Interior-Point Polynomial Algorithms in Convex Programming

Yurii Nesterov
Arkadii Nemirovskii

siam Studies in Applied Mathematics

Yurii Nesterov and Arkadii Nemirovskii

Interior-Point Polynomial Algorithms in Convex Programming

Society for Industrial and Applied Mathematics
siam *Philadelphia*

Second Printing, 1994. Third Printing, 2001.

Library of Congress Cataloging-in-Publication Data

Nesterov, IU. E.
Interior-point polynomial algorithms in convex programming /Yurii Nesterov, Arkadii Nemirovskii.
p. cm. — (SIAM studies in applied mathematics ; vol.13)
Includes bibliographical references and index.
ISBN 0-89871-319-6
ISBN 0-89871-515-6 (pbk.)
1. Programming (Mathematics) 2. Convex programming
I. Nemirovski, Arkadii Semenovich. II.Title. III. Series: SIAM studies in applied mathematics ; 13
QA402.5.N4615 1993
519.7'6--dc20 93-38912

Foreword

In this book, Nesterov and Nemirovskii describe the first *unified theory* of polynomial-time interior-point methods. Their approach provides a simple and elegant framework in which all known polynomial-time interior-point methods can be explained and analyzed. Perhaps more important for applications, their approach yields polynomial-time interior-point methods for a very wide variety of problems beyond the traditional linear and quadratic programs.

The book contains new and important results in the general *theory* of convex programming, e.g., their "conic" problem formulation in which duality theory is completely symmetric. For each algorithm described, the authors carefully derive precise bounds on the computational effort required to solve a given family of problems to a given precision. In several cases they obtain better problem complexity estimates than were previously known.

The detailed proofs and lack of "numerical examples" might suggest that the book is of limited value to the reader interested in the practical aspects of convex optimization, but nothing could be further from the truth. An entire chapter is devoted to potential reduction methods precisely because of their great efficiency in practice (indeed, some of these algorithms are worse than path-following methods from the complexity theorist's point of view). Although it is not reported in this book, several of the new algorithms described (e.g., the projective method) have been implemented, tested on "real world" problems, and found to be extremely efficient in practice.

Nesterov and Nemirovskii's work has profound implications for the applications of convex programming. In many fields of engineering we find convex problems that are not linear or quadratic programs, but are of the form readily handled by their methods. For example, convex problems involving matrix inequalities arise in control system engineering. Before Nesterov and Nemirovskii's work, we could observe that such problems can be solved in polynomial time (by, e.g., the ellipsoid method) and therefore are, at least in a *theoretical* sense, tractable. The methods described in this book make these problems tractable in *practice*.

Karmakar's contribution was to demonstrate the first algorithm that solves linear programs in polynomial time and with practical efficiency. Similarly, it is one of Nesterov and Nemirovskii's contributions to describe algorithms that solve, in polynomial time and with practical efficiency, an extremely wide class of convex problems beyond linear and quadratic programs.

Stephen Boyd *Stanford, California*

Acknowledgments

During our five-year work on this book, we had (and intensively used) the opportunity to discuss the subject and to collaborate with many of our colleagues. We are greatly indebted to Farid Alizadeh, Alfred Auslender, Aharon Ben-Tal, Stephen Boyd, Ronald Freund, Eugene Golsthein, Osman Guler, Dick den Hertog, Florian Jarre, Leonid Khachiyan, Ken Kortanek, Claude Lemarechal, Olvi Mangasarian, Boris Polyak, Roman Polyak, Cornelis Roos, Naum Shor, Linus Shrage, Michael Todd, Pravin Vaidya, Jean-Philippe Vial, and Yinyu Ye for their advice and remarks; in particular, the courses given by Michael Todd and Ronald Freund on the basis of initial version of the book helped us to hopefully improve the order of exposition. We are especially thankful to Stephen Boyd, Claude Lemarechal, and Olvi Mangasarian for encouraging us to write this monograph.

