrained optimization problems. Active set methods also have procedures for revising the active set (either deleting inequality constraints from it, or adding inequality constraints from the inactive set into it) in each step, based on information accumulated in the method so far.

Convex programming problem, nonconvex programming problem

A problem in which a convex objective function is to be minimized over a convex set (usually of the form: minimize $\theta(x)$, subject to $g_i(x) \geqq 0$, $i = 1$ to m and $h_t(x) = 0$, $t = 1$ to p; where all the functions are given and $\theta(x)$ is convex; $g_i(x)$ are concave for all i; and $h_t(x)$ is affine for all t) is said to be a **convex programming problem**. A **nonconvex programming problem** is one which is not convex, that is, does not belong to the above class. For a convex programming problem every local minimum is a global minimum. In general, it is very hard to find the global minimum in a nonconvex programming problem. Necessary and sufficient conditions for optimality are available for convex programming problems. For nonconvex programming problems we have some necessary conditions for a point to be a local minimum, and sufficient conditions for a given point to be a local minimum. No simple set of conditions which are both necessary and sufficient for a given point to be a local minimum, are known for general nonconvex programming problems.

Merit function

In a nonlinear program where an objective function defined on $\mathbf{R}^n$ is to be minimized subject to constraints, a **merit function** is a real valued function defined on $\mathbf{R}^n$, it consists of the objective function plus penalty terms for constraint violations. Usually the penalty terms come from either the absolute-value penalty function (L_1-penalty function) or the quadratic penalty function. Minimizing the merit function balances the two competing goals which result from the desire to decrease the objective function while reducing the amount by which the constraints fail to be satisfied. See Section 1.3.6.

Cauchy-Schwartz inequality

Let x, y be two column vectors in $\mathbf{R}^n$. Then $|x^T y| \leqq ||x||.||y||$, this inequality is known as the **Cauchy-Schwartz inequality**. To prove it consider the quadratic equation in one variable λ, $f(\lambda) = (\lambda x + y)^T(\lambda x + y) = \lambda^{\mathbf{2}}||x||^{\mathbf{2}} + 2\lambda x^T y + ||y||^{\mathbf{2}} = 0$. Since $f(\lambda) = ||\lambda x + y||^{\mathbf{2}}$, it is always $\geqq 0$. This implies that the equation $f(\lambda) = 0$ can have at most one real solution in λ. It is well known that the quadratic equation $a\lambda^{\mathbf{2}} + b\lambda + c = 0$ has at most one real solution iff $b^{\mathbf{2}} - 4ac \leqq 0$, applying this to the equation $f(\lambda) = 0$, we conclude that $(x^T y)^{\mathbf{2}} \leqq ||x||^{\mathbf{2}}.||y||^{\mathbf{2}}$, that is, $|x^T y| \leqq ||x||.||y||$. Also, the quadratic equation $a\lambda^{\mathbf{2}} + b\lambda + c = 0$ has exactly one real solution if $b^{\mathbf{2}} - 4ac = 0$. Applying this to $f(\lambda) = 0$, we conclude that $f(\lambda) = 0$ has a real solution if $|x^T y| = ||x||.||y||$, in this case since $f(\lambda) = ||\lambda x + y||^{\mathbf{2}} = 0$ for some real λ, we must have $\lambda x + y = 0$, or y is scalar multiple of the vector x. Thus, if the Cauchy-Schwartz inequality holds as an equation for two vectors $x, y \in \mathbf{R}^n$, one of these vectors must be a scalar multiple of the other.

Cholesky factor

If M is a square matrix of order n which is symmetric and positive definite, there exists a lower triangular matrix F of order n with positive diagonal elements, satisfying $M = FF^T$. This matrix F is known as the **Cholesky factor** of M. For efficient methods for computing Cholesky factors, see books on computational linear algebra, or [1.28, 2.26].

Homotopy method

To solve a system by a **homotopy method**, we continuously deform a simple system with a known solution, into the system we are trying to solve. For example, consider the problem of solving a smooth system of n equations in n unknowns "$g(x) = 0$". Let a be an initial point from $\mathbf{R}^n$, consider the simple system of equations "$x = a$" with a known solution. Let $F(x, \lambda) = \lambda g(x) + (1 - \lambda)(x - a)$, on $0 \leqq \lambda \leqq 1$, $x \in \mathbf{R}^n$, $F(x, \lambda)$ is continuous in x and λ. The system "$F(x, \lambda) = 0$", treated as a system of equations in x, with λ as a parameter with given value between 0 and 1; is the simple system when $\lambda = 0$, and the system we want to solve when $\lambda = 1$. As the parameter λ varies from 0 to 1, the system "$F(x, \lambda) = 0$" provides a **homotopy** (contiuous deformation) of the simple system "$x = a$" into the system "$g(x) = 0$". The method for solving "$g(x) = 0$" based on the homotopy $F(x, \lambda)$, would follow the curve $x(\lambda)$ (where $x(\lambda)$ is a solution of $F(x, \lambda) = 0$ as a function of the homotopy parameter λ) beginning with $x(0) = a$, until λ assumes the value 1 at which point we have a solution for "$g(x) = 0$".

