Topics in Convex Optimization: Interior-Point Methods, Conic Duality and Approximations

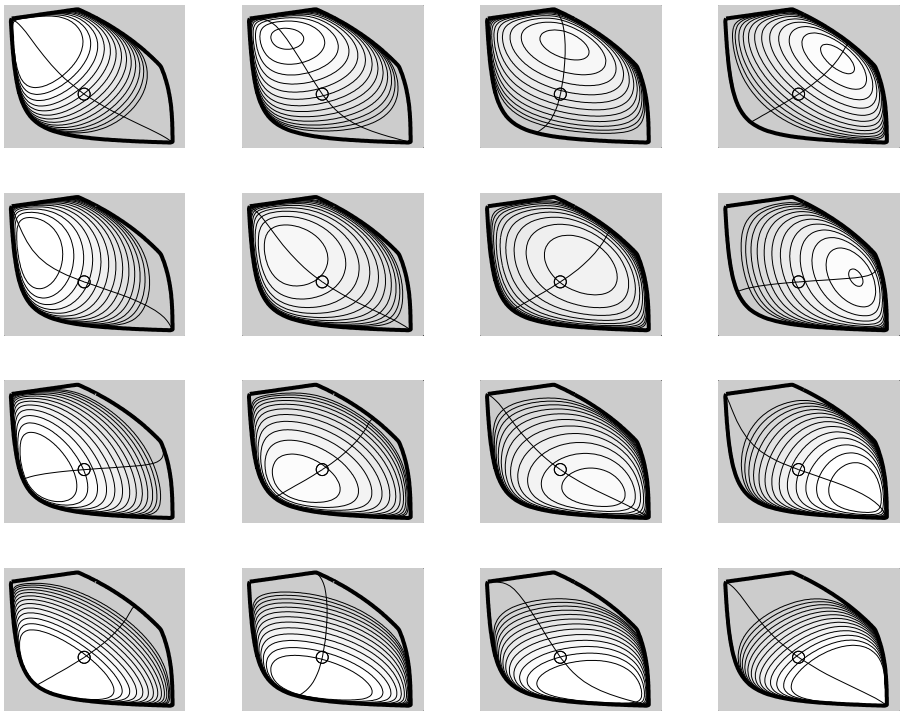
François Glineur

Topics in Convex Optimization: Interior-Point Methods, Conic Duality and Approximations

François Glineur

Service de Mathématique et de Recherche Opérationnelle,
Faculté Polytechnique de Mons,
Rue de Houdain, 9, B-7000 Mons, Belgium.

Francois.Glineur@fpms.ac.be
http://mathro.fpms.ac.be/~glineur/



January 2001

Co-directed by
Jacques Teghem
Tamás Terlaky

Contents

List of Figures

Preface

This work is dedicated to my wife, my parents and my grandfather,
for the love and support they gave me throughout the writing of this thesis.

First of all, I wish to thank my advisor Jacques Teghem, which understood early that the field of optimization would provide a stimulating and challenging area for my research. Both his guidance and support were crucial in the accomplishment of this doctoral degree. He also provided me with very valuable feedback during the final redaction of this thesis.

A great deal of the ideas presented in this thesis were originally developed during a research stay at the *Delft University of Technology* which took place in the first half of 1999. I am very grateful to Professors Kees Roos and Tamás Terlaky for their kind hospitality. They welcomed me in their Operations Research department, which provided me with a very stimulating research environment to work in.

Professor Tamás Terlaky accepted to co-direct this thesis. I wish to express him my deep gratitude for the numerous and fruitful discussions we had about my research. Many other researchers contributed directly or indirectly to my current understanding of optimization, sharing with me at various occasions their knowledge and insight about this field. Let me mention Professors Martine Labbé, Michel Goemans, Van Hien Nguyen, Jean-Jacques Strodiot and Philippe Toint, who made me discover some of the most interesting topics in optimization during my first year as doctoral student as well as Professor Yurii Nesterov, who was advisor in my thesis committee.

I also wish to express special thanks to the entire staff of the *Mathematics and Operations Research* department at the *Faculté Polytechnique de Mons*, for their constant kindness, availability and support.

I conducted this research as a research fellow supported by a grant from the *F.N.R.S.* (*Belgian National Fund for Scientific Research*), which also funded a trip to attend the International Mathematical Programming Symposium 2000 in Atlanta. My research stay at the *Delft University of Technology* was made possible with the aid of a travel grant awarded by the *Communauté Française de Belgique*, which also supported a trip to the INFORMS Spring 2000 conference in Salt Lake City.

Mons, December 2000.

Introduction

The main goal of *operations research* is to model real-life situations where some decisions have to be taken and help to identify the best one(s). One may for example want to choose between several available alternatives, tune numerical parameters in an engineering design or schedule the use of machines in a factory.

The concept of *best decision* depends of course on the problem considered and is not easy to define mathematically. The most common way to do this is to describe a decision as a set of parameters called *decision variables*, and try to minimize (or maximize) an *objective function* depending on these variables. This function may for example compute the cost associated to the decision. Moreover, we are most of the time in a situation where some combinations of parameters are not allowed (e.g. physical dimensions cannot be negative, a system must satisfy some performance requirements, ...), which leads us to consider a set of constraints acting on the decision variables.

Optimization is the field of mathematics whose goal is to minimize or maximize an objective function depending on several decision variables under a set of constraints. The main topic of thesis is a special category of optimization problems called *convex optimization*[1].

Why convex optimization ?

A fundamental difficulty in optimization is that it is not possible to solve all problems efficiently. Indeed, it is shown in [Nes96] that a hypothetical method that would be able to

[1] This class of problems is sometimes called *convex programming* in the literature. However, following other authors [RTV97, Ren00], we prefer to use the more natural word "optimization" since the term "programming" is nowadays strongly connected to computer science. The same treatment will be applied to the other classes of problems that will be considered in this thesis, such as linear optimization, geometric optimization, etc.

handle all optimization problems would require at least 10^{20} operations to solve with 1% accuracy some problems involving only 10 variables. There are basically two fundamentally different ways to react to this distressing fact:

a. Ignore it, i.e. design a method that can potentially solve all problems. Because of the above-mentioned result, it will be slow (or fail) on some problems, but hopefully will be efficient on most real-world problems we are interested in. This is the approach that generally prevails in the field of *nonlinear optimization*.

b. Restrict the set of problems that the method is supposed to solve. The goal is then to design a provably efficient method that is able to solve this restricted class of problems. This is for example the approach taken in *linear optimization*, where one requires the objective function and the constraints to be linear.

Each of these two approaches has its advantages and drawbacks. The major advantage of the first approach is its potentially very wide applicability, but this is counterbalanced by a less efficient analysis of the behaviour of the corresponding algorithms. In more technical terms, methods in first approach can usually only be proven to converge to an optimum (in some weak sense), while one can usually estimate the efficiency of methods designed for special categories of problems, i.e. bound the number of arithmetic operations they need to attain an optimum with a given accuracy. This is what led us to focus our research for this thesis on that second approach.

The next relevant question that has to be answered consists in asking ourselves which classes of problems we are going to study. It is rather clear that there is a tradeoff between generality and algorithmic efficiency: the more general your problem, the less efficient your methods. Linear optimization is in this respect an extreme case: it is a very particular (yet useful) type of problem for which very efficient algorithms are available (see Chapter 1).

However, some problems simply cannot be formulated within the framework of linear programs, which led us to consider a much broader class of problems called *convex optimization*. Basically, a problem belongs to this category if its objective function is convex and its constraints define a feasible convex set. As we will see in Chapter 2, very effective methods are available to solve these problems.

Unfortunately, checking that a given optimization problem is convex is far from straightforward (and it might even be more difficult than solving the problem itself). We have therefore to consider problems that are designed in a way that guarantees them to be convex. This is done by using specific classes of objective functions and constraints, and is called *structured convex optimization*. This is the central topic of this thesis, which is treated in Chapters 3–8.

To conclude, we mention that although it is not possible to model all problems of interest with a convex formulation, one can do it in a surprisingly high number of situations, either directly or using a equivalent reformulation. The reward for the added work of formulating the problem as a structured convex optimization problem is the great efficiency of the methods that can be then applied to it.

Overview of the thesis

We give here a short introduction to the research work presented in this thesis, which consists in three parts (we however refer the reader to the abstract and the introductory section placed at the beginning of each chapter for more detailed comments).

a. **Interior-point methods**. This first part deals with algorithms. We start with the case of linear optimization, for which an efficient method is known since the end of the fifties: the simplex method [Dan63]. However, another class of algorithms that could rival the simplex method was introduced in 1984 [Kar84]: the so-called *interior-point methods*, which are surveyed in Chapter 1 (this Chapter was published in [Gli98a], which is a translated and reworked version of [Gli97]). These methods can be generalized to handle any type of convex problems, provided a suitable barrier function is known. This is the topic of Chapter 2 [Gli00d], which gives a self-contained overview of the theory of self-concordant barriers for structured convex optimization [NN94].

b. **Conic duality**. The second part of this thesis is devoted to the study of duality issues for several classes of convex optimization problems. We first present in Chapter 3 conic optimization, a framework to describe convex optimization problems based on the use of convex cones. Convex problems expressed in this fashion feature a very symmetric duality theory, which is also presented in this Chapter. This setting is used in Chapters 4 [GT00] and 5 [Gli99], where we describe and study two classes of structured convex optimization problems known as l_p-norm optimization and geometric optimization.

   The approach used in these two chapters is very similar: we first define a suitable convex cone that allows us to express our problem with a conic formulation. The properties of this cone are then studied, which allows us to formulate the dual problem. One can then apply the conic duality theory described in Chapter 3 to give simplified proofs of all the duality properties that relate these primal and dual problems. Chapter 4 also presents a polynomial-time algorithm for l_p-norm optimization using a suitable self-concordant barrier and the results of Chapter 2.

   Despite some similarities, the convex cones introduced in Chapters 4 and 5 do not share the same structure. The goal of Chapter 6 [Gli00b] is to provide a different convex cone for geometric optimization that is more amenable to a common generalization with the cone for l_p-norm optimization presented in Chapter 4. This generalization is the topic of Chapter 7, which presents a very large class of so-called *separable* convex cones that unifies our formulations for geometric and l_p-norm optimization, as well as allowing the modelling of several others classes of convex problems.

c. **Approximations**. The last part of this thesis deals with various approximations of convex problems. Chapter 8 [Gli00a] uncovers an additional connection between geometric and l_p-norm optimization by showing that the former can be approximated by the latter. Basically, we are able to associate to a geometric optimization problem a family of l_p-norm optimization problems whose optimum solutions tend to the optimal solution of the original geometric problem. This also allows us to derive the duality properties of geometric optimization in a different way. Finally, Chapter 9 [Gli00c] presents computational experiments conducted with the polyhedral approximation of

the second-order cone presented in [BTN98]. This leads to a linearizing scheme that allows any second-order cone problem to be solved up to an arbitrary accuracy using linear optimization.

Part I

Interior-Point Methods

CHAPTER 1

Interior-point methods for linear optimization: a guided tour

The purpose of *mathematical optimization* is to minimize (or maximize) a function of several variables under a set of constraints. This is a very important problem arising in many real-world situations (e.g. cost or duration minimization).

When the function to optimize and its associated set of constraints are linear, we talk about *linear optimization*. The *simplex algorithm*, first developed by Dantzig in 1947, is a very efficient method to solve this class of problems [Dan63]. It has been thoroughly studied and improved since its first appearance, and is now widely used in commercial software to solve a great variety of problems (production planning, transportation, scheduling, etc.).

However, Karmarkar introduced in 1984 a new class of methods: the so-called *interior-point methods* [Kar84]. Most of the ideas underlying these new methods originate from the nonlinear optimization domain. These methods are both theoretically and practically efficient, can be used to solve large-scale problems and can be generalized to other types of convex optimization problems.

The purpose of this chapter is to give an overview of this rather new domain, providing a clear and understandable description of these methods, both from a theoretical and a practical point of view. This will provide a basis for the following chapters, which will present our contributions to the field.

1.1 Introduction

In this section, we present the standard formulations of a linear program and give a brief overview of the main differences between the simplex method, the traditional approach to solve these problems, and the recently developed class of interior-point methods, as well as a short historical account.

1.1.1 Linear optimization

The purpose of linear optimization is to optimize a linear objective function f depending on n decision variables under a set of linear (equality or inequality) constraints, which can be mathematically stated as (using matrix notation)

$$\min_{x \in \mathbb{R}^n} f(x) = c^T x \quad \text{s.t.} \quad \begin{cases} A_e x = b_e \\ A_i x \geq b_i \end{cases}, \tag{1.1}$$

where vector x contains the n decision variables, vector c defines the objective function and pairs (A_e, b_e) and (A_i, b_i) define the m_e equality and m_i inequality constraints. Column vectors x and c have size n, column vectors b_e and b_i have size m_e and m_i and matrices A_e and A_i have dimensions $m_e \times n$ and $m_i \times n$.

Many linear programs have simpler inequality constraints, e.g. nonnegativity constraints ($x \geq 0$) or bound constraints ($l \leq x \leq u$). The linear optimization *standard form* is a special case of linear program used for most theoretical developments of interior-point methods:

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{s.t.} \quad \begin{cases} Ax = b \\ x \geq 0 \end{cases}. \tag{1.2}$$

The only inequality constraints in this format are nonnegativity constraints for *all* variables, i.e. there are no *free* variables (we have thus that m_i is equal to n, A_i is the identity matrix and b_i is the null vector). It is furthermore possible to show that every linear program in the general form (1.1) admits an equivalent program in the standard form, obtainable by adding/removing variables/constraints (by *equivalent problem*, we mean that solving the transformed problem allows us to find the solution of the original one).

1.1.2 The simplex method

The set of all x satisfying the constraints in (1.2) is a polyhedron in $\mathbb{R}^n$. Since the objective is linear, parallel hyperplanes orthogonal to c are constant-cost sets and the optimal solution must be at one of the vertices of the polyhedron (it is also possible that a whole face of the polyhedron is optimal or that no solution exists, either because the constraints defining the polyhedron are inconsistent or because it is unbounded in the direction of the objective function).

The main idea behind the *simplex method* is to explore these vertices in an iterative way, moving from the current vertex to an adjacent one that improves the objective function value.

This is done using an algebraic characterization of a vertex called a *basis*. When such a move becomes impossible to make, the algorithm stops. Dantzig proved that this always happens after a finite number of moves, and that the resulting vertex is optimal [Dan63].

1.1.3 A first glimpse on interior-point methods

We are now able to give a first description of interior-point methods. As opposed to the simplex method which uses vertices, these methods start with a point that lies *inside* the set of feasible solutions. Using the standard form notation (1.2), we define the feasible set $\mathcal{P}$ to be the set of vectors x satisfying the constraints, i.e.

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax = b \text{ and } x \geq 0\} \ ,$$

and the associated set $\mathcal{P}^+$ to be the subset of $\mathcal{P}$ satisfying strict nonnegativity constraints

$$\mathcal{P}^+ = \{x \in \mathbb{R}^n \mid Ax = b \text{ and } x > 0\} \ .$$

$\mathcal{P}^+$ is called the *strictly feasible* set[1] and its elements are called *strictly feasible* points.

Interior-point methods are *iterative* methods that compute a sequence of iterates belonging to $\mathcal{P}^+$ and converging to an optimal solution. This is completely different from the simplex method, where an *exact* optimal solution is obtained after a finite number of steps. Interior-point iterates tend to an optimal solution but never attain it (since the optimal solutions do not belong to $\mathcal{P}^+$ but to $\mathcal{P} \setminus \mathcal{P}^+$). This apparent drawback is not really serious since

- ◇ Most of the time, an approximate solution (with e.g. 10^{-8} relative accuracy) is sufficient for most purposes.
- ◇ A *rounding* procedure can convert a nearly optimal interior point into an exact optimal vertex solution (see e.g. [RTV97]).

Another significant difference occurs when an entire face of $\mathcal{P}$ is optimal: interior-point methods converge to the interior of that face while the simplex method ends on one of its vertices.

The last difference we would like to point out at this stage is about algorithmic complexity. While the simplex method may potentially make a number of moves that grows exponentially with the problem size [KM72], interior-point methods need a number of iterations that is polynomially bounded by the problem size to attain a given accuracy. This property is with no doubt mainly responsible for the huge amount of research that has been carried out on the topic of interior-point methods for linear optimization.

1.1.4 A short historical account

The purpose of this paragraph is not to be exhaustive but rather to give some important milestones in the development of interior-point methods.

[1] $\mathcal{P}^+$ is in fact the relative interior of $\mathcal{P}$, see [Roc70a].

First steps of linear optimization.

1930–1940. First appearance of linear optimization formulations.

1939–1945. Second World War: operations research makes its debuts with military applications.

1947. Georges B. Dantzig publishes the first article about the simplex method for linear optimization [Dan63].

1970. V. Klee and G. Minty prove that the simplex method has exponential worst-case complexity [KM72].

First steps of interior-point methods.

1955. K. R. Frisch proposes a *barrier* method to solve nonlinear programs [Fri55].

1967. P. Huard introduces the *method of centers* to solve problems with nonlinear constraints [Hua67].

1968. A. V. Fiacco and G. P. McCormick develop barrier methods for convex nonlinear optimization [FM68].

1978. L. G. Khachiyan applies the *ellipsoid* method (developed by N. Shor in 1970 [Sho70]) to linear optimization and proves that it is polynomial [Kha79].

It is important to note that these barrier methods were developed as methods for nonlinear optimization. Although they are applicable to linear optimization, their authors do not consider them as viable competitors to the simplex method. We also point out that the complexity advantage of the ellipsoid method over the simplex algorithm is only of theoretical value, since the ellipsoid method turns out to be very slow in practice[2].

The interior-point revolution.

1984. N. Karmarkar discovers a polynomial interior-point method that is practically more efficient than the ellipsoid method. He also claims superior performance compared to the simplex method [Kar84].

1994. Y. Nesterov and A. Nemirovski publish a monograph on polynomial interior-point methods for convex optimization [NN94].

2000. Since Karmarkar's first breakthrough, more than 3000 articles have been published on the topic of interior point methods. A few textbooks have been published (see e.g. [Wri97, RTV97, Ye97]). Research is now concentrating on nonlinear optimization, especially on convex optimization.

Karmarkar's algorithm was not competitive with the best simplex implementations, especially on small-scale problems, but his announcement concentrated a stream of research on the topic.

[2]The simplex method only shows an exponential complexity on some hand-crafted linear programs and is much faster on real-world problems, while the ellipsoid method always achieves its worst-case polynomial number of iterations, which turns out to be slower than the simplex method.

We also point out that Khachiyan's method is not properly speaking the first polynomial algorithm for linear optimization, since Fiacco and McCormick's method has been shown *a posteriori* to be polynomial by Anstreicher [Ans90].

1.2 Building blocks

In this section, we are going to review the different concepts needed to get a correct understanding of interior-point methods. We start with the very well studied notion of duality for linear optimization (see e.g. [Sch86]).

1.2.1 Duality

Let us state again the standard form of a linear program

$$\min_{x\in\mathbb{R}^n} c^T x \quad \text{s.t.} \quad \left\{ \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \right. . \tag{LP}$$

Using the same data (viz. A, b and c) it is possible to describe another linear program

$$\max_{y\in\mathbb{R}^m} b^T y \quad \text{s.t.} \quad \left\{ \begin{array}{l} A^T y \leq c \\ y \text{ is free} \end{array} \right. . \tag{LD'}$$

As we will see later, this program is closely related to (LP) and is called the *dual* of LP (which will be called *primal* program). It is readily seen that this program may also be written as

$$\max_{y\in\mathbb{R}^m, s\in\mathbb{R}^n} b^T y \quad \text{s.t.} \quad \left\{ \begin{array}{l} A^T y + s = c \\ s \geq 0 \text{ and } y \text{ free} \end{array} \right. . \tag{LD}$$

This extra *slack* vector s will prove useful in simplifying our notation and we will therefore mainly use this formulation of the dual. We also define the dual feasible and strictly feasible sets $\mathcal{D}$ and $\mathcal{D}^+$ in a similar fashion to the sets $\mathcal{P}$ and $\mathcal{P}^+$

$$\begin{aligned} \mathcal{D} &= \left\{(y,s) \mid A^T y + s = c \text{ and } s \geq 0\right\} , \\ \mathcal{D}^+ &= \left\{(y,s) \mid A^T y + s = c \text{ and } s > 0\right\} . \end{aligned}$$

From now on, we will assume that matrix A has full row rank, i.e. that its rows are linearly independent[3]. Because of the equation $A^T y + s = c$, this implies a one-to-one correspondence between the y and s variables in the dual feasible set. In the following, we will thus refer to either (y, s), y or s as the dual variables.

We now state various important facts about duality:

[3]This is done without loss of generality: if a row of A is linearly dependent on some other rows, we have that the associated constraint is either redundant (and can be safely ignored) or impossible to satisfy (leading to an infeasible problem), depending on the value of the right-hand side vector b.

- ◇ If x is feasible for (LP) and (y, s) for (LD), we have $b^T y \leq c^T x$. This means that any feasible point of (LD) provides a lower bound for (LP) and that any feasible point of (LP) provides an upper bound for (LD). This is the *weak* duality property. The nonnegative quantity $c^T x - b^T y$ is called the *duality gap* and is equal to $x^T s$.
- ◇ x and (y, s) are optimal for (LP) and (LD) if and only if the duality gap is zero. This is the *strong* duality property. This implies that when both problems have optimal solutions, their objective values are equal. In that case, since $x^T s = 0$ and $x \geq 0$, $s \geq 0$, we have that all products $x_i s_i$ must be zero, i.e. at least one of x_i and s_i is zero for each i (this is known as *complementary slackness*).
- ◇ One of the following three situations occurs for problems (LP) and (LD)
  a. Both problems have finite optimal solutions.
  b. One problem is unbounded (i.e. its optimal value is infinite) and the other one is infeasible (i.e. its feasible set is empty). In fact, the weak duality property is easily seen to imply that the dual of an unbounded problem cannot have any feasible solution.
  c. Both problems are infeasible.

  This result is known as the *fundamental theorem of duality*.

Let us point out that it is possible to generalize most of these duality results to the class of convex optimization problems (see Chapter 3).

1.2.2 Optimality conditions

Karush-Kuhn-Tucker (KKT) conditions are necessary optimality conditions pertaining to nonlinear constrained optimization with a differentiable objective. Moreover, they are sufficient when the problem is convex, which is the case for linear optimization. For problem (LP) they lead to the following system

$$x \text{ is optimal for (LP)} \Leftrightarrow \exists\,(z,t) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} Ax & = & b \\ A^T z + t & = & c \\ x_i t_i & = & 0\ \forall i \\ x \text{ and } t & \geq & 0 \end{array} \right. . \qquad \text{(KKT)}$$

The second equation has exactly the same structure as the equality constraint for the dual problem (LD). Indeed, if we identify z with y and t with s we find

$$x \text{ is optimal for (LP)} \Leftrightarrow \exists\,(y,s) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} Ax & = & b \\ A^T y + s & = & c \\ x_i s_i & = & 0\ \forall i \\ x \text{ and } s & \geq & 0 \end{array} \right. .$$

Finally, using the definitions of $\mathcal{P}$ and $\mathcal{D}$ and the fact that when u and v are nonnegative

$$u_i v_i = 0\ \forall i \Leftrightarrow \sum_i u_i v_i = 0 \Leftrightarrow u^T v = 0$$

we have

$$x \text{ is optimal for (LP)} \;\Leftrightarrow\; \exists\,(y,s) \quad \text{s.t.} \quad \left\{ \begin{array}{rcl} x & \in & \mathcal{P} \\ (y,s) & \in & \mathcal{D} \\ x^T s & = & 0 \end{array} \right. .$$

This is in fact a confirmation of the strong duality theorem, revealing the deep connections between a problem and its dual: a necessary and sufficient condition for the optimality of a feasible primal solution is the existence of a feasible dual solution with zero duality gap (i.e. the same objective value).

Similarly, applying the KKT conditions to the dual problem would lead exactly to the same set of conditions, requiring the existence of a feasible primal solution with zero duality gap.

1.2.3 Newton's method

The fact that finding the optimal solution of a linear program is completely equivalent to solving the KKT conditions may suggest the use of a general method designed to solve systems of nonlinear equations[4]. The most popular of these methods is the *Newton's method*, whose principle is described in the following paragraph.

Let $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ be a differentiable nonlinear mapping. Newton's method is an iterative process aiming to find an $x \in \mathbb{R}^n$ such that $F(x) = 0$. For each iterate x_k, the method computes a first-order approximation to F around x_k and sets x_{k+1} to the zero of this linear approximation. Formally, if J is the Jacobian of F (assumed to be nonsingular), we have

$$F(x_k + \Delta x_k) \approx F(x_k) + J(x_k)\Delta x_k$$

and the Newton step Δx_k is chosen such that this linear approximation is equal to zero: we let thus $x_{k+1} = x_k + \Delta x_k$ where[5] $\Delta x_k = -J(x_k)^{-1}F(x_k)$. Convergence to a solution is guaranteed if the initial iterate x_0 lies in a suitable neighbourhood of one of the zeros of F.

Newton's method is also applicable to minimization problems in the following way: let $g : \mathbb{R}^n \mapsto \mathbb{R}$ be a function to minimize. We form a second-order approximation to $g(x)$ around x_k, namely

$$g(x_k + \Delta x_k) \approx g(x_k) + \nabla g(x_k)^T \Delta x_k + \frac{1}{2}\Delta {x_k}^T \nabla^2 g(x_k)\Delta x_k \; .$$

If the Hessian $\nabla^2 g(x_k)$ is positive definite, which happens when g is strictly convex, this approximation has a unique minimizer, which we take as next iterate. It is defined by $\Delta x_k = -\nabla^2 g(x_k)^{-1}\nabla g(x_k)$, which leads to a method that is basically equivalent to applying Newton's method to the gradient-based optimality condition $\nabla g(x) = 0$.

One problem with the application of Newton's method to the resolution of the KKT conditions is the nonnegativity constraints on x and s, which cannot directly be taken into

[4] Strictly speaking, the first two conditions are linear while only the $x_i s_i = 0$ equations are nonlinear. The nonnegativity constraints are not equations and cannot be handled by such a method.

[5] Computation of Δx_k is usually done with the linear system $J(x_k)\Delta x_k = -F(x_k)$ rather than computing explicitly $J(x_k)$'s inverse.

account via the mapping F. One way of incorporating these constraints is to use a *barrier* term, as described in the next paragraph.

1.2.4 Barrier function

A barrier function $\phi : \mathbb{R}^+ \mapsto \mathbb{R}$ is simply a differentiable function such that $\lim_{x\to 0^+} \phi(x) = +\infty$. Using such a barrier, it is possible to derive a parameterized family of unconstrained problems from an inequality-constrained problem in the following way

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{s.t.} \quad g_i(x) \geq 0 \; \forall i \tag{G}$$

$$\rightarrow \quad \min_{x\in\mathbb{R}^n} f(x) + \mu \sum_i \phi(g_i(x)) \; , \tag{G_μ}$$

where $\mu \in \mathbb{R}^+$. The purpose of the added barrier term is to drive the iterates generated by an unconstrained optimization method away from the infeasible zone (where one or more g_i's are negative). Of course, we should not expect the optimal solutions to (G_μ) to be equal to those of (G). In fact each value of μ gives rise to a different problem (G_μ) with its own optimal solutions.

However, if we solve a sequence of problems (G_μ) with μ decreasing to zero, we might expect the sequence of optimal solutions we obtain to converge to the optimum of the original problem (G), since the impact of the barrier term is less and less significant compared to the real objective function. The advantage of this procedure is that each optimal solution in the sequence will satisfy the strict inequality constraints $g_i(x) > 0$, leading to a feasible optimal solution to (G)[6].

The application of this technique to linear optimization will lead to a fundamental notion in interior-point methods: the *central path*.

1.2.5 The central path

Interior-point researchers use the following barrier function, called the *logarithmic barrier*:

$$\phi(x) = -\log(x) \; .$$

Using ϕ, let us apply a barrier term to the linear optimization problem (LP)

$$\min_{x\in\mathbb{R}^n} c^T x - \mu \sum_i \log(x_i) \quad \text{s.t.} \quad \left\{ \begin{array}{l} Ax = b \\ x > 0 \end{array} \right. \tag{P_μ}$$

and to its dual (LD) (since it is a maximization problem, we have to subtract the barrier term)

$$\max_{y\in\mathbb{R}^m} b^T y + \mu \sum_i \log(s_i) \quad \text{s.t.} \quad \left\{ \begin{array}{l} A^T y + s = c \\ s > 0 \text{ and } y \text{ free} \end{array} \right. \; . \tag{D_μ}$$

[6]The notion of barrier function was first investigated in [Fri55, FM68].

It is possible to prove (see e.g. [RTV97]) that both of these problems have unique optimal solutions x_μ and (y_μ, s_μ) for all $\mu > 0$ if and only if both $\mathcal{P}^+$ and $\mathcal{D}^+$ are nonempty[7]. In that case, we call the sets of optimal solutions $\{x_\mu \mid \mu > 0\} \subset \mathcal{P}^+$ and $\{(y_\mu, s_\mu) \mid \mu > 0\} \subset \mathcal{D}^+$ respectively the primal and dual central paths. These parametric curves have the following properties:

- ◇ The primal (resp. dual) objective value $c^T x$ (resp. $b^T y$) is monotonically decreasing (resp. increasing) along the primal (resp. dual) central path when $\mu \to 0$.
- ◇ The duality gap $c^T x_\mu - b^T y_\mu$ for the primal-dual solution (x_μ, y_μ, s_μ) is equal to $n\mu$. For this reason, μ will be called the *duality measure*. When a point (x, y, s) does not lie exactly on the central path, we can compute its *estimated* duality measure using $\mu = (c^T x - b^T y)/n$.
- ◇ The limit points $x_* = \lim_{\mu \to 0} x_\mu$ and $(y_*, s_*) = \lim_{\mu \to 0}(y_\mu, s_\mu)$ exist and hence are optimal solutions to problems (LP) and (LD) (because we have $c^T x_* - b^T y_* = 0$). Moreover, we have that $x_* + s_* > 0$, i.e. this optimal pair is strictly complementary[8].

1.2.6 Link between central path and KKT equations

To conclude this section we establish a link between the central path and the KKT equations. Applying the general KKT conditions to either problem (P_μ) or (D_μ) we find the following necessary and sufficient conditions

$$\left\{\begin{array}{rcl} Ax & = & b \\ A^T y + s & = & c \\ x_i s_i & = & \mu \; \forall i \\ x \text{ and } s & > & 0 \end{array}\right. \quad \Leftrightarrow \quad \left\{\begin{array}{rcl} x & \in & \mathcal{P}^+ \\ (y, s) & \in & \mathcal{D}^+ \\ x_i s_i & = & \mu \; \forall i \end{array}\right. . \qquad (\text{KKT}_\mu)$$

This system is very similar to the original KKT system, the only difference being the right-hand side of the third condition and the strict inequalities. This means in fact that the points on the central path satisfy a slightly perturbed version of the optimality KKT conditions for (LP) and (LD).

We now have all the tools we need to give a description of interior-point methods for linear optimization.

1.3 Interior-point algorithms

Since Karmarkar's breakthrough, many different interior-point methods have been developed. It is important to note that there exists in fact a whole collection of methods, sharing the same basic principles but whose individual characteristics may vary a lot.

[7] This condition is known as the *interior-point condition*.

[8] For optimal solutions (x, s) we always have $x_i s_i = 0$, i.e. at least one of x_i and s_i is zero. In the case of a strictly complementary solution, *exactly* one of x_i and s_i is zero.

Among the criteria that are commonly used to classify the methods, we have

- ◇ **Iterate space**. A method is said to be *primal*, *dual* or *primal-dual* when its iterates belong respectively to the primal space, the dual space or the Cartesian product of these spaces.

- ◇ **Type of iterate**. A method is said to be *feasible* when its iterates are feasible, i.e. satisfy both the equality and nonnegativity constraints. In the case of an *infeasible method*, the iterates need not satisfy the equality constraints, but are still required to satisfy the nonnegativity conditions.

- ◇ **Type of algorithm**. This is the main difference between the methods. Although the denominations are not yet fully standardized, we will distinguish *path-following* algorithms, *affine-scaling* algorithms and *potential reduction* algorithms. Sections 1.3.1, 1.3.2 and 1.3.3 will describe these three types of algorithms with more detail.

- ◇ **Type of step**. In order to preserve their polynomial complexity, some algorithms are obliged to take very small steps at each iteration, leading to a high total number of iterations when applied to practical problems[9]. These methods are called *short-step* methods and are mainly of theoretical interest. Therefore *long-step* methods, which are allowed to take much longer steps, have been developed and are the only methods used in practice.

It is not our purpose to give an exhaustive list of all the methods that have been developed up to now, but rather to present some representative algorithms, highlighting their underlying principles.

1.3.1 Path-following algorithms

We start with the most elegant category of methods, the path-following algorithms. As suggested by their denomination, the main idea behind these methods is to follow the central path up to its limit point. One could imagine the following naive conceptual algorithm (at this point, we want to keep generality and do not specify whether our method is primal, dual or primal-dual)

> **Given** an initial iterate v_0 and a sequence of duality measures monotonically decreasing to zero: $\mu_1 > \mu_2 > \mu_3 > \ldots > 0$ and $\lim_{k\to 0} \mu_k = 0$.
>
> **Repeat** for $k = 0, 1, 2, \ldots$
>
> Using v_k as starting point, compute v_{k+1}, the point on the central path with a duality measure equal to μ_{k+1}.
>
> **End**

[9] Please note that this is not in contradiction with the fact that this number of iterations is polynomially bounded by the size of the problem. This may simply mean that the polynomial coefficients are large.

It is clear from this scheme that v_k will tend to the limit point of the central path, which is an optimal solution to our problem.

However, the determination of a point on the central path requires the solution of a minimization problem like (P_μ) or the (KKT_μ) conditions, which potentially implies a lot of computational work. This is why path-following interior-point methods only try to compute points that are *approximately* on the central path, hopefully with much less computational work, and will thus only *loosely* follow the central path. Our conceptual algorithm becomes

> **Given** an initial iterate v_0 and a sequence of duality measures monotonically decreasing to zero: $\mu_1 > \mu_2 > \mu_3 > \ldots > 0$ and $\lim_{k\to 0} \mu_k = 0$.
>
> **Repeat** for $k = 0, 1, 2, \ldots$
>
> Using v_k as starting point, compute v_{k+1}, an approximation of the point on the central path with a duality measure equal to μ_{k+1}.
>
> **End**

The main task in proving the convergence and complexity of these methods will be to assess how well we approximate our targets on the central path (i.e. how close to the central path we stay).

Short-step primal-dual path-following algorithm

This specific algorithm is a primal-dual feasible method, which means that all the iterates lie in $\mathcal{P}^+ \times \mathcal{D}^+$. Let (x_k, y_k, s_k) be the current iterate with duality measure μ_k. We also suppose that this iterate is *close* to the point $(x_{\mu_k}, y_{\mu_k}, s_{\mu_k})$ on the central path. To compute the next iterate, we target $(x_{\mu_{k+1}}, y_{\mu_{k+1}}, s_{\mu_{k+1}})$, a point on the central path with a smaller duality measure μ_{k+1} (thus closer to the optimal limit point). The main two characteristics of the short-step method are

- ◇ The duality measure of the point we target is defined by $\mu_{k+1} = \sigma\mu_k$ where σ is a constant strictly between 0 and 1.
- ◇ The next iterate will be computed by applying *one single* Newton step to the perturbed primal-dual conditions ($KKT_{\sigma\mu_k}$) defining our target on the central path[10]

$$\left\{ \begin{array}{rcl} Ax & = & b \\ A^T y + s & = & c \\ x_i s_i & = & \sigma\mu_k \; \forall i \end{array} \right. . \tag{1.3}$$

Formally, we have presented Newton's method as a way to find a root of a function F and not as a way to solve a systems of equations, so that we have first to define a function whose roots are solution of the system (1.3). Indeed, considering

$$F_k : \mathbb{R}^{2n+m} \mapsto \mathbb{R}^{2n+m} : \begin{pmatrix} x_k \\ y_k \\ s_k \end{pmatrix} \mapsto \begin{pmatrix} Ax_k - b \\ A^T y_k + s_k - c \\ X_k S_k e - \sigma\mu_k e \end{pmatrix} ,$$

[10] Note that we have to ignore the nonnegativity conditions for the moment.

where e stands for the all-one vector and X_k and S_k are diagonal matrices made up with vectors x_k and s_k (these notations are standard in the field of interior-point methods), we find that the Newton step we take is defined by the following linear system

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X_k S_k e + \sigma \mu_k e \end{pmatrix} . \tag{1.4}$$

This leads to the following algorithm

Given an initial iterate $(x_0, y_0, s_0) \in \mathcal{P}^+ \times \mathcal{D}^+$ with duality measure μ_0 and a constant $0 < \sigma < 1$.

Repeat for $k = 0, 1, 2, \ldots$

Compute the Newton step $(\Delta x_k, \Delta y_k, \Delta s_k)$ using the linear system (1.4).

Let $(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + (\Delta x_k, \Delta y_k, \Delta s_k)$ and $\mu_{k+1} = \sigma \mu_k$.

End

We now sketch a proof of the correctness of this algorithm. For our path-following strategy to work, we have to ensure that our iterates (x_k, y_k, s_k) stay close to the points $(x_{\mu_k}, y_{\mu_k}, s_{\mu_k})$ on the central path, which guide us to an optimal solution. For this purpose we define a quantity that measures the proximity between a strictly feasible iterate $(x, y, s) \in \mathcal{P}^+ \times \mathcal{D}^+$ and the central point (x_μ, y_μ, s_μ). Since the main property of this central point is $x_i s_i = \mu \; \forall i$, which is equivalent to[11] $xs = \mu e$, the following measure (see e.g. [Wri97])

$$\delta(x, s, \mu) = \frac{1}{\mu} \|xs - \mu e\| = \left\| \frac{xs}{\mu} - e \right\|$$

seems adequate: it is zero if and only if (x, y, s) is equal to (x_μ, y_μ, s_μ) and increases as we move away from this central point. It is also interesting to note that the size of a neighbourhood defined by $\delta(x, s, \mu) < R$ decreases with μ, because of the leading term $\frac{1}{\mu}$.

Another possibility of proximity measure with the same properties is

$$\delta(x, s, \mu) = \frac{1}{2} \left\| \sqrt{\frac{xs}{\mu}} - \sqrt{\frac{\mu}{xs}} \right\|$$

where the square roots are taken componentwise (see [RTV97]).

The proof has the following steps [RTV97, Wri97]

a. **Strict Feasibility**. Prove that strict feasibility is preserved by the Newton step: if $(x_k, y_k, s_k) \in \mathcal{P}^+ \times \mathcal{D}^+$, we have $(x_{k+1}, y_{k+1}, s_{k+1}) \in \mathcal{P}^+ \times \mathcal{D}^+$. We have to be especially careful with the strict nonnegativity constraints, since they are not taken into account by Newton's method.

[11] xs denotes here the componentwise product of vectors x and s.

b. **Duality measure**. Prove that the target duality measure is attained after the Newton step: if (x_k, y_k, s_k) has a duality measure equal to μ_k, the next iterate $(x_{k+1}, y_{k+1}, s_{k+1})$ has a duality measure equal to $\sigma\mu_k$

c. **Proximity**. Prove that proximity to the central path targets is preserved: there is a constant τ such that if $\delta(x_k, s_k, \mu_k) < \tau$, we have $\delta(x_{k+1}, s_{k+1}, \mu_{k+1}) < \tau$ after the Newton step.

Adding the additional initial assumption that $\delta(x_0, s_0, \mu_0) < \tau$, this is enough to prove that the sequence of iterates will stay in a prescribed neighbourhood of the central path and will thus (approximately) converge to its limit point, which is a (strictly complementary) optimal solution. The last delicate question is to choose a suitable combination of constants σ and τ that allows us to prove the three statements above. For the first duality measure we presented the following values are acceptable (see [Wri97])

$$\sigma = 1 - \frac{0.4}{\sqrt{n}} \text{ and } \tau = 0.4 \,,$$

where n stands for the size of vectors x and s as usual, while for the second measure we may choose (see [RTV97])

$$\sigma = 1 - \frac{1}{2\sqrt{n}} \text{ and } \tau = \frac{1}{\sqrt{2}} \,.$$

To conclude this description, we specify how the algorithm terminates. Given an accuracy parameter ε, we stop our computations when the duality gap falls below ε, which happens when $n\mu_k < \varepsilon$. This guarantees that $c^T x$ and $b^T y$ approximate the true optimal objective value with an error smaller than ε. We now state this algorithm in its final form:

> **Given** an initial iterate $(x_0, y_0, s_0) \in \mathcal{P}^+ \times \mathcal{D}^+$ with duality measure μ_0, an accuracy parameter ε and suitable constants $0 < \sigma < 1$ and τ such that $\delta(x_0, y_0, s_0) < \tau$.
>
> **Repeat** for $k = 0, 1, 2, \ldots$
>
> Compute the Newton step $(\Delta x_k, \Delta y_k, \Delta s_k)$ using the linear system (1.4).
>
> Let $(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + (\Delta x_k, \Delta y_k, \Delta s_k)$ and $\mu_{k+1} = \sigma\mu_k$.
>
> **Until** $n\mu_{k+1} < \varepsilon$

Moreover, it is also possible to prove that in both cases, the solution with ε accuracy will be reached after a number of iterations N such that

$$N = O\left(\sqrt{n} \log \frac{n\mu_0}{\varepsilon}\right) \,. \tag{1.5}$$

This polynomial complexity bound on the number of iterations that varies like the square root of the problem size is the best attained so far for linear optimization.

However, it is important to note that values of σ presented above will always be in practice nearly equal to one, which means that the duality measures will decrease very slowly. Although its complexity is polynomial, this method requires a large number of iterations and is not very efficient from a practical point of view.

Dual short-step path-following methods

This second short-step method is very similar to the previous one but its iterates lie in the dual space $\mathcal{D}^+$. We keep the general principle of following the dual central path and targeting points (y_{μ_k}, s_{μ_k}) on it but we have to make the following adjustments[12]

- ◇ We cannot deduce the Newton step from the (KKT_μ) conditions any more, since they involve both primal and dual variables. We apply instead a single minimizing Newton step to the (D_μ) barrier problem, which gives the following $(n+m)\times(n+m)$ linear system
$$\begin{pmatrix} A^T & I \\ AS_k^{-2}A^T & 0 \end{pmatrix} \begin{pmatrix} \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{b}{\sigma\mu_k} - AS_k^{-1}e \end{pmatrix} . \tag{1.6}$$
- ◇ We have to modify our measure of proximity: we now define $\delta(s,\mu)$ with [RTV97]
$$\delta(s,\mu) = \min_x \{\delta(x,s,\mu) \mid Ax = b\} = \frac{1}{\mu} \min_x \{\|xs - \mu e\| \mid Ax = b\}$$
(we have that this measure is zero if and only if $s = s_\mu$).

Our algorithm simply becomes

> **Given** an initial iterate $(y_0, s_0) \in \mathcal{D}^+$ with duality measure μ_0, an accuracy parameter ε and suitable constants $0 < \sigma < 1$ and τ such that $\delta(y_0, s_0) < \tau$.
>
> **Repeat** for $k = 0, 1, 2, \ldots$
>
> Compute the Newton step $(\Delta y_k, \Delta s_k)$ using the linear system (1.6).
>
> Let $(y_{k+1}, s_{k+1}) = (y_k, s_k) + (\Delta y_k, \Delta s_k)$ and $\mu_{k+1} = \sigma\mu_k$.
>
> **Until** $n\mu_{k+1} < \varepsilon$

In this case we may for example choose
$$\sigma = 1 - \frac{1}{3\sqrt{n}} \text{ and } \tau = \frac{1}{\sqrt{2}} ,$$
which leads to the same complexity bound (1.5) for the total number of iterations.

Primal-dual long-step path-following methods

The long-step primal-dual method we are going to describe now is an attempt to overcome the main limitation of the short-step methods: their very small step size. As presented above, the fundamental reason for this slow progress is the value of σ that has to be chosen nearly equal to one in order to prove the polynomial complexity of the method.

[12] It is of course also possible to design a primal short-step path-following method in a completely similar fashion.

A simple idea to accelerate the method would simply be to decrease the duality measure more aggressively, i.e. still using $\mu_{k+1} = \sigma\mu_k$ but with a lower σ. However, this apparently small change breaks down the good properties we were able to prove for the short-step algorithms. Indeed, if our target on the central path is too far from our current iterate, we may have that

- ◇ The Newton step computed by (1.4) is no longer feasible. The reason for that is easy to understand. Newton's method is asked to solve the (KKT_μ) system, which is made of two linear equations and one mildly nonlinear equation. Because of this third equation, the linear system we solve is only an approximation of the real set of equations, and the further we are from the solution we target, the less accurate this approximation is. When our target is located too far away, the linear approximation becomes so bad that barrier term does not play its role and the Newton step jumps out of the feasible region by violating the nonnegativity constraints[13] $x > 0$ and $s > 0$.

  Since the iterates of an interior-point method must always satisfy the strict nonnegativity conditions, we have to take a so-called *damped* Newton step, i.e. reduce it with a factor $\alpha_k < 1$ in order to make it stay within the strictly feasible region $\mathcal{P}^+ \times \mathcal{D}^+$:

$$(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + \alpha_k(\Delta x_k, \Delta y_k, \Delta s_k) \ .$$

- ◇ This damping of the Newton step cancels the property that the duality measure we target is attained. It is indeed possible to show that the duality measure after a damped Newton step becomes $(1-\alpha_k(1-\sigma))\mu_k$, which varies linearly between μ_k and $\sigma\mu_k$ when α decreases from 1 to 0.

  There is unfortunately no way to circumvent this drawback, and we have to accept that our iterates never exactly achieve the targeted duality measures, unless a full Newton step is taken.

- ◇ We cannot guarantee that a single Newton step will keep the proximity to the central path in the sense of $\delta(x, s, \mu) < \tau$, for the same reasons as above (nonlinearity). In the long-step strategy we describe, we take several Newton steps with the same target duality measure until proximity to the central path is restored. Then we may choose another target and decrease μ.

Our long-step method may be described in the following way:

Given an initial iterate $(x_0, y_0, s_0) \in \mathcal{P}^+ \times \mathcal{D}^+$, an initial duality measure μ_0, an accuracy parameter ε and suitable constants $0 < \sigma < 1$ and τ such that $\delta(x_0, y_0, s_0) < \tau$.

Repeat for $k = 0, 1, 2, \ldots$

Compute the Newton step $(\Delta x_k, \Delta y_k, \Delta s_k)$ using the linear system (1.4).

Let $(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + \alpha_k(\Delta x_k, \Delta y_k, \Delta s_k)$ with a step length α_k chosen such that $(x_{k+1}, y_{k+1}, s_{k+1}) \in \mathcal{P}^+ \times \mathcal{D}^+$.

[13] Note that since the first two conditions $Ax = b$ and $A^T y + s = c$ are linear, they are always fulfilled after the Newton step.

If $\delta(x_{k+1}, s_{k+1}, \sigma\mu_k) < \tau$ **Then** let $\mu_{k+1} = \sigma\mu_k$ **Else** let $\mu_{k+1} = \mu_k$.
Until $n\mu_{k+1} < \varepsilon$

As opposed to the complexity analysis of the short-step method, we may choose here whatever value we want for the constant σ, in particular values much smaller than 1. It is the choice of τ and α_k that makes the method polynomial. The main task is here to analyse the number of iterations that is needed to restore proximity to the central path. Taking for σ a constant independent of n (like .5, .1 or .01), it is possible to prove that suitable choices of τ and α_k lead to the following number of iterations

$$N = O\left(n \log \frac{n\mu_0}{\varepsilon}\right) .$$

Let us point out an odd fact: although this method takes longer steps and is practically more efficient than the short-step methods, its theoretical complexity is worse than the short-step complexity (1.5).

1.3.2 Affine-scaling algorithms

The intensive stream of research on the topic of interior-point methods for linear optimization was triggered by Karmarkar's seminal article [Kar84]. His method used projective transformations and was not described in terms of central path or Newton's method. Later, researchers simplified this algorithm, removing the need for projective transformations, and obtained a class of methods called affine-scaling algorithms. It was later discovered that these methods had been previously proposed by Dikin in Russia, 17 years before Karmarkar [Dik67].

Affine-scaling algorithms do not explicitly follow the central path and do not even refer to it. The basic idea underlying these methods is the following: consider for example the primal problem (LP)

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{s.t.} \quad \left\{ \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \right. . \tag{LP}$$

This problem is hard to solve because of the nonnegativity constraints, which give the feasible region a polyhedral shape. Let us consider the current iterate x_k and replace the polyhedral feasible region by an inscribed ellipsoid centered at x_k. The idea is to minimize the objective on this ellipsoid, which should be easier than on a polyhedron, and take this minimum as next iterate.

How do we construct an ellipsoid that is centered at x_k and inscribed into the feasible region ? Consider a positive diagonal matrix D. It is easy to show that problem ($\mathrm{P_D}$)

$$\min_{w \in \mathbb{R}^n} (Dc)^T w \quad \text{s.t.} \quad \left\{ \begin{array}{l} ADw = b \\ w \geq 0 \end{array} \right. \tag{$\mathrm{P_D}$}$$

is equivalent to (LP), the x variable being simply scaled by $x = Dw$ (this scaling operation is responsible for the denomination of the method). Choosing a special diagonal matrix $D = X_k$, which maps the current iterate x_k to e, we obtain the following problem

$$\min_{w \in \mathbb{R}^n} (X_k c)^T w \quad \text{s.t.} \quad \left\{ \begin{array}{l} AX_k w = b \\ w \geq 0 \end{array} \right. .$$

We are now able to restrict the feasible region defined by $w \geq 0$ to a unit ball centered at e, according to the inclusion $\{w \mid \|w - e\| \leq 1\} \subset \{w \mid w \geq 0\}$. Our problem becomes

$$\min_{w \in \mathbb{R}^n} (X_k c)^T w \quad \text{s.t.} \quad \begin{cases} AX_k w = b \\ \|w - e\| \leq 1 \end{cases},$$

i.e. the minimization of a linear objective over the intersection of a unit ball and an affine subspace, whose solution can be easily computed analytically via a linear system. Back in the original space, this is equivalent to

$$\min_{x \in \mathbb{R}^n} c^T x \quad \text{s.t.} \quad \begin{cases} Ax = b \\ \|X_k^{-1} x - e\| \leq 1 \end{cases},$$

whose feasible region is an ellipsoid centered at x_k. This ellipsoid is called the *Dikin* ellipsoid and lies entirely inside $\mathcal{P}$. The minimum over this ellipsoid is given by $x_k + \Delta x_k$, where[14]

$$\Delta x_k = -\frac{X_k P_{AX_k} X_k c}{\|P_{AX_k} X_k c\|}. \tag{1.7}$$

Because our ellipsoid lies entirely within the feasible region, the step Δx_k is feasible and the next iterate $x_k + \Delta x_k$ is expected to be closer to the optimal solution than x_k.

Short- and long-step primal affine-scaling algorithms

Introducing a constant ρ to reduce the step size, we may state our algorithm as

Given an initial iterate $x_0 \in \mathcal{P}^+$ and a constant $0 < \rho < 1$.
Repeat for $k = 0, 1, 2, \ldots$
Compute the affine scaling step Δ_k with (1.7) and let $x_{k+1} = x_k + \rho \Delta_k$.
End

This scheme is known as the short-step primal affine-scaling algorithm. Convergence to a primal solution has been proved for $\rho = \frac{1}{8}$, but we still do not know whether this method has polynomial complexity[15]. It is of course possible to design a dual and even a primal-dual variant of this method (all we have to do is to define the corresponding Dikin ellipsoids).

It is also possible to make the algorithm more efficient by taking longer steps, i.e. moving outside of the Dikin ellipsoid. Keeping the same direction as for the short-step method, the maximum step we can take without leaving the primal feasible region is given by

$$\Delta x_k = -\frac{X_k P_{AX_k} X_k c}{\max [P_{AX_k} X_k c]}, \tag{1.8}$$

where $\max[v]$ stands for the maximum component of vector v, which leads to the following algorithm:

[14] P_Q denotes the projection matrix onto $\operatorname{Ker} Q$, the null space of Q, which can be written as $P_Q = I - Q^T (Q Q^T)^{-1} Q$ when Q has maximal rank.

[15] When certain nondegeneracy conditions hold, convergence has been proved for $0 < \rho < 1$.

Given an initial iterate x_0 and a constant $0 < \lambda < 1$.

Repeat for $k = 0, 1, 2, \ldots$

Compute the affine scaling step Δ_k with (1.8) and let $x_{k+1} = x_k + \lambda\Delta_k$.

End

The constant λ decides which fraction of the way to the boundary of the feasible region we move[16]. Global convergence has been proved when $0 < \lambda \leq 2/3$ but a surprising counterexample has been found with $\lambda = 0.999$ (see [Mas93]). Finally, as for the short-step method, we do not know whether this method has polynomial complexity.

Link with path-following algorithms

There is an interesting and unexpected link between affine-scaling methods and path-following algorithms. Taking for example the definition (1.6) of the dual Newton step in the path-following framework and letting σ tend to zero, i.e. letting the target duality measure tend to zero, we find that the resulting limit direction is exactly equal to the dual affine-scaling direction ! This surprising fact, which is also valid for their primal counterparts, gives us some insight about both methods:

- ◇ The affine-scaling method can be seen as an application of Newton's method that is targeting the limit point of the central path, i.e. that tries to jump directly to an optimal solution without following the central path.

- ◇ Looking at (1.6), it is possible to decompose the dual Newton step into two parts:

$$\Delta x_k = \frac{1}{\sigma\mu_k}\Delta^a x_k + \Delta^c x_k \ ,$$

  where

$$\begin{pmatrix} A^T & I \\ AS_k^{-2}A^T & 0 \end{pmatrix}\begin{pmatrix} \Delta^a y_k \\ \Delta^a s_k \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix} \text{ and } \begin{pmatrix} A^T & I \\ AS_k^{-2}A^T & 0 \end{pmatrix}\begin{pmatrix} \Delta^c y_k \\ \Delta^c s_k \end{pmatrix} = \begin{pmatrix} 0 \\ -AS_k^{-1}e \end{pmatrix} \ .$$

  - $\Delta^a x_k$ is called the affine-scaling component. It has the same direction as the affine-scaling method and is only seeking optimality.
  - $\Delta^c x_k$ is called the centering component. It is targeting a point on the central path with the same duality measure as the current iterate, i.e. only tries to improve proximity to the central path.

  It is possible to show that most interior-point methods follow in fact directions that are combinations of these two basic directions.

[16] This constant has to be strictly less than 1 since we want to stay in the interior of the feasible region.

1.3.3 Potential reduction algorithms

Instead of targeting a decreasing sequence of duality measures, the method of Karmarkar made use of a potential function to monitor the progress of its iterates. A potential function is a way to measure the worth of an iterate. Its main two properties are the following:

- ◇ It should tend to $-\infty$ if and only if the iterates tend to optimality.
- ◇ It should tend to $+\infty$ when the iterates tend to the boundary of the feasible region without tending to an optimal solution[17].

The main goal of a potential reduction algorithm is simply to reduce the potential function by a fixed amount δ at each step, hence its name. Convergence follows directly from the first property above.

Primal-dual potential reduction algorithm

We are going to describe the application of this strategy in the primal-dual case. The Tanabe-Todd-Ye primal-dual potential function is defined on the strictly feasible primal-dual space $\mathcal{P}^+ \times \mathcal{D}^+$ by

$$\Phi_\rho(x,s) = \rho \log x^T s - \sum_i \log x_i s_i \ ,$$

where ρ is a constant required to be greater than n. We may rewrite it as

$$\Phi_\rho(x,s) = (\rho - n) \log x^T s - \sum_i \log \frac{x_i s_i}{x^T s/n} + n \log n$$

and note the following

- ◇ The first term makes the potential tend to $-\infty$ when (x,s) tends to optimality, since we have then the duality gap $x^T s$ tending to 0.
- ◇ The second term measures *centrality* of the iterate. A perfectly centered iterate will have all its products $x_i s_i$ equal to their average value $x^T s/n$, making the second term equal to zero. As soon these products become different, this term increases, and tends to $+\infty$ if one of the products $x_i s_i$ tends to zero without $x^T s$ tending also to zero (which means exactly that we approach the boundary of the feasible region without tending to an optimal solution).

The search direction for this method is not new: it is the same as for the path-following algorithm, defined with a target duality measure $n\mu_k/\rho$ (i.e. with $\sigma = n/\rho$). However, in this case, μ_k will not follow a predefined decreasing sequence, but will have to be recomputed after each step (since this algorithm cannot guarantee that the duality measure targeted by the Newton step will be attained). The algorithm proceeds as follows:

[17] We cannot of course simply prevent the method from approaching the boundary of the feasible region, since our optimal solution lies on it.

Given an initial iterate $(x_0, y_0, s_0) \in \mathcal{P}^+ \times \mathcal{D}^+$ with duality measure μ_0 and a constant $\rho > n$. Define $\sigma = n/\rho$.

Repeat for $k = 0, 1, 2, \ldots$

Compute the Newton step $(\Delta x_k, \Delta y_k, \Delta s_k)$ using the linear system (1.4).

Let $(x_{k+1}, y_{k+1}, s_{k+1}) = (x_k, y_k, s_k) + \alpha_k(\Delta x_k, \Delta y_k, \Delta s_k)$ where α_k is defined by

$$\begin{aligned} \alpha_k &= \arg\min_{\alpha} \Phi_\rho(x_k + \alpha\Delta x_k, s_k + \alpha\Delta s_k) \\ &\qquad \text{s.t.} \quad (x_k, y_k, s_k) + \alpha(\Delta x_k, \Delta y_k, \Delta s_k) \in \mathcal{P}^+ \times \mathcal{D}^+ \ . \end{aligned}$$

Evaluate μ_{k+1} with $(x_{k+1}^T s_{k+1})/n$.

Until $n\mu_{k+1} < \varepsilon$

The principle of this method is thus to minimize the potential function along the search direction at each iteration. The main task in analysing the complexity of this method is to prove that this step will provide at least a fixed reduction of Φ_ρ at each iteration. Using $\rho = n + \sqrt{n}$, it is possible to prove that $\Phi_\rho(x_{k+1}, s_{k+1}) \leq \Phi_\rho(x_k, s_k) - \delta$ with $\delta = 0.16$ (see e.g. [Ans96]), leading to a total number of iterations equal to

$$N = O\left(\sqrt{n} \log \frac{n\mu_0}{\varepsilon}\right) \ ,$$

matching the best complexity results for the path-following methods.

It is in general too costly for a practical algorithm to minimize exactly the potential function along the search direction, since Φ_ρ is a highly nonlinear function. We may use instead one the following strategies

- ◇ Define a quadratic approximation of Φ_ρ along the search direction and take its minimizer as next iterate.
- ◇ Take a fixed percentage (e.g. 95%) of the maximum step along the search direction staying inside of the feasible region.

We note however that polynomial complexity is no longer guaranteed in these cases.

1.4 Enhancements

In the following, we present various enhancements that are needed to make the theoretical methods of the previous section work in practice.

1.4.1 Infeasible algorithms

All the algorithms we have described up to now are feasible methods, which means they need a strictly feasible iterate as starting point. However, such a point is not always available:

- ◇ For some problems, a *natural* strictly feasible point is not directly available and finding one may be as difficult as solving the whole linear program.

- ◇ Some problems have no strictly feasible points although they are perfectly valid and have finite optimal solutions. This situation happens in fact if and only if the optimal solution set is not bounded[18].

We can think of two different strategies to handle such cases: embed the problem into a larger one that admits a strictly feasible starting point (this will be developed in the next paragraph) or modify the algorithm to make it work with infeasible iterates. We are now going to give an overview of this second strategy.

We recall that the iterates of an infeasible method do not satisfy the equality constraints $Ax = b$ and $A^T y + s = c$ but are required to be nonnegative, i.e. $x > 0$ and $s > 0$. The main idea is simply to ask Newton's method to make the iterates feasible. This amounts to a simple modification of the linear system (1.4), which becomes

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} c - A^T y_k - s_k \\ b - Ax_k \\ -X_k S_k e + \sigma \mu_k e \end{pmatrix} . \tag{1.9}$$

The only difference with the feasible system is the right-hand side vector, which now incorporates the primal and dual residuals $b - Ax_k$ and $c - (A^T y_k + s_k)$. Newton steps will try to reduce both the duality measure and the iterate infeasibility at the same time.

Infeasible variants of both path-following and potential reduction methods have been developed using this search direction. Without going into the details, let us point out that an additional constraint on the step has to be enforced to ensure that infeasibility is reduced at least at the same pace as the duality measure (to avoid ending with an "optimal" solution that would be infeasible). The complexity results for these methods are the same as those of their feasible counterparts, although the analysis is generally much more involved.

1.4.2 Homogeneous self-dual embedding

As mentioned in the previous subsection, another way to handle infeasibility is to embed our problem into a larger linear program that admits a known feasible starting point. We choose a starting iterate (x_0, y_0, s_0) such that $x_0 > 0$ and $s_0 > 0$ and define the following quantities

$$\begin{aligned} \hat{b} &= b - Ax_0 \\ \hat{c} &= c - A^T y_0 - s_0 \\ \hat{g} &= b^T y_0 - c^T x_0 - 1 \\ \hat{h} &= x_0^T s_0 + 1 . \end{aligned}$$

[18] This is the case for example when a variable that is not bounded by the constraints is not present in the objective.

We consider the following problem, introduced in [YTM94]

$$\begin{array}{lrrrrrrrl} \min & & & & \hat{h}\,\theta & & & & \\ \text{s.t.} & & Ax & -b\,\tau & +\hat{b}\,\theta & & & = & 0 \\ & -A^T y & & +c\,\tau & -\hat{c}\,\theta & -s & & = & 0 \\ & b^T y & -c^T x & & -\hat{g}\,\theta & & -\kappa & = & 0 \\ & -\hat{b}^T y & +\hat{c}^T x & +\hat{g}\,\tau & & & & = & -\hat{h} \\ & & x \geq 0 & \tau \geq 0 & & s \geq 0 & \kappa \geq 0 & & \end{array} \quad . \tag{HSD}$$

It is easy to see find a strictly feasible starting point for this problem. Indeed, one can check that $(x, y, s, \tau, \kappa, \theta) = (x_0, y_0, s_0, 1, 1, 1)$ is a suitable choice. Without going into too many details, we give a brief description of the new variables involved in (HSD): τ is a homogenizing variable, θ is measuring infeasibility and κ refers to the duality gap in the original problem. We also point out that the first two equalities correspond to the feasibility constraints $Ax = b$ and $A^T y + s = c$.

This program has the following interesting properties (see [YTM94]):

- ◇ This program is homogeneous, i.e. its right-hand side is the zero vector (except for the last equality that is a homogenizing constraint).
- ◇ This program is self-dual, i.e. its dual is identical to itself (this is due to the fact that the coefficient matrix is skew-symmetric).
- ◇ The optimal value of (HSD) is 0 (i.e. $\theta_* = 0$).
- ◇ Given a strictly complementary solution $(x_*, y_*, s_*, \tau_*, \kappa_*, 0)$ to (HSD) we have either $\tau_* > 0$ or $\kappa_* > 0$.
- ◇ If $\tau_* > 0$ then $(x_*/\tau_*, y_*/\tau_*, s_*/\tau_*)$ is an optimal solution to our original problem.
- ◇ If $\kappa_* > 0$ then our original problem has no finite optimal solution. Moreover, we have in this case $b^T y_* - c^T x_* > 0$ and
  - When $b^T y_* > 0$, problem (LP) is infeasible.
  - When $-c^T x_* > 0$, problem (LD) is infeasible.

Since we know a strictly feasible starting point, we can apply a feasible path-following method to this problem that will converge to an optimal strictly complementary solution. Using the above-mentioned properties, it is then possible to compute an optimal solution to our original problem or detect its infeasibility.

This homogeneous self-dual program has roughly twice the size of our original linear program, which may be seen as a drawback. However, it is possible to take advantage of the self-duality property and use some algorithmic devices to solve this problem at nearly the same computational cost as the original program.

1.4.3 Theory versus implemented algorithms

We have already mentioned that a polynomial complexity result is not necessarily a guarantee of good practical behaviour. Short-step methods are definitely too slow because of the tiny reduction of the duality measure they allow. Long-step methods perform better but are still too slow. This is why practitioners have implemented various tricks to accelerate their practical behaviour. It is important to note that the complexity results we have mentioned so far do not apply to these modified methods, since they do not strictly follow the theory.

The infeasible primal-dual long-step path-following algorithm is by far the most commonly implemented interior-point method. The following tricks are usually added:

- ◇ The theoretical long-step method takes several Newton steps targeting the same duality measure until proximity to the central path is restored. Practical algorithms ignore this and take only a single Newton step, like short-step methods.
- ◇ Instead of choosing the step length recommended by the theory, practical implementations usually take a very large fraction of the maximum step that stays within the feasible region (common values are 99.5% or 99.9%). This modification works especially well with primal-dual methods.
- ◇ The primal and dual steps are taken with different step lengths, i.e. we take

  $$x_{k+1} = x_k + \alpha^P \Delta x_k \text{ and } (y_{k+1}, s_{k+1}) = (y_k, s_k) + \alpha^D (\Delta y_k, \Delta s_k) \; .$$

  These steps are chosen according to the previous trick, for example with $(\alpha^P, \alpha^D) = 0.995\,(\alpha^P_{\max}, \alpha^D_{\max})$. This modification alone is responsible for a substantial decrease of the total number of iterations, but is not theoretically justified.

1.4.4 The Mehrotra predictor-corrector algorithm

The description of the methods from the previous section has underlined the fact that the constant σ, defining the target duality measure $\sigma\mu_k$, has a very important role in determining the algorithm efficiency:

- ◇ Choosing σ nearly equal to 1 allows us to take a full Newton step, but this step is usually very short and does not make much progress towards the solution. However it has the advantage of increasing the proximity to the central path.
- ◇ Choosing a smaller σ produces a larger Newton step making more progress towards optimality, but this step is generally infeasible and has to be damped. Moreover this kind of step usually tends to move the iterate away from the central path.

We understand that the best choice of σ may vary according to the current iterate: small if a far target is easy to attain and large otherwise. Mehrotra has designed a very efficient way to choose σ according to this principle: the predictor-corrector primal-dual infeasible algorithm [Meh92].

This algorithm first computes an affine-scaling *predictor* step $(\Delta x_k^a, \Delta y_k^a, \Delta s_k^a)$, i.e. solves (1.9) with $\sigma = 0$, targeting directly the optimal limit point of the central path. The maximum feasible step lengths are then computed separately using

$$\begin{aligned} \alpha_k^{a,P} &= \arg\max \{\alpha \in [0,1] \mid x_k + \alpha \Delta x_k^a \geq 0\} \ , \\ \alpha_k^{a,D} &= \arg\max \{\alpha \in [0,1] \mid s_k + \alpha \Delta s_k^a \geq 0\} \ . \end{aligned}$$

Finally, the duality measure of the resulting iterate is evaluated with

$$\mu_{k+1}^a = \frac{(x_k + \alpha_k^{a,P} \Delta x_k^a)^T (\alpha_s^{a,D} \Delta s_k^a)}{n} \ .$$

This quantity measures how easy it is to progress towards optimality: if it is much smaller than the current duality measure μ_k, we can choose a small σ and hope to make much progress, on the other hand if it is just a little smaller, we have to be more careful and choose σ closer to one, in order to increase proximity to the central path and be in a better position to achieve a large decrease of the duality measure on the *next* iteration. Mehrotra suggested the following heuristic, which has proved to be very efficient in practice

$$\sigma = \left(\frac{\mu_{k+1}^a}{\mu_k} \right)^3 \ .$$

We now simply compute a *corrector* step $(\Delta x_k^c, \Delta y_k^c, \Delta s_k^c)$ using this σ and take the maximum feasible step lengths separately in the primal and dual spaces.

However, this algorithm can be improved a little further using the following fact. After a full predictor step, the pairwise product $x_i s_i$ is transformed into $(x_i + \Delta x_i^a)(s_i + \Delta s_i^a)$, which can be shown to be equal to $\Delta x_i^a \Delta s_i^a$. Since Newton's method was trying to make $x_i s_i$ equal to zero, this last product measures the error due to the nonlinearity of the equations we are trying to solve. The idea is simply to incorporate this error term in the computation of the corrector step, using the following modification to the right-hand side in (1.9)

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} c - A^T y_k - s_k \\ b - A x_k \\ -X_k S_k e - \Delta X_k^a \Delta S_k^a e + \sigma \mu_k e \end{pmatrix} \ . \tag{1.10}$$

This strategy of computing a step taking into account the results of a first-order prediction gives rise to a second-order method. The complete algorithm follows:

Given an initial iterate (x_0, y_0, s_0) with duality measure μ_0 such that $x_0 > 0$ and $s_0 > 0$, an accuracy parameter ε and a constant $\rho < 1$ (e.g. 0.995 or 0.999).

Repeat for $k = 0, 1, 2, \ldots$

Compute the predictor Newton step $(\Delta x_k^a, \Delta y_k^a, \Delta s_k^a)$ using the linear system (1.9) and $\sigma = 0$.

Compute the maximal step lengths and the resulting duality measure with

$$\begin{aligned} \alpha_k^{a,P} &= \arg\max \{\alpha \in [0,1] \mid x_k + \alpha \Delta x_k^a \geq 0\} \ , \\ \alpha_k^{a,D} &= \arg\max \{\alpha \in [0,1] \mid s_k + \alpha \Delta s_k^a \geq 0\} \ , \\ \mu_{k+1}^a &= \frac{(x_k + \alpha_k^{a,P} \Delta x_k^a)^T (s_k + \alpha_k^{a,D} \Delta s_k^a)}{n} \ . \end{aligned}$$

Compute the corrector Newton step $(\Delta x_k^c, \Delta y_k^c, \Delta s_k^c)$ using the modified linear system (1.10) and $\sigma = \left(\mu_{k+1}^a/\mu_k\right)^3$.

Compute the maximal step lengths with

$$\begin{aligned} \alpha_k^P &= \arg\max\{\alpha \in [0,1] \mid x_k + \alpha \Delta x_k^c \geq 0\} \ , \\ \alpha_k^D &= \arg\max\{\alpha \in [0,1] \mid s_k + \alpha \Delta s_k^c \geq 0\} \ . \end{aligned}$$

Let $x_{k+1} = x_k + \rho\, \alpha_k^P \Delta x_k^c$ and $(y_{k+1}, s_{k+1}) = (y_k, s_k) + \rho\, \alpha_k^D (\Delta y_k^c, \Delta s_k^c)$.

Evaluate μ_{k+1} with $(x_{k+1}^T s_{k+1})/n$.

Until $n\mu_{k+1} < \varepsilon$

It is important to note that the predictor step is only used to compute σ and the right-hand side of (1.10) and is not actually taken. This has a very important effect on the computational work, since the calculation of both the predictor and the corrector step is made with the same current iterate. This implies that the coefficient matrix in the linear systems (1.10) and (1.9) is the same, the only difference being the right-hand side vector. The resolution of the second system will then reuse the factorization of the coefficient matrix and will only need a computationally cheap additional backsubstitution. This property is responsible for the great efficiency of Mehrotra's algorithm: a clever heuristic to decrease the duality measure using very little additional computational work.

1.5 Implementation

We mention here some important facts about the implementation of interior-point algorithms.

1.5.1 Linear algebra

It is important to realize that the resolution of the linear system defining the Newton step takes up most of the computing time in interior-point methods (some authors report 80–90% of the total CPU time). It should be therefore very carefully implemented. Equations (1.9) are not usually solved in this format: some pivoting is done, leading first to the following system (where we define $D_k^2 = S_k^{-1} X_k$)

$$\begin{pmatrix} -D_k^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = \begin{pmatrix} c - A^T y_k - \sigma\mu_k X_k^{-1} e) \\ b - Ax_k \end{pmatrix} \tag{1.11}$$

$$\Delta s_k = -s_k + \sigma\mu_k X_k^{-1} e - D_k^{-2} \Delta x_k \ , \tag{1.12}$$

and then to this one

$$AD_k^2 A^T \Delta y_k = b - A(x_k - D_k^2 c + D_k^2 A^T y_k + \sigma\mu_k S_k^{-1} e) \tag{1.13}$$

$$\Delta s_k = c - A^T y_k - s_k - A^T \Delta y_k \tag{1.14}$$

$$\Delta x_k = -x_k + \sigma\mu_k S_k^{-1} e - D_k^2 \Delta s_k \ . \tag{1.15}$$

System (1.11) is called the *augmented* system and can be solved with a Bunch-Partlett factorization. However, the most usual way to compute the Newton step is to solve (1.13), called the *normal* equation, with a Cholevsky factorization, taking advantage of the fact that matrix $AD_k^2A^T$ is positive definite (see [AGMX96] for a discussion). At this stage, it is important to note that most real-world problems have very few nonzero entries in matrix A. It is thus very important to exploit this sparsity in order to reduce both computing times and storage capacity requirements. More specifically, one should try to find a reordering of the rows and columns of matrix $AD_k^2A^T$ that leads to the sparsest Cholevsky factor[19]. This permutation has to be computed only once, since the sparsity pattern of matrix $AD_k^2A^T$ is the same for all iterations.

On a side note, let us note that the complexity of solving this linear system is $\mathcal{O}\left(n^3\right)$ arithmetic iterations, which gives the best interior-point methods a total complexity of

$$\mathcal{O}\big(n^{3.5}\log\frac{n\mu_0}{\varepsilon}\big)$$

arithmetic operations[20].