It was our pleasure to collaborate with SIAM in processing the manuscript. We highly appreciate valuable comments of the anonymous referees, which helped to improve the initial text. We are greatly impressed by the professionalism of Acquisitions Editor Susan Ciambrano, and also by her care and patience.

Having finished this book in Paris, we express our gratitude to Claude Lemarechal and Jean-Philippe Vial for their hospitality and for the fine working facilities we were given.

Preface

The purpose of this book is to present the general theory of interior-point polynomial-time methods for convex programming. Since the publication of Karmarkar's famous paper in 1984, the area has been intensively developed by many researchers, who have focused on linear and quadratic programming. This monograph has given us the opportunity to present in one volume all of the major theoretical contributions to the theory of complexity for interior-point methods in optimization. Our aim is to demonstrate that all known polynomial-time interior-point methods can be explained on the basis of general theory, which allows these methods to extend into a wide variety of nonlinear convex problems. We also have presented for the first time a definition and analysis of the self-concordant barrier function for a compact convex body.

The abilities of the theory are demonstrated by developing new polynomial-time interior-point methods for many important classes of problems: quadratically constrained quadratic programming, geometrical programming, approximation in L_p norms, finding extremal ellipsoids, and solving problems in structural design. Problems of special interest covered by the approach are those with positive semidefinite matrices as variables. These problems include numerous applications in modern control theory, combinatorial optimization, graph theory, and computer science.

This book has been written for those interested in optimization in general, including theory, algorithms, and applications. Mathematicians working in numerical analysis and control theory will be interested, as will computer scientists who are developing theory for computation of solutions of problems by digital computers. We hope that mechanical and electrical engineers who solve convex optimization problems will find this a useful reference.

Explicit algorithms for the aforementioned problems, along with detailed theoretical complexity analysis, form the main contents of this book. We hope that the theory presented herein will lead to additional significant applications.

Contents

Chapter 1

Introduction

1.1 Subject

The introduction of polynomial-time interior-point methods is one of the most remarkable events in the development of mathematical programming in the 1980s. The first method of this family was suggested for linear programming in the landmark paper of Karmarkar (see [Ka 84]). An excellent complexity result of this paper, as well as the claim that the performance of the new method on real-world problems is significantly better than the one of the simplex method, made this work a sensation and subsequently inspired very intensive and fruitful studies.

Until now, the activity in the field of interior-point methods focuses mainly on linear programming. At the same time, we find that the nature of the methods, is in fact, independent of the specific properties of LP problems, so that these methods can be extended onto more general convex programs. The aim of this book is twofold:

- To present a general approach to the design of polynomial-time interior-point methods for nonlinear convex problems, and
- To illustrate the abilities of the approach by a number of important examples (quadratically constrained quadratic programming, geometrical programming, approximation in L_p norm, minimization of eigenvalues, among others).

1.2 Essence of the approach

After the seminal paper of Renegar (see [Re 86]), it became absolutely clear that the new polynomial-time algorithms belong to the traditional class of *interior penalty methods* studied in the classical monograph of Fiacco and McCormick (see [FMcC 68]). To solve a convex problem

$$(f) \qquad \text{minimize } f_0(x) \text{ s.t. } f_i(x) \leq 0,\ i = 1, ..., m(f), \qquad x \in \mathbf{R}^{n(f)}$$

by an interior penalty method, it is first necessary to form a barrier function for the feasible domain

$$G_f = \{x \mid f_i(x) \leq 0,\ i = 1, ..., m(f)\}$$

of the problem, i.e., smooth and strongly convex on the interior of the domain function F tending to infinity along each sequence of interior points converging to a boundary point of G_f. Given such a barrier, one approximates the constrained problem (f) by the family of unconstrained problems, e.g., by the *barrier-generated family*

$$(f_t) \qquad \text{minimize } f_t(x) = tf_0(x) + F(x),$$

where $t > 0$ is the penalty parameter. Under extremely mild restrictions, the solutions $x(t)$ to (f_t) tend to the optimal set of (f) as t tends to ∞. The classical scheme suggests following the trajectory $x(t)$ along certain sequence $t_i \to \infty$ of values of the penalty. By applying to (f_t) a method for unconstrained minimization, one forms "tight" approximations to $x(t_i)$, and these approximations are regarded as approximate solutions to (f). This scheme leads to *barrier methods.*

