Acta Numerica (2008), pp. 191–234
doi: 10.1017/S0962492906370018
© Cambridge University Press, 2008
Printed in the United Kingdom

Interior-point methods for optimization

Arkadi S. Nemirovski
School of Industrial and Systems Engineering,
Georgia Institute of Technology,
Atlanta, Georgia 30332, USA
E-mail: arkadi.nemirovski@isye.gatech.edu

Michael J. Todd
School of Operations Research and Information Engineering,
Cornell University, Ithaca, NY 14853, USA
E-mail: mjt7@cornell.edu

This article describes the current state of the art of interior-point methods (IPMs) for convex, conic, and general nonlinear optimization. We discuss the theory, outline the algorithms, and comment on the applicability of this class of methods, which have revolutionized the field over the last twenty years.

CONTENTS

1 Introduction 191
2 The self-concordance-based approach to IPMs 194
3 Conic optimization 210
4 IPMs for non-convex programming 228
5 Summary 230
References 231

1. Introduction

During the last twenty years, there has been a revolution in the methods used to solve optimization problems. In the early 1980s, sequential quadratic programming and augmented Lagrangian methods were favoured for nonlinear problems, while the simplex method was basically unchallenged for linear programming. Since then, modern interior-point methods (IPMs) have infused virtually every area of continuous optimization, and have forced great improvements in the earlier methods. The aim of this article is to describe interior-point methods and their application to convex programming, special conic programming problems (including linear and semidefinite programming), and general possibly non-convex programming.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

We have also tried to complement the earlier articles in this journal by Wright (1992), Lewis and Overton (1996), and Todd (2001).

Almost twenty-five years ago, Karmarkar (1984) proposed his projective method to solve linear programming problems: from a theoretical point of view, this was a polynomial-time algorithm, in contrast to Dantzig's simplex method. Moreover, with some refinements it proved a very worthy competitor in practical computation, and substantial improvements to both interior-point and simplex methods have led to the routine solution of problems (with hundreds of thousands of constraints and variables) that were considered untouchable previously. Most commercial software, for example CPlex (Bixby 2002) and XpressMP (Guéret, Prins and Sevaux 2002), includes interior-point as well as simplex options.

The majority of the early papers following Karmarkar's dealt exclusively with linear programming and its near-relatives, convex quadratic programming and the (monotone) linear complementarity problem. Gill, Murray, Saunders, Tomlin and Wright (1986) showed the strong connection to earlier barrier methods in nonlinear programming; Renegar (1988) and Gonzaga (1989) introduced path-following methods with an improved iteration complexity; and Megiddo (1989) suggested, and Monteiro and Adler (1989) and Kojima, Mizuno and Yoshise (1989) realized, primal–dual versions of these algorithms, which are the most successful in practice.

At the same time, Nesterov and Nemirovski were investigating the new methods from a more fundamental viewpoint: What are the basic properties that lead to polynomial-time complexity? It turned out that the key property is that the barrier function should be *self-concordant*. This seemed to provide a clear, complexity-based criterion to delineate the class of optimization problems that could be solved in a provably efficient way using the new methods. The culmination of this work was the book by Nesterov and Nemirovski (1994), whose complexity emphasis contrasted with the classic text on barrier methods by Fiacco and McCormick (1968).

Fiacco and McCormick describe the history of (exterior) penalty and barrier (sometimes called interior penalty) methods; other useful references are Nash (1998) and Forsgren, Gill and Wright (2002). Very briefly, Courant (1943) first proposed penalty methods, while Frisch (1955) suggested the logarithmic barrier method and Carroll (1961) the inverse barrier method (which inspired Fiacco and McCormick). While these methods were among the most successful for solving constrained nonlinear optimization problems in the 1960s, they lost favour in the late 1960s and 1970s when it became apparent that the subproblems that needed to be solved became increasingly ill-conditioned as the solution was approached.

The new research alleviated these fears to some extent, at least for certain problems. In addition, the ill-conditioning turned out to be relatively benign: see, *e.g.*, Wright (1992) and Forsgren *et al.* (2002). Moreover,

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

Nesterov and Nemirovski (1994) showed that, at least in principle, *any* convex optimization problem could be provided with a self-concordant barrier. This was purely an existence result, however, as the generated barrier could not be efficiently evaluated in general. (So we should qualify our earlier statement: the class of optimization problems to which the new methods can be efficiently applied consists of those with a *computationally tractable* self-concordant barrier.) To contrast with the general case, Nesterov and Nemirovski listed a considerable number of important problems where computationally tractable self-concordant barriers were available, and provided a calculus for constructing such functions for more complicated sets. A very significant special case was that of the positive semidefinite cone, leading to semidefinite programming. Independently, Alizadeh (1995) developed an efficient interior-point method for semidefinite programming, with the motivation of obtaining strong bounds for combinatorial optimization problems.

The theory of self-concordant barriers is limited to convex optimization. However, this limitation has become less burdensome as more and more scientific and engineering problems have been shown to be amenable to convex optimization formulations. Researchers in control theory have been much influenced by the ability to solve semidefinite programming problems (or linear matrix inequalities, in their terminology) arising in their field: see Boyd, El Ghaoui, Feron and Balakrishnan (1994). Moreover, a number of seemingly non-convex problems arising in engineering design can be reformulated as convex optimization problems: see Boyd and Vandenberghe (2004) and Ben-Tal and Nemirovski (2001).

Besides the books we have cited, other useful references include the lecture notes of Nemirovski (2004) and the books of Nesterov (2003) and Renegar (2001) for general convex programming; for mostly linear programming, the books of Roos, Terlaky and Vial (1997), Vanderbei (2007), Wright (1997) and Ye (1997); for semidefinite programming, the handbook of Wolkowicz, Saigal and Vandenberghe (2000); and for general nonlinear programming, the survey articles of Forsgren *et al.* (2002) and Gould, Orban and Toint (2005).

In Section 2, we discuss self-concordant barriers and their properties, and then describe interior-point methods for both general convex optimization problems and conic problems, as well as the calculus of self-concordant barriers. Section 3 treats conic optimization in detail, concentrating on symmetric or self-scaled cones, including the non-negative orthant (linear programming) and the positive semidefinite cone (semidefinite programming). We also briefly discuss some recent developments in hyperbolicity cones, global polynomial optimization, and copositive programming. Finally, Section 4 is concerned with the application of interior-point methods to general, possibly non-convex, nonlinear optimization. These methods are used in some of the most effective codes for such problems, such as IPOPT

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

(Wächter and Biegler 2006), KNITRO (Byrd, Nocedal and Waltz 2006), and LOQO (Vanderbei and Shanno 1999).

We have concentrated on the theory and application in structured convex programming of interior-point methods, since the polynomial-time complexity of these methods and its range of applicability have been a major focus of the research of the last twenty years. For further coverage of interior-point methods for general nonlinear programming we recommend the survey articles of Forsgren *et al.* (2002) and Gould, Orban and Toint (2005). Also, to convey the main ideas of the methods, we have given short shrift to important topics including attaining feasibility from infeasible initial points, dealing with infeasible problems, and superlinear convergence. The literature on interior-point methods is huge, and the area is still very active; the reader wishing to follow the latest research is advised to visit the Optimization Online website www.optimization-online.org/ and the Interior-Point Methods Online page at www-unix.mcs.anl.gov/otc/InteriorPoint/. A very useful source is Helmberg's semidefinite programming page www-user.tu-chemnitz.de/~helmberg/semidef.html. Software for optimization problems, including special-purpose algorithms for semidefinite and second-order cone programming, is available at the Network Enabled Optimization System (NEOS) homepage neos.mcs.anl.gov/neos/solvers/index.html.

2. The self-concordance-based approach to IPMs

Preliminaries

The first *path-following* interior-point polynomial-time methods for linear programming, analysed by Renegar (1988) and Gonzaga (1989), turned out to belong to the very well-known *interior penalty scheme* going back to Fiacco and McCormick (1968). Consider a convex program

$$\min\{c^T x : x \in X\}, \tag{2.1}$$

X being a closed convex domain (*i.e.*, a closed convex set with a non-empty interior) in $\mathbb{R}^n$; this is one of the universal forms of a convex program. In order to solve the problem with a path-following scheme, one equips X with an *interior penalty* or *barrier* function F – a smooth and strongly convex[1] function defined on $\operatorname{int} X$ such that $F(x_k) \to +\infty$ on every sequence of points $x_k \in \operatorname{int} X$ converging to a point $\bar{x} \in \partial X$ – and considers the *barrier family* of functions

$$F_t(x) = tc^T x + F(x), \tag{2.2}$$

where $t > 0$ is the *penalty parameter*. Under mild assumptions (*e.g.*, when X is bounded), every function F_t attains its minimum on $\operatorname{int} X$ at a unique point $x_*(t)$, and the *central path* $\{x_*(t) : t \geq 0\}$ converges, as $t \to \infty$,

[1] Hessian positive definite everywhere.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

to the optimal set of (2.1). The path-following scheme for solving (2.1) suggests 'tracing' this path as $t \to \infty$ according to the following conceptual algorithm:

> Given the current iterate $(t_k > 0, x_k \in \operatorname{int} X)$ with x_k 'reasonably close' to $x_*(t_k)$, we
>
> (a) replace the current value t_k of the penalty parameter with a larger value t_{k+1}; and
>
> (b) run an algorithm for minimizing $F_{t_{k+1}}(\cdot)$, starting at x_k, until a point x_{k+1} close to $x_*(t_{k+1}) = \operatorname{argmin}_{\operatorname{int} X} F_{t_{k+1}}(\cdot)$ is found.
>
> As a result, we get a new iterate (t_{k+1}, x_{k+1}) 'close to the path' and loop to step $k+1$.