1.5.2 Preprocessing

In most cases, the linear program we want to solve is not formulated in the standard form (1.2). The first task for an interior-point solver is thus to convert it by adding variables and constraints

- ◇ Inequality constraints can be transformed into equality constraints with a slack variable: $f^Tx \geq b \Leftrightarrow f^Tx - s = b$ with $s \geq 0$.
- ◇ A free variable can be split into two nonnegative variables: $x = x^+ - x^-$ with $x^+ \geq 0$ and $x^- \geq 0$. However this procedure has some drawbacks[21] and practical solvers usually include a modification of the algorithm to handle free variables directly.
- ◇ Lower bounds $l \leq x$ are handled using a translation $x = x' + l$ with $x' \geq 0$.
- ◇ Upper bounds $x \leq u$ could be handled using a slack variable, but practical solvers usually implement a variation of the standard form that takes these bounds directly into account.

After this initial conversion, it is not unusual that a series of simple transformations can greatly reduce the size of the problem

- ◇ Zero lines and columns are either redundant (and thus may be removed) or make the problem infeasible.

[19] Because the problem of finding the optimal reordering is NP-hard, heuristics have been developed, e.g. the *minimum degree* and minimum local fill-in heuristics.

[20] A technique of partial updating of the coefficient matrix $AD_k^2A^T$ in the normal equation can reduce this total complexity to $\mathcal{O}\left(n^3\right)$.

[21] It makes for example the optimal solution set unbounded and the primal-dual strictly feasible set empty.

- ◇ Equality constraints involving only one variable are removed and used to fix the value of this variable.
- ◇ Equality constraints involving exactly two variables can be used to pivot out one the variables.
- ◇ Two identical lines are either redundant (one of them may thus be removed) or inconsistent (and make the problem infeasible).
- ◇ Some constraints may allow us to compute lower and upper bounds for some variables. These bounds can improve existing bounds, detect redundant constraints or diagnose an infeasible problem.

Every practical solver applies these rules (and some others) repeatedly before starting to solve the problem.

1.5.3 Starting point and stopping criteria

The problem of finding a suitable starting point has already been addressed by the homogeneous self-dual embedding technique and the infeasible methods. In both cases, any iterate satisfying $x_0 > 0$ and $s_0 > 0$ can be chosen as starting point. However, the actual performance of the algorithm can be greatly influenced by this choice.

Although there is no theoretical justification for it, the following heuristic is often used to find a starting point. We first solve

$$\min_{x\in\mathbb{R}^n} c^T x + \frac{\omega}{2} x^T x \quad \text{s.t.} \quad Ax = b \quad \text{and} \quad \min_{(y,s)\in\mathbb{R}^m\times\mathbb{R}^n} b^T y + \frac{\omega}{2} s^T s \quad \text{s.t.} \quad A^T y + s = c\ .$$

These convex quadratic programs can be solved analytically at a cost comparable to a single interior-point iteration. The negative components of the optimal x and s are then replaced with a small positive constant to give x_0 and (y_0, s_0).

As described earlier, the stopping criteria is usually a small predefined duality gap ε_g. In the case of an infeasible method, primal and dual infeasibility are also monitored and are required to fall below some predefined value ε_i. One can use for example the following formulas

$$\frac{\|Ax - b\|}{\|b\| + 1} < \varepsilon_i, \quad \frac{\|A^T y + s - c\|}{\|c\| + 1} < \varepsilon_i, \quad \frac{\|c^T x - b^T y\|}{\|c^T x\| + 1} < \varepsilon_g\ .$$

The denominators are used to make these measures relative and the $+1$ constant to avoid division by zero. However, when dealing with an infeasible problem, infeasible methods tend to see their iterates diverging towards infinity. Practical solvers usually detect this behaviour and diagnose an infeasible problem.

1.6 Concluding remarks

The theory of interior-point methods for linear optimization is now well established ; several textbooks on the topic have been published (see e.g. [Wri97, RTV97, Ye97]). From a prac-

tical point of view, interior-point methods compete with the best simplex implementations, especially for large-scale problems.

However some unsatisfying issues remain, in particular the gap between theoretical and implemented algorithms. Another interesting point is the number of iterations that is practically observed, almost independent from the problem size or varying like $\log n$ or $n^{1/4}$, instead of the $\sqrt{n}$ theoretical bound.

Research is now concentrating on the adaptation of these methods to the nonlinear framework. Let us mention the following directions:

- ◇ *Semidefinite optimization* is a promising generalization of linear optimization in which the nonnegativity condition on a vector $x \geq 0$ is replaced by the requirement that a symmetric matrix X is positive semidefinite. This kind of problem has numerous applications in various fields, e.g. combinatorial optimization (with the famous Goemans-Williamson bound on the quality of a semidefinite MAXCUT relaxation [GW95]), control, classification (see [Gli98b] and Appendix A), structural optimization, etc. (see [VB96] for more information). The methods we have presented here can be adapted to semidefinite optimization with relatively little effort and several practical algorithms are able to solve this kind of problem quite efficiently.
- ◇ In their brilliant monograph [NN94], Nesterov and Nemirovski develop a complete theory of interior-point methods applicable to the whole class of *convex* optimization problems. They are able to prove polynomial complexity for several types of interior-point methods and relate their efficiency to the existence of a certain type of barrier depending on the problem structure, a so-called *self-concordant* barrier. This topic is further discussed in Chapter 2.

CHAPTER 2

Self-concordant functions

This chapter provides a self-contained introduction to the theory of self-concordant functions [NN94] and applies it to several classes of structured convex optimization problems. We describe the classical short-step interior-point method and optimize its parameters to provide its best possible iteration bound. We also discuss the necessity of introducing two parameters in the definition of self-concordancy, how they react to addition and scaling and which one is the best to fix. A lemma from [dJRT95] is improved and allows us to review several classes of structured convex optimization problems and evaluate their algorithmic complexity, using the self-concordancy of the associated logarithmic barriers.

2.1 Introduction

We start with a presentation of convex optimization.

2.1.1 Convex optimization

Convex optimization deals with the following problem

$$\inf_{x\in\mathbb{R}^n} f_0(x) \quad \text{s.t.} \quad x \in C\,, \tag{C}$$

where $C \subseteq \mathbb{R}^n$ is a closed convex set and $f_0 : C \mapsto \mathbb{R}$ is a convex function defined on C. Convexity of f_0 and C plays a very important role in this problem, since it is responsible for

the following two important properties [Roc70a, SW70]:

- ◇ Any local optimum for (C) is also a global optimum, which implies that the objective value is equal for all local optima. Moreover, all these optima can be shown to form a convex set.
- ◇ It is possible to use Lagrange duality to derive a dual problem strongly related to (C). Namely, this pair of problems satisfies a weak duality property (the objective value of any feasible solution for one of these problems provides a bound on the optimum objective value for the dual problem) and, under a Slater-type condition, a strong duality property (equality and attainment of the optimum objective values for the two problems). These properties are described with more detail in Section 3.2.

We first note that it can be assumed with any loss of generality that the objective function f_0 is linear, so that we can define it as $f_0(x) = c^T x$ using a vector $c \in \mathbb{R}^n$. Indeed, it is readily seen that problem (C) is equivalent to the following problem with a linear objective:

$$\inf_{x \in \mathbb{R}^n,\ t \in \mathbb{R}} t \quad \text{s.t.} \quad (x,t) \in \bar{C} \ ,$$

where $\bar{C} \subseteq \mathbb{R}^{n+1}$ is suitably defined as

$$\bar{C} = \left\{ (x,t) \in \mathbb{R}^{n+1} \mid x \in C \text{ and } f(x) \leq t \right\} \ .$$

We will thus consider in the rest of this chapter the problem

$$\inf_{x \in \mathbb{R}^n} c^T x \quad \text{s.t.} \quad x \in C \ . \tag{CL}$$

It is interesting to ask ourselves how one can specify the data of a problem cast in such a form, i.e. how one can describe its objective function and feasible set. While specifying the objective function is easily done by providing vector c, describing the feasible set C, which is responsible for the *structure* of problem (CL), can be done in several manners.

a. The traditional way to proceed in nonlinear optimization is to provide a list of convex constraints defining C, i.e.

$$C = \left\{ x \in \mathbb{R}^n \mid f_i(x) \leq 0 \ \forall i \in I = \{1, 2, \ldots, m\} \right\} \ , \tag{2.1}$$

where each of the m functions $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ is convex. This guarantees the convexity of C, as an intersection of convex level sets.

b. An alternative approach consists in considering the domain of a convex function. More precisely, we require the interior of C to be equal to the domain of a convex function. Extending the real line $\mathbb{R}$ with the quantity $+\infty$, we introduce the convex function $F : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ and define C as the closure of its effective domain, i.e.

$$C = \operatorname{cl} \operatorname{dom} F = \operatorname{cl} \left\{ x \in \mathbb{R}^n \mid F(x) < +\infty \right\} \ .$$

Most of the time, we will require in addition F to be a *barrier function* for the set C, according to the following definition.

Definition 2.1. A function F is a *barrier function* for the convex set C if and only if it satisfies the following assumptions:

(a) F is smooth (three times continuously differentiable for our purpose),

(b) F is strictly convex, i.e. $\nabla^2 F$ is positive definite,

(c) $F(x)$ tends to $+\infty$ whenever x tends to ∂C, the boundary of C (this is the barrier property).

Note 2.1. We also note that it is often possible to provide a suitable barrier function F for a convex set C given by a functional description (2.1) using the *logarithmic barrier* [Fri55] defined as

$$F : \mathbb{R}^n \mapsto \mathbb{R} : x \mapsto F(x) = -\sum_{i \in I} \log(-f_i(x)) \ ,$$

where we define $\log z = +\infty$ whenever $z \in \mathbb{R}_-$. We have indeed to check that F is strictly convex and is a barrier function for C, which is not always the case (for example, in the case of $C = \mathbb{R}_+$, taking $f_1(x) = |x| - x$ does not lead to a strictly convex F while $f_1(x) = -x^x$ leads to $F(x) = -x \log x$, which does not possess the barrier property).

c. It may also be worthwhile to consider the special case where C can be described as the intersection of a convex cone $\mathcal{C} \subseteq \mathbb{R}^n$ and an affine subspace $b + L$ (where L is a linear subspace)

$$C = \mathcal{C} \cap (b + L) = \{x \in \mathcal{C} \mid x - b \in L\} \ .$$

The resulting class of problems is known as conic optimization, and can be easily shown to be equivalent to convex optimization [NN94] (in practice, subspace $b + L$ would be defined with a set of linear equalities).

Special treatment for the linear constraints, i.e. their representation as an intersection with an affine subspace, can be justified by the fact that these constraints are easier to handle than general nonlinear constraints. In particular, let us mention that it is usually easy for algorithms to preserve feasibility with respect to these constraints, and that they cannot cause a nonzero duality gap, i.e. strong duality is valid without a Slater-type assumption for linear optimization. We will not need to use this approach in this chapter. It will nevertheless constitute the main tool used in the second part of this thesis, which focuses on the topic of duality (see Chapters 4–7).

2.1.2 Interior-point methods

Among the different types of algorithms that can be applied to solve problem (CL), the so-called *interior-point methods* have gained a lot of popularity in the last two decades. This is mainly due to the following facts:

◇ it is not only possible to prove convergence of these methods to an optimal solution but also to give a polynomial bound on the number of arithmetic operations needed to reach a solution within a given accuracy,

- ◇ these methods can be implemented and applied successfully to solve real-world problems, especially in the fields of linear (where they compare favourably with the simplex method), quadratic and semidefinite optimization.

A fundamental ingredient in the elaboration of these methods is the above-mentioned notion of barrier function F for the set C. Namely, let us consider the following parameterized family of unconstrained minimization problems:

$$\inf_{x\in\mathbb{R}^n} \frac{c^T x}{\mu} + F(x) \ , \tag{CL$_\mu$}$$

where parameter μ belongs to $\mathbb{R}_{++}$ and is called the *barrier parameter*. The constraint $x \in C$ of the original problem (CL) has been replaced by a penalty term $F(x)$ in the objective function, which tends to $+\infty$ as x tends to the boundary of C and whose purpose is to avoid that the iterates leave the feasible set (see the classical monograph [FM68]). Assuming existence of a minimizer $x(\mu)$ for each of these problems (strong convexity of F ensures uniqueness of such a minimizer), we call the set $\{x(\mu) \mid \mu > 0\} \subseteq C$ the central path for problem (CL).

It is intuitively clear that as μ tends to zero, the first term proportional to the original objective $\frac{c^T x}{\mu}$ becomes preponderant in the sum, which implies that the central path converges to a solution that is optimal for the original problem. The principle behind interior-point methods will thus be to follow this central path until an iterate that is sufficiently close to the optimum is found.

However, two questions remain pending: how do we compute $x(\mu)$ and how do we choose a suitable barrier F. The first question is readily answered: interior-point methods rely on Newton's method to compute these minimizers, which leads us to a refined version of the second question: is it possible to choose a barrier function F such that Newton's method is provably efficient in solving subproblems (CL$_\mu$) and has an algorithmic complexity that can be estimated ? This crucial question is thoroughly answered by the remarkable theory of self-concordant functions, first developed by Nesterov and Nemirovski [NN94], which we will present in the next section.

2.1.3 Organization of the chapter

The purpose of this chapter is to give a self-contained introduction to the theory of self-concordant functions and to apply it to several classes of structured convex optimization problems. Section 2.2 introduces a definition of self-concordant functions and presents several equivalent conditions. A short-step interior-point method using these functions is then presented along with an explanation of how the proof of polynomiality works. Our contribution at this stage is the computation of the best possible iteration bound for this method (Theorem 2.5).

Section 2.3 deals with the construction of self-concordant functions. Scaling and addition of self-concordant functions are considered, as well as a discussion on the utility of two parameters in the definition of self-concordancy and how to fix one of them in the best possible

way. We then present an improved version of a lemma from [dJRT95] (Lemma 2.3). This lemma is the main tool used in Section 2.4, where we review several classes of structured convex optimization problems and prove self-concordancy of the corresponding logarithmic barriers, improving the complexity results found in [dJRT95]. We conclude in Section 2.5 with some comments.

2.2 Self-concordancy

We start this section with a definition of a self-concordant function.

2.2.1 Definitions

We first recall the following piece of notation: the first, second and third differentials of a function $F : \mathbb{R}^n \mapsto \mathbb{R}$ evaluated at the point x will be denoted by $\nabla F(x)$, $\nabla^2 F(x)$ and $\nabla^3 F(x)$. These are linear mappings, and we have indeed

$$\begin{aligned}\nabla F(x) &: \mathbb{R}^n \mapsto \mathbb{R} : h_1 \mapsto \nabla F(x)[h_1] \\ \nabla^2 F(x) &: \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R} : (h_1, h_2) \mapsto \nabla^2 F(x)[h_1, h_2] \\ \nabla^3 F(x) &: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R} : (h_1, h_2, h_3) \mapsto \nabla^3 F(x)[h_1, h_2, h_3] \ .\end{aligned}$$

Definition 2.2. A function $F : C \mapsto R$ is called (κ, ν)-*self-concordant* for the convex set $C \subseteq \mathbb{R}^n$ if and only if F is a barrier function according to Definition 2.1 and the following two conditions hold for all $x \in \operatorname{int} C$ and $h \in \mathbb{R}^n$:

$$\nabla^3 F(x)[h,h,h] \leq 2\kappa \left(\nabla^2 F(x)[h,h]\right)^{\frac{3}{2}} \ , \tag{2.2}$$

$$\nabla F(x)^T (\nabla^2 F(x))^{-1} \nabla F(x) \leq \nu \tag{2.3}$$

(note that the square root in (2.2) is well defined since its argument $\nabla^2 F(x)[h,h]$ is positive because of the requirement that F is convex).

This definition does not exactly match the original definition of a self-concordant barrier in [NN94], but merely corresponds to the notion of strongly non-degenerate κ^{-2}-self-concordant barrier functional with parameter ν, that is general enough for our purpose.

Note 2.2. We would like to point out that no absolute value is needed in (2.2): while some authors usually require the apparently stronger condition

$$\left|\nabla^3 F(x)[h,h,h]\right| \leq 2\kappa \left(\nabla^2 F(x)[h,h]\right)^{\frac{3}{2}} \ , \tag{2.4}$$

this is not needed since it suffices to notice that inequality (2.2) also has to hold in the direction opposite to h, which gives

$$\nabla^3 F(x)[-h,-h,-h] \leq 2\kappa \left(\nabla^2 F(x)[-h,-h]\right)^{\frac{3}{2}} \Leftrightarrow -\nabla^3 F(x)[h,h,h] \leq 2\kappa \left(\nabla^2 F(x)[h,h]\right)^{\frac{3}{2}}$$

(using the fact that the n^{th}-order differential is homogeneous with degree n), which combined with (2.2) gives condition (2.4).

It is possible to reformulate conditions (2.2) and (2.3) into several equivalent inequalities that may prove easier to handle in some cases. However, before we list them, we would like to make a few comments about the use of inner products in our setting, following the line of thought of Renegar's monograph [Ren00].

It is indeed important to realize that the definitions of gradient and Hessian, i.e. first-order and second-order differentials are in fact dependent from inner product that is being used. Nevertheless, in most texts, it is customary to use the dot product[1] as standard inner product. This has the disadvantage to make all developments *a priori* dependent from the coordinate system. However, Renegar notices that it is possible to develop the theory of self-concordant functions in a completely coordinate-free manner, i.e. independently of a reference inner product. The is due to the fact that the two principal objects in this theory are indeed independent from the coordinate system: the Newton step $n(x)$ and the intrinsic inner product $\langle \cdot, \cdot \rangle_x$. Given a barrier function F and a point x belonging to its domain, these two objects are defined according to:

$$n(x) = -(\nabla^2 F(x))^{-1} \nabla F(x) \quad \text{and} \quad \langle \alpha, \beta \rangle_x = \langle \alpha, \nabla^2 F(x) \beta \rangle \ .$$

It is also convenient to introduce the intrinsic norm $\|\cdot\|_x$ based on the intrinsic inner product $\langle \cdot, \cdot \rangle_x$ according to the usual definition $\|a\|_x = \sqrt{\langle a, a \rangle_x}$.

Let $x \in \operatorname{int} C$ and $h \in \mathbb{R}^n$ and let us introduce the one-dimensional function $F_{x,h} : \mathbb{R} \mapsto \mathbb{R} : t \mapsto F(x+th)$, the restriction of F along the line $\{x + th \mid t \in \mathbb{R}\}$. We are now in position to state several reformulations of conditions (2.2) and (2.3), grouped in the following two theorems:

Theorem 2.1. *The following four conditions are equivalent:*

$$\nabla^3 F(x)[h,h,h] \leq 2\kappa \left(\nabla^2 F(x)[h,h]\right)^{\frac{3}{2}} \text{ for all } x \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n \tag{2.5a}$$

$$F'''_{x,h}(0) \leq 2\kappa F''_{x,h}(0)^{\frac{3}{2}} \text{ for all } x \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n \tag{2.5b}$$

$$F'''_{x,h}(t) \leq 2\kappa F''_{x,h}(t)^{\frac{3}{2}} \text{ for all } x + th \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n \tag{2.5c}$$

$$\left(-\frac{1}{\sqrt{F''_{x,h}(t)}}\right)' \leq \kappa \text{ for all } x + th \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n \ . \tag{2.5d}$$

Proof. Since $F_{x,h}(t) = F(x+th)$, we can write

$$F'_{x,h}(t) = \nabla F(x+th)[h], \ F''_{x,h}(t) = \nabla^2 F(x+th)[h,h] \text{ and } F'''_{x,h}(t) = \nabla^3 F(x+th)[h,h,h] \ .$$

Condition (2.5b) is thus simply condition (2.5a) written differently. Moreover, condition (2.5c) is equivalent to condition (2.5b) written for $x + th$ instead of x. Finally, we note that

$$\left(-\frac{1}{\sqrt{F''_{x,h}(t)}}\right)' \leq \kappa \Leftrightarrow \frac{1}{2} F''_{x,h}(t)^{-\frac{3}{2}} F'''_{x,h}(t) \leq \kappa \Leftrightarrow F'''_{x,h}(t) \leq 2\kappa F''_{x,h}(t)^{\frac{3}{2}} \ ,$$

which shows that (2.5d) and (2.5c) are equivalent. □

[1] The dot product of two vector x and y whose coordinates are $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $(\beta_1, \beta_2, \ldots, \beta_n)$ in a given coordinate system is equal to $\sum_{i=1}^n \alpha_i \beta_i$.

Theorem 2.2. *The following four conditions are equivalent:*

$$\begin{aligned}
\nabla F(x)^T(\nabla^2 F(x))^{-1}\nabla F(x) &\leq \nu \text{ for all } x \in \operatorname{int} C && \text{(2.6a)}\\
F'_{x,h}(0)^2 &\leq \nu F''_{x,h}(0) \text{ for all } x \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n && \text{(2.6b)}\\
F'_{x,h}(t)^2 &\leq \nu F''_{x,h}(t) \text{ for all } x+th \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n && \text{(2.6c)}\\
\Big(-\frac{1}{F'_{x,h}(t)}\Big)' &\geq \frac{1}{\nu} \text{ for all } x+th \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n\,. && \text{(2.6d)}
\end{aligned}$$

Proof. Proving these equivalences is a little more involved than for the previous theorem. We start by showing that condition (2.6b) implies condition (2.6a). We can write

$$\begin{aligned}
\nabla F(x)^T(\nabla^2 F(x))^{-1}\nabla F(x) &= \nabla F(x)[(\nabla^2 F(x))^{-1}\nabla F(x)] = F'_{x,(\nabla^2 F(x))^{-1}\nabla F(x)}(0)\\
&\leq \sqrt{\nu}\sqrt{F''_{x,(\nabla^2 F(x))^{-1}\nabla F(x)}(0)} \text{ using condition (2.6b)}\\
&= \sqrt{\nu}\sqrt{\nabla^2 F(x)[(\nabla^2 F(x))^{-1}\nabla F(x),(\nabla^2 F(x))^{-1}\nabla F(x)]}\\
&= \sqrt{\nu}\sqrt{\nabla F(x)^T(\nabla^2 F(x))^{-1}\nabla^2 F(x)(\nabla^2 F(x))^{-1}\nabla F(x)}\\
&= \sqrt{\nu}\sqrt{\nabla F(x)^T(\nabla^2 F(x))^{-1}\nabla F(x)}\,,
\end{aligned}$$

which implies condition (2.6a). Considering now the reverse implication, we have

$$\begin{aligned}
F'_{x,h}(0)^2 &= (\nabla F(x)[h])^2 = \big(\nabla F(x)^T h\big)^2\\
&= \big(\nabla F(x)^T(\nabla^2 F(x))^{-1}\nabla^2 F(x)h\big)^2 = \langle(\nabla^2 F(x))^{-1}\nabla F(x),h\rangle_x^2\\
&\leq \big\|(\nabla^2 F(x))^{-1}\nabla F(x)\big\|_x^2\,\|h\|_x^2 \text{ (using the Cauchy-Schwarz inequality)}\\
&= \big(\nabla F(x)^T(\nabla^2 F(x))^{-1}\nabla^2 F(x)(\nabla^2 F(x))^{-1}\nabla F(x)\big)\,\big(h^T\nabla^2 F(x)h\big)\\
&\leq \nu\nabla^2 F(x)[h,h] \text{ using condition (2.6a)}\\
&= \nu F''_{x,h}(0)\,.
\end{aligned}$$

Condition (2.6c) is condition (2.6b) written for $x+th$ instead of x, and we finally note that

$$\Big(-\frac{1}{F'_{x,h}(t)}\Big)' \geq \frac{1}{\nu} \Leftrightarrow F'_{x,h}(t)^{-2}F''_{x,h}(t) \geq \frac{1}{\nu} \Leftrightarrow \nu F''_{x,h}(t) \geq F'_{x,h}(t)^2\,,$$

which shows that (2.6d) and (2.6c) are equivalent. □

The first three reformulations for each condition are well-known and can be found for example in [NN94, Jar96, Ren00]. Conditions (2.5d) and (2.6d) are less commonly seen (they were however mentioned in [Bri00]).

2.2.2 Short-step method

As outlined in the introduction, interior-point methods for convex optimization rely on a barrier function and the associated central path to solve problem (CL). Ideally, we would like

our iterates to be a sequence of points on the central path $x(\mu_0),\ x(\mu_1), \ldots,\ x(\mu_k), \ldots$ for a sequence of barrier parameters μ_k tending to zero (and thus $x(\mu_k)$ tending to an optimal solution).

We already mentioned that Newton's method, applied to problems (CL_μ), will be the workhorse to compute those minimizers. However, it would be too costly to compute each of these points with high accuracy, so that interior-point methods require instead their iterates to lie in a prescribed neighbourhood of the central path and its exact minimizers.

Let x_k, the k^{th} iterate, be an approximation of $x(\mu_k)$. A good proximity measure would be $\|x_k - x(\mu_k)\|$ or, to be independent from the coordinate system, $\|x_k - x(\mu_k)\|_{x_k}$. However, these quantities involve the unknown central point $x(\mu_k)$, and are therefore difficult to work with. Nevertheless, another elegant proximity measure can be used for that purpose. Let us define $n_\mu(x)$ to be the Newton step trying to minimize the objective in problem (CL_μ), which is thus aiming at $x(\mu)$. Since this objective is equal to $F_\mu(x) = \frac{c^T x}{\mu} + F(x)$, we have

$$\begin{aligned} n_\mu(x) = -(\nabla^2 F_\mu(x))^{-1} \nabla F_\mu(x) &= -(\nabla^2 F(x))^{-1}(\frac{c}{\mu} + \nabla F(x)) \\ &= -\frac{1}{\mu}(\nabla^2 F(x))^{-1} c + n(x)\ . \end{aligned} \tag{2.7}$$

Let us now define $\delta(x, \mu)$, a measure of the proximity of x to the central point $x(\mu)$, as the intrinsic norm of the newton step $n_\mu(x)$, i.e. $\delta(x, \mu) = \|n_\mu(x)\|_x$. This quantity is indeed a good candidate to measure how far x lies from the minimizer $x(\mu)$, since the Newton step at x targeting $x(\mu)$ is supposed to be a good approximation of $x(\mu) - x$. The goal of a short-step interior-point method will be to trace the central path approximately, ensuring that the proximity $\delta(x_k, \mu_k)$ is kept below a predefined bound for each iterate.

We are now in position to sketch a short-step algorithm. Given a problem of type (CL), a barrier function F for C, an upper bound on the proximity measure $\tau > 0$, a decrease parameter $0 < \theta < 1$ and an initial iterate x_0 such that $\delta(x_0, \mu_0) < \tau$, we set $k \leftarrow 0$ and perform the following main loop:

a. $\mu_{k+1} \leftarrow \mu_k(1 - \theta)$

b. $x_{k+1} \leftarrow x_k + n_{\mu_{k+1}}(x_k)$

c. $k \leftarrow k + 1$

The key is to choose parameters τ and θ such that $\delta(x_k, \mu_k) < \tau$ implies $\delta(x_{k+1}, \mu_{k+1}) < \tau$, so that proximity to the central path is preserved. This is the moment where the self-concordancy of the barrier function F comes into play. Indeed, it is precisely this property that will guarantee that such a choice is always possible.

2.2.3 Optimal complexity

In order to relate the two proximities $\delta(x_k, \mu_k)$ and $\delta(x_{k+1}, \mu_{k+1})$, it is useful to introduce an intermediate quantity $\delta(x_k, \mu_{k+1})$, the proximity from an iterate to its next target on the central path. We have the following two properties:

Theorem 2.3. *Let F be a barrier function satisfying* (2.3)*, $x \in \operatorname{dom} F$ and $\mu^+ = (1-\theta)\mu$. We have*

$$\delta(x, \mu^+) \leq \frac{\delta(x,\mu) + \theta\sqrt{\nu}}{1-\theta} \ .$$

Proof. Using (2.7), we have

$$\begin{array}{rcl} & & \mu^+ n_{\mu^+}(x) - \mu^+ n(x) = -(\nabla^2 F(x))^{-1} c = \mu n_\mu(x) - \mu n(x) \\ \text{(dividing by } \mu) & \Leftrightarrow & (1-\theta) n_{\mu^+}(x) - (1-\theta) n(x) = n_\mu(x) - n(x) \\ & \Leftrightarrow & (1-\theta) n_{\mu^+}(x) = n_\mu(x) - \theta n(x) \\ & \Rightarrow & (1-\theta) \left\| n_{\mu^+}(x) \right\|_x \leq \left\| n_\mu(x) \right\|_x + \theta \left\| n(x) \right\|_x \\ & \Rightarrow & (1-\theta)\delta(x,\mu^+) \leq \delta(x,\mu) + \theta\sqrt{\nu} \ , \end{array}$$

which implies the desired inequality, where we used to derive the last implication the fact that

$$\begin{array}{rcl} \left\| n(x) \right\|_x = \sqrt{\langle n(x), n(x) \rangle_x} & = & \sqrt{\nabla F(x)^T (\nabla^2 F(x))^{-1} \nabla^2 F(x) (\nabla^2 F(x))^{-1} \nabla F(x)} \\ & = & \sqrt{\nabla F(x)^T (\nabla^2 F(x))^{-1} \nabla F(x)} \leq \sqrt{\nu} \ , \end{array}$$

because of condition (2.3). □

Theorem 2.4. *Let F be a barrier function satisfying* (2.2) *and $x \in \operatorname{dom} F$. Let us suppose $\delta(x,\mu) < \frac{1}{\kappa}$. We have that $x + n_\mu(x) \in \operatorname{dom} F$ and*

$$\delta(x + n_\mu(x), \mu) \leq \frac{\kappa \delta(x,\mu)^2}{(1 - \kappa\delta(x,\mu))^2} \ .$$

This proof is more technical and is omitted here ; it can be found in [NN94, Jar96, Ren00].

Note 2.3. It is now clear why the self-concordancy property relies on two separate conditions: one of them is responsible for the control of the increase of the proximity measure when the target on the central path is updated (Theorem 2.3), while the other guarantees that the proximity to the target can be restored, i.e. sufficiently decreased when taking a Newton step (Theorem 2.4).

Assuming for the moment that τ and θ can be chosen such that the proximity to central path is preserved at each iteration, we see that the number of iterations needed to attain a certain value of the barrier parameter μ_e depends solely on the ratio $\frac{\mu_0}{\mu_e}$ and the value of θ. Namely, since $\mu_k = (1-\theta)^k \mu_0$, it is readily seen that this number of iterations is equal to

$$\left\lceil \log_{(1-\theta)} \frac{\mu_e}{\mu_0} \right\rceil = \left\lceil \frac{1}{\log(1-\theta)} \log \frac{\mu_e}{\mu_0} \right\rceil \ . \tag{2.8}$$

Given a (κ, ν)-self-concordant function, we are now going to find a suitable pair of parameters τ and θ . Moreover, we will optimize this choice of parameters, i.e. try to provide the greatest reduction for the parameter μ at each iteration, in other words maximize θ in

order to get the lowest possible total iteration count. Letting $\delta = \delta(x_k, \mu_k)$, $\delta' = \delta(x_k, \mu_{k+1})$ and $\delta^+ = \delta(x_{k+1}, \mu_{k+1})$ and assuming $\delta \leq \tau$, we have to satisfy $\delta^+ \leq \tau$ with the greatest possible value for θ.

Let us assume first that $\delta' < \frac{1}{\kappa}$. Using Theorem 2.4, we find that

$$\delta^+ \leq \frac{\kappa\delta'^2}{(1-\kappa\delta')^2}$$

and therefore require that

$$\frac{\kappa\delta'^2}{(1-\kappa\delta')^2} \leq \tau \ .$$

This is equivalent to

$$\left(\frac{\kappa\delta'}{1-\kappa\delta'}\right)^2 \leq \kappa\tau \Leftrightarrow \left(\frac{1}{\kappa\delta'} - 1\right)^2 \geq \frac{1}{\kappa\tau} \Leftrightarrow \frac{1}{\kappa\delta'} \geq 1 + \frac{1}{\sqrt{\kappa\tau}}$$

(this also shows that the assumption $\kappa\delta' < 1$ we made in the beginning was valid). Using now Theorem 2.3, we know that

$$\delta' \leq \frac{\delta + \theta\sqrt{\nu}}{1-\theta} \Rightarrow \delta' \leq \frac{\tau + \theta\sqrt{\nu}}{1-\theta} \Leftrightarrow \frac{1}{\kappa\delta'} \geq \frac{1-\theta}{\kappa\tau + \theta\kappa\sqrt{\nu}}$$

and thus require that

$$\frac{1-\theta}{\kappa\tau + \theta\kappa\sqrt{\nu}} \geq 1 + \frac{1}{\sqrt{\kappa\tau}} \ .$$

Letting $\Gamma = \kappa\sqrt{\nu}$ and $\beta = \sqrt{\kappa\tau}$ we have

$$\frac{1-\theta}{\beta^2 + \theta\Gamma} \geq 1 + \frac{1}{\beta} \Leftrightarrow 1 - \theta \geq (1 + \frac{1}{\beta})(\beta^2 + \Gamma\theta) \Leftrightarrow 1 - \beta - \beta^2 \geq \theta\left(1 + \Gamma + \frac{\Gamma}{\beta}\right) \ ,$$

which means finally that we have to choose θ such that

$$\theta \leq \frac{1 - \beta - \beta^2}{1 + \Gamma + \frac{\Gamma}{\beta}} \tag{2.9}$$

in order to guarantee $\delta^+ \leq \tau$. We are now in position to optimize the value of θ, i.e. find the value of β that maximizes this upper bound. However, this value is likely to depend on Γ (and thus on κ and ν) in a complex way. We are therefore going to work with the following slightly worse upper bound, which has the advantage of allowing the optimization of β independently of Γ (we use the fact that $\Gamma = \kappa\sqrt{\nu} \geq 1$, see [NN94])

$$\theta \leq \frac{1}{\Gamma}\Big(\frac{1 - \beta - \beta^2}{2 + \frac{1}{\beta}}\Big) = \frac{f(\beta)}{\Gamma} \quad \Big(\leq \frac{1 - \beta - \beta^2}{1 + \Gamma + \frac{\Gamma}{\beta}}\Big) \ .$$

It is now straightforward to maximize $f(\beta)$: computing the derivative shows there is a unique maximizer when $\beta \approx 0.273$ (the exact value is the real root of $1 - 2\beta - 5\beta^2 - 4\beta^3$) and our upper bound in (2.9) becomes $\frac{0.65}{1+4.66\Gamma}$. Translating back into our original quantities τ, κ and ν we find that we can choose

$$\tau = \frac{\beta^2}{\kappa} \approx \frac{1}{13.42\kappa} \quad \text{and} \quad \theta = \frac{1 - \beta - \beta^2}{1 + \Gamma + \frac{\Gamma}{\beta}} \approx \frac{1}{1.53 + 7.15\kappa\sqrt{\nu}} \ , \tag{2.10}$$

which is the best result obtainable if we want β to be independent from κ and ν (more precisely, it essentially corresponds to the best result in the case where $\kappa\sqrt{\nu} = 1$). This improves several results from the literature, e.g. $\theta = \frac{1}{9\kappa\sqrt{\nu}}$ in [Jar96] and $\theta = \frac{1}{1+8\kappa\sqrt{\nu}}$ in [Ren00].

Before we conclude this section with a global complexity result, let us say a few words about termination of the algorithm. The most practical stopping criterion is a small target value μ_e for the barrier parameter, which gives the iteration bound (2.8). Our final iterate x_e will thus satisfy $\delta(x_e, \mu_e) \leq \tau$, which tells us it is not too far from $x(\mu_e)$, itself not too far from the optimum since μ_e is small. Indeed, using again the self-concordancy property of F, it is possible to derive the following bound on the accuracy of the final objective $c^T x_e$, i.e. its deviation from the optimal objective $c^T x^*$

$$c^T x_e - c^T x^* \leq \frac{\mu_e}{1 - 3\kappa\tau}\kappa\sqrt{\nu} \tag{2.11}$$

(proof of this fact is omitted here and can easily be obtained combining Theorems 2.2.5 and 2.3.3 in [Ren00]). We are now ready to state our final complexity result:

Theorem 2.5. *Given a convex optimization problem* (CL)*, a (κ, ν)-self-concordant barrier F for C and an initial iterate x_0 such that $\delta(x_0, \mu_0) < \frac{1}{13.42\kappa}$, one can find a solution with accuracy ϵ in*

$$\left\lceil (1.03 + 7.15\kappa\sqrt{\nu}) \log \frac{1.29\mu_0\kappa\sqrt{\nu}}{\epsilon} \right\rceil \text{ iterations.}$$

Proof. Using our optimal values for θ and τ from (2.10) and the bound on the objective accuracy in (2.11), we find that the stopping threshold on the barrier parameter μ_e must satisfy

$$\frac{\mu_e}{1 - 3/13.42}\kappa\sqrt{\nu} \leq \epsilon \Leftrightarrow 1.29\mu_e\kappa\sqrt{\nu} \leq \epsilon \Leftrightarrow \mu_e \leq \frac{\epsilon}{1.29\kappa\sqrt{\nu}} .$$

Plugging this value into (2.8) we find that the total number of iterations can be bounded by (omitting the rounding bracket for clarity)

$$\begin{aligned}
\frac{1}{\log(1-\theta)} \log \frac{\mu_e}{\mu_0} &\leq \frac{1}{\log(1-\theta)} \log \frac{\epsilon}{1.29\mu_0\kappa\sqrt{\nu}} \\
&= -\frac{1}{\log(1-\theta)} \log \frac{1.29\mu_0\kappa\sqrt{\nu}}{\epsilon} \\
&\leq \left(\frac{1}{\theta} - \frac{1}{2}\right) \log \frac{1.29\mu_0\kappa\sqrt{\nu}}{\epsilon} \\
&= (1.03 + 7.15\kappa\sqrt{\nu}) \log \frac{1.29\mu_0\kappa\sqrt{\nu}}{\epsilon} ,
\end{aligned}$$

as announced (the third line uses the inequality $\frac{1}{\log(1-\theta)} \geq \frac{1}{2} - \frac{1}{\theta}$, which can be easily derived using the Taylor series of $\log x$ around 1). □

2.3 Proving self-concordancy

The previous section has made clear that the self-concordancy property of the barrier function F is essential to derive a polynomial bound on the number of iterations of the short-step

method. Moreover, smaller values for parameters κ and ν imply a lower total complexity. The next question we may ask ourselves is how to find self-concordant barriers (ideally with low parameters).

2.3.1 Barrier calculus

An impressive result in [NN94] states that every convex set in $\mathbb{R}^n$ admits a (K, n)-self-concordant barrier, where K is a universal constant (independent of n). However, the universal barrier they provide in their proof is defined as a volume integral over an n-dimensional convex body, and is therefore difficult to evaluate in practice, even for simple sets in low-dimensional spaces. Another potential problem with this approach is that evaluating this barrier (and/or its gradient and Hessian) might take a number of arithmetic operations that grows exponentially with n, which would lead to an exponential algorithmic complexity for the short-step method, despite the polynomial iteration bound.

Another approach to find self-concordant function is to combine basic self-concordant functions using operations that are known to preserve self-concordancy (this approach is called *barrier calculus* in [NN94]). We are now going to describe two of these self-concordancy preserving operations, positive scaling and addition, and examine how the associated parameters are affected in the process.

Let us start with positive scalar multiplication.

Theorem 2.6. *Let F be a (κ, ν)-self-concordant barrier for $C \subseteq \mathbb{R}^n$ and $\lambda \in \mathbb{R}_{++}$ a positive scalar. Then (λF) is also a self-concordant barrier for C with parameters $(\frac{\kappa}{\sqrt{\lambda}}, \lambda\nu)$.*

Proof. It is clear that (λF) is also a barrier function (i.e. smoothness, strong convexity and the barrier property are obviously preserved by scaling). Looking at the restrictions $(\lambda F)_{x,h} = \lambda F_{x,h}$, we also have that

$$(\lambda F)'_{x,h} = \lambda F'_{x,h},\ (\lambda F)''_{x,h} = \lambda F''_{x,h} \text{ and } (\lambda F)'''_{x,h} = \lambda F'''_{x,h}\ .$$

Since F is (κ, ν)-self-concordant, we have (using conditions (2.5b) and (2.6b) from Theorems 2.1 and 2.2)

$$F'''_{x,h}(0) \leq 2\kappa F''_{x,h}(0)^{\frac{3}{2}} \text{ and } F'_{x,h}(0)^2 \leq \nu F''_{x,h}(0) \text{ for all } x \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n\ .$$

This is equivalent to

$$\lambda F'''_{x,h}(0) \leq 2\frac{\kappa}{\sqrt{\lambda}}(\lambda F''_{x,h}(0))^{\frac{3}{2}} \text{ and } (\lambda F'_{x,h}(0))^2 \leq \lambda\nu\lambda F''_{x,h}(0) \text{ for all } x \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n\ ,$$

which is precisely stating that (λF) is $(\frac{\kappa}{\sqrt{\lambda}}, \lambda\nu)$-self-concordant. □

This theorem show that self-concordancy is preserved by positive scalar multiplication, but that parameters κ and ν are both modified. It is interesting to note that these parameters do not occur individually in the iteration bound of Theorem 2.5 but are rather always

appearing together in the expression $\kappa\sqrt{\nu}$. This quantity, which we will call the *complexity value* of the barrier, is solely responsible for the polynomial iteration bound. Looking at what happens to it when F is scaled by λ, we find that the scaled complexity value is equal to $\frac{\kappa}{\sqrt{\nu}}\sqrt{\lambda\nu} = \kappa\sqrt{\nu}$, i.e. that the complexity value is invariant to scaling. This means *in fine* that scaling a self-concordant barrier does not influence the algorithmic complexity of the associated short-step method, a property than could reasonably be expected from the start.

Let us now examine what happens when two self-concordant barriers are added.

Theorem 2.7. *Let F be a (κ_1, ν_1)-self-concordant barrier for $C_1 \subseteq \mathbb{R}^n$ and G be a (κ_2, ν_2)-self-concordant barrier for $C_2 \subseteq \mathbb{R}^n$. Then $(F + G)$ is a self-concordant barrier for $C_1 \cap C_2$ (provided this intersection is nonempty) with parameters $(\max\{\kappa_1, \kappa_2\}, \nu_1 + \nu_2)$.*

Proof. It is straightforward to see that $(F + G)$ is a barrier function for $C_1 \cap C_2$. Looking at the restrictions $(F + G)_{x,h}$, we also have that

$$(F+G)'_{x,h} = F'_{x,h} + G'_{x,h},\ (F+G)''_{x,h} = F''_{x,h} + G''_{x,h} \text{ and } (F+G)'''_{x,h} = F'''_{x,h} + G'''_{x,h}\ .$$

We can write thus

$$\begin{aligned}(F+G)'''_{x,h} = F'''_{x,h} + G'''_{x,h} &\leq 2\kappa_1 {F''_{x,h}}^{\frac{3}{2}} + 2\kappa_2 {G''_{x,h}}^{\frac{3}{2}} \\ &\leq 2\max\{\kappa_1,\kappa_2\}({F''_{x,h}}^{\frac{3}{2}} + {G''_{x,h}}^{\frac{3}{2}}) \\ &\leq 2\max\{\kappa_1,\kappa_2\}(F''_{x,h} + G''_{x,h})^{\frac{3}{2}} = 2\max\{\kappa_1,\kappa_2\}(F+G)''_{x,h}\end{aligned}$$

(where we used for the third inequality the easily proven fact $x^{\frac{3}{2}} + y^{\frac{3}{2}} \leq (x+y)^{\frac{3}{2}}$ for $x, y \in \mathbb{R}_{++}$) and

$$\begin{aligned}\left|(F+G)'_{x,h}\right| = \left|F'_{x,h} + G'_{x,h}\right| &\leq \left|F'_{x,h}\right| + \left|G'_{x,h}\right| \\ &\leq \sqrt{\nu_1}\sqrt{F''_{x,h}} + \sqrt{\nu_2}\sqrt{G''_{x,h}} \\ &\leq \sqrt{\nu_1+\nu_2}\sqrt{F''_{x,h} + G''_{x,h}} = \sqrt{\nu_1+\nu_2}\sqrt{(F+G)''_{x,h}}\end{aligned}$$

(where we used for the third inequality the Cauchy-Schwarz inequality applied to vectors $(\sqrt{\nu_1}, \sqrt{\nu_2})$ and $(\sqrt{F''_{x,h}}, \sqrt{G''_{x,h}})$), which is precisely stating that $(F+G)$ is $(\max\{\kappa_1, \kappa_2\}, \nu_1 + \nu_2)$-self-concordant. □

2.3.2 Fixing a parameter

As mentioned above, scaling a barrier function with a positive scalar does not affect its self-concordancy, i.e. its suitability as a tool for convex optimization, and leaves its complexity value unchanged. One can thus make the decision to fix one of the two parameters κ and ν arbitrarily and only work with the corresponding subclass of barrier, without any real loss of generality. We describe now two choices of this kind that have been made in the literature.

First choice. Some authors [dJRT95, RT98, Jar89, dRT92] choose to work with the second parameter ν fixed to one. However, this choice is not made explicitly but results from the particular structure of the barrier functions that are considered. Indeed, these authors consider convex optimization problems whose feasible sets are given by a functional description like (2.1), i.e.

$$\inf_{x\in\mathbb{R}^n} c^T x \quad \text{s.t.} \quad f_i(x) \leq 0 \ \forall i \in I \ .$$

In order to apply the interior-point method methodology, a barrier function is needed and it is customary to use the logarithmic barrier as described in Note 2.1

$$F : \mathbb{R}^n \mapsto \mathbb{R} : x \mapsto F(x) = -\sum_{i\in I} \log(-f_i(x)) \ .$$

The following lemma will prove useful.

Lemma 2.1. *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a convex function and define $F : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\} : x \mapsto -\log(-f(x))$, whose effective domain is the set $C = \{x \in \mathbb{R}^n \mid f(x) < 0\}$. We have that F satisfies the second condition of self-concordancy (2.3) with parameter $\nu = 1$.*

Proof. Using the equivalent condition (2.6b) of Theorem 2.2, we have to evaluate for $x \in \operatorname{int} C$, $h \in \mathbb{R}^n$ and $t = 0$

$$F'_{x,h}(t) = -\frac{\nabla f(x+th)[h]}{f(x+th)} \text{ and } F''_{x,h}(t) = \frac{\nabla f(x+th)[h]^2 - \nabla^2 f(x+th)[h,h] f(x+th)}{f(x+th)^2} \ ,$$

which implies

$$F'_{x,h}(0)^2 = \frac{\nabla f(x)[h]^2}{f(x)^2} \leq \frac{\nabla f(x)[h]^2 - \nabla^2 f(x)[h,h] f(x)}{f(x)^2} = F''_{x,h}(0)$$

(where we have used the fact that $\nabla^2 f(x)[h,h] \geq 0$ because f is convex and $f(x) \leq 0$ because x belongs to the feasible set C), which implies that F satisfies the second self-concordancy condition (2.3) with $\nu = 1$. □

Since the complete logarithmic barrier is a sum of terms for which this lemma is applicable, we can use Theorem 2.7 to find that it satisfies the same condition with $\nu = |I| = m$, the number of constraints.

This means that we only have to check the first condition (2.2) involving κ to establish self-concordancy for the logarithmic barrier. Assuming that each individual term $-\log(-f_i(x))$ can be shown to satisfy it with $\kappa = \kappa_i$, we have that the whole logarithmic barrier is $(\max_{i\in I}\{\kappa_i\}, m)$-self-concordant, which leads to a complexity value equal to $\|\kappa\|_\infty \sqrt{m}$, where we have defined $\kappa = (\kappa_1, \kappa_2, \ldots, \kappa_m)$.

Second choice. Another arbitrary choice of self-concordance parameters that one encounters frequently in the literature consists in fixing $\kappa = 1$ in the first self-concordancy condition (2.2). This approach has been used increasingly in the recent years (see e.g.

[NN94, Ren00, Jar96]), and we propose to give here a justification of its superiority over the alternative presented above.

Let us consider the same logarithmic barrier, and suppose again that each individual term $F_i : x \mapsto -\log(-f_i(x))$ has been shown to satisfy the first self-concordancy condition (2.2) with $\kappa = \kappa_i$. Our previous discussion implies thus that F_i is $(\kappa_i, 1)$-self-concordant. Multiplying now F_i with κ_i^2, Theorem 2.6 implies that $\kappa_i^2 F_i$ is $(1, \kappa_i^2)$-self-concordant. The corresponding complete scaled logarithmic barrier

$$\tilde{F} : x \mapsto -\sum_{i \in I} \kappa_i^2 \log(-f_i(x))$$

is then $(1, \sum_{i \in I} \kappa_i^2)$-self-concordant by virtue of Theorem 2.7, which leads finally to a complexity value equal to $\sqrt{\sum_{i \in I} \kappa_i^2} = \|\kappa\|_2$. This quantity is always lower than the complexity value for the standard logarithmic barrier considered above because of the well-known norm inequality $\|\kappa\|_2 \leq \sqrt{m}\, \|\kappa\|_\infty$, which proves the superiority of this second approach (the only case where they are equivalent is when all parameters κ_i's are equal).

Note 2.4. The fundamental reason why the first approach is less efficient is that it makes us combine barriers with different κ parameters, with the consequence that only the largest value $\max_{i \in I}\{\kappa_i\}$ appears in the final complexity value (the other smaller values become completely irrelevant and do not influence the final complexity at all). The second approach avoids this situation by ensuring that κ is always equal to one, which means that κ's are equal for each combination and that the final complexity is well depending on the parameters of all the terms of the logarithmic barrier.

2.3.3 Two useful lemmas

We have seen so far how to construct self-concordant barrier by combining simpler functionals but still have no tool to prove self-concordancy of these basic barriers. The purpose of this section is to present two lemmas that can help us in that regard.

The first one deals with the second condition of self-concordancy with logarithmically homogeneous barriers [NN94].

Lemma 2.2. *Let us suppose F is a logarithmically homogeneous function with parameter α, i.e.*

$$F(tx) = F(x) - \alpha \log t \; . \tag{2.12}$$

We have that F satisfies the second condition of self-concordancy (2.3) *with parameter $\nu = \alpha$.*

Proof. This fact admits the following straightforward proof. We start by differentiating both sides of (2.12) with respect to t, to find

$$\nabla F(tx)[x] = -\frac{\alpha}{t} \; .$$

Fixing $t = 1$ gives

$$\nabla F(x)[x] = \nabla F(x)^T x = -\alpha \; . \tag{2.13}$$

Differentiating this last equality again, this time with respect to x, leads to

$$\nabla F(x) + \nabla^2 F(x)x = 0 \Leftrightarrow \nabla F(x) = -\nabla^2 F(x)x \; . \tag{2.14}$$

Looking now at the left-hand side in (2.3) we have

$$\nabla F(x)^T (\nabla^2 F(x))^{-1} \nabla F(x) = -\nabla F(x)^T (\nabla^2 F(x))^{-1} \nabla^2 F(x)x = -\nabla F(x)^T x = \alpha$$

(using successively (2.14) and (2.13)), which implies immediately that F satisfies the second condition of self-concordancy (2.3) with $\nu = \alpha$. It is worth to point out that this inequality is in this case always satisfied with equality. □

The second lemma we are going to present deals with the first self-concordancy condition. Let us first introduce two auxiliary functions r_1 and r_2, whose graphs are depicted in Figure 2.1:

$$r_1 : \mathbb{R} \mapsto \mathbb{R} : \gamma \mapsto \max\Big\{1, \frac{\gamma}{\sqrt{3 - 2/\gamma}}\Big\} \text{ and } r_2 : \mathbb{R} \mapsto \mathbb{R} : \gamma \mapsto \max\Big\{1, \frac{\gamma + 1 + 1/\gamma}{\sqrt{3 + 4/\gamma + 2/\gamma^2}}\Big\} \; .$$

Both of these functions are equal to 1 for $\gamma \leq 1$ and strictly increasing for $\gamma \geq 1$, with the asymptotic approximations $r_1(\gamma) \approx \frac{\gamma}{\sqrt{3}}$ and $r_2(\gamma) \approx \frac{\gamma+1}{\sqrt{3}}$ when γ tends to $+\infty$.

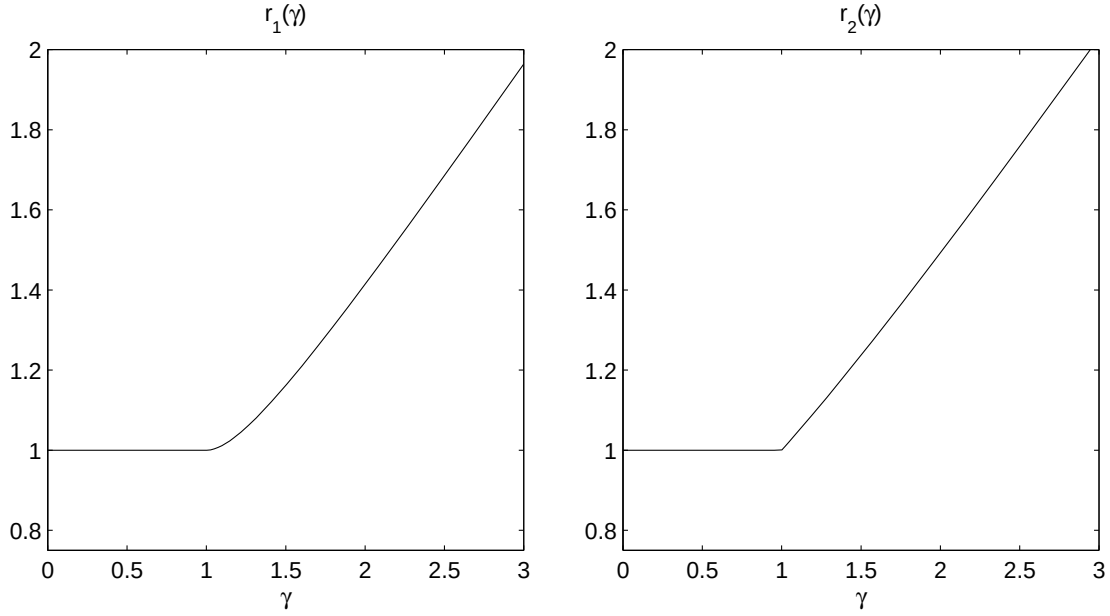


Figure 2.1: Graphs of functions r_1 and r_2

Lemma 2.3. *Let us suppose F is a convex function with effective domain $C \subseteq \mathbb{R}^n_+$ and that there exists a constant γ such that*

$$\nabla^3 F(x)[h,h,h] \leq 3\gamma \nabla^2 F(x)[h,h] \sqrt{\sum_{i=1}^n \frac{h_i^2}{x_i^2}} \text{ for all } x \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n \; . \tag{2.15}$$

We have that

$$F_1 : C \mapsto \mathbb{R} : x \mapsto F(x) - \sum_{i=1}^{n} \log x_i$$

satisfies the first condition of self-concordancy (2.2) *with parameter* $\kappa_1 = r_1(\gamma)$ *on its domain* C *and*

$$F_2 : C \times \mathbb{R} \mapsto \mathbb{R} : (x, u) \mapsto -\log\big(u - F(x)\big) - \sum_{i=1}^{n} \log x_i$$

satisfies the first condition of self-concordancy (2.2) *with parameter* $\kappa_2 = r_2(\gamma)$ *on its domain* $\operatorname{epi} F = \{(x, u) \mid F(x) \leq u\}$.

Note 2.5. A similar lemma is proved in [dJRT95], with parameters κ_1 and κ_2 both equal to $1 + \gamma$. The second result is improved in [Jar96], with κ_2 equal to $\max\{1, \gamma\}$, as a special case of a more general compatibility theory developed in [NN94]. However, it is easy to see that our result is better. Indeed, our parameters are strictly lower in all cases for F_1 and as soon as $\gamma > 1$ for r_2, with an asymptotical ratio of $\sqrt{3}$ when γ tends to $+\infty$.

Proof. We follow the lines of [dJRT95] and start with F_1: computing its second and third differentials gives

$$\nabla^2 F_1(x)[h, h] = \nabla^2 F(x)[h, h] + \sum_{i=1}^{n} \frac{h_i^2}{x_i^2} \text{ and } \nabla^3 F_1(x)[h, h, h] = \nabla^3 F(x)[h, h, h] - 2\sum_{i=1}^{n} \frac{h_i^3}{x_i^3} .$$

Introducing two auxiliary variables $a \geq 0$ and $b \geq 0$ such that

$$a^2 = \nabla^2 F[h, h] \quad \text{and} \quad b^2 = \sum_{i=1}^{n} \frac{h_i^2}{x_i^2}$$

(convexity of F guarantees that a is real), we can rewrite inequality (2.15) as

$$\nabla^3 F(x)[h, h, h] \leq 3\gamma a^2 b .$$

Combining it with the fact that

$$\left| \left(\sum_{i=1}^{n} \frac{h_i^3}{x_i^3} \right)^{\frac{1}{3}} \right| \leq \left(\sum_{i=1}^{n} \frac{|h_i|^3}{|x_i|^3} \right)^{\frac{1}{3}} \leq \left(\sum_{i=1}^{n} \frac{h_i^2}{x_i^2} \right)^{\frac{1}{2}} = b , \tag{2.16}$$

where the second inequality comes from the well-known relation $\|\cdot\|_3 \leq \|\cdot\|_2$ applied to vector $(\frac{h_1}{x_1}, \ldots, \frac{h_n}{x_n})$, we find that

$$\frac{\nabla^3 F_1(x)[h, h, h]}{2(\nabla^2 F_1(x)[h, h])^{\frac{3}{2}}} \leq \frac{3\gamma a^2 b + 2b^3}{2(a^2 + b^2)^{\frac{3}{2}}} .$$

According to (2.2), finding the best parameter κ for F_1 amounts to maximize this last quantity depending on a and b. Since $a \geq 0$ and $b \geq 0$ we can write $a = r\cos\theta$ and $b = r\sin\theta$ with $r \geq 0$ and $0 \leq \theta \leq \frac{\pi}{2}$, which gives

$$\frac{3\gamma a^2 b + 2b^3}{2(a^2 + b^2)^{\frac{3}{2}}} = \frac{3\gamma}{2}\cos^2\theta\sin\theta + \sin^3\theta = h(\theta) .$$

The derivative of h is

$$h'(\theta) = \frac{3\gamma}{2}\cos^3\theta - 3\gamma\sin^2\theta\cos\theta + 3\cos\theta\sin^2\theta = 3\cos\theta\left(\frac{\gamma}{2}\cos^2\theta + (1-\gamma)\sin^2\theta\right) .$$

When $\gamma \leq 1$, this derivative is clearly always nonnegative, which implies that the maximum is attained for the largest value of θ, which gives $h_{max} = h(\frac{\pi}{2}) = 1 = r_1(\gamma)$. When $\gamma > 1$, we easily see that h has a maximum when $\frac{\gamma}{2}\cos^2\theta + (1-\gamma)\sin^2\theta = 0$. This condition is easily seen to imply $\sin^2\theta = \frac{\gamma}{3\gamma-2}$, and h_{max} becomes

$$\begin{aligned} h_{max} = 3\frac{\gamma}{2}\cos^2\theta\sin\theta + \sin^3\theta &= \big(3(\gamma-1)+1\big)\sin^3\theta \\ &= (3\gamma-2)\left(\frac{\gamma}{3\gamma-2}\right)^{\frac{3}{2}} = \frac{\gamma}{\sqrt{3-2/\gamma}} = r_1(\gamma) . \end{aligned}$$

A similar but slightly more technical proof holds for F_2. Letting $\tilde{x} = (x,u), \tilde{h} = (h,v)$ and $G(\tilde{x}) = F(x) - u$, we have that $F_2(\tilde{x}) = -\log(-G(\tilde{x})) - \sum_{i=1}^n \log x_i$. G is easily shown to be convex and negative on $\operatorname{epi} F$, the domain of F_2. Since F and G only differ by a linear term, we also have that $\nabla^2 F(x)[h,h] = \nabla^2 G(\tilde{x})[\tilde{h},\tilde{h}]$ and $\nabla^3 F(x)[h,h,h] = \nabla^3 G(\tilde{x})[\tilde{h},\tilde{h},\tilde{h}]$. Looking now at the second differential of F_2 we find

$$\nabla^2 F_2(\tilde{x})[\tilde{h},\tilde{h}] = \frac{\nabla G(\tilde{x})[h]^2}{G(\tilde{x})^2} - \frac{\nabla^2 G(\tilde{x})[\tilde{h},\tilde{h}]}{G(\tilde{x})} + \sum_{i=1}^n \frac{h_i^2}{x_i^2} .$$

Let us define for convenience $a \in \mathbb{R}_+$, $b \in \mathbb{R}_+$ and $c \in \mathbb{R}$ with

$$a^2 = -\frac{\nabla^2 G(\tilde{x})[\tilde{h},\tilde{h}]}{G(\tilde{x})},\ b^2 = \sum_{i=1}^n \frac{h_i^2}{x_i^2} \text{ and } c = -\frac{\nabla G(\tilde{x})[h]}{G(\tilde{x})}$$

(convexity of G and the fact that it is negative on the domain of F_2 guarantee that a is real), which implies $\nabla^2 F_2(x)[\tilde{h},\tilde{h}] = a^2 + b^2 + c^2$. We can now evaluate the third differential

$$\begin{aligned} \nabla^3 F_2(\tilde{x})[\tilde{h},\tilde{h},\tilde{h}] &= -\frac{\nabla^3 G(\tilde{x})[\tilde{h},\tilde{h},\tilde{h}]}{G(\tilde{x})} + 3\frac{\nabla^2 G(\tilde{x})[\tilde{h},\tilde{h}]\nabla G(\tilde{x})[\tilde{h}]}{G(\tilde{x})^2} - 2\frac{\nabla G(\tilde{x})[\tilde{h}]^3}{G(\tilde{x})^3} - 2\sum_{i=1}^n \frac{h_i^3}{x_i^3} \\ &= -\frac{\nabla^3 G(\tilde{x})[\tilde{h},\tilde{h},\tilde{h}]}{G(\tilde{x})} + 3a^2c + 2c^3 - 2\sum_{i=1}^n \frac{h_i^3}{x_i^3} \\ &\leq -\frac{\nabla^3 G(\tilde{x})[\tilde{h},\tilde{h},\tilde{h}]}{\nabla^2 G(\tilde{x})[\tilde{h},\tilde{h}]}\frac{\nabla^2 G(\tilde{x})[\tilde{h},\tilde{h}]}{G(\tilde{x})} + 3a^2c + 2c^3 + 2b^3 \text{ using again (2.16)} \\ &= -\frac{\nabla^3 F(x)[h,h,h]}{\nabla^2 F(x)[h,h]}\frac{\nabla^2 G(\tilde{x})[\tilde{h},\tilde{h}]}{G(\tilde{x})} + 3a^2c + 2c^3 + 2b^3 \\ &\leq 3\gamma a^2 b + 3a^2c + 2c^3 + 2b^3 \text{ using condition (2.15)} . \end{aligned}$$

According to (2.2), finding the best parameter κ for F_2 amounts to maximize the following ratio

$$\frac{\nabla^3 F_2(\tilde{x})}{2(\nabla^3 F_2(\tilde{x}))^{\frac{3}{2}}} \leq \frac{3\gamma a^2 b + 3a^2c + 2c^3 + 2b^3}{2(a^2+b^2+c^2)^{\frac{3}{2}}} = \frac{\frac{3\gamma}{2}a^2b + \frac{3}{2}a^2c + c^3 + b^3}{(a^2+b^2+c^2)^{\frac{3}{2}}} .$$

Since this last quantity is homogeneous of degree 0 with respect to variables a, b and c, we can assume that $a^2 + b^2 + c^2 = 1$, which gives

$$\frac{3\gamma}{2}a^2b + \frac{3}{2}a^2c + c^3 + b^3 = \frac{3}{2}a^2(\gamma b + c) + c^3 + b^3 = \frac{3}{2}(1 - b^2 - c^2)(\gamma b + c) + b^3 + c^3 \ .$$

Calling this last quantity $m(b, c)$, we can now compute its partial derivatives with respect to b and c and find

$$\frac{\partial m}{\partial b} = -\frac{3}{2}\left((3\gamma - 2)b^2 + \gamma c^2 + 2bc - \gamma\right) \text{ and } \frac{\partial m}{\partial c} = -\frac{3}{2}\left(b^2 + c^2 + 2bc\gamma - 1\right) \ .$$

We have now to equate those two quantities to zero and solve the resulting system. We can for example write $\frac{\partial m}{\partial b} - \gamma\frac{\partial m}{\partial c} = 0$, which gives $(g - 1)b(b - c(\gamma + 1)) = 0$, and explore the resulting three cases. The solutions we find are

$$(b, c) = (0, \pm 1) \quad \text{and} \quad \left(\frac{\gamma + 1}{\sqrt{3\gamma^2 + 4\gamma + 2}}, \frac{1}{\sqrt{3\gamma^2 + 4\gamma + 2}}\right)$$

with an additional special case $b + c = 1$ when $\gamma = 1$. Plugging these values into $m(b, c)$, one finds after some computations the following potential maximum values

$$\pm 1 \quad \text{and} \quad \frac{\gamma^2 + \gamma + 1}{\sqrt{3\gamma^2 + 4\gamma + 2}} = \frac{\gamma + 1 + 1/\gamma}{\sqrt{3 + +4/\gamma + 2/\gamma^2}}$$

(and 1 in the special case $\gamma = 1$). One concludes that the maximum we seek is equal to $r_2(\gamma)$, as announced. □

While the lemma we have just proved is useful to tackle the first condition of self-concordancy (2.2), it does not say anything about the second condition (2.3). The following Corollary about the second barrier F_2 might prove useful in this respect.

Corollary 2.1. *Let F satisfy the assumptions of Lemma 2.3. Then the second barrier*

$$F_2 : C \times \mathbb{R} \mapsto \mathbb{R} : (x, u) \mapsto -\log\big(u - F(x)\big) - \sum_{i=1}^{n} \log x_i$$

is $(r_2(\gamma), n + 1)$-self-concordant.

Proof. Since $G(x, u) = F(x) - u$ is convex, $-\log(u - F(x)) = -\log(-G(x, u))$ is known to satisfy the second self-concordancy condition (2.3) with $\nu = 1$ by virtue of Lemma 2.1. Moreover, it is straightforward to check that each term $-\log x_i$ also satisfies that second condition with parameter $\nu = 1$. Using the addition Theorem 2.7 and combining with the result of Lemma 2.3, we can conclude that F_2 is $(r_2(\gamma), n + 1)$-self-concordant. □

Note 2.6. We would like to point out that no similar result can hold for the first function F_1, since we know nothing about the status of the second self-concordancy condition (2.3) on its first term $F(x)$. Indeed, taking the case of $F : \mathbb{R}_+ \mapsto \mathbb{R} : x \mapsto \frac{1}{x}$, we can check that $\nabla^2 F(x)[h, h] = 2\frac{h^2}{x^3}$ and $\nabla^3 F(x)[h, h, h] = -6\frac{h^3}{x^4}$, which implies that condition (2.15) holds with $\gamma = 1$ since

$$-6\frac{h^3}{x^4} \leq 3 \times 2\frac{h^2}{x^3}\frac{|h|}{x} \Leftrightarrow -h^3 \leq h^2\,|h|$$

is satisfied. On the other hand, the second self-concordancy condition (2.3) cannot hold for $F_1 : \mathbb{R}_+ \mapsto R : x \mapsto \frac{1}{x} - \log x$, since

$$\nabla F(x)^T (\nabla^2 F(x))^{-1} \nabla F(x) = \frac{F_1'(x)^2}{F_1''(x)} = \frac{\frac{(x+1)^2}{x^4}}{\frac{(2+x)}{x^3}} = \frac{(x+1)^2}{x(x+2)}$$

does not admit an upper bound (it tends to $+\infty$ when $x \to 0$).

To conclude this section, we mention that since condition (2.15) is invariant with respect to positive scaling of F, the results from Lemma 2.3 hold for barriers $F_{\lambda,1}(x) = \lambda F(x) - \sum_{i=1}^n \log x_i$ and $F_{\lambda,2}(x,u) = -\log(u - \lambda F(x)) - \sum_{i=1}^n \log x_i$ where λ is a positive constant.

2.4 Application to structured convex problems

In this section we rely on the work in [dJRT95], where several classes of structured convex optimization problems are shown to admit a self-concordant logarithmic barrier. However, Lemma 2.3 will allow us to improve the self-concordancy parameters and lower the resulting complexity values.

2.4.1 Extended entropy optimization

Let $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$. We consider the following problem

$$\inf_{x \in \mathbb{R}^n} \; c^T x + \sum_{i=1}^n g_i(x_i) \quad \text{s.t.} \quad Ax = b \text{ and } x \geq 0 \tag{EEO}$$

where scalar functions $g_i : \mathbb{R}_+ \mapsto \mathbb{R} : z \mapsto g_i(z)$ are required to satisfy

$$\left| g_i'''(z) \right| \leq \kappa_i \frac{g_i''(z)}{z} \; \forall z > 0 \tag{2.17}$$

(which by the way implies their convexity). This class of problems is studied in [HPY92, PY93]. Classical entropy optimization results as a special case when $g_i(x) = x \log x$ (in that case, it is straightforward to see that condition (2.17) holds with $\kappa_i = 1$).

Let us use Lemma 2.3 with $F_i : x_i \mapsto g_i(x_i)$ and $\gamma = \frac{\kappa_i}{3}$. Indeed, checking condition (2.15) amounts to write

$$h^3 g_i'''(x) \leq 3 \frac{\kappa_i}{3} h^2 g_i''(x) \frac{|h|}{x} \Leftrightarrow \frac{h}{|h|} g_i'''(x) \leq \kappa_i \frac{g_i''(x)}{x} \; ,$$

which is guaranteed by condition (2.17). Using the second barrier and Corollary 2.1, we find that

$$F_i : (x_i, u_i) \mapsto -\log\big(u_i - g_i(x_i)\big) - \log x_i$$

is $(r_2(\frac{\kappa_i}{3}), 2)$-self-concordant[2]. However, in order to use this barrier to solve problem (EEO), we need to reformulate it as

$$\inf_{x\in\mathbb{R}^n,\ u\in\mathbb{R}^n} c^T x + \sum_{i=1}^n u_i \quad \text{s.t.} \quad Ax = b,\ g_i(x_i) \le u_i\ \forall 1 \le i \le n \text{ and } x \ge 0\ ,$$

which is clearly equivalent. We are now able to write the complete logarithmic barrier

$$F : (x,u) \mapsto -\sum_{i=1}^n \log\big(u_i - g_i(x_i)\big) - \sum_{i=1}^n \log x_i\ ,$$

which is $(r_2(\frac{\max\{\kappa_i\}}{3}), 2n)$-self-concordant by virtue of Theorem 2.7. In light of Note 2.4, we can even do better with a different scaling of each term, to get

$$\tilde{F} : (x,u) \mapsto -\sum_{i=1}^n r_2(\frac{\kappa_i}{3})^2 \log\big(u_i - g_i(x_i)\big) - \sum_{i=1}^n r_2(\frac{\kappa_i}{3})^2 \log x_i$$

which is then $(1, \sqrt{2\sum_{i=1}^n r_2(\frac{\kappa_i}{3})^2})$-self-concordant. In the case of classical entropy optimization, these parameters become $(1, \sqrt{2n})$, since $r_2(\frac{1}{3}) = 1$.

2.4.2 Dual geometric optimization

Let $\{I_k\}_{k=1\ldots r}$ be a partition of $\{1, 2, \ldots, n\}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m\times n}$. The dual geometric optimization problem is (see Chapter 5 for a complete description)

$$\begin{aligned} &\inf_{x\in\mathbb{R}^n} c^T x + \sum_{k=1}^r \left[\sum_{i\in I_k} x_i \log(\frac{x_i}{\sum_{i\in I_k} x_i})\right] \\ &\text{s.t. } Ax = b \text{ and } x \ge 0 \end{aligned} \tag{GD}$$

It is shown in [dJRT95] that condition (2.15) holds for

$$F_k : (x_i)_{i\in I_k} \mapsto \sum_{i\in I_k} x_i \log(\frac{x_i}{\sum_{i\in I_k} x_i})$$

with $\gamma = 1$, so that the corresponding second barrier in Lemma 2.15 is $(1, |I_k| + 1)$-self-concordant. Using the same trick as for problem (EEO), we introduce additional variables u_k to find that the following barrier

$$F : (x,u) \mapsto \sum_{k=1}^r -\log\Big(u_k - \sum_{i\in I_k} x_i \log(\frac{x_i}{\sum_{i\in I_k} x_i})\Big) - \sum_{i=1}^n \log x_i$$

is a $(1, n + r)$-self-concordant barrier for a suitable reformulation of problem (GD).

