Lectures on Modern Convex Optimization

MPS/SIAM Series on Optimization

This series is published jointly by the Mathematical Programming Society and the Society for Industrial and Applied Mathematics. It includes research monographs, textbooks at all levels, books on applications, and tutorials. Besides being of high scientific quality, books in the series must advance the understanding and practice of optimization and be written clearly, in a manner appropriate to their level.

LECTURES ON MODERN CONVEX OPTIMIZATION

ANALYSIS, ALGORITHMS, AND ENGINEERING APPLICATIONS

Aharon Ben-Tal
Arkadi Nemirovski
Technion-Israel Institute of Technology
Haifa, Israel

siam
Society for Industrial and Applied Mathematics
Philadelphia

MPS
Mathematical Programming Society
Philadelphia

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data
Ben-Tal, A.
Lectures on modern convex optimization : analysis, algorithms, and engineering applications / Aharon Ben-Tal, Arkadi Nemirovski.
p. cm. — (MPS-SIAM series on optimization)
Includes bibliographical references and index.
ISBN 0-89871-491-5
1. Convex programming. 2. Mathematical optimization. I. Nemirovski, Arkadi Semenovich. II. Title. III. Series.

T57.815 .B46 2001
519.7'6—dc21

2001020818

This book is dedicated
to our friend and colleague, Jochem Zowe

Contents

Preface **xi**

1 Linear Programming **1**
1.1 Linear programming: Basic notions 1
1.2 Example: Tschebyshev approximation and its applications 2
1.2.1 Best uniform approximation 2
1.2.2 Application: Synthesis of filters 3
1.2.3 Filter synthesis revisited 5
1.2.4 Synthesis of arrays of antennae 7
1.3 Duality in linear programming 11
1.3.1 Certificates for solvability and insolvability 12
1.3.2 Dual to a linear programming program: Origin 16
1.3.3 Linear programming duality theorem 17
1.3.4 Illustration: Problem dual to the Tschebyshev approximation problem 19
1.3.5 Application: Truss topology design 21
1.4 Exercises to Lecture 1 30
1.4.1 Uniform approximation 30
1.4.2 Theorem on the alternative 34
1.4.3 Proof of the homogeneous Farkas lemma 35
1.4.4 Helley theorem 38
1.4.5 How many bars are needed in an optimal truss? 40

2 From Linear Programming to Conic Programming **43**
2.1 Orderings of $\mathbf{R}^m$ and convex cones 44
2.2 What is conic programming? 46
2.3 Conic duality 50
2.3.1 Geometry of the primal and dual problems 53
2.4 Conic duality theorem 57
2.4.1 Is something wrong with conic duality? 60
2.4.2 Consequences of the conic duality theorem 61
2.4.3 Robust solvability status 64
2.5 Conic duality revisited 67
2.6 Exercises to Lecture 2 73

2.6.1 Cones . . . 73
2.6.2 Conic problems . . . 76
2.6.3 Feasible and level sets of conic problems . . . 77

3 Conic Quadratic Programming 79
3.1 Conic quadratic problems: Preliminaries . . . 79
3.2 Examples of conic quadratic problems . . . 81
3.2.1 Best linear approximation of complex-valued functions . . 81
3.2.2 Contact problems with static friction . . . 82
3.3 What can be expressed via conic quadratic constraints? . . . 85
3.3.1 More examples of conic quadratic-representable functions and sets . . . 104
3.4 More applications . . . 109
3.4.1 Tschebyshev approximation in relative scale . . . 109
3.4.2 Robust linear programming . . . 110
3.4.3 Truss topology design . . . 120
3.5 Exercises to Lecture 3 . . . 131
3.5.1 Optimal control in discrete time linear dynamic system . . 131
3.5.2 Conic quadratic representations . . . 132
3.5.3 Does conic quadratic programming exist? . . . 137