Another "unconstrained approximation" of the constrained problem (f) is given by the family

$$(f_t^c) \qquad \text{minimize } f_t^c(x) = \phi(t - f_0(x)) + F(x),$$

where $t > f^*$ (f^* is the optimal value in (f)) and ϕ is a barrier for the nonnegative half-axis. As $t \to f^* + 0$, the solutions $x^c(t)$ to the problems (f_t^c) tend to the optimal set of (f), and one can follow the path $x^c(t)$ along a sequence $t_i \to f^* + 0$ by applying to (f_t^c) a method for unconstrained minimization. The latter scheme originating from Huard (see, e.g., [BH 66]) leads to what is called *methods of centers.*

Note that the above schemes possess two main "degrees of freedom": First, it is possible to use various barriers; second, one can implement any method for unconstrained minimization. Regarding the first issue, the classical recommendation, at least in the case of smooth convex constraints, is to use barriers that are compositions of constraints,

$$F(x) = \sum_{i=1}^{m(f)} \psi(-f_i(x)),$$

where $\psi(s)$ is a barrier for the nonnegative half-axis, e.g.,

$$\psi(s) = s^{-\kappa}, \quad \kappa > 0; \qquad \psi(s) = -\kappa \ln s, \quad \kappa > 0; \qquad \psi(s) = \mathrm{e}^{1/s}, \quad \text{etc.}$$

Regarding choice of the method for unconstrained minimization, there were almost no firm theoretical priorities; the computational experience was in favor of the Newton method, but this recommendation had no theoretical background.

Such a background was first given by Renegar in [Re 86]. Renegar demonstrated that in the case of a linear programming problem (f) (all f_i, $i =$

$1, \ldots, m(f)$, are linear), the method of centers associated with the *standard logarithmic barrier*

$$F(x) = -\sum_{i=1}^{m} \ln(-f_i(x))$$

for the feasible polytope G_f of the problem and with

$$\phi(s) = -\omega \ln(s), \qquad \omega > 0$$

allows us to decrease the residual $t_i - f^*$ at a linear rate at the cost of a *single* step of the Newton method as applied to $(f^c_{t_i})$. Under appropriate choice of the weight ω at the term $\ln(t - f_0(x))$ (namely, $\omega = O(m(f))$), one can force the residual $t_i - f^*$ to decrease as $\exp\{-O(1)i/m^{1/2}(f)\}$. Thus, to improve the accuracy of the current approximate solution by an absolute constant factor, it suffices to perform $O(m^{1/2}(f))$ Newton steps, which requires a polynomial in *the size* $(n(f), m(f))$ of the problem number of arithmetic operations; in other words, the method proves to be *polynomial*. Similar results for the barrier method associated with the same logarithmic barrier for a linear programming problem were established by Gonzaga [Go 87].

We see that the central role in the modern interior-point methods for linear programming is played by the standard logarithmic barrier for the feasible polytope of the problem. To extend the methods onto nonlinear problems, one should understand the properties of the barrier responsible for polynomiality of the associated interior-point methods. Our general approach originates in [Ns 88b], [Ns 88c], [Ns 89]. It is as follows: Among all various properties of the logarithmic barrier, only two are responsible for all nice features of the associated with F interior-point methods. These two properties are (i) the Lipschitz continuity of the Hessian F'' of the barrier with respect to the local Euclidean metric defined by the Hessian itself as

$$|D^3F(x)[h,h,h]| \leq \text{const}_1\{D^2F(x)[h,h]\}^{3/2}$$

for all x from the interior of G and all $h \in \mathbf{R}^n$; and (ii) the Lipschitz continuity of the barrier itself with respect to the same local Euclidean structure

$$|DF(x)[h]| \leq \text{const}_2\{D^2F(x)[h,h]\}^{1/2}$$

for the same as above x and h.