[2]This corrects the statement in [dJRT95] where it is mentioned that $g_i(x_i) - \log x_i$, i.e. the first barrier in Lemma 2.3, is self-concordant. As it is made clear in Note 2.6, this cannot be true in general

2.4.3 l_p-norm optimization

Let $\{I_k\}_{k=1\dots r}$ be a partition of $\{1,2,\dots,n\}$, $b \in \mathbb{R}^m$, $a_i \in \mathbb{R}^m$, $f_k \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}^r$ and $p \in \mathbb{R}^n$ such that $p_i \geq 1$. The primal l_p-norm optimization problem is (see Chapter 4 for a complete description)

$$\sup_{y\in\mathbb{R}^m} b^T y \quad \text{s.t.} \quad f_k(y) \leq 0 \ \text{ for all } k = 1,\dots,r\ , \tag{Pl_p}$$

where functions $f_k : \mathbb{R}^m \mapsto \mathbb{R}$ are defined according to

$$f_k : y \mapsto \sum_{i\in I_k} \frac{1}{p_i} \left|a_i^T y - c_i\right|^{p_i} + f_k^T y - d_k\ .$$

This problem can be reformulated as

$$\sup_{y\in\mathbb{R}^m,\ s\in\mathbb{R}^n,\ t\in\mathbb{R}^n} b^T y \quad \text{s.t.} \quad \left\{ \begin{array}{rcll} \left|a_i^T y - c_i\right| & \leq & s_i & \forall i = 1,\dots,n \\ s_i & \leq & t_i^{1/p_i} & \forall i = 1,\dots,n \\ \sum_{i\in I_k} \frac{t_i}{p_i} & \leq & d_k - f_k^T y & \forall k = 1,\dots,r \end{array} \right.$$

where each of the m constraints involving an absolute value is indeed equivalent to a pair of linear constraints $a_i^T y - c_i \leq s_i$ and $c_i - a_i^T y \leq s_i$. Once again, a self-concordant function can be found for the difficult part of the constraints, i.e. the nonlinear inequality $s_i \leq t_i^{1/p_i}$. Indeed, it is straightforward to check that $f_i : t_i \mapsto -t_i^{1/p_i}$ satisfies condition (2.15) with $\gamma = \frac{2p_i-1}{3p_i} < 1$, which implies in the same fashion as above that

$$-\log\left(t_i^{1/p_i} - s_i\right) - \log t_i$$

is $(1,2)$-self-concordant. Combining with the logarithmic barrier for the linear constraints, we have that

$$-\sum_{i=1}^m \log(s_i - a_i^T y + c_i) - \sum_{i=1}^m \log(s_i + a_i^T y - c_i) - \sum_{i=1}^m \log\left(t_i^{1/p_i} - s_i\right) \dots$$
$$\dots - \sum_{i=1}^m \log t_i - \sum_{k=1}^r \log\left(d_k - f_k^T y - \sum_{i\in I_k} \frac{t_i}{p_i}\right)$$

is $(1, 4m + r)$-self-concordant for our reformulation of problem (Pl_p) (since each linear constraint is $(1,1)$-self-concordant).

Let us mention that another reformulation is presented in [dJRT95], where Lemma 2.3 is applicable to the nonlinear constraint with parameter $\gamma = \frac{|p_i-2|}{3}$, with the disadvantage of having a parameter that depends on p_i (although $r_2(\gamma)$ will stay at its lowest value as long as $p_i \leq 5$).

We conclude this section by mentioning that very similar results hold for the dual l_p-norm optimization problem, and we refer the reader to [dJRT95] for the details[3].

[3]However, we would like to point out that the nonlinear function involved in these developments is wrongly stated to satisfy condition (2.15) with $\gamma = \frac{\sqrt{2}(q_i+1)}{3q_i}$, while a correct value is $\frac{\sqrt{5q_i^2-2q_i+2}}{3q_i}$.

2.5 Concluding remarks

We gave in this chapter an overview of the theory of self-concordant functions. We would like to point out that this very powerful framework relies on two different conditions (2.2) and (2.3) and the two corresponding parameters κ and ν, each with its own purpose (see the discussion in Note 2.3). However, the important quantity is the resulting complexity value $\kappa\sqrt{\nu}$, which is of the same order as the number of iterations that is needed to reduce the barrier parameter by a constant factor by the short-step interior-point algorithm.

It is possible to scale self-concordant barriers such that one of the parameters is arbitrarily fixed without any real loss of generality. We have shown that this is best done fixing parameter κ, considering the way the complexity value is affected when adding several self-concordant barriers. However, it is in our opinion better to keep two parameters all the time, in order to simplify the presentation (for example, Lemma 2.3 intrinsically deals with the κ parameter and would need a rather awkward reformulation to be written for parameter ν with κ fixed to 1).

Several important results help us prove self-concordancy of barrier functions: Lemmas 2.1 and 2.2 deal with the second self-concordancy condition (2.3), while our improved Lemma 2.3 pertains to the first self-concordancy condition (2.2). They are indeed responsible for most of the analysis carried out in Section 2.4, which is dedicated to several classes of structured convex optimization problems. Namely, it is proved that nearly all the nonlinear (i.e. corresponding to the nonlinear constraints) terms in the associated logarithmic barriers are self-concordant with $\kappa = 1$ (the exception being extended entropy optimization, which encompasses a very broad class of problems). We would also like to mention that since all the barriers that are presented are polynomially computable, as well as their gradient and Hessian, the short-step method applied to any of these problems would need to perform a polynomial number of arithmetic operations to provide a solution with a given accuracy.

To conclude, we would like to speculate on the possibility of replacing the two self-concordancy conditions by a single inequality. Indeed, since the complexity value $\kappa\sqrt{\nu}$ is the only quantity that really matters in the final complexity result, one could imagine to consider the following inequality

$$\frac{F'''_{x,h}(0)F'_{x,h}(0)}{F''_{x,h}(0)^2} \leq 2\Gamma \text{ for all } x \in \operatorname{int} C \text{ and } h \in \mathbb{R}^n \,, \tag{2.18}$$

which is satisfied with $\Gamma = \kappa\sqrt{\nu}$ for (κ, ν)-self-concordant barriers (to see that, simply multiply condition (2.5b) by the square root of condition (2.6b)). We point out the following two intriguing facts and leave their investigation for further research:

- ◇ Condition (2.18) appears to be central in the recent theory of *self-regular* functions [PRT00], an attempt at generalizing self-concordant functions.
- ◇ Following the same principles as for (2.5d) and (2.6d), condition (2.18) can be reformulated as

$$\left(-\frac{F'_{x,h}(t)}{F''_{x,h}(t)}\right)' \leq 2\Gamma - 1 \,,$$

where the quantity on the left-hand side is the derivative of the Newton step applied to the restriction $F_{x,h}$.

Part II

Conic Duality

CHAPTER 3

Conic optimization

In this section, we describe conic optimization and the associated duality theory. Conic optimization deals with a class of problems that is essentially equivalent to the class of convex problems, i.e. minimization of a convex function over a convex set. However, formulating a convex problem in a conic way has the advantage of providing a very symmetric form for the dual problem, which often gives a new insight about its structure, especially dealing with duality.

3.1 Conic problems

The results we present in this Chapter are well-known and we will skip most of the proofs. They can be found for example in the Ph.D. thesis of Sturm [Stu97, Stu99a] with similar notations, more classical references presenting equivalent results are [SW70] and [ET76, Chapter III, Section 5]).

The basic ingredient of conic optimization is a convex cone.

Definition 3.1. A set $\mathcal{C}$ is a *cone* if and only if it is closed under nonnegative scalar multiplication, i.e.

$$x \in \mathcal{C} \Rightarrow \lambda x \in \mathcal{C} \text{ for all } \lambda \in \mathbb{R}_+ .$$

Recall that a set is convex if and only if it contains the whole segment joining any two of its points. Establishing convexity is easier for cones than for general sets, because of the following elementary theorem [Roc70a, Theorem 2.6]:

Theorem 3.1. *A cone $\mathcal{C}$ is* convex *if and only if it is closed under addition, i.e.*

$$x \in \mathcal{C} \text{ and } y \in \mathcal{C} \Rightarrow x + y \in \mathcal{C} \ .$$

In order to avoid some technical nuisances, the convex cones we are going to consider will be required to be closed, pointed and solid, according to the following definitions. A cone is said to be *pointed* if it doesn't contain any straight line passing through the origin, which can be expressed as

Definition 3.2. A cone $\mathcal{C}$ is *pointed* if and only if $\mathcal{C} \cap -\mathcal{C} = \{0\}$, where $-\mathcal{C}$ stands for the set $\{x \mid -x \in \mathcal{C}\}$

Furthermore, a cone is said to be *solid* if it has a nonempty interior, i.e. it is full-dimensional.

Definition 3.3. A cone $\mathcal{C}$ is *solid* if and only if $\operatorname{int} \mathcal{C} \neq \emptyset$ (where $\operatorname{int} S$ denotes the interior of set S).

For example, the positive orthant is a pointed and solid convex cone. A linear subspace is a convex cone that is neither pointed, nor solid (except $\mathbb{R}^n$ itself).

We are now in position to define a conic optimization problem: let $\mathcal{C} \subseteq \mathbb{R}^n$ a pointed, solid, closed convex cone. The (primal) conic optimization problem is

$$\inf_{x \in \mathbb{R}^n} \ c^T x \quad \text{s.t.} \quad Ax = b \text{ and } x \in \mathcal{C} \ , \tag{CP}$$

where $x \in \mathbb{R}^n$ is the column vector we are optimizing and the problem data is given by cone $\mathcal{C}$, a $m \times n$ matrix A and two column vectors b and c belonging respectively to $\mathbb{R}^m$ and $\mathbb{R}^n$. This problem can be viewed as the minimization of a linear function over the intersection of a convex cone and an affine subspace. As an illustration, let us mention that a linear optimization problem in the standard form (1.2) is formulated by choosing cone $\mathcal{C}$ to be the positive orthant $\mathbb{R}^n_+$.

At this stage, we would like to emphasize the fact that although our cone $\mathcal{C}$ is closed, it may happen that the infimum in (CP) is not attained (some examples of this situation will be given in Subsection 3.3).

It is well-known that the class of conic problems is equivalent to the class of convex problems, see e.g. [NN94]. However, the usual Lagrangean dual of a conic problem can be expressed very nicely in a conic form, using the notion of dual cone.

Definition 3.4. The *dual* of a cone $\mathcal{C} \subseteq \mathbb{R}^n$ is defined by

$$\mathcal{C}^* = \left\{ x^* \in \mathbb{R}^n \mid x^T x^* \geq 0 \text{ for all } x \in \mathcal{C} \right\} \ .$$

For example, the dual of $\mathbb{R}^n_+$ is $\mathbb{R}^n_+$ itself. We say it is *self-dual*. Another example is the dual of the linear subspace L, which is $L^* = L^\perp$, the linear subspace orthogonal to L (note that in that case the inequality of Definition 3.4 is always satisfied with equality).

The following theorem stipulates that the dual of a closed convex cone is always a closed convex cone [Roc70a, Theorem 14.1].

Theorem 3.2. *If $\mathcal{C}$ is a closed convex cone, its dual $\mathcal{C}^*$ is another closed convex cone. Moreover, the dual $(\mathcal{C}^*)^*$ of $\mathcal{C}^*$ is equal to $\mathcal{C}$.*

Closedness is essential for $(\mathcal{C}^*)^* = \mathcal{C}$ to hold (without the closedness assumption on $\mathcal{C}$, we only have $(\mathcal{C}^*)^* = \operatorname{cl}\mathcal{C}$ where $\operatorname{cl} S$ denotes the closure of set S [Roc70a, Theorem 14.1]). The additional notions of solidness and pointedness also behave well when taking the dual of a convex cone: indeed, these two properties are dual to each other [Stu97, Corollary 2.1], which allows us to state the following theorem:

Theorem 3.3. *If $\mathcal{C}$ is a solid, pointed, closed convex cone, its dual $\mathcal{C}^*$ is another solid, pointed, closed convex cone.*

The dual of our primal conic problem (CP) is defined by

$$\sup_{y\in\mathbb{R}^m, s\in\mathbb{R}^n} b^T y \quad \text{s.t.} \quad A^T y + s = c \text{ and } s \in \mathcal{C}^* \,, \tag{CD}$$

where $y \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are the column vectors we are optimizing, the other quantities A, b and c being the same as in (CP). It is immediate to notice that this dual problem has the same kind of structure as the primal problem, i.e. it also involves optimizing a linear function over the intersection of a convex cone and an affine subspace. The only differences are the direction of the optimization (maximization instead of minimization) and the way the affine subspace is described (it is a translation of the range space of A^T, while primal involved a translation of the null space of A). It is also easy to show that the dual of this dual problem is equivalent to the primal problem, using the fact that $(\mathcal{C}^*)^* = \mathcal{C}$.

One of the reasons the conic formulation (CP) is interesting is the fact that we may view the constraint $x \in \mathcal{C}$ as a generalization of the traditional nonnegativity constraint $x \geq 0$ of linear optimization. Indeed, let us define the relation $\succeq$ on $\mathbb{R}^n \times \mathbb{R}^n$ according $x \succeq y \Leftrightarrow x - y \in \mathcal{C}$. This relation is reflexive, since $x \succeq x \Leftrightarrow 0 \in \mathcal{C}$ is always true. It is also transitive, since we have

$$x \succeq y \text{ and } y \succeq z \Leftrightarrow x - y \in \mathcal{C} \text{ and } y - z \in \mathcal{C} \Rightarrow (x - y) + (y - z) = x - z \in \mathcal{C} \Leftrightarrow x \succeq z$$

(where we used the fact that a convex cone is closed under addition, see Theorem 3.1). Finally, using the fact that $\mathcal{C}$ is pointed, we can write

$$x \succeq y \text{ and } y \succeq x \Leftrightarrow x - y \in \mathcal{C} \text{ and } -(x - y) \in \mathcal{C} \Rightarrow x - y = 0 \Rightarrow x = y \,,$$

which shows that relation $\succeq$ is antisymmetric and is thus a partial order on $\mathbb{R}^n \times \mathbb{R}^n$. Defining $\succeq^*$ to be the relation induced by the dual cone $\mathcal{C}^*$, we can rewrite our primal-dual pair (CP)–(CD) as

$$\begin{aligned} &\inf_{x\in\mathbb{R}^n} c^T x \qquad \text{s.t.} \qquad Ax = b \text{ and } x \succeq 0 \\ &\sup_{y\in\mathbb{R}^m} b^T y \qquad \text{s.t.} \qquad c \succeq^* A^T y \,, \end{aligned}$$

which looks very much like a generalization of the primal-dual pair of linear optimization problems (LP)–(LD').

For example, one of the most versatile cones used in convex optimization is the *positive semidefinite* cone $\mathbb{S}^n_+$.

Definition 3.5. The positive semidefinite cone $\mathbb{S}^n_+$ is a subset of $\mathbb{S}^n$, the set of symmetric $n \times n$ matrices. It consists of all positive semidefinite matrices, i.e.

$$M \in \mathbb{S}^n_+ \Leftrightarrow z^T M z \geq 0 \ \forall z \in \mathbb{R}^n \Leftrightarrow \lambda(M) \geq 0$$

where $\lambda(M)$ denotes the vector of eigenvalues of M.

It is straightforward to check that $\mathbb{S}^n_+$ is a closed, solid, pointed convex cone. A conic optimization problem of the form (CP) or (CD) that uses a cone of the type $\mathbb{S}^n_+$ is called a *semidefinite problem*[1]. This cone provides us with the ability to model many more types of constraints than a linear problem (see e.g. [VB96] or Appendix A for an application to classification).

3.2 Duality theory

The two conic problems of this primal-dual pair are strongly related to each other, as demonstrated by the duality theorems stated in this section. Conic optimization enjoys the same kind of rich duality theory as linear optimization, albeit with some complications regarding the strong duality property.

Theorem 3.4 (Weak duality). *Let x a feasible (i.e. satisfying the constraints) solution for* (CP)*, and (y, s) a feasible solution for* (CD)*. We have*

$$b^T y \leq c^T x \ ,$$

equality occurring if and only if the following orthogonality condition *is satisfied:*

$$x^T s = 0 \ .$$

This theorem shows that any primal (resp. dual) feasible solution provides an upper (resp. lower) bound for the dual (resp. primal) problem. Its proof is quite easy to obtain: elementary manipulations give

$$c^T x - b^T y = x^T c - (Ax)^T y = x^T (A^T y + s) - x^T A^T y = x^T s \ ,$$

this last inner product being always nonnegative because of $x \in \mathcal{C}$, $s \in \mathcal{C}^*$ and Definition 3.4 of the dual cone $\mathcal{C}^*$. The nonnegative quantity $x^T s = c^T x - b^T y$ is called the *duality gap*.

Obviously, a pair (x, y) with a zero duality gap must be optimal. It is well known that the converse is true in the case of linear optimization, i.e. that all primal-dual pairs of optimal

[1] The fact that our feasible points are in this case matrices instead of vectors calls for some explanation. Since our convex cones are supposed to belong to a real vector space, we have to consider that $\mathbb{S}^n$, the space of symmetric matrices, is isomorphous to $\mathbb{R}^{n(n+1)/2}$. In that setting, an expression such as the objective function $c^T x$, where c and x belong to $\mathbb{R}^{n(n+1)/2}$, is to be understood as the inner product of the corresponding symmetric matrices C and X in the space $\mathbb{S}^n$, which is defined by $\langle C, X \rangle = \operatorname{trace} CX$. Moreover, A can be seen in this case as an application (more precisely a tensor) that maps $\mathbb{S}^n$ to $\mathbb{R}^m$, while A^T is the adjoint of A which maps $\mathbb{R}^m$ to $\mathbb{S}^n$.

solutions for a linear optimization problem have a zero duality gap (see Section 1.2.1), but this is not in general the case for conic optimization.

Denoting by p^* and d^* the optimum objective values of problems (CP) and (CD), the previous theorem implies that $p^* - d^* \geq 0$, a nonnegative quantity which will be called the *optimal duality gap*. Under certain circumstances, it can be proved to be equal to zero, which shows that the optimum values of problems (CP) and (CD) are equal. Before describing the conditions guaranteeing such a situation, called *strong* duality, we need to introduce the notion of strictly feasible point.

Definition 3.6. A point x (resp. (y, s)) is said to be *strictly feasible* for the primal (resp. dual) problem if and only if it is feasible and belongs to the interior of the cone $\mathcal{C}$ (resp. $\mathcal{C}^*$), i.e.

$$Ax = b \text{ and } x \in \operatorname{int} \mathcal{C} \quad (\text{resp. } A^T y + s = c \text{ and } s \in \operatorname{int} \mathcal{C}^*) \ .$$

Strictly feasible points, sometimes called *Slater points*, are also said to satisfy the *interior-point* or *Slater condition*. Moreover, we will say that the primal (resp. dual) problem is *unbounded* if $p^* = -\infty$ (resp. $d^* = +\infty$), that it is *infeasible* if there is no feasible solution, i.e. when $p^* = +\infty$ (resp. $d^* = -\infty$), and that it is *solvable* or *attained* if the optimum objective value p^* (resp. d^*) is achieved by at least one feasible primal (resp. dual) solution.

Theorem 3.5 (Strong duality). *If the dual problem* (CD) *admits a strictly feasible solution, we have either*

- ◇ *an infeasible primal problem* (CP) *if the dual problem* (CD) *is unbounded, i.e.* $p^* = d^* = +\infty$
- ◇ *a feasible primal problem* (CP) *if the dual problem* (CD) *is bounded. Moreover, in this case, the primal optimum is finite and attained with a zero duality gap, i.e. there is at least an optimal feasible solution* x^* *such that* $c^T x^* = p^* = d^*$.

The first case in this theorem (see e.g. [Stu97, Theorem 2.7] for a proof) is a simple consequence of Theorem 3.4, which is also valid in the absence of a Slater point for the dual, as opposed to the second case which relies on the existence of such a point. It is also worth to mention that boundedness of the dual problem (CD), defining the second case, is implied by the existence of a feasible primal solution, because of the weak duality theorem (however, the converse implication is not true in general, since a bounded dual problem can admit an infeasible primal problem ; an example of this situation is provided in Subsection 5.3.4).

This theorem is important, because it provides us with way to identify when both the primal and the dual problems have the same optimal value, and when this optimal value is attained by one of the problems. Obviously, this result can be dualized, meaning that the existence of a strictly feasible primal solution implies a zero duality gap and dual attainment. The combination of these two theorems leads to the following well-known corollary:

Corollary 3.1. *If both the primal and the dual problems admit a strictly feasible point, we have a zero duality gap and attainment for both problems, i.e. the same finite optimum objective value is attained for both problems.*

When the dual problem has no strictly feasible point, nothing can be said about the duality gap (which can happen to be strictly positive) and about attainment of the primal optimum objective value. However, even in this situation, we can prove an alternate version of the strong duality theorem involving the notion of primal problem subvalue. The idea behind this notion is to allow a small constraint violation in the infimum defining the primal problem (CP).

Definition 3.7. The *subvalue* of primal problem (CP) is given by

$$p^- = \lim_{\epsilon \to 0^+} \left[\inf_x c^T x \quad \text{s.t.} \quad \|Ax - b\| < \epsilon \text{ and } x \in \mathcal{C} \right]$$

(a similar definition is holding for the dual subvalue d^-).

It is readily seen that this limit always exists (possibly being $+\infty$), because the feasible region of the infimum shrinks as ϵ tends to zero, which implies that its optimum value is a nonincreasing function of ϵ. Moreover, the inequality $p^- \leq p^*$ holds, because all the feasible regions of the infima defining p^- as ϵ tends to zero are larger than the actual feasible region of problem (CP).

The case $p^- = +\infty$, which implies that primal problem (CP) is infeasible (since we have then $p^* \geq p^- = +\infty$), is called primal *strong infeasibility*, and essentially means that the affine subspace defined by the linear constraints $Ax = b$ is strongly separated from cone $\mathcal{C}$. We are now in position to state the following alternate strong duality theorem:

Theorem 3.6 (Strong duality, alternate version). *We have either*

- $p^- = +\infty$ *and* $d^* = -\infty$ *when primal problem* (CP) *is strongly infeasible and dual problem* (CD) *is infeasible.*
- $p^- = d^*$ *in all other cases.*

This theorem (see e.g. [Stu97, Theorem 2.6] for a proof) states that there is no duality gap between p^- and d^*, except in the rather exceptional case of primal strong infeasibility and dual infeasibility. Note that the second case covers situations where the primal problem is infeasible but not strongly infeasible (i.e. $p^- < p^* = +\infty$).

To conclude this section, we would like to mention the fact that all the properties and theorems described in this section can be easily extended to the case of several conic constraints involving disjoint sets of variables.

Note 3.1. Namely, having to satisfy the constraints $x^i \in \mathcal{C}^i$ for all $i \in \{1, 2, \ldots, k\}$, where $\mathcal{C}^i \subseteq \mathbb{R}^{n_i}$, we will simply consider the Cartesian product of these cones $\mathcal{C} = \mathcal{C}^1 \times \mathcal{C}^2 \times \cdots \times \mathcal{C}^k \subseteq \mathbb{R}^{\sum_{i=1}^k n_i}$ and express all these constraints simultaneously as $x \in \mathcal{C}$ with $x = (x^1, x^2, \ldots, x^k)$. The dual cone of $\mathcal{C}$ will be given by

$$\mathcal{C}^* = (\mathcal{C}^1)^* \times (\mathcal{C}^2)^* \times \cdots \times (\mathcal{C}^k)^* \subseteq \mathbb{R}^{\sum_{i=1}^k n_i} ,$$

as implied by the following theorem:

Theorem 3.7. *Let $\mathcal{C}^1$ and $\mathcal{C}^2$ two closed convex cones, and $\mathcal{C} = \mathcal{C}^1 \times \mathcal{C}^2$ their Cartesian product. Cone $\mathcal{C}$ is also a closed convex cone, and its dual $\mathcal{C}^*$ is given by*

$$\mathcal{C}^* = (\mathcal{C}^1)^* \times (\mathcal{C}^2)^* .$$

3.3 Classification of conic optimization problems

In this last section, we describe all the possible types of conic programs with respect to feasibility, attainability of the optimum and optimal duality gap, and provide corresponding examples.

Given our standard primal conic program (CP), we define

$$\mathcal{F}_+ = \{x \in \mathbb{R}^n \mid Ax = b \text{ and } x \in \mathcal{C}\}$$

to be its feasible set and $\delta = \text{dist}(\mathcal{C}, L)$ the minimum distance between cone $\mathcal{C}$ and the affine subspace $L = \{x \mid Ax = b\}$ defined by the linear constraints. We also call $\mathcal{F}_{++}$ the set of strictly feasible solutions of (CP), i.e.

$$\mathcal{F}_{++} = \{x \in \mathbb{R}^n \mid Ax = b \text{ and } x \in \text{int}\,\mathcal{C}\}\ .$$

3.3.1 Feasibility

First of all, the distinction between feasible and infeasible conic problems is not as clear-cut as for linear optimization. We have the following cases[2]

- ◇ A conic program is infeasible. This means the feasible set $\mathcal{F}_+ = \emptyset$, and that $p^* = +\infty$. But we have to distinguish two subcases
  - $\delta = 0$, which means an infinitesimal perturbation of the problem data may transform the program into a feasible one. We call the program *weakly infeasible*(‡). This corresponds to the case of a finite subvalue, i.e. $p^- < p^* = +\infty$.
  - $\delta > 0$, which corresponds to the usual infeasibility as for linear optimization. We call the program *strongly infeasible*, which corresponds to an infinite subvalue $p^- = p^* = +\infty$.
- ◇ A conic program is feasible, which means $\mathcal{F}_+ \neq \emptyset$ and $p^* < +\infty$ (and thus $\delta = 0$). We also distinguish two subcases
  - $\mathcal{F}_{++} = \emptyset$, which implies that all feasible points belong to the boundary of the feasible set $\mathcal{F}_+$ (this corresponds indeed to the case where the affine subspace L is tangent to the cone $\mathcal{C}$). This also means that an infinitesimal perturbation of the problem data can make the program infeasible. We call the program *weakly feasible*.
  - $\mathcal{F}_{++} \neq \emptyset$. We call the program *strongly feasible*. This means there exists at least one feasible solution belonging to the interior of $\mathcal{C}$, which is the main hypothesis of the strong duality Theorem 3.5.

It is possible to characterize these situations by looking at the existence of certain types of directions in the dual problem (level direction, improving direction, improving direction sequence, see [Stu97]). Let us now illustrate these four situations with an example.

[2] In the following, we'll mark with a (‡) the cases which never happen in the case of linear optimization.

Example 3.1. Let us choose

$$\mathcal{C} = \mathbb{S}^2_+ \quad \text{and} \quad x = \begin{pmatrix} x_1 & x_3 \\ x_3 & x_2 \end{pmatrix} .$$

We have that $x \in \mathcal{C} \Leftrightarrow x_1 \geq 0, x_2 \geq 0$ and $x_1 x_2 \geq x_3^2$.

If we add the linear constraint $x_3 = 1$, the feasible set becomes the epigraph of the positive branch of the hyperbola $x_1 x_2 = 1$, i.e. $\mathcal{F}_+ = \{(x_1, x_2) \mid x_1 \geq 0 \text{ and } x_1 x_2 \geq 1\}$ as depicted on Figure 3.1.

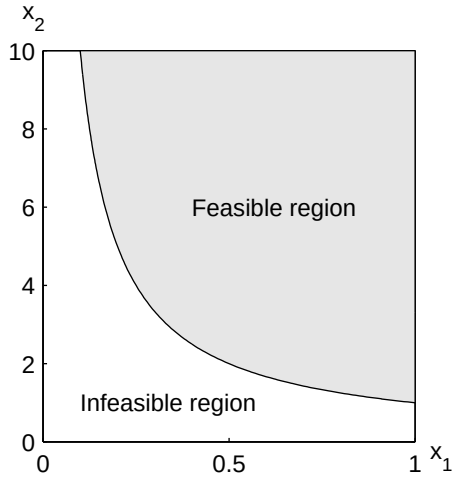


Figure 3.1: Epigraph of the positive branch of the hyperbola $x_1 x_2 = 1$

This problem is strongly feasible.

- ◇ If we add another linear constraint $x_1 = -1$, we get a strongly infeasible problem (since x_1 must be positive).
- ◇ If we add $x_1 = 0$, we get a weakly infeasible problem (since the distance between the axis $x_1 = 0$ and the hyperbola is zero but x_1 still must be positive).
- ◇ Finally, adding $x_1 + x_2 = 2$ leads to a weakly feasible problem (because the only feasible point, $x_1 = x_2 = x_3 = 1$, does not belong to the interior of $\mathcal{C}$).

3.3.2 Attainability

Let us denote by $\mathcal{F}^*$ the set of optimal solutions, i.e. feasible solutions with an objective equal to p^*

$$\mathcal{F}^* = \mathcal{F}_+ \cap \{x \in \mathbb{R}^n \mid c^T x = p^*\}$$

We have the following distinction regarding attainability of the optimum

- ◇ A conic program is *solvable* if $\mathcal{F}^* \neq \emptyset$.

- ◇ A conic program is *unsolvable* if $\mathcal{F}^* = \emptyset$, but we have two subcases
  - If $p^* = -\infty$, the program is *unbounded* (this is the only possibility in the case of linear optimization).
  - If p^* is finite, we have a feasible unsolvable bounded program (‡). This situation happens when the infimum defining p^* is not attained, i.e. there exists feasible solution with objective value arbitrarily close to p^* but no optimal solution.

Let us examine a little further the second situation. In this case, we have a sequence of feasible solutions whose objective value tends to p^*, but no optimal solution. This implies that at least one of the variables in this sequence of feasible solutions tends to infinity. Indeed, if it was not the case, that sequence would be bounded, and since the feasible set $\mathcal{F}$ is closed (it is the intersection of a closed cone and a affine subspace, which is also closed), its limit would also belong to the feasible set, hence would be a feasible solution with objective value p^*, i.e. an optimal solution, which is a contradiction.

Example 3.2. Let us consider the same strongly feasible problem as in Example 3.1 (epigraph of an hyperbola).

- ◇ If we choose a linear objective equal to $x_1 + x_2$,, $\mathcal{F}^*$ is reduced to the unique point $(x_1, x_2, x_3) = (1, 1, 1)$, and the problem is solvable ($p^* = 2$).
- ◇ If we choose another objective equal to $-x_1 - x_2$, $\mathcal{F}^* = \emptyset$ because $p^* = -\infty$, and the problem is unbounded.
- ◇ Finally, choosing x_1 as objective function leads to an unsolvable bounded problem: p^* is easily seen to be equal to zero but $\mathcal{F}^* = \emptyset$ because there is no feasible solution with $x_1 = 0$ since the product $x_1 x_2$ has to be greater than 1.

3.3.3 Optimal duality gap

Finally, we state the various possibilities about the *optimal duality gap*, which is equal to $p^* - d^*$:

- ◇ The optimal duality gap is strictly positive (‡)
- ◇ The optimal duality gap is zero but there is no optimal solution pair. In this case, there exists pairs (x, y) with an arbitrarily small duality gap (which means that the optimum is not attained for at least one of the two programs (LP) and (LD)) (‡)
- ◇ An optimal solution pair (x, y) has a zero duality gap, as for linear optimization

Of course, the first two cases can be avoided if we require our problem to satisfy the Slater condition. We can alternatively work with the subvalue p^-, for which there is no duality gap except when both problems are infeasible.

Example 3.3. The first problem described in Example 3.2 has its optimal value equal to $p^* = 2$. Its data can be described as

$$c = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \; A : \mathbb{S}^2 \mapsto \mathbb{R} : \begin{pmatrix} x_1 & x_3 \\ x_3 & x_2 \end{pmatrix} \mapsto x_3 \text{ and } b = 1 \; .$$

Using the fact that the adjoint of A can be written as[3]

$$A^T : \mathbb{R} \mapsto \mathbb{S}^2 : y_1 \mapsto \begin{pmatrix} 0 & y_1/2 \\ y_1/2 & 0 \end{pmatrix}$$

and the dual formulation (CD), we can state the dual as

$$\sup y_1 \quad \text{s.t.} \quad \begin{pmatrix} 0 & y_1/2 \\ y_1/2 & 0 \end{pmatrix} + \begin{pmatrix} s_1 & s_3 \\ s_3 & s_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \begin{pmatrix} s_1 & s_3 \\ s_3 & s_2 \end{pmatrix} \in \mathbb{S}^2_+$$

or equivalently, after eliminating the s variables,

$$\sup y_1 \quad \text{s.t.} \quad \begin{pmatrix} 1 & -y_1/2 \\ -y_1/2 & 1 \end{pmatrix} \in \mathbb{S}^2_+ \; .$$

The optimal value d^* of this problem is equal to 2, because the semidefinite constraint is equivalent to $y_1^2 \leq 4$), and the optimal duality gap $p^* - d^*$ is zero as expected.

Changing the primal objective to $c = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, we get an unsolvable bounded problem

$$\inf x_1 \quad \text{s.t.} \quad x_3 = 1 \text{ and } x_1 x_2 \geq 1$$

whose optimal value is $p^* = 0$ but is not attained. The dual becomes

$$\sup y_1 \quad \text{s.t.} \quad \begin{pmatrix} 1 & -y_1/2 \\ -y_1/2 & 0 \end{pmatrix} \in \mathbb{S}^2_+$$

which admits only one feasible solution, namely $y_1 = 0$, and has thus an optimal value $d^* = 0$. In this case, the optimal duality gap is zero but is not attained (because the primal problem is unsolvable).

Finally, we give here an example where the optimal duality gap is nonzero. Choosing a nonnegative parameter λ and

$$\mathcal{C} = \mathbb{S}^3_+, \; c = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & \lambda \end{pmatrix}, \; A : \mathbb{S}^3 \mapsto \mathbb{R}^2 : \begin{pmatrix} x_1 & x_4 & x_5 \\ x_4 & x_2 & x_6 \\ x_5 & x_6 & x_3 \end{pmatrix} \mapsto \begin{pmatrix} x_3 + x_4 \\ x_2 \end{pmatrix} \text{ and } b = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \; ,$$

we have for the primal

$$\inf \lambda x_3 - 2x_4 \quad \text{s.t.} \quad x_3 + x_4 = 1, \; x_2 = 0 \text{ and } \begin{pmatrix} x_1 & x_4 & x_5 \\ x_4 & x_2 & x_6 \\ x_5 & x_6 & x_3 \end{pmatrix} \in \mathbb{S}^3_+ \; .$$

[3]To check this, simply write $\langle Ax, y \rangle = \langle x, A^T y \rangle$, where the first inner product is the usual dot product on $\mathbb{R}^n$ but the second inner product is the trace inner product on $\mathbb{S}^n$.

The fact that $x_2 = 0$ implies $x_4 = x_6 = 0$, which in turn implies $x_3 = 1$. We have thus that all solutions have the form

$$\begin{pmatrix} x_1 & 0 & x_5 \\ 0 & 0 & 0 \\ x_5 & 0 & 1 \end{pmatrix}$$

which is feasible as soon as $x_1 \geq x_5^2$. All these feasible solutions have an objective value equal λ, and hence are all optimal: we have $p^* = \lambda$. Using the fact that the adjoint of A is

$$A^T : \mathbb{R}^2 \mapsto \mathbb{S}^3 : \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \mapsto \begin{pmatrix} 0 & y_1/2 & 0 \\ y_1/2 & y_2 & 0 \\ 0 & 0 & y_1 \end{pmatrix}$$

we can write the dual (after eliminating the s variables with the linear equality constraints) as

$$\sup y_1 \quad \text{s.t.} \quad \begin{pmatrix} 0 & -1 - y_1/2 & 0 \\ -1 - y_1/2 & -y_2 & 0 \\ 0 & 0 & \lambda - y_1 \end{pmatrix} \in \mathbb{S}^3_+$$

The above matrix can only be positive semidefinite if $y_1 = -2$. In that case, any nonnegative value for y_2 will lead to a feasible solution with an objective equal to -2, i.e. all these solutions are optimal and $d^* = -2$. The optimal duality gap is equal to $p^* - d^* = \lambda + 2$, which is strictly positive fear all values of λ. Note that in this case, as expected from the theory, none of the two problems satisfies the Slater condition since every feasible primal or dual solution has at least a zero on its diagonal, which implies a zero eigenvalue and hence that it does not belong to the interior of $\mathbb{S}^3_+$.

CHAPTER 4

l_p-norm optimization

In this chapter, we formulate the l_p-norm optimization problem as a conic optimization problem, derive its standard duality properties and show it can be solved in polynomial time.

We first define an *ad hoc* closed convex cone $\mathcal{L}^p$, study its properties and derive its dual. This allows us to express the standard l_p-norm optimization primal problem as a conic problem involving $\mathcal{L}^p$. Using the theory of conic duality described in Chapter 3 and our knowledge about $\mathcal{L}^p$, we proceed to derive the dual of this problem and prove the well-known regularity properties of this primal-dual pair, i.e. zero duality gap and primal attainment. Finally, we prove that the class of l_p-norm optimization problems can be solved up to a given accuracy in polynomial time, using the framework of interior-point algorithms and self-concordant barriers.

4.1 Introduction

l_p-norm optimization problems form an important class of convex problems, which includes as special cases linear optimization, quadratically constrained convex quadratic optimization and l_p-norm approximation problems.

A few interesting duality results are known for l_p-norm optimization. Namely, a pair of feasible primal-dual l_p-norm optimization problems satisfies the weak duality property, which is a mere consequence of convexity, but can also be shown to satisfy two additional properties that cannot be guaranteed in the general convex case: the optimum duality gap is equal to

zero and at least one feasible solution attains the optimum primal objective. These results were first presented by Peterson and Ecker [PE70a, PE67, PE70b] and later greatly simplified by Terlaky [Ter85], using standard convex duality theory (e.g. the convex Farkas theorem).

The aim of this chapter is to derive these results in a completely different setting, using the machinery of conic convex duality described in Chapter 3. This new approach has the advantage of further simplifying the proofs and giving some insight about the reasons why this class of problems has better properties than a general convex problem. We also show that this class of optimization problems can be solved up to a given accuracy in polynomial time, using the theory of self-concordant barriers in the framework of interior-point algorithms (see Chapter 2).

4.1.1 Problem definition

Let us start by introducing the primal l_p-norm optimization problem [PE70a, Ter85], which is basically a slight modification of a linear optimization problem where the use of l_p-norms applied to linear terms is allowed within the constraints. In order to state its formulation in the most general setting, we need to introduce the following sets: let $K = \{1, 2, \ldots, r\}$, $I = \{1, 2, \ldots, n\}$ and let $\{I_k\}_{k \in K}$ be a partition of I into r classes, i.e. satisfying

$$\cup_{k \in K} I_k = I \text{ and } I_k \cap I_l = \emptyset \text{ for all } k \neq l \ .$$

The problem data is given by two matrices $A \in \mathbb{R}^{m \times n}$ and $F \in \mathbb{R}^{m \times r}$ (whose columns will be denoted by a_i, $i \in I$ and f_k, $k \in K$) and four column vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}^r$ and $p \in \mathbb{R}^n$ such that $p_i > 1 \ \forall i \in I$. Our primal problem consists in optimizing a linear function of a column vector $y \in \mathbb{R}^m$ under a set of constraints involving l_p-norms of linear forms, and can be written as

$$\sup \ b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} \frac{1}{p_i} \left| c_i - a_i^T y \right|^{p_i} \leq d_k - f_k^T y \quad \forall k \in K \ . \tag{Pl_p}$$

It is readily seen that this formulation is quite general. Indeed,

- ◇ linear optimization problems can be modelled by taking $n = 0$ (and thus $I_k = \emptyset \ \forall k \in K$), which gives
$$\sup \ b^T y \quad \text{s.t.} \quad F^T y \leq d \ ,$$
- ◇ problems of approximation in l_p-norm correspond to the case $f_k = 0 \ \forall k \in K$, described in [PE70a, Ter85] and [NN94, Section 6.3.2],
- ◇ a convex quadratic constraint can be modelled with a constraint involving an l_2-norm. Indeed, $\frac{1}{2} y^T Q y + f^T y + g \leq 0$ (where Q is positive semidefinite) is equivalent to $\frac{1}{2} \left\| H^T y \right\|^2 \leq -f^T y - g$, where H is a $m \times s$ matrix such that $Q = HH^T$ (whose columns will be denoted by h_i), and can be modelled as
$$\sum_{i=1}^{s} \frac{1}{2} \left| h_i^T y \right|^2 \leq -g - f^T y \ ,$$

which has the same form as one constraint of problem (Pl_p) with $p_i = 2$ and $c_i = 0$. This implies that linearly and quadratically constrained convex quadratic optimization problems can be modelled as l_p-norm optimization problems (since a convex quadratic objective can be modelled using an additional variable, a linear objective and a convex quadratic constraint).

Defining a vector $q \in \mathbb{R}^n$ such that $\frac{1}{p_i} + \frac{1}{q_i} = 1$ for all $i \in I$, the dual problem for (Pl_p) can be defined as (see e.g. [Ter85])

$$\inf\ \psi(x,z) = c^T x + d^T z + \sum_{k \in K | z_k > 0} z_k \sum_{i \in I_k} \frac{1}{q_i} \left| \frac{x_i}{z_k} \right|^{q_i} \text{ s.t. } \begin{cases} Ax + Fz = b \text{ and } z \geq 0\,, \\ z_k = 0 \Rightarrow x_i = 0\ \forall i \in I_k\,. \end{cases} \quad (\text{D}l_p)$$

We note that a special convention has been taken to handle the case when one or more components of z are equal to zero: the associated terms are left out of the first sum (to avoid a zero denominator) and the corresponding components of x have to be equal to zero. When compared with the primal problem (Pl_p), this problem has a simpler feasible region (mostly defined by linear equalities and nonnegativity constraints) at the price of a highly nonlinear (but convex) objective.

4.1.2 Organization of the chapter

The rest of this chapter is organized as follows. In order to use the setting of conic optimization, we define in Section 4.2 an appropriate convex cone that will allow us to express l_p-norm optimization problems as conic programs. We also study some aspects of this cone (closedness, interior, dual). We are then in position to formulate the primal-dual pair (Pl_p)–(Dl_p) using a conic formulation and apply in Section 4.3 the general duality theory for conic optimization, in order to prove the above-mentioned duality results about l_p-norm optimization. Section 4.4 deals with algorithmic complexity issues and presents a self-concordant barrier construction for our problem. We conclude with some remarks in Section 4.5.

4.2 Cones for l_p-norm optimization

Let us now introduce the $\mathcal{L}^p$ cone, which will allow us to give a conic formulation of l_p-norm optimization problems.

4.2.1 The primal cone

Definition 4.1. Let $n \in \mathbb{N}$ and $p \in \mathbb{R}^n$ with $p_i > 1$. We define the following set

$$\mathcal{L}^p = \left\{ (x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \sum_{i=1}^{n} \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} \leq \kappa \right\}$$

using in the case of a zero denominator the following convention:

$$\frac{|x_i|}{0} = \begin{cases} +\infty & \text{if } x_i \neq 0 \ , \\ 0 & \text{if } x_i = 0 \ . \end{cases}$$

This convention means that if $(x, \theta, \kappa) \in \mathcal{L}^p$, $\theta = 0$ implies $x = 0^n$. We start by proving that $\mathcal{L}^p$ is a convex cone.

Theorem 4.1. *$\mathcal{L}^p$ is a convex cone.*

Proof. Let us first introduce the following function

$$f_p : \mathbb{R}^n \times \mathbb{R}_+ \mapsto \mathbb{R}_+ \cup \{+\infty\} : (x, \theta) \mapsto \sum_{i=1}^{n} \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} \ .$$

With the convention mentioned above, its effective domain is $\mathbb{R}^n \times \mathbb{R}_{++} \cup 0^n \times 0$. It is straightforward to check that f_p is positively homogeneous, i.e. $f_p(\lambda x, \lambda\theta) = \lambda f_p(x, \theta)$ for $\lambda \geq 0$. Moreover, f_p is subadditive, i.e. $f_p(x + x', \theta + \theta') \leq f_p(x, \theta) + f_p(x', \theta')$. In order to show it, we only need to prove the following inequality for all $x, x' \in \mathbb{R}$ and $\theta, \theta' \in \mathbb{R}_+$:

$$\frac{|x|^{p_i}}{\theta^{p_i - 1}} + \frac{|x'|^{p_i}}{\theta'^{p_i - 1}} \geq \frac{|x + x'|^{p_i}}{(\theta + \theta')^{p_i - 1}} \ .$$

First observe that this inequality is obviously true if θ or θ' is equal to 0. When θ and θ' are both different from 0, we use the well known fact that x^{p_i} is a convex function on $\mathbb{R}_+$ for $p_i \geq 1$, implying that $\lambda a^{p_i} + \lambda' a'^{p_i} \geq (\lambda a + \lambda' a')^{p_i}$ for any nonnegative a, a', λ and λ' satisfying $\lambda + \lambda' = 1$. Choosing $a = \frac{1}{\theta} |x|$, $a' = \frac{1}{\theta'} |x'|$, $\lambda = \frac{\theta}{\theta + \theta'}$ and $\lambda' = \frac{\theta'}{\theta + \theta'}$, we find that

$$\begin{aligned} \frac{\theta}{\theta + \theta'} \Big(\frac{|x|}{\theta}\Big)^{p_i} + \frac{\theta'}{\theta + \theta'} \Big(\frac{|x'|}{\theta'}\Big)^{p_i} &\geq \left(\frac{\theta}{\theta + \theta'} \frac{|x|}{\theta} + \frac{\theta'}{\theta + \theta'} \frac{|x'|}{\theta'}\right)^{p_i} \\ \frac{1}{\theta + \theta'} \left(\frac{|x|^{p_i}}{\theta^{p_i - 1}} + \frac{|x'|^{p_i}}{\theta'^{p_i - 1}}\right) &\geq \left(\frac{|x| + |x'|}{\theta + \theta'}\right)^{p_i} \\ \frac{|x|^{p_i}}{\theta^{p_i - 1}} + \frac{|x'|^{p_i}}{\theta'^{p_i - 1}} &\geq \frac{(|x| + |x'|)^{p_i}}{(\theta + \theta')^{p_i - 1}} \geq \frac{|x + x'|^{p_i}}{(\theta + \theta')^{p_i - 1}} \ . \end{aligned}$$

Positive homogeneity and subadditivity imply that f_p is a convex function. Since $f_p(x, \theta) \geq 0$ for all x and θ, we notice that $\mathcal{L}^p$ is the epigraph of f_p, i.e.

$$\operatorname{epi} f_p = \Big\{(x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R} \mid f_p(x, \theta) \leq \kappa\Big\} = \mathcal{L}^p \ .$$

$\mathcal{L}^p$ is thus the epigraph of a convex positively homogeneous function, hence a convex cone. □

In order to characterize strictly feasible points, we would like to identify the interior of this cone.

Theorem 4.2. *The interior of $\mathcal{L}^p$ is given by*

$$\operatorname{int} \mathcal{L}^p = \Big\{(x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R}_{++} \mid \sum_{i=1}^{n} \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} < \kappa\Big\} \ .$$

Proof. According to Lemma 7.3 in [Roc70a] we have

$$\operatorname{int} \mathcal{L}^p = \operatorname{int} \operatorname{epi} f_p = \{(x, \theta, \kappa) \mid (x, \theta) \in \operatorname{int} \operatorname{dom} f_p \text{ and } f_p(x, \theta) < \kappa\} \ .$$

The stated result then simply follows from the fact that $\operatorname{int} \operatorname{dom} f_p = \mathbb{R}^n \times \mathbb{R}_{++}$. □

Corollary 4.1. *The cone $\mathcal{L}^p$ is solid.*

Proof. It suffices to prove that there exists at least one point that belongs to $\operatorname{int} \mathcal{L}^p$, for example by taking the point $(e, 1, n)$, where e stands for the n-dimensional all-one vector. Indeed, we have $\sum_{i=1}^n \frac{|1|^1}{p_i 1^{p_i - 1}} = \sum_{i=1}^n \frac{1}{p_i} < \sum_{i=1}^n 1 = n$. □

Note 4.1. When $n = 0$, our cone $\mathcal{L}^p$ is readily seen to be equivalent to the two-dimensional positive orthant $\mathbb{R}^2_+$. We also notice that in the special case where $p_i = 2$ for all i, our cone $\mathcal{L}^p$ becomes

$$\mathcal{L}^{(2,\cdots,2)} = \left\{(x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \sum_{i=1}^n x_i^2 \leq 2\theta\kappa\right\} ,$$

which is usually called the *hyperbolic* or *rotated* second-order cone [LVBL98, Stu99a](it is a simple linear transformation of the usual second-order cone, see Chapter 9).

To illustrate our purpose, we provide in Figure 4.1 the three-dimensional graphs of the boundary surfaces of $\mathcal{L}^{(5)}$ and $\mathcal{L}^{(2)}$ (corresponding to the case $n = 1$).

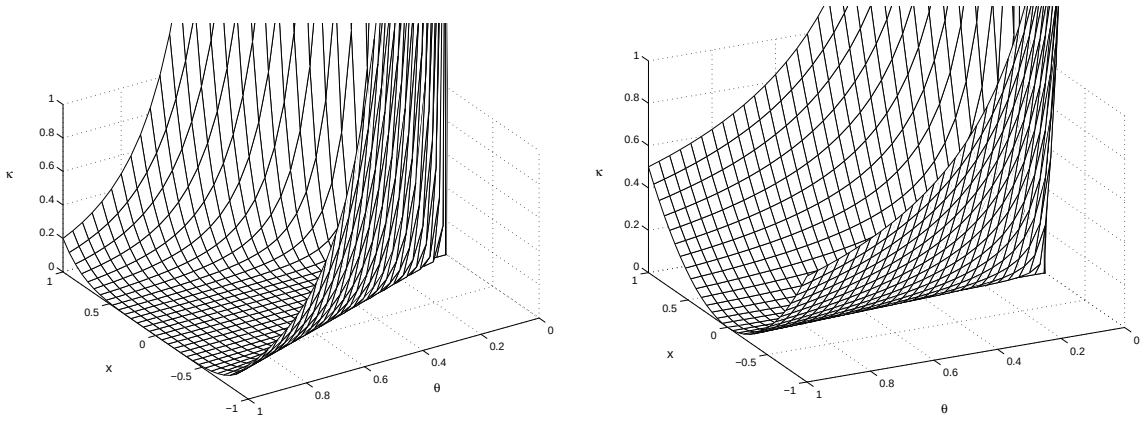


Figure 4.1: The boundary surfaces of $\mathcal{L}^{(5)}$ and $\mathcal{L}^{(2)}$ (in the case $n = 1$).

4.2.2 The dual cone

We are now going to determine the dual cone of $\mathcal{L}^p$. Let us first recall the following well-known result, known as the weighted arithmetic-geometric inequality.

Lemma 4.1. *Let $x \in \mathbb{R}^n_{++}$ and $\delta \in \mathbb{R}^n_{++}$ such that $\sum_{i=1}^n \delta_i = 1$. We have*

$$\prod_{i=1}^n x_i^{\delta_i} \leq \sum_{i=1}^n \delta_i x_i \ ,$$

equality occurring if and only if all x_i's are equal.

This result is easily proved, applying for example Jensen's inequality [Roc70a, Theorem 4.3] to the convex function $x \mapsto e^x$.

We now introduce a useful inequality, which lies at the heart of duality for $\mathcal{L}^p$ cones [Ter85, NN94]. In order to keep our exposition self-contained, we also include its proof.

Lemma 4.2. *Let $a, b \in \mathbb{R}_+$ and $\alpha, \beta \in \mathbb{R}_{++}$ such that $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. We have the inequality*

$$\frac{a^\alpha}{\alpha} + \frac{b^\beta}{\beta} \geq ab \;,$$

with equality holding if and only if $a^\alpha = b^\beta$.

Proof. The cases where $a = 0$ or $b = 0$ are obvious. When $a, b \in \mathbb{R}_{++}$, we can simply apply Lemma 4.1 on a^α and b^β with weights $\frac{1}{\alpha}$ and $\frac{1}{\beta}$ (whose sum is equal to one), which gives

$$\frac{a^\alpha}{\alpha} + \frac{b^\beta}{\beta} \geq (a^\alpha)^{1/\alpha}(b^\beta)^{1/\beta} = ab \;,$$

with equality if and only if $a^\alpha = b^\beta$. □

For ease of notation, we also introduce the *switched* cone $\mathcal{L}^p_s$ as the $\mathcal{L}^p$ cone with its last two components exchanged, i.e.

$$(x, \theta, \kappa) \in \mathcal{L}^p_s \;\Leftrightarrow\; (x, \kappa, \theta) \in \mathcal{L}^p \;.$$

We are now ready to describe the dual of $\mathcal{L}^p$.

Theorem 4.3 (Dual of $\mathcal{L}^p$). *Let $p, q \in \mathbb{R}^n_{++}$ such that $\frac{1}{p_i} + \frac{1}{q_i} = 1$ for each i. The dual of $\mathcal{L}^p$ is $\mathcal{L}^q_s$.*

Proof. By definition of the dual cone, we have

$$(\mathcal{L}^p)^* = \left\{ v^* \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mid v^T v^* \geq 0 \text{ for all } v \in \mathcal{L}^p \right\} \;.$$

We start by showing that $\mathcal{L}^q_s \subseteq (\mathcal{L}^p)^*$.

Let $v^* = (x^*, \theta^*, \kappa^*) \in \mathcal{L}^q_s$ and $v = (x, \theta, \kappa) \in \mathcal{L}^p$. We are going to prove that $v^T v^* \geq 0$, which will imply the desired inclusion. The case when $\theta = 0$ is easily handled: we have then $x = 0$ implying $v^T v^* = \kappa\kappa^* \geq 0$. Similarly we can eliminate the case where $\kappa^* = 0$. In the remaining cases, we use the definitions of $\mathcal{L}^p$ and $\mathcal{L}^q_s$ to get

$$f_p(x, \theta) = \sum_{i=1}^n \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} \leq \kappa \text{ and } f_q(x^*, \kappa^*) = \sum_{i=1}^n \frac{|x_i^*|^{q_i}}{q_i \kappa^{*q_i - 1}} \leq \theta^* \;.$$

Dividing respectively by θ and κ^* and adding the resulting inequalities we find

$$\sum_{i=1}^{n}\Big(\frac{|x_i|^{p_i}}{p_i\theta^{p_i}}+\frac{|x_i^*|^{q_i}}{q_i\kappa^{*q_i}}\Big)\leq\frac{\kappa}{\theta}+\frac{\theta^*}{\kappa^*}\,. \tag{4.1}$$

Applying now Lemma 4.2 to each pair $\dfrac{|x_i|}{\theta},\dfrac{|x_i^*|}{\kappa^*}$ we get

$$\sum_{i=1}^{n}\frac{|x_i|}{\theta}\frac{|x_i^*|}{\kappa^*}\leq\frac{\kappa}{\theta}+\frac{\theta^*}{\kappa^*}\,, \tag{4.2}$$

which is equivalent to

$$\sum_{i=1}^{n}|x_i|\,|x_i^*|\leq\kappa\kappa^*+\theta\theta^*\,.$$

Finally, noting that $x_ix_i^*\geq-|x_i|\,|x_i^*|$ we conclude that

$$v^Tv^*=x^Tx^*+\kappa\kappa^*+\theta\theta^*=\sum_{i=1}^{n}x_ix_i^*+\kappa\kappa^*+\theta\theta^*\geq\sum_{i=1}^{n}-|x_i|\,|x_i^*|+\kappa\kappa^*+\theta\theta^*\geq0\,, \tag{4.3}$$

showing that $\mathcal{L}_s^q\subseteq(\mathcal{L}^p)^*$.

Let us prove now the reverse inclusion, i.e. $(\mathcal{L}^p)^*\subseteq\mathcal{L}_s^q$.

Let $v^*=(x^*,\theta^*,\kappa^*)\in(\mathcal{L}^p)^*$. We have to show that $v^*\in\mathcal{L}_s^q$, using that $v^Tv^*\geq0$ for every $v=(x,\theta,\kappa)\in\mathcal{L}^p$. Choosing $v=(0,0,1)$, we first ensure that $v^Tv^*=\kappa^*\geq0$. We distinguish the cases $\kappa^*=0$ and $\kappa^*>0$. If $\kappa^*=0$, we have that $v^Tv^*=x^Tx^*+\theta\theta^*\geq0$ for every $v=(x,\theta,\kappa)\in\mathcal{L}^p$. Choosing $\theta=1$ and $\kappa\geq f_p(x,1)$ for any $x\in\mathbb{R}^n$, we find that $x^Tx^*+\theta^*\geq0$ for all $x\in\mathbb{R}^n$, which implies $x^*=0$ and $\theta^*\geq0$ and thus $v^*\in\mathcal{L}_s^q$. When $\kappa^*>0$, we can always choose a $v\in\mathcal{L}^p$ such that

$$\frac{|x_i|^{p_i}}{\theta^{p_i}}=\frac{|x_i^*|^{q_i}}{\kappa^{*q_i}}\quad,\quad x_ix_i^*\leq0\quad\text{and}\quad f_p(x,\theta)=\sum_{i=1}^{n}\frac{|x_i|^{p_i}}{p_i\theta^{p_i-1}}=\kappa\,. \tag{4.4}$$

Writing

$$\begin{aligned}0\leq\frac{v^Tv^*}{\theta\kappa^*}&=\Big(\frac{x}{\theta}\Big)^T\Big(\frac{x^*}{\kappa^*}\Big)^T+\frac{\theta^*}{\kappa^*}+\frac{\kappa}{\theta}\\&=\sum_{i=1}^{n}\frac{x_i}{\theta}\frac{x_i^*}{\kappa^*}+\frac{\theta^*}{\kappa^*}+\frac{\kappa}{\theta}\\&=\sum_{i=1}^{n}-\frac{|x_i|}{\theta}\frac{|x_i^*|}{\kappa^*}+\frac{\theta^*}{\kappa^*}+\frac{\kappa}{\theta}\,,\end{aligned}$$

using the case of equality of Lemma 4.2 on the pairs $\dfrac{|x_i|}{\theta},\dfrac{|x_i^*|}{\kappa^*}$ and the choice of v in (4.4),

$$\begin{aligned}&=-\sum_{i=1}^{n}\Big(\frac{|x_i|^{p_i}}{p_i\theta^{p_i}}+\frac{|x_i^*|^{q_i}}{q_i\kappa^{*q_i}}\Big)+\frac{\theta^*}{\kappa^*}+\frac{\kappa}{\theta}\\&=\frac{\theta^*}{\kappa^*}-\sum_{i=1}^{n}\frac{|x_i^*|^{q_i}}{q_i\kappa^{*q_i}}\,,\end{aligned}$$

and finally multiplying by κ^* leads to

$$\sum_{i=1}^{n} \frac{|x_i^*|^{q_i}}{q_i \kappa^{*q_i - 1}} \leq \theta^* \ ,$$

i.e. $v^* \in \mathcal{L}_s^q$, showing that $(\mathcal{L}^p)^* \subseteq \mathcal{L}_s^q$ and thus $(\mathcal{L}^p)^* = \mathcal{L}_s^q$. □

The dual of a $\mathcal{L}^p$ cone is thus equal, up to a permutation of two variables, to another $\mathcal{L}^p$ cone with a *dual* vector of exponents.

Corollary 4.2. *We also have* $(\mathcal{L}_s^p)^* = \mathcal{L}^q$, $(\mathcal{L}^q)^* = \mathcal{L}_s^p$ *and* $(\mathcal{L}_s^q)^* = \mathcal{L}^p$.

Proof. Obvious considering both the symmetry between $\mathcal{L}^p$ and $\mathcal{L}_s^q$ and the symmetry between p and q. □

Corollary 4.3. $\mathcal{L}^p$ *and* $\mathcal{L}_s^q$ *are solid and pointed.*

Proof. We have already proved that $\mathcal{L}^p$ is solid which, for obvious symmetry reasons, implies that its switched counterpart $\mathcal{L}_s^q$ is also solid. Since pointedness is the property that is dual to solidness (Theorem 3.3), noting that $\mathcal{L}^p = (\mathcal{L}_s^q)^*$ and $\mathcal{L}_s^q = (\mathcal{L}^p)^*$ is enough to prove that $\mathcal{L}^p$ and $\mathcal{L}_s^q$ are also pointed. □

Corollary 4.4. $\mathcal{L}^p$ *and* $\mathcal{L}_s^q$ *are closed.*

Proof. Starting with $(\mathcal{L}^p)^* = \mathcal{L}_s^q$ and taking the dual of both sides, we find $((\mathcal{L}^p)^*)^* = (\mathcal{L}_s^q)^*$. Since $(\mathcal{L}_s^q)^* = \mathcal{L}^p$ by Corollary 4.2 and $((\mathcal{L}^p)^*)^* = \operatorname{cl} \mathcal{L}^p$ [Roc70a, page 121], we have $\operatorname{cl} \mathcal{L}^p = \mathcal{L}^p$, hence $\mathcal{L}^p$ is closed. The switched cone $\mathcal{L}_s^q$ is obviously closed as well. □

We can also provide a direct proof of the closedness of $\mathcal{L}^p$: using the fact that it is the epigraph of f_p, it is enough to show that f_p is a lower semicontinuous function [Roc70a, Theorem 7.1]. Being convex, f_p is continuous on the interior of its effective domain, i.e. when $\theta > 0$. When $\theta = 0$, we have to prove that

$$\lim_{(x,\theta) \to (x^0, 0^+)} f_p(x, \theta) \geq f_p(x^0, 0) \ .$$

On the one hand, if $x_i^0 \neq 0$ for some index i, we have that $f_p(x^0, 0) = +\infty$ but also that $\lim_{(x,\theta)\to(x^0,0^+)} f_p(x,\theta) = +\infty$, since the term $\frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}}$ tends to $+\infty$ when (x_i, θ) tends to $(x_i^0, 0)$, hence the inequality is true. On the other hand, if $x^0 = 0$, we have to check that $\lim_{(x,\theta)\to(0,0^+)} f_p(x,\theta) \geq f_p(0,0) = 0$, which is obviously also true. From this we can conclude that f_p is lower semicontinuous and hence $\mathcal{L}^p$ is closed.

Note however that f_p is not continuous in $(0, 0)$. Choosing an arbitrary positive constant M and defining for example $x_i(\theta) = (Mp_i)^{1/p_i} \theta^{1/q_i}$, so that $x(\theta) \to 0$ when $\theta \to 0^+$, we have that $\lim_{\theta \to 0^+} f(x(\theta), \theta) = nM \neq f(0,0) = 0$. The limit of f_p at $(0, 0)$ can indeed take any positive value[1].

[1]However, taking $x(\theta)$ proportional to θ, namely $x_i(\theta) = L_i \theta$, we have $\lim_{\theta \to 0^+} f(x(\theta), \theta) = f(0, 0) = 0$, i.e. f_p is continuous on its restrictions to lines passing through the origin.

Note 4.2. As special cases, we note that when $n = 0$, $(\mathcal{L}^p)^*$ is equivalent to $\mathbb{R}^2_+$, which is the usual dual for $\mathcal{L}^p = \mathbb{R}^2_+$. In the case of $p_i = 2\ \forall i$, we find

$$(\mathcal{L}^{(2,\cdots,2)})^* = \mathcal{L}_s^{(2,\cdots,2)} = \left\{(x,\theta,\kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \sum_{i=1}^n x_i^2 \leq 2\theta\kappa\right\},$$

which is the expected result. Note that apart from these two special cases, $\mathcal{L}^p$ is in general not self-dual.

Note 4.3 *(Self-duality of $\mathcal{L}^p$ cones with $n = 1$).* Let us examine the special case of three-dimensional $\mathcal{L}^p$ cones, i.e. assume $n = 1$. Figure 4.2, representing $\mathcal{L}^{(\frac{5}{4})}$, illustrates our point: up to a permutation of variables, it is equal to $(\mathcal{L}^{(5)})^*$ (since $1/5 + 1/\frac{5}{4} = 1$) and is different from $\mathcal{L}^{(5)}$, and hence these cones are not self-dual. However, in the particular case where $n = 1$, this difference is not as great as it could be. Namely, one can show easily that $\mathcal{L}^{(p)}$ and its dual are equal up to a simple scaling of some of the variables. Indeed, we have

$$\begin{aligned}(x,\theta,\kappa) \in \mathcal{L}^{(p)} &\Leftrightarrow |x|^p \leq p\kappa\theta^{p-1} \\ &\Leftrightarrow |x|^q \leq p^{\frac{q}{p}}\kappa^{\frac{q}{p}}\theta^{(p-1)\frac{q}{p}}\end{aligned}$$

using $\frac{q}{p} = q\frac{1}{p} = q(1-\frac{1}{q}) = q-1$ and $(p-1)\frac{q}{p} = (1-\frac{1}{p})q = \frac{1}{q}q = 1$

$$\begin{aligned}&\Leftrightarrow |x|^q \leq p^{q-1}\kappa^{q-1}\theta \\ &\Leftrightarrow |x|^q \leq q(p\kappa)^{q-1}\frac{\theta}{q} \\ &\Leftrightarrow (x, \tfrac{\theta}{q}, p\kappa) \in \mathcal{L}_s^{(q)} = (\mathcal{L}^{(p)})^*\,.\end{aligned}$$

From another point of view, we could also state that these two cones are self-dual with respect to a modified inner product that takes this scaling of the variables into account.

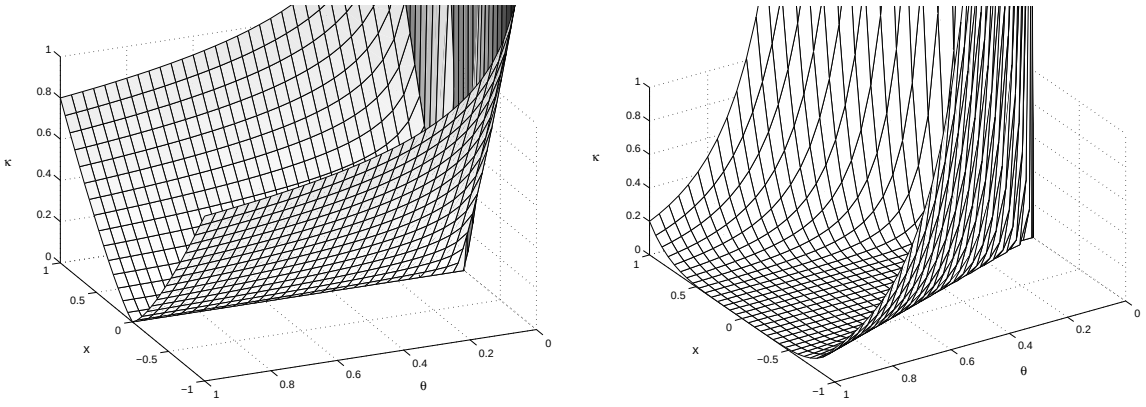


Figure 4.2: The boundary surfaces of $\mathcal{L}^{(\frac{5}{4})}$ and $\mathcal{L}^{(5)}$ (in the case $n = 1$).

Our last theorem in this section describes the cases where two vectors from $\mathcal{L}^p$ and $\mathcal{L}_s^q$ are orthogonal to each other, which will be used in the study of the duality properties.

Theorem 4.4 (orthogonality conditions). *Let $v = (x, \theta, \kappa) \in \mathcal{L}^p$ and $v^* = (x^*, \theta^*, \kappa^*) \in \mathcal{L}^q_s$. We have $v^T v^* = 0$ if and only if the following set of conditions holds*

$$\kappa^*(f_p(x, \theta) - \kappa) = 0 \tag{4.5a}$$
$$\theta(f_q(x^*, \kappa^*) - \theta^*) = 0 \tag{4.5b}$$
$$\kappa^* \frac{|x_i|^{p_i}}{\theta^{p_i - 1}} = \theta \frac{|x_i^*|^{q_i}}{\kappa^{* q_i - 1}} \tag{4.5c}$$
$$x_i x_i^* \leq 0 \text{ for all } i\ . \tag{4.5d}$$

Proof. When $\theta > 0$ and $\kappa^* > 0$, a careful reading of the first part of the proof of Theorem 4.3 shows that equality occurs if and only if all conditions in (4.5) are fulfilled. Namely, (4.5a) and (4.5b) are responsible for equality in (4.1), (4.5c) ensures that we are in the case of equality of Lemma 4.2 for inequality (4.2) and the last condition (4.5d) is necessary for equality in (4.3).

When $\theta = 0$ but $\kappa^* > 0$, we have $x = 0$ and thus $v^T v^* = \kappa\kappa^*$. This quantity is zero if and only if $\kappa = 0$, which is equivalent in this case to $f_p(x, \theta) = \kappa$ and occurs if and only if (4.5a) is satisfied (all the other conditions being trivially fulfilled). A similar reasoning takes care of the case $\theta > 0,\ \kappa^* = 0$.

Finally, when $\theta = \kappa^* = 0$, we have $x = x^* = 0$ and $v^T v^* = 0$, while the set of conditions (4.5) is also always satisfied. □

4.3 Duality for l_p-norm optimization

This is the main section , where we show how a primal-dual pair of l_p-norm optimization problems can be modelled using the $\mathcal{L}^p$ and $\mathcal{L}^q_s$ cones and how this allows us to derive the relevant duality properties.

4.3.1 Conic formulation

Let us restate here for convenience the definition of the standard primal l_p-norm optimization problem (Pl_p).

$$\sup\ b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} \frac{1}{p_i} \left| c_i - a_i^T y \right|^{p_i} \leq d_k - f_k^T y \quad \forall k \in K \tag{Pl_p}$$

(where $K = \{1, 2, \ldots, r\}$, $I = \{1, 2, \ldots, n\}$, $\{I_k\}_{k \in K}$ is a partition of I into r classes, $A \in \mathbb{R}^{m \times n}$ and $F \in \mathbb{R}^{m \times r}$ (whose columns will be denoted by $a_i,\ i \in I$ and $f_k,\ k \in K$), $y \in \mathbb{R}^m$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}^r$ and $p \in \mathbb{R}^n$ such that $p_i > 1\ \forall i \in I$).

Let us now model problem (Pl_p) with a conic formulation. The following notation will be useful in this context: v_S (resp. M_S) denotes the restriction of column vector v (resp. matrix M) to the components (resp. rows) whose indices belong to set S.

We start by introducing an auxiliary vector of variables $x^* \in \mathbb{R}^n$ to represent the argument of the power functions, namely we let

$$x_i^* = c_i - a_i^T y \text{ for all } i \in I \text{ or, in matrix form, } x^* = c - A^T y \ ,$$

and we also need additional variables $z^* \in \mathbb{R}^r$ for the linear term forming the right-hand side of the inequalities

$$z_k^* = d_k - f_k^T y \text{ for all } k \in K \text{ or, in matrix form, } z^* = d - F^T y \ .$$

Our problem is now equivalent to

$$\sup\ b^T y \quad \text{s.t.} \quad A^T y + x^* = c,\ F^T y + z^* = d \text{ and } \sum_{i \in I_k} \frac{1}{p_i} |x_i^*|^{p_i} \leq z_k^* \quad \forall k \in K \ ,$$

where we can easily plug our definition of the $\mathcal{L}^p$ cone, provided we fix variables θ to 1

$$\sup\ b^T y \quad \text{s.t.} \quad A^T y + x^* = c,\ F^T y + z^* = d \text{ and } (x_{I_k}^*, 1, z_k^*) \in \mathcal{L}^{p^k}\ \forall k \in K$$

(where for convenience we defined vectors $p^k = (p_i \mid i \in I_k)$ for $k \in K$). We finally introduce an additional vector of fictitious variables $v^* \in \mathbb{R}^r$ whose components are fixed to 1 by linear constraints to find

$$\sup\ b^T y \quad \text{s.t.} \quad A^T y + x^* = c,\ F^T y + z^* = d,\ v^* = e \text{ and } (x_{I_k}^*, v_k^*, z_k^*) \in \mathcal{L}^{p^k}\ \forall k \in K$$

(where e stands again for the all-one vector). Rewriting the linear constraints with a single matrix equality, we end up with

$$\sup\ b^T y \quad \text{s.t.} \quad \begin{pmatrix} A^T \\ F^T \\ 0 \end{pmatrix} y + \begin{pmatrix} x^* \\ z^* \\ v^* \end{pmatrix} = \begin{pmatrix} c \\ d \\ e \end{pmatrix} \text{ and } (x_{I_k}^*, v_k^*, z_k^*) \in \mathcal{L}^{p^k}\ \forall k \in K \ , \qquad (\mathrm{CP}l_p)$$

which is exactly a conic optimization problem in the dual[2] form (CD), using variables $(\tilde{y}, \tilde{s})$, data $(\tilde{A}, \tilde{b}, \tilde{c})$ and a cone $\mathcal{C}^*$ such that

$$\tilde{y} = y,\ \tilde{s} = \begin{pmatrix} x^* \\ z^* \\ v^* \end{pmatrix},\ \tilde{A} = \begin{pmatrix} A & F & 0 \end{pmatrix},\ \tilde{b} = b,\ \tilde{c} = \begin{pmatrix} c \\ d \\ e \end{pmatrix} \text{ and } \mathcal{C}^* = \mathcal{L}^{p^1} \times \mathcal{L}^{p^2} \times \cdots \times \mathcal{L}^{p^r} \ ,$$

where $\mathcal{C}^*$ has been defined according to Note 3.1, since we have to deal with multiple conic constraints involving disjoint sets of variables.