4 Semidefinite Programming 139
4.1 Semidefinite cone and semidefinite programs . . . 139
4.1.1 Preliminaries . . . 139
4.2 What can be expressed via linear matrix inequalities? . . . 144
4.3 Applications I: Combinatorics . . . 159
4.3.1 Shor's semidefinite relaxation scheme . . . 161
4.3.2 Stability number, Shannon capacity, and Lovasz capacity of a graph . . . 164
4.3.3 MAXCUT problem . . . 170
4.3.4 Extensions . . . 172
4.3.5 $\mathcal{S}$-lemma . . . 175
4.4 Applications II: Stability analysis . . . 178
4.4.1 Dynamic stability in mechanics . . . 178
4.4.2 Lyapunov stability analysis and synthesis . . . 180
4.4.3 Interval stability analysis and synthesis . . . 189
4.5 Applications III: Robust quadratic programming . . . 202
4.6 Applications IV: Synthesis of filters and antennae arrays . . . 210
4.7 Applications V: Design of chips . . . 219
4.7.1 Building the model . . . 220
4.7.2 Wire sizing . . . 226
4.8 Applications VI: Structural design . . . 227
4.8.1 Building a model . . . 228
4.8.2 Standard case . . . 233
4.8.3 Semidefinite reformulation of the standard SSD problem 236
4.8.4 From primal to dual . . . 243

4.8.5 Back to primal . . . 248
4.8.6 Explicit forms of the standard truss and shape problems. . 252
4.9 Applications VII: Extremal ellipsoids . . . 257
4.9.1 Ellipsoidal approximations of unions and intersections of ellipsoids . . . 262
4.9.2 Approximating sums of ellipsoids . . . 264
4.10 Exercises to Lecture 4 . . . 276
4.10.1 Around positive semidefiniteness, eigenvalues, and $\succeq$-ordering . . . 276
4.10.2 Semidefinite representations of epigraphs of convex polynomials . . . 291
4.10.3 Lovasz capacity number and semidefinite relaxations of combinatorial problems . . . 293
4.10.4 Lyapunov stability analysis . . . 299
4.10.5 $\mathcal{S}$-lemma . . . 300
4.10.6 Antenna synthesis . . . 323
4.10.7 Ellipsoidal approximations . . . 326

5 Computational Tractability of Convex Programs 335
5.1 Numerical solution of optimization programs—preliminaries . . . 335
5.1.1 Mathematical programming programs . . . 335
5.1.2 Solution methods and efficiency . . . 336
5.2 Black box–represented convex programs . . . 342
5.3 Polynomial solvability of convex programming . . . 352
5.3.1 Polynomial solvability of convex programming . . . 359
5.4 Difficult problems and NP-completeness . . . 363
5.4.1 CCT—a quick introduction . . . 363
5.4.2 From the real arithmetic complexity theory to the CCT and back . . . 367

6 Interior Point Polynomial Time Methods for Linear Programming, Conic Quadratic Programming, and Semidefinite Programming 377
6.1 Motivation . . . 377
6.1.1 Interior point methods . . . 378
6.2 Newton method and the interior penalty scheme . . . 379
6.2.1 Unconstrained minimization and the Newton method . . . 379
6.2.2 Classical interior penalty scheme: Construction . . . 380
6.2.3 Classical interior penalty scheme: Drawbacks . . . 382
6.2.4 But. 382
6.3 Interior point methods for linear programming, conic quadratic programming, and semidefinite programming: Building blocks . . . 384
6.3.1 Canonical cones and canonical barriers . . . 384
6.3.2 Elementary properties of canonical barriers . . . 387
6.4 Primal-dual pair of problems and the primal-dual central path . . . 389
6.4.1 Problem(s) . . . 389
6.4.2 Central path(s) . . . 390