Now (i) and (ii) do not explicitly involve the polyhedral structure of the feasible domain G of the problem; given an arbitrary closed convex domain G, we can consider a interior penalty function for G with these properties (such a function will be called a *self-concordant barrier* for G). The essence of the theory is that, given a self-concordant barrier F for a closed convex domain G, we can associate with this barrier interior point methods for minimizing *linear* objectives over G in the same way as is done in the case of the standard logarithmic barrier for a polytope. Moreover, all polynomial-time interior-point methods known for LP admit the above extension. To improve the accuracy of a given approximate solution by an absolute constant factor, the resulting methods require the amount of steps that depends only on the *parameter of the barrier*, i.e., on certain combination of the above const_1 and const_2, while each of the steps is basically a step of the Newton minimization method as applied to F.

Note that the problem of minimizing a linear objective over a closed convex domain is universal for convex programming: Each convex program can be reformulated in this form. It follows that the possibility to solve convex programs with the aid of interior-point methods is limited only by our ability to point out self-concordant barriers for the resulting feasible domains. The result is that such a barrier always exists (with the parameter being absolute constant times the dimension of the domain); unfortunately, to obtain nice complexity results, we need a barrier with moderate arithmetic cost of computing the gradient and the Hessian, which is not always the case. Nevertheless, in many cases we can to point out "computable" self-concordant barriers, so that we can develop efficient methods for a wide variety of nonlinear convex problems of an appropriate analytical structure.

Thus, we see that there exist not only heuristic, but also theoretical reasons for implementing the Newton minimization method in the classical schemes of the barrier method and the method of centers. Moreover, we understand how to use the freedom in choice of the barrier: It should be self-concordant, and we are interested in this intrinsic property, in contrast to the traditional recommendations where we are offered a number of possibilities for constructing the barrier but have no priorities for choosing one of them.

1.3 Motivation

In our opinion, the main advantage of interior-point machinery is that, in many important cases, it allows us to utilize the knowledge of the analytical structure of the problem under consideration to develop an efficient algorithm. Consider a family $\mathcal{A}$ of solvable optimization problems of the type (f) with convex finite (say, on the whole $\mathbf{R}^{n(f)}$) objective and constraints.

Assume that we have fixed analytical structure of the functionals involved into our problems, so that each problem instance (f) belonging to $\mathcal{A}$ can be identified by a finite-dimensional real vector $\mathsf{D}(f)$ ("the set of coefficients of the instance"). Typical examples here are the classes of linear programming problems, linearly/quadratically constrained convex quadratic problems, and so forth. Assume that, when solving (f), the set of data $\mathsf{D}(f)$ form the input to the algorithm, and we desire to solve (f) to a prescribed accuracy ε, i.e., to find an approximate solution x_ε satisfying the relations

$$f_0(x_\varepsilon) \le f^* + \varepsilon, \quad f_i(x_\varepsilon) \le \varepsilon, \quad i = 1, \ldots, m(f),$$

where f^* is the optimal value in (f).

An algorithm that transforms the input $(\mathsf{D}(f), \varepsilon)$ into an ε-solution to (f) in a finite number of operations of precise real arithmetic will be called *polynomial*, if the total amount of these operations for all $(f) \in \mathcal{A}$ and all $\varepsilon > 0$ is bounded from above by $p(m(f), n(f), \dim\{\mathsf{D}(f)\}) \ln(V(f)/\varepsilon)$, where p is a polynomial. Here $V(f)$ is certain *scale parameter*, which can depend on the magnitudes of coefficients involved into (f) (a reasonable choice of the parameter is specific for the family under consideration). The ratio $\varepsilon/V(f)$ can be regarded as the relative accuracy, so that $\ln(V(f)/\varepsilon)$ is something like the amount of accuracy digits in an ε-solution. Thus, a polynomial-time algorithm is a procedure in which the arithmetic cost "per accuracy digit" does not exceed a polynomial of the *problem size* $(m(f), n(f), \dim\{\mathsf{D}(f)\})$. Polynomiality usually is treated as theoretical equivalent to the unformal notion "an effective computational procedure," and the efficiency of a polynomial-time algorithm, from the theoretical viewpoint, is defined by the corresponding "cost per digit" $p(m(f), n(f), \dim\{\mathsf{D}(f)\})$.