The main advantage of the scheme described above is that $x_*(t)$ is, essentially, the unconstrained minimizer of F_t, which allows the use in (b) of basically any method for smooth convex unconstrained minimization, *e.g.*, the Newton method. Note, however, that the classical theory of the path-following scheme did not suggest its polynomiality; rather, the standard theory of unconstrained minimization predicted slow-down of the process as the penalty parameter grows. In sharp contrast to this common wisdom, both Renegar and Gonzaga proved that, when applied to the *logarithmic barrier* $F(x) = -\sum_i \ln(b_i - a_i^T x)$ for a polyhedral set $X = \{x : a_i^T x \le b_i, 1 \le i \le m\}$, a Newton-method-based implementation of the path-following scheme can be made polynomial. These breakthrough results were obtained via an *ad hoc* analysis of the behaviour of the Newton method as applied to the logarithmic barrier (augmented by a linear term). In a short time Nesterov realized what intrinsic properties of the standard log-barrier are responsible for this polynomiality, and this crucial understanding led to the general *self-concordance-based* theory of polynomial-time interior-point methods developed in Nesterov and Nemirovski (1994); this theory explained the nature of existing interior-point methods (IPMs) for LP and allowed the extension of these methods to the entire field of convex programming. We now provide an overview of the basic results of this theory.[2]

2.1. Self-concordance

In retrospect, the notion of self-concordance can be extracted from analysis of the classical results on the local quadratic convergence of Newton's

[2] Up to minor refinements which can be found in Nemirovski (2004), all results quoted in the next subsection without explicit references are taken from Nesterov and Nemirovski (1994).

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

method as applied to a smooth convex function f with non-singular Hessian. These results state that a quantitative description of the domain of quadratic convergence depends on (a) the condition number of $\nabla^2 f$ evaluated at the minimizer x_*, and (b) the Lipschitz constant of $\nabla^2 f$. In hindsight, such a description seems unnatural, since it is 'frame-dependent': it heavily depends on an *ad hoc* choice of the Euclidean structure in $\mathbb{R}^n$; indeed, both the condition number of $\nabla^2 f(x_*)$ and the Lipschitz constant of $\nabla^2 f(\cdot)$ depend on this structure, which is in sharp contrast to the affine invariance of the Newton method itself. At the same time, a smooth strongly convex function f by itself defines at every point x a Euclidean structure $\langle u, v\rangle_{f,x} = D^2 f(x)[u, v]$. With respect to this structure, $\nabla^2 f(x)$ is as well-conditioned as it could be – it is just the unit matrix. The idea of Nesterov was to use this local Euclidean structure, intrinsically linked to the function f we intend to minimize, in order to quantify the Lipschitz constant of $\nabla^2 f$, with the ultimate goal of getting a 'frame-independent' description of the behaviour of the Newton method. The resulting notion of self-concordance is defined as follows.

Definition 2.1. Let $X \subset \mathbb{R}^n$ be a closed convex domain. A function $f : \operatorname{int} X \to \mathbb{R}$ is called *self-concordant* (SC) on X if

(i) f is a three times continuously differentiable convex function with $f(x_k) \to \infty$ if $x_k \to \bar{x} \in \partial X$; and
(ii) f satisfies the differential inequality

$$|D^3 f(x)[h, h, h]| \le 2\big(D^2 f(x)[h, h]\big)^{3/2}, \quad \forall x \in \operatorname{int} X,\ h \in \mathbb{R}^n. \tag{2.3}$$

Given a real $\vartheta \ge 1$, F is called a *ϑ-self-concordant barrier* (ϑ-SCB) for X if F is self-concordant on X and, in addition,

$$|DF(x)[h]| \le \vartheta^{1/2}\big(D^2 F(x)[h, h]\big)^{1/2}, \quad \forall x \in \operatorname{int} X,\ h \in \mathbb{R}^n. \tag{2.4}$$

(As above, we will use f for a general SC function and F for an SCB in what follows.) Note that the powers 3/2 and 1/2 in (2.3) and (2.4) are a must, since both sides of the inequalities should be of the same homogeneity degree with respect to h. In contrast to this, the two sides of (2.3) are of different homogeneity degrees with respect to f, meaning that if f satisfies a relation of the type (2.3) with some constant factor on the right-hand side, we can always make this factor equal to 2 by scaling f appropriately. The advantage of the specific factor 2 is that with this definition, the function $x \mapsto -\ln(x) : \mathbb{R}_{++} \to \mathbb{R}$ becomes a 1-SCB for $\mathbb{R}_+$ directly, without any scaling, and this function is the main building block of the theory we are presenting. Finally, we remark that (2.3) and (2.4) have a very transparent interpretation: they mean that $D^2 f$ and F are Lipschitz-continuous, with constants 2 and $\vartheta^{1/2}$, in the local Euclidean (semi)norm $\|h\|_{f,x} = \sqrt{\langle h, h\rangle_{f,x}} = \sqrt{h^T \nabla^2 f(x) h}$ defined by f or similarly by F.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

It turns out that self-concordant functions possess nice local properties and are perfectly well suited to Newton minimization. We are about to present the most important of the related results. In what follows, f is an SC function on a closed convex domain X.

2.1.0. Bounds on third derivatives and the recession space of SC *functions*
For all $x \in \operatorname{int} X$ and all $h_1, h_2, h_3 \in \mathbb{R}^n$, we have

$$|D^3 f(x)[h_1, h_2, h_3]| \le 2\|h_1\|_{f,x}\|h_2\|_{f,x}\|h_3\|_{f,x}.$$

The *recession subspace* $E_f = \{h : D^2 f(x)[h,h] = 0\}$ of f is independent of $x \in \operatorname{int} X$, and $X = X + E_f$. In particular, if $\nabla^2 f(x)$ is positive definite at some point in $\operatorname{int} X$, then $\nabla^2 f(x)$ is positive definite for all $x \in \operatorname{int} X$ (in this case, f is called a *non-degenerate* SC function; this is always the case when X does not contain lines).

It is convenient to write $A \succ 0$ ($A \succeq 0$) to denote that the symmetric matrix A is positive definite (semidefinite), and $A \succeq B$ and $B \preceq A$ ($A \succ B$ and $B \prec A$) if $A - B \succeq 0$ ($A - B \succ 0$).

2.1.1. Dikin's ellipsoid and the local behaviour of f
For every $x \in \operatorname{int} X$, the *unit Dikin ellipsoid* of f $\{y : \|y - x\|_{f,x} \le 1\}$ is contained in X, and within this ellipsoid, f is nicely approximated by its second-order Taylor expansion:

$$\begin{aligned} &r := \|h\|_{f,x} < 1 \;\Rightarrow \\ &(1-r)^2\nabla^2 f(x) \;\preceq\; \nabla^2 f(x+h) \;\preceq\; \frac{1}{(1-r)^2}\nabla^2 f(x), \qquad (2.5)\\ f(x) + \nabla f(x)^T h + \rho(-r) \;\le\; &f(x+h) \;\le\; f(x) + \nabla f(x)^T h + \rho(r), \end{aligned}$$

where $\rho(s) := -\ln(1-s) - s = s^2/2 + s^3/3 + \cdots$. (Indeed, the lower bound in the last line holds true for all h such that $x + h \in \operatorname{int} X$.)

2.1.2. The Newton decrement and the damped Newton method
Let f be non-degenerate. Then $\|\cdot\|_{f,x}$ is a norm, and its conjugate norm is $\|\eta\|^*_{f,x} = \max\{h^T\eta : \|h\|_{f,x} \le 1\} = \sqrt{\eta^T[\nabla^2 f(x)]^{-1}\eta}$. The quantity

$$\begin{aligned} \lambda(x,f) := \|\nabla f(x)\|^*_{f,x} &= \|[\nabla^2 f(x)]^{-1}\nabla f(x)\|_{f,x} \\ &= \max_h\{Df(x)[h] : D^2 f(x)[h,h] \le 1\}, \end{aligned}$$

called the *Newton decrement* of f at x, is a finite continuous function of $x \in \operatorname{int} X$ which vanishes exactly at the (unique, if any) minimizer x_f of f on $\operatorname{int} X$; this function can be considered as the 'observable' measure

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

of proximity of x to x_f. The Newton decrement possesses the following properties:

$$\lambda(x,f)<1 \Rightarrow \begin{cases} \operatorname{argmin}_{\operatorname{int} X} f \neq \emptyset, & (a)\\ f(x)-\min_{\operatorname{int} X} f \leq \rho(\lambda(x,f)), & (b)\\ \|x_f-x\|_{f,x} \leq \frac{\lambda(x,f)}{1-\lambda(x,f)}, & (c)\\ \|x_f-x\|_{f,x_f} \leq \frac{\lambda(x,f)}{1-\lambda(x,f)}. & (d) \end{cases} \tag{2.6}$$

In particular, when it is at most $1/2$, the Newton decrement is, within an absolute constant factor, the same as $\|x-x_f\|_{f,x}$, $\|x-x_f\|_{f,x_f}$, and $\sqrt{f(x)-\min_{\operatorname{int} X} f}$.

The *damped Newton method* as applied to f is the iterative process

$$x_{k+1}=x_k-\frac{1}{1+\lambda(x_k,f)}[\nabla^2 f(x_k)]^{-1}\nabla f(x_k) \tag{2.7}$$

starting at a point $x_0 \in \operatorname{int} X$. The damped Newton method is well defined: all its iterates belong to $\operatorname{int} X$. Besides this, setting $\lambda_j := \lambda(x_j,f)$, we have

$$\lambda_{k+1} \leq 2\lambda_k^2 \quad \text{and} \quad f(x_k)-f(x_{k+1}) \geq \rho(-\lambda_k)=\lambda_k-\ln(1+\lambda_k). \tag{2.8}$$

As a consequence of (2.8) and (2.7), we get the following 'frame- and data-independent' description of the convergence properties of the damped Newton method as applied to an SC function f: the domain of quadratic convergence is $\{x : \lambda(x,f) \leq 1/4\}$; after this domain is reached, every step of the method nearly squares the Newton decrement, the $\|\cdot\|_{f,x_f}$-distance to the minimizer and the residual in terms of f. Before the domain is reached, every step of the method decreases the objective by at least $\Omega(1)=1/4-\ln(5/4)$. It follows that a non-degenerate SC function admits its minimum on the interior of its domain if and only if it is bounded below, and if and only if $\lambda(x,f)<1$ for certain x. Whenever this is the case, for every $\epsilon \in (0,0.1]$ the number of steps N of the damped Newton method which ensures that $f(x_k) \leq \min_{\operatorname{int} X} f+\epsilon$ does not exceed $O(1)\big[\ln\ln(1/\epsilon)+f(x_0)-\min_{\operatorname{int} X} f\big]$. (Here and below, $O(1)$ denotes a suitably chosen absolute constant.)