Using properties of $\mathcal{L}^p$ proved in the previous section, it is straightforward to show that $\mathcal{C}^*$ is a solid, pointed, closed convex cone whose dual is

$$(\mathcal{C}^*)^* = \mathcal{C} = \mathcal{L}_s^{q^1} \times \mathcal{L}_s^{q^2} \times \cdots \times \mathcal{L}_s^{q^r} \ ,$$

another solid, pointed, closed convex cone (where we have defined a vector $q \in \mathbb{R}^n$ such that $\frac{1}{p_i} + \frac{1}{q_i} = 1$ for all $i \in I$ and vectors q^k such that $q^k = (q_i \mid i \in I_k)$ for $k \in K$). This allows

[2]This is the reason why we added a * superscript to the notation of our additional variables, in order to emphasize the fact that the primal l_p-norm optimization problem ($\mathrm{P}l_p$) is in fact in the dual conic form (CD).

us to derive a dual problem to (CPl_p) in a completely mechanical way and find the following conic optimization problem, expressed in the primal form (CP) (since the dual of a problem in dual form is a problem in primal form):

$$\inf \begin{pmatrix} c^T & d^T & e^T \end{pmatrix} \begin{pmatrix} x \\ z \\ v \end{pmatrix} \quad \text{s.t.} \quad \begin{pmatrix} A & F & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ v \end{pmatrix} = b \text{ and } (x_{I_k}, v_k, z_k) \in \mathcal{L}_s^{q^k} \text{ for all } k \in K \ ,$$

which is equivalent to

$$\inf c^T x + d^T z + e^T v \quad \text{s.t.} \quad Ax + Fz = b \text{ and } (x_{I_k}, v_k, z_k) \in \mathcal{L}_s^{q^k} \text{ for all } k \in K \ , \qquad (\text{CD}l_p)$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^r$ and $v \in \mathbb{R}^r$ are the dual variables we optimize. This problem can be simplified: making the conic constraints explicit, we find

$$\inf c^T x + d^T z + e^T v \quad \text{s.t.} \quad Ax + Fz = b, \ \sum_{i \in I_k} \frac{|x_i|^{q_i}}{q_i z_k^{q_i - 1}} \leq v_k \ \forall k \in K \text{ and } z \geq 0 \ ,$$

keeping in mind the convention on zero denominators that in effect implies $z_k = 0 \Rightarrow x_{I_k} = 0$. Finally, we can remove the v variables from the formulation since they are only constrained by the sum inequalities, which have to be tight at any optimal solution. We can thus directly incorporate these sums into the objective function, which leads to

$$\inf \ \psi(x,z) = c^T x + d^T z + \sum_{k \in K | z_k > 0} z_k \sum_{i \in I_k} \frac{1}{q_i} \left| \frac{x_i}{z_k} \right|^{q_i} \quad \text{s.t.} \quad \begin{cases} Ax + Fz = b \text{ and } z \geq 0 \ , \\ z_k = 0 \Rightarrow x_i = 0 \ \forall i \in I_k \ . \end{cases} \qquad (\text{D}l_p)$$

Unsurprisingly, the dual formulation (Dl_p) we have just found without much effort is exactly the standard form for a dual l_p-norm optimization problem [Ter85].

4.3.2 Duality properties

We are now able to prove the weak duality property for the l_p-norm optimization problem.

Theorem 4.5 (Weak duality). *If y is feasible for (Pl_p) and (x, z) is feasible for (Dl_p), we have $\psi(x, z) \geq b^T y$. Equality occurs if and only if for all $k \in K$ and $i \in I_k$*

$$z_k \Big(\sum_{i \in I_k} \frac{1}{p_i} \left| c_i - a_i^T y \right|^{p_i} + f_k^T y - d_k \Big) = 0, \quad x_i (c_i - a_i^T y) \leq 0, \quad z_k \left| c_i - a_i^T y \right|^{p_i} = \frac{|x_i|^{q_i}}{z_k^{q_i - 1}} \ . \quad (4.6)$$

Proof. Let y and (x, z) be feasible for (Pl_p) and (Dl_p). Choosing $v_k = f_{q^k}(x_{I_k}, z_k)$ for all $k \in K$, we have that (x, z, v) is feasible for (CDl_p) with the same objective function, i.e. with $c^T x + d^T z + e^T v = \psi(x, z)$. Moreover, computing (x^*, z^*, v^*) from y in order to satisfy the linear constraints in (CPl_p), i.e. according to

$$x_i^* = c_i - a_i^T y, \quad z_k^* = d_k - f_k^T y, \quad v_k^* = 1 \ , \qquad (4.7)$$

we have that (x^*, z^*, v^*, y) is feasible for (CPl_p). The standard weak duality property for the conic pair (CPl_p)–(CDl_p) from Theorem 3.4 then states that $c^T x + d^T z + e^T v \geq b^T y$, which in turn implies $\psi(x, z) \geq b^T y$.

We proceed now to investigate the equality conditions. At the optimum, variables v_k must assume their lower bounds so that we can still assume that $v_k = f_{q^k}(x_{I_k}, z_k)$ holds for all $k \in K$. We also keep variables (x^*, z^*, v^*) defined by (4.7). From the weak duality Theorem 3.4, we know that equality can only occur if the primal and dual vectors of variables are orthogonal to each other for each conic constraint, i.e. $(x^*_{I_k}, z^*_k, v^*_k)^T(x_{I_k}, z_k, v_k) = 0$ for all $k \in K$.

Having $(x^*_{I_k}, v^*_k, z^*_k)^T \in \mathcal{L}^{p^k}$ and $(x_{I_k}, v_k, z_k) \in \mathcal{L}_s^{q^k}$, Theorem 4.4 gives us the necessary and sufficient conditions for equality to happen

$$z_k(f_{p^k}(x^*_{I_k}, v^*_k) - z^*_k) = 0,\ v^*_k(f_{q^k}(x_{I_k}, z_k) - v_k) = 0,\ z_k\frac{|x^*_i|^{p_i}}{v^{*p_i-1}_k} = v^*_k\frac{|x_i|^{q_i}}{z^{q_i-1}_k},\ x_i x^*_i \leq 0 \qquad (4.8)$$

for all $i \in I_k$ and $k \in K$. The second condition is always satisfied while the other three conditions can be readily simplified using (4.7) to give the announced inequalities (4.6). □

The weak duality property is a rather straightforward consequence of the convexity of the problems, and in fact can be proved without too many difficulties without sophisticated tools from duality theory. However, this is not the case with the next theorem, which deals with a strong duality property.

In the case of a general pair of primal and dual conic problems, the duality gap at the optimum is not always equal to zero, neither are the primal or dual optimum objective values always attained by feasible solutions (see the examples in Section3.3). However, it is well-known that in the special case of linear optimization, we always have a zero duality gap and attainment of both optimum objective values. The status of l_p-norm optimization lies somewhere between these two situations: the duality gap is always equal zero but attainment of the optimum objective value can only be guaranteed for the primal problem.

In the course of our proof, we will need to use the well-known Goldman-Tucker theorem [GT56] for linear optimization, which we state here for reference.

Theorem 4.6 (Goldman-Tucker). *Let us consider the following primal-dual pair of linear optimization problems in standard form:*

$$\min c^T x \quad s.t. \quad Ax = b \text{ and } x \geq 0 \qquad \text{and} \qquad \max b^T y \quad s.t. \quad A^T y + s = c \text{ and } s \geq 0\ .$$

If both problems are feasible, there exists a unique partition $(\mathcal{B}, \mathcal{N})$ of the index set common to vectors x and s such that

- ◇ *every optimal solution $\hat{x}$ to the primal problem satisfies $\hat{x}_\mathcal{N} = 0$.*
- ◇ *every optimal solution $(\hat{y}, \hat{s})$ to the dual problem satisfies $\hat{s}_\mathcal{B} = 0$.*

This partition is called the optimal partition. *Moreover, there exists at least an optimal primal-dual solution $(\hat{x}, \hat{y}, \hat{s})$ such that $\hat{x} + \hat{s} > 0$, hence satisfying $\hat{x}_\mathcal{B} > 0$ and $\hat{s}_\mathcal{N} > 0$. Such a pair is called a* strictly complementary *pair*[3].

[3]This optimal partition can be computed in polynomial time by interior-point methods. Indeed, it is possible to prove for example that the short-step algorithm presented in Chapter 2 converges to a strictly complementary solution, and thus allows us to identify the optimal partition unequivocally.

This theorem is central to the theory of duality for linear optimization. Its most important consequence is the fact that any pair of primal-dual optimal solutions $\hat{x}$ and $(\hat{y}, \hat{s})$ must have a zero duality gap. Indeed, the duality gap is equal to $\hat{x}^T\hat{s}$ (see Theorem 3.4) and the theorem implies that $\hat{x}_{\mathcal{N}} = 0$ and $\hat{s}_{\mathcal{B}} = 0$, which leads to

$$\hat{x}^T\hat{s} = \sum_{i\in\mathcal{B}} \hat{x}_i\hat{s}_i + \sum_{i\in\mathcal{N}} \hat{x}_i\hat{s}_i = 0$$

since $(\mathcal{B}, \mathcal{N})$ is a partition of the index set of the variables. One can also consider this theorem as a version of the strong duality Theorem 3.5 specialized for linear optimization, with the important difference that it is valid even when no Slater point exists.

The strong duality theorem for l_p-norm optimization we are about to prove is the following:

Theorem 4.7 (Strong duality). *If both problems* (Pl_p) *and* (Dl_p) *are feasible, the primal optimal objective value is attained with a zero duality gap, i.e.*

$$\begin{aligned} p^* &= \max b^T y \quad s.t. \quad \sum_{i\in I_k} \frac{1}{p_i} \left|c_i - a_i^T y\right|^{p_i} \leq d_k - f_k^T y \quad \forall k \in K \\ &= \inf \psi(x,z) \quad s.t. \quad \begin{cases} Ax + Fz = b \text{ and } z \geq 0 \\ z_k = 0 \Rightarrow x_i = 0 \ \forall i \in I_k \end{cases} \qquad = d^* \ . \end{aligned}$$

Proof. The strong duality Theorem 3.5 tells us that zero duality gap and primal attainment are guaranteed by the existence of a strictly interior dual feasible solution (excluding the case of an unbounded dual). Let (x, z) be a feasible solution for (Dl_p). We would like to complement it with a vector v such that the corresponding solution (x, z, v) is strictly feasible for the conic formulation (CDl_p).

Since cone $\mathcal{C}$ is the cartesian products of the set of cones $\mathcal{L}_s^{q^k}$ for $k \in K$, we need in fact for (x, z, v) to be a strictly feasible solution of (CDl_p) that $(x_{I_k}, z_k, v_k) \in \operatorname{int} \mathcal{L}_s^{q^k}$ holds for all $k \in K$. Using now Theorem 4.2 to identify the interior of the $\mathcal{L}_s^q$ cones, we see that both conditions $v_k > f_{p^k}(x_{I_k}, z_k)$ and $z_k > 0$ have to be valid for all $k \in K$.

Since vector v contains only free variables and is not constrained by the linear constraints, it is always possible to choose it such that $v_k > f_{p^k}(x_{I_k}, z_k)$ for all $k \in K$. However, the situation is much different for z: it is unfortunately not always possible to find a strictly positive z, since it may happen that the linear constraints combined with the nonnegativity constraint on z force one or more of the components z_k to be equal to zero for all primal feasible solutions. Here is an outline of the three-step strategy we are going to follow:

a. Since some components of z may prevent the existence of a strictly feasible solution to (CDl_p), we are going to define a *restricted* version of (CDl_p) where those problematic components of z and the associated variables x have been removed. Hopefully, this restricted problem (RDl_p) will not behave too differently from the original because the zero components of z and x did not play a crucial role in it.

b. Since this restricted problem will now admit a strictly feasible solution, its dual problem (RPl_p) (which is a problem in primal form) has a duality gap equal to zero with its optimal objective value attained by some solution.

c. The last step of our proof will be to convert this optimal solution with a zero duality gap for the restricted primal problem (RPl_p) into an optimal solution for the original primal problem (CPl_p).

The whole procedure can be summarized with the following diagram:

$$\begin{array}{ccccccc} (\mathrm{P}l_p) & \equiv & (\mathrm{CP}l_p) & \overset{\text{Weak}}{\longleftrightarrow} & (\mathrm{CD}l_p) & \equiv & (\mathrm{D}l_p) \\ & & \text{c.}\updownarrow & & \updownarrow\text{a.} & & \\ & & (\mathrm{RP}l_p) & \overset{\text{Strong (zero gap)}}{\longleftrightarrow} & (\mathrm{RD}l_p) & & \\ & & \downarrow & & \uparrow & & \\ & & \text{(Attainment)} & \text{b.} & \text{(Strictly feasible)} & & \end{array}$$

Let us first identify the problematic z_k's that are identically equal to zero for all feasible solutions. This can be done by solving the following linear optimization problem:

$$\min 0 \quad \text{s.t.} \quad Ax + Fz = b \text{ and } z \geq 0\ . \tag{ALP}$$

This problem has the same feasible region as our dual problem (Dl_p) (actually, its feasible region can be slightly larger from the point of view of the x variables, since the special constraints $z_k = 0 \Rightarrow x_{I_k} = 0$ have been omitted, but this does not have any effect on our reasoning). We are thus looking for components of z that are equal to zero on the whole feasible region of (ALP).

Since this problem has a zero objective function, all its feasible solutions are optimal and we can therefore deduce that if a variable z_k is zero for all feasible solutions to problem (ALP), it is zero for all optimal solution to problem (ALP). In order to use the Goldman-Tucker theorem, we also write the dual[4] of problem (ALP):

$$\max\ b^T y \quad \text{s.t.} \quad A^T y = 0, \quad F^T y + z^* = 0 \quad \text{and} \quad z^* \geq 0\ . \tag{ALD}$$

Both (ALP) and (ALD) are feasible (the former because (Dl_p) is assumed to be feasible, the latter because $(y, z^*) = (0, 0)$ is always a feasible solution), which means that the Goldman-Tucker theorem is applicable. Having now the optimal partition $(\mathcal{B}, \mathcal{N})$ at hand, we observe that the index set $\mathcal{N}$ defines exactly the set of variables z_i that are identically zero on the feasible regions of problems (ALP) and (Dl_p). We are thus now ready the apply the strategy outlined above.

a. Let us introduce the reduced primal-dual pair of l_p-norm optimization problems where variables z_k and x_{I_k} with $k \in \mathcal{N}$ have been removed. We start with the dual problem

$$\inf c_{I_\mathcal{B}}^T x_{I_\mathcal{B}} + d_\mathcal{B}^T z_\mathcal{B} + e_\mathcal{B}^T v_\mathcal{B} \quad \text{s.t.} \quad A_{I_\mathcal{B}} x_{I_\mathcal{B}} + F_\mathcal{B} z_\mathcal{B} = b,\ (x_{I_k}, v_k, z_k) \in \mathcal{L}_s^{q^k}\ \forall k \in \mathcal{B}\ , \tag{RD}l_p)$$

[4] Although problem (ALP) is not exactly formulated in the standard form used to state Theorem 4.6, the same results hold in the case of a general linear optimization problem.

where $I_{\mathcal{B}}$ stands for $\cup_{k\in\mathcal{B}} I_k$. It is straightforward to check that this problem is completely equivalent to problem (CDl_p), since the variables $z_{\mathcal{N}}$ and $x_{I_{\mathcal{N}}}$ we removed, being forced to zero for all feasible solutions, had no contribution to the objective or to the linear constraints in (CDl_p).

The corresponding conic constraints become $(0, v_k, 0) \in \mathcal{L}_s^{q^k} \Leftrightarrow v_k \geq 0\ \forall k \in \mathcal{N}$, which imply at the optimum that $v_k = 0\ \forall k \in \mathcal{N}$, showing that variables $v_{\mathcal{N}}$ can also be safely removed without changing the optimum objective value. We can thus conclude that $\inf(\text{RD}l_p) = \inf(\text{CD}l_p) = \inf(\text{D}l_p)$.

b. Because of the second part of the Goldman-Tucker theorem, there is at least one feasible solution to (ALP) such that $z_{\mathcal{B}} > 0$. Combining the $(x_{I_{\mathcal{B}}}, z_{\mathcal{B}})$ part of this solution with a vector $v_{\mathcal{B}}$ with sufficiently large components gives us a strictly feasible solution for (RDl_p) ($z_k > 0$ and $v_k > f_{q^k}(x_{I_k}, z_k)$ for all $k \in \mathcal{B}$), which is exactly what we need to apply our strong duality Theorem 3.5. Let us first write down the dual problem of (RDl_p), the restricted primal:

$$\sup\ b^T y \quad \text{s.t.} \quad \left\{ \begin{array}{c} A_{I_{\mathcal{B}}}^T y + x^*_{I_{\mathcal{B}}} = c_{I_{\mathcal{B}}},\ F_{\mathcal{B}}^T y + z^*_{\mathcal{B}} = d_{\mathcal{B}},\ v^*_{\mathcal{B}} = e, \\ (x^*_{I_k}, v^*_k, z^*_k) \in \mathcal{L}^{p^k}\ \forall k \in \mathcal{B}\ . \end{array} \right. \qquad (\text{RP}l_p)$$

We cannot be in the first case of the strong duality Theorem 3.5, since unboundedness of (RDl_p) would imply unboundedness of the original problem (Dl_p) which in turn would prevent the existence of a feasible primal solution (simple consequence of the weak duality theorem). We can thus conclude that there exists an optimal solution to (RPl_p) $(\hat{x}^*_{I_{\mathcal{B}}}, \hat{z}^*_{\mathcal{B}}, \hat{v}^*_{\mathcal{B}}, \hat{y})$ such that $b^T\hat{y} = \max(\text{RP}l_p) = \inf(\text{RD}l_p)$.

c. Combining the results obtained so far, we have proved that $\max(\text{RP}l_p) = \inf(\text{D}l_p)$. The last step we need to perform is to prove that $\max(\text{P}l_p) = \max(\text{RP}l_p)$, i.e. that the optimum objective of (Pl_p) is attained and that it is equal to the optimal objective value of (RPl_p). Unfortunately, the apparently most straightforward way to do this, namely using the optimal solution $\hat{y}$ we have at hand for problem (RPl_p), does not work since it is not necessarily feasible for problem (CPl_p). The reason is that (CPl_p) contains additional conic constraints (the ones corresponding to $k \in \mathcal{N}$) which are not guaranteed to be satisfied by the optimal solution $\hat{y}$ of the restricted problem. We can however overcome this difficulty by perturbing this solution by a suitably chosen vector such that

- ◇ feasibility for the constraints $k \in \mathcal{B}$ is not lost,
- ◇ feasibility for the constraints $k \in \mathcal{N}$ can be gained.

Let us consider $(\bar{x}, \bar{z}, \bar{y}, \bar{z}^*)$, a strictly complementary solution to the primal-dual pair (ALP)–(ALD) whose existence is guaranteed by the Goldman-Tucker theorem. We have thus $\bar{z}^*_{\mathcal{N}} > 0$ and $\bar{z}^*_{\mathcal{B}} = 0$. Since all primal solutions have a zero objective, the optimal dual objective value also satisfies $b^T\bar{y} = 0$. Summarizing the properties of $\bar{y}$ obtained so far, we can write

$$b^T\bar{y} = 0, \quad A^T\bar{y} = 0, \quad F_{\mathcal{B}}^T\bar{y} = -\bar{z}^*_{\mathcal{B}} = 0 \text{ and } F_{\mathcal{N}}^T\bar{y} = -\bar{z}^*_{\mathcal{N}} < 0\ .$$

Let us now consider $y = \hat{y} + \lambda\bar{y}$ with $\lambda \geq 0$ as a solution of (CPl_p) and compute the value of x^* and z^* given by (4.7), distinguishing the $\mathcal{B}$ and $\mathcal{N}$ parts (we already know that $v^* = e$):

$$\begin{array}{rclcll} x^*_{I_\mathcal{B}} & = & c_{I_\mathcal{B}} - A^T_{I_\mathcal{B}} y & = & c_{I_\mathcal{B}} - A^T_{I_\mathcal{B}} \hat{y} & = \hat{x}^*_{I_\mathcal{B}} \quad (\text{using } A^T_{I_\mathcal{B}} \bar{y} = 0) \\ z^*_\mathcal{B} & = & d_\mathcal{B} - F^T_\mathcal{B} y & = & d_\mathcal{B} - F^T_\mathcal{B} \hat{y} & = \hat{z}^*_\mathcal{B} \quad (\text{using } F^T_\mathcal{B} \bar{y} = 0) \\ x^*_{I_\mathcal{N}} & = & c_{I_\mathcal{N}} - A^T_{I_\mathcal{N}} y & = & c_{I_\mathcal{N}} - A^T_{I_\mathcal{N}} \hat{y} & = \hat{x}^*_{I_\mathcal{N}} \quad (\text{using } A^T_{I_\mathcal{N}} \bar{y} = 0) \\ z^*_\mathcal{N} & = & d_\mathcal{N} - F^T_\mathcal{N} y & = & d_\mathcal{N} - F^T_\mathcal{N} \hat{y} + \lambda \bar{z}^*_\mathcal{N} & \quad (\text{using } -F^T_\mathcal{N} \bar{y} = \bar{z}^*_\mathcal{N}) \,. \end{array}$$

The conic constraints corresponding to $k \in \mathcal{B}$ remain valid for all λ, since the associated variables do not vary with λ. Considering now the constraints for $k \in \mathcal{N}$, we see that $x^*_{I_\mathcal{N}}$ does not depend on λ, while $z^*_\mathcal{N}$ can be made arbitrarily large by increasing λ, due to the fact that $\bar{z}^*_\mathcal{N} > 0$. Choosing a sufficiently large λ, we can force $(x^*_{I_k}, 1, z^*_k) \in \mathcal{L}^{q^k}_s$ for $k \in \mathcal{N}$ and thus make (x^*, v^*, z^*, y) feasible for (CPl_p). Obviously, we also have that y is feasible for (Pl_p) with the same objective value.

Evaluating this objective value, we find that $b^T y = b^T\hat{y} + \lambda b^T\bar{y} = b^T\hat{y} = \max(\text{RP}l_p)$, i.e. the feasible solution y we constructed has the same objective value for (CPl_p) and (Pl_p) as $\hat{y}$ for (RPl_p). This proves that $\max(\text{RP}l_p) \leq \sup\ (\text{P}l_p)$, which combined with our previous results gives $d^* = \inf(\text{D}l_p) = b^T\hat{y} = \max(\text{RP}l_p) \leq \sup\ (\text{P}l_p) = p^*$. Finally, using the weak duality of Theorem 4.5, i.e. $p^* \leq d^*$, we obtain $d^* = \inf(\text{D}l_p) = b^T\hat{y} = \sup\ (\text{P}l_p) = p^*$, which implies that $\hat{y}$ is optimum for (Pl_p), $\sup\ (\text{P}l_p) = \max(\text{P}l_p)$ and finally the desired result $p^* = \max(\text{P}l_p) = \inf(\text{D}l_p) = d^*$.

□

4.3.3 Examples

We conclude this section by providing a few examples of the possible situations that can arise for a couple of primal-dual l_p-norm optimization problems. Let us consider the following problem data:

$$r = 1,\ K = \{1\},\ n = 1,\ I_1 = \{1\},\ m = 1,\ A = 1,\ F = 0,\ c = 5,\ d \in \mathbb{R},\ b = 1,\ p = 3$$

(d_1 is left unspecified), which translates into the following primal problem:

$$\sup\ y_1 \quad \text{s.t.} \quad \frac{1}{3}|5 - y_1|^3 \leq d_1 \,. \tag{Pl_p}$$

Noting $q = \frac{3}{2}$, we can also write down the dual

$$\inf 5x_1 + d_1 z_1 + z_1 \frac{1}{3/2}\left|\frac{x_1}{z_1}\right|^{3/2} \quad \text{s.t.} \quad x_1 = 1,\ z_1 \geq 0, z_1 = 0 \Rightarrow x_1 = 0 \,. \tag{Dl_p}$$

This pair of problems can readily be simplified to

$$\sup\ y_1 \quad \text{s.t.} \quad |5 - y_1| \leq \sqrt[3]{3d_1} \quad \text{and} \quad \inf 5 + d_1 z_1 + \frac{2}{3\sqrt{z_1}} \quad \text{s.t.} \quad z_1 > 0$$

- When $d = 9$, our primal constraint becomes $|5 - y_1| \leq 3$, which gives a primal optimum equal to $y_1 = 8$. Looking at the dual, we have

$$9z_1 + \frac{2}{3\sqrt{z_1}} = \frac{1}{3}(27z_1) + \frac{2}{3}(\frac{1}{\sqrt{z_1}}) \leq (27z_1)^{\frac{1}{3}}(\frac{1}{\sqrt{z_1}})^{\frac{2}{3}} = 3$$

  (using the weighted arithmetic-geometric mean), which shows that the dual optimum is also equal to 8, and is attained for $(x, z) = (1, \frac{1}{9})$. This is the most common situation: both optimum values are finite and attained, with a zero duality gap.

- When $d = 0$, our primal constraint becomes $|5 - y_1| \leq 0$, which implies that the only feasible solution is $y_1 = 5$, giving a primal optimum equal to 5. The dual optimum value is then $\inf 5 + \frac{2}{3\sqrt{z_1}} = 5$, equal to the primal but not attained ($z_1 \to +\infty$). This shows that there are problems for which the dual optimum is not attained, i.e. we do not have the perfect duality of linear optimization (one can observe that in this case the primal had no strict interior).

- Finally, when $d = -1$, the primal becomes infeasible while the dual is unbounded (take again $z \to +\infty$).

4.4 Complexity

The goal of this section is to prove it is possible to solve an l_p-norm optimization problem up to a given accuracy in polynomial time. According to the theoretical framework of Nesterov and Nemirovski [NN94], which was presented in Chapter 2, in order to solve the conic problem described in Chapter 3

$$\inf_x c^T x \quad \text{s.t.} \quad Ax = b \text{ and } x \in \mathcal{C} \,, \tag{CP}$$

we only need to find a computable self-concordant barrier function for the cone $\mathcal{C}$, according to Definition 2.2. Indeed, we can apply for example the following variant of Theorem 2.5.

Theorem 4.8. *Given a (κ, ν)-self-concordant barrier for the cone $\mathcal{C} \subseteq \mathbb{R}^n$ and a feasible interior starting point $x_0 \in \operatorname{int}\mathcal{C}$ satisfying $\delta(x_0, \mu_0) < \frac{1}{13.42\kappa}$, a short-step interior-point algorithm can solve problem* (CP) *up to ϵ accuracy within*

$$\mathcal{O}\left(\kappa\sqrt{\theta}\log\frac{\mu_0\kappa\sqrt{\nu}}{\epsilon}\right) \text{ iterations,}$$

such that at each iteration the self-concordant barrier and its first and second derivatives have to be evaluated and a linear system has to be solved in $\mathbb{R}^n$ (i.e. the Newton step for the barrier problem has to be computed).

We are now going to describe a self-concordant barrier that allows us to solve conic problems involving our $\mathcal{L}^p$ cone (we follow an approach similar to the one used in [XY00]). The following convex cone

$$\left\{(x, y) \in \mathbb{R} \times \mathbb{R}_+ \mid |x|^p \leq y\right\}$$

(with $p > 1$) admits the well-known self-concordant barrier

$$f_p : \mathbb{R} \times \mathbb{R}_{++} \mapsto \mathbb{R} : (x, y) \mapsto -2 \log y - \log(y^{2/p} - x^2)$$

with parameters $(1, 4)$ (see [NN94, Propostion 5.3.1], note we are using here the convention $\kappa = 1$). Let $n \in \mathbb{N}$, $p \in \mathbb{R}^n$ and $I = \{1, 2, \ldots, n\}$. We have that

$$\left\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n_+ \mid |x_i|^{p_i} \leq y_i \ \forall i \in I\right\}$$

admits

$$f_p : \mathbb{R}^n \times \mathbb{R}^n_{++} \mapsto \mathbb{R} : (x, y) \mapsto \sum_{i=1}^{n} \left(-2 \log y_i - \log(y_i^{2/p_i} - x_i^2)\right)$$

with parameters $(1, 4n)$ (using [NN94, Propostion 5.1.2]). This also implies that the set

$$\mathcal{S}_p = \left\{(x, y, \kappa) \in \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathbb{R} \mid |x_i|^{p_i} \leq y_i \ \forall i \in I \text{ and } \kappa = \sum_{i=1}^{n} \frac{y_i}{p_i}\right\}$$

admits a self-concordant barrier $f'_p(x, y, \kappa) = f_p(x, y)$ with parameters $(1, 4n)$ (taking the cartesian product with $\mathbb{R}$ essentially leaves the self-concordant barrier unchanged, taking the intersection with an affine subspace does not influence self-concordancy). Finally, we use another result from Nesterov and Nemirovski to find a self-concordant barrier for the *conic hull* of $\mathcal{S}_p$, which is defined by

$$\begin{aligned}
\mathcal{H}_p &= \text{cl}\left\{(x, t) \in \mathcal{S}_p \times \mathbb{R}_{++} \mid \frac{x}{t} \in \mathcal{S}_p\right\} \\
&= \text{cl}\left\{(x, y, \kappa, \theta) \in \mathcal{S}_p \times \mathbb{R}_{++} \mid (\frac{x}{\theta}, \frac{y}{\theta}, \frac{\kappa}{\theta}) \in \mathcal{S}_p\right\} \\
&= \text{cl}\left\{(x, y, \kappa, \theta) \in \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathbb{R} \times \mathbb{R}_{++} \mid \left|\frac{x_i}{\theta}\right|^{p_i} \leq \frac{y_i}{\theta} \ \forall i \in I \text{ and } \frac{\kappa}{\theta} = \sum_{i=1}^{n} \frac{y_i}{p_i \theta}\right\} \\
&= \text{cl}\left\{(x, y, \kappa, \theta) \in \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathbb{R} \times \mathbb{R}_{++} \mid \frac{|x_i|^{p_i}}{\theta^{p_i - 1}} \leq y_i \ \forall i \in I \text{ and } \kappa = \sum_{i=1}^{n} \frac{y_i}{p_i}\right\} \\
&= \left\{(x, y, \kappa, \theta) \in \mathbb{R}^n \times \mathbb{R}^n_+ \times \mathbb{R} \times \mathbb{R}_+ \mid \frac{|x_i|^{p_i}}{\theta^{p_i - 1}} \leq y_i \ \forall i \in I \text{ and } \kappa = \sum_{i=1}^{n} \frac{y_i}{p_i}\right\}
\end{aligned}$$

(to find the last equality, you have to consider accumulation points with $\theta = 0$, which in fact must satisfy $x = 0$, which in turn can be seen to match exactly the convention about zero denominators we chose in Definition 4.1), and find that

$$h_p : \mathbb{R}^n \times \mathbb{R}^n_{++} \times \mathbb{R} \times \mathbb{R}_{++} \mapsto \mathbb{R} : (x, y, \kappa, \theta) \mapsto \left(f_p(\frac{x}{\theta}, \frac{y}{\theta}) - 8n \log \theta\right)$$

is a self-concordant barrier for $\mathcal{H}_p$ with parameter $(20, 8n)$ (see [NN94, Proposition 5.1.4]). We now make the following interesting observation linking $\mathcal{H}_p$ to our cone $\mathcal{L}^p$.

Theorem 4.9. *The $\mathcal{L}^p$ cone is equal to the projection of $\mathcal{H}_p$ on the space of (x, κ, θ), i.e.*

$$(x, \theta, \kappa) \in \mathcal{L}^p \quad \Leftrightarrow \quad \exists y \in \mathbb{R}^n_+ \mid (x, y, \kappa, \theta) \in \mathcal{H}_p \ .$$

Proof. This proof is straightforward. First note that both sets take the same convention in case of a zero denominator. Let $(x, \theta, \kappa) \in \mathcal{L}^p$. Choosing y such that $y_i = \frac{|x_i|^{p_i}}{\theta^{p_i - 1}}$ for all $i \in I$ ensures that

$$\sum_{i=1}^{n} \frac{y_i}{p_i} = \sum_{i=1}^{n} \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} \leq \kappa$$

(this last inequality because of the definition of $\mathcal{L}^p$). It is now possible to increase y_1 until the equality $\kappa = \sum_{i=1}^{n} \frac{y_i}{p_i}$ is satisfied, which shows $(x, y, \kappa, \theta) \in \mathcal{H}_p$. For the reverse inclusion, suppose $(x, y, \kappa, \theta) \in \mathcal{H}_p$. This implies that

$$\kappa = \sum_{i=1}^{n} \frac{y_i}{p_i} \geq \sum_{i=1}^{n} \frac{|x_i|^{p_i}}{p_i \theta^{p_i - 1}} \ ,$$

which is exactly the defining inequality of $\mathcal{L}^p$. □

Suppose now we have now to solve

$$\inf_x c^T x \quad \text{s.t.} \quad Ax = b \text{ and } x \in \mathcal{L}^p \ . \tag{4.9}$$

In light of the previous theorem, it is equivalent to solve

$$\inf_{(x,y)} c^T x \quad \text{s.t.} \quad Ax = b \text{ and } (x, y) \in \mathcal{H}_p \ ,$$

for which we know a self-concordant barrier with parameter $(20, 8n)$. This implies that it is possible to find an approximate solution to problem (4.9) with accuracy ϵ in $\mathcal{O}\big(\sqrt{n} \log \frac{1}{\epsilon}\big)$ iterations. Moreover, since it is possible to compute in polynomial time the value of h_p and of its first two derivatives, we can conclude that problem (4.9) is solvable in polynomial time.

This argument is rather easy to generalize to the case of the cartesian product of several $\mathcal{L}^p$ cones or dual $\mathcal{L}^q_s$ cones, which shows eventually that any primal or dual l_p-norm optimization can be solved up to a given accuracy in polynomial time.

4.5 Concluding remarks

In this chapter, we have formulated l_p-norm optimization problems in a conic way and applied results from the standard conic duality theory to derive their special duality properties.

This leads in our opinion to clearer proofs, the specificity of the class of problems under study being confined to the convex cone used in the formulation. Moreover, the fundamental reason why this class of optimization problems has better duality properties than a general convex problem becomes clear: this is essentially due to the existence of a strictly interior dual solution (even if a reduction procedure involving an equivalent regularized problem has to be introduced when the original dual lacks a strictly feasible point).

It is also worthy to note that this is an example of nonsymmetric conic duality, i.e. involving cones that are not self-dual, unlike the very well-studied cases of linear, second-order and semidefinite optimization.

Another advantage of this approach is the ease to prove polynomial complexity for our problems: finding a suitable self-concordant barrier is essentially all that is needed.

In the special case where all p_i's are equal, one might think it is possible to derive those duality results with a simpler formulation relying on the standard cone involving p-norms, i.e. the p-cone defined as

$$\mathbb{L}_p^n = \left\{(x,\kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \mid \|x\|_p \leq \kappa\right\} = \left\{(x,\kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \mid \sum_{i=1}^{n} |x_i|^p \leq \kappa^p\right\}.$$

However, we were note able to reach that goal, the reason being that the homogenizing variables θ and κ^* appear to play a significant role in our approach and cannot be avoided.

Finally, we mention that this framework is general enough to be applied to other classes of structured convex problems. Chapter 5 will indeed deal with the class of problems known as geometric optimization.

CHAPTER 5

Geometric optimization

Geometric optimization is an important class of problems that has many applications, especially in engineering design. In this chapter, we provide new simplified proofs for the well-known associated duality theory, using conic optimization. After introducing suitable convex cones and studying their properties, we model geometric optimization problems with a conic formulation, which allows us to apply the powerful duality theory of conic optimization and derive the duality results valid for geometric optimization.

5.1 Introduction

Geometric optimization forms an important class of problems that enables practitioners to model a large variety of real-world applications, mostly in the field of engineering design. We refer the reader to [DPZ67, Chapter V] for two detailed case studies in mechanical engineering (use of sea power) and electrical engineering (design of a transformer).

Although not convex itself, a geometric optimization problem can be easily transformed into a convex problem, for which a Lagrangean dual can be explicitly written. Several duality results are known for this pair of problems, some being mere consequences of convexity (e.g. weak duality), others being specific to this particular class of problems (e.g. the absence of a duality gap).

These properties were first studied in the sixties, and can be found for example in the reference book of Duffin, Peterson and Zener [DPZ67]. The aim of this chapter is to derive

these results using the machinery of duality for conic optimization of Chapter 3, which has in our opinion the advantage of simplifying and clarifying the proofs.

In order to use this setting, we start by defining an appropriate convex cone that allows us to express geometric optimization problems as conic programs. The first step we take consists in studying some properties of this cone (e.g. closedness) and determine its dual. We are then in position to apply the general duality theory for conic optimization described in Chapter 3 to our problems and find in a rather seamless way the various well-known duality theorems of geometric optimization.

This chapter is organized as follows: we define and study in Section 5.2 the convex cones needed to model geometric optimization. Section 5.3 constitutes the main part of this chapter and presents new proofs of several duality theorems based on conic duality. Finally, we provide in Section 5.4 some hints on how to establish the link between our results and the classical theorems found in the literature, as well as some concluding remarks.

The approach we follow here is quite similar to the one we used in Chapter 4. However, geometric optimization differs from l_p-norm optimization in some important respects, which will be detailed later in this chapter.

5.2 Cones for geometric optimization

Let us introduce the geometric cone $\mathcal{G}^n$, which will allow us to give a conic formulation of geometric optimization problems.

5.2.1 The geometric cone

Definition 5.1. Let $n \in \mathbb{N}$. The *geometric cone* $\mathcal{G}^n$ is defined by

$$\mathcal{G}^n = \left\{ (x, \theta) \in \mathbb{R}^n_+ \times \mathbb{R}_+ \mid \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} \leq 1 \right\}$$

using in the case of a zero denominator the following convention:

$$e^{-\frac{x_i}{0}} = 0 \, .$$

We observe that this convention results in $(x, 0) \in \mathcal{G}^n$ for all $x \in \mathbb{R}^n_+$. As special cases, we mention that $\mathcal{G}^0$ is the nonnegative real line $\mathbb{R}_+$, while $\mathcal{G}^1$ is easily shown to be equal to the 2-dimensional nonnegative orthant $\mathbb{R}^2_+$.

In order to use the powerful duality theory outlined in Chapter 3, we first have to prove that $\mathcal{G}^n$ is a convex cone.

Theorem 5.1. *$\mathcal{G}^n$ is a convex cone.*

Proof. To prove that a set is a convex cone, it suffices to show that it is closed under addition and nonnegative scalar multiplication (Definition 3.1 and Theorem 3.1). Indeed, if $(x, \theta) \in \mathcal{G}^n$, $(x', \theta') \in \mathcal{G}^n$ and $\lambda \geq 0$, we have

$$\sum_{i=1}^{n} e^{-\frac{\lambda x_i}{\lambda \theta}} = \begin{cases} \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} \leq 1 & \text{if } \lambda > 0 \\ 0 \leq 1 & \text{if } \lambda = 0 \end{cases}$$

which shows that $\lambda(x, \theta) \in \mathcal{G}^n$. Looking now at $(x, \theta) + (x', \theta')$, we first consider the case $\theta > 0$ and $\theta' > 0$ and write

$$\sum_{i=1}^{n} e^{-\frac{x_i + x'_i}{\theta + \theta'}} = \sum_{i=1}^{n} \left(e^{-\frac{x_i}{\theta}}\right)^{\frac{\theta}{\theta+\theta'}} \left(e^{-\frac{x'_i}{\theta'}}\right)^{\frac{\theta'}{\theta+\theta'}} .$$

We can now apply Lemma 4.1 on each term of the sum, using vector $(e^{-\frac{x_i}{\theta}}, e^{-\frac{x'_i}{\theta'}})$ and weights $(\frac{\theta}{\theta+\theta'}, \frac{\theta'}{\theta+\theta'})$, satisfying $\frac{\theta}{\theta+\theta'} + \frac{\theta'}{\theta+\theta'} = 1$, to obtain

$$\begin{aligned} \sum_{i=1}^{n} e^{-\frac{x_i + x'_i}{\theta + \theta'}} &\leq \sum_{i=1}^{n} \frac{\theta}{\theta + \theta'}(e^{-\frac{x_i}{\theta}}) + \frac{\theta'}{\theta + \theta'}(e^{-\frac{x'_i}{\theta'}}) \\ &= \frac{\theta}{\theta + \theta'} \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} + \frac{\theta'}{\theta + \theta'} \sum_{i=1}^{n} e^{-\frac{x'_i}{\theta'}} \\ &\leq \frac{\theta}{\theta + \theta'} 1 + \frac{\theta'}{\theta + \theta'} 1 = 1 , \end{aligned}$$

while in the case of $\theta' = 0$ we have

$$\sum_{i=1}^{n} e^{-\frac{x_i + x'_i}{\theta + \theta'}} = \sum_{i=1}^{n} e^{-\frac{x_i + x'_i}{\theta}} \leq \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} \leq 1$$

(the case $\theta = 0$ is similar). We have thus shown that $(x + x', \theta + \theta') \in \mathcal{G}^n$ in all cases, and therefore that $\mathcal{G}^n$ is a convex cone. □

We now proceed to prove some properties of the geometric cone $\mathcal{G}^n$.

Theorem 5.2. *$\mathcal{G}^n$ is closed.*

Proof. Let $\{(x^k, \theta^k)\}$ a sequence of points in $\mathbb{R}^{n+1}$ such that $(x^k, \theta^k) \in \mathcal{G}^n$ for all k and $\lim_{k\to\infty}(x^k, \theta^k) = (x^\infty, \theta^\infty)$. In order to prove that $\mathcal{G}^n$ is closed, it suffices to show that $(x^\infty, \theta^\infty) \in \mathcal{G}^n$. Let us distinguish two cases:

- ⋄ $\theta^\infty > 0$. Using the easily proven fact that functions $(x_i, \theta) \mapsto e^{-\frac{x_i}{\theta}}$ are continuous on $\mathbb{R}_+ \times \mathbb{R}_{++}$, we have that

$$\sum_{i=1}^{n} e^{-\frac{x_i^\infty}{\theta^\infty}} = \sum_{i=1}^{n} \lim_{k\to\infty} e^{-\frac{x_i^k}{\theta^k}} = \lim_{k\to\infty} \sum_{i=1}^{n} e^{-\frac{x_i^k}{\theta^k}} \leq 1 ,$$

  which implies $(x^\infty, \theta^\infty) \in \mathcal{G}^n$.

- $\theta^\infty = 0$. Since $(x^k, \theta^k) \in \mathcal{G}^n$, we have $x^k \geq 0$ and thus $x^\infty \geq 0$, which implies that $(x^\infty, 0) \in \mathcal{G}^n$.

In both cases, $(x^\infty, \theta^\infty)$ is shown to belong to $\mathcal{G}^n$, which proves the claim. □

In order to use the strong duality theorem, we now proceed to identify the interior of the geometric cone.

Theorem 5.3. *The interior of $\mathcal{G}^n$ is given by*

$$\operatorname{int} \mathcal{G}^n = \left\{ (x, \theta) \in \mathbb{R}^n_{++} \times \mathbb{R}_{++} \mid \sum_{i=1}^n e^{-\frac{x_i}{\theta}} < 1 \right\} .$$

Proof. A point x belongs to the interior of a set S if and only if there exists an open ball centered at x entirely included in S. Let $(x, \theta) \in \mathcal{G}^n$. We first note that $(x, 0)$ cannot belong to $\operatorname{int} \mathcal{G}^n$, because every open ball centered at $(x, 0)$ contains a point with a negative θ component, which does not belong to the cone $\mathcal{G}^n$. Suppose $\theta > 0$ and the inequality in the definition of $\mathcal{G}^n$ is satisfied with equality, i.e.

$$\sum_{i=1}^n e^{-\frac{x_i}{\theta}} = 1 .$$

Every open ball centered at (x, θ) contains a point (x', θ') with $x' < x$ and $\theta' > \theta$, which satisfies then

$$\sum_{i=1}^n e^{-\frac{x'_i}{\theta'}} > \sum_{i=1}^n e^{-\frac{x_i}{\theta}} = 1$$

and is thus outside of $\mathcal{G}^n$, implying $(x, \theta) \notin \operatorname{int} \mathcal{G}^n$. We now show that all the remaining points that do not satisfy one of the two conditions mentioned above, i.e. the points with $\theta > 0$ satisfying the strict inequality, belong to the interior of $\mathcal{G}^n$. Let (x, θ) one of these points, and $\mathcal{B}(\epsilon)$ the open ball centered at (x, θ) with radius ϵ. Restricting ϵ to sufficiently small values (i.e. choosing $\epsilon < \theta$), we have for all points $(x', \theta') \in \mathcal{B}(\epsilon)$

$$x_i - \epsilon \leq x'_i \leq x_i + \epsilon \text{ and } 0 < \theta - \epsilon \leq \theta' \leq \theta + \epsilon ,$$

which implies

$$\frac{x'_i}{\theta'} \geq \frac{x - \epsilon}{\theta + \epsilon} \quad \text{and thus} \quad \sum_{i=1}^n e^{-\frac{x'_i}{\theta'}} \leq \sum_{i=1}^n e^{-\frac{x_i - \epsilon}{\theta + \epsilon}} \text{ for all } (x', \theta') \in \mathcal{B}(\epsilon) . \tag{5.1}$$

Taking the limit of the last right-hand side when $\epsilon \to 0$, we find

$$\lim_{\epsilon \to 0} \sum_{i=1}^n e^{-\frac{x_i - \epsilon}{\theta + \epsilon}} = \sum_{i=1}^n e^{-\frac{x_i}{\theta}} < 1$$

(because of the continuity of functions $(x_i, \theta) \mapsto e^{-\frac{x_i}{\theta}}$ on $\mathbb{R}_+ \times \mathbb{R}_{++}$). Therefore we can assume the existence of a value ϵ^* such that

$$\sum_{i=1}^n e^{-\frac{x_i - \epsilon^*}{\theta + \epsilon^*}} < 1 ,$$

which because of (5.1) will imply that

$$\sum_{i=1}^{n} e^{-\frac{x_i'}{\theta'}} < 1$$

for all $(x', \theta') \in \mathcal{B}(\epsilon^*)$. This inequality, combined with $\theta' > 0$, is sufficient to prove that the open ball $\mathcal{B}(\epsilon^*)$ is entirely included in $\mathcal{G}^n$, hence that $(x, \theta) \in \operatorname{int} \mathcal{G}^n$. □

Theorem 5.4. *$\mathcal{G}^n$ is solid and pointed.*

Proof. The fact that $0 \in \mathcal{G}^n \subseteq \mathbb{R}^{n+1}_+$ implies that $\mathcal{G}^n \cap -\mathcal{G}^n = \{0\}$, i.e. $\mathcal{G}^n$ is pointed (Definition 3.2). To prove it is solid (Definition 3.3), we simply provide a point belonging to its interior, for example $(e, \frac{1}{n})$ (where e stands for the all-one vector). We have then

$$\sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} = ne^{-n} < 1 \,,$$

because $e^n > n$ for all $n \in \mathbb{N}$, and therefore $(e, \frac{1}{n}) \in \operatorname{int} \mathcal{G}^n$. □

To summarize, $\mathcal{G}^n$ is a solid pointed close convex cone, hence suitable for conic optimization.

5.2.2 The dual geometric cone

In order to express the dual of a conic problem involving the geometric cone $\mathcal{G}^n$, we need to find an explicit description of its dual.

Theorem 5.5. *The dual of $\mathcal{G}^n$ is given by*

$$(\mathcal{G}^n)^* = \left\{ (x^*, \theta^*) \in \mathbb{R}^n_+ \times \mathbb{R} \mid \theta^* \geq \sum_{i | x_i^* > 0} x_i^* \log \frac{x_i^*}{\sum_{i=1}^n x_i^*} \right\}.$$

Proof. Using Definition 3.4 for the dual cone, we have

$$(\mathcal{G}^n)^* = \left\{ (x^*, \theta^*) \in \mathbb{R}^n \times \mathbb{R} \mid (x, \theta)^T (x^*, \theta^*) \geq 0 \text{ for all } (x, \theta) \in \mathcal{G}^n \right\}$$

(the * superscript on variables x^* and θ^* is a reminder of their dual nature). This condition on (x^*, θ^*) is equivalent to saying that the following infimum

$$\delta(x^*, \theta^*) = \inf x^T x^* + \theta\theta^* \quad \text{s.t.} \quad (x, \theta) \in \mathcal{G}^n \,.$$

has to be nonnegative. Let us distinguish the cases $\theta = 0$ and $\theta > 0$: we have that

$$\delta(x^*, \theta^*) = \min\{\delta_1(x^*, \theta^*), \delta_2(x^*, \theta^*)\}$$

with

$$\begin{cases} \delta_1(x^*, \theta^*) = \inf \; x^T x^* + \theta\theta^* & \text{s.t.} \quad (x, \theta) \in \mathcal{G}^n \text{ and } \theta = 0 \\ \delta_2(x^*, \theta^*) = \inf \; x^T x^* + \theta\theta^* & \text{s.t.} \quad (x, \theta) \in \mathcal{G}^n \text{ and } \theta > 0 \end{cases} .$$

The first of these infima can be rewritten as

$$\inf x^T x^* \quad \text{s.t.} \quad x \geq 0 \ ,$$

since $(x, 0) \in \mathcal{G}^n \Leftrightarrow x \geq 0$. It is easy to see that this infimum is equal to 0 if $x^* \geq 0$ and to $-\infty$ when $x^* \ngeq 0$. Since we are looking for points with a nonnegative infimum $\delta(x^*, \theta^*)$, we will require in the rest of this proof x^* to be nonnegative and only consider the second infimum, which is equal to

$$\inf \ \theta \left[\frac{x^T x^*}{\theta} + \theta^* \right] \quad \text{s.t.} \quad \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} \leq 1 \text{ and } (x, \theta) \in \mathbb{R}^n_+ \times \mathbb{R}_{++} \ . \tag{5.2}$$

Let us again distinguish two cases. When $x^* = 0$, this infimum becomes

$$\inf \theta\theta^* \quad \text{s.t.} \quad \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} \leq 1 \text{ and } (x, \theta) \in \mathbb{R}^n_+ \times \mathbb{R}_{++} \ ,$$

which is nonnegative if and only if $\theta^* \geq 0$, since θ can take any value in the open positive interval $]0 + \infty[$. On the other hand, if $x^* \neq 0$, we have $\sum_{i=1}^n x_i^* > 0$ and can define the auxiliary variables w_i^* by

$$w_i^* = \frac{x_i^*}{\sum_{i=1}^n x_i^*}$$

(in order to simplify notations). We write the following chain of inequalities

$$1 \geq \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} \geq \sum_{i|w_i^*>0} e^{-\frac{x_i}{\theta}} = \sum_{i|w_i^*>0} w_i^* \left(\frac{e^{-\frac{x_i}{\theta}}}{w_i^*} \right) \geq \prod_{i|w_i^*>0} \left(\frac{e^{-\frac{x_i}{\theta}}}{w_i^*} \right)^{w_i^*} \tag{5.3}$$

The second inequality comes from the fact that each term of the sum is positive (we remove some terms), and the third one uses Lemma 4.1 with weights w_i^*, noting that $\sum_{i|w_i^*>0} w_i^* = \sum_{i=1}^n w_i^* = 1$. From this last inequality we derive successively

$$\begin{aligned}
\prod_{i|w_i^*>0} e^{-\frac{x_i w_i^*}{\theta}} &\leq \prod_{i|w_i^*>0} w_i^{*w_i^*} \ , \\
-\sum_{i|w_i^*>0} \frac{x_i w_i^*}{\theta} &\leq \sum_{i|w_i^*>0} w_i^* \log w_i^* \quad \text{(taking the logarithms)} \ , \\
\sum_{i=1}^{n} \frac{x_i x_i^*}{\theta} &\geq -\sum_{i|x_i^*>0} x_i^* \log w_i^* \quad \text{(multiplying by } -\textstyle\sum_{i=1}^n x_i^*) \ , \\
\frac{x^T x^*}{\theta} + \theta^* &\geq \theta^* - \sum_{i|x_i^*>0} x_i^* \log w_i^* \ , \text{ and finally} \\
\inf_{(x,\theta)\in\mathcal{G}^n|\theta>0} \frac{x^T x^*}{\theta} + \theta^* &\geq \theta^* - \sum_{i|x_i^*>0} x_i^* \log w_i^* \ .
\end{aligned}$$

Examining carefully the chain of inequalities in (5.3), we observe that a suitable choice of (x, θ) can lead to attainment of this last infimum: namely, we need to have

- $\diamond$ $\sum_{i=1}^n e^{-\frac{x_i}{\theta}} = 1$, for the first inequality in (5.3),
- $\diamond$ $x_i \to +\infty$ for all indices i such that $w_i^* = 0$, in order to have $e^{-\frac{x_i}{\theta}} \to 0$ when $w_i^* = 0$ for the second inequality in (5.3),
- $\diamond$ all terms $(\frac{e^{-\frac{x_i}{\theta}}}{w_i^*})$ with indices such that $w_i^* > 0$ equal to each other, for the third inequality in (5.3).

These conditions are compatible: summing up the constant terms, we find

$$\frac{e^{-\frac{x_i}{\theta}}}{w_i^*} \text{ (when } w_i^* > 0) = \frac{\sum_{i|w_i^*>0} e^{-\frac{x_i}{\theta}}}{\sum_{i|w_i^*>0} w_i^*} = \sum_{i|w_i^*>0} e^{-\frac{x_i}{\theta}} \;\to\; \sum_{i=1}^n e^{-\frac{x_i}{\theta}} = 1 \,,$$

which gives $e^{-\frac{x_i}{\theta}} = w_i^*$ for all i such that $w_i^* > 0$. Summarizing, we can choose x according to

$$\begin{cases} x_i = -\theta \log w_i^* & \text{when } w_i^* > 0 \\ x_i \to +\infty & \text{when } w_i^* = 0 \end{cases},$$

which proves that

$$\inf_{(x,\theta)\in\mathcal{G}^n|\theta>0} \frac{x^T x^*}{\theta} + \theta^* = \theta^* - \sum_{i|x_i^*>0} x_i^* \log w_i^* \,. \tag{5.4}$$

Since the additional multiplicative θ in (5.2) doesn't change the sign of this infimum (because $\theta > 0$), we may conclude that it is nonnegative if and only if

$$\theta^* - \sum_{i|x_i^*>0} x_i^* \log w_i^* \geq 0 \,.$$

Combining with the special case $x^* = 0$ and the constraint $x^* \geq 0$ implied by the first infimum, we conclude that the dual cone is given by

$$(\mathcal{G}^n)^* = \left\{ (x^*, \theta^*) \in \mathbb{R}_+^n \times \mathbb{R} \mid \theta^* \geq \sum_{i|x_i^*>0} x_i^* \log w_i^* \right\},$$

as announced. □

As special cases, since $\mathcal{G}^0 = \mathbb{R}_+$ and $\mathcal{G}^1 = \mathbb{R}_+^2$, we may check that $(\mathcal{G}^0)^* = (\mathbb{R}_+)^* = \mathbb{R}_+$ and $(\mathcal{G}^1)^* = (\mathbb{R}_+^2)^* = \mathbb{R}_+^2$, as expected. These two cones are thus self-dual, but it is easy to see that geometric cones of higher dimension are not self-dual any more. To illustrate our purpose, we provide in Figure 5.1 the three-dimensional graphs of the boundary surfaces of $\mathcal{G}^2$ and $(\mathcal{G}^2)^*$.

Note 5.1. Since we have $0 \leq w_i^* \leq 1$ for all indices i, each logarithmic term appearing in this definition is nonpositive, as well as their sum, which means that $(x^*, \theta^*) \in (\mathcal{G}^n)^*$ as soon as x^* and θ^* are nonnegative. This fact could have been guessed prior to any computation: noticing that $\mathcal{G}^n \subseteq \mathbb{R}_+^{n+1}$ and $(\mathbb{R}_+^{n+1})^* = \mathbb{R}_+^{n+1}$, we immediately have that $(\mathcal{G}^n)^* \supseteq \mathbb{R}_+^{n+1}$, because taking the dual of a set inclusion reverses its direction.

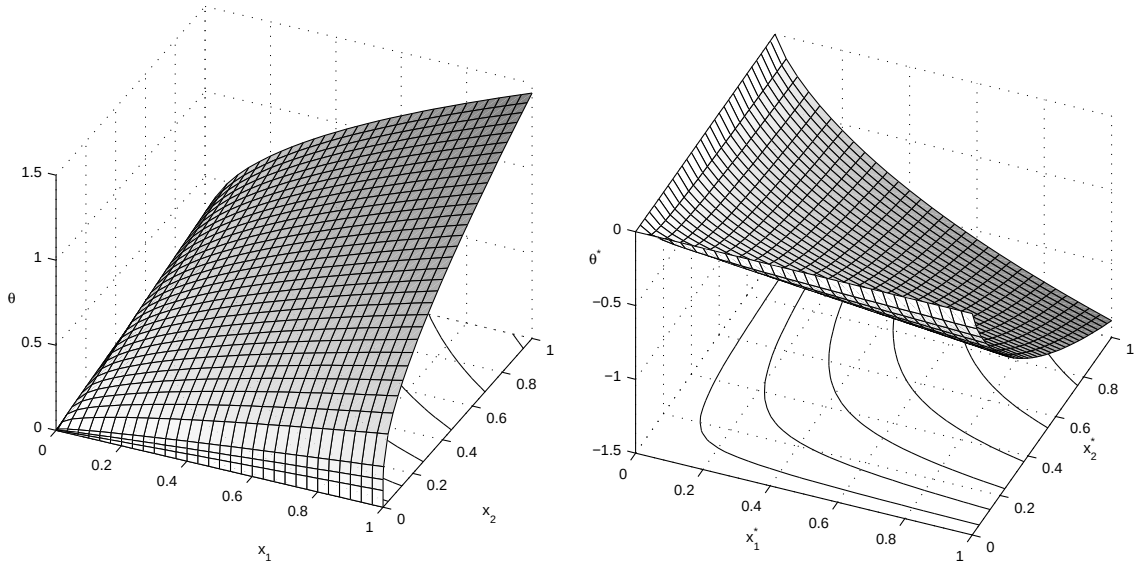


Figure 5.1: The boundary surfaces of $\mathcal{G}^2$ and $(\mathcal{G}^2)^*$.

Finding the dual of $\mathcal{G}^n$ was a little involved, but establishing its properties is straightforward.

Theorem 5.6. *$(\mathcal{G}^n)^*$ is a solid, pointed, closed convex cone. Moreover, $((\mathcal{G}^n)^*)^* = \mathcal{G}^n$.*

Proof. The proof of this fact is immediate by Theorem 3.3 since $(\mathcal{G}^n)^*$ is the dual of a solid, pointed, closed convex cone. □

The interior of $(\mathcal{G}^n)^*$ is also rather easy to obtain:

Theorem 5.7. *The interior of $(\mathcal{G}^n)^*$ is given by*

$$\operatorname{int}(\mathcal{G}^n)^* = \left\{ (x^*, \theta^*) \in \mathbb{R}^n_{++} \times \mathbb{R} \mid \theta^* > \sum_{i=1}^n x_i^* \log \frac{x_i^*}{\sum_{i=1}^n x_i^*} \right\}.$$

Proof. We first note that $(\mathcal{G}^n)^*$, a convex set, is the epigraph of the following function

$$f_n : \mathbb{R}^n_+ \mapsto \mathbb{R} : x \mapsto \sum_{i|x_i^*>0} x_i^* \log \frac{x_i^*}{\sum_{i=1}^n x_i^*},$$

which implies that f_n is convex (by definition of a convex function). Hence we can apply Lemma 7.3 in [Roc70a] to get

$$\operatorname{int}(\mathcal{G}^n)^* = \operatorname{int} \operatorname{epi} f_n = \{(x^*, \theta^*) \in \operatorname{int} \operatorname{dom} f_n \times \mathbb{R} \mid \theta^* > f_n(x^*)\},$$

which is exactly our claim since $\operatorname{int} \mathbb{R}^n_+ = \mathbb{R}^n_{++}$. □

The last piece of information we need about the pair of cones $(\mathcal{G}^n, (\mathcal{G}^n)^*)$ is its set of orthogonality conditions.

Theorem 5.8 (orthogonality conditions). *Let $v = (x, \theta) \in \mathcal{G}^n$ and $v^* = (x^*, \theta^*) \in (\mathcal{G}^n)^*$. We have $v^T v^* = 0$ if and only if one of these two sets of conditions is satisfied*

$$\theta = 0 \quad \text{and} \quad x_i x_i^* = 0 \text{ for all } i$$
$$\theta > 0 \quad \text{and} \quad \begin{cases} \sum_{i|x_i^*>0} x_i^* \log w_i^* &= \theta^* \\ (\sum_{i=1}^n x_i^*) e^{-\frac{x_i}{\theta}} &= x_i^* \text{ for all } i \end{cases} .$$

Proof. To prove this fact, we merely have to reread carefully the proof of Theorem 5.5, paying attention to the cases where the infimum is equal to zero. In the first case examined, $\theta = 0$, we have $v^T v^* = x^T x^*$. Since x and x^* are two nonnegative vectors, we have $v^T v^* = 0$ if and only if $x_i x_i^* = 0$ for every index i, which gives the first set of conditions of the theorem.

When $\theta > 0$, we first have the special case $x^* = 0$ which gives $v^T v^* = \theta\theta^*$. This quantity can only be zero if $\theta^* = 0$, i.e. when $(x^*, \theta^*) = 0$. When $x^* \neq 0$, the proof of Theorem 5.5 shows that $v^T v^*$ can only be zero when the infimum (5.4) is equal to zero and attained, which implies $\theta^* = \sum_{i|x_i^*>0} x_i^* \log w_i^*$. However, this infimum is not always attained by a finite vector (x, θ), because of the condition $x_i \to +\infty$ that is required when $w_i^* = 0$. The scalar product $v^T v^*$ is thus equal to zero only if all w_i^*'s are positive, i.e. when all x_i^*'s are positive: in this case, the two sets of equalities $\theta^* = \sum_{i|x_i^*>0} x_i^* \log w_i^*$ (to have a zero infimum) and $e^{-\frac{x_i}{\theta}} = w_i^*$ (to attain the infimum) must be satisfied.

Rephrasing this last equality as $(\sum_{i=1}^n x_i^*) e^{-\frac{x_i}{\theta}} = x_i^*$ to take into account the special case $(x^*, \theta^*) = 0$, we find the second set of conditions of our theorem. □

5.3 Duality for geometric optimization

In this section, we introduce a form of geometric optimization problems that is suitable to our purpose and prove several duality properties using the previously defined primal-dual pair of convex cones. These results are well-known and can be found e.g. in [DPZ67]. However, our presentation differs and handles problems expressed in a slightly different (but equivalent) format, and hence provides results adapted to the formulation we use. We refer the reader to Subsection 5.4.1 where the connection is made between our results and their classical counterparts.

5.3.1 Conic formulation

We start with the original formulation of a geometric optimization problem (see e.g. [DPZ67]). Let us define two sets $K = \{0, 1, 2, \ldots, r\}$ and $I = \{1, 2, \ldots, n\}$ and let $\{I_k\}_{k \in K}$ be a partition of I into $r + 1$ classes, i.e. satisfying

$$\cup_{k \in K} I_k = I \text{ and } I_k \cap I_l = \emptyset \text{ for all } k \neq l .$$

The primal geometric optimization problem is the following:

$$\inf G_0(t) \quad \text{s.t.} \quad t \in \mathbb{R}^m_{++} \text{ and } G_k(t) \leq 1 \text{ for all } k \in K \setminus \{0\} \ , \tag{OGP}$$

where t is the m-dimensional column vector we want to optimize and the functions G_k defining the objective and the constraints are so-called posynomials, given by

$$G_k : \mathbb{R}^m_{++} \mapsto \mathbb{R}_{++} : t \mapsto \sum_{i \in I_k} C_i \prod_{j=1}^{m} t_j^{a_{ij}} \ ,$$

where exponents a_{ij} are arbitrary real numbers and coefficients C_i are required to be strictly positive (hence the name *posy*nomial). These functions are very well suited for the formulation of constraints that come from the laws of physics or economics (either directly or using an empirical fit).

Although not convex itself (choose for example $G_0 : t \mapsto t^{1/2}$ as the objective, which is not a convex function), a geometric optimization problem can be easily transformed into a convex problem, for which a Lagrangean dual can be explicitly written. This transformation uses the following change of variables:

$$t_j = e^{y_j} \text{ for all } j \in \{1, 2, \ldots, m\} \ , \tag{5.5}$$

to become

$$\inf g_0(y) \quad \text{s.t.} \quad g_k(y) \leq 1 \text{ for all } k \in K \setminus \{0\} \ . \tag{OGP'}$$

The functions g_k are defined to satisfy $g_k(y) = G_k(t)$ when (5.5) holds, which means

$$g_k : \mathbb{R}^m \mapsto \mathbb{R}_{++} : y \mapsto \sum_{i \in I_k} C_i \prod_{j=1}^{m} (e^{y_j})^{a_{ij}} = \sum_{i \in I_k} e^{-c_i + \sum_{j=1}^{m} y_j a_{ij}} = \sum_{i \in I_k} e^{a_i^T y - c_i} \ ,$$

where the coefficient vector $c \in \mathbb{R}^n$ is given by $c_i = -\log C_i$ and $a_i = (a_{i1}, a_{i2}, \ldots, a_{im})^T$ is an m-dimensional column vector. Note that unlike the original variables t and coefficients C, variables y and coefficients c are not required to be strictly positive and can take any real value.

It is straightforward to check that functions g_k are now convex, hence that (OGP') is a convex optimization problem. However, we will not establish convexity directly but rather derive it from the fact that problem (OGP') can be cast as a conic optimization problem. Moreover, following others [Kla74, dJRT95, RT98], we will not use this formulation but instead work with a slight variation featuring a linear objective:

$$\sup b^T y \quad \text{s.t.} \quad g_k(y) \leq 1 \text{ for all } k \in K \ , \tag{GP}$$

where $b \in \mathbb{R}^m$ and 0 has been removed from set K.

It will be shown later that problems in the form (OGP') (and (OGP)) can be expressed in this format, and the results we are going to obtain about problem (GP) will be translated back to these more traditional settings later in Subsection 5.4.1. We can focus our attention on formulation (GP) without any loss of generality.

Let us now model problem (GP) with a conic formulation. As in Chapter 4, we will use the following useful convention: v_S (resp. M_S) denotes the restriction of column vector v (resp. matrix M) to the components (resp. rows) whose indices belong to set S. We introduce a vector of auxiliary variables $s \in \mathbb{R}^n$ to represent the exponents used in functions g_k, more precisely we let

$$s_i = c_i - a_i^T y \text{ for all } i \in I \text{ or, in matrix form, } s = c - A^T y \ ,$$

where A is a $m \times n$ matrix whose columns are a_i. Our problem becomes then

$$\sup b^T y \quad \text{s.t.} \quad s = c - A^T y \text{ and } \sum_{i \in I_k} e^{-s_i} \leq 1 \text{ for all } k \in K \ ,$$

which is readily seen to be equivalent to the following, using the definition of $\mathcal{G}^n$ (where variables θ have been fixed to 1),

$$\sup b^T y \quad \text{s.t.} \quad A^T y + s = c \text{ and } (s_{I_k}, 1) \in \mathcal{G}^{\#I_k} \text{ for all } k \in K \ ,$$

and finally to

$$\sup b^T y \quad \text{s.t.} \quad \begin{pmatrix} A^T \\ 0 \end{pmatrix} y + \begin{pmatrix} s \\ v \end{pmatrix} = \begin{pmatrix} c \\ e \end{pmatrix} \text{ and } (s_{I_k}, v_k) \in \mathcal{G}^{n_k} \text{ for all } k \in K \ , \qquad \text{(CGP)}$$

where e is the all-one vector in $\mathbb{R}^r$, $n_k = \#I_k$ and an additional vector of fictitious variables $v \in \mathbb{R}^r$ has been introduced, whose components are fixed to 1 by part of the linear constraints. This is exactly a conic optimization problem, in the dual form (CD), using variables $(\tilde{y}, \tilde{s})$, data $(\tilde{A}, \tilde{b}, \tilde{c})$ and a cone K^* such that

$$\tilde{y} = y, \ \tilde{s} = \begin{pmatrix} s \\ v \end{pmatrix}, \ \tilde{A} = \begin{pmatrix} A & 0 \end{pmatrix}, \ \tilde{b} = b, \ \tilde{c} = \begin{pmatrix} c \\ e \end{pmatrix} \text{ and } K^* = \mathcal{G}^{n_1} \times \mathcal{G}^{n_2} \times \cdots \times \mathcal{G}^{n_r} \ ,$$

where K^* has been defined according to Note 3.1, since we have to deal with multiple conic constraints involving disjoint sets of variables.

Using properties of $\mathcal{G}^n$ and $(\mathcal{G}^n)^*$ proved in the previous section, it is straightforward to show that K^* is a solid, pointed, closed convex cone whose dual is

$$(K^*)^* = K = (\mathcal{G}^{n_1})^* \times (\mathcal{G}^{n_2})^* \times \cdots \times (\mathcal{G}^{n_r})^* \ ,$$

another solid, pointed, closed convex cone, according to Theorem 5.6. This allows us to derive a dual problem to (CGP) in a completely mechanical way and find the following conic optimization problem, expressed in the primal form (CP):

$$\inf \begin{pmatrix} c^T & e^T \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} \quad \text{s.t.} \quad \begin{pmatrix} A & 0 \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} = b \text{ and } (x_{I_k}, z_k) \in (\mathcal{G}^{n_k})^* \text{ for all } k \in K \ , \qquad \text{(CGD)}$$

where $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^r$ are the vectors we optimize. This problem can be simplified: making the conic constraints explicit, we find

$$\inf c^T x + e^T z \quad \text{s.t.} \quad Ax = b, \ x_{I_k} \geq 0 \text{ and } z_k \geq \sum_{i \in I_k | x_i > 0} x_i \log \frac{x_i}{\sum_{i \in I_k} x_i} \text{ for all } k \in K \ ,$$

which can be further reduced to

$$\inf c^T x + \sum_{k\in K} \sum_{i\in I_k | x_i>0} x_i \log \frac{x_i}{\sum_{i\in I_k} x_i} \quad \text{s.t.} \quad Ax = b \text{ and } x \geq 0 \; . \tag{GD}$$

Indeed, since each variable z_k is free except for the inequality coming from the associated conic constraint, these inequalities must be satisfied with equality at each optimum solution and variables z can therefore be removed from the formulation. As could be expected, the dual problem we have just found using conic duality and our primal-dual pair of cones $(\mathcal{G}^n, (\mathcal{G}^n)^*)$ corresponds to the usual dual for problem (GP) found in the literature [Kla76, dJRT95]. We will also show later in Subsection 5.4.1 that it also allows us to derive the dual problem in the traditional formulations (OGP) and (OGP'). We end this section by pointing out that, up to now, our reasoning has been completely similar to the one used for l_p-norm optimization in Chapter 4.

5.3.2 Duality theory

We are now about to apply the various duality theorems described in Chapter 3 to geometric optimization. Our strategy will be the following: in order to prove results about the pair (GP)–(GD), we are going to apply our theorems to the conic primal-dual pair (CGP)–(CGD) and use the equivalence that holds between (CGP) and (GP) and between (CGD) and (GD). We start with the weak duality theorem.

Theorem 5.9 (Weak duality). *Let y a feasible solution for primal problem* (GP) *and x a feasible solution for dual problem* (GD)*. We have*

$$b^T y \leq c^T x + \sum_{k\in K} \sum_{i\in I_k | x_i>0} x_i \log \frac{x_i}{\sum_{i\in I_k} x_i} \; , \tag{5.6}$$

equality occurring if and only if

$$\Big(\sum_{i\in I_k} x_i\Big) e^{a_i^T y - c_i} = x_i \text{ for all } i \in I_k, k \in K \; .$$

Proof (the original proof can be found in [Roc70b] or [Kla74, §1]). On the one hand, we note that y can be easily converted to a feasible solution (y, s, v) for the conic problem (CGP), simply by choosing vectors s and v according to the linear constraints. On the other hand, x can also be converted to a feasible solution (x, z) for the conic problem (CGD), admitting the same objective value, by choosing

$$z_k = \sum_{i\in I_k | x_i>0} x_i \log \frac{x_i}{\sum_{i\in I_k} x_i} \text{ for all } k \in K \; . \tag{5.7}$$

Applying now the weak duality Theorem 3.4 to the conic primal-dual pair (CGP)–(CGD) with feasible solutions (x, z) and (y, s, v), we find the announced inequality

$$b^T y \leq c^T x + \sum_{k\in K} \sum_{i\in I_k | x_i>0} x_i \log \frac{x_i}{\sum_{i\in I_k} x_i} \; ,$$

equality occurring if and only if the orthogonality conditions given in Theorem 5.8 are satisfied for each conic constraint. Since θ corresponds here to v_k, which is always equal to 1 because of the linear constraints, we can rule out the first set of equalities (occurring where $\theta = 0$) and keep only the second set of conditions. The first of these equalities being always satisfied because of our choice of z_k, we finally conclude that equality (5.6) can occur if and only if the following set of remaining equalities is satisfied, namely

$$\Big(\sum_{i\in I_k} x_i\Big) e^{-\frac{s_i}{v_i}} = x_i \text{ for all } i \in I_k, k \in K \ ,$$

which is equivalent to our claim because of the linear constraints on s_i and v_i. □

The following theorem is an application of the strong duality Theorem 3.5, and requires therefore the existence of a specific primal feasible solution.

Theorem 5.10. *If there exists a feasible solution for the primal problem* (GP) *satisfying strictly the inequality constraints, i.e. a vector y such that*

$$g_k(y) < 1 \text{ for all } k \in K$$

we have either

- ◇ *an infeasible dual problem* (GD) *if primal problem* (GP) *is unbounded*
- ◇ *a feasible dual problem* (GD) *whose optimum objective value is attained by a feasible vector x if primal problem* (GP) *is bounded. Moreover, the optimum objective values of* (GP) *and* (GD) *are equal.*

Proof (a more classical proof can be found in [Kla74, §2]). Choosing again vectors s and v according to the linear constraints, we find a feasible solution (y, s, v) for the primal conic problem (CGP). Moreover, recalling the description of $\operatorname{int}\mathcal{G}^n$ given by Theorem 5.3, the conditions $v_k = 1 > 0$ and $g_k(y) = \sum_{i\in I_k} e^{-s_i} < 1$ ensure that (y, s, v) is a strictly feasible solution for (CGP). The strong duality Theorem 3.5 implies then that we have either

- ◇ an infeasible dual problem (CGD) if primal problem (CGP) is unbounded: this is equivalent to the first part of our claim, since it is clear that (CGP) is unbounded if and only if (GP) is unbounded and that (CGD) is infeasible if and only if (GD) is infeasible (indeed, (x, z) feasible for (CGD) implies x feasible for (GD), while x feasible for (GD) implies $(x, 0)$ feasible for (CGD)). This fact could also have been obtained as a simple consequence of weak duality Theorem 5.9.
- ◇ a feasible dual problem (CGD) whose optimum objective value is attained by a feasible vector (x, z) if primal problem (CGP) is bounded. Moreover, the optimum objective values of (CGP) and (CGD) are equal. Obviously, the finite optimum objective values of (CGP) and (GP) are equal. It is also clear that optimal variables z_k in (CGD) must attain the lower bounds defined by the conic constraints, as in (5.7), which implies that vector x is optimum for problem (GD) and has the same objective value as (x, z) in (CGD). This proves the second part of our claim.