The concept of a polynomial-time algorithm was introduced by Edmonds [Ed 65] and Cobham [Co 65] (see also Aho et al. [AHU 76], Garey and Johnson [GJ 79], and Karp [Kr 72], [Kr 75]). This initial concept was oriented onto discrete problems; in the case of continuous problems with real data, it seems to be more convenient to deal with the above (relaxed) version of this concept.

Note that polynomial-time algorithms do exist in a sense, for "all" convex problems. Indeed, there are procedures (e.g., the ellipsoid method; see [NY 79]) that solve all convex problems (f) to relative (in a reasonable scale) accuracy ε at the cost of $O(p(n,m)\ln(n/\varepsilon))$ arithmetic operations and $O(q(n,m)\ln(n/\varepsilon))$ computations of the values and subgradients of the objective and the constraints, where p and q are polynomials (for the ellipsoid method, $p(n,m) = n^3(m+n)$, $q(n,m) = n^2$). Now, if our class of problems $\mathcal{A}$ is such that, given the data $\mathsf{D}(f)$, we can compute the above values and subgradients at a given point x in polynomial in $m(f)$, $n(f)$, $\dim\{\mathsf{D}(f)\}$ number of arithmetic operations, then the above procedure proves to be polynomial on $\mathcal{A}$.

A conceptual drawback of the latter scheme is that, although from the very beginning we possess complete information about the problem instance, we make only "local" conclusions from this "global" information; in fact, in this scheme, we ignore our knowledge of the analytical structure of the problem under consideration (more accurately, this information is used only when computing the values and the subgradients of f_i). At the same time, the interior-point machinery is now the only known way to utilize the knowledge of analytical structure to improve—sometimes significantly—the theoretical efficiency of polynomial-time algorithms. Indeed, as already mentioned, the efficiency of a polynomial-time interior-point method is defined first by the parameter of the underlying barrier and second by the arithmetic cost at which one can form and solve the corresponding Newton systems; both these quantities depend more on the analytical structure of the objective and constraints than on the dimensions $m(f)$ and $n(f)$ of the problem.

1.4 Overview of the contents

Chapter 2 forms the technical basis of the book. Here we introduce and study our main notions of self-concordant functions and barriers.

Chapter 3 is devoted to the path-following interior-point methods. In their basic form, these methods allow us to minimize a linear objective f over a bounded closed convex domain G, provided that we are given a self-concordant barrier for the domain and a starting point belonging to the interior of the domain. In a path-following method, the barrier and the objective generate certain penalty-type family of functions and, consequently, the trajectory of minimizers of these functions; this trajectory converges to the optimal set of the problem. The idea of the method is to follow this path of minimizers: Given a strictly feasible approximate solution close, in a sense, to the point of the path corresponding to a current value of the penalty parameter, we vary the parameter in the desired direction and then compute the Newton iterate of the current approximate solution to restore the initial closeness between the updated approximate solution and the new point of the path. Of course, this scheme is quite traditional, and, generally speaking, it does not result in polynomial-time procedure. The latter feature is provided by self-concordance of the functions comprising the family.

We demonstrate that path-following methods known for LP (i.e., for the case when G is a polytope) can be easily explained and extended onto the case of general convex domains G. We prove that the efficiency ("cost per digit") of these methods is $O(\vartheta^{1/2})$, where ϑ is the parameter of the barrier (for the standard logarithmic barrier for an m-facet polytope one has $\vartheta = m$).