6.5 Tracing the central path 397
6.5.1 Path-following scheme 397
6.5.2 Speed of path-tracing 398
6.5.3 Primal and dual path-following methods 402
6.5.4 Semidefinite programming case 405
6.6 Complexity bounds for linear programming, conic quadratic programming, and semidefinite programming 421
6.6.1 Complexity of linear programming 422
6.6.2 Complexity of conic quadratic programming 423
6.6.3 Semidefinite programming 423
6.7 Concluding remarks 424
6.8 Exercises to Lecture 6 426
6.8.1 Canonical barriers 426
6.8.2 Scalings of canonical cones 427
6.8.3 Dikin ellipsoid 429
6.8.4 More on canonical barriers 431
6.8.5 Primal path-following method 432
6.8.6 Infeasible start path-following method 435

Solutions to Selected Exercises **443**
Exercises to Lecture 1 443
Exercises to Lecture 2 446
Exercises to Lecture 3 449
Exercises to Lecture 4 459
Exercises to Lecture 6 475

Index **485**

Preface

The goals. To make decisions optimally is a basic human desire. Whenever the situation and the objectives can be described quantitatively, this desire can be satisfied, to some extent, by using mathematical tools, specifically those provided by optimization theory and algorithms. For our purposes, a sufficiently general mathematical setting of an optimization problem is offered by *mathematical programming*:

$$\begin{array}{lrcl} \text{Minimize} & f_0(x) & & \\ \text{subject to (s.t.)} & & & \\ & f_i(x) & \leq & b_i,\ i = 1, \ldots, m, \\ & x & \in & X \subset \mathbf{R}^n. \end{array} \tag{P}$$

In this problem, we are given an *objective* $f_0(x)$ and finitely many *functional constraints* $f_i(x)$, $i = 1, \ldots, m$, which are real-valued functions of n-dimensional *design vector* x varying in a given domain X. Our goal is to minimize the objective over the *feasible set* of the problem—the set that is cut off the domain X by the system of inequalities $f_i(x) \leq b_i$, $i = 1, \ldots, m$.

In typical engineering applications, the design vector specifies a decision to be made (e.g., the physical sizes of the bars in a trusslike structure), the domain X is the set of "meaningful" values of the design vector, and the functional constraints represent design specifications—restrictions (physical, technical, financial) on certain characteristics of the decision.

The last decade has witnessed major progress in optimization, especially in the area of convex programming. Powerful modeling languages and database technology extended our abilities to model large-scale, complex real-life problems; progress in complexity theory improved our understanding of the advantages of certain algorithms, and the limitations of others, and resulted in development of efficient interior point algorithms for a large family of convex programs. Combined with the dramatic improvement of computers, this progress enables us today to solve problems that were considered out of reach for optimization just 10 years ago.

Regrettably, this promising state of affairs is yet unrealized, and consequently not utilized, by potential end users (engineers, managers, etc.). This book presents modern optimization, combining it with applications (mainly from engineering) and thus helping to bridge the gap between researchers and practitioners. This ultimate goal determines our approach and dictates specific targets on which we should focus our attention.

Theoretically, what modern optimization can solve well are *convex optimization problems*. In essence, the two-decade-long investigation of complexity issues in optimization can be summarized as follows:

> *From the viewpoint of the numerical processing of problem* (P)*, there exists a "solvable case"—the one of* convex *optimization problems, those where the domain X is a closed convex subset of $\mathbf{R}^n$, and the objective $f_0(x)$ and the functional constraints $f_i(x)$, $i = 1, \dots, m$, are convex functions on X.*
>
> *Under minimal additional computability assumptions* (which are satisfied in basically all applications)*, a convex optimization problem is computationally tractable—the computational effort required to solve the problem to a given accuracy grows moderately with the dimensions of the problem and the required number of accuracy digits.*
>
> *In contrast to this, general-type nonconvex problems are too difficult for numerical solution; the computational effort required to solve such a problem, by the best numerical methods known, grows prohibitively fast with the dimensions of the problem and the number of accuracy digits. Moreover, there are serious theoretical reasons to conjecture that this is an intrinsic feature of nonconvex problems rather than a drawback of the existing optimization techniques.*

As an example, consider the pair of optimization problems (A) and (B). The first is the nonconvex problem