2.1.3. Self-concordance and Legendre transformations

Let f be non-degenerate. Then the domain $\{y : f_*(y)<\infty\}$ of the (modified) Legendre transformation

$$f_*(y)=\sup_{x \in \operatorname{int} X}\big[-y^T x-f(x)\big]$$

of f is an open convex set, f_* is self-concordant on the closure X_* of this set, and the mappings $x \mapsto -\nabla f(x)$ and $y \mapsto -\nabla f_*(y)$ are bijections of $\operatorname{int} X$ and $\operatorname{int} X_*$ that are inverse to each other. Besides this, X_* is a closed cone with a non-empty interior, specifically, the cone dual to the recession cone of X.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

We next list specific properties of SCBs not shared by more general SC functions. In what follows, F is a non-degenerate ϑ-SCB for a closed convex domain X.

2.1.4. Non-degeneracy, semiboundedness, attaining minimum
F is non-degenerate if and only if X does not contain lines. We have

$$\forall(x \in \operatorname{int} X, y \in X) : \nabla F(x)^T(y-x) \le \vartheta \tag{2.9}$$

(semiboundedness) and

$$\forall(x \in \operatorname{int} X, y \in X \text{ with } \nabla F(x)^T(y-x) \ge 0) : \|y-x\|_{F,x} \le \vartheta + 2\sqrt{\vartheta}. \tag{2.10}$$

F attains its minimum on $\operatorname{int} X$ if and only if X is bounded; otherwise $\lambda(x,F) \ge 1$ for all $x \in \operatorname{int} X$.

2.1.5. Useful bounds
For $x \in \operatorname{int} X$, let $\pi_x(y) = \inf\{t : t > 0, x + t^{-1}(y-x) \in X\}$ be the Minkowski function of X with respect to x. We have

$$\forall(x, y \in \operatorname{int} X) : \begin{cases} F(y) \le F(x) + \vartheta \ln\left(\frac{1}{1-\pi_x(y)}\right), \\ F(y) \ge F(x) + \nabla F(x)^T(y-x) + \ln\left(\frac{1}{1-\pi_x(y)}\right) - \pi_x(y). \end{cases} \tag{2.11}$$

2.1.6. Existence of the central path and its convergence to the optimal set
Consider problem (2.1) and assume that the domain X of the problem is equipped with a self-concordant barrier F, and the level sets of the objective $\{x \in X : c^T x \le \alpha\}$ are bounded. In the situation in question, F is non-degenerate, $c^T x$ attains its minimum on X, the central path

$$x_*(t) := \operatorname*{argmin}_{x \in \operatorname{int} X} F_t(x), \quad F_t(x) := tc^T x + F(x), \quad t > 0,$$

is well-defined, all functions F_t are self-concordant on X and

$$\epsilon(x_*(t)) := c^T x_*(t) - \min_{x \in X} c^T x \le \frac{\vartheta}{t}, \quad t > 0. \tag{2.12}$$

Moreover, if $\lambda(x, F_t) \le \frac{1}{2}$ for some $t > 0$, then

$$\epsilon(x) \le \frac{\vartheta + \sqrt{\vartheta}}{t}. \tag{2.13}$$

Let us derive the claims in 2.1.6 from the preceding facts, mainly in order to explain why these facts are important. By 2.1.1, F is non-degenerate, since X does not contain lines. The fact that all F_t are SC is evident from the definition: self-concordance is clearly preserved when adding a linear or convex quadratic function to an SC one. Further, the level sets of the

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

objective on X are bounded, so that the objective attains its minimum over X at some point x_* and, as is easily seen, is coercive on X: $c^T x \geq \alpha + \beta\|x\|$ for all $x \in X$ with appropriate constants $\beta > 0$ and α ($\|\cdot\|$, without subscripts, always denotes the Euclidean norm). Now fix a point $\bar{y}$ in $\operatorname{int} X$; then $\pi_x(\bar{y}) \leq \frac{\|\bar{y}-x\|}{r+\|\bar{y}-x\|}$ for all $x \in \operatorname{int} X$, where $r > 0$ is such that a $\|\cdot\|$-ball of radius r centred at $\bar{y}$ belongs to X. Invoking the first line of (2.11) with $y = \bar{y}$, we conclude that $F(x) \geq F(\bar{y}) + \vartheta \ln(\frac{r}{r+\|x-\bar{y}\|})$ for all $x \in \operatorname{int} X$. Recalling that the objective is coercive, we conclude that $F_t(x) \to \infty$ as $x \in X$ and $\|x\| \to \infty$, so that the level sets of F_t are bounded. Since F_t is, along with F, an interior penalty for X, these sets are in fact compact subsets of $\operatorname{int} X$, whence F_t attains its minimum on $\operatorname{int} X$. Since F_t is convex and non-degenerate along with F, the minimizer is unique; thus, the central path is well-defined. To verify (2.12), note that $\nabla F(x_*(t)) = -tc$, whence $c^T(x_*(t) - y) = t^{-1}\nabla F(x_*(t)^T(y - x_*(t))$. By (2.9), the right-hand side in this equality is at most ϑ/t, provided $y \in X$, and (2.12) follows. Finally, when $\lambda(x, F_t) \leq 1/2$, then $\|x - x_*(t)\|_{F_t, x_*(t)} \leq 1$ by (2.6.d) as applied to F_t instead of F. Since $\|\cdot\|_{F_t,u} \equiv \|\cdot\|_{F,u}$, we get $\|x - x_*(t)\|_{F,x_*(t)} \leq 1$, whence

$$\begin{aligned} c^T(x - x_*(t)) &\leq \|c\|^*_{F,x_*(t)}\|x - x_*(t)\|_{F,x_*(t)} \\ &\leq \|c\|^*_{F,x_*(t)} = t^{-1}\|\nabla F(x_*(t))\|^*_{F,x_*(t)} \leq t^{-1}\sqrt{\vartheta}, \end{aligned}$$

which combines with (2.12) to imply (2.13).

2.2. A primal polynomial-time path-following method

As an immediate consequence of the above results, we arrive at the following important result.

Theorem 2.1. Consider problem (2.1) and assume that the level sets of the objective are bounded, and we are given a ϑ-SCB F for X; according to 2.1.6, c and F define a central path $x_*(\cdot)$. Suppose we also have at our disposal a starting pair ($t_0 > 0, x_0 \in \operatorname{int} X$) which is close to the path in the sense that $\lambda(x_0, F_{t_0}) \leq 0.1$, and consider the following implementation (the basic path-following algorithm) of the path-following scheme:

$$(t_k, x_k) \mapsto \begin{cases} t_{k+1} = (1 + 0.1\vartheta^{-1/2})t_k, \\ x_{k+1} = x_k - \frac{1}{1+\lambda(x_k, F_{t_{k+1}})}[\nabla^2 F(x_k)]^{-1}\nabla F_{t_{k+1}}(x_k). \end{cases} \tag{2.14}$$

This recurrence is well-defined (*i.e.*, $x_k \in \operatorname{int} X$ for all k), maintains closeness to the path (*i.e.*, $\lambda(x_k, F_{t_k}) \leq 0.1$ for all k) and ensures the efficiency estimate

$$\forall k : c^T x_k - \min_{x \in X} c^T x \leq \frac{\vartheta + \sqrt{\vartheta}}{t_k} \leq \frac{\vartheta + \sqrt{\vartheta}}{t_0} \exp\left\{-\frac{0.095}{\sqrt{\vartheta}}k\right\}. \tag{2.15}$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

In particular, for every $\epsilon > 0$, it takes at most

$$N(\epsilon) = O(1)\sqrt{\vartheta}\ln\left(\frac{\vartheta}{t_0\epsilon} + 2\right)$$

steps of the recurrence to get a strictly feasible (*i.e.*, in $\operatorname{int} X$) solution to the problem with residual in terms of the objective at most ϵ.

Proof. In view of 2.1.2 and 2.1.6, all we need to prove by induction on k is that (2.14) maintains closeness to the path. Assume that $(t = t_k, x = x_k)$ is close to the path, and let us verify that the same is true for $(t_+ = t_{k+1}, x_+ = x_{k+1})$. Taking into account that $\|\cdot\|_{F_t,u} = \|\cdot\|_{F,u}$ and similarly for the conjugate norms, we have $0.1 \geq \lambda(x, F_t) = \|tc + \nabla F(x)\|^*_{F,x}$ so that

$$\begin{aligned} t\|c\|^*_{F,x} &\leq 0.1 + \|\nabla F(x)\|^*_{F,x} \\ &\leq 0.1 + \vartheta^{1/2}, \end{aligned}$$

using the fact that F is a ϑ-SCB. This implies that

$$\begin{aligned} \lambda(x, F_{t_+}) &= \|[tc + \nabla F(x)] + (t_+ - t)c\|^*_{F,x} \\ &\leq \|tc + \nabla F(x)\|^*_{F,x} + (t_+/t - 1)t\|c\|^*_{F,x} \\ &\leq 0.1 + 0.1\vartheta^{-1/2}t\|c\|^*_{F,x} \leq 0.1[1 + \vartheta^{-1/2}[0.1 + \vartheta^{1/2}]] \leq 0.21. \end{aligned}$$

Finally, we obtain

$$\lambda(x_+, F_{t_+}) \leq 2\lambda^2(x, F_{t_+}) \leq 0.1,$$

using (2.8). □

Remarks. **A** The algorithm presented in Theorem 2.1 is, in a sense, incomplete: it does not explain how to approach the central path in order to start path-tracing. There are many ways to resolve this issue. Assume, *e.g.*, that X is bounded and we know in advance a point $y \in \operatorname{int} X$. When X is bounded, every linear form $g^T x$ generates a central path, and we can easily find such a path passing through y: with $g = -\nabla F(y)$, the corresponding path passes through y when $t = 1$. Now, as $t \to +0$, all paths converge to the minimizer x_F of F over X, and thus approach each other. At the same time, we can as easily trace the paths backwards as trace them forwards – with the parameter updating rule $t_{k+1} = (1 - 0.1\vartheta^{-1/2})t_k$, the recurrence in Theorem 2.1 still maintains closeness to the path, now along a sequence of values of the parameter t decreasing geometrically. Thus, we can trace the auxiliary path passing through y backwards until coming close to the path of interest, and then start tracing the latter path forwards. A simple analysis demonstrates that with simple on-line termination and switching rules, the resulting algorithm, for every $\epsilon > 0$, produces a strictly feasible

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

ϵ-solution to the problem at the price of no more than

$$O(1)\sqrt{\vartheta}\ln\left(\frac{\vartheta\mathcal{V}}{(1-\pi_{x_F}(y))\epsilon}+2\right)$$

Newton steps of both phases, where $\mathcal{V}=\max_{x\in X}c^Tx-\min_{x\in X}c^Tx$.