□

Let us note again that a sufficient condition for the second case of this theorem to happen is the existence of a feasible solution for the dual problem (GD), because of the weak duality property.

The strong duality theorem can also be applied on the dual side.

Theorem 5.11. *If there exists is a strictly positive feasible solution for the dual problem* (GD)*, i.e. a vector* x *such that*

$$Ax = b \text{ and } x > 0 \ ,$$

we have either

- ◇ *an infeasible primal problem* (GP) *if dual problem* (GD) *is unbounded*
- ◇ *a feasible primal problem* (GP) *whose optimum objective value is attained by a feasible vector* y *if dual problem* (GD) *is bounded. Moreover, the optimum objective values of* (GD) *and* (GP) *are equal.*

Proof (a traditional proof can be found in [Kla74, §5]). As for the previous theorem, the first part of our claim is a direct consequence of Theorem 5.9, that does not really rely on the existence of a strictly positive x. Let us prove the second part of our claim and suppose that problem (GD) is bounded. Problem (CGD) cannot be unbounded, because each feasible solution (x, z) for (CGD) leads to a feasible x for (GD) with a lower objective (because of the conic constraints), which would also lead to an unbounded (GD). Using the description of $\text{int}(\mathcal{G}^n)^*$ given by Theorem 5.7, we find that a feasible $x > 0$ for (GD) can be easily converted to a strictly feasible solution (x, z) for (CGD), taking sufficiently large values for variables z_k (letting $z_k = 1$ for example is enough). The strong duality theorem implies thus, since (CGD) has been shown to be bounded, that problem (CGP) is feasible with an optimum objective value attained by a feasible vector (y, s, v) and equal to the dual optimum objective value of (CGD). Obviously, on the one hand, vector y is a feasible optimum solution to problem (GP), attaining the same objective value as (y, s, v) in (CGD). On the other hand, the finite optimum objective values of (CGD) and (GD) must be equal, even if no feasible solution is actually optimum (since x feasible for (GD) implies (x, z) feasible for (CGD) with the same objective value and (x, z) feasible for (CGD) implies x feasible for (GD) with a smaller or equal objective value). This is enough to prove the second part of our claim. □

To conclude this section, we prove a last theorem that involves the alternate version of the strong duality theorem. Let us introduce the following family of optimization problems, parameterized by a strictly positive parameter δ:

$$\hat{p}(\delta) \quad = \quad \sup b^T y \quad \text{s.t.} \quad g_k(y) \leq e^\delta \text{ for all } k \in K \ . \qquad (\text{GP}_\delta)$$

It is clear that each of these problems is a (strict) relaxation of problem (GP), because $e^\delta > 1$ for $\delta > 0$, hence we have $\hat{p}(\delta) \geq p^*$ for all δ. Moreover, since the feasible region of these

problems shrinks as δ tends to zero, $\hat{p}(\delta)$ is a nondecreasing function of δ and we can always define the following limit

$$\hat{p} = \lim_{\delta \to 0^+} \hat{p}(\delta) \ ,$$

which we will call the *subvalue* of problem (GP). We have the following theorem

Theorem 5.12. *If there exists a feasible solution to the dual problem* (GD)*, the subvalue of the primal problem* (GP) *is equal to the optimum objective value of the dual problem* (GD)*.*

Proof. We are going to show in fact that the primal subvalue $\hat{p}$ is equal to the subvalue p^- of the primal conic optimization problem (CGP) according to Definition 3.7. Using Theorem 3.6 on the primal-dual conic pair (CGP)–(CGD), we will find that $p^- = d^*$ (the first case of the theorem cannot happen since (GD), and hence (CGD), is feasible by hypothesis). Noting finally that the optimum objective values of (CGD) and (GD) are equal (which has been shown in the course of the previous proof) will conclude our proof.

Let us restate the definition of the subvalue p^- for problem (CGP). Defining the following family of problems, parameterized by a strictly positive parameter ϵ,

$$\sup_{(y,s,v)} b^T y \quad \text{s.t.} \quad \left\| \begin{pmatrix} A^T \\ 0 \end{pmatrix} y + \begin{pmatrix} s \\ v \end{pmatrix} - \begin{pmatrix} c \\ e \end{pmatrix} \right\| < \epsilon, \ (s_{I_k}, v_k) \in \mathcal{G}^{n_k} \ \forall k \in K \ , \qquad (\mathrm{CGP}_\epsilon)$$

whose optimum objective values will be denoted by $\bar{p}(\epsilon)$, we have that the subvalue p^- of the primal problem (CGP) is defined by

$$p^- = \lim_{\epsilon \to 0^+} \bar{p}(\epsilon) \ .$$

We first show that for all $\delta > 0$, the inequality $\hat{p}(\delta) \leq \bar{p}(\epsilon)$ holds for some well chosen value of ϵ. Let y a feasible solution for problem (GP_δ). Using the definition of g_k, constraints $g_k(y) \leq e^\delta$ easily give

$$\sum_{i \in I_k} e^{a_i^T y - c_i - \delta} \leq 1 \ ,$$

which shows that the following choice of vectors s and v

$$s_i = c_i - a_i^T y + \delta \text{ for all } i \in I \text{ and } v_k = 1 \text{ for all } k \in K$$

will be feasible for problem (CGP_ϵ) with $\epsilon = \delta\sqrt{n}$, since we have then $(s_{I_k}, v_k) \in \mathcal{G}^{n_k} \forall k \in K$ and

$$\left\| \begin{pmatrix} A^T \\ 0 \end{pmatrix} y + \begin{pmatrix} s \\ v \end{pmatrix} - \begin{pmatrix} c \\ e \end{pmatrix} \right\| = \left\| \begin{pmatrix} \delta \\ 0 \end{pmatrix} \right\| = \delta\sqrt{n} \ .$$

Since every feasible solution y for (GP_δ) gives a feasible solution (y, s, v) for (CGP_ϵ) with the same objective value, the latter problem cannot have a smaller optimum objective value and we have $\hat{p}(\delta) \leq \bar{p}(\delta\sqrt{n})$. Taking the limit when $\delta \to 0$, this shows that $\hat{p} \leq p^-$.

Let us now work in the opposite direction and let (y, s, v) a feasible solution to problem (CGP_ϵ). We have thus

$$\sum_{i \in I_k} e^{-\frac{s_i}{v_k}} \leq 1 \text{ for all } k \in K \text{ and } \left\| \begin{pmatrix} A^T y + s - c \\ v - e \end{pmatrix} \right\| < \epsilon \ ,$$

which implies

$$\begin{cases} \left|a_i^T y + s_i - c_i\right| < \epsilon \text{ for all } i \in I \\ |v_k - 1| < \epsilon \text{ for all } k \in K \end{cases} .$$

We write

$$1 \geq \sum_{i \in I_k} e^{-\frac{s_i}{v_i}} > \sum_{i \in I_k} e^{-\frac{c_i - a_i^T y + \epsilon}{1-\epsilon}} ,$$

since $v_k > 1 - \epsilon$, $s_i < c_i - a_i^T y + \epsilon$ and $x \mapsto e^{-x}$ is a monotonic decreasing function. Defining $\tilde{y} = \frac{y}{1-\epsilon}$, we have

$$\begin{aligned} \frac{c_i - a_i^T y + \epsilon}{1 - \epsilon} &= c_i - a_i^T \tilde{y} + \frac{c_i + \epsilon}{1 - \epsilon} - c_i \\ &= c_i - a_i^T \tilde{y} + \frac{\epsilon}{1 - \epsilon} (c_i + 1) \\ &\leq c_i - a_i^T \tilde{y} + \frac{\epsilon}{1 - \epsilon} (\max c_i + 1) \\ &\leq c_i - a_i^T \tilde{y} + \frac{C\epsilon}{1 - \epsilon} , \end{aligned}$$

where $C = \max c_i + 1$. We have thus

$$1 > \sum_{i \in I_k} e^{-\frac{c_i - a_i^T y + \epsilon}{1-\epsilon}} \geq \sum_{i \in I_k} e^{a_i^T \tilde{y} - c_i - \frac{C\epsilon}{1-\epsilon}} = e^{-\frac{C\epsilon}{1-\epsilon}} \sum_{i \in I_k} e^{a_i^T \tilde{y} - c_i} ,$$

which shows that

$$\sum_{i \in I_k} e^{a_i^T \tilde{y} - c_i} < e^{\frac{C\epsilon}{1-\epsilon}} ,$$

i.e. $\tilde{y}$ is a feasible solution to problem (GP_δ) with $\delta = \frac{C\epsilon}{1-\epsilon}$. Since this solution has an objective value $b^T \tilde{y}$ equal to $b^T y$ divided by $1-\epsilon$, this means that $\bar{p}(\epsilon) \leq (1-\epsilon)\hat{p}(\frac{C\epsilon}{1-\epsilon})$. Taking the limit when $\epsilon \to 0$, this shows that $p^- \leq \hat{p}$, and we may conclude that $p^- = \hat{p}$, as announced. □

5.3.3 Refined duality

The properties we have proved so far about our pair of primal-dual geometric optimization problems (GP)–(GD) are merely more or less direct consequences of their convex nature, hence valid for all convex optimization problems. In this section, we are going further and prove a result that does not hold in the general convex case, namely we show that our pair of primal-dual problems cannot have a strictly positive duality gap.

Theorem 5.13. *If both problems* (GP) *and* (GD) *are feasible, their optimum objective values are equal (but not necessarily attained).*

Proof (the original proof can be found in [Kla74, §7]). In Theorem 5.11, we proved the existence of a zero duality gap using some assumption on the dual, namely the existence of a strictly positive feasible vector. What we are going to show here is that if such a point does not exist, i.e. one or more components of vector x are zero for all feasible dual solutions, our

primal-dual pair can be reduced to an equivalent pair of problems where these components have been removed, in other words a primal-dual pair with a strictly positive feasible dual solution and a zero duality gap.

In order to use this strategy, we start by identifying the components of x that are identically equal to zero on the dual feasible region. This can be done with the following linear optimization problem:

$$\min 0 \quad \text{s.t.} \quad Ax = b \text{ and } x \geq 0 \,. \tag{BLP}$$

Since this problem has a zero objective function, all feasible solutions are optimal and we deduce that if a variable x_i is zero for all feasible solutions to problem (GD), it is zero for all optimal solution to problem (BLP). We are going to need the Goldman-Tucker Theorem 4.6 previously used in Chapter 4.

Writing the dual of problem (BLP)

$$\max b^T y \quad \text{s.t.} \quad A^T y + s = 0 \text{ and } s \geq 0 \,, \tag{BLD}$$

we find that both (BLP) and (BLD) are feasible (the former because (GD) is feasible, the latter because $(y, s) = (0, 0)$ is always a feasible solution), and thus that the Goldman-Tucker theorem is applicable.

Having now the optimal partition $(\mathcal{B}, \mathcal{N})$ at hand, we observe that the index set $\mathcal{N}$ defines exactly the set of variables x_i that are identically zero on the feasible region of problem (GD). We are now able to introduce a reduced primal-dual pair of geometric optimization problems, where variables x_i with $i \in \mathcal{N}$ have been removed. We start with the dual problem

$$\inf c_{\mathcal{B}}^T x_{\mathcal{B}} + \sum_{k \in K} \sum_{i \in I_k \cap \mathcal{B} | x_i > 0} x_i \log \frac{x_i}{\sum_{i \in I_k \cap \mathcal{B}} x_i} \quad \text{s.t.} \quad A_{\mathcal{B}} x_{\mathcal{B}} = b \text{ and } x_{\mathcal{B}} \geq 0 \,. \tag{RGD}$$

It is straightforward to check that this problem is completely equivalent to problem (GD), since the variables we removed had no contribution to the objective or to the constraints in (GD). Indeed, there is a one-to-one correspondence preserving objective values between feasible solutions $x_{\mathcal{B}}$ for (RGD) and feasible solutions x for (GD), the latter satisfying always $x_{\mathcal{N}} = 0$. Our primal geometric optimization problem becomes

$$\sup \; b^T y \quad \text{s.t.} \quad g_k^{\mathcal{B}}(y) \leq 1 \text{ for all } k \in K \,, \tag{RGP}$$

where functions $g_k^{\mathcal{B}}$ are now defined over the sets $I_k \cap \mathcal{B}$, i.e.

$$g_k^{\mathcal{B}} : \mathbb{R}^m \mapsto \mathbb{R}_{++} : y \mapsto \sum_{i \in I_k \cap \mathcal{B}} e^{a_i^T y - c_i} \,.$$

Since the Goldman-Tucker theorem implies the existence of a feasible vector x^* such that $x_{\mathcal{B}}^* > 0$ and $x_{\mathcal{N}}^* = 0$, we find that $x_{\mathcal{B}}^*$ is a strictly positive feasible solution to (RGD), which allows us to apply Theorem 5.11. Knowing that (GP) is feasible, problems (GD) and (RGD) must be bounded: we are in the second case of the theorem and can conclude that problem (RGP) attains an optimum objective value equal to the optimum objective value of problem (RGD). The last thing we have to show in order to finish our proof is that the optimum values of primal problem (GP) and its reduced version (RGP) are equal.

Let us start with $\bar{y}$, one of the optimal solutions to (RGP) that are known to exist. Our goal is thus to prove that problem (GP) has an optimum objective value equal to $b^T\bar{y}$. Unfortunately, $\bar{y}$ is not always feasible for problem (GP), since the additional terms in g_k corresponding to indices $i \in \mathcal{N}$ result in $g_k(\bar{y}) > g_k^{\mathcal{B}}(\bar{y})$ and possibly $g_k(\bar{y}) > 1$.

To solve this problem, we are going to perturb $\bar{y}$ with a suitably chosen vector, in order to make it feasible. The existence of this perturbation vector will be again derived from the Goldman-Tucker theorem, in the following manner. Let (x^*, y^*, s^*) a strictly complementary pair for problems (BLP)–(BLD). Since the optimum primal objective value is obviously equal to zero, we also have that the optimum dual objective $b^T y^*$ is equal to zero. Moreover, we have that $A^T y^* + s^* = 0$, which gives

$$A_{\mathcal{B}}^T y^* = -s_{\mathcal{B}}^* = 0 \text{ and } A_{\mathcal{N}}^T y^* = -s_{\mathcal{N}}^* < 0 \ .$$

Considering a vector y defined by $y = \bar{y} + \lambda y^*$, where λ is a positive parameter that is going to tend to $+\infty$, it is easy to check that

$$\begin{aligned} g_k(y) &= g_k^{\mathcal{B}}(y) + g_k^{\mathcal{N}}(y) \\ &= \sum_{i \in I_k \cap \mathcal{B}} e^{a_i^T y - c_i} + \sum_{i \in I_k \cap \mathcal{N}} e^{a_i^T y - c_i} \\ &= \sum_{i \in I_k \cap \mathcal{B}} e^{a_i^T \bar{y} + \lambda a_i^T y^* - c_i} + \sum_{i \in I_k \cap \mathcal{N}} e^{a_i^T \bar{y} + \lambda a_i^T y^* - c_i} \\ &= \sum_{i \in I_k \cap \mathcal{B}} e^{a_i^T \bar{y} - c_i} + \sum_{i \in I_k \cap \mathcal{N}} e^{a_i^T \bar{y} - c_i - \lambda s_i^*} \\ &= g_k^{\mathcal{B}}(\bar{y}) + \sum_{i \in I_k \cap \mathcal{N}} e^{a_i^T \bar{y} - c_i - \lambda s_i^*} \ , \end{aligned}$$

which means that

$$\lim_{\lambda \to +\infty} g_k(y) = g_k^{\mathcal{B}}(\bar{y}) \leq 1 \text{ for all } k \in K \ ,$$

since $s_i^* > 0$ for all $i \in \mathcal{N}$ implies that all the exponents in the second sum are tending to $-\infty$. Moreover, the objective value $b^T y$ is equal to $b^T\bar{y} + \lambda b^T y^* = b^T\bar{y}$ for all values of λ, since $b^T y^* = 0$.

Until now, our proof has followed the lines of the corresponding proof for l_p-norm optimization (Theorem 4.7). However, an additional difficulty arises in the case of geometric optimization. Namely, our vector y is not necessarily feasible for problem (GP) (we may have $g_k^{\mathcal{B}}(\bar{y}) = 1$ for some k and thus $g_k(y) > 1$ for all λ), and cannot therefore help us in proving that its optimum objective value is equal to $b^T\bar{y}$. We have to use a second trick, namely to "mix" y with a feasible solution to make it feasible.

Let y^0 a feasible solution to problem (GP). We know thus that

$$g_k(y^0) = g_k^{\mathcal{B}}(y^0) + g_k^{\mathcal{N}}(y^0) \leq 1 \ ,$$

which implies

$$g_k^{\mathcal{B}}(y^0) < 1$$

since $g_k^{\mathcal{N}}(y^0)$ is strictly positive. Considering now the vector $y = \delta y^0 + (1-\delta)\bar{y} + \lambda y^*$, we may write

$$\begin{aligned} g_k(y) &= g_k^{\mathcal{B}}(y) + g_k^{\mathcal{N}}(y) \\ &= g_k^{\mathcal{B}}(\delta y^0 + (1-\delta)\bar{y} + \lambda y^*) + g_k^{\mathcal{N}}(\delta y^0 + (1-\delta)\bar{y} + \lambda y^*) \\ &= g_k^{\mathcal{B}}(\delta y^0 + (1-\delta)\bar{y}) + g_k^{\mathcal{N}}(\delta y^0 + (1-\delta)\bar{y} + \lambda y^*) \ , \end{aligned}$$

this last line using again the fact that $A_{\mathcal{B}}^T y^* = 0$. We have thus

$$\lim_{\lambda \to +\infty} g_k(y) = g_k^{\mathcal{B}}(\delta y^0 + (1-\delta)\bar{y})$$

for the same reasons as above (exponents in $g_k^{\mathcal{N}}$ tending to $-\infty$). Since we know that functions g_k are convex, we have that

$$g_k^{\mathcal{B}}(\delta y^0 + (1-\delta)\bar{y}) \leq \delta g_k^{\mathcal{B}}(y^0) + (1-\delta) g_k^{\mathcal{B}}(\bar{y}) < \delta + (1-\delta) = 1 \ ,$$

which finally implies

$$\lim_{\lambda \to +\infty} g_k(y) < 1 \ .$$

Taking now a sufficiently large value of λ, we can ensure that $g_k(y) < 1$ for all k, i.e. that y is feasible for problem (GP). The objective value associated to such a solution is equal to

$$b^T y = \delta b^T y^0 + (1-\delta) b^T \bar{y} + \lambda b^T y^* = \delta b^T y^0 + (1-\delta) b^T \bar{y} \ .$$

Letting finally δ tend to zero, we obtain a sequence of solutions y, feasible for problem (GP), whose objective values converge to $b^T \bar{y}$, the optimum objective value of the reduced problem (RGP), itself equal to the optimum objective value of the dual problem (GD). This is enough to prove that the primal-dual pair of problems (GP)–(GD) has a zero duality gap. □

We also have the following corollary about the subvalue p^- of problem (GP).

Corollary 5.1. *When both problems* (GP) *and* (GD) *are feasible, the optimum objective value of problem* (GP) *is equal to its subvalue.*

Proof. Indeed, we have in general $p^* \leq p^- \leq d^*$. Since the last theorem implies $p^* = d^*$, we obtain $p^* = p^-$. □

5.3.4 Summary and examples

Let us summarize the possible situations about the primal problem (GP), and give corresponding examples to show that the results obtained so far cannot be sharpened.

- ◇ In the best possible situation, the dual problem has a strictly positive solution and is bounded: our primal problem is guaranteed by Theorem 5.11 to be feasible and have at least one finite optimal solution with a zero duality gap. Taking for example

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} , \ b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} , \ c = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ and } I_1 = \{1, 2\} \ ,$$

our primal-dual pair (CGP)–(CGD) becomes

$$\begin{array}{ll} \sup\ y_1 + y_2 & \text{s.t.} \quad e^{y_1} + e^{y_2} \leq 1 \\ \inf\ 0 + x_1 \log \dfrac{x_1}{x_1 + x_2} + x_2 \log \dfrac{x_2}{x_1 + x_2} & \text{s.t.} \quad x_1 = 1, x_2 = 1 \text{ and } x \geq 0\ . \end{array}$$

The only feasible dual solution is strictly positive, giving a bounded optimum objective value $d^* = 2 \log \frac{1}{2} = -2 \log 2$, and we may easily check (using Lemma 4.1) that $y_1 = y_2 = -\log 2$ is the only optimum primal solution, giving also $p^* = -2 \log 2$.

◇ In the case of an unbounded dual, the primal problem has to be infeasible because of the weak duality theorem. Choosing

$$A = \begin{pmatrix} 0 & 1 \end{pmatrix},\ b = 1,\ c = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \text{ and } I_1 = \{1, 2\}\ ,$$

our primal-dual pair becomes

$$\begin{array}{ll} \sup\ y_1 & \text{s.t.} \quad e^1 + e^{y_1} \leq 1 \\ \inf\ -x_1 + x_1 \log \dfrac{x_1}{x_1 + x_2} + x_2 \log \dfrac{x_2}{x_1 + x_2} & \text{s.t.} \quad x_2 = 1 \text{ and } x \geq 0\ . \end{array}$$

The dual is unbounded: the feasible solution $x = (\lambda, 1)$ for all $\lambda > 0$ has an objective value equal to $-\lambda + \lambda \log \frac{\lambda}{\lambda+1} + \log \frac{1}{\lambda+1}$, which is easily shown to tend to $(-\infty - 1 - \infty) = -\infty$ when $\lambda \to +\infty$. The primal problem is obviously infeasible, as expected.

◇ When both the primal and the dual problems are feasible but the dual does not have a strictly feasible solution, the duality gap is guaranteed by Theorem 5.13 to be equal to zero with a finite common optimal objective value, but not necessarily with attainment. Adding a third variable to our previous examples

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix},\ b = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},\ c = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } I_1 = \{1, 2, 3\}\ ,$$

our primal-dual pair becomes

$$\begin{array}{ll} \sup\ y_1 + y_2 & \text{s.t.} \quad e^{y_1} + e^{y_2} + e^{y_2 + 2y_3 - 1} \leq 1 \\ \inf\ x_3 + \displaystyle\sum_{i | x_i > 0} x_i \log \frac{x_i}{\sum_{i=1}^{3} x_i} & \text{s.t.} \quad x_1 = 1, x_2 + x_3 = 1, 2x_3 = 0 \text{ and } x \geq 0\ . \end{array}$$

The only feasible dual solution $x = (1, 1, 0)$ has a zero component and gives $d^* = -2 \log 2$. It is not too difficult to find a sequence of primal feasible solutions tending to $y = (-\log 2, -\log 2, -\infty)$ that establishes that the supremum of the primal problem is also equal to $p^* = -2 \log 2$. However, this value cannot be attained: the primal constraint implies $e^{y_1} + e^{y_2} < 1$, which in turn can be shown to force $y_1 + y_2 < -2 \log 2$ using Lemma 4.1.

◇ Our last example will demonstrate the worst situation that can happen: a feasible bounded dual problem with an infeasible primal problem. Taking

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \; b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \; c = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \text{ and } I_1 = \{1, 2\}, \; I_2 = \{3\} \; ,$$

our primal-dual pair becomes (after some simplifications in the dual objective)

$$\begin{array}{ll} \sup \; y_1 & \text{s.t.} \quad e^{y_1 - 1} + e^{y_2} \leq 1 \text{ and } e^{1 - y_1} \leq 1 \\ \inf \; x_1 - x_3 + x_1 \log \dfrac{x_1}{x_1 + x_2} & \text{s.t.} \quad x_1 - x_3 = 1, x_2 = 0 \text{ and } x \geq 0 \; . \end{array}$$

All the feasible dual solution have at least one zero component and it is not difficult to compute that $d^* = 1$ (when $x = (1, 0, 0)$, for example). It is also easy to check that the primal problem is infeasible: the first constraint implies $e^{y_1 - 1} < 1$ and thus $y_1 < 1$, while the second constraint forces $y_1 \geq 1$. However, Theorem 5.12 tells us that the primal problem has a subvalue p^- equal to d^*. Indeed, relaxing the primal problem to

$$\sup \; y_1 \quad \text{s.t.} \quad e^{y_1 - 1} + e^{y_2} \leq e^{\delta} \text{ and } e^{1 - y_1} \leq e^{\delta}$$

for any $\delta > 0$, we find $y_1 < 1 + \delta$ and $y_1 \geq 1 - \delta$, implying $1 - \delta \leq \bar{p}(\delta) < 1 + \delta$ and leading to a subvalue p^- equal to 1, as expected.

5.4 Concluding remarks

5.4.1 Original formulation

In Subsection 5.3.1, we presented a conic formulation for the primal-dual pair of geometric optimization problems (GP)–(GD) involving linear objective functions, which allowed us to derive several duality theorems. However, the traditional formulation of geometric optimization usually involves a posynomial objective function, as in (OGP) or in (OGP'), its convexified variant. In this subsection, we show that such problems can be cast as problems with a linear objective, and outline how these duality results can be translated into this traditional formulation.

Let us restate for convenience the convexified problem (OGP')

$$\inf g_0(y) \quad \text{s.t.} \quad g_k(y) \leq 1 \text{ for all } k \in K \setminus \{0\} \; , \tag{OGP'}$$

which is readily seen to be equivalent to

$$\inf e^{-y_0} \quad \text{s.t.} \quad g_0(y) \leq e^{-y_0} \text{ and } g_k(y) \leq 1 \text{ for all } k \in K \setminus \{0\} \; ,$$

introducing a new variable y_0 to express the posynomial objective. Noticing that minimizing e^{-y_0} amounts to maximizing y_0, we can rewrite this last problem as

$$\sup y_0 \quad \text{s.t.} \quad e^{y_0} g_0(y) \leq 1 \text{ and } g_k(y) \leq 1 \text{ for all } k \in K \setminus \{0\} \; ,$$

which can now be expressed in the format of (GP) as

$$\sup \tilde{b}^T \tilde{y} \quad \text{s.t.} \quad \tilde{g}_k(\tilde{y}) \leq 1 \text{ for all } k \in K \ ,$$

where vector of variables $\tilde{y} \in \mathbb{R}^{m+1}$, objective vector $\tilde{b} \in \mathbb{R}^{m+1}$ and posynomials $\tilde{g}_k$ are defined by

$$\tilde{y} = \begin{pmatrix} y_0 \\ y \end{pmatrix}, \ \tilde{b} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ \tilde{g}_0(\tilde{y}) = e^{y_0} g_0(y) \text{ and } \tilde{g}_k(\tilde{y}) = g_k(y) \text{ for all } k \in K \setminus \{0\} \ .$$

This last definition of posynomials $\tilde{g}_k$ corresponds to the following choice of column vectors $\tilde{a}_i$ (constants c_i are left unchanged):

$$\tilde{a}_i = \begin{pmatrix} 1 \\ a_i \end{pmatrix} \text{ for all } i \in I_0 \text{ and } \tilde{a}_i = \begin{pmatrix} 0 \\ a_i \end{pmatrix} \text{ for all } i \in I \setminus I_0 \ .$$

It is now easy to find a dual for problem (OGP'), based on the known dual for (GP) and our special choice of $\tilde{a}_i$ and $\tilde{b}$. Defining a matrix $\tilde{A}$ whose columns are the a_i's, i.e.

$$\tilde{A} = \begin{pmatrix} \overbrace{1, \ldots, 1}^{I_0} & \overbrace{0, \ldots, 0}^{I_k \ \forall k \neq 0} \\ & A \end{pmatrix} \ ,$$

we find the dual problem

$$\inf c^T x + \sum_{k \in K} \sum_{i \in I_k | x_i > 0} x_i \log \frac{x_i}{\sum_{i \in I_k} x_i} \quad \text{s.t.} \quad \tilde{A}x = \tilde{b} \text{ and } x \geq 0$$

or, equivalently,

$$\inf c^T x + \sum_{k \in K} \sum_{i \in I_k | x_i > 0} x_i \log \frac{x_i}{\sum_{i \in I_k} x_i} \quad \text{s.t.} \quad Ax = 0 \ , \sum_{i \in I_0} x_i = 1 \text{ and } x \geq 0 \ .$$

We can manipulate further the second part of the objective function

$$\begin{aligned} \sum_{k \in K} \sum_{i \in I_k | x_i > 0} x_i \log \frac{x_i}{\sum_{i \in I_k} x_i} &= \sum_{k \in K} \sum_{i \in I_k | x_i > 0} \Big(x_i \log x_i - x_i \log \sum_{i \in I_k} x_i \Big) \\ &= \sum_{i \in I} x_i \log x_i - \sum_{k \in K} \Big(\sum_{i \in I_k} x_i \Big) \log \Big(\sum_{i \in I_k} x_i \Big) \ , \end{aligned}$$

with the convention that $0 \log 0 = 0$, and find

$$\inf c^T x + \sum_{i \in I} x_i \log x_i - \sum_{k \in K \setminus \{0\}} \Big(\sum_{i \in I_k} x_i \Big) \log \Big(\sum_{i \in I_k} x_i \Big) \quad \text{s.t.} \quad Ax = 0 \ , \sum_{i \in I_0} x_i = 1 \text{ and } x \geq 0$$

(we could remove the term for $k = 0$ in the second sum because of the linear constraint $\sum_{i \in I_0} x_i = 1$). Noting finally that the objective of (OGP') is actually e^{-y_0} and not y_0, we find after some easy transformations the final dual problem (using $c_i = -\log C_i$)

$$\sup \prod_{i \in I} \left(\frac{C_i}{x_i} \right)^{x_i} \prod_{k \in K \setminus \{0\}} \Big(\sum_{i \in I_k} x_i \Big)^{\sum_{i \in I_k} x_i} \quad \text{s.t.} \quad Ax = 0 \ , \sum_{i \in I_0} x_i = 1 \text{ and } x \geq 0 \ . \qquad \text{(OGD')}$$

This dual problem is identical to the usual formulation that can be found in the literature [DPZ67, Chapter III]. To close this discussion, we give a few hints on how to establish links between the classical theory elaborated in [DPZ67] and the results presented in Subsections 5.3.2 and 5.3.3.

The *main lemma* in [DPZ67, Chapter IV] is essentially our weak duality theorem with its associated set of orthogonality conditions. The *first and second duality theorems* from [DPZ67, Chapter III] are basically coming from Theorems 5.10 and 5.11, i.e. the application of the strong duality theorem to the primal and the dual problems (note that the hypotheses of the first duality theorem suppose primal attainment while our version only requires primal boundedness, which is a weaker condition). We also note that the notion of *subinfimum* in [DPZ67, Chapter VI] for the primal problem is equivalent to our concept of *subvalue*. Finally, the *strong duality* theorems in [DPZ67, Chapter VI] are closely related to our Theorem 5.13, stating that a nonzero duality gap cannot occur ; the notion of *canonical* problem that is heavily used in the associated proofs corresponds to the case $\mathcal{N} = \emptyset$ in the optimal partition of problem (BLP), i.e. existence of a strictly feasible dual solution.

5.4.2 Conclusions

In this chapter, we have shown how to use the duality theory of conic optimization to derive results about geometric optimization. This process involved the introduction of a dedicated pair of convex cones $\mathcal{G}^n$ and $(\mathcal{G}^n)^*$. We would like to point out that conic optimization had so far been mostly applied to self-dual cones, i.e. to linear, second-order cone and semidefinite optimization. We hope to have demonstrated here that this theory can be equally useful in the case of a less symmetric duality.

The results we obtained can be classified into two distinct categories: most of them are direct consequences of the convex nature of geometric optimization (weak and strong duality theorems), while some of them are specific to this class of problems (absence of a duality gap). The set of problems we studied differed in fact slightly from the classical formulation of geometric optimization, because of the linear objective function.

We would like to point out that this variation in the formulation was necessary since conic optimization cannot be applied directly to geometric optimization problems cast in the traditional form. Indeed, problem (OGP) is not convex, which already prevents us from applying Lagrange duality, while the pair of problems (OGP')–(OGD') does not feature a linear objectives and hence is not suitable for a conic formulation. However, extension of our results to the case of a posynomial objective function is straightforward, as outlined in Subsection 5.4.1. We also consider the results associated to our formulation more natural than their traditional counterparts. For example, looking at the structure of the linear constraints in the dual problem (OGD'), we understand that the presence of the *normalizing* constraint $\sum_{i \in I_0} x_1 = 1$ in (OGD') is essentially a consequence of the posynomial objective, while our dual problem (GD) features a simpler set of linear constraints $Ax = b$.

The proofs presented in this chapter possess in our opinion several advantages over the classical ones: in addition to being shorter, they allow us to confine the specificity of the class of problems under study to the convex cones used in the formulation. Moreover, the

reason why geometric optimization has better duality properties than a general conic problem becomes clear: this is essentially due to the existence of a strictly feasible dual solution. Indeed, even if such an interior solution does not always exist, a regularization procedure involving an equivalent reduced problem can always be carried out and allows us to prove the absence of a duality gap in all cases (we note however that the property of primal attainment, satisfied when there exists a strictly feasible dual solution, is lost in this process and is thus no longer valid in the general case).

Duality for geometric optimization is a little weaker than for l_p-norm optimization. Namely, we do not have the primal attainment property of Theorem 4.7. The reason for this became clear in the proof of Theorem 5.13: because the solutions of the restricted primal problem were not necessarily feasible for the original primal problem, we had to perturb them with a feasible solution. Decreasing the size of this perturbation term led to a sequence of feasible solutions y, whose objective values tended to the optimal objective value of problem, but attainment was lost with this procedure since this sequence does not necessarily have a finite limit point. Indeed, the third example in Section 5.3.4 demonstrates a situation when such a sequence of feasible points tending to optimality has one component tending to $+\infty$.

A last advantage of our conic formulation is that it allows us to benefit with minimal work from the theory of polynomial interior-point methods for convex optimization developed in [NN94]. Indeed, finding a computable self-concordant barrier for our geometric cone $\mathcal{G}^n$, would be all that is needed to build an algorithm able to solve a geometric optimization problem up to a given accuracy within a polynomial number of arithmetic operations. However, the definition of cone $\mathcal{G}^n$ is not convenient and Chapter 6 will provide an alternative cone suitable for geometric optimization, which will prove much more suitable for the purpose of finding a self-concordant barrier.

CHAPTER 6

A different cone for geometric optimization

Chapters 4 and 5 have presented a new way of formulating two classical classes of structured convex problems, l_p-norm and geometric optimization, using dedicated convex cones. This approach has some advantages over the traditional formulation: it simplifies the proofs of the well-known associated duality properties (i.e. weak and strong duality) and the design of a polynomial algorithm becomes straightforward.

In this chapter, we make a step towards the description of a common framework that would include these two classes of problems. Indeed, we introduce a variant of the cone for geometric optimization $\mathcal{G}^n$ used in Chapter 5 and show it is equally suitable to formulate this class of problems. This new cone has the additional advantage of being very similar to the cone $\mathcal{L}^p$ used for l_p-norm optimization 4, which opens the way to a common generalization.

6.1 Introduction

In Chapter 5, we defined an appropriate convex cone that allowed us to express geometric optimization problems as conic programs, the aim being to apply the general duality theory for conic optimization from Chapter 3 to these problems and prove in a seamless way the various well-known duality theorems of geometric optimization. The goal of this chapter is to introduce a variation of this convex cone that preserves its ability to model geometric optimization problems but bears more resemblance with the cone that was introduced for l_p-norm optimization in Chapter 4, hinting for a common generalization of these two families

of cones.

This chapter is organized as follows: Section 6.2 introduces the convex cones needed to model geometric optimization and studies some of their properties. Section 6.4 constitutes the main part of this chapter and demonstrates how the above-mentioned cones enable us to model primal and dual geometric optimization problems in a seamless fashion. Modelling the primal problem with our first cone is rather straightforward and writing down its dual is immediate, but some work is needed to prove the equivalence with the traditional formulation of a dual geometric optimization problem. Finally, concluding remarks in Section 6.5 provide some insight about the relevance of our approach and pave the way to Chapter 7, where it is applied to a much larger class of cones.

6.2 The extended geometric cone

Let us introduce the extended geometric cone $\mathcal{G}_2^n$, which will allow us to give a conic formulation of geometric optimization problems.

Definition 6.1. Let $n \in \mathbb{N}$. The *extended geometric cone* $\mathcal{G}_2^n$ is defined by

$$\mathcal{G}_2^n = \left\{ (x, \theta, \kappa) \in \mathbb{R}_+^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \theta \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} \leq \kappa \right\}$$

using in the case of a zero denominator the following convention:

$$e^{-\frac{x_i}{0}} = 0 \, .$$

We observe that this convention results in $(x, 0, \kappa) \in \mathcal{G}_2^n$ for all $x \in \mathbb{R}_+^n$ and $\kappa \in \mathbb{R}_+$. As a special case, we mention that $\mathcal{G}_2^0$ is the 2-dimensional nonnegative orthant $\mathbb{R}_+^2$. The main difference between this cone and the original geometric cone $\mathcal{G}^n$ described in Chapter 5 is the addition of a variable κ.

In order to use the conic formulation from Chapter 3, we first prove that $\mathcal{G}_2^n$ is a convex cone.

Theorem 6.1. *$\mathcal{G}_2^n$ is a convex cone.*

Proof. Let us first introduce the following function

$$f_n : \mathbb{R}_+^n \times \mathbb{R}_+ \mapsto \mathbb{R}_+ : (x, \theta) \mapsto \sum_{i=1}^{n} \theta e^{-\frac{x_i}{\theta}} \, .$$

With the convention mentioned above, its effective domain is $\mathbb{R}_+^{n+1}$. It is straightforward to check that f_n is positively homogeneous, i.e. $f_n(\lambda x, \lambda\theta) = \lambda f_n(x, \theta)$ for $\lambda \geq 0$. Moreover, f_n is subadditive, i.e. $f_n(x + x', \theta + \theta') \leq f_n(x, \theta) + f_n(x', \theta')$. In order to show this property, we can work on each term of the sum separately, which means that we only need to prove the following inequality for all $x, x' \in \mathbb{R}$ and $\theta, \theta' \in \mathbb{R}_+$:

$$\theta e^{-\frac{x}{\theta}} + \theta' e^{-\frac{x'}{\theta'}} \geq (\theta + \theta') e^{-\frac{x+x'}{\theta+\theta'}} \, .$$

First observe that this inequality holds when $\theta = 0$ or $\theta' = 0$. For example, when $\theta = 0$, we have to check that $\theta' e^{-\frac{x'}{\theta'}} \geq \theta' e^{-\frac{x+x'}{\theta'}}$, which is a consequence of the fact that $x \mapsto e^{-x}$ is a decreasing function. When $\theta + \theta' > 0$, we use the well-known fact that $x \mapsto e^{-x}$ is a convex function on $\mathbb{R}_+$, implying that $\lambda e^{-a} + \lambda' e^{-a'} \geq e^{-(\lambda a + \lambda a')}$ for any nonnegative a, a', λ and λ' satisfying $\lambda + \lambda' = 1$. Choosing $a = \frac{x}{\theta}$, $a' = \frac{x'}{\theta'}$, $\lambda = \frac{\theta}{\theta+\theta'}$ and $\lambda' = \frac{\theta'}{\theta+\theta'}$, we find that

$$\tfrac{\theta}{\theta+\theta'} e^{-\frac{x}{\theta}} + \tfrac{\theta'}{\theta+\theta'} e^{-\frac{x'}{\theta'}} \geq e^{-\frac{\theta}{\theta+\theta'}\frac{x}{\theta} - \frac{\theta'}{\theta+\theta'}\frac{x'}{\theta'}} ,$$

which, after multiplying by $(\theta + \theta')$, lead to the desired inequality

$$\theta e^{-\frac{x}{\theta}} + \theta' e^{-\frac{x'}{\theta'}} \geq (\theta + \theta') e^{-\frac{x+x'}{\theta+\theta'}} .$$

Positive homogeneity and subadditivity imply that f_n is a convex function. Since $f_n(x, \theta) \geq 0$ for all $x \in \mathbb{R}^n_+$ and $\theta \in \mathbb{R}_+$, we notice that $\mathcal{G}^n_2$ is the epigraph of f_n, i.e.

$$\operatorname{epi} f_n = \Big\{ (x, \theta, \kappa) \in \mathbb{R}^n_+ \times \mathbb{R}_+ \times \mathbb{R} \mid f_n(x, \theta) \leq \kappa \Big\} = \mathcal{G}^n_2 .$$

$\mathcal{G}^n_2$ is thus the epigraph of a convex positively homogeneous function, hence a convex cone [Roc70a]. □

Note that the above proof bears much more resemblance with the corresponding proof for the $\mathcal{L}^p$ cone of l_p-norm optimization than the original geometric cone $\mathcal{G}^n$. We now proceed to prove some properties of the extended geometric cone $\mathcal{G}^n_2$.

Theorem 6.2. *$\mathcal{G}^n_2$ is closed.*

Proof. Let $\big\{(x^k, \theta^k, \kappa^k)\big\}$ a sequence of points in $\mathbb{R}^{n+2}_+$ such that $(x^k, \theta^k, \kappa^k) \in \mathcal{G}^n_2$ for all k and $\lim_{k\to\infty}(x^k, \theta^k, \kappa^k) = (x^\infty, \theta^\infty, \kappa^\infty)$. In order to prove that $\mathcal{G}^n_2$ is closed, it suffices to show that $(x^\infty, \theta^\infty, \kappa^\infty) \in \mathcal{G}^n_2$. Let us distinguish two cases:

- ◇ $\theta^\infty > 0$. Using the easily proven fact that functions $(x_i, \theta) \mapsto \theta e^{-\frac{x_i}{\theta}}$ are continuous on $\mathbb{R}_+ \times \mathbb{R}_{++}$, we have that

  $$\theta^\infty \sum_{i=1}^n e^{-\frac{x_i^\infty}{\theta^\infty}} = \sum_{i=1}^n \theta^\infty e^{-\frac{x_i^\infty}{\theta^\infty}} = \sum_{i=1}^n \lim_{k\to\infty} \theta^k e^{-\frac{x_i^k}{\theta^k}} = \lim_{k\to\infty} \sum_{i=1}^n \theta^k e^{-\frac{x_i^k}{\theta^k}} \leq \lim_{k\to\infty} \kappa^k = \kappa^\infty ,$$

  which implies $(x^\infty, \theta^\infty) \in \mathcal{G}^n_2$.

- ◇ $\theta^\infty = 0$. Since $(x^k, \theta^k, \kappa^k) \in \mathcal{G}^n_2$, we have $x^k \geq 0$ and $\kappa^k \geq 0$, which implies that $x^\infty \geq 0$ and $k^\infty \geq 0$. This shows that $(x^\infty, 0, \kappa^\infty) \in \mathcal{G}^n_2$.

In both cases, $(x^\infty, \theta^\infty, \kappa^\infty)$ is shown to belong to $\mathcal{G}^n_2$, which proves the claim. □

It is also interesting to identify the interior of this cone.

Theorem 6.3. *The interior of $\mathcal{G}_2^n$ is given by*

$$\operatorname{int} \mathcal{G}_2^n = \left\{ (x, \theta, \kappa) \in \mathbb{R}^n_{++} \times \mathbb{R}_{++} \times \mathbb{R}_{++} \mid \theta \sum_{i=1}^{n} e^{-\frac{x_i}{\theta}} < \kappa \right\} .$$

Proof. According to Lemma 7.3 in [Roc70a] we have

$$\operatorname{int} \mathcal{G}_2^n = \operatorname{int} \operatorname{epi} f_n = \{ (x, \theta, \kappa) \mid (x, \theta) \in \operatorname{int} \operatorname{dom} f_n \text{ and } f_n(x, \theta) < \kappa \} .$$

The above-stated result then simply follows from the fact that $\operatorname{int} \operatorname{dom} f_n = \mathbb{R}^{n+1}_{++}$. □

Corollary 6.1. *The cone $\mathcal{G}_2^n$ is solid.*

Proof. It suffices to prove that there exists at least one point that belongs to $\operatorname{int} \mathcal{G}_2^n$ (Definition 3.3). Taking for example the point $(e, \frac{1}{n}, 1)$, where e stands for the n-dimensional all-one vector, we have

$$\sum_{i=1}^{n} \theta e^{-\frac{x_i}{\theta}} = e^{-n} < 1 = \kappa ,$$

and therefore $(e, \frac{1}{n}, 1) \in \operatorname{int} \mathcal{G}_2^n$. □

We also have the following fact:

Theorem 6.4. *$\mathcal{G}_2^n$ is pointed.*

Proof. The fact that $0 \in \mathcal{G}_2^n \subseteq \mathbb{R}^{n+2}_+$ implies that $\mathcal{G}_2^n \cap -\mathcal{G}_2^n = \{0\}$, i.e. $\mathcal{G}_2^n$ is pointed (Definition 3.2). □

To summarize, $\mathcal{G}_2^n$ is a solid pointed closed convex cone, hence suitable for conic optimization.

6.3 The dual extended geometric cone

In order to express the dual of a conic problem involving the extended geometric cone $\mathcal{G}_2^n$, we need to find an explicit description of its dual.

Theorem 6.5. *The dual of $\mathcal{G}_2^n$ is given by*

$$(\mathcal{G}_2^n)^* = \left\{ (x^*, \theta^*, \kappa^*) \in \mathbb{R}^n_+ \times \mathbb{R} \times \mathbb{R}_+ \mid \theta^* \geq \sum_{0 < x_i^* < \kappa^*} \left(x_i^* \log \frac{x_i^*}{\kappa^*} - x_i^* \right) - \sum_{x_i^* \geq \kappa^*} \kappa^* \right\} .$$

Proof. Using Definition 3.4 for the dual cone, we have

$$(\mathcal{G}_2^n)^* = \big\{(x^*, \theta^*, \kappa^*) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mid (x, \theta, \kappa)^T (x^*, \theta^*, \kappa^*) \geq 0 \text{ for all } (x, \theta, \kappa) \in \mathcal{G}_2^n\big\}$$

(the * superscript on variables x^* and θ^* is a reminder of their dual nature). We first note that in the case $\theta = 0$, we may choose any $x \in \mathbb{R}^n_+$ and $\kappa \in \mathbb{R}_+$ and have $(x, 0, \kappa) \in \mathcal{G}_2^n$, which means that the product

$$(x, \theta, \kappa)^T (x^*, \theta^*, \kappa^*) = x^T x^* + \theta\theta^* + \kappa\kappa^* = x^T x^* + \kappa\kappa^*$$

has to be nonnegative for all $(x, \kappa) \in \mathbb{R}^{n+1}_+$ and is easily seen to imply that x^* and κ^* are nonnegative. We may now suppose $\theta > 0$, $(x^*, \kappa^*) \geq 0$ and write

$$\begin{aligned}
& x^T x^* + \theta\theta^* + \kappa\kappa^* \geq 0 \quad \text{for all} \quad (x, \theta, \kappa) \in \mathcal{G}_2^n \\
\Leftrightarrow \quad & x^T x^* + \theta\theta^* + \big(\theta \sum_{i=1}^n e^{-\frac{x_i}{\theta}}\big)\kappa^* \geq 0 \quad \text{for all} \quad (x, \theta) \in \mathbb{R}^n_+ \times \mathbb{R}_{++} \\
\Leftrightarrow \quad & \theta^* \geq -\frac{x^T x^*}{\theta} - \kappa^* \sum_{i=1}^n e^{-\frac{x_i}{\theta}} \quad \text{for all} \quad (x, \theta) \in \mathbb{R}^n_+ \times \mathbb{R}_{++} \\
\Leftrightarrow \quad & \theta^* \geq -t^T x^* - \kappa^* \sum_{i=1}^n e^{-t_i} \quad \text{for all} \quad t \in \mathbb{R}^n_+ \\
\Leftrightarrow \quad & \theta^* \geq -\sum_{i=1}^n \big(t_i x_i^* + \kappa^* e^{-t_i}\big) \quad \text{for all} \quad t \in \mathbb{R}^n_+ \,,
\end{aligned}$$

where we have defined $t_i = \frac{x_i}{\theta}$ for convenience. We now proceed to seek the greatest possible lower bound on θ^*, examining each term of the sum separately: we have thus to seek the minimum of

$$t_i x_i^* + \kappa^* e^{-t_i} \,.$$

The derivative of this quantity with respect to t_i being equal to $x_i^* - \kappa^* e^{-t_i}$, we have a minimum when $t_i = -\log \frac{x_i^*}{\kappa^*}$, but we have to take into account the fact that t_i has to be nonnegative, which leads us to distinguish the following three cases

- $\diamond$ $\kappa^* = 0$: in this case, the minimum is always equal to 0,
- $\diamond$ $\kappa^* > 0$ and $x_i^* \leq \kappa^*$: in this case, the minimum is attained for a nonnegative t_i and is equal to $-x_i^* \log \frac{x_i^*}{\kappa^*} + x_i^*$, this quantity being taken as equal to zero in the case of $x_i^* = 0$,
- $\diamond$ $\kappa^* > 0$ and $x_i^* > \kappa^*$: in this case, the minimum value for a nonnegative t is attained for $t = 0$ and is equal to κ^*.

These three cases can be summarized with

$$\inf_{t_i \geq 0} \big(t_i x_i^* + \kappa^* e^{-t_i}\big) = \begin{cases} -x_i^* \log \frac{x_i^*}{\kappa^*} + x_i^* & \text{when } x_i^* < \kappa^* \\ \kappa^* & \text{when } x_i^* \geq \kappa^* \end{cases} .$$

Since all of these lower bounds can be simultaneously attained with a suitable choice of t, we can state the final defining inequalities of our dual cone as

$$x^* \geq 0,\ \kappa^* \geq 0 \text{ and } \theta^* \geq \sum_{0<x_i^*<\kappa^*} \left(x_i^* \log \frac{x_i^*}{\kappa^*} - x_i^*\right) - \sum_{x_i^* \geq \kappa^*} \kappa^* \ .$$

□

As a special case, since $\mathcal{G}_2^0 = \mathbb{R}_+^2$, we check that $(\mathcal{G}_2^0)^* = (\mathbb{R}_+^2)^* = \mathbb{R}_+^2$, as expected.

Note 6.1. It can be easily checked that the lower bound on θ^* appearing in the definition is always nonpositive, which means that $(x^*, \theta^*, \kappa^*) \in (\mathcal{G}_2^n)^*$ as soon as x^* and θ^* are nonnegative. This fact could have been guessed prior to any computation: noticing that $\mathcal{G}_2^n \subseteq \mathbb{R}_+^{n+2}$ and $(\mathbb{R}_+^{n+2})^* = \mathbb{R}_+^{n+2}$, we immediately have that $(\mathcal{G}_2^n)^* \supseteq \mathbb{R}_+^{n+2}$, because taking the dual of a set inclusion reverses its direction.

Finding the dual of $\mathcal{G}_2^n$ was a little involved, but establishing its properties is straightforward.

Theorem 6.6. *$(\mathcal{G}_2^n)^*$ is a solid, pointed, closed convex cone. Moreover, $((\mathcal{G}_2^n)^*)^* = \mathcal{G}_2^n$.*

Proof. The proof of this fact is immediate by Theorem 3.3 since $(\mathcal{G}_2^n)^*$ is the dual of a solid, pointed, closed convex cone. □

The interior of $(\mathcal{G}_2^n)^*$ is also rather easy to obtain:

Theorem 6.7. *The interior of $(\mathcal{G}_2^n)^*$ is given by*

$$\operatorname{int}(\mathcal{G}_2^n)^* = \left\{ (x^*, \theta^*, \kappa^*) \in \mathbb{R}_{++}^n \times \mathbb{R} \times \mathbb{R}_{++} \mid \theta^* > \sum_{0<x_i^*<\kappa^*} \left(x_i^* \log \frac{x_i^*}{\kappa^*} - x_i^*\right) - \sum_{x_i^* \geq \kappa^*} \kappa^* \right\} .$$

Proof. We first note that $(\mathcal{G}_2^n)^*$, a convex set, is the epigraph of the following function

$$f_n : \mathbb{R}_+^n \times \mathbb{R}_+ \mapsto \mathbb{R} : (x^*, \kappa^*) \mapsto \sum_{0<x_i^*<\kappa^*} \left(x_i^* \log \frac{x_i^*}{\kappa^*} - x_i^*\right) - \sum_{x_i^* \geq \kappa^*} \kappa^* \ ,$$

which implies that f_n is convex (by definition of a convex function). Hence we can apply Lemma 7.3 from [Roc70a] to get

$$\operatorname{int}(\mathcal{G}_2^n)^* = \operatorname{int} \operatorname{epi} f_n = \left\{ (x^*, \kappa^*, \theta^*) \in \operatorname{int} \operatorname{dom} f_n \times \mathbb{R} \mid \theta^* > f_n(x^*, \kappa^*) \right\} ,$$

which is exactly our claim since $\operatorname{int}(\mathbb{R}_+^n \times \mathbb{R}_+) = \mathbb{R}_{++}^n \times \mathbb{R}_{++}$. □

6.4 A conic formulation

This is the main section of this chapter, where we show how a primal-dual pair of geometric optimization problems can be modelled using the $\mathcal{G}_2^n$ and $(\mathcal{G}_2^n)^*$ cones.

6.4.1 Modelling geometric optimization

Let us restate here for convenience the definition of the standard primal geometric optimization problem.

$$\sup\ b^T y \quad \text{s.t.} \quad g_k(y) \leq 1 \text{ for all } k \in K\ , \tag{GP}$$

where functions g_k are defined by

$$g_k : \mathbb{R}^m \mapsto \mathbb{R}_{++} : y \mapsto \sum_{i \in I_k} e^{a_i^T y - c_i}\ .$$

We first introduce a vector of auxiliary variables $s \in \mathbb{R}^n$ to represent the exponents used in functions g_k, more precisely we let

$$s_i = c_i - a_i^T y \text{ for all } i \in I \text{ or, in matrix form, } s = c - A^T y\ ,$$

where A is a $m \times n$ matrix whose columns are a_i. Our problem becomes then

$$\sup\ b^T y \quad \text{s.t.} \quad s = c - A^T y \text{ and } \sum_{i \in I_k} e^{-s_i} \leq 1 \text{ for all } k \in K\ ,$$

which is readily seen to be equivalent to the following, using the definition of $\mathcal{G}_2^n$ (where both variables κ and θ have been fixed to 1),

$$\sup\ b^T y \quad \text{s.t.} \quad A^T y + s = c \text{ and } (s_{I_k}, 1, 1) \in \mathcal{G}_2^{\#I_k} \text{ for all } k \in K\ ,$$

and finally to

$$\sup\ b^T y \quad \text{s.t.} \quad \begin{pmatrix} A^T \\ 0 \\ 0 \end{pmatrix} y + \begin{pmatrix} s \\ v \\ w \end{pmatrix} = \begin{pmatrix} c \\ e \\ e \end{pmatrix} \text{ and } (s_{I_k}, v_k, w_k) \in \mathcal{G}_2^{n_k} \text{ for all } k \in K\ , \tag{CG$_2$P}$$

where e is the all-one vector in $\mathbb{R}^r$, $n_k = \#I_k$ and two additional vectors of fictitious variables $v, w \in \mathbb{R}^r$ have been introduced, whose components are fixed to 1 by part of the linear constraints. This is exactly a conic optimization problem, in the dual form (CD), using variables $(\tilde{y}, \tilde{s})$, data $(\tilde{A}, \tilde{b}, \tilde{c})$ and a cone K^* such that

$$\tilde{y} = y,\ \tilde{s} = \begin{pmatrix} s \\ v \\ w \end{pmatrix},\ \tilde{A} = \begin{pmatrix} A & 0 & 0 \end{pmatrix},\ \tilde{b} = b,\ \tilde{c} = \begin{pmatrix} c \\ e \\ e \end{pmatrix} \text{ and } K^* = \mathcal{G}_2^{n_1} \times \mathcal{G}_2^{n_2} \times \cdots \times \mathcal{G}_2^{n_r}\ ,$$

where K^* has been defined as the Cartesian product of several disjoint extended geometric cones, according to Note 3.1, in order to deal with multiple conic constraints involving disjoint sets of variables. We also note that the fact that we have been able to model geometric optimization with a convex cone is a proof that these problems are convex.

6.4.2 Deriving the dual problem

Using properties of $\mathcal{G}_2^n$ and $(\mathcal{G}_2^n)^*$ proved in the previous section, it is straightforward to show that K^* is a solid, pointed, closed convex cone whose dual is

$$(K^*)^* = K = (\mathcal{G}_2^{n_1})^* \times (\mathcal{G}_2^{n_2})^* \times \cdots \times (\mathcal{G}_2^{n_r})^* \ ,$$

another solid, pointed, closed convex cone, according to Theorem 3.3. This allows us to derive a dual problem to ($\mathrm{CG_2P}$) in a completely mechanical way and find the following conic optimization problem, expressed in the primal form (CP):

$$\inf \begin{pmatrix} c \\ e \\ e \end{pmatrix}^T \begin{pmatrix} x \\ z \\ u \end{pmatrix} \quad \text{s.t.} \quad \begin{pmatrix} A & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ u \end{pmatrix} = b \text{ and } (x_{I_k}, z_k, u_k) \in (\mathcal{G}_2^{n_k})^* \ \forall k \in K \ , \quad (\mathrm{CG_2D})$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^r$ and $u \in \mathbb{R}^r$ are the vectors we optimize. This problem can be simplified: making the conic constraints explicit, we find

$$\inf c^T x + e^T z + e^T u \quad \text{s.t.} \quad \begin{cases} Ax = b, \ x_{I_k} \geq 0, \ u_k \geq 0 \ , \\ z_k \geq \sum_{i \in I_k | 0 < x_i < u_k} \left(x_i \log \frac{x_i}{u_k} - x_i \right) - \sum_{i \in I_k | x_i \geq u_k} u_k \ \forall k \in K \ , \end{cases}$$

which can be further reduced to

$$\inf c^T x + e^T u + \sum_{k \in K} \Big(\sum_{i \in I_k | 0 < x_i < u_k} \left(x_i \log \frac{x_i}{u_k} - x_i \right) - \sum_{i \in I_k | x_i \geq u_k} u_k \Big) \text{ s.t. } Ax = b, \ u \geq 0 \text{ and } x \geq 0 \ .$$

Indeed, since each variable z_k is free except for the inequality coming from the associated conic constraint, these inequalities must be satisfied with equality at each optimum solution and variables z can therefore be removed from the formulation. At this point, the formulation we have is simpler than the pure conic dual but is still different from the usual geometric optimization dual problem (GD) one can find in the literature. A little bit of calculus will help us to bridge the gap: let us fix k and consider the corresponding terms in the objective

$$c_{I_k}^T x_{I_k} + u_k + \sum_{i \in I_k | 0 < x_i < u_k} \left(x_i \log \frac{x_i}{u_k} - x_i \right) - \sum_{i \in I_k | x_i \geq u_k} u_k \ .$$

We would like to eliminate variable u_k, i.e. find for which value of u_k the previous quantity is minimum. It is first straightforward to check that such a value of u_k must satisfy $x_i < u_k$ for all $i \in I_k$, i.e. will only involve the first summation sign (since the value $-u_k$ in the second sum is attained as a limit case in the first sum when x_i tends to u_k from below). Taking the derivative with respect to u_k and equating it to zero we find

$$0 = 1 + \sum_{i \in I_k} x_i \frac{u_k}{x_i} (-\frac{x_i}{u_k^2}) = 1 - \frac{\sum_{i \in I_k} x_i}{u_k}, \text{ which implies } u_k = \sum_{i \in I_k} x_i \ .$$

Our objective terms become equal to

$$c_{I_k}^T x_{I_k} + \sum_{i \in I_k} x_i + \sum_{i \in I_k} \left(x_i \log \frac{x_i}{\sum_{i \in I_k} x_i} - x_i \right) = c_{I_k}^T x_{I_k} + \sum_{i \in I_k} x_i \log \frac{x_i}{\sum_{i \in I_k} x_i} \ ,$$

and leads to the following simplified dual problem

$$\inf c^T x + \sum_{k\in K} \sum_{i\in I_k | x_i>0} x_i \log \frac{x_i}{\sum_{i\in I_k} x_i} \quad \text{s.t.} \quad Ax = b \text{ and } x \geq 0 \,, \tag{GD}$$

which is, as we expected, the traditional form of a dual geometric optimization problem (see Chapter 5). This confirms the relevance of our pair of primal-dual extended geometric cones as a tool to model the class of geometric optimization problems.

6.5 Concluding remarks

In this chapter, we have formulated geometric optimization problems in a conic way using some suitably defined convex cones $\mathcal{G}_2^n$ and $(\mathcal{G}_2^n)^*$. This approach has the following advantages:

- ◇ Classical results from the standard conic duality theory can be applied to derive the duality properties of a pair of geometric optimization problems, including weak and strong duality. This was done in Chapters 4 and 5 and could be done here in a very similar fashion.
- ◇ Proving that geometric optimization problems can be solved in polynomial time can now be done rather easily: finding a suitable (i.e. computable) self-concordant barrier for cones $\mathcal{G}_2^n$ and $(\mathcal{G}_2^n)^*$ is essentially all that is needed.
- ◇ Unlike the cones $\mathcal{G}^n$ and $(\mathcal{G}^n)^*$ introduced in Chapter 5, the pair of cones we have introduced in this chapter bears some strong similarities with the cones $\mathcal{L}^p$ and $\mathcal{L}_s^q$ used in Chapters 4 for l_p-norm optimization. We can indeed write the following equivalent definition of the cone $\mathcal{L}^p$

  $$\mathcal{L}^p = \left\{(x,\theta,\kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \theta \sum_{i=1}^n \frac{1}{p_i} \left|\frac{x_i}{\theta}\right|^{p_i} \leq \kappa\right\}$$

  and compare it to

  $$\mathcal{G}_2^n = \left\{(x,\theta,\kappa) \in \mathbb{R}_+^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \theta \sum_{i=1}^n e^{-\frac{x_i}{\theta}} \leq \kappa\right\}.$$

  The only difference between those two definitions is the function that is applied to the quantities $\frac{x_i}{\theta}$ for each term of the sum: the extended geometric cone $\mathcal{G}_2^n$ uses $x \mapsto e^{-x}$ while the l_p-norm cone $\mathcal{L}^p$ is based on $x \mapsto \frac{1}{p_i}|x|^{p_i}$. This observation is the first step towards the design of a common framework that would encompass geometric optimization, l_p-norm optimization and several other kinds of structured convex problems, which is the topic of Chapter 7.

CHAPTER 7

A general framework for separable convex optimization

In this chapter, we introduce the notion of separable cone $\mathcal{K}^f$ to generalize the cones $\mathcal{L}^p$ and $\mathcal{G}_2^n$ presented in Chapters 4 and 6 to model l_p-norm and geometric optimization. We start by giving a suitable definition for this new class of cones, and then proceed to investigate their properties and compute the corresponding dual cones, which share the same structure as their primal counterparts. Special care is taken to handle in a correct manner the boundary of these cones.

This allows us to present a new class of primal-dual convex problems using the conic formulation of Chapter 3, with the potential to model many different types of constraints.

7.1 Introduction

Chapter 4 and Chapter 5 were devoted to the study of l_p-norm optimization and geometric optimization using a conic formulation. The reader has probably noticed a lot of similarity between these two chapters. Indeed, in both cases, we started by defining an *ad hoc* convex cone, studied its properties (i.e. proved closedness, solidness, pointedness and identified its interior), computed the corresponding dual cone and listed the associated orthogonality conditions.

The primal cone allowed us to model the traditional primal formulation of these two classes of problems, while the dual cone allowed us to find in a straightforward manner the classical dual associated to these problems. Furthermore, this setting allowed us to prove the associated duality properties (using the theory of conic duality, see Chapter 3) and in the case

of l_p-norm optimization to describe an interior-point polynomial-time algorithm (using the framework of self-concordant barriers, see Chapter 2). This new approach had the advantage of simplifying the proofs and giving some insight on the duality properties of these two classes of problems, which are better than in the case of a general convex problem.

The purpose of this chapter is to show that this process can be generalized to a great extent. Indeed, Chapter 6 started to bridge the gap between l_p-norm optimization and geometric optimization by giving an alternate formulation for the latter. We recall here the last remark of Section 6.5, whose purpose was to compare the following equivalent definition of the cone $\mathcal{L}^p$

$$\mathcal{L}^p = \left\{(x,\theta,\kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \theta \sum_{i=1}^n \frac{1}{p_i} \left|\frac{x_i}{\theta}\right|^{p_i} \leq \kappa\right\}$$

with the definition of the extended geometric cone

$$\mathcal{G}_2^n = \left\{(x,\theta,\kappa) \in \mathbb{R}_+^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \theta \sum_{i=1}^n e^{-\frac{x_i}{\theta}} \leq \kappa\right\}.$$

We noticed that the only difference between those two definitions was the function that was applied to the quantities $\frac{x_i}{\theta}$ for each term of the sum: the extended geometric cone $\mathcal{G}_2^n$ used the negative exponential $x \mapsto e^{-x}$ while the l_p-norm cone $\mathcal{L}^p$ was based on $x \mapsto \frac{1}{p_i}|x|^{p_i}$. The purpose of this chapter is to generalize these two cones, based on the use of an arbitrary convex function in the definition of a cone with the same structure as $\mathcal{L}^p$ and $\mathcal{G}_2^n$.

This chapter is organized as follows. In order to use the setting of conic optimization, we define in Section 7.2 a large class of convex cones called separable cones. Section 7.3 is devoted to the computation of the corresponding dual cone. Section 7.4 provides an alternate and more explicit definition of these cones. Section 7.5 shows that the class of separable cones is indeed a generalization of the l_p-norm and geometric cones presented in previous chapters. Section 7.6 presents the primal-dual pair of conic optimization problems built with our separable cones and finally Section 7.7 concludes with some possible directions for future research.

7.2 The separable cone

Let $n \in \mathbb{N}$ and let us consider a set of n convex scalar functions

$$\{f_i : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\} : x \mapsto f_i(x) \text{ for all } 1 \leq i \leq n\},$$

which can be conveniently assembled into an n-dimensional function of $\mathbb{R}$

$$f : \mathbb{R} \mapsto (\mathbb{R} \cup \{+\infty\})^n : x \mapsto \big(f_1(x), f_2(x), \ldots, f_n(x)\big).$$

Function f is obviously also convex. We will also require functions f to be proper and closed, according to the following definitions (see e.g. [Roc70a]).

Definition 7.1. A convex function $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is *proper* if it is not identically equal to $+\infty$ on $\mathbb{R}^n$, i.e. if there exists at least a point $y \in \mathbb{R}^n$ such that $f(y)$ is finite.

Definition 7.2. A convex function $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is *closed* if and only if its epigraph is closed, i.e. if

$$\{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\} = \mathrm{cl}\{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\} .$$

Theorem 7.1 in [Roc70a] states that a function f is closed if and only if it is lower semi-continuous, according to the following definition:

Definition 7.3. A function $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is *lower semi-continuous* if and only if

$$f(x) \leq \lim_{k \mapsto +\infty} f(x^k)$$

for every sequence such that x^k converges to x and the limit of $f(x^1)$, $f(x^2)$, ... exists in $\mathbb{R} \cup \{+\infty\}$.

Let us now consider the following set

$$\mathcal{K}^{\circ f} = \left\{(x,\theta,\kappa) \in \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R} \mid \theta \sum_{i=1}^{n} f_i(\frac{x_i}{\theta}) \leq \kappa\right\} .$$

The closure of this set will be define the separable cone $\mathcal{K}^f$.

Definition 7.4. The *separable cone* $\mathcal{K}^f \subseteq \mathbb{R}^{n+2}$ is defined by

$$\mathcal{K}^f = \mathrm{cl}\,\mathcal{K}^{\circ f} = \mathrm{cl}\left\{(x,\theta,\kappa) \in \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R} \mid \theta \sum_{i=1}^{n} f_i(\frac{x_i}{\theta}) \leq \kappa\right\} .$$

Comparing this with the definitions of cones $\mathcal{L}^p$, $\mathcal{G}^n$ and $\mathcal{G}_2^n$, we notice that we did not have to introduce an arbitrary convention for the case of a zero denominator, since the definition of $\mathcal{K}^{\circ f}$, which is based on the potentially undefined argument $\frac{x_i}{\theta}$, only uses strictly positive values of θ.

We first show that $\mathcal{K}^f$ is a closed convex cone, i.e. that it will be suitable for conic optimization.

Theorem 7.1. *$\mathcal{K}^f$ is a closed convex cone.*

Proof. Since $\mathcal{K}^f$ is obviously a closed set, we only have to prove that it is closed under addition and nonnegative scalar multiplication. Let us first suppose $y \in \mathcal{K}^f$ and consider $\lambda > 0$. Since $\mathcal{K}^f = \mathrm{cl}\,\mathcal{K}^{\circ f}$, we have that there exists a sequence $y^1, y^2, \ldots$ converging to y such that $y^k \in \mathcal{K}^{\circ f}$ for all k. Letting $y^k = (x^k, \theta^k, \kappa^k)$, we immediately see that $\lambda y^k = (\lambda x^k, \lambda\theta^k, \lambda\kappa^k)$ also belongs to $\mathcal{K}^{\circ f}$, since

$$\theta^k \in \mathbb{R}_{++} \Leftrightarrow \lambda\theta^k \in \mathbb{R}_{++} \text{ and } \theta^k \sum_{i=1}^{n} f_i(\frac{x_i^k}{\theta^k}) \leq \kappa^k \Leftrightarrow \lambda\theta^k \sum_{i=1}^{n} f_i(\frac{\lambda x_i^k}{\lambda\theta^k}) \leq \lambda\kappa^k \text{ for all } \lambda > 0 .$$

Taking now the limit of the sequence λy^k, we find that $\lim_{k\to+\infty} \lambda y^k = \lambda \lim_{k\to+\infty} y^k = \lambda y$ belongs to $\mathcal{K}^f$, because of the closure operation.

We also have to handle the case $\lambda = 0$, i.e. prove that 0 always belongs to $\mathcal{K}^f$. Indeed, recalling that functions f_i are proper, we have for each index i a real $\hat{x}_i$ such that $f_i(\hat{x}_i) < +\infty$. This is easily seen to imply that the point $(\hat{x}, 1, \sum_{i=1}^n f_i(\hat{x}_i))$ belongs to $\mathcal{K}^{\circ f}$. Using the above discussion, we immediately also have that $(\mu\hat{x}, \mu, \mu\sum_{i=1}^n f_i(\hat{x}_i)) \in \mathcal{K}^{\circ f}$ for all $\mu > 0$. Letting μ tend to 0, we find that the limit point of this sequence is $(0, 0, 0)$ and has to belong to the closure of $\mathcal{K}^{\circ f}$, i.e. that $0 \in \mathcal{K}^f$.

Let us now consider another point z belonging to $\mathcal{K}^f$, which implies the existence of a sequence $z^1, z^2, \ldots$ converging to z such that $z^k \in \mathcal{K}^{\circ f}$ for all k. We would like to show that $y^k + z^k$ belongs to $\mathcal{K}^{\circ f}$, since it would then imply that

$$\lim_{k\to+\infty}(y^k + z^k) = \lim_{k\to+\infty} y^k + \lim_{k\to+\infty} z^k = y + z \;,$$

which belongs to $\operatorname{cl}\mathcal{K}^{\circ f} = \mathcal{K}^f$. Indeed, letting $z^k = (x'^k, \theta'^k, \kappa'^k)$, we first check that $\theta^k + \theta'^k > 0$. Convexity of functions f_i implies then that

$$f_i\big(\frac{x_i^k + x_i'^k}{\theta^k + \theta'^k}\big) = f_i\big(\frac{\theta^k}{\theta^k + \theta'^k}\frac{x_i^k}{\theta^k} + \frac{\theta'^k}{\theta^k + \theta'^k}\frac{x_i'^k}{\theta'^k}\big) \le \frac{\theta^k}{\theta^k + \theta'^k} f_i\big(\frac{x_i^k}{\theta^k}\big) + \frac{\theta'^k}{\theta^k + \theta'^k} f_i\big(\frac{x_i'^k}{\theta'^k}\big) \;,$$

since we have $\frac{\theta^k}{\theta^k+\theta'^k} + \frac{\theta'^k}{\theta^k+\theta'^k} = 1$. This shows that

$$(\theta^k + \theta'^k)\sum_{i=1}^n f_i\big(\frac{x_i^k + x_i'^k}{\theta^k + \theta'^k}\big) \le \theta^k \sum_{i=1}^n f_i\big(\frac{x_i^k}{\theta^k}\big) + \theta'^k \sum_{i=1}^n f_i\big(\frac{x_i'^k}{\theta'^k}\big) \le \kappa^k + \kappa'^k \;,$$

i.e. that $(y^k + z^k)$ belongs to $\mathcal{K}^{\circ f}$, which concludes this proof (which was quite similar to the one we used to show that $\mathcal{L}^p$ is convex). □

Let us now identify the interior of the separable cone $\mathcal{K}^f$.