Principal rearrangement of a square matrix

Let M be a given square matrix of order n. Let $p = (i_1, \ldots, i_n)$ be a permutation of $(1, \ldots, n)$. The square matrix P of order n whose rows are $I_{i_1 .}$, $I_{i_2 .}, \ldots, I_{i_n .}$ in that order is the permutation matrix corresponding to p. P is obtained by essentially permuting the rows of the unit matrix I of order n using the permutation p. The matrix $M' = PMP^T$ is known as the principal rearrangement of M according to the permutation p. Clearly M' is obtained by first rearranging the rows of M according to the permutation p, and in the resulting matrix, rearranging the columns again accordng to the same permutation p. See Section 3.2.1.

Euclidean distance, rectilinear distance

Let $x = (x_j)$, $y = (y_j)$ be two point in $\mathbf{R}^n$. The Euclidean distance between x and y is $||x - y|| = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^{\mathbf{2}}}$. The rectilinear distance between x and y is $\sum_{j=1}^{n} |x_j - y_j|$.

Steepest descent direction at a feasible solution, in a continuous minimization problem.

First, consider an unconstrained minimization problem

$$\text{minimize } \theta(x) \text{ over } x \in \mathbf{R}^n \qquad \text{(i)}$$

where $\theta(x)$ is a real valued continuous function defined over $\mathbf{R}^n$.

Given any direction $y \in \mathbf{R}^n$, $y \neq 0$, the directional derivative of $\theta(x)$ at a point $\overline{x}$ in the direction y is defined to be

$$\text{limit } \frac{\theta(\overline{x} + \alpha y) - \theta(\overline{x})}{\alpha}$$

as $\alpha \to 0^+$, and denoted by $\theta'(\overline{x}; y)$, when it exists. If $\theta(x)$ is differentiable at $\overline{x}$, then $\theta'(\overline{x}; y) = \nabla\theta(\overline{x})y$. In general, $\theta'(\overline{x}, y)$ may exist even if $\theta(x)$ is not differentiable at $\overline{x}$.

$\theta'(\overline{x}; y)$ measures the rate of change in $\theta(x)$ at $x = \overline{x}$, when moving in the direction y.

The direction y is said to be a descent direction at $\overline{x}$ for problem (i), if $\theta'(\overline{x}; y) < 0$.

If $\overline{x}$ is a local minimum for (i), there is no descent direction for (i) at $\overline{x}$, and hence no steepest descent direction. Unfortunately, the converse of this statement may not always be true, that is, the absence of a descent direction at a point $\overline{x}$ does not imply that $\overline{x}$ is a local minimum. See Exercise 20 in Appendix 6. This just means that descent methods are not always guaranteed to find a local minimum.

If $\overline{x}$ is not a local minimum for (i), an optimum solution of

$$\text{minimize } \theta'(\overline{x}; y) \text{ subject to norm } (y) = 1 \qquad \text{(ii)}$$

is called a steepest descent direction at $\overline{x}$ for (i), under the particular norm used, if it is a descent direction at $\overline{x}$ for (i). In (ii), norm (y) is a function which measures the distance between the points 0 and y is $\mathbf{R}^n$. Different norms may lead to different steepest descent directions.

In optimization literature, usually norm (y) is taken as $y^T Ay$ where A is some specified symmetric PD matrix of order n (taking $A = I$, the unit matrix of order n, leads to the Euclidean norm).

Now consider a constrained continuous minimization
problem. Let $\mathbf{K} \subset \mathbf{R}^n$ denote its set of feasible solutions. Then this problem is of the form

$$\text{minimize } \theta(x) \text{ subject to } x \in \mathbf{K} \qquad \text{(iii)}$$

where the objective function $\theta(x)$ is a real valued continuous function defined over $\mathbf{R}^n$. Let $\overline{x} \in \mathbf{K}$ be a given feasible solution.
Again, if $\overline{x}$ is a local minimum for (iii), there is no descent direction and hence no steepest descent direction for (iii) at $\overline{x}$. If $\overline{x}$ is not a local minimum for (iii), any optimum solution of

$$\begin{aligned} &\text{minimize } \theta'(\overline{x}; y) \\ &\text{subject to norm of } (y) = 1, \\ &\quad \text{and } y \text{ is a feasible direction} \\ &\quad \text{at } \overline{x} \text{ for } \mathbf{K}, \text{ and a descent} \\ &\quad \text{direction for } \theta(x) \text{ at } \overline{x} \end{aligned} \qquad \text{(iv)}$$

is known as a steepest descent direction for (iii) at the feasible solution $\overline{x}$.