B The outlined path-following algorithm, using properly chosen SC barriers, yields the currently best polynomial-time complexity bounds for basically all 'well-structured' generic convex programs, such as those of linear, second-order cone, semidefinite, and geometric programming, to name just a few. At the same time, from a practical perspective a severe shortcoming of the algorithm is its worst-case-oriented nature: as presented, it will *always* perform according to its worst-case theoretical complexity bounds. There exist implementations of IPMs that are much more powerful in practice, using more aggressive parameter updating policies that are adjusted during the course of the algorithm. All known algorithms of this type are *primal–dual*: they work simultaneously on the problem and its dual, and nearly all of them, including all those implemented so far in professional software, work with conic problems, specifically, those of linear, second-order cone, and semidefinite programming (the only exceptions are the cone-free primal–dual methods proposed in Nemirovski and Tunçel (2005); these methods, however, have not yet been implemented). Our next goal is to describe the general theory of primal–dual interior-point methods for conic problems.

2.3. Interior-point methods for conic problems

Interior-point methods for conic problems are associated with specific ϑ-SC barriers for cones, those satisfying the so-called *logarithmic homogeneity* condition.

Definition 2.2. Let $\mathbf{K}\subset\mathbb{R}^n$ be a cone (from now on, all cones are closed and convex, have non-empty interiors, and contain no lines). A ϑ-self-concordant barrier F for $\mathbf{K}$ is called *logarithmically homogeneous* (an LHSCB), if

$$\forall(\tau>0, x\in\operatorname{int}\mathbf{K}): F(\tau x)=F(x)-\vartheta\ln\tau. \tag{2.16}$$

In fact, *every* self-concordant function on a cone $\mathbf{K}$ satisfying the identity (2.16) is automatically a ϑ-SCB for $\mathbf{K}$, since whenever a smooth function F satisfies the identity (2.16), we have

$$\forall(x\in\operatorname{int}K): \nabla F(x)^Tx=-\vartheta,\quad \nabla^2F(x)x=-\nabla F(x); \tag{2.17}$$

it follows that when F is self-concordant, we have

$$\lambda(x,F)=\sqrt{\nabla F(x)^T[\nabla^2F(x)]^{-1}\nabla F(x)}=\sqrt{-\nabla F(x)^Tx}=\sqrt{\vartheta},$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

meaning that F is indeed a ϑ-SCB for $\mathbf{K}$. A nice and important fact is that the (modified) Legendre transformation $F_*(s)$ of a ϑ-LHSCB F for a cone $\mathbf{K}$ is a ϑ-LHSCB for the cone

$$\mathbf{K}_* := \{s \in \mathbb{R}^n : s^T x \geq 0,\ \forall x \in \mathbf{K}\} \tag{2.18}$$

dual to $\mathbf{K}$. The resulting symmetry of LHSCBs complements the symmetry between cones and their duals. Moreover, we have the following result.

Proposition 2.1. The mappings $x \mapsto -\nabla F(x)$ and $s \mapsto -\nabla F_*(s)$ are inverse bijections between $\operatorname{int}\mathbf{K}$ and $\operatorname{int}\mathbf{K}_*$, and these bijections are homogeneous of degree -1: $-\nabla F(\tau x) = -\tau^{-1}\nabla F(x)$, $x \in \operatorname{int}\mathbf{K}$, $\tau > 0$, and similarly for F_*. Finally, $\nabla^2 F$ and $\nabla^2 F_*$ are homogeneous of degree -2, with $\nabla^2 F_*(-\nabla F(x)) = [\nabla^2 F(x)]^{-1}$ and $\nabla^2 F(-\nabla F_*(s)) = [\nabla^2 F_*(s)]^{-1}$.

Now assume that we want to solve a primal–dual pair of conic problems

$$\begin{aligned} &\min_x \{c^T x : Ax = b, x \in \mathbf{K}\} \qquad (P), \\ &\max_{y,s} \{b^T y : A^T y + s = c,\ s \in \mathbf{K}_*\} \ (D), \end{aligned} \tag{2.19}$$

where the rows of A are linearly independent and both problems have strictly feasible solutions (*i.e.*, feasible solutions with $x \in \operatorname{int}\mathbf{K}$ and $s \in \operatorname{int}\mathbf{K}_*$). Assume also that we have at our disposal a ϑ-LHSCB F for $\mathbf{K}$ along with its Legendre transformation F_*, which is a ϑ-LHSCB for $\mathbf{K}_*$. (P) can be treated as a problem of the form (2.1), with the affine set $L = \{x : Ax = b\}$ playing the role of the 'universe' $\mathbb{R}^n$ and $\mathbf{K} \cap L$ in the role of X. It is easily seen that the restriction of F to $\operatorname{int}\mathbf{K} \cap L$ is a ϑ-SCB for the resulting problem (2.1) (see rule **D** in Section 2.4), and that this is a problem with bounded level sets. As a result, we can define the *primal central path* $\{x_*(t)\}$, which comprises strictly feasible solutions to (P) and converges, as $t \to \infty$, to the primal optimal set. Similarly, setting $Y = \{y : c - A^T y \in \mathbf{K}_*\}$, the dual problem can be written in the form of (2.1), namely, as $\min_{y \in Y}[-b]^T y$. The domain Y of this problem can also be equipped with a ϑ-SCB, namely, $F_*(c - A^T y)$, and again the problem has bounded level sets, so that we can define the associated central path $\{y_*(t)\}$. This induces the *dual central path* $\{s_*(t) := c - A^T y_*(t)\}$; the latter path comprises interior points of $\mathbf{K}_*$. We have arrived at the *primal–dual central path* $\{z_*(t) := (x_*(t), s_*(t))\}$ 'living' in the interior of $\mathbf{K} \times \mathbf{K}_*$. It is easily seen that *for every $t > 0$, the point $z_*(t)$ is uniquely defined by the following restrictions on its components x, s:*

$$\begin{aligned} &x \in X^o := \operatorname{int}\mathbf{K} \cap \{x : Ax = b\} \text{ [strict primal feasibility]}, \\ &s \in S^o := \operatorname{int}\mathbf{K}_* \cap \{s : \exists y \text{ such that } A^T y + s = c\} \text{ [strict dual feasibility]}, \\ &\left.\begin{aligned} s &= -t^{-1}\nabla F(x) \\ x &= -t^{-1}\nabla F_*(s) \end{aligned}\right\} \text{ [augmented complementary slackness]}. \end{aligned} \tag{2.20}$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

Note that, by Proposition 2.1, each of the complementary slackness equations implies the other, so that we could eliminate either one of them; we keep both to highlight the primal–dual symmetry.

Primal–dual path-following interior-point methods trace simultaneously the primal and dual central paths basically in the same fashion as the method described in Theorem 2.1. It turns out that tracing the paths together is much more advantageous than tracing only one of them. In our general setting these advantages permit, for example,

- adaptive long-step strategies for path-tracing (Nesterov 1997);
- an elegant way ('self-dual embedding'; see, *e.g.*, Ye, Todd and Mizuno (1994), Xu, Hung and Ye (1996), Andersen and Ye (1999), Luo, Sturm and Zhang (2000), de Klerk, Roos and Terlaky (1997) and Potra and Sheng (1998)) to initialize path-tracing even in the case when no strictly feasible solutions to (P) and (D) are available in advance; and
- building certificates of strict (*i.e.*, preserved by small perturbations of the data) primal or dual infeasibility (Nesterov, Todd and Ye 1999) when it holds, *etc.*

Primal–dual IPMs achieve their full power when the underlying cones are *self-scaled*, which is the case in linear, second-order cone, and semidefinite programming, considered in depth in Section 3. In the remaining part of this subsection, we overview, still in the general setting, another family of primal–dual IPMs, those based on *potential reduction*.

Potential-reduction interior-point methods

We now present two potential-reduction IPMs which are straightforward conic generalizations, developed by Nesterov and Nemirovski (1994), of algorithms originally proposed for LP.

Karmarkar's Algorithm. The first polynomial-time interior-point method for LP was discovered by Karmarkar (1984). The conic generalization of the algorithm is as follows. Assume that we want to solve a strictly feasible problem (P) in the following special setting: the primal feasible set $X := \{x \in \mathbf{K} : Ax = b\}$ is bounded, the optimal value is known to be 0, and we know a strictly feasible primal starting point $\bar{x}$ (using conic duality, every strictly primal–dual feasible conic problem can be transformed into this form). Lastly, $\mathbf{K}$ is equipped with a ϑ-LHSCB F. It is immediately seen that under our assumptions $b \neq 0$, so that we lose nothing when assuming that the first equality constraint reads $e^T x = 1$ for some vector e. Subtracting this equality with appropriate coefficients from the remaining equality constraints in (P), we can make all these constraints homogeneous, thus representing the problem in the form

$$\min_x \{c^T x : x \in L \cap \mathbf{K}, e^T x = 1\},$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