In Chapter 4 we extend onto the general convex case the potential reduction interior-point methods for LP problems; we mean the method of Karmarkar [Ka 84], the projective method [Nm 87], the primal-dual method of Todd and Ye [TY 87], and Ye [Ye 88a], [Ye 89]. The efficiency of the resulting method is $O(\vartheta)$ (for the generalized method of Karmarkar and the projective method) or $O(\vartheta^{1/2})$ (the generalized primal-dual method), where ϑ denotes the parameter of the underlying self-concordant barrier. Thus, the potential reduction methods, theoretically, have no advantages as compared to the path-following algorithms. From the computational viewpoint, however, these methods are much more attractive. The reason is that, for a potential reduction method, one can point out an explicit Lyapunov's function, and the accuracy of a feasible approximate solution can be expressed in terms of the potential (the less the potential, the better the approximate solution). At each strictly feasible solution, the theory prescribes a direction and a stepsize, which allows us to obtain a new strictly feasible solution with the value of the potential being "considerably" less than that at the previous approximate solution. To ensure the theoretical efficiency estimate, it suffices to perform this theoretical step, but we are not forbidden to achieve a deeper decreasing of the potential, say, with the aid of one-dimensional minimization of the potential in the direction prescribed by the theory. In real-world problems, these "large steps" significantly accelerate the method. In contrast to this, in a path-following method, we should maintain closeness to the corresponding trajectory, which, at least theoretically, is an obstacle for "large steps."

To extend potential reduction interior-point methods onto the general convex case, we use a special reformulation of a convex programming problem, the so-called conic setting of it (where we should minimize a linear functional over the intersection of an affine subspace and a closed convex cone). An important role in the extension is played by duality, which for conic problems attains very symmetric form and looks quite similar to the usual LP duality. Another advantage of the "conic format" of convex programs, which is especially important to the design of polynomial-time methods, is that this format allows us to exploit the widest group of transformations preserving convexity of feasible domains; we mean the projective transformations (to subject a conic problem to such a transformation is basically the same as to intersect the cone with another affine subspace).

As already mentioned, to solve a convex problem by an interior-point method, we should first reduce the problem to one of minimizing a linear objective over convex domain (which is quite straightforward) and, second, point out a "computable" self-concordant barrier for this domain (which is the crucial point for the approach). As shown in Chapter 2, every n-dimensional closed convex domain admits a self-concordant barrier with the parameter of order of n; unfortunately, the corresponding "universal barrier" is given by a multivariate integral and therefore cannot be treated as "computable." Nevertheless, the result is that there exists a kind of calculus of "computable" self-concordant barriers, which forms the subject of Chapter 5. We first point out "simple" self-concordant barriers for a number of standard domains arising in convex programming (epigraphs of standard functions on the axis, level sets of convex quadratic forms, the epigraph of the Euclidean norm, the cone of positive semidefinite matrices, and so forth). Second, we demonstrate that all standard (preserving convexity) operations with convex domains (taking images/inverse images under affine mappings and projective transformations, intersection, taking direct products, and so forth) admit simple rules for combining self-concordant barriers for the operands into a self-concordant barrier for the resulting domain. This calculus involves "rational linear algebra" tools only and, as applied to our "raw materials"—concrete self-concordant barriers for "standard" convex sets—allows us to form "computable" self-concordant barriers for a wide variety of convex domains arising in convex programming.