Descent methods

Descent methods for smooth minimization problems
have the following features. They are initiated with a feasible solution, x^0, for the problem, and generate a sequence $\{x^r : r = 0, 1, 2, \ldots\}$ of feasible points. For each r, the objective value at x^{r+1} is strictly less than the objective value at x^r. For $r \geqq 0$, step $r + 1$ of the method consists of the following two substeps.

1. Generate a feasible direction, y^r, for the problem at the present feasible point x^r, which is a descent direction for the objective function.

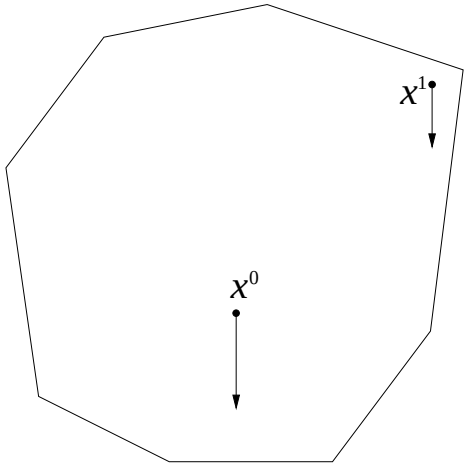
2. Carry out a line search on the half-line $\{x : x = x^r + \lambda y^r, \lambda \geqq 0\}$ for improving the objective value. For this, one has to determine the maximum value of λ, say $\overline{\lambda}$, such that $x^r + \lambda y^r$ remains feasible to the problem for all $0 \leqq \lambda \leqq \overline{\lambda}$ and then solve the line minimization problem of minimizing the objective function over $\{x : x = x^r + \lambda y^r, 0 \leqq \lambda \leqq \overline{\lambda}\}$, the output of which is the next point in the sequence, x^{r+1}.

If there exists no feasible descent direction at x^r, the method terminates with x^r while carrying out substep 1 (unfortunately, this does not guarantee that x^r is even a local minimum for the problem, it just means that we are unable to improve on the point x^r using descent methods.) If subsetp 1 does produce a direction y^r, from the definition of feasible descent directions, $\overline{\lambda}$ is guaranteed to be positive in substep 2 (it may happen that $\overline{\lambda} = \infty$). Different descent methods use different procedures for carrying out substeps 1, 2.

Therefore, the important feature of descent methods is that each move is made along a straight line, and results in a strict improvement in objective value. Since the objective value strictly improves in each step (assuming that the method does not terminate in that step), the sequence of points generated by a descent method is called a **descent sequence**.

Karmarkar's algorithm for LP and an intuitive justification for it

A detailed description of Karmarkar's algorithm, including complete proofs of its polynomial boundedness are provided in Section 11.4. Here we give a statement of this algorithm, with an intuitive justification, for someone interested in an overview without all the technical details and the proofs.
Consider the problem of minimizing a linear function on a convex polytope.



One can improve the current solution substantially by moving in the steepest descent direction, if the current solution is near the center of the feasible region, as in x^0 in the figure given above; but not so if it is near the boundary, as in x^1.
The main ideas behind Karmarkar's algorithm are the following:
i) If the current feasible solution is near the center of the feasible region, it makes sense to move in the steepest descent direction.
ii) If it is possible to transform the problem without changing it in an essential way, that moves the current feasible solution near the center of the feasible region, do it. Karmarkar uses a projective scaling transformation to do exactly this.
A (relative) interior feasible solution to an LP is one which satisfies all inequality constraints as strict inequalities. The basic strategy of Karmarkar's algorithm is to start at a (relative) interior feasible solution, and to carry out a projective scaling transformation to move the current solution to the center.

In the transformed problem, move in the steepest descent direction from this center, but not all the way to the (relative) boundary. Repeat as often as necessary.

Karmarkar considers linear programming problems in the following form

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & Ax = 0 \\ & e^T x = 1 \\ & x \geqq 0 \end{array} \qquad \text{(P)}$$

where A is a given matrix of order $m \times n$, and e^T is the row vector of all 1's in $\mathbf{R}^n$. The set $\mathbf{S} = \{x : x \in \mathbf{R}^n \text{ and } e^T x = 1, x \geqq 0\}$ is the standard $(n-1)$ dimensional simplex in $\mathbf{R}^n$. The problem (P) is assumed to satisfy the following assumptions.