where L is a linear subspace in $\mathbb{R}^n$. Note that since X is bounded, we have $e^T x > 0$ for every $0 \neq x \in (L \cap \mathbf{K})$. If we exclude the trivial case $c^T \bar{x} = 0$ (here already $\bar{x}$ is an optimal solution), $c^T x$ is positive on the relative interior $X^o := X \cap \operatorname{int} \mathbf{K}$ of X, so that the projective transformation $x \mapsto p(x) := x/c^T x$ is well defined on X^o; this transformation maps X^o onto the relative interior $Z^o := Z \cap \operatorname{int} \mathbf{K}$ of the set $Z := \{z : z \in L \cap \mathbf{K}, c^T z = 1\}$, the inverse transformation being $z \mapsto z/e^T z$. The point is that Z *is unbounded*, since otherwise the linear form $e^T z$ would be bounded and positive on Z due to $e^T x > 0$ for $0 \neq x \in L \cap \mathbf{K}$, and so $c^T x$ would be bounded away from 0 on X^o, which is not the case. All we need is to generate a sequence $z_k \in Z^o$ such that $\|z_k\| \to \infty$ as $k \to \infty$; indeed, for such a sequence we clearly have $e^T z_k \to \infty$ and $c^T z_k = 1$, whence the points $x_k = z_k/e^T z_k$, which are feasible solutions to the problem of interest, satisfy $c^T x_k \to 0 = \min_{x \in X} c^T x$ as $k \to \infty$. This is how we 'run to ∞ along Z' using Karmarkar's algorithm. Let $G(z)$ be the restriction of F to Z^o. Treating Z as a subset of its affine hull $\operatorname{Aff}(Z)$, so that Z is a closed convex domain in a certain $\mathbb{R}^n$, we find that G is a ϑ-SCB for Z (see rule **D** in Section 2.4). Since Z, along with $\mathbf{K}$, does not contain lines, G is non-degenerate and therefore $\lambda(z, G) \geq 1$ for all $z \in Z^o$ by 2.1.4 (recall that Z is unbounded); applying 2.1.2, we conclude that *the step $z \mapsto z^+(z)$ of the damped Newton method as applied to G, z maps Z^o into Z^o and reduces G by at least the absolute constant $\delta := 1 - \ln(2) > 0$*. It follows that applying the damped Newton method to G, we push G to $-\infty$, and therefore indeed run to ∞ along Z. To get an explicit efficiency estimate, let us look at the *Karmarkar potential function* $\phi : X^o \to \mathbb{R}$ defined by $\phi(x) = \vartheta \ln(c^T x) + F(x)$; note that $\phi(x) = G(p(x))$ due to the ϑ-logarithmical homogeneity of F. It follows that *the basic Karmarkar step $x \mapsto x_+(x) = p^{-1}(z^+(p(x)))$ maps X^o into itself and reduces the potential by at least δ*. In Karmarkar's algorithm, one iterates this step (usually augmented by a line search aimed at getting a larger reduction in the potential than the guaranteed reduction δ) starting with $x_0 := \bar{x}$, thus generating a sequence $\{x_k\}_{k=0}^{\infty}$ of strictly feasible solutions to (P) such that $\phi(x_k) \leq \phi(\bar{x}) - k\delta = F(\bar{x}) + \vartheta \ln(c^T \bar{x}) - k\delta$. Recalling that X is bounded, so that F is bounded below on X^o by 2.1.4, we have also $F(x) \geq \widehat{F} := \min_{x \in X^o} F(x)$, whence $F(\bar{x}) + \vartheta \ln(c^T \bar{x}) - k\delta \geq \phi(x_k) \geq \widehat{F} + \vartheta \ln(c^T x_k)$. We arrive at the efficiency estimate

$$c^T x_k = c^T x_k - \min_{x \in X} c^T x \leq c^T \bar{x} \exp\left(\frac{F(\bar{x}) - \widehat{F} - k\delta}{\vartheta}\right),$$

meaning that, for every $\epsilon \in (0, 1)$, at most $\left\lfloor \frac{[F(\bar{x}) - \widehat{F}] + \vartheta \ln(1/\epsilon)}{\delta} \right\rfloor + 1$ steps of the method are needed to arrive at a strictly feasible solution x_k with $c^T x_k = c^T x_k - \min_X c^T x \leq \epsilon c^T \bar{x}$. The advantage of the Karmarkar algorithm, as compared to that in Theorem 2.1, is that our only interest now

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

is driving the (on-line-observable) potential function $\phi(x)$ to $-\infty$ as rapidly as possible, while staying strictly feasible at all times; this can be done, *e.g.*, by augmenting the basic step with an appropriate line search, which usually leads to a much larger reduction in $\phi(\cdot)$ at each step than the reduction δ guaranteed by the theory. As a result, the practical performance of Karmarkar's algorithm is typically much better than predicted by the theoretical complexity estimate above. On the negative side, the latter estimate is worse than that for the basic path-following method from Theorem 2.1: now the complexity is proportional to ϑ rather than to $\vartheta^{1/2}$ and $\vartheta \geq 1$ may well be large. To circumvent this difficulty, we now present a *primal–dual* potential-reduction algorithm, extending to the general conic case the algorithm of Ye (1991) originally developed for LP.

Primal–dual potential-reduction algorithm. This algorithm is a 'genuine primal–dual one'; it works on a strictly feasible pair (2.19) of conic problems and associates with this pair the generalized Tanabe–Todd–Ye (Tanabe 1988, Todd and Ye 1990) *primal–dual potential function* $p : X^o \times S^o \to \mathbb{R}$ defined by

$$p(x,s) := (\vartheta + \sqrt{\vartheta}) \ln(s^T x) + F(x) + F_*(s) =: p_0(x,s) + \sqrt{\vartheta} \ln(s^T x). \quad (2.21)$$

It is easily seen that p_0 is bounded below on $X^o \times S^o$ and the set of minimizers of p_0 on $X^o \times S^o$ is exactly the primal–dual central path, where p_0 takes the value $p_* = \vartheta \ln(\vartheta) - \vartheta$. It follows that *for* $(x,s) \in X^o \times S^o$, *the duality gap* $s^T x$ *can be bounded in terms of* p:

$$(x,s) \in X^o \times S^o \Rightarrow s^T x \leq \exp\{\vartheta^{-1/2}[p(x,s) - p_*]\}. \quad (2.22)$$

(It can be readily checked – see Proposition 3.1 – that $c^T x - b^T y = s^T x \geq 0$ for any feasible x and (y,s), so $s^T x$ bounds the distance from optimality of both the primal and dual objective function values.)

Hence all we need in order to approach primal–dual optimality is a 'basic primal–dual step': an update $(x,s) \mapsto (x_+, s_+) : X^o \times S^o \to X^o \times S^o$ which 'substantially' reduces the potential p, at least by a positive absolute constant δ. Iterating this update (perhaps augmented by a line search aimed at further reduction in p) starting with a given initial point $(x_0, s_0) \in X^o \times S^o$, we get a sequence of strictly feasible primal solutions x_k and dual slacks $s_k \in \operatorname{int} \mathbf{K}_*$ (which can be immediately extended to dual feasible solutions (y_k, s_k)), such that $p(x_k, s_k) \leq p(x_0, s_0) - k\delta$, which combines with (2.22) to yield the efficiency estimate

$$s_k^T x_k \leq \exp\{\vartheta^{-1/2}[p_0(x_0, s_0) - p_*]\} \exp\{-\delta \vartheta^{-1/2} k\} s_0^T x_0.$$

Now it takes only $O(1)\sqrt{\vartheta}$ steps to reduce the (upper bound on the) duality gap by an absolute constant factor, and we end up with complexity bounds almost identical to those in Theorem 2.1.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

It remains to explain how to make a basic primal–dual step. This can be done as follows. With a fixed positive threshold $\bar{\lambda}$, given $(x, s) \in X^o \times S^o$, we linearize the logarithmic term in the potential in x and in s, thus getting the functions

$$\xi \mapsto p^x(\xi) = (\vartheta + \sqrt{\vartheta})\frac{s^T\xi}{s^Tx} + F(\xi) + \text{const}_x : \text{int}\,\mathbf{K} \to \mathbb{R},$$

$$\sigma \mapsto p^s(\sigma) = (\vartheta + \sqrt{\vartheta})\frac{\sigma^Tx}{s^Tx} + F_*(\sigma) + \text{const}_s : \text{int}\,\mathbf{K}_* \to \mathbb{R},$$

which are non-degenerate self-concordant functions on $\mathbf{K}$ and $\mathbf{K}_*$, respectively. We compute the Newton direction $d_x = \text{argmin}_d\{d^T\nabla p^x(x) + \frac{1}{2}d^T\nabla^2 p^x(x)d : x + d \in \text{Aff}(X)\}$ of $p^x\big|_X$ at $\xi = x$ along with the corresponding Newton decrement $\lambda := \lambda(x, p^x\big|_X) = \sqrt{-\nabla p^x(x)^T d_x}$. When $\lambda \geq \bar{\lambda}$, one can set $s_+ = s$ and take for x_+ the damped Newton iterate $x + (1 + \lambda)^{-1}d_x$ of x, the Newton method being applied to $p^x\big|_X$. When $\lambda < \bar{\lambda}$, one can set

$$x_+ = x \quad \text{and} \quad s_+ = \frac{s^Tx}{\vartheta + \sqrt{\vartheta}}[-\nabla F(x) - \nabla^2 F(x)d_x].$$

It can be shown that with a properly chosen *absolute constant* $\bar{\lambda} > 0$, this update indeed ensures that $(x_+, s_+) \in X^o \times S^o$ and $p(x_+, s_+) \leq p(x, s) - \delta$, where $\delta > 0$ depends solely on $\bar{\lambda}$. Note that the same is true for the 'symmetric' updating obtained by similar construction with the primal and dual problems swapped, and one is welcome to use the better (the one with a larger reduction in the potential) of these two updates or their line-search augmentations.

2.4. The calculus of self-concordant barriers

The practical significance of the nice results we have described depends heavily on our ability to equip the problem we are interested in (a convex program (2.1), or a primal–dual pair of conic programs (2.19)) with self-concordant barrier(s). *In principle* this can always be done: every closed convex domain $X \subset \mathbb{R}^n$ admits an $O(1)n$-SCB; when the domain is a cone, this barrier can be chosen to be logarithmically homogeneous. Assuming without loss of generality that X does not contain lines, one can take as such a barrier the function

$$F(x) = O(1)\ln \text{mes}_n\{y : y^T(z - x) \leq 1,\ \forall z \in X\},$$

where mes_n denotes n-dimensional (Jordan or Lebesgue) measure. This function has a transparent geometric interpretation: the set whose measure we are taking is the polar of $X - x$. When X is a cone (closed, convex, containing no lines and with a non-empty interior), the *universal barrier* given by the expression above is automatically logarithmically homogeneous.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

From a practical perspective, the existence theorem just formulated is not of much interest – the universal barrier is usually pretty difficult to compute, and in the rare cases when this is possible, it may be non-optimal in terms of its self-concordance parameter. Fortunately, there exists a simple and fully algorithmic 'calculus' of self-concordant barriers which allows us to build systematically explicit efficiently computable SCBs for seemingly all generic convex programs associated with 'computationally tractable' domains. We start with the list of the most basic rules (essentially, the only ones needed in practice) of 'self-concordant calculus'.

A If F is a ϑ-SCB for X and $\alpha \geq 1$, then αF is an $(\alpha\vartheta)$-SCB for X.