Theorem 7.2. *The interior of $\mathcal{K}^f$ is given by*

$$\begin{aligned}\operatorname{int}\mathcal{K}^f &= \operatorname{int}\mathcal{K}^{\circ f} \\ &= \Big\{(x,\theta,\kappa) \in \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R} \mid x_i \in \operatorname{int}\operatorname{dom} f_i \;\forall 1 \le i \le n \text{ and } \theta\sum_{i=1}^n f_i(\frac{x_i}{\theta}) < \kappa\Big\} \;.\end{aligned}$$

Proof. The first equality is obvious, since $\operatorname{int}\operatorname{cl} S = \operatorname{int} S$ for any set S. We note $\mathcal{K}^{\circ f}$ can be seen as the epigraph of a function g defined by

$$g : \mathbb{R}^n \times \mathbb{R}_{++} \mapsto \mathbb{R} : (x,\theta) \mapsto \theta\sum_{i=1}^n f_i(\frac{x_i}{\theta}) \;,$$

i.e. $(x,\theta,\kappa) \in \mathcal{K}^{\circ f} \Leftrightarrow g(x,\theta) \le \kappa$. Moreover, the effective domain of g is easily seen to be equal to $\operatorname{dom} f_1 \times \operatorname{dom} f_2 \times \cdots \times \operatorname{dom} f_n \times \mathbb{R}_{++}$. Using now Lemma 7.3 in [Roc70a], we find that

$$\operatorname{int}\mathcal{K}^{\circ f} = \{(x,\theta,\kappa) \in \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R} \mid x_i \in \operatorname{int}\operatorname{dom} f_i \text{ for all } 1 \le i \le n \text{ and } g(x,\theta) < \kappa\Big\} \;,$$

which is exactly what we wanted to prove. □

At this point, we make an additional assumption on our scalar functions f_i, namely we require that $\operatorname{int}\operatorname{dom} f_i \neq \emptyset$. Recall that properness of f_i only implies $\operatorname{dom} f_i \neq \emptyset$. Since we know that $\operatorname{dom} f_i$ is a convex subset in $\mathbb{R}$ [Roc70a, p. 23], i.e. an interval, we see that the only effect of this assumption is to exclude the case where $\operatorname{dom} f_i = \{a\}$, i.e. the situation where f_i is infinite everywhere except at a single point. With this assumption, we have that

Corollary 7.1. *The separable cone $\mathcal{K}^f$ is solid.*

Proof. It suffices to prove that there exists at least one point (x, θ, κ) that belongs to $\operatorname{int}\mathcal{K}^f$. The previous theorem shows this is trivially done by taking $x_i \in \operatorname{int}\operatorname{dom} f_i$ for all $1 \leq i \leq n$, $\theta = 1$ and a sufficiently large κ. □

7.3 The dual separable cone

We are now going to determine the dual cone of $\mathcal{G}^f$. In order to do that, we have to introduce the notion of conjugate function (see e.g. [Roc70a]).

Definition 7.5. The *conjugate* of the convex function $f : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\}$ is the function

$$f^* : \mathbb{R}^n \mapsto \mathbb{R} \cup \{+\infty\} : x^* \mapsto \sup_{x \in \mathbb{R}^n} \{x^T x^* - f(x)\} .$$

Theorem 12.2 in [Roc70a] states that the conjugate of a closed proper convex function is also closed, proper and convex, and that the conjugate of that conjugate is equal to the original function. We will require in addition that $\operatorname{int}\operatorname{dom} f_i^* \neq \emptyset$ as for functions f_i.

Just as we did in Chapter 4 for the $\mathcal{L}^p$ cone, it is convenient to introduce a *switched* separable cone $\mathcal{K}_s^f$, which is obtained by taking the opposite x variables and exchanging the roles of variables θ and κ (note that in the case of the $\mathcal{L}^p$ cone, the opposite sign of the dual x^* variables was hidden by the fact that the conjugate functions f_i^* were even).

Definition 7.6. The switched separable cone $\mathcal{K}_s^f \subseteq \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}_+$ is defined by

$$(x, \theta, \kappa) \in \mathcal{K}_s^f \;\Leftrightarrow\; (-x, \kappa, \theta) \in \mathcal{K}^f .$$

We are now ready to describe the dual of $\mathcal{K}^f$.

Theorem 7.3. *Let us define f^* as*

$$f^* : \mathbb{R} \mapsto (\mathbb{R} \cup \{+\infty\})^n : x \mapsto \big(f_1^*(x), f_2^*(x), \ldots, f_n^*(x)\big)$$

where f_i^* is the scalar function that is conjugate to f_i. The dual of $\mathcal{K}^f$ is $\mathcal{K}_s^{f^*}$.

Proof. Using first the fact that $(\operatorname{cl}\mathcal{C})^* = \mathcal{C}^*$ [Roc70a, p. 121], we have $(\mathcal{K}^f)^* = (\operatorname{cl}\mathcal{K}^{\circ f})^* = (\mathcal{K}^{\circ f})^*$. By Definition 3.4 of the dual cone, we have then

$$(\mathcal{K}^f)^* = \left\{ v^* \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mid v^T v^* \geq 0 \text{ for all } v \in \mathcal{K}^{\circ f} \right\} , \tag{7.1}$$

which translates into

$$(x^*, \theta^*, \kappa^*) \in (\mathcal{K}^f)^* \Leftrightarrow x^T x^* + \theta\theta^* + \kappa\kappa^* \geq 0 \text{ for all } (x, \theta, \kappa) \in \mathcal{K}^{\circ f} \ .$$

Let us suppose first that $\kappa^* > 0$. We find that

$$x^T x^* + \theta\theta^* + \kappa\kappa^* \geq 0 \ \forall (x, \theta, \kappa) \in \mathcal{K}^{\circ f} \Leftrightarrow \frac{x^T}{\theta}\frac{x^*}{\kappa^*} + \frac{\theta^*}{\kappa^*} + \frac{\kappa}{\theta} \geq 0 \ \forall (x, \theta, \kappa) \in \mathcal{K}^{\circ f} \ ,$$

which, since $\theta > 0$ and κ is only restricted to its lower bound in the definition of $\mathcal{K}^{\circ f}$, is equivalent to

$$\frac{x^T}{\theta}\frac{x^*}{\kappa^*} + \frac{\theta^*}{\kappa^*} + \frac{\theta \sum_{i=1}^n f_i(\frac{x_i}{\theta})}{\theta} \geq 0 \Leftrightarrow \frac{\theta^*}{\kappa^*} \geq -\frac{x^T}{\theta}\frac{x^*}{\kappa^*} - \sum_{i=1}^n f_i(\frac{x_i}{\theta}) \ \forall (x, \theta) \text{ s.t. } \frac{x_i}{\theta} \in \operatorname{dom} f_i \ ,$$

where we could replace condition $(x, \theta, \kappa) \in \mathcal{K}^{\circ f}$ with the simpler requirement that x_i/θ belongs to the domain of f_i for all $1 \leq i \leq n$. The key insight to have here is to note that the maximum of the right-hand side for all valid x and θ can be expressed with the conjugate functions f_i^*, since

$$f_i^*(-\frac{x_i^*}{\kappa^*}) = \sup_{y \in \mathbb{R}} \{-y\frac{x_i^*}{\kappa^*} - f_i(y)\} = \sup_{y \in \operatorname{dom} f_i} \{-y\frac{x_i^*}{\kappa^*} - f_i(y)\} = \sup_{(x_i/\theta) \in \operatorname{dom} f_i} \{-\frac{x_i}{\theta}\frac{x_i^*}{\kappa^*} - f_i(\frac{x_i}{\theta})\} \ .$$

Our condition is thus equivalent to

$$\frac{\theta^*}{\kappa^*} \geq \sum_{i=1}^n f_i^*(-\frac{x_i^*}{\kappa^*}) \Leftrightarrow \kappa^* \sum_{i=1}^n f_i^*(-\frac{x_i^*}{\kappa^*}) \leq \theta^* \ ,$$

which is exactly the same as saying that $(-x^*, \kappa^*, \theta^*) \in \mathcal{K}^{\circ f^*}$ or, using our definition of the switched cone, $(x^*, \theta^*, \kappa^*) \in \mathcal{K}_s^{\circ f^*}$.

We have finally to examine the case $\kappa^* = 0$, which will be done using an indirect approach. We have just shown that $(\mathcal{K}^f)^* \cap H = \mathcal{K}_s^{\circ f^*}$, where H is the open half-space defined by $\kappa^* > 0$, i.e. $H = \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}_{++}$. We are going to make use of Theorem 6.5 in [Roc70a], which essentially states that

$$\operatorname{cl}(C_1 \cap C_2) = \operatorname{cl} C_1 \cap \operatorname{cl} C_2 \text{ provided } \operatorname{int} C_1 \cap \operatorname{int} C_2 \neq \emptyset \ ,$$

i.e. that the closure of the intersection of two sets is the intersection of their closures, provided the intersection of their interiors is nonempty. We would like to apply this theorem to sets $(\mathcal{K}^f)^*$ and H. We first check that $\operatorname{int}(\mathcal{K}^f)^* \cap \operatorname{int} H \neq \emptyset$. Indeed, we first have that $\operatorname{int} H = H$. Moreover, it is easy to see that $\operatorname{int} \mathcal{K}_s^{\circ f^*} \cap H \neq \emptyset$ (see Theorem 7.2), which implies that $\operatorname{int}(\mathcal{K}^f)^* \cap H \neq \emptyset$ since $\mathcal{K}_s^{\circ f^*} \subseteq (\mathcal{K}^f)^*$. This allows us to apply the theorem and find that $\operatorname{cl}((\mathcal{K}^f)^* \cap H) = \operatorname{cl}(\mathcal{K}^f)^* \cap \operatorname{cl} H$ and, since $(\mathcal{K}^f)^*$ is closed, $\operatorname{cl}((\mathcal{K}^f)^* \cap H) = (\mathcal{K}^f)^* \cap \operatorname{cl} H$.

However, we cannot have a point with $\kappa^* < 0$ in $(\mathcal{K}^f)^*$. Indeed, choosing any point (x, θ, κ) in $\mathcal{K}^{\circ f}$, we have that $(x, \theta, \kappa') \in \mathcal{K}^{\circ f}$ for all $\kappa' \geq \kappa$. If $\kappa^* < 0$, we see that the quantity $x^T x^* + \theta\theta^* + \kappa\kappa^*$ can be made arbitrarily negative when $\kappa' \to +\infty$, meaning that the point $(x^*, \theta^*, \kappa^*)$ does not belong to our dual cone. Using the fact that $\operatorname{cl} H$ is the closed half-space defined by $\kappa^* \geq 0$ allows us to write that $(\mathcal{K}^f)^* \cap \operatorname{cl} H = (\mathcal{K}^f)^*$, which combined with the previous result shows that $\operatorname{cl}((\mathcal{K}^f)^* \cap H) = (\mathcal{K}^f)^*$.

Using finally the fact that $(\mathcal{K}^f)^* \cap H = \mathcal{K}_s^{\circ f^*}$, we can conclude that $(\mathcal{K}^f)^* = \operatorname{cl} \mathcal{K}_s^{\circ f^*}$, i.e. $(\mathcal{K}^f)^* = \mathcal{K}_s^{f^*}$. □

We note this proof is simpler than its counterpart for the $\mathcal{L}^p$ or the $\mathcal{G}_2^n$ cones, because the adequate use of $\mathcal{K}^{\circ f}$ instead of $\mathcal{K}^f$ which allows an elegant treatment of the case $\kappa^* = 0$.

The dual of a separable cone is thus equal, up to a change of sign and a permutation of two variables, to another separable cone based on conjugate functions.

Corollary 7.2. *We also have* $(\mathcal{K}_s^f)^* = \mathcal{K}^{f^*}$, $(\mathcal{K}^{f^*})^* = \mathcal{K}_s^f$ *and* $(\mathcal{K}_s^{f^*})^* = \mathcal{K}^f$.

Proof. Immediate considering on the one hand the symmetry between $\mathcal{K}^f$ and $\mathcal{K}_s^f$ and on the other hand the symmetry between f and f^*. □

Corollary 7.3. $\mathcal{K}^f$ *and* $\mathcal{K}_s^{f^*}$ *are solid and pointed.*

Proof. We have already proved that $\mathcal{K}^f$ is solid which, for obvious symmetry reasons, implies that its switched counterpart $\mathcal{K}_s^{f^*}$ is also solid. Since pointedness is the property that is dual to solidness (Theorem 3.3), noting that $\mathcal{K}^f = (\mathcal{K}_s^{f^*})^*$ and $\mathcal{K}_s^{f^*} = (\mathcal{K}^f)^*$ is enough to prove that $\mathcal{K}^f$ and $\mathcal{K}_s^{f^*}$ are also pointed. □

7.4 An explicit definition of $\mathcal{K}^f$

A drawback in our Definition 7.4 is the fact that it expresses $\mathcal{K}^f$ as the closure of another set, namely $\mathcal{K}^{\circ f}$. Since $\mathcal{K}^{\circ f} \subseteq \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R}$, we immediately have that $\mathcal{K}^f = \operatorname{cl} \mathcal{K}^{\circ f} \subseteq \mathbb{R}^n \times \mathbb{R}_{+} \times \mathbb{R}$, which shows that $\mathcal{K}^f$ can have points with a θ component equal to 0. This relates to the various conventions that had to be taken to handle the case of a zero denominator in the definitions of cones $\mathcal{L}^p$ and $\mathcal{G}_2^n$.

The next theorem gives an explicit definition of $\mathcal{K}^f$. It basically states that the points of $\mathcal{K}^f$ with a strictly positive θ are exactly the points of $\mathcal{K}^{\circ f}$, while the points with $\theta = 0$ can be identified using the domain of the conjugate functions f_i^*.

Theorem 7.4. *We have*

$$\mathcal{K}^f = \mathcal{K}^{\circ f} \cup \left\{ (x, 0, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R} \mid x^T x^* \leq \kappa \text{ for all } x_i^* \in \operatorname{dom} f_i^*,\ 1 \leq i \leq n \right\}.$$

Proof. A point (x, θ, κ) belongs to $\mathcal{K}^f$ if and only if there exists a sequence of points $(x^k, \theta^k, \kappa^k)$ belonging to $\mathcal{K}^{\circ f}$ such that $\theta^k \to \theta$, $x^k \to x$ and $\kappa^k \to \kappa$. Let us suppose first that $\theta > 0$.

It is obvious that points belonging to $\mathcal{K}^{\circ f}$ satisfy $\theta > 0$ and also belong to $\mathcal{K}^f$. Let us show there are no other points in $\mathcal{K}^f$ with $\theta > 0$. Using the fact that

$$\theta^k \sum_{i=1}^{n} f_i\big(\frac{x_i^k}{\theta^k}\big) \leq \kappa^k \,,$$

we can take the limit and write

$$\lim_{k \to +\infty} \theta^k \sum_{i=1}^{n} f_i\big(\frac{x_i^k}{\theta^k}\big) \leq \lim_{k \to +\infty} \kappa^k = \kappa \,. \tag{7.2}$$

Using now the lower-semicontinuity of f_i we have that

$$\lim_{k\to+\infty} \theta^k \sum_{i=1}^n f_i\big(\frac{x_i^k}{\theta^k}\big) = \theta \sum_{i=1}^n \lim_{k\to+\infty} f_i\big(\frac{x_i^k}{\theta^k}\big) \geq \theta \sum_{i=1}^n f_i\big(\frac{x_i}{\theta}\big) \ ,$$

since x_i^k/θ^k converges to x_i/θ, which shows eventually that

$$\theta \sum_{i=1}^n f_i\big(\frac{x_i}{\theta}\big) \leq \kappa \ ,$$

i.e. that (x, θ, κ) belongs to $\mathcal{K}^{\circ f}$. The sets $\mathcal{K}^f$ and $\mathcal{K}^{\circ f}$ are thus identical when $\theta > 0$.

Let us now examine the case $\theta = 0$. Using Corollary 7.2, we have that $\mathcal{K}^f = (\mathcal{K}_s^{f^*})^*$. Looking now at equation (7.1) in the proof of Theorem 7.3, we see that points of $\mathcal{K}^f$ satisfying $\theta = 0$ can be characterized by

$$(x, 0, \kappa) \in \mathcal{K}^f \Leftrightarrow x^T x^* + \kappa\kappa^* \geq 0 \text{ for all } (x^*, \theta^*, \kappa^*) \in \mathcal{K}_s^{\circ f^*} \ ,$$

which is equivalent to

$$\begin{aligned}
(x, 0, \kappa) \in \mathcal{K}^f \quad &\Leftrightarrow \quad x^T x^* + \kappa\kappa^* \geq 0 \text{ for all } (-x^*, \kappa^*, \theta^*) \in \mathcal{K}^{\circ f^*} \\
&\Leftrightarrow \quad x^T(x^*/\kappa^*) + \kappa \geq 0 \text{ for all } (-x^*, \kappa^*, \theta^*) \in \mathcal{K}^{\circ f^*} \ (\text{using } \kappa^* > 0) \\
&\Leftrightarrow \quad \kappa \geq -x^T(x^*/\kappa^*) \text{ for all } (-x^*, \kappa^*, \theta^*) \in \mathcal{K}^{\circ f^*} \\
&\Leftrightarrow \quad \kappa \geq -x^T(x^*/\kappa^*) \text{ for all } (-x_i^*/\kappa^*) \in \operatorname{dom} f_i^*, \ 1 \leq i \leq n \\
&\Leftrightarrow \quad \kappa \geq x^T x'^* \text{ for all } x_i'^* \in \operatorname{dom} f_i^*, \ 1 \leq i \leq n \ (\text{where } x'^* = x^*/\kappa^*) \ ,
\end{aligned}$$

which equivalent to the announced result. □

7.5 Back to geometric and l_p-norm optimization

Let us check that our the separable cone $\mathcal{K}^f$ generalizes the cones $\mathcal{L}^p$ and $\mathcal{G}_2^n$ introduced in Chapters 4 and 6 for l_p-norm and geometric optimization. Special care will be taken to justify the conventions we had to introduce in order to handle the cases where $\theta = 0$.

As mentioned in the introduction of this chapter, the $\mathcal{L}^p$ cone corresponds to the choice of $f_i : x \mapsto \frac{1}{p_i}|x|^{p_i}$, which is easily seen to be a proper closed convex function. Let us compute the conjugate of this function: we have

$$f_i^*(x^*) : x^* \mapsto f_i^*(x^*) = \sup_{x\in\mathbb{R}} \big\{xx^* - \frac{|x|^{p_i}}{p_i}\big\} \ .$$

Introducing parameters q_i such that $1/p_i + 1/q_i = 1$, we perform the maximization by letting the derivative of quantity appearing inside of the supremum equal to zero, which leads to $x^* = |x|^{p_i}/x$ and a supremum equal to

$$xx^* - \frac{|x|^{p_i}}{p_i} = xx^* - \frac{xx^*}{p_i} = xx^*\big(1 - \frac{1}{p_i}\big) = \frac{xx^*}{q_i} = \frac{|x|^{p_i}}{q_i} \ .$$

Using now

$$x^* = |x|^{p_i}/x \Rightarrow |x^*| = |x|^{p_i - 1} \Leftrightarrow |x^*|^{q_i} = |x|^{q_i(p_i-1)}$$

and

$$q_i(p_i - 1) = \frac{p_i - 1}{1 - 1/p_i} = (p_i - 1)\frac{p_i}{p_i - 1} = p_i \ ,$$

we find that $|x^*|^{q_i} = |x|^{p_i}$ and finally have that

$$f_i^*(x^*) = \frac{|x^*|^{q_i}}{q_i} \ .$$

Let us check our convention when $\theta = 0$. In light of Theorem 7.4, a point $(x, 0, \kappa)$ will belong to $\mathcal{K}^f$ if and only if $x^T x^* \leq \kappa$ for all $x_i^* \in \operatorname{dom} f_i^*,\ 1 \leq i \leq n$. Since $\operatorname{dom} f_i^* = \mathbb{R}$, we see that this is possible if and only if $x_i = 0$, in which case we must have $\kappa \geq 0$. This shows that

$$\mathcal{K}^f = \left\{(x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R} \mid \sum_{i=1}^n \frac{1}{p_i} \frac{|x_i|^{p_i}}{\theta^{p_i - 1}} \leq \kappa\right\} ,$$

with the convention

$$\frac{|x|}{0} = \begin{cases} +\infty & \text{if } x \neq 0 \ , \\ 0 & \text{if } x = 0 \ , \end{cases}$$

which is exactly the definition of $\mathcal{L}^p$ given in Chapter 4 (one can also easily check that the dual $\mathcal{L}_s^q$ is equivalent to $\mathcal{K}_s^{f^*}$).

The geometric cone $\mathcal{G}_2^n$ is based on $f_i : x \mapsto e^{-x}$ but features a slight difference with our separable cone $\mathcal{K}^f$ since it requires $x \geq 0$. However, the same effect can be obtained by restricting the effective domain of f_i to $\mathbb{R}_+$, i.e. letting

$$f_i : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\} : x \mapsto \begin{cases} e^{-x} & \text{when } x \geq 0 \ , \\ +\infty & \text{when } x < 0 \ . \end{cases}$$

It is straightforward to check that this function is convex proper and closed (note that the alternative choice $f_i(0) = +\infty$ does not lead to a closed function). Its conjugate function can be computed in a straightforward manner, to find

$$f_i^*(x^*) : x^* \mapsto f_i^*(x^*) = \begin{cases} -1 & \text{when } x^* \leq -1 \ , \\ x^* - x^* \log(-x^*) & \text{when } -1 < x^* < 0 \ , \\ 0 & \text{when } x^* = 0 \ , \\ +\infty & \text{when } 0 < x^* \ . \end{cases}$$

According to Theorem 7.4, a point $(x, 0, \kappa)$ will belong to $\mathcal{G}_2^n$ if and only if the product $x^T x^*$ is smaller than κ for all $x_i^* \in \operatorname{dom} f_i^*,\ 1 \leq i \leq n$. Since $\operatorname{dom} f_i^* = \mathbb{R}_-$, we see that κ can only be finite when $x \geq 0$, in which case it must satisfy $\kappa \geq 0$. This justifies the convention $e^{-\frac{x_i}{0}} = 0$ that was made in Chapter 6, since it leads to

$$\mathcal{K}^f = \left\{(x, \theta, \kappa) \in \mathbb{R}_+^n \times \mathbb{R}_+ \times \mathbb{R}_+ \mid \theta \sum_{i=1}^n e^{-\frac{x_i}{\theta}} \leq \kappa\right\} ,$$

which is exactly the original definition of $\mathcal{G}_2^n$. Let us compute its dual: we have

$$(\mathcal{K}^f)^* = \left\{(x^*, \theta^*, \kappa^*) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}_+ \mid \theta^* \sum_{i=1}^n f_i^*(-\frac{x_i^*}{\kappa^*}) \leq \theta^*\right\},$$

which is equivalent to

$$\begin{aligned} & \left\{(x^*, \theta^*, \kappa^*) \in \mathbb{R}_+^n \times \mathbb{R} \times \mathbb{R}_+ \mid \theta^* \geq \kappa^* \sum_{0<x_i^*<\kappa^*} (-\frac{x_i^*}{\kappa^*} + \frac{x_i^*}{\kappa^*} \log \frac{x_i^*}{\kappa^*}) + \kappa^* \sum_{x_i^* \geq \kappa^*} (-1)\right\} \\ = & \left\{(x^*, \theta^*, \kappa^*) \in \mathbb{R}_+^n \times \mathbb{R} \times \mathbb{R}_+ \mid \theta^* \geq \sum_{0<x_i^*<\kappa^*} (x_i^* \log \frac{x_i^*}{\kappa^*} - x_i^*) - \sum_{x_i^* \geq \kappa^*} \kappa^*\right\}, \end{aligned}$$

the original definition of the dual cone $(\mathcal{G}_2^n)^*$. We first note that the effective domain of f_i^* is responsible for restricting x^* to $\mathbb{R}_+^n$ and that we had to distinguish the cases $-x_i^*/\kappa^* \leq -1$ and $-x_i^*/\kappa^* > 1$). Moreover, the special case $\kappa^* = 0$ is handled correctly: we must have in that case $-x^T x^* \leq \theta^*$ for all $x \in \operatorname{dom} f_i$, which implies $x^* \geq 0$ and $\theta^* \geq 0$, which is exactly what is expressed by our definition.

To conclude this section, we note that it is possible to give a simpler variant of our geometric cone $\mathcal{G}_2^n$. Indeed, one can consider the negative exponential function on the whole real line, i.e. choose $f_i : x \mapsto e^{-x}$, which is again closed, proper and convex. The expression of its conjugate function is simpler

$$f_i^*(x^*) : x^* \mapsto f_i^*(x^*) = \begin{cases} x^* - x^* \log(-x^*) & \text{when } x^* < 0\,, \\ 0 & \text{when } x^* = 0\,, \\ +\infty & \text{when } 0 < x^*\,, \end{cases}$$

and leads to the following primal-dual pair of cones

$$\begin{aligned} \mathcal{K}^f &= \left\{(x, \theta, \kappa) \in \mathbb{R}^n \times \mathbb{R}_+ \times \mathbb{R} \mid \theta \sum_{i=1}^n e^{-\frac{x_i}{\theta}} \leq \kappa\right\} \\ (\mathcal{K}^f)^* &= \left\{(x^*, \theta^*, \kappa^*) \in \mathbb{R}_+^n \times \mathbb{R} \times \mathbb{R}_+ \mid \theta^* \geq \sum_{x_i^*>0} (x_i^* \log \frac{x_i^*}{\kappa^*} - x_i^*)\right\} \end{aligned}$$

(note that negative components of x are now allowed in the primal, and that the distinction between $x_i^* < \kappa^*$ and $x_i^* \geq \kappa^*$ has disappeared in the dual ; the convention $e^{-\frac{x_i}{0}} = 0$ stays valid when $x_i \geq 0$ but has to be transformed to $e^{-\frac{x_i}{0}} = +\infty$ for $x_i < 0$).

7.6 Separable convex optimization

The previous sections have introduced and studied the notion of separable cone, which encompasses the extended geometric cone $\mathcal{G}_2^n$ as well as the $\mathcal{L}^p$ cone used to model l_p-norm optimization. These separable cones are convex, closed, pointed and solid, and have a well-identified dual, which makes them perfect candidates to be used in the framework of conic optimization described in Chapter 3.

We define now the class of *separable convex optimization* and show how its primal and dual problems can be modelled using the $\mathcal{K}^f$ and $(\mathcal{K}^f)^*$ cones.

As can be expected from the above developments, the structure of this class of problems is very similar to that of l_p-norm and geometric optimization. Indeed, we define two sets $K = \{1, 2, \ldots, r\}$, $I = \{1, 2, \ldots, n\}$ and let $\{I_k\}_{k \in K}$ be a partition of I into r classes. We also choose n closed, proper convex scalar functions $f_i : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\}$, whose conjugates will be denoted by f_i^*. Finally, we assume that both $\operatorname{int} \operatorname{dom} f_i$ and $\operatorname{int} \operatorname{dom} f_i^*$ are nonempty for all $i \in I$.

The data of our problems is given by two matrices $A \in \mathbb{R}^{m \times n}$ and $F \in \mathbb{R}^{m \times r}$ (whose columns will be denoted by a_i, $i \in I$ and f_k, $k \in K$) and three column vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $d \in \mathbb{R}^r$. The primal separable convex optimization problem consists in optimizing a linear function of a column vector $y \in \mathbb{R}^m$ under a set of constraints involving functions f_i applied to linear forms, and can be written as

$$\sup \; b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} f_i(c_i - a_i^T y) \leq d_k - f_k^T y \quad \forall k \in K \; . \tag{SP}$$

Let us now model this problem with a conic formulation. We start by introducing an auxiliary vector of variables $x^* \in \mathbb{R}^n$ to represent the linear arguments of functions f_i, namely we let

$$x_i^* = c_i - a_i^T y \text{ for all } i \in I \text{ or, in matrix form, } x^* = c - A^T y \; ,$$

and we also introduce additional variables $z^* \in \mathbb{R}^r$ for the linear right-hand side of the inequalities

$$z_k^* = d_k - f_k^T y \text{ for all } k \in K \text{ or, in matrix form, } z^* = d - F^T y \; .$$

Our problem is now equivalent to

$$\sup \; b^T y \quad \text{s.t.} \quad A^T y + x^* = c, \; F^T y + z^* = d \text{ and } \sum_{i \in I_k} f_i(x_i^*) \leq z_k^* \quad \forall k \in K \; ,$$

where it is easy to plug our definition of the separable cone $\mathcal{K}^f$, provided variables θ are fixed to one

$$\sup \; b^T y \quad \text{s.t.} \quad A^T y + x^* = c, \; F^T y + z^* = d \text{ and } (x_{I_k}^*, 1, z_k^*) \in \mathcal{K}^{f^k} \; \forall k \in K$$

(where for convenience we defined $f^k = (f_i \mid i \in I_k)$ for $k \in K$). We finally introduce an additional vector of fictitious variables $v^* \in \mathbb{R}^r$ whose components are fixed to one by additional linear constraints to find

$$\sup \; b^T y \quad \text{s.t.} \quad A^T y + x^* = c, \; F^T y + z^* = d, \; v^* = e \text{ and } (x_{I_k}^*, v_k^*, z_k^*) \in \mathcal{K}^{f^k} \; \forall k \in K$$

(where e stands for the all-one vector). We point out that the description of the points belonging our separable cone when $\theta = 0$ is not used here, since variables v_r cannot be equal to zero. Rewriting the linear constraints with a single matrix equality, we end up with

$$\sup \; b^T y \quad \text{s.t.} \quad \begin{pmatrix} A^T \\ F^T \\ 0 \end{pmatrix} y + \begin{pmatrix} x^* \\ z^* \\ v^* \end{pmatrix} = \begin{pmatrix} c \\ d \\ e \end{pmatrix} \text{ and } (x_{I_k}^*, v_k^*, z_k^*) \in \mathcal{K}^{f^k} \; \forall k \in K \; , \tag{CSP}$$

which is exactly a conic optimization problem in the dual form (CD) of Chapter 3, using variables $(\tilde{y}, \tilde{s})$, data $(\tilde{A}, \tilde{b}, \tilde{c})$ and a cone $\mathcal{C}^*$ such that

$$\tilde{y} = y,\ \tilde{s} = \begin{pmatrix} x^* \\ z^* \\ v^* \end{pmatrix},\ \tilde{A} = \begin{pmatrix} A & F & 0 \end{pmatrix},\ \tilde{b} = b,\ \tilde{c} = \begin{pmatrix} c \\ d \\ e \end{pmatrix} \text{ and } \mathcal{C}^* = \mathcal{K}^{f^1} \times \mathcal{K}^{f^2} \times \cdots \times \mathcal{K}^{f^r} ,$$

where $\mathcal{C}^*$ has been defined according to Note 3.1, since we have to deal with multiple conic constraints involving disjoint sets of variables.

Using properties of $\mathcal{K}^f$ proved in the first part of this chapter, it is straightforward to show that $\mathcal{C}^*$ is a solid, pointed, closed convex cone whose dual is

$$(\mathcal{C}^*)^* = \mathcal{C} = \mathcal{K}_s^{f^{1*}} \times \mathcal{K}_s^{f^{2*}} \times \cdots \times \mathcal{K}_s^{f^{r*}} ,$$

another solid, pointed, closed convex cone (where we have defined $f^{k*} = \left(f^{i*} \mid i \in I_k\right)$ for $k \in K$). This allows us to derive a dual problem to (CSP) in a completely mechanical way and find the following conic optimization problem, expressed in the primal form (CP) (since the dual of a problem in dual form is a problem in primal form):

$$\inf \begin{pmatrix} c^T & d^T & e^T \end{pmatrix} \begin{pmatrix} x \\ z \\ v \end{pmatrix} \quad \text{s.t.} \quad \begin{pmatrix} A & F & 0 \end{pmatrix} \begin{pmatrix} x \\ z \\ v \end{pmatrix} = b \text{ and } (x_{I_k}, v_k, z_k) \in \mathcal{K}_s^{f^{k*}} \text{ for all } k \in K ,$$

which is equivalent to

$$\inf c^T x + d^T z + e^T v \quad \text{s.t.} \quad Ax + Fz = b \text{ and } (x_{I_k}, v_k, z_k) \in \mathcal{K}_s^{f^{k*}} \text{ for all } k \in K , \qquad \text{(CSD)}$$

where $x \in \mathbb{R}^n$, $z \in \mathbb{R}^r$ and $v \in \mathbb{R}^r$ are the dual variables we optimize. This problem can be simplified: developing the conic constraints, we find

$$\inf c^T x + d^T z + e^T v \quad \text{s.t.} \quad \begin{cases} Ax + Fz = b,\ z \geq 0 & \\ z_k \sum_{i \in I_k} f_i^*\left(-\frac{x_i}{z_k}\right) \leq v_k & \forall k \in K \mid z_k > 0 \\ -x_{I_k}^T x_{I_k}^* \leq v_k\ \forall x_{I_k}^* \in \operatorname{dom} f_{I_k} & \forall k \in K \mid z_k = 0 \end{cases}$$

(where $\operatorname{dom} f_{I_k}$ is the cartesian product of all $\operatorname{dom} f_i$ such that $i \in I_k$), using the explicit definition of $\mathcal{K}^f$ given by Theorem 7.4. Finally, we can remove the v variables from the formulation since they are only lower bounded by the conic constraints, and have thus to attain this lower bound at any optimal solution. We can thus directly incorporate these terms into the objective function, which leads to the final dual separable optimization problem

$$\begin{aligned} \inf\ \psi(x,z) = c^T x + d^T z + \sum_{k \in K \mid z_k > 0} z_k \sum_{i \in I_k} f_i^*\left(-\frac{x_i}{z_k}\right) - \sum_{k \in K \mid z_k = 0} \inf_{x_{I_k}^* \in \operatorname{dom} f_{I_k}} x_{I_k}^T x_{I_k}^* \qquad \text{(SD)} \\ \text{s.t. } Ax + Fz = b \text{ and } z \geq 0 . \end{aligned}$$

Finally, we note that similarly to the case of geometric optimization, the special situation where $F = 0$ can lead to a further simplification of this dual problem. Indeed, since variables z_k do not appear in the linear constraints any more, they can be optimized separately and possibly be replaced in the objective function by a closed form of their optimal value.

7.7 Concluding remarks

In this chapter, we have generalized the cones $\mathcal{G}_2^n$ and $\mathcal{L}^p$ for geometric and l_p-norm optimization with the notion of separable cone $\mathcal{K}^f$. This allowed us to present a new pair of primal-dual problems (SP)–(SD).

It is obvious that much more has to be said about this topic. We mention the following suggestions for further research:

- ◇ Duality for the pair of primal-dual problems (SP)–(SD) can be studied using the theory presented in Chapter 3. Proving weak duality should be straightforward, as well as establishing the equivalent of the strong duality Theorem 3.5. Our feeling is that it should also be possible to prove that a zero duality gap can be guaranteed without any constraint qualification, because of the scalar nature of the functions used in the formulation.
- ◇ Similarly to what was done in Chapter 4, it should be straightforward to build a self-concordant barrier for the separable cone $\mathcal{K}^f$, using as building blocks self-concordant barriers for the 2-dimensional epigraphs of functions f_i.
- ◇ Finally, this formulation has the potential to model many more classes of convex problems. We mention the following three possibilities (see [Roc70a, p. 106])
  - Let $a \in \mathbb{R}_{++}$. Functions of the type
    $$f : x \mapsto \begin{cases} -\sqrt{a^2 - x^2} & \text{if } |x| \leq a \\ +\infty & \text{if } |x| > a \end{cases} \text{ and } f^* : x^* \mapsto a\sqrt{1 + x^{*2}}$$
    are conjugate to each other, and could help modelling problems involving square roots or describing circles and ellipses.
  - Let $0 < p < 1$ and $-\infty < q < 0$ such that $1/p + 1/q = 1$. Functions of the type
    $$f : x \mapsto \begin{cases} -\frac{1}{p}x^p & \text{if } x \geq 0 \\ +\infty & \text{if } x < 0 \end{cases} \text{ and } f^* : x^* \mapsto \begin{cases} -\frac{1}{q}(-x^*)^q & \text{if } x^* < 0 \\ +\infty & \text{if } x^* \geq 0 \end{cases}$$
    are conjugate to each other, and appear to be able to model so-called *CES* functions [HvM97], which happen to be useful in production and consumer theory [Sat75].
  - Functions
    $$f : x \mapsto \begin{cases} -\frac{1}{2} - \log x & \text{if } x > 0 \\ +\infty & \text{if } x \leq 0 \end{cases} \text{ and } f^* : x^* \mapsto \begin{cases} -\frac{1}{2} - \log(-x^*) & \text{if } x^* < 0 \\ +\infty & \text{if } x^* \geq 0 \end{cases}$$
    are conjugate to each other, and could be used in problems involving logarithms. They also feature the property that $f^*(x^*) = f(-x^*)$, which could add another level of symmetry between the corresponding primal $\mathcal{K}^f$ and dual $\mathcal{K}_s^{f^*}$ cones.

  We also point out that the definition of our separable convex optimization problems allows the use of different types of cones within the same constraint, which can lead for example to the formulation of a mixed geometric-l_p-norm optimization problem.

Part III

Approximations

CHAPTER 8

Approximating geometric optimization with l_p-norm optimization

In this chapter, we demonstrate how to approximate geometric optimization with l_p-norm optimization. These two classes of problems are well known in structured convex optimization. We describe a family of l_p-norm optimization problems that can be made arbitrarily close to a geometric optimization problem, and show that the dual problems for these approximations are also approximating the dual geometric optimization problem. Finally, we use these approximations and the duality theory for l_p-norm optimization to derive simple proofs of the weak and strong duality theorems for geometric optimization.

8.1 Introduction

Let us recall first for convenience the formulation of the primal l_p-norm optimization problem (Pl_p) presented in chapter 4. Given two sets $K = \{1, 2, \ldots, r\}$ and $I = \{1, 2, \ldots, n\}$, we let $\{I_k\}_{k \in K}$ be a partition of I into r classes. The problem data is given by two matrices $A \in \mathbb{R}^{m \times n}$ and $F \in \mathbb{R}^{m \times r}$ (whose columns are be denoted by a_i, $i \in I$ and f_k, $k \in K$) and four column vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}^r$ and $p \in \mathbb{R}^n$ such that $p_i > 1 \ \forall i \in I$. The primal l_p-norm optimization problem is

$$\sup \ b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} \frac{1}{p_i} \left| c_i - a_i^T y \right|^{p_i} \leq d_k - f_k^T y \quad \forall k \in K \ . \tag{Pl_p}$$

The purpose of this chapter is to show that this category of problems can be used to approximate another famous class of problems known as geometric optimization [DPZ67], presented in Chapter 5.

Using the same notations as above for sets K and I_k, $k \in K$, matrix A and vectors b, c and a_i, $i \in I$, we recall for convenience that the primal geometric optimization problem can be stated as

$$\sup\ b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} e^{a_i^T y - c_i} \leq 1\ \forall k \in K \tag{GP}$$

We will start by presenting in Section 8.2 an approximation of the exponential function, which is central in the definition of the constraints of a geometric optimization problem. This will allow us to present a family of l_p-norm optimization problems which can be made arbitrarily close to a primal geometric optimization problem. We derive in Section 8.3 a dual problem for this approximation, and show that the limiting case for these dual approximations is equivalent to the traditional dual geometric optimization problem. Using this family of pairs of primal-dual problems and the weak and strong duality theorems for l_p-norm optimization, we will then show how to derive the corresponding theorems for geometric optimization in a simple manner. Section 8.4 will conclude and present some topics for further research.

8.2 Approximating geometric optimization

In this section, we will show how geometric optimization problems can be approximated with l_p-norm optimization.

8.2.1 An approximation of the exponential function

A key ingredient in our approach is the function that will be used to approximate the exponential terms that arise within the constraints of (GP). Let $\alpha \in \mathbb{R}_{++}$ and let us define

$$g_\alpha : \mathbb{R}_+ \mapsto \mathbb{R}_+ : x \mapsto \left|1 - \frac{x}{\alpha}\right|^\alpha .$$

We have the following lemma relating $g_\alpha(x)$ to e^{-x}:

Lemma 8.1. *For any fixed $x \in \mathbb{R}_+$, we have that*

$$g_\alpha(x) \leq e^{-x}\ \forall \alpha \geq x \quad \textit{and} \quad e^{-x} < g_\alpha(x) + \alpha^{-1}\ \forall \alpha > 0\ , \tag{8.1}$$

where the first inequality is tight if and only if $x = 0$. Moreover, we have

$$\lim_{\alpha \to +\infty} g_\alpha(x) = e^{-x}\ .$$

Proof. Let us fix $x \in \mathbb{R}_+$. When $0 < \alpha < x$, we only have to prove the second inequality in (8.1), which is straightforward: we have $e^{-x} < e^{-\alpha} < \alpha^{-1} < g_\alpha(x) + \alpha^{-1}$, where we used the obvious inequalities $e^\alpha > \alpha$ and $g_\alpha(x) > 0$. Assuming $\alpha \geq x$ for the rest of this proof, we

define the auxiliary function $h : \mathbb{R}_{++} \mapsto \mathbb{R} : \alpha \mapsto \log g_\alpha(x)$. Using the Taylor expansion of $\log(1-x)$ around $x = 0$

$$\log(1-x) = -\sum_{i=1}^{\infty} \frac{x^i}{i} \quad \text{for all } x \text{ such that } |x| \leq 1 \tag{8.2}$$

we have

$$h(\alpha) = \alpha \log\Big|1 - \frac{x}{\alpha}\Big| = \alpha \log\Big(1 - \frac{x}{\alpha}\Big) = -\sum_{i=1}^{\infty} \frac{x^i}{i\alpha^{i-1}} = -x - \sum_{i=2}^{\infty} \frac{x^i}{i\alpha^{i-1}} \tag{8.3}$$

(where we used the fact that $\frac{x}{\alpha} \leq 1$ to write the Taylor expansion). It is now clear that $h(\alpha) \leq -x$, with equality if and only if $x = 0$, which in turn implies that $g_\alpha(x) \leq e^{-x}$, with equality if and only if $x = 0$, which is the first inequality in (8.1).

The second inequality is equivalent, after multiplication by e^x, to

$$1 < e^x g_\alpha(x) + e^x \alpha^{-1} \Leftrightarrow 1 - e^x\alpha^{-1} < e^x e^{h_\alpha(x)} \Leftrightarrow 1 - e^x\alpha^{-1} < e^{x+h_\alpha(x)} \ .$$

This last inequality trivially holds when its left-hand side is negative, i.e. when $\alpha \leq e^x$. When $\alpha > e^x$, we take the logarithm of both sides, use again the Taylor expansion (8.2) and the expression for $h_\alpha(x)$ in (8.3) to find

$$\log\big(1 - e^x\alpha^{-1}\big) < x + h_\alpha(x) \Leftrightarrow -\sum_{i=1}^{\infty} \frac{e^{xi}}{i\alpha^i} < -\sum_{i=2}^{\infty} \frac{x^i}{i\alpha^{i-1}} \Leftrightarrow 0 < \sum_{i=1}^{\infty} \frac{1}{\alpha^i}\left(\frac{e^{xi}}{i} - \frac{x^{i+1}}{i+1}\right) \ .$$

This last inequality holds since each of the coefficients between parentheses can be shown to be strictly positive: writing the well-known inequality $e^a > \frac{a^n}{n!}$ for $a = xi$ and $n = i+1$, we find

$$e^{xi} > \frac{(xi)^{i+1}}{(i+1)!} \Leftrightarrow \frac{e^{xi}}{i} > \frac{x^{i+1}}{(i+1)}\,\frac{i^i}{i!} \Rightarrow \frac{e^{xi}}{i} > \frac{x^{i+1}}{i+1} \Leftrightarrow \frac{e^{xi}}{i} - \frac{x^{i+1}}{i+1} > 0$$

(where we used $i^i \geq i!$ to derive the third inequality).

To conclude this proof, we note that (8.3) implies that $\lim_{\alpha\to+\infty} h(\alpha) = -x$, which gives $\lim_{\alpha\to+\infty} g_\alpha(x) = e^{-x}$, as announced. This last property can also be easily derived from the two inequalities in (8.1). □

The first inequality in (8.1) and the limit of $g_\alpha(x)$ are well-known, and are sometimes used as definition for the real exponential function, while the second inequality in (8.1) is much less common.

8.2.2 An approximation using l_p-norm optimization

The formulation of the primal geometric optimization problem (GP) relies heavily on the exponential function. Since Lemma 8.1 shows that it is possible to approximate e^{-x} with increasing accuracy using the function g_α, we can consider using this function to formulate an

approximation of problem (GP). The key observation we make here is that this approximation can be expressed as an l_p-norm optimization problem.

Indeed, let us fix $\alpha \in \mathbb{R}_{++}$ and write the approximate problem

$$\sup\ b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} \left(g_\alpha(c_i - a_i^T y) + \alpha^{-1}\right) \leq 1\ \forall k \in K\ . \tag{GP$_\alpha$}$$

We note that this problem is a restriction of the original problem (GP), i.e. that any y that is feasible for (GP$_\alpha$) is also feasible for (GP), with the same objective value. This is indeed a direct consequence of the second inequality in (8.1), which implies for any y feasible for (GP$_\alpha$)

$$\sum_{i \in I_k} e^{c_i - a_i^T y} < \sum_{i \in I_k} \left(g_\alpha(c_i - a_i^T y) + \alpha^{-1}\right) \leq 1\ .$$

We need now to transform the expressions $g_\alpha(c_i - a_i^T y) + \alpha^{-1}$ to fit the format of the constraints of an l_p-norm optimization problem. Assuming that $\alpha > 1$ for the rest of this chapter, we write

$$\begin{aligned}
\sum_{i \in I_k} \left(g_\alpha(c_i - a_i^T y) + \alpha^{-1}\right) \leq 1 \quad &\Leftrightarrow \quad \sum_{i \in I_k} g_\alpha(c_i - a_i^T y) \leq 1 - n_k \alpha^{-1} \\
&\Leftrightarrow \quad \sum_{i \in I_k} \left| 1 - \frac{c_i - a_i^T y}{\alpha} \right|^\alpha \leq 1 - n_k \alpha^{-1} \\
&\Leftrightarrow \quad \sum_{i \in I_k} \left| \alpha - c_i + a_i^T y \right|^\alpha \leq \alpha^\alpha (1 - n_k \alpha^{-1}) \\
&\Leftrightarrow \quad \sum_{i \in I_k} \frac{1}{\alpha} \left| c_i - \alpha - a_i^T y \right|^\alpha \leq \alpha^{\alpha - 1} (1 - n_k \alpha^{-1})
\end{aligned}$$

(where n_k is the number of elements in I_k), which allows us to write (GP$_\alpha$) as

$$\sup\ b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} \frac{1}{\alpha} \left| c_i - \alpha - a_i^T y \right|^\alpha \leq \alpha^{\alpha - 1} (1 - n_k \alpha^{-1})\ \forall k \in K\ . \tag{GP$'_\alpha$}$$

This is indeed an l_p-norm optimization in the form (Pl_p): dimensions m, n and r are the same in both problems, sets I, K and I_k are identical, the vector of exponents p satisfies $p_i = \alpha > 1$ for all $i \in I$, matrix A and vector b are the same for both problems while matrix F is equal to zero. The only difference consists in vectors $\tilde{c}$ and d, which satisfy $\tilde{c}_i = c_i - \alpha$ and $d_k = \alpha^{\alpha - 1}(1 - n_k \alpha^{-1})$.

We have thus shown how to approximate a geometric optimization problem with a standard l_p-norm optimization problem. Solving this problem for a fixed value of α will give a feasible solution to the original geometric optimization problem. Letting α tend to $+\infty$, the approximations $g_\alpha(c_i - a_i^T y)$ will be more and more accurate, and the corresponding feasible regions will approximate the feasible region of (GP) better and better. We can thus expect the optimal solutions of problems (GP$'_\alpha$) to tend to an optimal solution of (GP). Indeed, this is the most common situation, but it does not happen in all the cases, as will be showed in the next section.

8.3 Deriving duality properties

The purpose of this section is to study the duality properties of our geometric optimization problem and its approximations. Namely, using the duality properties of l_p-norm optimization problems, we will derive the corresponding properties for geometric optimization, using our family of approximate problems.

8.3.1 Duality for l_p-norm optimization

Defining a vector $q \in \mathbb{R}^n$ such that $\frac{1}{p_i} + \frac{1}{q_i} = 1$ for all $i \in I$, we recall from Chapter 4 that the dual problem for (Pl_p) consists in finding two vectors $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^r$ that maximize a highly nonlinear objective while satisfying some linear equalities and nonnegativity constraints:

$$\inf\ \psi(x,z) = c^T x + d^T z + \sum_{k\in K|z_k>0} z_k \sum_{i\in I_k} \frac{1}{q_i} \left|\frac{x_i}{z_k}\right|^{q_i} \text{ s.t. } \begin{cases} Ax + Fz = b \text{ and } z \geq 0\ , \\ z_k = 0 \Rightarrow x_i = 0\ \forall i \in I_k\ . \end{cases} \quad (\mathrm{D}l_p)$$

Let us recall here for convenience from Chapter 4 the following duality properties for the pair of problems (Pl_p)–(Dl_p):

Theorem 8.1 (Weak duality). *If y is feasible for* (Pl_p) *and (x,z) is feasible for* (Dl_p)*, we have $\psi(x,z) \geq b^T y$.*

Theorem 8.2 (Strong duality). *If both problems* (Pl_p) *and* (Dl_p) *are feasible, the primal optimal objective value is attained with a zero duality gap, i.e.*

$$\begin{aligned} p^* = \max b^T y \quad s.t. \quad & \sum_{i\in I_k} \frac{1}{p_i} \left|c_i - a_i^T y\right|^{p_i} \leq d_k - f_k^T y \quad \forall k \in K \\ = \inf\ \psi(x,z) \quad s.t. \quad & \begin{cases} Ax + Fz = b \text{ and } z \geq 0 \\ z_k = 0 \Rightarrow x_i = 0\ \forall i \in I_k \end{cases} \qquad = d^*\ . \end{aligned}$$

We would like to bring the reader's attention to an interesting special case of dual l_p-norm optimization problem. When matrix F is identically equal to 0, i.e. when there are no pure linear terms in the constraints, and when all exponents p_i corresponding to same set I_k are equal to each other, i.e. when we have $p_i = p^k\ \forall i \in I_k$ for all $k \in K$, problem (Dl_p) becomes

$$\inf\ \psi(x,z) = c^T x + d^T z + \sum_{k\in K|z_k>0} \frac{z_k^{1-q^k}}{q^k} \sum_{i\in I_k} |x_i|^{q^k} \text{ s.t. } \begin{cases} Ax = b \text{ and } z \geq 0\ , \\ z_k = 0 \Rightarrow x_i = 0\ \forall i \in I_k\ . \end{cases} \quad (\mathrm{D}l'_p)$$

This kind of formulation arises in problems of approximation in l_p-norm, see [NN94, Section 6.3.2] and [Ter85, Section 11, page 98].

Since variables z_k do not appear any more in the linear constraints but only in the objective function $\psi(x,z)$, we may try to find a closed form for their optimal value. Looking at one variable z_k at a time and isolating the corresponding terms in the objective, one finds

$d_k z_k + \frac{1}{q^k} z_k^{1-q^k} \sum_{i \in I_k} |x_i|^{q^k}$, whose derivative is equal to $d_k + \frac{1-q^k}{q^k} z_k^{-q^k} \sum_{i \in I_k} |x_i|^{q^k}$. One easily sees that this quantity admits a single maximum when

$$z_k = (p^k\, d_k)^{-\frac{1}{q^k}}\, \|x_{I_k}\|_{q^k}$$

(where $\|\cdot\|_p$ corresponds to the usual p-norm defined by $\|x\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$ and x_{I_k} denotes the vector made of the components of x whose indices belong to I_k), which always satisfies the nonnegativity constraint in ($\mathrm{D}l'_p$) and gives after some straightforward computations a value of

$$d_k z_k + \frac{1}{q^k} z_k^{1-q^k} \sum_{i \in I_k} |x_i|^{q^k} = \ldots = \big(1 + \frac{p^k}{q^k}\big) d_k z_k = p^k\, d_k z_k = (p^k\, d_k)^{\frac{1}{p^k}}\, \|x_{I_k}\|_{q^k}$$

for the two corresponding terms in the objective. Our dual problem ($\mathrm{D}l'_p$) becomes then

$$\inf\ \psi(x) = c^T x + \sum_{k \in K} (p^k\, d_k)^{\frac{1}{p^k}}\, \|x_{I_k}\|_{q^k} \quad \text{s.t.} \quad Ax = b\,, \tag{$\mathrm{D}l''_p$}$$

a great simplification when compared to ($\mathrm{D}l'_p$). One can check that the special treatment for the case $z_k = 0$ is well handled: indeed, $z_k = 0$ happens when $x_{I_k} = 0$, and the implication that is stated in the constraints of ($\mathrm{D}l'_p$) is thus satisfied.

It is interesting point out that problem($\mathrm{D}l''_p$) is essentially unconstrained, since it is well-known that linear equalities can be removed from an optimization problem that does not feature other types constraints (assuming matrix A has rank l, one can for example use these equalities to express l variables as linear combinations of the other variables and pivot these l variables out of the formulation). We also observe that in this case a primal problem with p-norms leads to a dual problem with q-norms, a situation which is examined by Dax and Sreedharan in [DS97].

8.3.2 A dual for the approximate problem

We are now going to write the dual for the approximate problem (GP'_α). Since we are in the case where $F = 0$ and all p_i's are equal to α, we can use the simplified version of the dual problem ($\mathrm{D}l''_p$) and write

$$\inf\ \psi_\alpha(x) = c^T x - \alpha e_n^T x + \sum_{k \in K} \big(\alpha\, \alpha^{\alpha-1}(1 - n_k \alpha^{-1})\big)^{\frac{1}{\alpha}}\, \|x_{I_k}\|_\beta \quad \text{s.t.} \quad Ax = b$$

(where e_n is a notation for the all-one n-dimensional column vector and $\beta > 1$ is a constant such that $\frac{1}{\alpha} + \frac{1}{\beta} = 1$), which can be simplified to give

$$\inf\ \psi_\alpha(x) = c^T x - \alpha e_n^T x + \alpha \sum_{k \in K} (1 - n_k \alpha^{-1})^{\frac{1}{\alpha}}\, \|x_{I_k}\|_\beta \quad \text{s.t.} \quad Ax = b\,. \tag{GD_α}$$

We observe that the constraints and thus the feasible region of this problem are independent from α, which only appears in the objective function $\psi_\alpha(x)$. Intuitively, since problems (GP'_α)

become closer and closer to (GP) as α tends to $+\infty$, the corresponding dual problems (GP'_α) should approximate the dual of (GP) better and better. It is thus interesting to write down the limiting case for these problems, i.e. find the limit of ψ_α when $\alpha \to +\infty$. Looking first at the terms that are related to single set of indices I_k, we write

$$\begin{aligned}
\psi_{k,\alpha}(x) &= c_{I_k}^T x_{I_k} - \alpha e_{n_k}^T x_{I_k} + \alpha(1 - n_k\alpha^{-1})^{\frac{1}{\alpha}} \left\|x_{I_k}\right\|_\beta \\
&= c_{I_k}^T x_{I_k} - \alpha e_{n_k}^T x_{I_k} + \alpha \left\|x_{I_k}\right\|_1 - \alpha \left\|x_{I_k}\right\|_1 + \alpha(1 - n_k\alpha^{-1})^{\frac{1}{\alpha}} \left\|x_{I_k}\right\|_\beta \\
&= c_{I_k}^T x_{I_k} + \alpha \left[\left\|x_{I_k}\right\|_1 - e_{n_k}^T x_{I_k}\right] + \alpha \left[(1 - n_k\alpha^{-1})^{\frac{1}{\alpha}} \left\|x_{I_k}\right\|_\beta - \left\|x_{I_k}\right\|_1\right] \\
&= c_{I_k}^T x_{I_k} + \alpha \left[\left\|x_{I_k}\right\|_1 - e_{n_k}^T x_{I_k}\right] + \frac{\beta}{\beta - 1} \left[(1 - n_k\alpha^{-1})^{\frac{1}{\alpha}} \left\|x_{I_k}\right\|_\beta - \left\|x_{I_k}\right\|_1\right]
\end{aligned}$$

(where we used at the last line the fact that $\alpha = \frac{\beta}{\beta-1}$). When α tends to $+\infty$ (and thus $\beta \to 1$), we have that the limit of $\psi_{k,\alpha}(x)$ is equal to

$$\begin{aligned}
& c_{I_k}^T x_{I_k} + \lim_{\alpha\to+\infty} \alpha \left[\left\|x_{I_k}\right\|_1 - e_{n_k}^T x_{I_k}\right] + \lim_{\substack{\alpha\to+\infty \\ \beta\to1}} \frac{\beta}{\beta - 1} \left[(1 - n_k\alpha^{-1})^{\frac{1}{\alpha}} \left\|x_{I_k}\right\|_\beta - \left\|x_{I_k}\right\|_1\right] \\
= \; & c_{I_k}^T x_{I_k} + \lim_{\alpha\to+\infty} \alpha \left[\left\|x_{I_k}\right\|_1 - e_{n_k}^T x_{I_k}\right] + \lim_{\beta\to1} \frac{\left\|x_{I_k}\right\|_\beta - \left\|x_{I_k}\right\|_1}{\beta - 1}
\end{aligned}$$

The last term in this limit is equal to the derivative of the real function $m_k : \beta \mapsto \left\|x_{I_k}\right\|_\beta$ at the point $\beta = 1$. We can check with some straightforward but lengthy computations that

$$m_k'(\beta) = \frac{\left\|x_{I_k}\right\|_\beta^{\frac{1}{\beta}-1}}{\beta^2} \left[\beta \sum_{i\in I_k | x_i>0} |x_i|^\beta \log|x_i| - \left\|x_{I_k}\right\|_1 \log\left\|x_{I_k}\right\|_1\right] ,$$

which gives for $\beta = 1$

$$m_k'(1) = \sum_{i\in I_k | x_i>0} |x_i| \log \frac{|x_i|}{\left\|x_{I_k}\right\|_1} ,$$

and leads to

$$\lim_{\substack{\alpha\to+\infty \\ \beta\to1}} \psi_{k,\alpha}(x) = c_{I_k}^T x_{I_k} + \lim_{\alpha\to+\infty} \alpha \left[\left\|x_{I_k}\right\|_1 - e_{n_k}^T x_{I_k}\right] + \sum_{i\in I_k | x_i>0} |x_i| \log \frac{|x_i|}{\left\|x_{I_k}\right\|_1} .$$

It is easy to see that $\left\|x_{I_k}\right\|_1 - e_{n_k}^T x_{I_k} \geq 0$, with equality if and only if $x_{I_k} \geq 0$. This means that the limit of our objective $\psi_{k,\alpha}(x)$ will be $+\infty$ unless $x_{I_k} \geq 0$. An objective equal to $+\infty$ for a minimization problem can be assimilated to an unfeasible problem, which means that the limit of our dual approximations (GD_α) admits the hidden constraint $x_{I_k} \geq 0$. Gathering now all terms in the objective, we eventually find the limit of problems (GD_α) when $\alpha \to +\infty$ to be

$$\inf \phi(x) = c^T x + \sum_{k\in K} \sum_{i\in I_k | x_i>0} x_i \log \frac{x_i}{\sum_{i\in I_k} x_i} \quad \text{s.t.} \quad Ax = b \text{ and } x \geq 0 , \qquad (\mathrm{GD})$$

which is exactly the dual geometric optimization problem that was presented in Chapter 5.

8.3.3 Duality for geometric optimization

Before we start to prove duality results for geometric optimization, we make a technical assumption on problem (GP), whose purpose will become clear further in this section: we assume that $n_k \geq 2$ for all $k \in K$, i.e. forbid problems where a constraint is defined with a single exponential term. This can be done without any loss of generality, since a constraint of the form $e^{a_i^T y - c_i} \leq 1$ can be equivalently rewritten as $e^{a_i^T y - c_i - \log 2} + e^{a_i^T y - c_i - \log 2} \leq 1$.

Let us now prove the weak duality Theorem 5.9 for geometric optimization:

Theorem 8.3 (Weak duality). *If y is feasible for* (GP) *and x is feasible for* (GD)*, we have $\phi(x) \geq b^T y$.*

Proof. Our objective is to prove this theorem using our family of primal-dual approximate problems (GP'_α)–(GD_α). We first note that x is feasible for (GD_α) for every α, since the only constraints for this family of problems are the linear constraints $Ax = b$, which are also present in (GD). The situation is a little different on the primal side: the first inequality in (8.1) and feasibility of y for (GP) imply

$$\sum_{i \in I_k} g_\alpha(c_i - a_i^T y) \leq \sum_{i \in I_k} e^{a_i^T y - c_i} \leq 1 \ ,$$

with equality if and only if $c_i - a_i^T y = 0$ for all $i \in I_k$. But this cannot happen, since we would have $\sum_{i \in I_k} e^{a_i^T y - c_i} = \sum_{i \in I_k} 1 = n_k > 1$, because of our assumption on n_k, which contradicts the feasibility of y. We can conclude that the following strict inequality holds for all $k \in K$:

$$\sum_{i \in I_k} g_\alpha(c_i - a_i^T y) < 1 \ .$$

Since the set K is finite, this means that there exists a constant M such that for all $\alpha \geq M$,

$$\sum_{i \in I_k} g_\alpha(c_i - a_i^T y) \leq 1 - n_k \alpha^{-1} \ \forall k \in K \ ,$$

which in turn implies feasibility of y for problems (GP'_α) as soon as $\alpha \geq M$. Feasibility of both y and x for their respective problem allows us to apply the weak duality Theorem 8.1 of l_p-norm optimization to our pair of approximate problems (GP'_α)–(GD_α), which implies $\psi_\alpha(x) \geq b^T y$ for all $\alpha \geq M$. Taking now the limit of $\psi_\alpha(x)$ for α tending to $+\infty$, which is finite and equal to $\phi(x)$ since $x \geq 0$, we find that $\phi(x) \geq b^T y$, which is the announced inequality. □

The strong duality Theorem 5.13 for geometric optimization is stated below. We note that contrary to the class of l_p-norm optimization problems, attainment cannot be guaranteed for any of the primal and dual optimum objective values.

Theorem 8.4. *If both problems* (GP) *and* (GD) *are feasible, their optimum objective values p^* and d^* are equal.*

Proof. As shown in the proof of the previous theorem, the existence of a feasible solution for (GP) and (GD) implies that problems (GP'_α) and (GD_α) are both feasible for all α greater than some constant M. Denoting by p^*_α (resp. d^*_α) the optimal objective value of problem (GP'_α) (resp. (GD_α)), we can thus apply the strong duality Theorem 8.2 of l_p-norm optimization to these pairs of problems to find that $p^*_\alpha = d^*_\alpha$ for all $\alpha \geq M$. Since all the dual approximate problems $p^*_\alpha = d^*_\alpha$ share the same feasible region, it is clear that the optimal value corresponding to the limit of the objective ψ_α when $\alpha \to +\infty$ is equal to the limit of the optimal objective values d^*_α for $\alpha \to +\infty$. Since the problem featuring this limiting objective has been shown to be equivalent to (GD) in Section 8.3.2 (including the hidden constraint $x \geq 0$), we must have $d^* = \lim_{\alpha \to +\infty} d^*_\alpha$. On the other hand, Theorem 8.2 guarantees for each of the problems (GP'_α) the existence of an optimal solution y_α that satisfies $b^T y_\alpha = p^*_\alpha$. Since each of these solutions is also a feasible solution for (GP) (since problems (GP'_α) are restrictions of (GP)), which shares the same objective function, we have that the optimal objective value of (GP) p^* is at least equal to $b^T y_\alpha$ for all $\alpha \geq M$, which implies $p^* \geq \lim_{\alpha \to +\infty} b^T y_\alpha = \lim_{\alpha \to +\infty} p^*_\alpha = \lim_{\alpha \to +\infty} d^*_\alpha = d^*$. Combining this last inequality with the easy consequence of the weak duality Theorem 8.3 that states $d^* \geq p^*$, we end up with the announced equality $p^* = d^*$. □

The reason why attainment of the primal optimum objective value cannot be guaranteed is that the sequence y_α may not have a finite limit point, a justification that is very similar to the one that was given in the concluding remarks of Chapter 5.

8.4 Concluding remarks

In this chapter, we have shown that the important class of geometric optimization problems can be approximated with l_p-norm optimization.

We have indeed described a parameterized family of primal and dual l_p-norm optimization problems, which can be made arbitrarily close to the geometric primal and dual problems. It is worth to note that the primal approximations are restrictions of the original geometric primal problem, sharing the same objective function, while the dual approximations share essentially the same constraints as the original geometric dual problem (except for the nonnegativity constraints) but feature a different objective.

Another possible approach would be to work with relaxations instead of restrictions on the primal side, using the first inequality in (8.1) instead of the second one, leading to the following problem:

$$\sup \; b^T y \quad \text{s.t.} \quad \sum_{i \in I_k} g_\alpha(c_i - a_i^T y) \leq 1 \; \forall k \in K \; .$$

However, two problems arise in this setting:

- ◇ the first inequality in (8.1) is only valid when $\alpha \geq x$, which means we would have to add a set of explicit linear inequalities $c_i - a_i^T y \leq \alpha$ to our approximations, which would make them and their dual problems more difficult to handle,

- ◇ following the same line of reasoning as in the proof of Theorem 8.2, we would end up with another family of optimal solutions y_α for the approximate problems; however, since all of these problems are relaxations, we would have no guarantee that any of the optimal vectors y_α are feasible for the original primal geometric optimization problem, which would prevent us to conclude that the duality gap is equal to zero. This would only show that there is a family of asymptotically feasible primal solutions with their objective values tending to the objective value of the dual, a fact that is always true in convex optimization (this is indeed the essence of the alternate strong duality Theorem 3.6, related to the notion of subvalue, see Chapter 3).

To conclude, we note that our approximate problems belong to a very special subcategory of l_p-norm optimization problem, since they satisfy $F = 0$. It might be fruitful to investigate which class of generalized geometric optimization problems can be approximated with general l_p-norm optimization problems, a topic we leave for further research.

CHAPTER 9

Computational experiments with a linear approximation of second-order cone optimization

In this chapter, we present and improve a polyhedral approximation of the second-order cone due to Ben-Tal and Nemirovski [BTN98]. We also discuss several ways of reducing the size of this approximation. This construction allows us to approximate second-order cone optimization problems with linear optimization.

We implement this scheme and conduct computational experiments dealing with two classes of second-order cone problems: the first one involves truss-topology design and uses a large number of second-order cones with relatively small dimensions, while the second one models convex quadratic optimization problems with a single large second-order cone.

9.1 Introduction

Chapter 3 deals with conic optimization, which is a powerful setting that relies on convex cones to formulate convex problems. We recall here the standard conic primal-dual pair for

convenience

$$\inf_{x\in\mathbb{R}^n} \; c^T x \quad \text{s.t.} \quad Ax = b \text{ and } x \in \mathcal{C} \tag{CP}$$

$$\sup_{y\in\mathbb{R}^m, x^*\in\mathbb{R}^n} \; b^T y \quad \text{s.t.} \quad A^T y + x^* = c \text{ and } x^* \in \mathcal{C}^* \;, \tag{CD}$$

where x and (y, x^*) are the primal and dual variables, A is a $m \times n$ matrix, b and c are m and n-dimensional column vectors, $\mathcal{C} \subseteq \mathbb{R}^n$ is a closed pointed solid convex cone and $\mathcal{C}^* \subseteq \mathbb{R}^n$ is its dual cone, defined by $\mathcal{C}^* = \{x^* \in \mathbb{R}^n \mid x^T x^* \geq 0 \; \forall x \in \mathcal{C}\}$.

Different types of convex cones lead to different classes of problems: for example, linear optimization uses the nonnegative orthant $\mathbb{R}^n_+$ while semidefinite optimization relies on the set of positive semidefinite matrices $\mathbb{S}^n_+$ (see Chapter 3). In this chapter, we will focus the *second-order* cone, also known as *Lorentz* cone or *ice-cream cone*, which leads to *second-order cone optimization*. It is defined as follows:

Definition 9.1. The second order cone $\mathbb{L}^n$ is the subset of $\mathbb{R}^{n+1}$ defined by

$$\mathbb{L}^n = \{(r, x) \in \mathbb{R} \times \mathbb{R}^n \mid \|x\| \leq r\} \;,$$

where $\|\cdot\|$ denotes the usual Euclidean norm on $\mathbb{R}^n$.

It is indeed straightforward to check that this is a closed pointed solid convex cone (it is in fact the epigraph of the Euclidean norm). Another interesting property of $\mathbb{L}^n$ is the fact that it is self-dual, i.e. $(\mathbb{L}^n)^* = \mathbb{L}^n$.

The standard second-order cone problems are based on the cartesian product of several second-order cones, which can be formalized using r constants $n_k \in \mathbb{N}, 1 \leq k \leq r$ such that $\sum_{k=1}^r (n_k+1) = n$ and defining $\mathcal{C} = \mathbb{L}^{n_1} \times \mathbb{L}^{n_2} \times \cdots \times \mathbb{L}^{n_r}$. This set is obviously also a self-dual closed convex cone, which allows us to rewrite problems (CP) and (CD) as

$$\inf \; c^T x \qquad \text{s.t.} \qquad Ax = b \text{ and } x^k \in \mathbb{L}^{n_k} \; \forall k = 1, 2, \ldots, r \tag{9.1}$$

$$\sup \; b^T y \qquad \text{s.t.} \qquad A^T y + x^* = c \text{ and } x^{k*} \in \mathbb{L}^{n_k} \; \forall k = 1, 2, \ldots, r \;, \tag{9.2}$$

where vectors x and x^* have been split into r subvectors $(x^1, x^2, \ldots, x^r)$ and $(x^{1*}, x^{2*}, \ldots, x^{r*})$ with $x^k \in \mathbb{R}^{n_k+1}$ and $x^{k*} \in \mathbb{R}^{n_k+1}$ for all $k = 1, \ldots, r$. It is usually more practical to pivot out variables x^* in the dual problem (9.2), i.e. write them as a function of vector y. Splitting matrix A into $(A^1, A^2, \ldots, A^r)$ with $A^k \in \mathbb{R}^{m\times(n_k+1)}$ and vector c into $(c^1, c^2, \ldots, c^r)$ with $c^k \in \mathbb{R}^{n_k+1}$, we have $x^{k*} = c^k - A^{kT} y$. The last step is to isolate the first column in A^k and the first component in c^k, i.e. letting $A^k = (f^k, G^k)$ with $f^k \in \mathbb{R}^m$ and $G^k \in \mathbb{R}^{m\times n_k}$ and $c^k = (d^k, h^k)$ with $d^k \in R$ and $h^k \in \mathbb{R}^{n_k}$, we can rewrite the dual problem (9.2) as

$$\sup b^T y \quad \text{s.t.} \quad \left\|G^{kT} y + h^k\right\| \leq f^{kT} y + d^k \; \forall k = 1, 2, \ldots, r \;,$$

which is more convenient to formulate real-world problems (we also note that these constraints bear a certain similarity to l_p-norm optimization constraints, see Chapter 4).

Second-order optimization admits many different well-known classes of optimization problems as special cases, such as linear optimization, linearly and quadratically constrained

convex quadratic optimization, robust linear optimization, matrix-fractional problems and problems with hyperbolic constraints (see the survey [LVBL98]). Applications arise in various fields such as engineering (antenna array design, finite impulse response filter design, truss design) and finance (portfolio optimization), see again [LVBL98].

From the computational point of view, second-order cone optimization is a relatively young field if compared to linear and quadratic optimization (for example, the leading commercial linear and quadratic solvers do not yet offer the option of solving second-order cone optimization problems). This observation led Ben-Tal and Nemirovski to develop an interesting alternative approach to solving second-order cone problems: they show in [BTN98] that it is possible to write a polyhedral approximation of the second order cone $\mathbb{L}^n$ with a prescribed accuracy ϵ using a number of variables and constraints that is polynomial in n and $\log \frac{1}{\epsilon}$. This implies that second-order cone optimization problems can be approximated with an arbitrarily prescribed accuracy by linear optimization problems using this polyhedral approximation.

This potentially allows the approximate resolution of large-scale second-order cone problems using state of the art linear solvers, capable of handling problems with hundreds of thousands of variables and constraints.

This chapter is organized as follows: Section 9.2 presents a polyhedral approximation of the second-order cone. This construction relies on a decomposition scheme based on three-dimensional second order cones. We present first an efficient way to approximate these cones and then show how to combine them in order to approximate a second-order cone of higher dimension, which ultimately gives a method to approximate any second-order cone optimization problem with a linear problem. Section 9.3 reports our computational experiments with this scheme. After a presentation of our implementation and some related issues, we describe two classes of second-order problems: truss-topology design problems and convex quadratic optimization problems. We present and discuss the results of our computational experiments, highlighting when necessary the particular features of each class of problems (guaranteed accuracy, alternative formulations). We conclude this chapter with a few remarks and suggestions for further research.

9.2 Approximating second-order cone optimization

In this section, we present a polyhedral approximation of the second-order cone $\mathbb{L}^n$ which allows us to derive a linearizing scheme for second-order cone optimization. It is a variation of the construction of Ben-Tal and Nemirovski that features slightly better properties.