(1) The point $a^0 = (1/n)e = (1/n, \ldots, 1/n)^T$, the center of $\mathbf{S}$, is feasible to (P).

(2) The problem (P) has an optimum solution, and the optimum objective value in (P) is zero.

Methods for transforming any LP into the form (P) satisfying conditions (1), (2), are discussed in Section 11.4. This is the initialization work before applying Karmarkar's algorithm on an LP. While these initialization methods are simple and mathematically correct, they can ruin the practical efficiency unless done in a clever way. Practically efficient initialization techniques in implementing Karmarkar's algorithm, are the object of intense research investigations at the moment.

Let us now consider the LP (P) satisfying (1) and (2). Karmarkar's method generates a sequence of feasible solutions for (P), $x^0 = a^0, x^1, x^2, \ldots$, all of them in the relative interior of $\mathbf{S}$ (i. e., $x^r > 0$ for all r), with cx^r monotonic decreasing. The method is terminated when we reach a t such that the objective value cx^t is sufficiently close to the optimum objective value of 0. So the terminal solution x^t is a near optimum solution to (P). A pivotal method (needing at most n pivot steps) that leads to an optimum extreme point solution of (P) from a near optimum solution, is discussed in Section 11.4, it

can be used in a final step if necessary. We now provide the general step.

General step $r+1$ in Karmarkar's algorithm: Let $x^r = a = (a_1, \ldots, a_n)^T > 0$ be the current feasible solution of (P). Define D as the $n \times n$ diagonal matrix with diagonal entries $a_1, \ldots, a_n$, that is

$$D = \begin{pmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{pmatrix}.$$

Since the matrix D depends on the current solution, you get a different D in each step. Use the projective transformation $T : \mathbf{S} \to \mathbf{S}$, defining new variables $y = (y_1, \ldots, y_n)$ by

$$y = T(x) = \frac{D^{-1}x}{e^T D^{-1}x}.$$

Since D is a diagonal matrix with positive diagonal entries, D^{-1} is the diagonal matrix whose ith diagonal entry is $(1/a_i)$. For every $x \in \mathbf{S}$, $T(x) \in \mathbf{S}$. Also, points in the relative interior of $\mathbf{S}$ in the x-space map into points in the relative interior of $\mathbf{S}$ in the y-space. The current feasible solution a of (P) in the x-space, maps into the solution $a^0 = (1/n, \ldots, 1/n)$, the center of the simplex $\mathbf{S}$ in the y-space, under this transformation.
To transform the problem (P), we use the inverse transformation

$$x = T^{-1}(y) = \frac{Dy}{e^T Dy}.$$

It can be verified that this transforms the original LP into

$$\begin{aligned} \text{minimize} \quad & \frac{cDy}{e^T Dy} = \theta(y) \\ \text{subject to} \quad & ADy = 0 \\ & e^T y = 1 \\ & y \geqq 0. \end{aligned} \tag{Q}$$

The constraints remain linear and essentially in the

same form as those in (P), but the objective function in (Q) is nonlinear.
Since the current solution for (Q) is a^0, the center of $\mathbf{S}$, it makes sense to move from a^0, in the steepest descent direction in (Q) at a^0. Since $a^0 > 0$, the set of feasible directions for (Q) at a^0 is $\{\xi : \xi \in \mathbf{R}^n, AD\xi = 0, e^T\xi = 0\}$. Let

$$B = \begin{pmatrix} AD \\ \dots \\ e^T \end{pmatrix}.$$

At a^0, the denominator in $\theta(y)$, $e^T Dy$, is equal to $(1/n)$, and it remains quite constant in a small neighborhood of a^0. So, the steepest descent direction for (Q) at the current point a^0 can be approximated by the steepest descent direction for the objective function cDy subject to the same constraints as in (Q), this is the solution of

$$\begin{array}{ll} \text{minimize} & cD\xi \\ \text{subject to} & B\xi = 0 \\ & \|\xi\| = 1. \end{array}$$

The optimum solution of this problem is $\hat{c}_p/\|\hat{c}_p\|$, where

$$\hat{c}_p = cD(I - B^T(BB^T)^{-1}B)$$

$\hat{c}_p$ is the orthogonal projection of cD onto the subspace $\{\xi : B\xi = 0\}$. So, the next point for (Q) is of the form

$$y' = a^0 - \beta\hat{c}_p/\|\hat{c}_p\|$$

where β is a positive step length. β can be chosen as large as possible, but keeping $y' > 0$. This leads to the new solution x^{r+1} for the original problem (P), where

$$x^{r+1} = \frac{Dy'}{e^T Dy'}.$$

If cx^{r+1} is sufficiently close to 0, terminate with cx^{r+1} as a near optimum solution for (P), otherwise, go to the next step with x^{r+1} as the current solution.