B *Direct products.* Let F_i, $i = 1, \ldots, m$, be ϑ_i-SCBs for closed convex domains $X_i \subset \mathbb{R}^{n_i}$. The 'direct sum' $F(x^1, \ldots, x^m) = \sum_i F_i(x^i)$ of these barriers is a $(\sum_i \vartheta_i)$-SCB for the direct product $X = X_1 \times \cdots \times X_m$ of the sets.

C *Intersection.* Let F_i, $i = 1, \ldots, m$, be ϑ_i-SCBs for closed convex domains $X_i \subset \mathbb{R}^n$, and let the set $X = \bigcap_i X_i$ possess a non-empty interior. Then $F(x) = \sum_i F_i(x)$ is a $(\sum_i \vartheta_i)$-SCB for X.

D *Inverse affine image.* Let F be a ϑ-SCB for a closed convex domain $X \subset \mathbb{R}^n$, and $y \mapsto Ay + b : \mathbb{R}^k \to \mathbb{R}^n$ be an affine mapping whose image intersects $\operatorname{int} X$. Then the function $G(y) = F(Ay + b)$ is a ϑ-SCB for the closed convex domain $Y = \{y : Ay + b \in X\}$.

When the operands in the rules are cones and the original SCBs are logarithmically homogeneous, so are the resulting barriers (in the case of **D**, provided that $b = 0$). All the statements remain true when, instead of SCBs, we are speaking about SC functions; in this case, the parameter-related parts should be skipped, and what remains become statements on preserving self-concordance.

Essentially all we need in addition to the outlined (and nearly evident) elementary calculus rules, are two more advanced rules, as follows.

E *Taking the conic hull.* Let $X \subset \mathbb{R}^n$ be a closed convex domain and let F be a ϑ-SCB for X. Then, with a properly chosen absolute constant κ, the function $F_+(x,t) = \kappa[F(x/t) - 2\vartheta \ln t]$ is a $2\kappa\vartheta$-SCB for the conic hull

$$X_+ := \operatorname{cl}\left\{(x,t) \in \mathbb{R}^n \times \mathbb{R} : t > 0, x/t \in \operatorname{int} X\right\}$$

of X.

To present the last calculus rule, which can be skipped on a first reading, we need to introduce the notion of compatibility, as follows. Let $K \subset \mathbb{R}^N$ and $G_- \subseteq \mathbb{R}^n$ be a closed convex cone and a closed convex domain, respectively, let $\beta \geq 1$, and let $\mathcal{A}(x) : \operatorname{int} G_- \to \mathbb{R}^N$ be a mapping. We say that $\mathcal{A}$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

is *β-compatible with* K, if $\mathcal{A}$ is three times continuously differentiable on $\operatorname{int} G_-$, is K-concave (that is, $D^2\mathcal{A}(x)[h,h] \in -K$ for all $x \in \operatorname{int} G_-$ and all $h \in \mathbb{R}^n$) and

$$D^3\mathcal{A}(x)[h,h,h] \leq_K -3\beta D^2\mathcal{A}(x)[h,h]$$

for all $x \in \operatorname{int} G_-$ and $h \in \mathbb{R}^n$ with $x \pm h \in G_-$, where $a \leq_K b$ means that $b - a \in K$. The calculus rule in question reads as follows.

F Let $G_- \subset \mathbb{R}^n$, $G_+ \subset \mathbb{R}^N$ be closed convex domains and $\mathcal{A} : \operatorname{int} G_- \to \mathbb{R}^N$ be a mapping, β-compatible with the recession cone of G_+, whose image intersects $\operatorname{int} G_+$. Given $\vartheta_\pm$-SCBs $F_\pm$ for G_+ and G_-, respectively, let us define $F : X^o := \{x \in \operatorname{int} G_- : \mathcal{A}(x) \in \operatorname{int} G_+\} \to \mathbb{R}$ by

$$F(x) = F_+(\mathcal{A}(x)) + \beta^2 F_-(x).$$

Then F is a $(\vartheta_+ + \beta^2\vartheta_-)$-SCB for $X = \operatorname{cl} X^o$.

The most non-trivial and important example of a mapping which can be used in the context of rule **E** is the *fractional-quadratic substitution*. Specifically, let T, E, F be Euclidean spaces, let $Q(x,z) : E \times E \to F$ be a symmetric bilinear mapping, and let $A(t)$ be a symmetric linear operator on E, affinely depending on $t \in T$, and such that the bilinear form $Q(A(t)x, z)$ on $E \times E$ is symmetric in x, z for every $t \in T$. Further, let K be a closed convex cone in F such that $Q(x,x) \in K$ for all x, and let H be a closed convex domain in T such that $A(t)$ is positive definite for all $t \in \operatorname{int} H$. It turns out that the mapping $\mathcal{A}(y,x,t) = y - Q([A(t)]^{-1}x, x)$ with the domain $F \times E \times \operatorname{int} H$ is 1-compatible with K.

It turns out (see examples in Nesterov and Nemirovski (1994) and Nemirovski (2004)) that the combination rules **A**–**F** used 'from scratch' (from the sole observation that the function $-\ln x$ is a 1-LHSCB for the non-negative ray) permit one to build 'good' SCBs/LHSCBs almost without calculation for all interesting convex domains/cones, including epigraphs of numerous convex functions (*e.g.*, the elementary univariate functions such as powers and the exponential, and the multivariate p-norms), sets given by finite systems of convex linear and quadratic inequalities, and much more. This list includes, in particular, ϑ-LHSCBs underlying:

(a) the non-negative orthant $\mathbb{R}^n_+$ $(F(x) = -\sum_j \ln x_j,\ \vartheta = n)$,

(b) Lorentz cones

$$\mathbf{L}^q = \{(\xi, x) \in \mathbb{R} \times \mathbb{R}^q : \xi \geq \|x\|_2\} \quad (F(\xi,x) = -\ln(\xi^2 - x^T x),\ \vartheta = 2),$$

(c) semidefinite cones $\mathbf{S}^p_+$ (the cone of all symmetric positive definite matrices of order p) $(F(x) = -\ln\det(x),\ \vartheta = p)$, and

(d) matrix norm cones $\{(\xi, x) : \xi \geq 0, x \in \mathbb{R}^{p\times q} : \xi^2 I_q \succeq x^T x\}$ (assuming without loss of generality $p \geq q$, $F(\xi,x) = -\ln\det(\xi I_q - \xi^{-1}x^T x) - \ln\xi$, $\vartheta = q + 1$).

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

With regard to (a)–(d), (a) is self-evident, and the remaining three barriers can be obtained from (a) and **F**, *without calculations*, via the above result on fractional-quadratic substitution. For example, to get (b), we set $T = F = \mathbb{R}$, $E = \mathbb{R}^n = \mathbb{R}^{n\times 1}$, $Q(x,z) = x^T z$, $A(\xi) = \xi I_n$, $K = H = \mathbb{R}_+$, thus concluding that the mapping $\mathcal{A}(y,x,\xi) = y - \xi^{-1}x^T x$ with the domain $\operatorname{int} G_-$, $G_- = \{y \in \mathbb{R}\} \times \{x \in \mathbb{R}^n\} \times \{\xi \geq 0\}$, taking values in $\mathbb{R}$, is 1-compatible with $K = \mathbb{R}_+$. Applying **F** to G_- and with $G_+ = \mathbb{R}_+$, $F_-(y,x,\xi) = -\ln\xi$, $F_+(s) = -\ln s$, and using (a) and **D** to conclude that F_- and F_+ are SCBs for G_- and G_+ with the parameters $\vartheta_- = \vartheta_+ = 1$, we see that $-\ln(y - \xi^{-1}x^T x) - \ln\xi$ is a 2-SCB for the set $\{(y,x,\xi) : y\xi \geq x^T x, y, \xi \geq 0\}$. It remains to note that $\mathbf{L}_n$ is the inverse affine image of the latter set under the linear mapping $(\xi, x) \mapsto (\xi, x, \xi)$, and to apply **D**.

Note that (a)–(c) combine with **B** to induce LHSCBs for the direct products K of non-negative rays, Lorentz and semidefinite cones. All cones K one can get in this fashion are self-dual, and the resulting barriers F turn out to be 'self-symmetric' ($F_*(\cdot) = F(\cdot) + \text{const}_K$), thus giving rise to primal–dual IPMs for linear, conic quadratic, and semidefinite programming. Moreover, it turns out that the barriers in question are 'optimal', with provably minimum possible values of the self-concordance parameter ϑ.

3. Conic optimization

Here we treat in more detail the case of the primal–dual conic problems in (2.19). We restate the primal problem:

$$(P) \qquad \begin{array}{rl} \min\limits_{x} & c^T x \\ Ax &= b, \\ x &\in \mathbf{K}, \end{array}$$

where again $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and **K** is a closed convex cone in $\mathbb{R}^n$. We call this the conic programming problem in primal or standard form, since when **K** is the non-negative orthant, it becomes the standard-form linear programming problem.

Recall the dual cone defined by

$$\mathbf{K}_* := \{s \in \mathbb{R}^n : s^T x \geq 0, \quad \text{for all } x \in \mathbf{K}\}. \tag{3.1}$$

Then we can construct the conic programming problem in dual form using the same data:

$$(D) \qquad \begin{array}{rl} \max\limits_{y,s} & b^T y \\ A^T y &+ s = c, \\ & s \in \mathbf{K}_*, \end{array}$$

with $y \in \mathbb{R}^m$, where we have introduced the dual slack variable s to make

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

the later analysis cleaner. In terms of the variables y, we have the conic constraints $c - A^T y \in \mathbf{K}_*$, corresponding to the linear inequality constraints $c - A^T y \geq 0$ when (P) is the standard linear programming problem.

In fact, it is easy to see that (D) is the Lagrangian dual

$$\max_y \left\{ \min_{x \in \mathbf{K}} \{c^T x - (Ax - b)^T y\} \right\}$$

of (P), using the fact that $\min\{u^T x : x \in \mathbf{K}\}$ is 0 if $u \in \mathbf{K}_*$ and $-\infty$ otherwise. We can also easily check weak duality, as follows.

Proposition 3.1. If x is feasible in (P) and (y, s) in (D), then

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

with equality if and only if $s^T x = 0$.

Proof. Indeed,

$$c^T x - b^T y = (A^T y + s)^T x - (Ax)^T y = s^T x \geq 0, \tag{3.2}$$

with the inequality following from the definition of the dual cone. □

In the case of linear programming, when $\mathbf{K}$ (and then also $\mathbf{K}_*$) is the nonnegative orthant, then whenever (P) or (D) is feasible, we have equality of their optimal values (possibly $\pm\infty$), and if both are feasible, we have strong duality: no duality gap, and both optimal values attained.