In Chapter 6 we illustrate the abilities of the developed technique. Namely, we present polynomial-time interior-point algorithms for a number of classes of nonlinear convex programs, including quadratically constrained quadratic programming, geometrical programming (in exponential form), approximation in L_p-norm, and minimization of the operator norm of a matrix linearly depending on the control vector. An especially interesting application is semidefinite programming, i.e., minimization of a linear functional of a symmetric matrix subjected to positive semidefiniteness restriction and a number of linear equality constraints. Note that, first, semidefinite programming is a nice field for interior-point methods (all path-following and potential reduction methods can be easily implemented for this class); second, semidefinite programming covers many important problems arising in various areas, from control theory to combinatorial optimization (e.g., the problem of minimizing the largest eigenvalue or the sum of k largest eigenvalues of a symmetric matrix). We conclude Chapter 6 with developing polynomial-time interior-point algorithms for two geometrical problems concerning extremal ellipsoids (the problems are to inscribe the maximum volume ellipsoid into a given polytope and to cover a given finite set in $\mathbf{R}^n$ by the ellipsoid of minimum volume). The first of these is especially interesting for nonsmooth convex optimization (it arises as an auxiliary problem in the *inscribed ellipsoid method* (see Khaciyan et al. [KhTE 88])).

Chapter 7 is devoted to variational inequalities with monotone operators. Here we extend the notion of self-concordance onto monotone operators and develop a polynomial-time path-following method for inequalities involving operators "compatible," in a sense, with a self-concordant barrier for the feasible domain of the inequality. Although the compatibility condition is a rather severe restriction, it is automatically satisfied for linear monotone operators, as well as for some interesting nonlinear operators (e.g., the operator arising, under some natural assumptions, in the pure exchange model of Arrow–Debreu).

In Chapter 8 we consider possibilities for acceleration of the path-following algorithms as applied to linearly constrained convex quadratic (in particular, LP) problems. Until now, the only known acceleration strategies were more or less straightforward modifications of the Karmarkar speed-up (see [Ka 84]) based on recursive updatings of approximate inverses to the matrices arising at the sequential Newton-type steps of the procedure under consideration. We describe four more strategies: Three are based on following the path with the aid of (prescaled) multistep methods for smooth optimization; in the fourth strategy, to find an approximate solution of a Newton system, we use the prescaled conjugate gradient method. All our strategies lead to the same worst-case complexity estimates as the known ones, but the new strategies seem to be more flexible and therefore can be expected to be more efficient in practice.

We conclude the exposition with Bibliography Comments. It seems to be impossible to give a detailed survey of the activity in the very intensively developing area of polynomial-time interior-point methods. Therefore we have restricted ourselves only with the papers closely related to the monograph. We realize that the "level of completeness" of our comments is far from being perfect and apologize in advance for possible lacunae.

The methods presented in the book are new, and we believe that they are promising for practical computations. The very preliminary experience we now possess supports this hope, but it in no sense is sufficient to make definite conclusions. Therefore our decision was to completely omit any numerical results.

Of course, we realize that it is computational experience, not theoretical results alone, that proves practical potential of an algorithm, and we hope that, by the methods presented in this book, this experience can soon be gained.

1.5 How to read this book

Basically, there are two ways of reading this book, depending on whether the reader is interested in the interior-point theory itself or its applications to concrete optimization problems. The theoretical aspect is detailed in Chapters 2–5, while applications of variational inequalities, in addition to the theory, can be found in Chapter 7 (it is possible to exclude Chapters 4 and 5 here). For specific explications of the theory of linear and linearly constrained quadratic programming, refer to Chapter 8.

Chapters 2 and 3 deal in general theory rather than in concrete applications; the reader interested in applications is expected to be familiar with the main concepts and results found there (with the exception of §2.5 and possible §2.4), but not necessarily with the proofs. Note that, for some applications (e.g., geometrical programming, approximation in L_p-norm, and finding extremal ellipsoids), only path-following methods are developed, and, consequently, those interested in these applications may move from chapter 3 directly to Chapter 6. Quadratically constrained quadratic problems, especially semidefinite programming with applications to control theory, can be found in §2.4 and Chapter 4 (at the level of concepts and schemes). If one wishes to deal with concrete applications that are not explicitly presented in this book and would like to attempt to develop new interior-point methods, refer to Chapter 5 for the techniques of constructing self-concordant barriers.