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{n} x_i \\ \text{subject to} & \\ & \begin{array}{rcl} x_i^2 - x_i & = & 0, \;\; i = 1, \dots, n; \\ x_i x_j & = & 0 \;\;\; \forall (i,j) \in \Gamma, \end{array} \end{array} \tag{A}$$

where Γ is a given set of pairs (i, j) of indices i, j. This is a fundamental combinatorial problem of computing the *stability number* of a graph. It arises, e.g., in the following *channel communication problem*:

There is an alphabet of n letters a_i, $i = 1, 2, \dots, n$, say, the 256 usual bytes. A letter a_i can be sent through a communication channel; when passing through it, it either remains unchanged or can be converted to another letter a_j due to transmission errors. The errors are assumed to be "symmetric" (if a_i can be converted into a_j, then a_j can be converted into a_i as well), and Γ is the set of (indices of) those pairs of letters that can be converted from one into another. Assume that we are interested in a "nonconfusing" communication, where the addressee either gets a correct letter or is able to conclude that a transmission error has occurred but never misreads the input letter. In this case we should restrict the subalphabet S to be *independent*, meaning that no two distinct letters from S can be converted from one to another by the channel. To get the most from the channel, we would like to use an independent subalphabet of maximal cardinality. It turns out that the optimal value in (A) is exactly the cardinality of such a maximal independent subalphabet.

The second problem is the convex program

$$
\begin{array}{l}
\text{minimize } x_0 \\
\text{subject to} \\
\qquad \begin{array}{rcl}
\lambda_{\min}\begin{pmatrix} x_1 & & & x_1^\ell \\ & \ddots & & \vdots \\ & & x_m & x_m^\ell \\ x_1^\ell & \cdots & x_m^\ell & x_0 \end{pmatrix} & \geq & 0,\ \ell = 1, \ldots, k, \\
\displaystyle\sum_{j=1}^{m} a_j x_j^\ell & = & b^\ell,\ \ \ell = 1, \ldots, k, \\
\displaystyle\sum_{j=1}^{m} x_j & = & 1,
\end{array}
\end{array}
\tag{B}
$$

where $\lambda_{\min}(A)$ denotes the minimum eigenvalue of a symmetric matrix A. This problem originates from design of a *truss* (a mechanical construction built from thin elastic bars, like an electric mast, a bridge, or the Eiffel Tower) able to withstand k nonsimultaneous loads.

Looking at the analytical forms of problems (A) and (B), it seems that the first problem is easier than the second: the constraints in (A) are simple explicit quadratic equations, while the constraints in (B) involve much more complicated functions of the design variables—the eigenvalues of certain matrices depending on the design vector. The truth, however, is that the first problem is in a sense as difficult as an optimization problem can be, and the worst-case computational effort to solve it within absolute inaccuracy 0.5 is about 2^n operations for all known optimization methods. For $n = 256$ ("alphabet of bytes"), the complexity $2^n \approx 10^{77}$ is, for all practical purposes, the same as $+\infty$. In contrast to this, the second problem is quite computationally tractable. For example, for $k = 6$ (six loading scenarios) and $m = 100$ (a 100-bar construction), the problem has 701 variables (2.7 times the number of variables in the byte version of (A)); however, it can be reliably solved within six accuracy digits in a couple of minutes. The dramatic difference in the computational effort required to solve (A) and (B) is due to the fact that (A) is a nonconvex optimization problem, while (B) is convex.

The above discussion explains the words *Convex Programming* in the title of our book. We now explain the word *modern*. The emphasis in the book is on *well-structured convex problems* such as linear programming (LP), conic quadratic programming (CQP), and semidefinite programming (SDP). These are the areas most affected by the recent progress in optimization, areas where we possess well-developed techniques for building large-scale models, analyzing them theoretically ("on paper") and processing them numerically. Except for LP, these areas did not exist 10 years ago; thus most users who could potentially benefit from recent developments are not aware of their existence, let alone their usefulness. In enlarging the scope of classical optimization (LP, general nonlinear programming) by introduction of CQP and SDP, new application areas were created. Examples include semidefinite relaxations of difficult combinatorial problems, linear matrix inequality–based techniques in control, and mathematical programming with uncertain data. These new applications create synergies between optimization, computer science, and engineering, with potentially far-reaching consequences for our ability to solve complex real-life decision problems.