In the case of more general conic programming, these properties no longer hold (we will provide examples in the next subsection), and we need further regularity conditions. Nesterov and Nemirovski (1994, Theorem 4.2.1) derive the next result.

Theorem 3.1. If either (P) or (D) is bounded and has a strictly feasible solution (*i.e.*, a feasible solution where x (respectively, s) lies in the interior of $\mathbf{K}$ (respectively, $\mathbf{K}_*$)), then their optimal values are equal. If both have strictly feasible solutions, then strong duality holds.

The existence of an easily stated dual problem provides one motivation for considering problems in conic form (but its usefulness depends on having a closed form expression for the dual cone). We will also see that many important applications naturally lead to conic optimization problems. Finally, there are efficient primal–dual interior-point methods for this class of problems, or at least for important subclasses.

In Section 3.1, we consider several interesting special cases of (P) and (D). Section 3.2 discusses path-following interior-point methods. In Section 3.3, we consider a special class of conic optimization problems allowing symmetric primal–dual methods. Finally, Section 3.4 addresses recent extensions.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

3.1. Examples of conic programming problems

First of all, it is worth pointing out that any convex programming problem can be put into conic form. Without loss of generality, after introducing a new variable if necessary to represent a convex nonlinear objective function, we can assume that the original problem is

$$\min_x \{c^T x : x \in X\},$$

with X a closed convex subset of $\mathbb{R}^n$. This is equivalent to the conic optimization problem, but for one dimension higher:

$$\min_{x,\xi} \{c^T x : \xi = 1,\ (x,\xi) \in \mathbf{K}\},$$

where $\mathbf{K} := \mathrm{cl}\{(x,\xi) \in \mathbb{R}^n \times \mathbb{R} : \xi > 0,\ x/\xi \in X\}$. However, this formal equivalence may not be very useful practically, partly because $\mathbf{K}$ and $\mathbf{K}_*$ may not be easy to work with. More importantly, even if we have a good self-concordant barrier for X, it may be hard to obtain an efficient self-concordant barrier for $\mathbf{K}$ (although general, if usually overconservative, procedures are available: see rule **E** in Section 2.4 and Freund, Jarre and Schaible (1996)).

Let us turn to examples with very concrete and useful cones. The first example is of course linear programming, where $\mathbf{K} = \mathbb{R}^n_+$. Then it is easy to see that $\mathbf{K}_*$ is also $\mathbb{R}^n_+$, and so the dual constraints are just $A^T y \le c$. The significance and wide applicability of linear programming are well known. Our first case with a non-polyhedral cone is what is known as *second-order cone programming* (SOCP). Here $\mathbf{K}$ is a second-order, or Lorentz, or 'ice-cream' cone,

$$\mathbf{L}^q := \{(\xi,\bar{x}) \in \mathbb{R} \times \mathbb{R}^q : \xi \ge \|\bar{x}\|\},$$

or the product of such cones. It is not hard to see, using the Cauchy–Schwarz inequality, that such cones are also self-dual, *i.e.*, equal to their duals. We now provide an example showing the usefulness of SOCP problems (many more examples can be found in Lobo, Vandenberghe, Boyd and Lebret (1998) and in Ben-Tal and Nemirovski (2001)), and also a particular instance demonstrating that strong duality does not always hold for such problems.

Suppose we are interested in solving a linear programming problem $\max\{b^T y : A^T y \le c\}$, but the constraints are not known exactly: for the jth constraint $a_j^T y \le c_j$, we just know that $(c_j; a_j) \in \{(\bar{c}_j; \bar{a}_j) + P_j u_j : \|u_j\| \le 1\}$, an ellipsoidal *uncertainty set* centred at the nominal values $(\bar{c}_j; \bar{a}_j)$. (We use the MATLAB-like notation $(u; v)$ to denote the concatenation of the vectors u and v.) Here P_j is a suitable matrix that determines the shape and size of this uncertainty set. We would like to choose our decision variable y so that it is feasible no matter what the constraint coefficients turn out

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

to be, as long as they are in the corresponding uncertainty sets; with this limitation, we would like to maximize $b^T y$. This is (a particular case of) the so-called *robust* linear programming problem. Since the minimum of $c_j - a_j^T y = (c_j; a_j)^T (1; -y)$ over the jth uncertainty set is

$$\begin{aligned}(\bar{c}_j; \bar{a}_j)^T (1; -y) + \min\{(P_j u_j)^T (1; -y) : \|u_j\| \le 1\} \\ = (\bar{c}_j; \bar{a}_j)^T (1; -y) - \|P_j^T (1; -y)\|,\end{aligned}$$

this robust linear programming problem can be formulated as

$$\begin{array}{lrcll} \max & b^T y & & & \\ & -\bar{c}_j + \bar{a}_j^T y & + & s_{j1} = 0, & j = 1, \ldots, m, \\ & P_j^T (1; -y) & + & \bar{s}_j = 0, & j = 1, \ldots, m, \\ & & & (s_{j1}; \bar{s}_j) \in \mathbf{K}_j, & j = 1, \ldots, m, \end{array}$$

where each $\mathbf{K}_j$ is a second-order cone of appropriate dimension. This is a SOCP problem in dual form.

Next, consider the SOCP problem in dual form with data

$$A = \begin{pmatrix} -1 & 0 & -1 \\ -1 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad c = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

and $\mathbf{K}$ the second-order cone in $\mathbb{R}^3$. It can be checked that y is feasible in (D) if and only if y_1 and y_2 are positive, and $4y_1 y_2 \ge 1$. Subject to these constraints, we wish to maximize $-y_1$, so the problem is feasible, with objective function bounded above, but there is no optimal solution! In this case, the optimal values of primal and dual are equal: $(\xi; \bar{x}) = (1/2; 0; 1/2)$ is the unique feasible solution to (P), with zero objective function value.

The second class of non-polyhedral cones we consider gives rise to semidefinite programming problems. These correspond to the case when $\mathbf{K}$ is the cone of positive semidefinite matrices of a given order (or possibly a Cartesian product of such cones). Here we will restrict ourselves to the case of real symmetric matrices, and we use $\mathbf{S}^p$ to denote the space of all such matrices of order p. Of course, this can be identified with $\mathbb{R}^n$ for $n := p(p+1)/2$, by making a vector from the entries m_{ii} and $\sqrt{2} m_{ij}$, $i < j$. We use the factor $\sqrt{2}$ so that the usual scalar product of the vectors corresponding to two symmetric matrices U and V equals the Frobenius scalar product

$$U \bullet V := \mathrm{Tr}(U^T V) = \sum_{i,j} u_{ij} v_{ij}$$

of the matrices. However, we will just state these problems in terms of the matrices for clarity. We write $\mathbf{S}^p_+$ for the cone of (real symmetric) positive semidefinite matrices, and sometimes write $X \succeq 0$ to denote that X lies in

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

this cone for appropriate p. As in the case of the non-negative orthant and the second-order cone, $\mathbf{S}^p_+$ is self-dual. This can be shown using the spectral decomposition of a symmetric matrix. We note that the case of complex Hermitian positive semidefinite matrices can also be considered, and this is important in some applications.

In matrix form, the constraint $AX = b$ is defined using an operator A from $\mathbf{S}^p$ to $\mathbb{R}^m$, and we can find matrices $A_i \in \mathbf{S}^p$, $i = 1, \ldots, m$, so that $AX = (A_i \bullet X)_{i=1}^m$; A^T is then the adjoint operator from $\mathbb{R}^m$ to $\mathbf{S}^p$ defined by $A^T y = \sum_i y_i A_i$. The primal and dual semidefinite programming problems then become

$$\min C \bullet X, \quad A_i \bullet X = b_i,\ i = 1, \ldots, m, \quad X \succeq 0, \tag{3.3}$$

and

$$\max b^T y, \quad \sum_i y_i A_i + S = C, \quad S \succeq 0. \tag{3.4}$$

Once again, we give examples of the importance of this class of conic optimization problems, and also an instance demonstrating the failure of strong duality.

Let us first describe a very simple example that illustrates techniques used in optimal control. Suppose we have a linear dynamical system

$$\dot{z}(t) = A(t)z(t),$$

where the $p \times p$ matrices $A(t)$ are known to lie in the convex hull of a number $A_1, \ldots, A_k$ of given matrices. We want conditions that guarantee that the trajectories of this system stay bounded. Certainly a sufficient condition is that there is a positive definite matrix $Y \in \mathbf{S}^p$ so that the Lyapunov function $L(z(t)) := z(t)^T Y z(t)$ remains bounded. And this will hold as long as $\dot{L}(z(t)) \leq 0$. Now using the dynamical system, we find that

$$\dot{L}(z(t)) = z(t)^T (A(t)^T Y + Y A(t)) z(t),$$

and since we do not know where the current state might be, we want $-A(t)^T Y - Y A(t)$ to be positive semidefinite whatever $A(t)$ is, and so we are led to the constraints

$$-A_i^T Y - Y A_i \succeq 0,\ i = 1, \ldots, k, \quad Y - I_p \succeq 0,$$

where the last constraint ensures that Y is positive definite. (Here I_p denotes the identity matrix of order p. Since the first constraints are homogeneous in Y, we can assume that Y is scaled so its minimum eigenvalue is at least 1.) To make an optimization problem, we could for instance minimize the condition number of Y by adding the constraint $\eta I_p - Y \succeq 0$ and then maximizing $-\eta$. This is a semidefinite programming problem in dual form. Note that the variables y are the entries of the symmetric matrix Y and the

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

scalar η, and the cone is the product of $k+2$ copies of $\mathbf{S}^p_+$. We can similarly find sufficient conditions for $z(t)$ to decay exponentially to zero.