9.2.1 Principle

The principle that lies behind their approximation is twofold:

a. **Decomposition**. Since the Lorentz cone $\mathbb{L}^n$ is a $n+1$-dimensional subset, any circumscribed polyhedral cone around $\mathbb{L}^n$ is bound to have its number of facets growing

exponentially with the dimension n, i.e. will need an exponential number of linear inequalities to be defined. The remedy is to decompose the second-order cone into a polynomial number of smaller second-order cones with fixed dimension, for which a good polyhedral approximation can be found. In the present case, $\mathbb{L}^n$ can be decomposed into $n-1$ three-dimensional second-order cones $\mathbb{L}^2$, at the price of introducing $n-2$ additional variables (see Section 9.2.2).

b. **Projection**. Even the three-dimensional second-order cone $\mathbb{L}^2$ is not too easy to approximate: the most obvious way to proceed, a regular circumscribed polyhedral cone, requires hundreds of inequalities even for an approximation with modest accuracy (see Section 9.2.3). The key idea to lower the number of inequalities is to introduce several additional variables, i.e. lift the approximating polyhedron into a higher dimensional space and consider its projection onto a $(n+1)$-dimensional subspace as the approximation of $\mathbb{L}^n$ (see Section 9.2.4).

To summarize, the introduction of a certain number of additional variables, combined with a projection, can be traded against a much lower number of inequality constraints defining the polyhedron. We first concentrate on the decomposition of $\mathbb{L}^n$ into smaller second-order cones.

9.2.2 Decomposition

Let us start with the following equivalent definition of $\mathbb{L}^n$

$$\mathbb{L}^n = \left\{(r, x_1, x_2, \ldots, x_n) \in \mathbb{R}_+ \times \mathbb{R}^n \mid \sum_{i=1}^{n} x_i^2 \leq r^2\right\}.$$

Introducing a vector of $\lfloor \frac{n}{2} \rfloor$ additional variables $y = (y_1, y_2, \ldots, y_{\lfloor \frac{n}{2} \rfloor})$, we consider the set $\mathbb{L}^{n\prime}$ defined by

$$\left\{(r, x, y) \in \mathbb{R}_+ \times \mathbb{R}^{n+\lfloor \frac{n}{2} \rfloor} \mid x_{2i-1}^2 + x_{2i}^2 \leq y_i^2,\ 1 \leq i \leq \lfloor \frac{n}{2} \rfloor,\ \begin{cases} \sum_{i=1}^{\frac{n}{2}} y_i^2 & \leq r^2 \ (n \text{ even}) \\ \sum_{i=1}^{\frac{n}{2}} y_i^2 + x_n^2 & \leq r^2 \ (n \text{ odd}) \end{cases}\right\}.$$

It it straightforward to prove that the projection of this set on the subspace of its first $n+1$ variables $(r, x_1, \ldots, x_n)$ is equal to $\mathbb{L}^n$, i.e.

$$(r, x) \in \mathbb{L}^n \quad \Leftrightarrow \quad \exists y \in \mathbb{R}^{\lfloor \frac{n}{2} \rfloor} \quad \text{s.t.} \quad (r, x, y) \in \mathbb{L}^{n\prime}.$$

It is also worth to point out that all the constraints defining $\mathbb{L}^{n\prime}$ are second-order cone constraints, i.e. that $\mathbb{L}^{n\prime}$ can also be written as

$$\left\{(r, x, y) \in \mathbb{R}_+ \times \mathbb{R}^{n+\lfloor \frac{n}{2} \rfloor} \mid (y_i, x_{2i-1}, x_{2i}) \in \mathbb{L}^2,\ 1 \leq i \leq \lfloor \frac{n}{2} \rfloor,\ \begin{cases} (r, y) & \in \mathbb{L}^{\lceil \frac{n}{2} \rceil} \ (n \text{ even}) \\ (r, y, x_n) & \in \mathbb{L}^{\lceil \frac{n}{2} \rceil} \ (n \text{ odd}) \end{cases}\right\}. \tag{9.3}$$

This means that $\mathbb{L}^n$ can be decomposed into $\lfloor \frac{n}{2} \rfloor$ 3-dimensional second-order cones and a single $\mathbb{L}^{\lceil \frac{n}{2} \rceil}$ second-order cone, at the price of introducing $\lfloor \frac{n}{2} \rfloor$ auxiliary variables. This procedure

can be applied recursively to the largest of the remaining second-order cone $\mathbb{L}^{\lceil \frac{n}{2} \rceil}$ until it also becomes equal to $\mathbb{L}^2$.

It is not too difficult to see that there are in the final expression $n-1$ second-order cones $\mathbb{L}^2$ and $n-2$ additional y_i variables. Indeed, the addition of each small cone $\mathbb{L}^2$ reduces the size of the largest cone by one, since we remove two variables from this cone (the last two variables in $\mathbb{L}^2$) but replace them with a single new variable (the first variable in $\mathbb{L}^2$).

Since we start with this largest cone equal $\mathbb{L}^n$ and stop when its size is equal to 2, we need $n-2$ small cones along with $n-2$ auxiliary variables to reduce the cone to $\mathbb{L}^2$. But this last $\mathbb{L}^2$ cone also has to be counted, which gives then a total number of cones equal to $n-1$.

The existence of this decomposition implies that any second-order cone optimization problem can be transformed into a problem using only 3-dimensional second-order cones, using the construction above. We note however that strictly speaking, the resulting formulation is not a conic problem, since some variables belong to two different cones at the same time. It is nonetheless possible to add a extra variable for each shared variable, along with a constraint to make them equal on the feasible region, to convert this formulation into the strict conic format (CP)–(CD).

9.2.3 A first approximation of $\mathbb{L}^2$

The previous section has shown that we can focus our attention on approximations of the 3-dimensional second-order. Moreover, it seems reasonable to require this approximation to be a cone itself too. Taking into account this additional assumption, we can take advantage of the homogeneity property of these cones to write

$$(r, x_1, x_2) \in \mathbb{L}^2 \Leftrightarrow (1, \frac{x_1}{r}, \frac{x_2}{r}) \in \mathbb{L}^2 \ , \tag{9.4}$$

which basically means we can fix $r = 1$ and look for a polyhedral approximation of the resulting set

$$\left\{x \in \mathbb{R}^2 \mid (1, x) \in \mathbb{L}^2\right\} = \left\{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 1\right\} = \mathcal{B}_2(1) \ ,$$

which is exactly the disc of radius one in $\mathbb{R}^2$. Any approximating polyhedron for $\mathcal{B}_2(2)$ will be then later straightforwardly converted into a polyhedral cone approximating $\mathbb{L}^2$, using the additional homogenizing variable r.

At this point, we have to introduce a measure of the quality of our approximations. A natural choice for this measure is to state that a polyhedron $\mathcal{P} \subseteq \mathbb{R}^2$ is a ϵ-approximation of $\mathcal{B}_2(1)$ if and only we have the double inclusion $\mathcal{B}_2(1) \subseteq \mathcal{P} \subseteq \mathcal{B}_2(1+\epsilon)$, i.e. the polyhedron contains the unit disc but lies entirely within the disc of radius $1+\epsilon$.

The most obvious approximation of the unit disc is the regular m-polyhedron $\mathcal{P}_m$, which is described by m linear inequalities. We have the following theorem:

Theorem 9.1. *The regular polyhedron with m sides is an approximation of the unit disc $\mathcal{B}_2(1)$ with accuracy $\epsilon = \cos(\frac{\pi}{m})^{-1} - 1$.*

Proof. The proof is quite straightforward: looking at Figure 9.1 (which represents the case $m = 8$), we see that angle $\angle AOM$ is equal to $\frac{\pi}{m}$ and thus that $|OA|\cos(\frac{\pi}{m}) = |OM| = 1$. Our measure of quality is then equal to $\epsilon = |OA| - 1 = \cos(\frac{\pi}{m})^{-1} - 1$, as announced. □

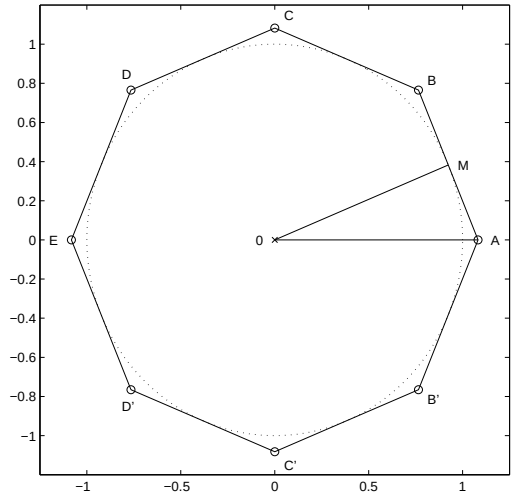


Figure 9.1: Approximating $\mathcal{B}_2(1)$ with a regular octagon.

This result is not very satisfying: since $\cos(x)^{-1} \approx 1 + \frac{x^2}{2}$ when x is small, we have that $\epsilon \approx \frac{\pi^2}{2m^2}$ when m is large, which means that doubling the number of inequalities only divides the accuracy by four. For example, approximating $\mathcal{B}_2(1)$ with the relatively modest accuracy 10^{-4} would already take a 223-sided polyhedron, i.e. more than 200 linear inequalities.

9.2.4 A better approximation of $\mathbb{L}^2$

As outlined in Section 9.2.1, the key idea introduced by Ben-Tal and Nemirovski to obtain a better polyhedral approximation is to consider the projection of a polyhedron belonging to a higher dimensional space. The construction we are going to present here is a variation of the one described in [BTN98], featuring slightly better parameters and a more transparent proof.

Let us introduce an integer parameter $k \geq 2$ and consider the set $\mathcal{D}_k \subseteq \mathbb{R}^{2k+2}$ defined as

$$\left\{(\alpha_0,\ldots,\alpha_k,\beta_0,\ldots,\beta_k) \in \mathbb{R}^{2k+2} \mid \left\{\begin{array}{rcl} \alpha_{i+1} &=& \alpha_i \cos\frac{\pi}{2^i} + \beta_i \sin\frac{\pi}{2^i} \\ \beta_{i+1} &\geq& \beta_i \cos\frac{\pi}{2^i} - \alpha_i \sin\frac{\pi}{2^i} \\ -\beta_{i+1} &\leq& \beta_i \cos\frac{\pi}{2^i} - \alpha_i \sin\frac{\pi}{2^i} \\ 1 &=& \alpha_k \cos\frac{\pi}{2^k} + \beta_k \sin\frac{\pi}{2^k} \end{array} \quad \forall\, 0 \leq i < k \right\}.$$

This set is obviously a polyhedron[1], since its defining constraints consist in $k+1$ linear equalities and $2k$ inequalities. The following theorem gives some insight about the structure of this set.

[1]Strictly speaking, this set is not a full-dimensional polyhedron in $\mathbb{R}^{2k+2}$ because of the additional linear constraints but this has no incidence on our purpose.

Theorem 9.2. *The projection of the set $\mathcal{D}_k$ on the subspace of its two variables (α_0, β_0) is equal to the regular 2^k-sided polyhedron, i.e. we have*

$$(\alpha_0, \beta_0) \in \mathcal{P}_{2^k} \;\Leftrightarrow\; \exists(\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k) \in \mathbb{R}^{2k} \mid (\alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_k) \in \mathcal{D}_k \,.$$

Proof. To fix ideas, we are going to present some figures corresponding to the case $k = 3$, but our reasoning will of course be valid for all $k \geq 2$. Looking at Figure 9.1, which depicts $\mathcal{P}_{2^3}$, we see that the last equality in the definition of $\mathcal{D}_k$ describes the line AM. Indeed, we have $A = (\cos(\frac{\pi}{2^k})^{-1}, 0)$ and $M = (\cos(\frac{\pi}{2^k}), \sin(\frac{\pi}{2^k}))$ and it is straightforward that both of these points satisfy the last equality in the definition of $\mathcal{D}_k$.

Recall now that the application

$$\mathcal{R}_\theta : \mathbb{R}^2 \mapsto \mathbb{R}^2 : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \mathcal{R}_\theta \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x\cos\theta + y\sin\theta \\ -x\sin\theta + y\cos\theta \end{pmatrix}$$

is a clockwise rotation around the origin with angle θ. Calling P_i the point of $\mathbb{R}^2$ whose coordinates are (α_i, β_i) and $\hat{P}_i = (\hat{\alpha}_i, \hat{\beta}_i)$ the image of P_i by rotation $\mathcal{R}_{\pi/2^i}$, we have that the first three constraints in the definition of $\mathcal{D}_k$ are equivalent to $\alpha_{i+1} = \hat{\alpha}_i$, $\beta_{i+1} \geq \hat{\beta}_i$ and $-\beta_{i+1} \leq \hat{\beta}_i$. These last two inequalities rewritten as $-\beta_{i+1} \leq \hat{\beta}_i \leq \beta_{i+1}$ immediately imply that β_{i+1} has to be nonnegative. Under this assumption, we call $\bar{P}_i$ the points whose coordinates are $(\alpha_i, -\beta_i)$ and find that these three constraints are equivalent to saying that $\hat{P}_i \in [P_{i+1}\ \bar{P}_{i+1}]$. In other words, the point $\hat{P}_i$ has to belong to a vertical segment $[P_{i+1}\ \bar{P}_{i+1}]$ such that P_{i+1} has its second coordinate nonnegative. Since $\hat{P}_i$ is the image of P_i by a rotation of angle $\pi/2^i$, saying that $\hat{P}_i$ belongs to some set is equivalent to saying that P_i belongs to the image of this set by the inverse rotation. In our case, this means *in fine* that P_i has to belong to the image by a rotation of angle $-\pi/2^i$ of a segment $[P_{i+1}\ \bar{P}_{i+1}]$ such that P_{i+1} has its second coordinate nonnegative.

We can now specialize this result to $i = k - 1$. Recall that P_k is known to belong to the line AB. According to the above discussion, we have first to restrict this set to its points with a nonnegative β_k, which gives the half line $[AB$. Taking the union of all segments $[P_k\ \bar{P}_k]$ for all possible P_k's gives the region bounded by half lines $[AB$ and $[AB'$. Taking finally the image of this set by a rotation of angle $-\pi/2^{k-1}$, we find that P_{k-1} has to belong to the region bounded by half lines $[BA$ and $[BC$.

We can now iterate this procedure and describe the set of points P_{k-2}, P_{k-3}, etc. Indeed, using exactly the same reasoning, we find that the set of points P_{i-1} can be deduced from the set of points P_i with a three-step procedure:

a. Restrict the set of points P_i to those with a nonnegative β_i coordinate.

b. Consider the union of segments $[P_i\ \bar{P}_i]$ where P_i belongs to the above restricted set, i.e. add for each point (α_i, β_i) the set of points (α_i, x) for all x ranging from $-\beta_i$ to β_i.

c. Rotate this union counterclockwise around the origin with an angle equal to $\pi/2^i$ to find the set of points P_{i-1}.

In the case of our example with $k = 3$, we have already shown that the set $\{P_k\} = \{P_3\} = [AB$ and $\{P_{k-1}\} = \{P_2\}$ is the region bounded by $[BA \cup [BC$. Going on with the procedure described above, we readily find that $\{P_{k-2}\} = \{P_1\}$ is the region bounded by the polygonal line $[ABCDE]$ while $\{P_{k-3}\} = \{P_0\}$ is the complete octagon $ABCDED'C'B'A$, which is the expected result (see Figure 9.2 for the corresponding pictures). It is not difficult to see that in the general case $\{P_{k-i}\}$ is a set bounded by 2^i consecutive sides of $\mathcal{P}_{2^k}$, which means we always end up with $\{P_0\}$ equal to the whole regular 2^k-sided polyhedron $\mathcal{P}_{2^k}$. This completes the proof since the set of points P_0 is the projection of $\mathcal{D}_k$ on the subspace of the two variables (α_0, β_0). □

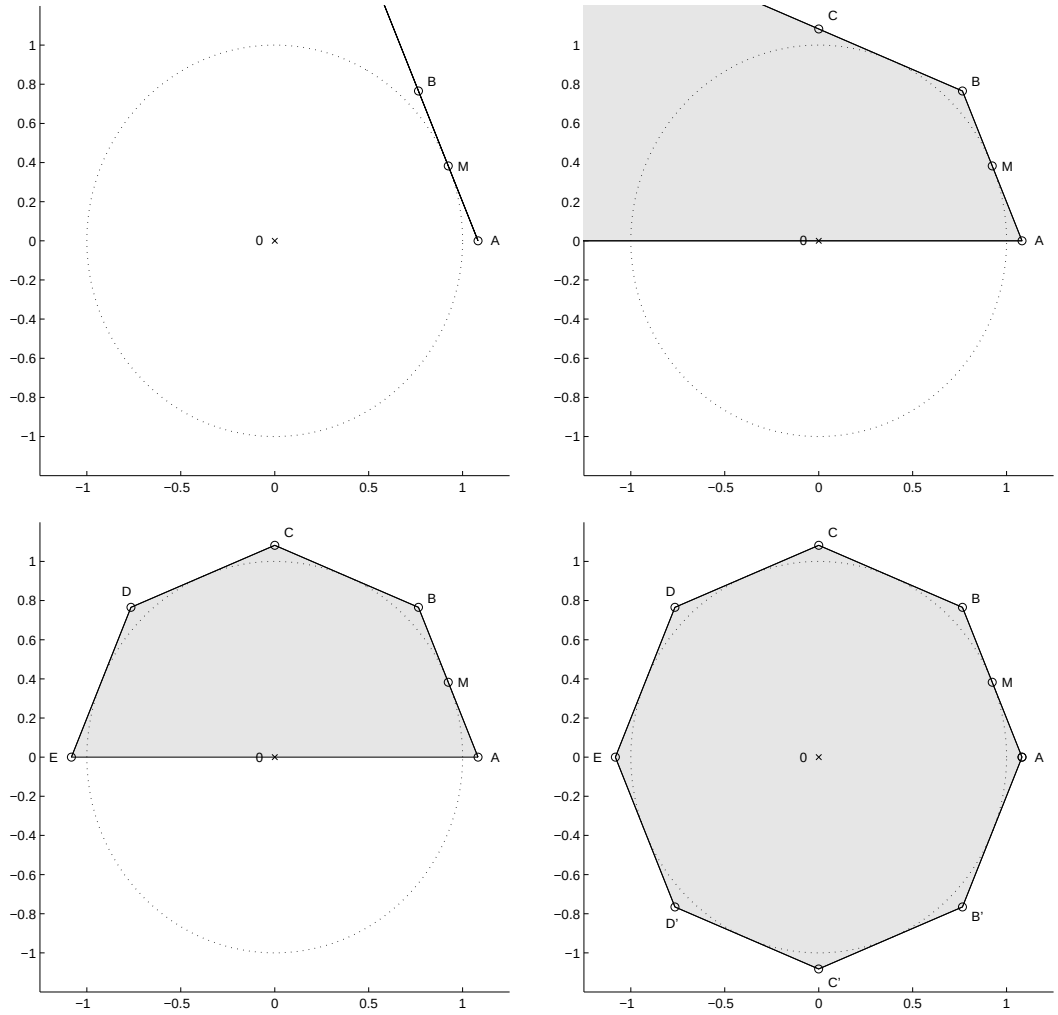


Figure 9.2: The sets of points P_3, P_2, P_1 and P_0 when $k = 3$.

This theorem allows us to derive quite easily a polyhedral approximation for $\mathcal{B}_2(1)$.

Corollary 9.1. *The projection of $\mathcal{D}_k$ on the subspace of its two variables (α_0, β_0) is a polyhedral approximation of $\mathcal{B}_2(1)$ with accuracy $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$.*

Proof. Straightforward application of Theorems 9.1 and 9.2. □

This approximation is much better than the previous one: we have here that $\epsilon \approx \frac{\pi^2}{2^{2k+1}}$, which means that dividing the accuracy by four can be achieved by increasing k by 1, which corresponds to adding 2 variables, 1 equality and 2 inequality constraints (compare to the previous situation which needed to double the number of inequalities to reach to same goal). For example, an accuracy of $\epsilon = 10^{-4}$ can be obtained with $k = 8$, i.e. with 16 inequalities, 9 equalities and 18 variables (as opposed to 223 inequalities with the previous approach).

We are now in position to convert this polyhedral approximation of $\mathcal{B}_2(1)$ into an approximation of $\mathbb{L}^2$. We define the set $\mathcal{L}_k \in \mathbb{R}^{2k+3}$ as

$$\left\{(r, \alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_k) \in \mathbb{R}^{2k+3} \mid \left\{ \begin{array}{rcrcl} \alpha_{i+1} & = & \alpha_i \cos\frac{\pi}{2^i} & + & \beta_i \sin\frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \beta_i \cos\frac{\pi}{2^i} & - & \alpha_i \sin\frac{\pi}{2^i} \\ -\beta_{i+1} & \leq & \beta_i \cos\frac{\pi}{2^i} & - & \alpha_i \sin\frac{\pi}{2^i} \\ r & = & \alpha_k \cos\frac{\pi}{2^k} & + & \beta_k \sin\frac{\pi}{2^k} \end{array} \right. \forall\ 0 \leq i < k \right\}.$$

Note the close resemblance between this set and $\mathcal{D}_k$, the only difference being the introduction of an additional variable r in the last equality constraint. This set $\mathcal{L}_k$ is our final polyhedral approximation of $\mathbb{L}^2$. Obviously, before we give proof of this fact, we need a measure of the quality of an approximation in the case of a second-order cone. This is the purpose of the next definition.

Definition 9.2. A set $S \subseteq \mathbb{R}^{n+1}$ is said to be an ϵ-approximation of the second-order cone $\mathbb{L}^n$ if and only if we have

$$\mathbb{L}^n \subseteq S \subseteq \mathbb{L}^n_\epsilon = \{(r, x) \in \mathbb{R} \times \mathbb{R}^n \mid \|x\| \leq (1+\epsilon)r\}$$

where $\mathbb{L}^n_\epsilon$ is an ϵ-relaxed second-order cone.

This definition extends our definition of ϵ-approximation for the unit disc $\mathcal{B}_2(1)$. The next theorem demonstrates how Corollary 9.1 on the accuracy of the polyhedral approximation $\mathcal{D}_k$ for $\mathcal{B}_2(1)$ can be converted into a result on the accuracy of $\mathcal{L}_k$ for $\mathbb{L}^2$.

Theorem 9.3. *The projection of $\mathcal{L}_k$ on the subspace of its three variables (r, α_0, β_0) is a polyhedral approximation of $\mathbb{L}^2$ with accuracy $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$.*

Proof. Assuming $r > 0$ for the moment, we first establish a link between $\mathcal{D}_k$ and $\mathcal{L}_k$. It is indeed straightforward to check using the corresponding definitions that the following equivalence holds

$$(r, \alpha_0, \ldots, \alpha_k, \beta_0, \ldots, \beta_k) \in \mathcal{L}_k \Leftrightarrow (\frac{\alpha_0}{r}, \ldots, \frac{\alpha_k}{r}, \frac{\beta_0}{r}, \ldots, \frac{\beta_k}{r}) \in \mathcal{D}_k\ , \tag{9.5}$$

since

$$\left\{ \begin{array}{rcrcl} \alpha_{i+1} & = & \alpha_i \cos\frac{\pi}{2^i} & + & \beta_i \sin\frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \beta_i \cos\frac{\pi}{2^i} & - & \alpha_i \sin\frac{\pi}{2^i} \\ -\beta_{i+1} & \leq & \beta_i \cos\frac{\pi}{2^i} & - & \alpha_i \sin\frac{\pi}{2^i} \\ r & = & \alpha_k \cos\frac{\pi}{2^k} & + & \beta_k \sin\frac{\pi}{2^k} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{rcrcl} \frac{\alpha_{i+1}}{r} & = & \frac{\alpha_i}{r} \cos\frac{\pi}{2^i} & + & \frac{\beta_i}{r} \sin\frac{\pi}{2^i} \\ \frac{\beta_{i+1}}{r} & \geq & \frac{\beta_i}{r} \cos\frac{\pi}{2^i} & - & \frac{\alpha_i}{r} \sin\frac{\pi}{2^i} \\ -\frac{\beta_{i+1}}{r} & \leq & \frac{\beta_i}{r} \cos\frac{\pi}{2^i} & - & \frac{\alpha_i}{r} \sin\frac{\pi}{2^i} \\ 1 & = & \frac{\alpha_k}{r} \cos\frac{\pi}{2^k} & + & \frac{\beta_k}{r} \sin\frac{\pi}{2^k} \end{array} \right.$$

which means that $\mathcal{L}_k$ is nothing more than the homogenized polyhedral cone corresponding to $\mathcal{D}_k$.

Let us now suppose $(r, x_1, x_2) \in \mathbb{L}^2$. Equivalence (9.4) implies $(\frac{x_1}{r}, \frac{x_2}{r}) \in \mathcal{B}_2(1)$, which in turn implies by Corollary 9.1 that there exists a vector $(\alpha, \beta) \in \mathbb{R}^{2k}$ such that $(\frac{x_1}{r}, \alpha, \frac{x_2}{r}, \beta)$ belongs to $\mathcal{D}_k$. Using the link (9.5), this last inclusion is equivalent to $(r, x_1, r\alpha, x_2, r\beta) \in \mathcal{L}_k$, which means that (r, α_0, β_0) belongs to the projection of $\mathcal{L}_k$ on the subspace (r, α_0, β_0). We have thus shown that this projection is a relaxation of $\mathbb{L}^2$, the first condition for it to be an ϵ-approximation of $\mathbb{L}^2$.

Supposing now (r, x_1, x_2) belongs to the projection of $\mathcal{L}_k$, there exists a vector $(\alpha, \beta) \in \mathbb{R}^{2k}$ such that $(r, x_1, \alpha, x_2, \beta) \in \mathcal{L}_k$. The equivalence (9.5) implies then that $(\frac{x_1}{r}, \frac{\alpha}{r}, \frac{x_2}{r}, \frac{\beta}{r}) \in \mathcal{D}_k$, which means that $(\frac{x_1}{r}, \frac{x_2}{r})$ belongs to the projection of $\mathcal{D}_k$ on its subspace (α_0, β_0). Using now Corollary 9.1, which states that this projection is an ϵ-approximation of $\mathcal{B}_2(1)$ with $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$, we can write that $\left\|(\frac{x_1}{r}, \frac{x_2}{r})\right\| \leq 1 + \epsilon$, which can be rewritten as $\|(x_1, x_2)\| \leq (1 + \epsilon)r$, which is the exactly the second condition for this projection to be an ϵ-approximation of $\mathbb{L}^2$.

The last task we have to accomplish is to check what happens in the case where $r \leq 0$. Suppose $(r, x_1, \alpha, x_2, \beta) \in \mathcal{L}_k$. Looking at the definition of $\mathcal{L}_k$, and using the same reasoning as in the proof of Theorem 9.2, it is straightforward to show that the variables α_0 and β_0 can only be equal to 0 when $r = 0$, and that they cannot satisfy the constraints if $r < 0$ (i.e. in the first case the set $\{P_0\}$ is equal to $\{(0, 0)\}$ while in the second case $\{P_0\} = \emptyset$). Since this is also the situation of the second-order cone $\mathbb{L}^2$, our approximation is exact when $r \leq 0$, and we can conclude that the projection of $\mathcal{L}_k$ on the subspace of its three variables (r, α_0, β_0) is an ϵ-approximation of the three dimensional second-order cone $\mathbb{L}^2$ with $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$. □

9.2.5 Reducing the approximation

Our polyhedral approximation $\mathcal{L}_k$ features $2k + 3$ variables, $2k$ linear inequalities and $k + 1$ linear equalities. It is possible to reduce these numbers by pivoting out a certain number of variables. Namely, using the set of constraints $\alpha_{i+1} = \alpha_i \cos\frac{\pi}{2^i} + \beta_i \sin\frac{\pi}{2^i}$ for $0 \leq i < k$, we can replace α_k by a linear combination of α_{k-1} and β_{k-1}, then replace α_{k-1} by a linear combination of α_{k-2} and β_{k-2}, and so on until all variables α_i have been replaced except α_0 (which cannot and should not be pivoted out since it belongs to the projected approximation). The last equality $r = \alpha_k \cos\frac{\pi}{2^k} + \beta_k \sin\frac{\pi}{2^k}$ can also be used to pivot out β_k.

The resulting polyhedron has then $k+2$ variables $(r, \alpha_0, \beta_0, \ldots, \beta_{k-1})$, $2k$ linear inequalities and no linear equality. However, it should be noted that the constraint matrix describing the reduced polyhedron is denser than in the original approximation, i.e. it contains many more nonzero elements, as depicted on Figure 9.3 in the case $k = 15$ (which also mentions the number of nonzero elements in each case).

This denser constraint matrix has of course a negative impact on the efficiency of the algorithm used to solve the approximation problems, so that computational experiments are needed to decide whether this is enough to counterbalance the advantage of a reduced number of equalities and variables Indeed, preliminary testing on a few problems representative of the ones we are going to consider in Section 9.3 led us to the conclusion that pivoting out the variables is beneficial, leading roughly to a 20% reduction of computing times.

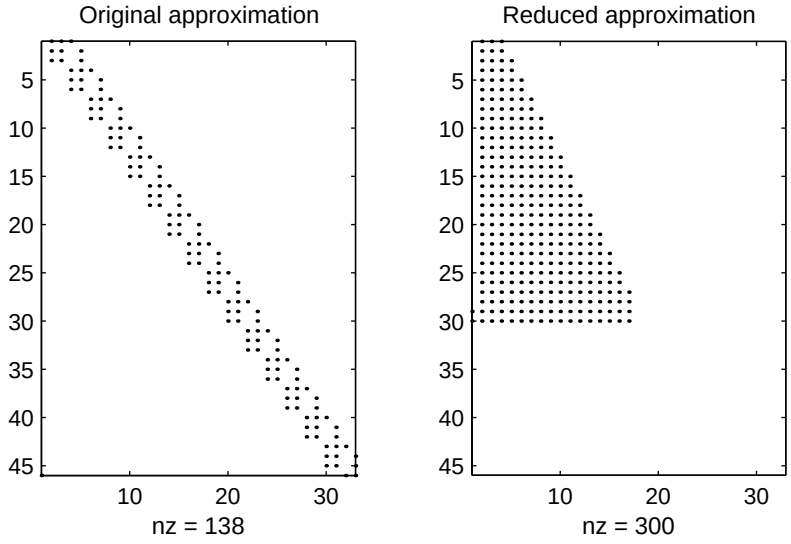


Figure 9.3: Constraint matrices for $\mathcal{L}_{15}$ and its reduced variant.

Another interesting remark can be made when we have to approximate a second-order cone whose components are restricted to be nonnegative. Namely, if we know beforehand that x_1 and x_2 cannot be negative, the polyhedral approximation of $\mathbb{L}^2$ can be reduced. Indeed, looking back at the proof of Theorem 9.2, we see that the set of points P_2 is bounded by 2^{k-2} consecutive sides of the regular 2^k-sided polyhedron (see for example the set $\{P_2\}$ depicted in Figure 9.2). Combining this with the restriction that α_2 and β_2 are nonnegative, we have that the set $\{P_2\}$ is exactly equal to the restriction of $\mathcal{P}_{2^k}$ to the positive orthant, and is thus a valid ϵ-approximation of $\mathbb{L}^2$ on this positive orthant with $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$. This observation leads to the formulation of a reduced polyhedral approximation $\mathcal{L}'_k$ defined by

$$\left\{(r, \alpha_2, \ldots, \alpha_k, \beta_2, \ldots, \beta_k) \in \mathbb{R}^{2k-1} \mid \left\{ \begin{array}{rcl} \alpha_{i+1} & = & \alpha_i \cos\frac{\pi}{2^i} + \beta_i \sin\frac{\pi}{2^i} \\ \beta_{i+1} & \geq & \beta_i \cos\frac{\pi}{2^i} - \alpha_i \sin\frac{\pi}{2^i} \\ -\beta_{i+1} & \leq & \beta_i \cos\frac{\pi}{2^i} - \alpha_i \sin\frac{\pi}{2^i} \\ r & = & \alpha_k \cos\frac{\pi}{2^k} + \beta_k \sin\frac{\pi}{2^k} \end{array} \right. \forall\ 2 \leq i < k \right\},$$

whose projection of the subspace of (r, α_2, β_2) approximates the nonnegative part $\mathbb{L}^2$. This approximation features $2k-1$ variables, $2k-4$ linear inequalities and $k-1$ linear equalities and can be reduced to k variables, $2k-4$ linear inequalities and no linear equality if we perform the pivoting described above.

At this stage, we would like to compare our approximation with the one presented in [BTN98]. Both of these feature the same accuracy $\epsilon = \cos(\frac{\pi}{2^k})^{-1} - 1$ (with parameter ν in [BTN98] equal to $k-1$ in our setting). However, Ben-Tal and Nemirovski do not make explicit that the projection of their polyhedral approximation is equal to the regular 2^k-sided polyhedron in $\mathbb{R}^2$ and only prove the corresponding accuracy result.

Table 9.1 compares the sizes of the polyhedral approximations in three cases: the original approximation, the reduced approximation where variables α_i are pivoted out and the nonnegative approximation $\mathcal{L}'_k$ (also with variables α_i pivoted out).

Table 9.1: Comparison of our approximation $\mathcal{L}_k$ with [BTN98].

	Original		Reduced		Nonnegative	
	[BTN98]	$\mathcal{L}_k$	[BTN98]	$\mathcal{L}_k$	[BTN98]	$\mathcal{L}'_k$
Variables	$2k+3$	$2k+3$	$k+4$	$k+2$	$k+2$	k
Inequalities	$2k+4$	$2k$	$2k+4$	$2k$	$2k$	$2k-4$
Equalities	$k-1$	$k+1$	0	0	0	0

Our version uses 4 less inequality constraints in all three cases. It also features 2 more equality constraints in the original approximation, which turns out to be an advantage since it allows us to pivot out more variables in the reduced versions. Both the reduced and the nonnegative versions of $\mathcal{L}_k$ use 2 less variables than their counterparts in the original article of Ben-Tal and Nemirovski.

9.2.6 An approximation of $\mathbb{L}^n$

We are now going to use the decomposition presented in Section 9.2.2 and our polyhedral approximation $\mathcal{L}_k$ for $\mathbb{L}^2$ to build an approximation for $\mathbb{L}^n$. Recall that expression (9.3) decomposed $\mathbb{L}^n$ into $\lfloor n/2 \rfloor$ three-dimensional second-order cones $\mathbb{L}^2$ and a single larger cone $\mathbb{L}^{\lceil \frac{n}{2} \rceil}$. Applying this decomposition recursively, we can decompose $\mathbb{L}^{\lceil \frac{n}{2} \rceil}$ into $\lfloor \lceil n/2 \rceil / 2 \rfloor$ second-order cones $\mathbb{L}^2$ with a remaining larger cone $\mathbb{L}^{\lceil \lceil n/2 \rceil / 2 \rceil}$, which can be again decomposed into $\lfloor \lceil \lceil n/2 \rceil / 2 \rceil / 2 \rfloor$ cones $\mathbb{L}^2$, etc.

Calling q_k the number of three-dimensional second-order cones appearing in the decomposition at each stage of this procedure and r_k the corresponding size of the remaining cone, we have initially $q_0 = 0$, $r_0 = n$ and

$$q_k = \lfloor \frac{r_{k-1}}{2} \rfloor \text{ and } r_k = \lceil \frac{r_{k-1}}{2} \rceil \ \forall \ k > 0 \ .$$

Obviously, r_k is strictly decreasing and we must eventually end up with r_k equal to 2. Indeed, it is easy to see that $2^{i-1} < r_{k-1} \leq 2^i$ implies $2^{i-2} < r_k \leq 2^{i-1}$ and a simple recursive argument shows then that if $2^{m-1} < n \leq 2^m$ we have $2^{m-k-1} < r_k \leq 2^{m-k}$ and thus that $r_{m-1} = 2$. At this stage, the remaining second-order cone is $\mathbb{L}^2$, which we can add to the decomposition in the last stage with $q_m = 1$ to have $r_m = 0$. Our decomposition has thus in total m stages. We also showed in Section 9.2.2 that the total of $\mathbb{L}^2$ cones in the final decomposition is equal to $\sum_{i=1}^m q_i = n - 1$, and we also note for later use that $2^{m-k} < r_{k-1} \leq 2^{m-k+1}$ implies $2^{m-k-1} \leq q_k \leq 2^{m-k}$.

We ask ourselves now what happens if each of the second-order cones appearing in this decomposition is replaced by an ϵ-approximation. Namely, suppose each of the $\lfloor n/2 \rfloor$ second-order cones $\mathbb{L}^2$ in expression (9.3) is replaced by an $\epsilon^{(i)}$-approximation ($0 \leq i \leq \lfloor n/2 \rfloor$), while the remaining larger cone is replaced by an ϵ'-approximation. We end up with the set

$$\left\{ (r, x, y) \in \mathbb{R}_+ \times \mathbb{R}^{n + \lfloor \frac{n}{2} \rfloor} \mid (y_i, x_{2i-1}, x_{2i}) \in \mathbb{L}^2_{\epsilon^{(i)}}, \ 1 \leq i \leq \lfloor \frac{n}{2} \rfloor, \ \begin{cases} (r, y) \in \mathbb{L}^{\lceil n/2 \rceil}_{\epsilon'} & (n \text{ even}) \\ (r, y, x_n) \in \mathbb{L}^{\lceil n/2 \rceil}_{\epsilon'} & (n \text{ odd}) \end{cases} \right\} ,$$

whose constraints are equivalent to

$$x_{2i-1}^2 + x_{2i}^2 \leq (1+\epsilon^{(i)})^2 y_i^2,\ 1 \leq i \leq \lfloor \frac{n}{2} \rfloor,\ \left\{ \begin{array}{ll} \sum_{i=1}^{\frac{n}{2}} y_i^2 & \leq (1+\epsilon')^2 r^2 \ (n \text{ even}) \\ \sum_{i=1}^{\frac{n}{2}} y_i^2 + x_n^2 & \leq (1+\epsilon')^2 r^2 \ (n \text{ odd}) \end{array} \right\}.$$

Ideally, we would like this decomposition to be an ϵ-approximation of $\mathbb{L}^n$. We already know that it is a relaxation of $\mathbb{L}^n$, since each approximation of $\mathbb{L}^2$ is itself a relaxation. We have thus to concentrate on the second condition defining an ϵ-approximation, $\|x\| \leq (1+\epsilon)r$. Writing

$$\textstyle\sum_{i=1}^{2\lfloor n/2 \rfloor} x_i^2 \leq \sum_{i=1}^{\lfloor n/2 \rfloor} (1+\epsilon^{(i)})^2 y_i^2\ ,$$

we would like to bound the quantity on the right hand-side. Unfortunately, we only know a bound on the sum of y_i^2's, which forces us to write

$$\textstyle\sum_{i=1}^{2\lfloor n/2 \rfloor} x_i^2 \leq (1+\max_i\ \epsilon^{(i)})^2 \sum_{i=1}^{\lfloor n/2 \rfloor} y_i^2\ \Rightarrow\ \sum_{i=1}^{n} x_i^2 \leq (1+\max_i\ \epsilon^{(i)})^2 (1+\epsilon')^2 r^2\ .$$

This shows that our decomposition is an approximation of $\mathbb{L}^n$ with accuracy $\epsilon = (1 + \max_i\ \epsilon_i)(1+\epsilon')$. This immediately implies that there is no point in approximating with different accuracies the $\lfloor n/2 \rfloor$ small second-order cones $\mathbb{L}^2$ appearing in the decomposition, since only the largest of these accuracies has an influence on the resulting approximation for $\mathbb{L}^n$. Applying now our decomposition recursively to the remaining cone, and choosing at each stage k a unique accuracy ϵ_k for all the $\mathbb{L}^2$ cones, we find that $1+\epsilon = \prod_{k=1}^{m}(1+\epsilon_k)$,, i.e. that the final accuracy of our polyhedral approximation is the product of the accuracies chosen at each stage of the decomposition (note that, unlike the situation for a single stage, there is no reason here to choose all ϵ_k accuracies to be equal to each other).

9.2.7 Optimizing the approximation

The previous section has shown how to build a polyhedral approximation of $\mathbb{L}^n$ and how its quality depends on the accuracy of the approximations used at each stage of the decomposition. Our goal is here to optimize these quantities, i.e. given a target accuracy ϵ for $\mathbb{L}^n$, find the values of ϵ_k $(1 \leq k \leq m)$ that lead to the smallest polyhedral approximation, i.e. the one with the smallest number of variables and constraints.

Let us suppose we use at stage k the approximation $\mathcal{L}_{u_k}$ with $u_k + 2$ variables and $2u_k$ linear inequalities (i.e. with variables α pivoted out of the formulation), which has an accuracy $\epsilon_k = \cos(\frac{\pi}{2^{u_k}})^{-1} - 1$. Recalling notation q_k for the number of cones $\mathbb{L}^2$ introduced at stage k of the decomposition, the final polyhedral approximation has thus an accuracy equal to $\prod_{k=1}^{m} \cos(\frac{\pi}{2^{u_k}})^{-1}$ with $2\sum_{k=1}^{m} q_k u_k$ inequalities and $n + \sum_{k=1}^{m} q_k u_k$ variables. Indeed, we have n original x_i variables and u_k additional variables for each of the q_k approximations at stage k, since the first two variables in these approximations are coming from the previous stage. We observe that the main quantity to be minimized is $\sum_{k=1}^{m} q_k u_k$ for both the number of variables and inequalities, which leads to the following optimization problem:

$$\sigma_{n,\epsilon} = \min_{u \in \mathbb{N}^m} \sum_{k=1}^{m} q_k u_k \quad \text{s.t.} \quad \prod_{k=1}^{m} \cos(\frac{\pi}{2^{u_k}})^{-1} \leq 1+\epsilon\ . \tag{9.6}$$

A possible choice for variables u_k is to take them all equal. Plugging this unique value into the accuracy constraint, we readily find that u_k has to be equal to

$$u_k = \lceil \log_2\big(\pi/\arccos(1+\epsilon)^{-1/m}\big) \rceil$$

and that when the dimension of the cone $\mathbb{L}^n$ (and thus m) tends to $+\infty$, we have $u_k = \mathcal{O}\left(\log \frac{m}{\epsilon}\right)$ and $\sigma_{n,\epsilon} = \mathcal{O}\left(n \log \frac{m}{\epsilon}\right)$.

This obviously does not lead to an optimal solution of (9.6). Indeed, since the number of approximations is decreasing as we move from one stage of the decomposition to the next, it is intuitively clear that trading a lower accuracy for first stages against a higher accuracy for the last stages will be beneficial, since the lowering of the number of variables and inequalities in the first stages will affect many more constraints than the increase of size for the last stages. This implies that the components u_k of any optimal solution of (9.6) will have to be in increasing order.

Finding a closed form optimal solution of (9.6) does not appear to be possible, but we can find a good suboptimal solution using some approximations. We first introduce variables v_k such that $v_k = 4^{-u_k} \Leftrightarrow u_k = -\log_4 v_k$ and rewrite problem (9.6) as

$$\sigma_{n,\epsilon} = \min_{v \in \mathbb{R}^m} -\log_4 \prod_{k=1}^m v_k^{q_k} \quad \text{s.t.} \quad \sum_{k=1}^m \log(\cos(\pi\sqrt{v_k})^{-1}) \leq \log(1+\epsilon) \ .$$

Since $u_k \geq 2$, we have $\pi\sqrt{v_k} \leq \frac{\pi}{4}$ and we can use the easily proven[2] inequality

$$\log(\cos(x)^{-1}) \leq (\frac{3x}{4})^2$$

valid for all $0 \leq x \leq \frac{\pi}{4}$ to write

$$\sigma_{n,\epsilon} = -\max_{v \in \mathbb{R}^m} \log_4 \prod_{k=1}^m v_k^{q_k} \quad \text{s.t.} \quad \sum_{k=1}^m v_k \leq \frac{16}{9}\pi^{-2}\log(1+\epsilon) = K(\epsilon) \ ,$$

which is thus a restriction of our original problem. It amounts to maximizing a product of variables whose sum is bounded, a problem whose optimality conditions are well-known. In our case, they can written as

$$\frac{v_1}{q_1} = \frac{v_2}{q_2} = \ldots \frac{v_m}{q_m} = \frac{\sum_{k=1}^m v_k}{\sum_{k=1}^m q_k} = \frac{K(\epsilon)}{n-1} \Rightarrow v_k = \frac{q_k}{n-1}K(\epsilon) \Leftrightarrow u_k = \log_4 \frac{n-1}{q_k} - \log_4 K(\epsilon) \ .$$

However u_k must be integer, so that we have to degrade this solution further and round it towards a larger integer. Using that fact that $n-1 \leq 2^m$ and $q_k \geq 2^{m-k-1}$, we have

$$\frac{n-1}{q_k} \leq \frac{2^m}{2^{m-k-1}} = 2^{k+1} \Rightarrow \log_4 \frac{n-1}{q_k} \leq \log_4 2^{k+1} = \frac{k+1}{2} \ ,$$

so that we can take $u_k = \lceil \frac{k+1}{2} \rceil - \lfloor \log_4 K(\epsilon) \rfloor$ as our suboptimal integer solution for (9.6).

[2] This inequality can be easily checked by plotting the graphs of its two sides on the interval $[0\ \frac{\pi}{4}]$.

Let us plug now these values into the objective function $\sigma_{n,\epsilon}$: we find

$$\begin{aligned}
\sum_{k=1}^{m} q_k u_k &= \sum_{k=1}^{m} q_k \lceil \frac{k+1}{2} \rceil - \sum_{k=1}^{m} q_k \lfloor \log_4 K(\epsilon) \rfloor \\
&\leq \sum_{k=1}^{m} q_k (\frac{k}{2} + 1) - (n-1)\lfloor \log_4 K(\epsilon) \rfloor \quad (\text{using } \sum_{k=1}^{m} q_k = n-1) \\
&\leq \sum_{k=1}^{m} 2^{m-k} (\frac{k}{2} + 1) - (n-1)\lfloor \log_4 K(\epsilon) \rfloor \quad (\text{using } q_k \leq 2^{m-k}) \\
&\leq 2^{m+1} - \frac{m}{2} - 2 - (n-1)\lfloor \log_4 K(\epsilon) \rfloor \\
&\leq 4(n-1) - (n-1)\lfloor \log_4 K(\epsilon) \rfloor \quad (\text{using } 2^{m-1} \leq n-1) \\
&\leq (n-1)\lceil 4 - \log_4 K(\epsilon) \rceil = (n-1)\lceil 4 - \log_4 \frac{16}{9} + \log_4 \pi^2 - \log_4 \log(1+\epsilon) \rceil \\
&\leq (n-1)\lceil 5.3 - \log_4 \log(1+\epsilon) \rceil
\end{aligned}$$

(where we have used at the fourth line the fact that $\sum_{k=1}^{m} 2^{m-k}(\frac{k}{2}+1) = 2^{m+1} - \frac{m}{2} - 2$, which is easily proved recursively). We can wrap this result into the following theorem:

Theorem 9.4. *For every $\epsilon < \frac{1}{2}$, there exists a polyhedron with no more than*

$$2 + (n-1)\lceil 5.3 - \log_4 \log(1+\epsilon) \rceil = \mathcal{O}\big(n \log \frac{1}{\epsilon}\big) \text{ variables}$$

and

$$2 + 2(n-1)\lceil 4.3 - \log_4 \log(1+\epsilon) \rceil = \mathcal{O}\big(n \log \frac{1}{\epsilon}\big) \text{ inequalities}$$

whose projection on a certain subspace of $n+1$ variables is an ϵ-approximation of the second-order cone $\mathbb{L}^n \subseteq \mathbb{R}^{n+1}$.

Proof. This a consequence of the previous derivation, which showed that choosing $u_k = \lceil \frac{k+1}{2} \rceil - \lfloor \log_4 K(\epsilon) \rfloor$ lead to an ϵ-approximation of $\mathbb{L}^n$ with $n + \sigma_{n,\epsilon}$ variables and $2\sigma_{n,\epsilon}$ linear inequalities, with $\sigma_{n,\epsilon} = (n-1)\lceil 5.3 - \log_4 \log(1+\epsilon) \rceil$. However, the size of this polyhedron can be further reduced using $\mathcal{L}'_k$, the polyhedral approximation of the nonnegative part of $\mathbb{L}^2$. Indeed, looking at the decomposition (9.3), we see that all the y variables used in the second stage of the decomposition are guaranteed to be nonnegative, since we have in our approximation $(y_i, x_{2i-1}, x_{2i}) \in \mathcal{L}_{u_1}$ which implies $y_i \geq 0$. This means that we can use for the second stage and the following our reduced approximation $\mathcal{L}'_{u_k}$, known to be valid when its first two variables are restricted to the nonnegative orthant $\mathbb{L}^2$, which uses 2 less variables and 4 less inequalities per cone. Since there are $\frac{n}{2}$ cones in the first stage of the decomposition and $n-1$ cone in total, we can use $\frac{n}{2} - 1$ reduced approximations $\mathcal{L}'$, which give us a total saving of $n-2$ variables and $2n-4$ constraints[3]. Combining this with the value of $\sigma_{n,\epsilon}$, we find that our approximation has $2 + (n-1)\lceil 5.3 - \log_4 \log(1+\epsilon) \rceil$ variables and $2 + \lceil 4.3 - \log_4 \log(1+\epsilon) \rceil$ inequalities.

[3]This reasoning was made for an even n. In the case of an odd n, we have one cone in the decomposition for which only the first variable is known to be nonnegative. It is possible to show that there exists a polyhedral approximation adapted to this situation that uses 1 less variable and 2 less inequalities than the regular approximation, which allows us to write exactly the same results as for an even n.

We also have to prove the asymptotic behaviour of $\sigma_{n,\epsilon}$ when n tends to infinity. Indeed, we have $\log(1+\epsilon) \geq \frac{\epsilon}{2}$ when $\epsilon < \frac{1}{2}$, which implies $-\log_4 \log(1+\epsilon) \leq \log_4 \frac{1}{\epsilon}$. This leads to $\lceil 5.3 - \log_4 \log(1+\epsilon) \rceil = \mathcal{O}\big(\log \frac{1}{\epsilon}\big)$, which is enough to prove the theorem. □

This result is better than the one we previously obtained choosing all u_k's equal to each other: indeed, we had in that case $\sigma_{n,\epsilon} = \mathcal{O}\big(n \log \frac{m}{\epsilon}\big) = \mathcal{O}\big(n \log \frac{\log n}{\epsilon}\big)$ while we have here $\mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$. For a fixed accuracy ϵ, our first choice translates into $\sigma_{n,\epsilon} = \mathcal{O}\left(n \log\log n\right)$ when n tends to infinity while our optimized choice of u_k's leads to $\sigma_{n,\epsilon} = \mathcal{O}\left(n\right)$, which is better. We note however that if we fix n and let ϵ tends to 0, the asymptotic behaviour is the same in both cases , namely we have $\sigma_{n,\epsilon} = \mathcal{O}\big(\log \frac{1}{\epsilon}\big)$.

Ben-Tal and Nemirovski achieve essentially the same result in [BTN98], albeit in the special case when n is a power of two. Our proof has the additional advantage of providing a closed form for the parameters u_k as well as for the total size of the polyhedral approximation, for all values of n. They also prove that the number of inequalities of a ϵ-approximation of $\mathbb{L}^n$ must be greater that $\mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$, i.e. that the order of the result of Theorem 9.4 is not improvable.

9.2.8 An approximation of second-order cones optimization

The previous sections have proven the existence of a polyhedral approximation of the second-order cone with a moderate size (growing linearly with the dimension of the cone and the logarithm of the accuracy). However, we have to point out that these polyhedrons are not strictly speaking approximations of the second-order cone: more precisely, it is their projection on a certain subspace than is an ϵ-approximation of $\mathbb{L}^n$.

This does not pose any problem when trying to approximate a second-order cone optimization problem with linear optimization. Let us suppose we want to approximate problem (9.1), which we recall here for convenience,

$$\inf \; c^T x \quad \text{s.t.} \quad Ax = b \text{ and } x^k \in \mathbb{L}^{n_k} \; \forall k = 1, 2, \ldots, r \tag{9.1}$$

with ϵ-approximations of the second order cones $\mathbb{L}^{n_k}$. Theorem 9.4 implies the existence of a polyhedron

$$\mathcal{Q}_k = \left\{ (x^k, y^k) \in \mathbb{R}^{n_k+1} \times \mathbb{R}^{\mathcal{O}\left(n_k \log \frac{1}{\epsilon}\right)} \mid A_k (x^k, y^k)^T \geq 0 \right\}$$

with

$$A_k \in \mathbb{R}^{\mathcal{O}\left(n_k \log \frac{1}{\epsilon}\right) \times \mathcal{O}\left(n_k \log \frac{1}{\epsilon}\right)} \; ,$$

whose projection on the subspace (r, x) is an ϵ-approximation of $\mathbb{L}^{n_k}$, which allows us to write the following linear optimization problem[4]

$$\min \; c^T x \quad \text{s.t.} \quad Ax = b \text{ and } A_k (x^k, y^k)^T \geq 0 \; \forall k = 1, 2, \ldots, r \; . \tag{9.7}$$

We note that the fact that our approximations are projections is handled in a seamless way by this formulation: the only difference with the use of a direct approximation of the cones

[4] We could replace the inf of problem (9.1) by a min since it is well-known that linear optimization problems always attain their optimal objectives, see Chapter 3.

$\mathbb{L}^{n_k}$ is the addition of the auxiliary variables y^k to the formulation. This problem features $\sum_{k=1}^{r} \mathcal{O}\big(n_k \log \frac{1}{\epsilon}\big) = \mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$ variables, m equality constraints and $\sum_{k=1}^{r} \mathcal{O}\big(n_k \log \frac{1}{\epsilon}\big) = m + \mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$ homogeneous inequality constraints. We also point out as a minor drawback of this formulation the fact that it involves irrational coefficients, namely the quantities $\sin \frac{\pi}{2^i}$ and $\cos \frac{\pi}{2^i}$ occurring in the definition of $\mathcal{L}_k$. However, it rational coefficients are really needed (for example if one wants to work with a complexity model based on exact arithmetic), it is possible to replace those quantities with rational approximations while keeping an essentially equivalent accuracy for the resulting polyhedral approximation, i.e. featuring the same asymptotic behaviour.

To conclude this section, we are going to compare the algorithmic complexity of solving problem (9.1) either directly or using our polyhedral approximation. The best complexity obtained so far[5] for solving a linear program with v variables up to accuracy ϵ is $\mathcal{O}\big(v^{3.5} \log(\frac{1}{\epsilon})\big)$ arithmetic operations (using for example a short-step path-following method, see Chapter 1). In our case, assuming we solve the approximate problem (9.7) up to the same accuracy that the one used to approximate the second-order cones, this leads to a complexity equal to $\mathcal{O}\big(n^{3.5} \log(\frac{1}{\epsilon})^{4.5}\big)$.

On the other hand, solving problem (9.1) can be done using $\mathcal{O}\big(\sqrt{r}\, n^3 \log(\frac{1}{\epsilon})\big)$ arithmetic operations, using for example a potential reduction approach, see e.g. [LVBL98]. If $r = \mathcal{O}(1)$, i.e. if the number of cones used in the formulation is bounded, the second complexity is better, both if $n \to +\infty$ or $\epsilon \to 0$. However, if $r = \mathcal{O}(n)$, which means that the dimension of the cones used in the formulation is bounded, both complexity become equivalent from the point of view of the dimension n, but the second one is still better when letting the accuracy tend to 0. We conclude that the direct solving of (9.1) as a second-order cone problem is superior from the point of view of algorithmic complexity. The purpose of the second part of this chapter will be to test whether this claim is also valid for computational experiments.

9.2.9 Accuracy of the approximation

The linearizing scheme for second-order cone optimization presented in the previous section is based on a polyhedral approximation whose accuracy is guaranteed in the sense of Definition 9.2. It is important to realize that this bound on the accuracy of the approximation does not imply a bound on the accuracy of the solutions (or the objective value) of the approximated problem.

Indeed, let us consider the following set:

$$\left\{(r, x_1, x_2) \in \mathbb{R}^3 \mid r - x_2 = \frac{1}{2} \text{ and } (r, x_1, x_2) \in \mathbb{L}^2\right\} .$$

This set can be seen as the feasible region of a second-order cone problem. Using the fact that

$$x_1^2 + x_2^2 \leq r^2 \Leftrightarrow x_1^2 + x_2^2 \leq (x_2 + \frac{1}{2})^2 \Leftrightarrow x_1^2 - \frac{1}{4} \leq x_2 ,$$

we find that the projection of this set on the subspace (x_1, x_2) is the epigraph of the parabola $x \mapsto x^2 - \frac{1}{4}$. Let us now replace $\mathbb{L}^2$ by the polyhedral approximation $\mathcal{L}_k$. Since the resulting

[5] Using standard linear algebra and without partial updating.

set will be polyhedral, its projection on the subspace (x_1, x_2) will also be polyhedral, and we can deduce without difficulties that it is the epigraph of a piecewise linear function, as shown by Figure 9.4 (depicting the cases $k = 1$, 2, 3 and 4).

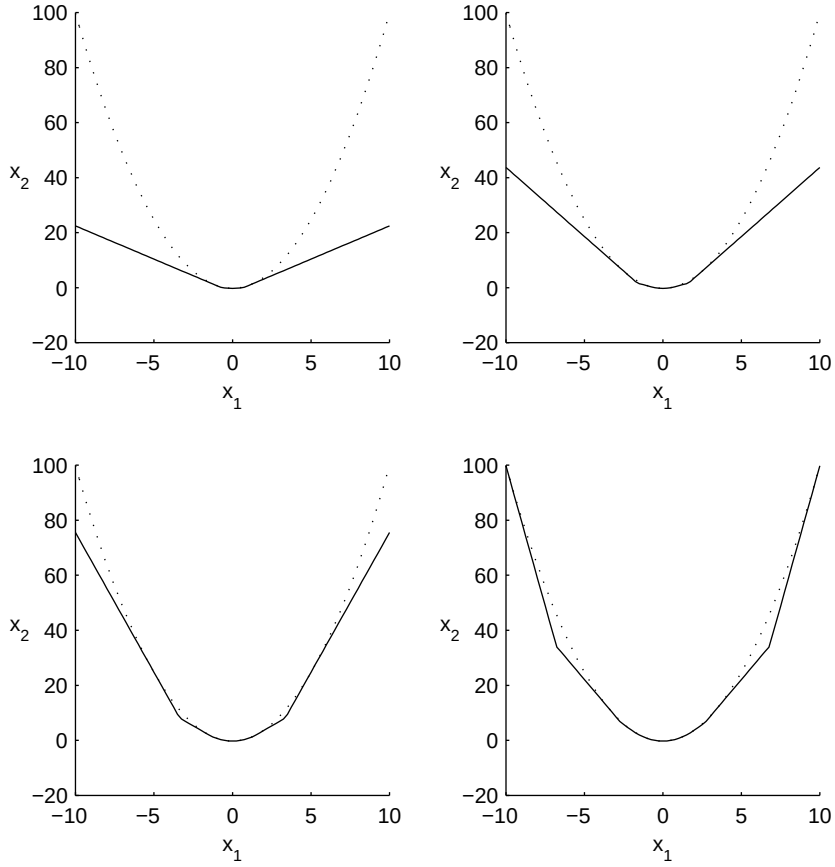


Figure 9.4: Linear approximation of a parabola using $\mathcal{L}_k$ for $k = 1, 2, 3, 4$.

Because a polyhedron has a finite number of vertices, this piecewise linear function must have a finite number of segments. Considering the rightmost piece, i.e. whose x_1 values span an interval of the type $[\alpha + \infty[$, it is obvious that it cannot approximate the parabola with a guaranteed accuracy. Indeed, the difference between the approximation and the parabola grows quadratically on this segment, which means that even the ratio of variable x_2 between the parabola and its linear approximation is not bounded.

Let us now consider the following parameterized family of second-order cone optimization problems

$$\min\ x_2 \quad \text{s.t.} \quad x_1 = \lambda,\ r - x_2 = \frac{1}{2} \text{ and } (r, x_1, x_2) \in \mathbb{L}^2\ , \tag{PB$_\lambda$}$$

which is using the same feasible set as above with an additional constraint fixing the variable x_1 to λ. Denoting the optimal objective value of (PB$_\lambda$) by $p^*(\lambda)$, we have in light of the previous discussion that $p^*(\lambda) = \lambda^2 - \frac{1}{4}$. However, we also showed that the optimal objective value $p_k^*(\lambda)$ of the approximated problem

$$\min\ x_2 \quad \text{s.t.} \quad x_1 = \lambda,\ r - x_2 = \frac{1}{2} \text{ and } (r, x_1, x_2, y) \in \mathcal{L}_k$$

must be a piecewise linear function of λ with a finite number of segments. Indeed, simple computations[6] show that the endpoints of these segments occur for

$$\lambda = \frac{\sin(i\theta)}{2\cos\frac{\theta}{2} - 2\cos(i\theta)} \text{ for } i = 1, 2, \ldots, 2^k - 1 \text{ with } \theta = \frac{\pi}{2^{k-1}} ,$$

which shows that $p^*(\lambda)$ is linear as soon as $\lambda \geq \frac{\sin\theta}{2\cos\frac{\theta}{2} - 2\cos\theta}$.

The discrepancy between the real optimum $p^*(\lambda)$ and the approximated optimum $p_k^*(\lambda)$ is thus unbounded when λ goes to infinity. Moreover, we have that the relative accuracy of $p_k^*(\lambda)$ tends to 1, the worst possible value, i.e.

$$\frac{p^*(\lambda) - p_k^*(\lambda)}{p^*(\lambda)} \to 1 .$$

Another interesting feature of this small example is that performing a complete parametric analysis for parameter λ ranging from $-\infty$ to $+\infty$ would lead to $2^k - 1$ different break points.

We conclude that we cannot give an *a priori* bound on the accuracy of the optimal objective value of the linear approximation of a second-order cone optimization problem (this remark is also valid for accuracy of the optimal solution itself, since we have in our example (PB_λ) that the optimal value of x_2 is equal to p^*).

9.3 Computational experiments

In this section, we present computational experiments with an implementation of the linearizing scheme for second-order cone optimization we have just described.

9.3.1 Implementation

The computer used to conduct those experiments is an Intel 500 MHz Pentium III with 128 megabytes of memory. We chose to use the MATLAB programming environment, developed by The MathWorks, for the following reasons:

- ◇ MATLAB is a flexible and modular environment for technical computing, two very important characteristics when developing research code. Although MATLAB may be somehow slower than a pure C or FORTRAN approach, we think that this loss of performance is more than compensated by the ease of development (especially from the point of view of graphic capabilities and debugging). Moreover, the critical (i.e. time consuming) parts of the algorithms can be coded separately in C or FORTRAN and used in MATLAB via MEX files (this is the approach taken by the solvers we mention below), which allows a well designed MATLAB program to be nearly as efficient as an equivalent pure C or FORTRAN program.

[6] Simply observe that the extremal rays of $\mathcal{L}_k$ obey to the relation $x_2 = x_1 \tan i\theta$ with $i = 1, 2, \ldots, 2^k$ and $\theta = \pi/2^{k-1}$.

◇ Efficient interior-point solvers are available on the MATLAB platform. Indeed, we used in our experiments

- The MOSEK optmization toolbox for MATLAB by EKA Consulting ApS, a full-featured optimization package including a simplex solver and primal-dual interior-point solvers for linear optimization, convex linearly and quadratically constrained optimization, second-order cone optimization, linear least square problems, linear l_1 and l_∞-norm optimization and geometric and entropy optimization [AA99, ART00]. When compared with the standard optimization toolbox from MATLAB, MOSEK is particularly efficient on large-scale and sparse problems. MOSEK can be downloaded for research and evaluation purposes at `http://www.mosek.com`.
- SeDuMi by Jos Sturm [Stu99b], another primal-dual interior-point solver which is able to handle linear, second-order cone and semidefinite optimization problems. SeDuMi is designed to take into account sparsity and complex values, and has the advantage of dealing with the very important class of semidefinite optimization, but is a little more restrictive than MOSEK concerning the input format, since problems must be entered in the standard conic form (9.1). SeDuMi can be downloaded at `http://www.unimaas.nl/~sturm/`.

The main routines we implemented are the following (source code is available in the appendix):

◇ `PolySOC2(k)` generates the polyhedron $\mathcal{L}_k$ with accuracy $\epsilon_k = \cos(\frac{\pi}{2^{u_k}})^{-1} - 1$. Variables α_i are pivoted out, so that this routine returns a polyhedron with $k+2$ variables and $2k$ inequalities. An optional parameter is available to use the reduced approximation $\mathcal{L}'_k$, valid on the nonnegative restriction of $\mathbb{L}^2$.

◇ `Steps(q, e)` computes the optimal choice for the size of the cones at each stage of the decomposition of $\mathbb{L}^n$. Indeed, `q` contains our vector q (i.e. the number of cones at each stage) and `e` is the target accuracy ϵ.

◇ `PolySOCN(n, e)` generates a e-approximation of $\mathbb{L}^n$. It uses the output of `PolySOC2` and the optimal sizes for the cones computed by `Steps`.

◇ `PolySOCLP(p, e)` linearizes the second-order cone optimization problem `p`, replacing each second-order cone constraint with a polyhedral e-approximation using `PolySOCN` and outputting a linear optimization problem.

The procedure `Steps` we implemented features some improvements when compared with the theory we presented in the previous section. Indeed, Theorem 9.4 shows that the choice $u_k = \lceil \frac{k+1}{2} \rceil - \lfloor \log_4 K(\epsilon) \rfloor$ leads to a polyhedron of size $\mathcal{O}\big(n \log \frac{1}{\epsilon}\big)$, but is not optimal for two reasons:

◇ We approximated the formula giving the accuracy of the approximation to derive u_k (namely, we used $\log(\cos(x)^{-1}) \leq (\frac{3x}{4})^2$).

◇ The optimal solution for this approximated accuracy was not guaranteed to be integer and had to be rounded to the smallest greater integer.

However, one can easily improve this choice in practice as follows. Let us suppose theory predicts some optimal values v_k for the sizes of the cones at stage k, which have to be rounded to $\lceil v_k \rceil$. Because of this rounding, the actual accuracy of the approximation will be much better than our target ϵ. Recalling now that this accuracy is equal to $(1+\epsilon_1)(1+\epsilon')$, where ϵ_1 is the accuracy of the cones in the first stage, and is thus equal to $\cos(\frac{\pi}{2^{\lceil v_1 \rceil}})^{-1} - 1$, and ϵ' is the accuracy from the cone modelled by all the remaining stages $\mathbb{L}^{\lceil n/2 \rceil}$, we can compute an upper bound for ϵ', according to

$$(1+\epsilon_1)(1+\epsilon') \leq \epsilon \Leftrightarrow \epsilon' \leq \frac{\epsilon}{1+\epsilon_1} - 1$$

which will be better (i.e. higher) than in the theoretical derivation since it takes into account the exact accuracy ϵ_1 of the first stage, rounding included. We can now apply this procedure to the second stage, i.e. computing a theoretical value for ϵ_2 and an upper bound on the accuracy of $\mathbb{L}^{\lceil \lceil n/2 \rceil /2 \rceil}$, and so on, obtaining in the end a smaller polyhedral approximation, since the required accuracies for the cones at every stage (except the first one) are higher and hence need less constraints and variables.

Still, this improved rounding does not address the first reason why our u_k's are not optimal, the fact that we do not optimize the actual formula for the accuracy. Since it seems impossible to deal with it in closed form, we implemented a dynamic programming approach to optimize it. This algorithm uses the theoretical suboptimal solution described above (including the improved rounding procedure) to provide bounds on the optimal solution and therefore reduce the computing time.

Figure 9.5 presents the graphs of the size (measured by $\sigma_{n,\epsilon}$) of our approximations in two situations: fixed dimension ($n = 50, 200, 800$) with accuracy ranging from 10^{-1} to 10^{-8} and fixed accuracy ($\epsilon = 10^{-2}, 10^{-5}, 10^{-8}$) with dimension ranging from 10 to 1000. The asymptotic behaviour of $\sigma_{n,\epsilon}$ is very clear on these graphs: we have a linear increase when n tends to $+\infty$ for a fixed accuracy and a logarithmic increase when ϵ tends to 0 for a fixed dimension (since the first graph has a logarithmic scale of abscissas).

Finally, in order to give an idea of the efficiency of our improved rounding procedure and dynamic programming resolution, we provide in Table 9.2 the value of $\sigma_{n,\epsilon}$ for different strategies, using a target accuracy $\epsilon = 10^{-8}$.

Table 9.2: Different approaches to optimize the size of a 10^{-8}-approximation of $\mathbb{L}^n$

n	Theory	All equal (rounded)	Theory (rounded)	Dynamic programming
10	171	139	141	139
100	1881	1584	1552	1537
1000	18981	15987	15688	15522
10000	189981	160140	157063	155392

The first column represents the theoretical value $\sigma_{n,\epsilon} = (n-1)\lceil 5.3 - \log_4 \log(1+\epsilon) \rceil$, the second column describes the choice of all u_k's, equal to each other, albeit using the improved rounding procedure presented above, the third column reports the choice of the theoretical

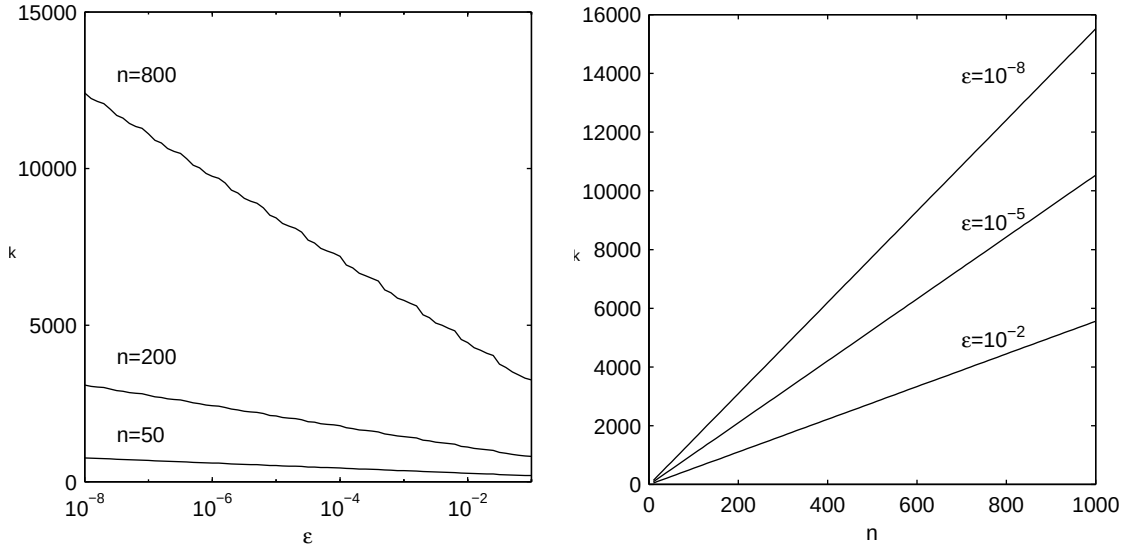


Figure 9.5: Size of the optimal approximation versus accuracy (left) and dimension (right).

value for u_k, this time with the improved rounding procedure, and the last column gives the true optimal value via our dynamic programming approach. We observe that our iterative rounding procedure improves the theoretical value of $\sigma_{n,\epsilon}$ in a noticeable way, lowering it by approximately 15%. The differences between the last three columns are less important, the dynamic programming approach giving a few additional percents of decrease in the size of the approximation.

9.3.2 Truss-topology design

We first tested our linearizing scheme for second-order cone optimization on a series of truss topology design problems. A *truss* is a structure composed of elastic bars connecting a set of nodes, like a railroad bridge or the Eiffel tower. The task consists in determining the size (i.e. the cross sectional areas) of the bars that lead to the stiffest truss when submitted to a set of forces, subject to a total weight limit. The problem we want to solve here is a multi-load truss topology design problem, which means we are simultaneously considering a set of k loading scenarios. This problem can be formulated as follows (see [BTN94]):

$$\min \sum_{i=1}^{n} \sigma_i \quad \text{s.t.} \quad \|(q_{i1}, \ldots, q_{ik})\| \leq \sigma_i \ \forall 1 \leq i \leq n \text{ and } B \begin{pmatrix} q_{1j} \\ q_{2j} \\ \vdots \\ q_{nj} \end{pmatrix} = f_j \ \forall 1 \leq j \leq k \ , \quad \text{(TTD)}$$

where n is the number of bars, k is the number of loading, vector $\sigma \in \mathbb{R}^n$ and matrix $Q \in \mathbb{R}^{n \times k}$ are the design variables, $B \in \mathbb{R}^{m \times n}$ is a matrix describing the physical configuration of the truss and $f_j \in \mathbb{R}^m, 1 \leq j \leq k$ are vectors of forces describing the loadings scenarios.