At this point, we want to address experts in optimization rather than general readers. The history of convex programming, as far as applied aspects are concerned, started with the invention of LP (Dantzig, circa 1948). LPs possess the simplest possible and transparent analytical structure, which from the beginning was heavily exploited both theoretically (the completely algorithmic LP duality theory) and computationally (the simplex method). With subsequent nonlinear extensions, the focus was shifted, in one giant step, from the simplest possible linear structure to the most general one, where all we know about the entities occurring in (P) is that they are called f_i, $i = 0, 1, \dots, m$, and X, that they are convex, and that f_i are $0/1/2/\dots$ times continuously differentiable. At the theoretical level, the focus on this general setting yielded very deep results in convex analysis (Lagrange duality, Karush–Kuhn–Tucker (KKT) optimality conditions, etc.); however, the price paid for these fundamental achievements was a lack of an *algorithmic* content of the subject. For example, the Lagrange dual of a general-type convex program (P) is something that exists and possesses very nice properties; this "something," however, normally cannot be written explicitly (in sharp contrast with the "fully algorithmic" LP duality). At the computational level, the emphasis on generality resulted in general-purpose "near-sighted" (and thus slow, as compared to LP algorithms) optimization methods, those utilizing purely local information on the problem.

To some extent, recent trends (the last decade or so) in convex optimization stem from the realization that there is something between the relatively narrow LP and the completely unstructured universe of convex programming; what is between are well-structured generic convex optimization problems like CQP and SDP. Needless to say, interest in special cases is not a complete novelty for our area (recall linearly constrained convex quadratic and geometric programming); what is a novelty is the recent emphasis on well-structured generic problems, along with outstanding achievements resulting from this emphasis. The most remarkable of these achievements is, of course, the interior point revolution, which has extended dramatically our abilities to process convex programs numerically, while creating a completely new, challenging, and attractive area of theoretical research. The emphasis on well-structured special cases has, however, another, less-evident (perhaps not less-important) consequence, which can be summarized as follows:

• *When restricted to "nice" generic convex programs, like LP, CQP, and SDP, convex analysis becomes an algorithmic calculus—as algorithmic as in the LP case*. For example, the SDP duality is as explicit and symmetric as the LP one. In fact, the same can be said about all other basic operations of convex analysis, from the simplest (like taking intersections and affine images of convex sets, or sums of convex functions) to the more sophisticated ones (like passing from a convex set to its polar or from a convex function to its Legendre transformation). Whenever the operands of such a construction can be represented, in a properly defined precise sense, via, say, SDP, the same is true for the resulting entity, and the SDP representation of the result is readily given by those of the operands.

As a result,

• *An instance of a nice generic convex problem can be processed, up to a certain point, on paper (and in a routine fashion). In many cases this allows one to obtain important qualitative information on the problem or to convert it into an equivalent one, better suited for subsequent numerical processing.* For example, applying SDP duality, one can reduce dramatically the design dimension of the truss topology design (TTD) problem and, as a result, increase by orders of magnitude the sizes of the TTD problems that can be processed numerically in practice.

Moreover,

• *nice generic convex problems, like CQP and especially SDP, possess vast expressive abilities*, which allow one to utilize the above advantages in a very wide spectrum of applications, much wider than the one covered by LP.

When writing this book, our major concern was to emphasize the issues just raised, and this emphasis is perhaps the most characteristic (and, hopefully, to some extent novel) feature of the book.

Restricting ourselves to well-structured convex optimization problems, it becomes logical to skip a significant part of what is traditionally thought of as the theory and algorithms of mathematical programming. Readers interested in the gradient descent, quasi-Newton methods, and even sequential quadratic programming, are kindly advised to use the excellent existing textbooks on these important subjects; our book should be thought of as a self-contained complement to, and not as an extended version of, these textbooks. We even have dared to omit the KKT optimality conditions in their standard form, since they are too general to be algorithmic; the role of the KKT conditions in our exposition is played by their particular (and truly algorithmic) case, expressed by the so-called conic duality theorem.

The book is addressed primarily to potential users (mainly engineers). Consequently, our emphasis is on building and processing instructive engineering models rather than on describing the details of optimization algorithms. The underlying motivation is twofold. First, we wish to convince engineers that optimization indeed has something to do with their professional interests. Second, we believe that a crucial necessary condition for successful practical applications of optimization is understanding what is desirable and what should be avoided at the *modeling* stage. Thus, important questions to be addressed are (a) *What optimization models can be efficiently processed* (to a certain point on paper and then on a computer)? and (b) *How one can convert* (provided that it is possible) *a seemingly bad initial description of a problem into a tractable and well-structured optimization model*?

We believe that the best way to provide relevant insight for potential users of optimization is to present, along with general concepts and techniques, many applications of these concepts and techniques. We believe that the presentation of *optimization algorithms* in a user-oriented book should be as nontechnical as possible (to drive a car, no knowledge of engines is necessary). The section devoted to algorithms presents the ellipsoid method (due to its simplicity, combined with its capability to answer affirmatively the fundamental question of whether convex programming is computationally tractable) and an overview of polynomial-time interior-point methods for LP, CQP, and SDP.

Although the book is user oriented, it is a mathematical book. Our goal is to demonstrate that when processing “meaningful” *mathematical* models by *rigorous* mathematical methods (not by their engineering surrogates), one can obtain results that have meaningful and instructive engineering interpretation. Whether we have reached this goal, is another story; this judgment rests upon the reader.

Last, but not least, a reader should keep in mind that what follows are *lecture notes*; our intention is to highlight those issues that we find most interesting and instructive, rather than to present a complete overview of convex optimization. Consequently, we are ready to take the blame for being boring or unclear or for focusing on issues of minor importance, in any material in the book. However, we do not accept "money back" requests based on claims that something (however important) is *not* included. Along with immunity with regard to what is absent, a lecture notes–type book provides us with some other privileges, like a style which is a bit more vivid compared to the academic standards, and a rather short list of bibliography references (embedded as footnotes in the body of the text). In this latter respect, a reader should be aware that if a statement appears in the text without a reference, this does not mean that we are claiming authorship; it merely reflects that our focus is on the state of convex optimization rather than on its history.

Audience and prerequisites. Formally, readers should know basic facts from linear algebra and analysis—those presented in standard undergraduate mathematical courses for engineers. For optimization-related areas, readers are assumed to know not much more than the definitions of a convex set and a convex function and to have heard (no more than that!) about mathematical programming problems. Informally, it is highly desirable that a reader is in possession of the basic elements of mathematical culture.

The exercises. Exercises accompanying each lecture form a significant part of the book. They are organized in small groups, each devoted to a specific topic related to the corresponding lecture. Typically, the task in an exercise is to prove something. Most of the exercises are not easy. The results stated by the exercises are used in the subsequent parts of the book in the same way as the statements presented and proved in the main body of the book; consequently, a reader is kindly asked to at least read all exercises carefully. The order of exercises is of primary importance: in many cases preceding exercises contain information and hints necessary to succeed in subsequent ones.

Acknowledgments. A significant part of the applications discussed in our book is taken from the papers of Prof. Stephen Boyd of Stanford University and his colleagues. We are greatly indebted to Prof. Boyd for providing us with access to this material and for stimulating discussions. Applications related to structural design were developed in tight collaboration with Prof. Jochem Zowe and Dr. Michael Kočvara of Erlangen University. Parts of the book were written when the authors were visiting the Statistics and Operations Research Department of the Technical University of Delft, and we are thankful to our hosts, Prof. Kees Roos and Prof. Tamas Terlaky.

Aharon Ben-Tal and Arkadi Nemirovski

August 2000, Haifa, Israel