It is easily cast as a second-order cone problem in the form (9.1), since the norm constraints can be modelled as $(\sigma_i, q_{i1}, \ldots, q_{ik}) \in \mathbb{L}^k$ for all $1 \leq i \leq n$. Indeed, we have that the

variables $x \in \mathbb{R}^{k(n+1)}$, the objective $c \in \mathbb{R}^{k(n+1)}$ and the equality constraints $Ax = b$ with $A \in \mathbb{R}^{km \times k(n+1)}$ and $b \in \mathbb{R}^{km}$ are given by

$$x = \begin{pmatrix} \sigma \\ q_{11} \\ q_{21} \\ \vdots \\ q_{nk} \end{pmatrix}, \; c = \begin{pmatrix} 1^{n\times 1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \; A = \begin{pmatrix} 0^{m\times n} & B & & \\ 0^{m\times n} & & B & \\ \vdots & & & \ddots \\ 0^{m\times n} & & & B \end{pmatrix} \text{ and } b = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{pmatrix}$$

to give the following second-order cone optimization problem equivalent to (TTD):

$$\min \begin{pmatrix} 1^{n\times 1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}^T \begin{pmatrix} \sigma \\ q_{11} \\ q_{21} \\ \vdots \\ q_{nk} \end{pmatrix} \quad \text{s.t.} \quad \left\{ \begin{array}{c} \begin{pmatrix} 0^{m\times n} & B & & \\ 0^{m\times n} & & B & \\ \vdots & & & \ddots \\ 0^{m\times n} & & & B \end{pmatrix} \begin{pmatrix} \sigma \\ q_{11} \\ q_{21} \\ \vdots \\ q_{nk} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{pmatrix} \\ (\sigma_i, q_{i1}, \ldots, q_{ik}) \in \mathbb{L}^k \text{ for all } 1 \le i \le n \end{array} \right. . \tag{TP}$$

This allows us to write a dual problem in the form (9.2) in a straightforward manner:

$$\max \sum_{j=1}^{k} f_j^T y_j \quad \text{s.t.} \quad \left\{ \begin{array}{c} \begin{pmatrix} 0^{m\times n} & 0^{m\times n} & \cdots & 0^{m\times n} \\ B^T & & & \\ & B^T & & \\ & & \ddots & \\ & & & B^T \end{pmatrix} \begin{pmatrix} y_\sigma \\ y_1 \\ \vdots \\ y_k \end{pmatrix} + \begin{pmatrix} \sigma^* \\ q_{11}^* \\ q_{21}^* \\ \vdots \\ q_{nk}^* \end{pmatrix} = \begin{pmatrix} 1^{n\times 1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ (\sigma_i^*, q_{i1}^*, \ldots, q_{ik}^*) \in \mathbb{L}^k \text{ for all } 1 \le i \le n \end{array} \right. \tag{TD}$$

where $\sigma^* \in \mathbb{R}^n$, $Q^* \in \mathbb{R}^{n\times k}$, $y_\sigma \in \mathbb{R}^m$ and $y_j \in \mathbb{R}^m$ for all $1 \le j \le k$ are the dual variables. Variables σ^* and Q^* can be pivoted out of the formulation using the linear constraints

$$\sigma^* = 1^{n\times 1} \text{ and } q_{ij}^* = -b_i^T y_j \; \forall 1 \le i \le n, \; 1 \le j \le k$$

(where $b_i \in \mathbb{R}^m$ is the i^{th} column of B), which gives then

$$\max \sum_{j=1}^{k} f_j^T y_j \quad \text{s.t.} \quad (1, -b_i^T y_1, \ldots, -b_i^T y_k) \in \mathbb{L}^k \text{ for all } 1 \le i \le n$$

and finally

$$\max \sum_{j=1}^{k} f_j^T y_j \quad \text{s.t.} \quad \sum_{j=1}^{k} (b_i^T y_j)^2 \le 1 \text{ for all } 1 \le i \le n \,, \tag{TQC}$$

which is a convex quadratically constrained problem with a linear objective. We can thus solve a truss-topology design problem in at least three different manners: either solving the second-order cone optimization problems (TP) or (TD) or solving the quadratically constrained problem (TQC).

The problems we used for our computational experiments were randomly created using a generator developed by A. Nemirovski. Given three integers p, q and k, it produced the

Table 9.3: Dimensions of the truss-topology design problems.

Problem description	Formulation	Primal	Dual
2×2 grid with 2 loads	5 cones $\mathbb{L}^2$	15×8	23×15
2×2 grid with 4 loads	5 cones $\mathbb{L}^4$	25×16	41×25
2×2 grid with 8 loads	5 cones $\mathbb{L}^8$	45×32	77×45
2×2 grid with 16 loads	5 cones $\mathbb{L}^{16}$	85×64	149×85
2×2 grid with 32 loads	5 cones $\mathbb{L}^{32}$	165×128	293×165
4×4 grid with 2 loads	114 cones $\mathbb{L}^2$	342×48	390×342
4×4 grid with 4 loads	114 cones $\mathbb{L}^4$	570×96	666×570
4×4 grid with 6 loads	114 cones $\mathbb{L}^6$	798×144	942×798
6×6 grid with 2 loads	615 cones $\mathbb{L}^2$	1845×120	1965×1845
6×6 grid with 4 loads	615 cones $\mathbb{L}^4$	3075×240	3315×3075
8×8 grid with 2 loads	1988 cones $\mathbb{L}^2$	5964×224	6188×5964

matrix B and vectors f_j corresponding to k loading scenarios for a truss using a 2-dimensional $p \times q$ nodal grid, with $n \approx \frac{1}{2}p^2q^2$ and $m \approx 2pq$. We tested 11 combinations of parameters p, q and k. The dimensions of the corresponding problems are reported in Table 9.3 (the last two columns report the number of variables $\times$ the number of constraints). We see that the last problems involve a fairly large number of small second-order cones.

Polyhedral approximations of these problems were computed for three different accuracies, namely $\epsilon = 10^{-2}, 10^{-5}$ and 10^{-8}. The dimensions of the resulting linear optimization problems are reported in Table 9.4. It is interesting to note that problems with accuracy 10^{-8} are only approximately three times larger than problems with accuracy 10^{-2} and 50% larger than problems with accuracy 10^{-5} (with several dozens of thousands of variables for the largest among them).

Before we present computing times, we have to mention a special feature of this class of problems. Contrary to the general assertion that is stated in Section 9.2.9, it is possible to give here an estimation of the quality of the optimum objective value of the approximated problem. Indeed, let us call t^* the optimal objective value of problem (TTD) and t^*_ϵ the optimal objective value of the approximated problem with accuracy ϵ. Since our approximation is a relaxation, we obviously have $t^*_\epsilon \leq t^*$, and the optimal solution of the approximated problem $(Q^*_\epsilon, \sigma^*_\epsilon)$ is not necessarily feasible for the original problem. However, Definition 9.2 of a ϵ-approximation of a second-order cone implies in our case that

$$\left\|(q^*_{\epsilon,i1}, \ldots, q^*_{\epsilon,ik})\right\| \leq (1+\epsilon)\sigma^*_{\epsilon,i} \ \forall 1 \leq i \leq n \ ,$$

which means that $(Q^*_\epsilon, (1+\epsilon)\sigma^*_\epsilon)$ is feasible for the original problem, with an objective value equal to $(1+\epsilon)t^*_\epsilon$. Since we must have then $t^* \leq (1+\epsilon)t^*_\epsilon$, we conclude that

$$\frac{t^*}{1+\epsilon} \leq t^*_\epsilon \leq t^* \Leftrightarrow 0 \leq \frac{t^* - t^*_\epsilon}{t^*} \leq \frac{\epsilon}{1+\epsilon} \ ,$$

i.e. we have a bound on the relative accuracy of our approximated optimum objective value.

Table 9.4: Dimensions of the approximated problems (primal above, dual below).

$p \times q \times k$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-8}$
$2 \times 2 \times 2$	35×58	60×108	85×158
$2 \times 2 \times 4$	85×146	160×296	235×446
$2 \times 2 \times 8$	200×352	370×692	545×1042
$2 \times 2 \times 16$	420×744	795×1494	1165×2234
$2 \times 2 \times 32$	860×1528	1635×3078	2410×4628
$4 \times 4 \times 2$	798×1188	1368×2328	1938×3468
$4 \times 4 \times 4$	1938×3060	3648×6480	5358×9900
$4 \times 4 \times 6$	3306×5388	6156×11088	9006×16788
$6 \times 6 \times 2$	4305×6270	7380×12420	10455×18570
$6 \times 6 \times 4$	10455×16230	19680×34680	28905×53130
$8 \times 8 \times 2$	13916×20104	23856×39984	33796×59864

$p \times q \times k$	$\epsilon = 10^{-2}$	$\epsilon = 10^{-5}$	$\epsilon = 10^{-8}$
$2 \times 2 \times 2$	43×65	68×115	93×165
$2 \times 2 \times 4$	101×155	176×305	251×455
$2 \times 2 \times 8$	232×365	402×705	577×1055
$2 \times 2 \times 16$	484×765	859×1515	1229×2255
$2 \times 2 \times 32$	988×1565	1763×3115	2538×4665
$4 \times 4 \times 2$	846×1482	1416×2622	1986×3762
$4 \times 4 \times 4$	2034×3534	3744×6954	5454×10374
$4 \times 4 \times 6$	3450×6042	6300×11742	9150×17442
$6 \times 6 \times 2$	4425×7995	7500×14145	10575×20295
$6 \times 6 \times 4$	10695×19065	19920×37515	29145×55965
$8 \times 8 \times 2$	14140×25844	24080×45724	34020×65604

Table 9.5: Computing times to solve truss-topology problems using different approaches.

	QCO	SOCO			10^{-2}		10^{-5}		10^{-8}	
$p \times q \times k$	(D)	(P)	(D)	(D')	(P)	(D)	(P)	(D)	(P)	(D)
$2 \times 2 \times 2$	0.00	0.00	0.00	0.58	0.01	0.00	0.01	0.02	0.04	0.03
$2 \times 2 \times 4$	0.01	0.01	0.00	0.10	0.02	0.02	0.05	0.05	0.11	0.12
$2 \times 2 \times 8$	0.01	0.01	0.01	0.12	0.05	0.05	0.14	0.14	0.33	0.34
$2 \times 2 \times 16$	0.01	0.02	0.01	0.19	0.10	0.11	0.37	0.37	0.85	0.83
$2 \times 2 \times 32$	0.02	0.03	0.03	0.35	0.26	0.27	0.87	0.90	1.96	1.86
$4 \times 4 \times 2$	0.09	0.06	0.29	0.60	0.30	0.21	0.73	0.69	1.54	1.52
$4 \times 4 \times 4$	0.11	0.13	0.74	1.74	1.52	0.92	3.19	2.86	9.30	5.61
$4 \times 4 \times 6$	0.20	0.26	2.22	5.03	3.86	2.09	7.98	5.50	13.50	10.84
$6 \times 6 \times 2$	0.47	0.61	1.96	30.13	2.54	1.88	21.59	5.16	34.95	10.45
$6 \times 6 \times 4$	1.28	1.32	6.48	475.73	19.08	7.82	40.80	23.89	396.04	43.43
$8 \times 8 \times 2$	4.30	2.76	11.81	339.66	12.08	8.89	53.29	24.55	127.68	48.42

We generated three random problems for each of the 11 combinations of parameters (p, q, k) presented in Table 9.3, and report in Table 9.5 the average computing time in seconds. Each column in this table corresponds to a different way to solve the truss-topology design problem:

a. the first column reports computing times using MOSEK on the quadratically constrained formulation (TQC),

b. the following three columns report computing times using MOSEK on the primal and the dual second-order cone formulations (TP)–(TD) in columns (P) and (D), as well as the results of SeDuMi on the dual formulation in column (D').

c. the last six columns report computing times using the interior-point code in MOSEK to solve the polyhedral approximations of the primal and dual second-order cone problems (TP)–(TD) with three different accuracies.

Our first constatation is that solving the quadratically constrained formulation (TQC) and the primal second-order cone formulation (TP) directly are the two fastest methods (with similar computing times). The quadratically constrained formulation has less variables and constraints, but this advantage seems to be counterbalanced by a more efficient second-order cone solver.

Solving the dual second-order cone formulation (TP) directly is also very fast with a 2×2 nodal grid but noticeably slower on the larger problems (3 to 8 times slower). This is most probably due to the greater dimensions of the problem. The SeDuMi solver is much less efficient to solve these dual problems, and is really slow on the three largest problems.

Let us now look at the approximated problems. First of all, we checked whether the accuracy of the optimum approximated objective was below the theoretical bound $\frac{\epsilon}{1+\epsilon}$, since rounding errors in the computations and handling of irrational coefficients could affect this

result. Unsurprisingly, the accuracy was below the theoretical threshold for all experiments. Computing times are worse than for the direct approaches, even with the low accuracy 10^{-2}. The difference grows up to one or two orders of magnitude for the larger problems.

We also observe that despite slightly greater dimensions, solving the approximated dual problem is more efficient than solving the primal problem, especially with the largest problems. The reasons for this behaviour, which is opposite to the situation for direct resolutions, are unclear to us, but could be related to sparsity issues.

Finally, let us mention that we also tried to solve the linear approximations using the simplex algorithm instead of an interior-point method. This lead to surprisingly bad computing times: for example, solving problem $4 \times 4 \times 4$ with accuracy 10^{-2} using the MOSEK[7] simplex code took 21.57 seconds, instead of 1.52 seconds with the interior-point algorithm. We believe this disastrous behaviour of the simplex algorithm is due to the presence of an exponential number of vertices in the approximation, which leads to very slow progress.

9.3.3 Quadratic optimization

Second-order cone formulations of truss-topology design problems feature a relatively large number of small cones. Since our approximation procedure has not proven to be more efficient than direct methods on these problems, we would like to turn our attention to the opposite configuration, i.e. a small number of large cones. We are going to show that convex quadratic optimization can be formulated such as to meet this requirement.

More specifically, we are going to consider linearly constrained convex quadric optimization problems. Such problems can be formulated as

$$\min \ \frac{1}{2}x^T Q x + c^T x + c_0 \quad \text{s.t.} \quad l_c \leq Ax \leq u_c \text{ and } l_x \leq x \leq u_x \ , \tag{QO}$$

where $x \in \mathbb{R}^n$ denotes the vector of design variable. The objective is defined by a matrix $Q \in \mathbb{R}^{n \times n}$, required to be positive semidefinite to ensure convexity of the problem, a vector $c \in \mathbb{R}^n$ and a scalar $c_0 \in \mathbb{R}$. Variables are bounded by two vectors $l_x \in \mathbb{R}^n$ and $u_x \in \mathbb{R}^n$ (note that some components of l_x or u_x can be equal to $-\infty$ or $+\infty$ if a variable has no lower or upper bound). Finally, the linear constraints are described by a matrix $A \in \mathbb{R}^{m \times n}$ and two vectors $l_c \in \mathbb{R}^m$ and $u_c \in \mathbb{R}^m$ (with the same remark holding about possible infinite values for some components of l_c and u_c).

In order to model problem (QO) with a second-order cone formulation, we first write the Cholevsky factorization of matrix Q. Indeed, we have $Q = L^T L$ with $L \in \mathbb{R}^{k \times n}$, where $k \leq n$ is the rank of Q. Introducing a vector of auxiliary variables $z \in \mathbb{R}^k$ such that $z = Lx$, we have that $x^T Q x = x^T L^T L x = (Lx)^T Lx = z^T z$, which allows us to write the following problem:

$$\min \frac{r+v}{2} + c^T x + c_0 \quad \text{s.t.} \quad (r, v, z) \in \mathbb{L}^{k+1},\ r - v = 1,\ l_c \leq Ax \leq u_c \text{ and } l_x \leq x \leq u_x \ . \tag{QO'}$$

[7] In order to make sure that this behaviour was not caused by a flaw in the MOSEK simplex solver, we performed a similar comparison with the CPLEX solver, which lead to the same conclusion.

It is indeed equivalent to (QO), since the conic constraint $(r, v, z) \in \mathbb{L}^{k+1}$ combined with the equality $r - v = 1$ leads to

$$v^2 + \sum_{i=1}^{k} z_i^2 \leq r^2 \Leftrightarrow \sum_{i=1}^{k} z_i^2 \leq r^2 - v^2 \Leftrightarrow z^T z \leq (r - v)(r + v) \Leftrightarrow z^T z \leq r + v \ ,$$

which is why the quadratic term $\frac{1}{2} x^T Q x$ in the objective of (QO) could be replaced by $\frac{r+v}{2}$ in (QO').

The problems we tested come from the convex quadratic optimization library QPDATA, collected by Maros and Mészáros [MM99]. As for our tests with truss-topology design problems, we decided to formulate approximations with three different accuracies $10^{-2}, 10^{-5}$ and 10^{-8}. Table 9.6 lists for each problem its original size (variables $\times$ constraints), the number of nonzero elements in the constraint matrix A, the number of nonzero elements in the upper triangular part of Q and the size (variables $\times$ constraints) of each of the three polyhedral approximations.

Table 9.7 reports computing times (in seconds) needed to solve these convex quadratic optimization problems in three different ways:

a. the first column reports computing times using MOSEK directly on the original quadratic formulation (QO),

b. the following two columns report computing times needed to solve the second-order cone formulation (QO') of these problems with MOSEK and SeDuMi (in columns labelled (SOCO) and (SOCO') respectively),

c. the last three columns report computing times using MOSEK to solve the polyhedral approximations of the second-order cone problem (QO') with three different accuracies.

Once again, the direct approach, i.e. solving (QO), is the most efficient method. Solving these problems with a second-order cone formulation is slower, especially on larger problems. Using SeDuMi instead of MOSEK degrades further the computing times.

It is also manifest that the linear approximations take more time than the direct approach to provide a solution, even with the lowest accuracy 10^{-2} (however, we note that this low accuracy approximation is faster than the SeDuMi resolution on a few problems).

Although we only tested small-scale and medium-scale problems, it is pretty clear from the trend present for the last problems that large-scale problems would also be most efficiently solved directly as convex quadratic optimization problems.

We pointed out in Section 9.2.9 that our bound on the accuracy of the polyhedral approximation did not imply anything on the quality of the optimum of the approximated problems. Indeed, Table 9.8 reports the relative accuracy for a few representative approximated problems. Some problems (`GENHS28`, `DUALC5`) behave very well, with a relative accuracy well below the target accuracy. Other problems (`GOULDQP2`, `MOSARQP1`) have higher relative accuracies, but still decreasing when the target accuracy is decreased. Problem `CVXQP3S` shows a worse

Table 9.6: Statistics for the convex quadratic optimization problems.

Name	Size	ANZ	QNZ	Size 10^{-2}	Size 10^{-5}	Size 10^{-8}
TAME	2×1	2	3	9×13	14×23	19×33
HS21	2×1	2	2	14×22	24×42	34×62
ZECEVIC2	2×2	4	1	9×14	14×24	19×34
HS35	3×1	3	5	20×31	35×61	50×91
HS35MOD	3×1	3	5	20×31	35×61	50×91
HS52	5×3	7	7	29×46	49×86	69×126
HS76	4×3	10	6	28×46	48×86	68×126
HS51	5×3	7	7	29×46	49×86	69×126
HS53	5×3	7	7	29×46	49×86	69×126
S268	5×5	25	15	34×57	59×107	84×157
HS268	5×5	25	15	34×57	59×107	84×157
GENHS28	10×8	24	19	61×100	106×190	151×280
LOTSCHD	12×7	54	6	47×70	76×128	106×188
HS118	15×17	39	15	99×169	174×319	248×467
QPCBLEND	83×74	491	83	545×914	959×1742	1373×2570
CVXQP2S	100×25	74	386	647×1020	1135×1996	1624×2974
CVXQP1S	100×50	148	386	647×1045	1135×2021	1624×2999
CVXQP3S	100×75	222	386	647×1070	1135×2046	1624×3024
QPCBOEI2	143×166	1196	143	939×1614	1652×3040	2366×4468
DUALC5	8×278	2224	36	54×61	94×441	134×521
PRIMALC1	230×9	2070	229	1505×2329	2646×4611	3789×6897
PRIMALC5	287×8	2296	286	1878×2903	3304×5755	4732×8611
DUAL4	75×1	75	2799	493×761	867×1509	1241×2257
GOULDQP2	699×349	1047	697	2635×3872	4370×7342	6107×10816
DUAL1	85×1	85	3558	558×861	982×1709	1406×2557
PRIMALC8	520×8	4160	519	3410×5268	5996×10440	8586×15620
GOULDQP3	699×349	1047	1395	4584×7420	8062×14376	11546×21344
DUAL2	96×1	96	4508	632×976	1110×1932	1589×2890
MOSARQP2	900×600	2390	945	5911×9721	10395×18689	14887×27673

Table 9.7: Computing times to solve convex quadratic optimization problems

Name	QO	SOCO	SOCO'	10^{-2}	10^{-5}	10^{-8}
TAME	0.06	0.11	0.46	0.01	0.01	0.02
HS21	0.00	0.01	0.17	0.01	0.01	0.02
ZECEVIC2	0.01	0.00	0.101	0.00	0.00	0.03
HS35	0.00	0.00	0.1	0.01	0.01	0.02
HS35MOD	0.00	0.03	0.361	0.01	0.01	0.02
HS52	0.00	0.01	0.09	0.01	0.02	0.03
HS76	0.00	0.00	0.11	0.00	0.02	0.04
HS51	0.00	0.01	0.08	0.00	0.02	0.04
HS53	0.01	0.00	0.09	0.00	0.02	0.03
S268	0.02	0.01	0.20	0.01	0.04	0.07
HS268	0.01	0.00	0.20	0.01	0.04	0.07
GENHS28	0.00	0.01	0.08	0.01	0.03	0.07
LOTSCHD	0.02	0.01	0.16	0.02	0.03	0.08
HS118	0.00	0.01	0.25	0.05	0.11	0.17
QPCBLEND	0.03	0.12	0.83	0.41	1.43	2.75
CVXQP2S	0.02	0.23	1.34	0.58	1.91	3.58
CVXQP1S	0.03	0.14	1.55	0.57	1.88	4.47
CVXQP3S	0.05	0.21	1.86	0.60	2.32	3.93
QPCBOEI2	0.08	0.59	4.01	1.26	5.53	11.55
DUALC5	0.01	0.01	1.56	0.04	0.04	0.09
PRIMALC1	0.04	0.91	2.17	1.27	7.93	17.85
PRIMALC5	0.03	0.83	0.69	3.15	9.49	20.83
DUAL4	0.04	0.13	0.57	0.39	0.83	1.42
GOULDQP2	1.21	0.41	1.87	3.54	10.14	20.65
DUAL1	0.06	0.18	0.72	0.72	1.66	2.53
PRIMALC8	0.08	4.72	4.32	3.97	30.32	50.99
GOULDQP3	0.98	12.16	6.32	9.16	38.44	74.15
DUAL2	0.07	0.16	0.93	0.88	2.03	3.03
MOSARQP2	0.38	1.64	8.512	7.69	39.03	76.59

Table 9.8: Relative accuracy of the optimum of some approximated problems.

Name	10^{-2}	10^{-5}	10^{-8}
GENHS28	5.2e-4	8.8e-7	7.6e-10
HS21	3.3e-10	3.3e-10	3.3e-10
CVXQP3S	6.7e-3	7.2e-4	4.7e-4
DUALC5	3.9e-3	7.2e-7	3e-10
GOULDQP2	5.2e-1	5.6e-3	2.1e-5
MOSARQP2	8.6e-1	2.5e-4	8.5e-7

behaviour, with virtually no improvement between the second and the third approximation. Finally, HS21 is a toy problem with the surprising property that its approximation is exact for any accuracy.

We were able to compute these relative accuracies because the true optimal objective values were known by other means. In a real-world situation where such a piece information would not be available, it would be still possible to estimate roughly this accuracy. Indeed, since our approximation is a relaxation, we have that the approximated optimal objective p_ϵ^* is lower than the true optimum p^*. On the other hand, the optimal solution x_ϵ^* of the approximation must be feasible, since it satisfies the linear constraints. Computing the objective function corresponding to this solution, i.e. letting $p_\epsilon'^* = \frac{1}{2} x_\epsilon^{*T} Q x_\epsilon^* + c^T x_\epsilon^* + c_0$, we have finally that $p_\epsilon^* \leq p^* \leq p_\epsilon'^*$, which allows us to estimate *a posteriori* the true optimum objective value.

However, in the special case where the objective is purely quadratic, i.e. when $c = 0$ and $c_0 = 0$, it is possible to slightly modify the formulation so that we have a bound on the accuracy of the objective[8]. Indeed, letting again $z = Lx$, we add this time the conic constraint $(r, z) \in \mathbb{L}^k$, which implies to $z^T z \leq r^2 \Leftrightarrow x^T Q x \leq r^2$. We can now choose r as our objective, which is equivalent to minimizing $\sqrt{x^T Q x}$ and is obviously the same thing as minimizing the true quadratic objective $\frac{1}{2} x^T Q x$. This leads to a situation that is very similar to the case of truss-topology design problems, and one can show without difficulties that this approximated problem provides an estimation of the true optimum with a relative accuracy equal to $(\frac{\epsilon}{1+\epsilon})^2$.

9.4 Concluding remarks

In this chapter, we presented a polyhedral approximation of the second-order cone originally developed by Ben-Tal and Nemirovski [BTN98]. Our presentation features several improvements, including smaller dimensions for the approximation, a more transparent proof of its correctness, complete developments valid for any size of the second-order cone (i.e. not limited to powers of two) and explicit constants in the derivation of a theoretical bound on the size of the approximation (Theorem 9.4).

[8] A similar improvement can be made in the case when c belongs to the column space of L^T, the Cholevsky factor of Q. However, we have been unable to generalize this construction to the case where c does not belong to this column space, e.g. for an objective equal to $x_1^2 + x_2$.

This scheme was implemented in MATLAB and optimized as much as possible. Indeed, we developed several approaches to reduce the size of the resulting linear problems (including pivoting out some variables and using dynamic programming to choose the best accuracies for each stage of the decomposition). Our experiments mainly showed that solving the original second-order cone problems or alternative equivalent formulations is more efficient than solving the linear approximations, even at low accuracies. On a side note, we noticed that these approximate problems are particularly difficult to solve with the simplex algorithm.

However, we would like to point out this approximating scheme can still prove very useful in certain well-defined circumstances, such as a situation where a user is equipped with a solver that is only able to solve linear optimization problems. In this case, this procedure provides him with an inexpensive and relatively straightforward way to test improved versions of his linear models that make use of second-order cones.

Moreover, we have to admit that we tested two very specific classes of second-order cone optimization problems for which either a simplified formulation or a well-understood dedicated algorithm was available. It might well be possible that this linearizing scheme becomes competitive for other types of difficult (i.e. that cannot be simplified and for which no dedicated solver is available) second-order cones optimization problems.

We would also like to insist on the fact that it is not possible to guarantee *a priori* the accuracy of a linear approximation of a general second-order optimization problem (see the example in Section 9.2.9). It is nevertheless possible to provide such a bound in some special cases (e.g. truss-topology design problems or convex quadratic optimization problems with a pure quadratic objective).

It is worth to point out that a straightforward modification of our polyhedral approximation of $\mathbb{L}^2$ can lead to a restriction instead of a relaxation of second-order cone optimization problems. This would then provide an upper bound instead of a lower bound on the true optimum objective value, and optimal solutions of the approximate problems would always be feasible for the original problem. However, this approach can be problematic in some cases since it might happen that the approximated problem is infeasible, even if the original problem admits some feasible solutions.

An interesting topic for further research is the generalization of the polyhedral approximation of $\mathbb{L}^2$ or, more precisely, of the unit ball $\mathcal{B}_2(1)$, to other convex sets. Indeed, finding a similar polyhedral approximation for a set like $\{(x_1, x_2) \in \mathbb{R}^2 \mid |x_1|^p + |x_2|^p \leq 1\}$ with $p > 1$, i.e. the unit ball for the p-norm, would lead to linearizing scheme for other classes of convex problems, such as l_p-norm optimization (see Chapter 4). However, it is unclear to us at this stage whether this goal is achievable or not, since the symmetry of the standard unit ball, which is not present for other norms, seems to play a great role in the construction of the approximation.

Part IV

Conclusions

Concluding remarks and future research directions

We give here some concluding remarks about the research presented in this thesis, highlighting our personal contributions and hinting at some possible directions for further research (we however refer the reader to the last section of each chapter for more detailed comments).

Interior-point methods

Chapters 1 and 2 presented a survey of interior-point methods for linear optimization and a self-contained overview of the theory of self-concordant functions for structured convex optimization. We contributed some new results in Chapter 2, namely the computation of the optimal complexity of the short-step method and the improvement of a very useful Lemma to prove self-concordancy. We also gave a detailed explanation of why the definition of self-concordancy that is most commonly used nowadays is the best possible.

A very promising research direction in this area consists in investigating other types of barriers functions that lead to polynomial-time algorithms for convex optimization, possibly using the single condition (2.18) instead of the two inequalities (2.2) and (2.3) that characterize a self-concordant function.

Conic duality

Chapter 3 presented the framework of conic optimization and the associated duality theory, which is heavily used in the rest of this thesis. The approach we take in Chapters 4–6 to study l_p-norm and geometric optimization and give simplified proofs of their duality properties is completely new. The corresponding convex cones $\mathcal{L}^p$, $\mathcal{G}^n$, $\mathcal{G}_2^n$ were to the best of our knowledge

never studied before.

Chapter 7 generalizes our conic formulations of geometric and l_p-norm optimization with the notion of separable cone and is the culminating point of our study of convex problems with a nonsymmetric dual. We believe that most of the structured convex optimization that one can encounter in practice can be formulated within this framework (with the notable exceptions of second-order cone and semidefinite optimization).

It is obvious that much more research has to be done in this area. First of all, it would be highly desirable to study the duality properties relating the primal-dual pair of separable problems (SP)–(SD). Proving weak duality and strong duality in the presence of a Slater point should be straightforward. Moreover, we believe the zero duality gap property can probably also be proved (possibly with some minor technical assumptions), because of the inherent separability that is present in the definition of the $\mathcal{K}^f$ cone (i.e. the fact that all the functions that are used within this definition are scalar functions).

Another promising approach consists in generalizing the self-concordant barrier we designed for the $\mathcal{L}^p$ cone to the whole class of separable cones $\mathcal{K}^f$ and implementing the corresponding interior-point algorithms. Based on the results of existing conic solvers for linear, second-order and semidefinite optimization, our feeling is that the conic approach could lead to significant improvements in computational efficiency over more traditional methods.

Approximations

Chapter 8 demonstrated that it is possible to approximate geometric optimization using l_p-norm optimization. Despite the large amount of similarities between these two problems that was noticed by several authors, it is to the best of our knowledge the first time that such a strong link between these two classes of problems is presented.

Finally, Chapter 9 described a linearizing scheme for second-order cone optimization first introduced in [BTN98]. Our presentation features several improvements over the original construction, such as smaller dimensions for the polyhedral approximation, a more transparent proof of its correctness, complete developments valid for any size of the second-order cone (i.e. not limited to powers of two) and explicit constants in the derivation of a theoretical bound on the size of the approximation. We also contributed a careful implementation of this procedure using the MATLAB programming environment.

Although the computational experiments we conducted tend to show that solving the approximated problems is not as efficient as solving directly the original problem, we would like to stress the nonintuitive fact, demonstrated in this chapter, that it is possible, albeit with a relative loss of efficiency, to solve second-order cone and quadratic optimization problems with a linear optimization solver. Another interesting topic for further research in this area would be to generalize the principle of this polyhedral approximation to other types of convex sets.

Part V

Appendices

APPENDIX A

An application to classification

We present here a summary of our research on the application of semidefinite optimization to classification, which was the topic of our master's thesis [Gli98b].

A.1 Introduction

Machine learning is a scientific discipline whose purpose is to design computer procedures that are able to perform classification tasks. For example, given a certain number of medical characteristics about a patient (e.g. age, weight, blood pressure), we would like to infer automatically whether he or she is healthy or not.

A special case of machine learning problem is the separation problem, which asks to find a way to classify patterns that are known to belong to different well-defined classes. This is equivalent to finding a procedure that is able to recognize to which class each pattern belongs. The obvious utility of such a procedure is its use on unknown patterns, in order to determine to which one of the classes they are most likely to belong.

In this chapter, we present a new approach for this question based on two fundamental ideas: use ellipsoids to perform the pattern separation and solve the resulting problems with semidefinite optimization.

A.2 Pattern separation

Let us suppose we are faced with a set of objects. Each of these objects is completely described by an n-dimensional vector. We call this vector a *pattern*. To each component in this vector corresponds in fact a numerical characteristic about the objects. We assume that the only knowledge we have about an object is its pattern vector.

Let us imagine there is a natural way to group those objects into c classes. The pattern separation problem is simply the problem of separating these classes, i.e. finding a partition of the whole pattern space $\mathbb{R}^n$ into c disjoint components such that the patterns associated to each class belong to the corresponding component of the partition.

The main use for such a partition is of course classification: suppose we have some well-known objects that we are able to group into classes and some other objects for which we don't know the correct class. Our classification process will take place as follows:

a. Separate the patterns of well-known objects. This is called the *learning* phase[1].

b. Use the partition found above to classify the unknown objects. This is called the *generalization* phase.

We might ask ourselves what is a good separation. A good algorithm should of course be able to separate correctly the well-known objects, but is only really useful if it classifies correctly the unknown patterns. The generalization capability is thus the ultimate criteria to judge a separation algorithm.

We list here a few examples of common classification tasks.

- ◇ Medical diagnosis. This is one of the most important applications. The pattern vectors represent various measures of a patient's condition (e.g. age, temperature, blood pressure, etc.). We want here to separate the class of ill people from the class of healthy people.
- ◇ Species identification. The pattern vector represent various characteristics (e.g. colour, dimensions) of a plant or animal. Our objective is to classify them into different species.
- ◇ Credit screening. A company is trying to evaluate applicants for a credit card. The pattern contains information about the customer (e.g. type of job, monthly income, owns a house) and the goal is to identify for which applicants it is financially safe to give a credit card.

Ellipsoid representation. The main idea of this chapter is to use *ellipsoids* to separate our classes. Assuming we want to separate two classes of patterns[2], this means that we would

[1] Some authors refer to it as *supervised* learning phase. In fact, one may want to separate patterns without knowing *a priori* the classes they belong to, which is then called *unsupervised* learning. This is in fact a *clustering* problem, completely different from ours, and won't be discussed further in this work.

[2] It is shown in [Gli98b] that we can restrict our attention to the problem of separating two classes without loss of generality.

like to compute a separating ellipsoid such that the points from one class belong to the interior of the ellipsoid while the points from the other class lie outside of this ellipsoid. Let us explain this idea with Figure A.1.

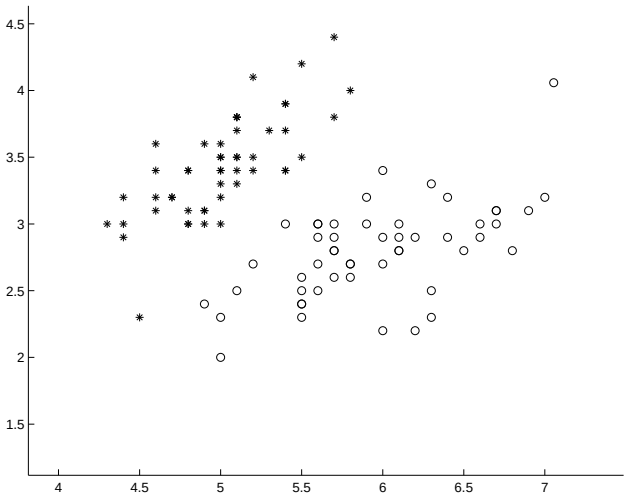


Figure A.1: A bidimensional separation problem.

This example is an easy bidimensional separation problem taken from a species classification data set (known as Fisher's Iris test set), using only the first two characteristics. The patterns from the first class appear as small circles, while the other class appear as small crosses. Computing a separating ellipsoid leads to the situation depicted on Figure A.2.

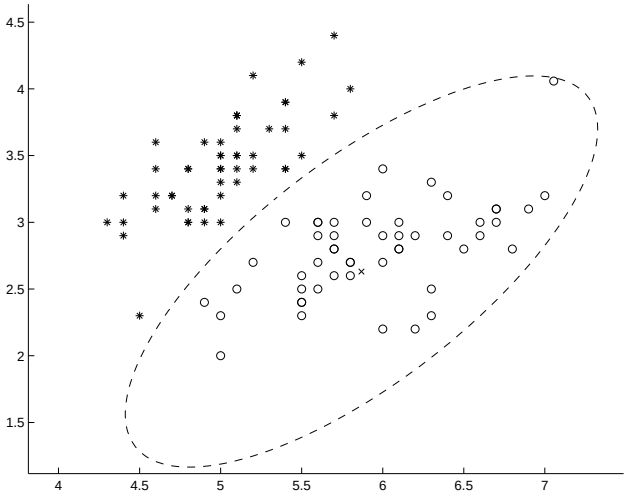


Figure A.2: A separating ellipsoid.

We decided to use ellipsoids for the following reasons:

- ◇ We expect patterns from the same class to be *close* to each other. This suggests enclosing them in some kind of hull, possibly a ball. But we also want our procedure to be scaling invariant. This is why we use the affine deformations of balls, which are the ellipsoids.
- ◇ Ellipsoids are the simplest convex sets (besides affine sets, which obviously do not fit our purpose).
- ◇ The set of points lying between two parallel hyperplanes is a (degenerate) ellipsoid. This means our separation procedures will generalize procedures that use a hyperplane to separate patterns.
- ◇ We know that some geometrical problems involving ellipsoids can be modelled using semidefinite optimization (this is due to the fact that an ellipsoid can be conveniently described using a positive semidefinite matrix).

Separating patterns. Our short presentation has avoided two difficulties that may arise with a pattern separation algorithm using ellipsoids, namely

a. Most of the time, the separating ellipsoid is not unique. How do we choose one ?
b. It may happen that there exists no separating ellipsoid.

Both of these issues can be addressed with the use optimization. Each ellipsoid is *a priori* a feasible solution. The objective function of our program will measure how well this ellipsoid separates our points. Ideally, non separating ellipsoids should have a high objective value (since we *minimize* our objective), while separating ellipsoids should have a lower objective value. With this kind of formulation, the conic program will always give us a solution, even when there is no separating ellipsoid.

We have thus to find an objective function that adequately represents the quality of the ellipsoid separation.

A.3 Maximizing the separation ratio

Let us consider the simple example depicted on Figure A.3: we want to include the small circles in an ellipsoid in order to obtain the best separation from the small crosses. A way to express this is to ask for two different separating ellipsoids. We want these ellipsoids to share the same center and axis directions (i.e. we want them to be geometrically similar), but the second one will be larger by a factor ρ, which we will subsequently call the *separation ratio*. Figure A.4 shows such a pair of ellipsoids with a ρ equal to $\frac{3}{2}$.

We now use the separation ratio to assess the quality of the separation: the higher the value of ρ, the better the separation. Our goal will be to maximize ρ over the set of separating ellipsoids. Figure A.5 shows the optimal pair of ellipsoids, with the maximal ρ equal to 1.863. However, we don't need two ellipsoids, so we finally partition the pattern

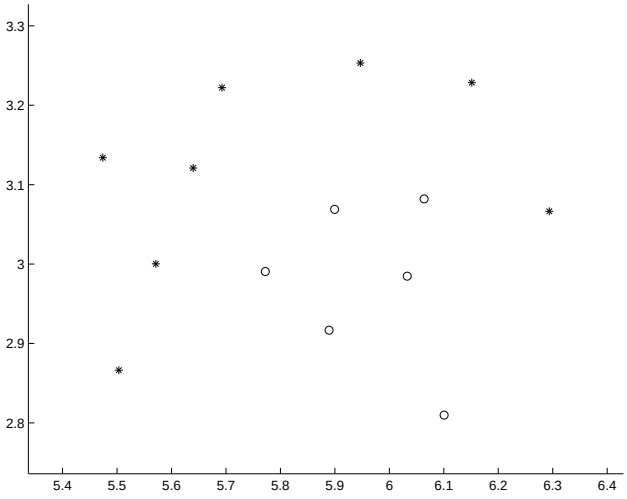


Figure A.3: A simple separation problem.

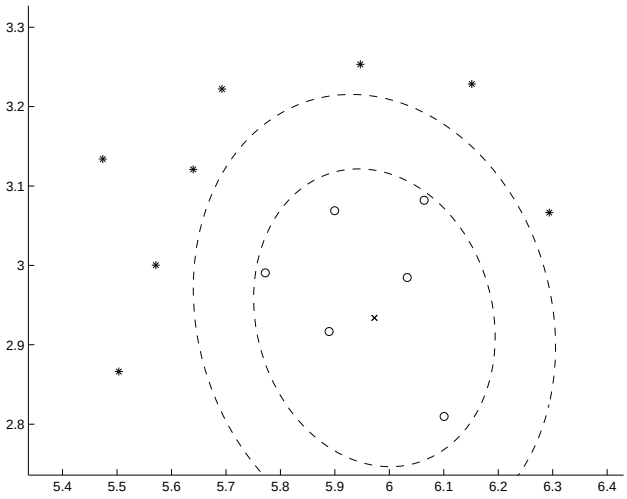


Figure A.4: A pair of ellipsoids with ρ equal to $\frac{3}{2}$.

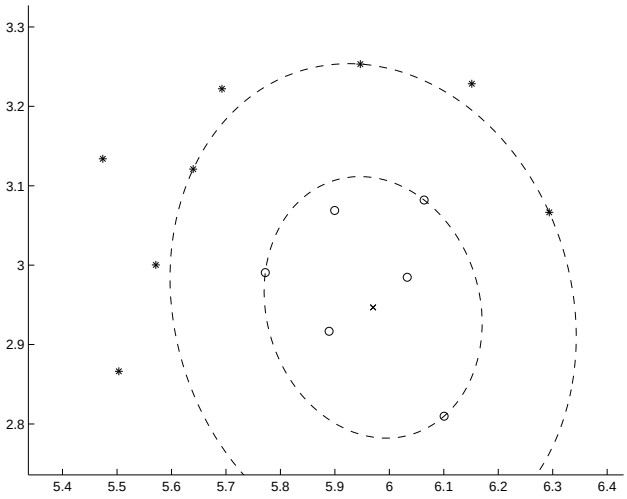


Figure A.5: The optimal pair of separating ellipsoids.

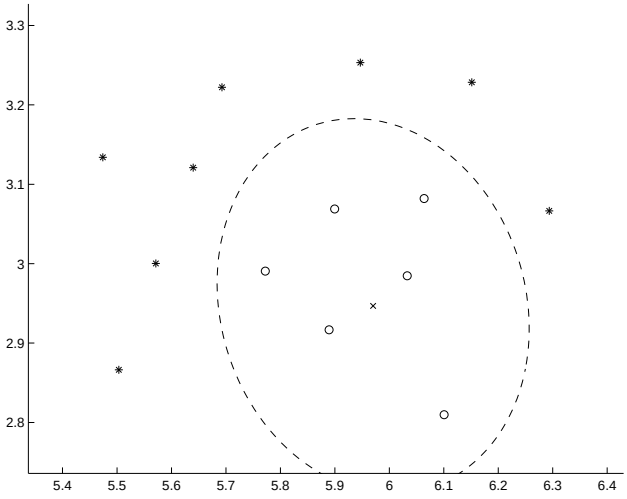


Figure A.6: The final separating ellipsoid.

space using an intermediate ellipsoid whose size is the mean size of our two ellipsoids, as depicted on Figure A.6.

It is possible to use semidefinite optimization to model the problem of finding the pair of ellipsoids with the best separation ratio. However, a straightforward formulation does not work because it leads to non-convex constraints and a technique of homogenization has to be introduced. We refer the reader to [Gli98b] for a thorough description of this formulation featuring the relevant mathematical details.

A.4 Concluding remarks

We have sketched in this chapter the principles of pattern separation using ellipsoids. It is obviously possible to enhance the basic method we presented in several different ways (for example to handle the case where the patterns cannot be completely separated by an ellipsoid). Three variants of this method are indeed described in [Gli98b] (*minimum volume* method, *maximum sum* method and *minimum squared sum* method).

We also refer the reader to [Gli98b] for the presentation and analysis of extensive computational results involving these methods on standard test sets. The main conclusion that can be drawn from this study is that these methods provide a viable way to classify patterns. As far as comparison with other classification procedures is concerned, it is fair to say that separating patterns using ellipsoids with semidefinite optimization occasionally delivers excellent results (significantly better than any other existing procedure) and gives competitive error rates on the majority of data sets.

To conclude this chapter, we mention that this approach has been recently applied to the problem of predicting the success or failure of students at their final exams.

Indeed, using only the results of preliminary tests carried out by first-year undergraduate students in late November, our separating ellipsoid is able to predict with a 11% error rate which students are going to pass and be allowed to enter the second-year, a decision which in fact depends on a series of exams that occur 2, 5 and even in some cases 7 months later (see the forthcoming report [DG00] for a complete description of these experiments).

APPENDIX B

Source code

We provide here the source code of the main routines used in the linearizing scheme for second-order cone optimization described in Chapter 9.

```
function Epsilon = Accuracy(Steps);
% Accuracy   Compute the accuracy of a polyhedral SOC approximation
%            Epsilon = Accuracy(Steps) returns the accuracy of a polyhedral
%            approximation of a second-order cone using a pyramidal
%            construction based on approximations of 3-dimensional SOC,
%            where Steps contains the number of steps used for the
%            approximation made at each level of the pyramidal construction.

Epsilon = 1/prod(cos(pi * (1/2) .^ Steps)) - 1;
```

```
function Levels = Levels(SizeCone)
% Levels    Computes the size of each level in the pyramidal approximation.
%           Levels = Levels(SizeCone) computes the number of cones
%           needed at each level in the pyramidal construction leading
%           to the polyhedral approximation of a second-order cone.
```

```
Levels = []; while SizeCone > 1
   Half = floor(SizeCone/2);
   Levels = [Levels Half];
   SizeCone = SizeCone - Half;
end
```

```
function theSteps = Steps(Levels, Epsilon, Method)
% Steps    Computes the number of steps for each level of the approximation
%          theSteps = Steps(Levels, Epsilon, Method) computes the number of
%          steps for each of the Levels in order to get accuracy equal to
%          Epsilon with the specified Method:
%            'AllEqual' -> number of steps is the same for each level
%            'Theory'   -> use formula with theoretical bound 'n log(1/e)'
%            'Optimal'  -> compute lowest possible total number of steps

if nargin < 3
   Method = 'Optimal';
end switch Method case 'AllEqual'
   theSteps = ceil(log2(pi/acos((1+Epsilon)^(-1/length(Levels)))));
   if length(Levels) > 1
      D = Accuracy(theSteps);
      theSteps = [Steps(Levels(1:end-1), (Epsilon-D)/(1+D), 'AllEqual') theSteps];
   end
case 'Theory'
   theSteps = ceil(log2(sum(Levels)/Levels(end)*9/16*pi^2/log(1+Epsilon))/2);
   if length(Levels) > 1
      D = Accuracy(theSteps);
      theSteps = [Steps(Levels(1:end-1), (Epsilon-D)/(1+D), 'Theory') theSteps];
   end
case 'Optimal'
   if length(Levels) == 1
      theSteps = Steps(1, Epsilon, 'AllEqual');
   else
      AE = Steps(Levels, Epsilon, 'AllEqual');
      TH = Steps(Levels, Epsilon, 'Theory');
      UpperBound = floor(min(AE*Levels', TH*Levels')/sum(Levels));
      LowerBound = Steps(1, Epsilon, 'AllEqual');
      theSteps = []; BestSize = inf;
      index = LowerBound;
      while index <= UpperBound
         D = Accuracy(index);
         S = [index Steps(Levels(2:end), (Epsilon-D)/(1+D), 'Optimal')];
         if S*Levels' < BestSize
            theSteps = S;
            BestSize = theSteps*Levels';
            UpperBound = min(UpperBound, floor(BestSize/sum(Levels)));
         end
         index = index + 1;
```

```
     end
   end
otherwise
   error('Unknown method');
end
```

```
function resLP = PolySOC2(Steps, SkipSteps)
% PolySOC2  Computes a polyhedral approximation of the 3-dimensional Lorentz cone
%           resLP = PolySOC2(Steps, SkipSteps) computes a polyhedral approximation
%           of the 3-dimensional SOC using the Ben-Tal/Nemirovski construction with
%           a number of steps equal to Steps. A number of the first steps of the
%           construction can be skipped using the optional parameter SkipSteps.
%           The resulting approximation will have:
%                 n+2 variables (i.e. n-1 additional variables),
%                 2n  inequality constraints,
%           where n is the total number of steps in the construction (i.e.
%           Steps-SkipSteps). There are also two global options available:
%             - useRestriction to use a restriction of the SOC instead of a
%               relaxation,
%             - doNotPivotOut to stop pivoting out variables from the equality
%               constraints, which gives n more variables and n equality
%               constraints but a more sparse constraint matrix.

% Global options
global doNotPivotOut useRestriction;
persistent PolySOC2Cache;
if nargin < 2
   SkipSteps = 0;
else
   Steps = Steps-SkipSteps;
end
if [Steps+1 SkipSteps+1] <= size(PolySOC2Cache) & ...
      ~isempty(PolySOC2Cache{Steps+1, SkipSteps+1})
   resLP = PolySOC2Cache{Steps+1, SkipSteps+1};
   return;
end
Angles = pi * (1/2).^(SkipSteps+(0:Steps))';
indexX = repmat([1 1 1 2 2 2 3 3 3], Steps, 1) + repmat((0:3:3*(Steps-1))', 1, 9);
indexY = repmat([1 2 3 1 2 4 1 2 4], Steps, 1) + repmat((1:2:2*(Steps)-1)', 1, 9);
indexVal = [ cos(Angles(1:end-1))  sin(Angles(1:end-1)) -ones(Steps, 1) ...
             sin(Angles(1:end-1)) -cos(Angles(1:end-1)) -ones(Steps, 1) ...
            -sin(Angles(1:end-1))  cos(Angles(1:end-1)) -ones(Steps, 1) ];
if ~isempty(useRestriction) & useRestriction
   rootCoef = cos(Angles(end));
else
   rootCoef = 1;
end
A = sparse([indexX(:) ; 3*(Steps) + [1;1;1]], ...
```

```
           [indexY(:) ; 2*(Steps) + [2;3] ; 1], ...
           [indexVal(:) ; cos(Angles(end)) ; sin(Angles(end)) ; -rootCoef]);
if isempty(doNotPivotOut) | ~doNotPivotOut
   for index = 1:Steps % alpha variables
      A = A + A(:, 3+index) * A(2*index-1, :);
      A(2*index-1, :) = [];
      A(:, 3+index) = [];
   end
   A = A - 1/A(end,end) * A(:, end) * A(end, :); % last beta_k variable
   A(end, :) = [];
   A(:, end) = [];
   resLP = lp([], A, [-inf*ones(2*Steps, 1)], zeros(2*(Steps), 1));
else
   resLP = lp([], A, [repmat([0 ; -inf ; -inf], Steps, 1) ; 0], ...
              zeros(3*(Steps)+1, 1));
end
PolySOC2Cache{Steps+1, SkipSteps+1} = resLP;
```

```
function [resLP, theAccuracy, theSteps] = PolySOCN(SizeCone, Epsilon)
% PolySOC2N  Computes a polyhedral approximation of a second-order cone
%            [resLP, theAccuracy, theSteps] = PolySOCN(SizeCone, Epsilon) computes
%            a polyhedral approximation with accuracy Epsilon of SOC of dimension
%            SizeCone (not counting the root) using a pyramidal construction
%            involving (SizeCone-1) 3-dimensional SOC approximations. theAccuracy
%            will contain the resulting accuracy (smaller or equal to Epsilon)
%            while theSteps provides the number of steps used for the approximation
%            at each level of the pyramidal construction.

switch SizeCone
case 0             % Special case: linear program, not handled by this construction
   resLP = lp([], 1, 0);
   theAccuracy = 0;
   theSteps = [];
case 1             % Special case: linear program, not handled by this construction
   resLP = lp([], [1 -1;1 1], [0 0]');
   theAccuracy = 0;
   theSteps = [];
otherwise
   theLevels = Levels(SizeCone);
   theSteps = Steps(theLevels, Epsilon, 'Optimal');
   theAccuracy = Accuracy(theSteps);
   CurrentVars = 1+(1:SizeCone);
   resLP = lp([], zeros(0, SizeCone+1));
   index = 1;
   OddLeft = mod(SizeCone, 2);
   for index = 1:length(theLevels)
      if index == 1
         addLP = PolySOC2(theSteps(index));
```

```
         [addLP, baseVars, rootVars] = DupPolySOCN(addLP, 2, theLevels(index));
      elseif OddLeft & theLevels(index-1) ~= 2*theLevels(index)
         OddLeft = 0;
         addLP = PolySOC2(theSteps(index), 2);
         [addLP, baseVars, rootVars] = DupPolySOCN(addLP, 2, theLevels(index)-1);
         oddLP = PolySOC2(theSteps(index), 1);
         rootVars = [1 rootVars+dims(oddLP, 2)];
         baseVars = [2 3 baseVars+dims(oddLP, 2)];
         addLP = add(oddLP, addLP, []);
      else
         addLP = PolySOC2(theSteps(index), 2);
         [addLP, baseVars, rootVars] = DupPolySOCN(addLP, 2, theLevels(index));
      end
      reOrder = NaN*ones(1, dims(addLP, 2));
      reOrder(baseVars) = CurrentVars(end-theLevels(index)*2+1:end);
      CurrentVars = [CurrentVars(1:end-theLevels(index)*2) dims(resLP, 2) + ...
         rootVars-(0:2:2*theLevels(index)-2)];
      if index == length(theLevels)
         reOrder(1) = 1;
      end
      resLP = add(resLP, addLP, reOrder);
   end
end
```

```
function [resLP, baseVars, rootVars] = DupPolySOCN(theLP, SizeCone, N);
% DupPolySOCN  Concatenate polyhedral approximations of second-order cones.
%              [resLP, baseVars, rootVars] = DupPolySOCN(theLP, SizeCone, N)
%              computes a concatenation of N polyhedral approximations of
%              a SizeCone-dimensional second-order cone contained in theLP.
%              rootVars contains the indices of the N root cone variables, while
%              baseVars contains the indices of the N*SizeCone other cone variables.

if N == 0
   rootVars = [];
   baseVars = [];
   resLP = lp;
else
   rootVars = 1;
   baseVars = 2:SizeCone+1;
   resLP = theLP;
   nSteps = floor(log2(N));
   N = N - 2^nSteps;
   for index = nSteps-1:-1:0
      Delta = dims(resLP, 2);
      resLP = add(resLP, resLP, []);
      baseVars = [baseVars Delta+baseVars];
      rootVars = [rootVars Delta+rootVars];
      if N >= 2^index
```

```
         N = N - 2^index;
         Delta = dims(resLP, 2);
         resLP = add(resLP, theLP, []);
         baseVars = [baseVars Delta+(2:SizeCone+1)];
         rootVars = [rootVars Delta+1];
      end
   end
end

% Alternate recursive version :
% if N == 0
%    rootVars = [];
%    baseVars = [];
%    resLP = lp;
% elseif N == 1
%    rootVars = 1;
%    baseVars = 2:SizeCone+1;
%    resLP = theLP;
% elseif mod(N, 2) == 0
%    [resLP baseVars rootVars] = DupPolySOCN(theLP, SizeCone, N/2);
%    Delta = dims(resLP, 2);
%    resLP = add(resLP, resLP, []);
%    baseVars = [baseVars Delta+baseVars];
%    rootVars = [rootVars Delta+rootVars];
% else
%    [resLP baseVars rootVars] = DupPolySOCN(theLP, SizeCone, (N-1));
%    Delta = dims(resLP, 2);
%    resLP = add(resLP, theLP, []);
%    baseVars = [baseVars Delta+(2:SizeCone+1)];
%    rootVars = [rootVars Delta+1];
% end
```

```
function apxLP = PolySOCLP(theLP, coneInfo, Epsilon, printLevel)
%PolySOCLP   Computes a polyhedral approximation of a second-ordre cone program.
%            apxLP = PolySOCLP(theLP, coneInfo, Epsilon, printLevel) computes a
%            polyhedral approximation with accuracy Epsilon of the second-order
%            cone program described by theLP (objective and linear constraints)
%            and coneInfo (list of second-order cones).
%            Optional parameter printLevel = 0 => no output
%                                            1 => outputs a summary
%                                            2 => info for each cone (default)

if nargin < 4
   printLevel = 2;
end if printLevel
   disp(sprintf(['Approximating with %4.2g epsilon SOCP with %d cones, ' ...
                 '%d variables and %d constraints.'], Epsilon, ...
                 size(coneInfo, 2), dims(theLP, 2), dims(theLP, 1)));
```

```
end

maxEpsilon = inf;
apxLP = theLP;
for indexCone = 1:length(coneInfo)
   coneSize(indexCone) = length(coneInfo(indexCone).memb) - 1;
end
[Sorted Order] = sort([coneSize]);
indexCone = 1;
while indexCone <= length(coneInfo)
   [coneLP theEpsilon theSteps] = PolySOCN(Sorted(indexCone), Epsilon);
   if theEpsilon < maxEpsilon
      maxEpsilon = theEpsilon;
   end
   nCones = max(find(Sorted == Sorted(indexCone))) - indexCone + 1;
   if printLevel >= 2
      disp([sprintf(['-> %d SOC of dimension %d : %g epsilon ' ...
                     'with %d variables, %d constraints ('], nCones, ...
                     Sorted(indexCone), theEpsilon, dims(coneLP, 2), ...
                     dims(coneLP, 1)) mat2str(theSteps) ' steps).']);
   end
   [NconeLP, baseVars, rootVars] = DupPolySOCN(coneLP, Sorted(indexCone), nCones);
   theCones = [coneInfo(Order(indexCone:indexCone+nCones-1)).memb];
   reOrder = NaN*ones(1, max([baseVars rootVars]));
   reOrder(rootVars) = theCones(1, :);
   reOrder(baseVars) = theCones(2:end, :);
   apxLP = add(apxLP, NconeLP, reOrder);
   indexCone = indexCone + nCones;
end
if printLevel
   disp(sprintf(['Final approximation has %4.2g epsilon with %d variables ' ...
                 'and %d constraints.'], maxEpsilon, dims(apxLP, 2), ...
                 dims(apxLP, 1)));
end
```

Bibliography

[AA99] E. D. Andersen and K. D. Andersen, *The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm*, High Performance Optimization (H. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds.), Applied optimization, vol. 33, Kluwer Academic Publishers, 1999.

[AGMX96] E. D. Andersen, J. Gondzio, Cs. Mészáros, and X. Xu, *Implementation of interior-point methods for large scale linear programs*, Interior Point Methods of Mathematical Programming (T. Terlaky, ed.), Applied Optimization, vol. 5, Kluwer Academic Publishers, 1996, pp. 189–252.

[Ans90] K. M. Anstreicher, *On long step path following and SUMT for linear and quadratic programming*, Tech. report, Yale School of Management, Yale University, New Haven, CT, 1990.

[Ans96] ———, *Potential reduction algorithms*, Interior Point Methods of Mathematical Programming (T. Terlaky, ed.), Applied Optimization, vol. 5, Kluwer Academic Publishers, 1996, pp. 125–158.

[ART00] E. D. Andersen, C. Roos, and T. Terlaky, *On implementing a primal-dual interior-point method for conic quadratic optimization*, in preparation, 2000.

[Bri00] J. Brinkhuis, Communication at the International Symposium on Mathematical Programming, Atlanta, August 2000.

[BTN94] A. Ben-Tal and A. Nemirovski, *Potential reduction polynomial-time method for truss topology design*, SIAM Journal of Optimization **4** (1994), 596–612.

[BTN98] ———, *On polyhedral approximations of the second-order cone*, Tech. report, Minerva Optimization Center, Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa, Israel, 1998, to appear in Mathematics of Operations Research.

[Dan63] G. B. Dantzig, *Linear programming and extensions*, Princeton University Press, Princeton, N.J., 1963.

[DG00] B. Diricq and Fr. Glineur, *Prédire la réussite en première candidature en sciences appliquées : mathématiques ou médiumnité ?*, in preparation, 2000.

[Dik67] I. I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Doklady Akademii Nauk SSSR **174** (1967), 747–748.

[dJRT95] D. den Hertog, F. Jarre, C. Roos, and T. Terlaky, *A sufficient condition for self-concordance with application to some classes of structured convex programming problems*, Mathematical Programming, Series B **69** (1995), no. 1, 75–88.

[DPZ67] R. J. Duffin, E. L. Peterson, and C. Zener, *Geometric programming*, John Wiley & Sons, New York, 1967.

[dRT92] D. den Hertog, C. Roos, and T. Terlaky, *On the classical logarithmic barrier method for a class of smooth convex programming problems*, Journal of Optimization Theory and Applications **73** (1992), no. 1, 1–25.

[DS97] A. Dax and V. P. Sreedharan, *On theorems of the alternative and duality*, Journal of Optimization Theory and Applications **94** (1997), no. 3, 561–590.

[ET76] I. Ekeland and R. Temam, *Convex analysis and variational problems*, Studies in mathematics and its applications, vol. 1, North-Holland publishing company, Amsterdam, Oxford, 1976.

[FM68] A. V. Fiacco and G. P. McCormick, *Nonlinear programming: Sequential unconstrained minimization techniques*, John Wiley & Sons, New York, 1968, Reprinted in *SIAM Classics in Applied Mathematics*, SIAM Publications, 1990.

[Fri55] K. R. Frisch, *The logarithmic potential method of convex programming*, Tech. report, University Institute of Economics, Oslo, Norway, 1955.

[Gli97] Fr. Glineur, *Etude des méthodes de point intérieur appliquées à la programmation linéaire et à la programmatiuon semidéfinie*, Travail de fin d'études études, Faculté Polytechnique de Mons, Mons, Belgium, June 1997.

[Gli98a] ———, *Interior-point methods for linear programming: a guided tour*, Belgian Journal of Operations Research, Statistics and Computer Science **38** (1998), no. 1, 3–30.

[Gli98b] ———, *Pattern separation via ellipsoids and conic programming*, Mémoire de D.E.A., Faculté Polytechnique de Mons, Mons, Belgium, September 1998.

[Gli99] ———, *Proving strong duality for geometric optimization using a conic formulation*, IMAGE Technical Report 9903, Faculté Polytechnique de Mons, Mons, Belgium, October 1999, to appear in Annals of Operations Research.

[Gli00a] ———, *Approximating geometric optimization with l_p-norm optimization*, IMAGE Technical Report 0008, Faculté Polytechnique de Mons, Mons, Belgium, November 2000, submitted to Operations Research Letters.

[Gli00b] ———, *An extended conic formulation for geometric optimization*, IMAGE Technical Report 0006, Faculté Polytechnique de Mons, Mons, Belgium, May 2000, submitted to Foundations of Computing and Decision Sciences.

[Gli00c] ———, *Polyhedral approximation of the second-order cone: computational experiments*, IMAGE Technical Report 0001, Faculté Polytechnique de Mons, Mons, Belgium, January 2000, revised November 2000.

[Gli00d] ———, *Self-concordant functions in structured convex optimization*, IMAGE Technical Report 0007, Faculté Polytechnique de Mons, Mons, Belgium, October 2000, submitted to European Journal of Operations Research.

[GT56] A. J. Goldman and A. W. Tucker, *Theory of linear programming*, Linear Equalities and Related Systems (H. W. Kuhn and A. W. Tucker, eds.), Annals of Mathematical Studies, vol. 38, Princeton University Press, Princeton, New Jersey, 1956, pp. 53–97.

[GT00] Fr. Glineur and T. Terlaky, *A conic formulation for l_p-norm optimization*, IMAGE Technical Report 0005, Faculté Polytechnique de Mons, Mons, Belgium, May 2000, submitted to Journal of Optimization Theory and Applications.

[GW95] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, Journal of Association for Computing Machinery **42** (1995), no. 6, 1115–1145.

[HPY92] C. Han, P. Pardalos, and Y. Ye, *Implementation of interior-point algorithms for some entropy optimization problems*, Optimization Methods and Software **1** (1992), 71–80.

[Hua67] P. Huard, *Resolution of mathematical programming with nonlinear constraints by the method of centers*, Nonlinear Programming (J. Abadie, ed.), North Holland, Amsterdam, The Netherlands, 1967, pp. 207–219.

[HvM97] T. Terlaky H. van Maaren, *Inverse barriers and ces-functions in linear programming*, Operations Research Letters **20** (1997), 15–20.

[Jar89] F. Jarre, *The method of analytic centers for smooth convex programs*, Dissertation, Institut für Angewandte Mathematik und Statistik, Universität Wärzburg, Germany, 1989.

[Jar96] ———, *Interior-point methods for classes of convex programs*, Interior Point Methods of Mathematical Programming (T. Terlaky, ed.), Applied Optimization, vol. 5, Kluwer Academic Publishers, 1996, pp. 255–296.

[Kar84] N. K. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica **4** (1984), 373–395.

[Kha79] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Mathematics Doklady **20** (1979), 191–194.

[Kla74] E. Klafszky, *Geometric programming and some applications*, Ph.D. thesis, Tanulmányok, No. 8, 1974.

[Kla76] ———, *Geometric programming*, Seminar Notes, no. 11.976, Hungarian Committee for Systems Analysis, Budapest, 1976.

[KM72] V. Klee and G. J. Minty, *How good is the simplex algorithm ?*, Inequalities, O. Shisha ed., pp. 159–175, Academic Press, New York, 1972.

[LVBL98] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, *Applications of second-order cone programming*, Linear Algebra and its Applications **284** (1998), 193–228.

[Mas93] W. F. Mascarenhas, *The affine scaling algorithm fails for $\lambda = 0.999$*, Tech. report, Universidade Estadual de Campinas, Campinas S. P., Brazil, October 1993.

[Meh92] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization **2** (1992), 575–601.

[MM99] I. Maros and Cs. Mészáros, *A repository of convex quadratic programming problems*, Optimization Methods and Software **11-12** (1999), 671–681, special issue on interior-point methods (CD supplement with software), guest editors: Florian Potra, Cornelis Roos and Tamás Terlaky.

[Nes96] Y. Nesterov, *Nonlinear optimization*, Notes from a lecture given at CORE, UCL, Belgium, 1996.

[NN94] Y. E. Nesterov and A. S. Nemirovski, *Interior-point polynomial methods in convex programming*, SIAM Studies in Applied Mathematics, SIAM Publications, Philadelphia, 1994.

[PE67] E. L. Peterson and J. G. Ecker, *Geometric programming: Duality in quadratic programming and l_p approximation II*, SIAM Journal on Applied Mathematics **13** (1967), 317–340.

[PE70a] ———, *Geometric programming: Duality in quadratic programming and l_p approximation I*, Proceedings of the International Symposium of Mathematical Programming (Princeton, New Jersey) (H. W. Kuhn and A. W. Tucker, eds.), Princeton University Press, 1970.

[PE70b] ———, *Geometric programming: Duality in quadratic programming and l_p approximation III*, Journal on Mathematical Analysis and Applications **29** (1970), 365–383.

[PRT00] J. Peng, C. Roos, and T. Terlaky, *Self-regular proximities and new search directions for linear and semidefinite optimization*, Technical report, Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada, March 2000, submitted to Mathematical Programming.

[PY93] F. Potra and Y. Ye, *A quadratically convergent polynomial interior-point algorithm for solving entropy optimization problems*, SIAM Journal on Optimization **3** (1993), 843–860.

[Ren00] J. Renegar, *A mathematical view of interior-point methods in convex optimization*, to be published by in the MPS/SIAM Series on Optimization, SIAM, New York, 2000.

[Roc70a] R. T. Rockafellar, *Convex analysis*, Princeton University Press, Princeton, N. J., 1970.

[Roc70b] ———, *Some convex programs whose duals are linearly constrained*, Non-linear Programming (J. B. Rosen, ed.), Academic Press, 1970.

[RT98] C. Roos and T. Terlaky, *Nonlinear optimization*, Delft University of Technology, The Netherlands, 1998, Course WI387.

[RTV97] C. Roos, T. Terlaky, and J.-Ph. Vial, *Theory and algorithms for linear optimization. an interior point approach*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Chichester, UK, 1997.

[Sat75] K. Sato, *Production functions and aggregation*, North-Holland, Amsterdam, 1975.

[Sch86] A. Schrijver, *Theory of linear and integer programming*, Wiley-Interscience series in discrete mathematics, John Wiley & sons, 1986.

[Sho70] N. Z. Shor, *Utilization of the operation of space dilatation in the minimization of convex functions*, Kibernetika **1** (1970), 6–12.

[Stu97] J. F. Sturm, *Primal-dual interior-point approach to semidefinite programming*, Ph.D. thesis, Erasmus Universiteit Rotterdam, The Netherlands, 1997, published in [Stu99a].

[Stu99a] ———, *Duality results*, High Performance Optimization (H. Frenk, C. Roos, T. Terlaky, and S. Zhang, eds.), Applied optimization, vol. 33, Kluwer Academic Publishers, 1999, pp. 21–60.

[Stu99b] ———, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software **11-12** (1999), 625–653, special issue on interior-point methods (CD supplement with software), guest editors: Florian Potra, Cornelis Roos and Tamás Terlaky.

[SW70] J. Stoer and Ch. Witzgall, *Convexity and optimization in finite dimensions I*, Springer Verlag, Berlin, 1970.

[Ter85] T. Terlaky, *On l_p programming*, European Journal of Operations Research **22** (1985), 70–100.

[VB96] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review **38** (1996), 49–95.

[Wri97] S. J. Wright, *Primal-dual interior-point methods*, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[XY00] G. Xue and Y. Ye, *An efficient algorithm for minimizing a sum of p-norms*, SIAM Journal on Optimization **10** (2000), no. 2, 551–579.

[Ye97] Y. Ye, *Interior point algorithms, theory and analysis*, John Wiley & Sons, Chichester, UK, 1997.

[YTM94] Y. Ye, M. J. Todd, and S. Mizuno, *An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm*, Mathematics of Operations Research **19** (1994), 53–67.

Summary

Optimization is a scientific discipline that lies at the boundary between pure and applied mathematics. Indeed, while on the one hand some of its developments involve rather theoretical concepts, its most successful algorithms are on the other hand heavily used by numerous companies to solve scheduling and design problems on a daily basis.

Our research started with the study of the *conic formulation* for convex optimization problems. This approach was already studied in the seventies but has recently gained a lot of interest due to development of a new class of algorithms called *interior-point methods*. This setting is able to exploit the two most important characteristics of convexity:

- ◇ a very rich duality theory (existence of a dual problem that is strongly related to the primal problem, with a very symmetric formulation),
- ◇ the ability to solve these problems efficiently, both from the theoretical (polynomial algorithmic complexity) and practical (implementations allowing the resolution of large-scale problems) point of views.

Most of the research in this area involved so-called *self-dual cones*, where the dual problem has exactly the same structure as the primal: the most famous classes of convex optimization problems (linear optimization, convex quadratic optimization and semidefinite optimization) belong to this category. We brought some contributions in this field:

- ◇ a survey of interior-point methods for linear optimization, with an emphasis on the fundamental principles that lie behind the design of these algorithms,
- ◇ a computational study of a method of linear approximation of convex quadratic optimization (more precisely, the second-order cone that can be used in the formulation of

quadratic problems is replaced by a polyhedral approximation whose accuracy that can be guaranteed a priori),

- ◇ an application of semidefinite optimization to classification, whose principle consists in separating different classes of patterns using ellipsoids defined in the feature space (this approach was successfully applied to the prediction of student grades).

However, our research focussed on a much less studied category of convex problems which does not rely on self-dual cones, i.e. structured problems whose dual is formulated very differently from the primal. We studied in particular

- ◇ *geometric optimization*, developed in the late sixties, which possesses numerous application in the field of engineering (entropy optimization, used in information theory, also belongs to this class of problems)
- ◇ *l_p-norm optimization*, a generalization of linear and convex quadratic optimization, which allows the formulation of constraints built around expressions of the form $|ax + b|^p$ (where p is a fixed exponent strictly greater than 1).

For each of these classes of problems, we introduced a new type of convex cone that made their formulation as standard conic problems possible. This allowed us to derive very simplified proofs of the classical duality results pertaining to these problems, notably weak duality (a mere consequence of convexity) and the absence of a duality gap (strong duality property without any constraint qualification, which does not hold in the general convex case). We also uncovered a very surprising result that stipulates that geometric optimization can be viewed as a limit case of l_p-norm optimization. Encouraged by the similarities we observed, we developed a general framework that encompasses these two classes of problems and unifies all the previously obtained conic formulations.

We also brought our attention to the design of interior-point methods to solve these problems. The theory of polynomial algorithms for convex optimization developed by Nesterov and Nemirovsky asserts that the main ingredient for these methods is a computable *self-concordant* barrier function for the corresponding cones. We were able to define such a barrier function in the case of l_p-norm optimization (whose parameter, which is the main determining factor in the algorithmic complexity of the method, is proportional to the number of variables in the formulation and independent from p) as well as in the case of the general framework mentioned above.

Finally, we contributed a survey of the self-concordancy property, improving some useful results about the value of the complexity parameter for certain categories of barrier functions and providing some insight on the reason why the most commonly adopted definition for self-concordant functions is the best possible.

About the cover

The drawing depicted on the cover and the variant that is presented on the next page are meant to illustrate some of the topics presented in this thesis, namely the fundamental notions of central path and barrier function for interior-point methods, as well as the existence of multiple types of convex constraints. Each of the small frames represents a convex optimization problem involving two variables (x, y) and the following four constraints:

a. a first linear constraint $5y - 0.9x \leq 4.5$, which defines the upper left boundary of the feasible zone,

b. a second hyperbolic constraint $32xy \geq 1$, which can be modelled as a second-order cone constraint (see Example 3.1 in Chapter 3) and is responsible for the lower left boundary of the feasible region,

c. a third l_p-norm constraint $|x|^{3/2} + |y|^{3/2} \leq 0.9$ (see Chapter 4), which defines the lower right boundary of the feasible set,

d. and finally a fourth geometric constraint $e^x + e^y \leq 4.15$ (see Chapter 5) to determine the shape of the upper right boundary of the feasible area.

Although they share the same feasible region, the different problems represented in these frames differ by their objective function: each of them has been endowed with a linear objective function pointing towards the direction of the relative position of the frame on the page. For example, the objective functions in the first and second pictures on the cover point towards the north-west and north-north-west directions.

We have drawn for each of these problems some level sets of a suitable barrier function combined with the objective function (more precisely, it is the objective function of problem (CL_μ) from Chapter 2 with $\mu = 1$) and the central path corresponding to this barrier function (see again Chapter 2). The endpoints of this central path correspond to the minimum and the maximum of the corresponding objective function on the feasible region.

One can notice that the level sets tend to be shifted in the direction of the objective function, and that the central path can sometimes take surprising turns before reaching its optimal endpoints.