Our second example is a relaxation of a quadratic optimization problem with quadratic constraints. Notice that we did not stipulate that the problem be convex, so we can include constraints like $x_j^2 = x_j$, which implies that x_j is 0 or 1, *i.e.*, we have included binary integer programming problems. Any quadratic function can be written as a linear function of a certain symmetric matrix (depending quadratically on x). Specifically, we see that

$$\begin{aligned}\alpha + 2b^T x + x^T C x &= \begin{pmatrix}1\\ x\end{pmatrix}^T \begin{pmatrix}\alpha & b^T\\ b & C\end{pmatrix}\begin{pmatrix}1\\ x\end{pmatrix}\\ &= \begin{pmatrix}\alpha & b^T\\ b & C\end{pmatrix} \bullet \left(\begin{pmatrix}1\\ x\end{pmatrix}\begin{pmatrix}1\\ x\end{pmatrix}^T\right)\\ &= \begin{pmatrix}\alpha & b^T\\ b & C\end{pmatrix} \bullet \begin{pmatrix}1 & x^T\\ x & xx^T\end{pmatrix}.\end{aligned}$$

The set of all matrices $\left(\begin{smallmatrix}1 & x^T\\ x & xx^T\end{smallmatrix}\right)$ is certainly a subset of the set of all positive semidefinite matrices with top left entry equal to 1, and so we can obtain a relaxation of the original hard problem in x by optimizing over a matrix X that is subject to the constraints defining this superset. This technique has been very successful in a number of combinatorial problems, and has led to worthwhile approximations to the stable set problem, various satisfiability problems, and notably the max-cut problem. Further details can be found, for example, in Goemans (1997) and Ben-Tal and Nemirovski (2001).

Let us give an example of two dual semidefinite programming problems where strong duality fails. The primal problem is

$$\min_{X \succeq 0} \begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{pmatrix} \bullet X, \quad \begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} \bullet X = 0, \quad \begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 2\end{pmatrix} \bullet X = 2,$$

where the first constraint implies that x_{11}, and hence x_{12} and x_{21}, are zero, and so the second constraint implies that x_{33} is 1. Hence one optimal solution is $X = \mathrm{Diag}(0; 0; 1)$ with *optimal value* 1. The dual problem is

$$\max\ 2y_2, \quad S = \begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 1\end{pmatrix} - y_1 \begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{pmatrix} - y_2 \begin{pmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 2\end{pmatrix} \succeq 0,$$

so the dual slack matrix S has $s_{22} = 0$, implying that s_{12} and s_{21} must be zero, so y_2 must be zero. So an optimal solution is $y = (0; 0)$ with *optimal value* 0. Hence, while both problems have optimal solutions, their optimal values are not equal. Note that neither problem has a strictly feasible solution, and arbitrary small perturbations in the data can make the optimal values jump.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

3.2. Basic interior-point methods for conic problems

Recall that, for conic problems, we want to use logarithmically homogeneous SCBs, those satisfying (2.16):

$$F(\tau x) = F(x) - \vartheta \ln \tau.$$

Examples of such ϑ-LHSCBs are

$$\begin{aligned} F(x) &:= -\sum_j \ln x_j, && x \in \operatorname{int} \mathbb{R}^n_+, \\ F(\xi;\bar{x}) &:= -\ln(\xi^2 - \|\bar{x}\|^2), && (\xi;\bar{x}) \in \operatorname{int} \mathbf{L}^q, \\ F(X) &:= -\ln \det X, && X \in \operatorname{int} \mathbf{S}^p_+, \end{aligned}$$

as in Section 2.4, with values of ϑ equal to n, 2, and p respectively. Each of these cones is self-dual, and it is easy to check that the corresponding dual barriers are $F_*(s) = F(s) - n$, $F_*(\sigma;\bar{s}) = F(\sigma;\bar{s}) + 2\ln 2 - 2$, and $F_*(S) = F(S) - p$.

Henceforth, F and F_* are ϑ-LHSCBs for the cones $\mathbf{K}$ and $\mathbf{K}_*$ respectively. The key properties of such functions are listed after (2.16) and in Proposition 2.1, and from these we easily obtain the following result.

Proposition 3.2. For $x \in \operatorname{int} K$, $s \in \operatorname{int} K_*$, and positive t, we have

$$s + t^{-1}\nabla F(x) = 0 \quad \text{if and only if} \quad x + t^{-1}\nabla F_*(s) = 0,$$

and if these hold,

$$s^T x = t^{-1}\vartheta \quad \text{and} \quad t^{-1}\nabla^2 F_*(s) = [t^{-1}\nabla^2 F(x)]^{-1}. \tag{3.5}$$

Proof. If $ts = -\nabla F(x)$, $x = -\nabla F_*(ts)$ since $-\nabla F$ and $-\nabla F_*$ are inverse bijections. Using the homogeneity of ∇F_*, we obtain $x + t^{-1}\nabla F_*(s) = 0$. The reverse implication follows the same reasoning. If $ts = -\nabla F(x)$, then $s^T x = -t^{-1}\nabla F(x)^T x = t^{-1}\vartheta$ by (2.17) and $\nabla^2 F_*(ts) = [\nabla^2 F(x)]^{-1}$, and the final claim follows from the homogeneity of $\nabla^2 F_*$ of degree -2. □

We now examine in more detail the path-following methods described in Section 2.3, both to see the computation involved and to see how these basic methods can be modified in some cases for increased efficiency. We assume that both (P) and (D) have strictly feasible solutions available. As we noted, the basic primal path-following algorithm can be applied to the restriction of F to the relative interior of $\{x \in \mathbf{K} : Ax = b\}$, which amounts to tracing the path of solutions for positive t to the primal barrier problems:

$$(PB_t) \qquad \begin{aligned} \min_x \quad & tc^T x + F(x) \\ & Ax = b, \\ & x \in \mathbf{K}. \end{aligned}$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

If we associate Lagrange multipliers $\lambda \in \mathbb{R}^m$ with the constraints, and then define $y := -t^{-1}\lambda$, we see that the optimality conditions for (PB_t) are

$$t(c - A^T y) + \nabla F(x) = 0, \quad Ax = b,\ x \in \operatorname{int} \mathbf{K}.$$

Since $-\nabla F$ maps $\operatorname{int} \mathbf{K}$ into $\operatorname{int} \mathbf{K}_*$, we see that $s := c - A^T y$ lies in $\operatorname{int} \mathbf{K}_*$, and so we have

$$\begin{aligned} A^T y \ + \ s &= c, \quad s \in \operatorname{int} \mathbf{K}_*, \\ Ax \qquad\quad &= b, \quad x \in \operatorname{int} \mathbf{K}, \\ \nabla F(x) \qquad + ts &= 0. \end{aligned} \tag{3.6}$$

These equations define the primal–dual central path $\{(x_*(t), s_*(t))\}$ as in Section 2.3. Note also that, using (3.2) and (3.5), the duality gap associated with $x_*(t)$ and $(y_*(t), s_*(t))$ is $s_*(t)^T x_*(t) = t^{-1}\vartheta$. In view of Proposition 3.2, the conditions above are remarkably symmetric. Indeed, let us consider the dual barrier problem

$$(DB_t) \qquad \begin{aligned} \min_{y,s} \quad -tb^T y \ + \ & F_*(s) \\ A^T y \ + \ & s = c, \\ & s \in \operatorname{int} \mathbf{K}_*. \end{aligned}$$

If we associate Lagrange multipliers $\mu \in \mathbb{R}^n$ with the constraints, and then define $x := t^{-1}\mu$, we see that the optimality conditions for (DB_t) are

$$-tb + tAx = 0, \quad \nabla F_*(s) + tx = 0, \quad A^T y + s = c,\ s \in \operatorname{int} \mathbf{K}_*.$$

We can now conclude that $x \in \operatorname{int} \mathbf{K}$, and so the optimality conditions can be written as (3.6) again, where the last equation is replaced by its equivalent form $tx + \nabla F_*(s) = 0$.

This nice symmetry is not preserved at first sight when we consider Newton-like algorithms to trace the central path. Suppose we have strictly feasible solutions x and (y, s) to (P) and (D), approximating a point on the central path: $(x, y, s) \approx (x_*(t), y_*(t), s_*(t))$ for some $t > 0$. We wish to find strictly feasible points approximating a point further along the central path, say corresponding to $t_+ > t$. Let us make a quadratic approximation to the objective function in (PB_{t_+}); for future analysis, we use the Hessian of F at a point $v \in \operatorname{int} \mathbf{K}$ which may or may not equal x. If we let the variable be $x_+ =: x + \Delta x$, we have

$$(PQP) \qquad \begin{aligned} \min_{\Delta x} \quad & t_+ c^T \Delta x + \nabla F(x)^T \Delta x + \tfrac{1}{2} \Delta x^T \nabla^2 F(v) \Delta x \\ & A\Delta x = 0. \end{aligned}$$

Let $\bar{\lambda} \in \mathbb{R}^m$ be the Lagrange multipliers for this problem, and define $\bar{y}_+ := -t_+^{-1}\bar{\lambda}$. Then the optimality conditions for (PQP) can be written as

$$t_+(c - A^T \bar{y}_+) + \nabla F(x) + \nabla^2 F(v)\Delta x = 0, \quad A\Delta x = 0,$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

and if we define $\overline{\Delta y} := \bar{y}_+ - y$ and $\overline{\Delta s} := -A^T\overline{\Delta y}$, we obtain

$$(PQPOC) \qquad \begin{array}{rrrl} & & A^T\overline{\Delta y} + \overline{\Delta s} &= 0, \\ & A\Delta x & & = 0, \\ t_+^{-1}\nabla^2 F(v)\Delta x & & + \overline{\Delta s} &= -s - t_+^{-1}\nabla F(x). \end{array}$$

This system also arises as giving the Newton step for (3.6) with t_+ replacing t, where $\nabla^2 F(v)$ is used instead of $\nabla^2 F(x)$. We will discuss the solution of this system of equations after comparing it with the corresponding system for the dual problem.

Hence let us make a quadratic approximation to the objective function of (DB_{t_+}), again evaluating the Hessian of F_* at a point $u \in \operatorname{int} \mathbf{K}_*$ which may or may not equal s for future analysis. If we make the variables of the problem $y_+ =: y + \Delta y$ and $s_+ =: s + \Delta s$, we obtain

$$(DQP) \qquad \begin{array}{rll} \min\limits_{\Delta y, \Delta s} & -t_+ b^T \Delta y + \nabla F_*(s)^T \Delta s + \frac{1}{2}\Delta s^T \nabla^2 F_*(u)\Delta s & \\ & A^T \Delta y + \Delta s = 0. & \end{array}$$

Let $\bar{\mu} \in \mathbb{R}^n$ be the Lagrange multipliers for (DQP), and define $\bar{x}_+ := t_+^{-1}\bar{\mu}$. Then the optimality conditions become

$$-t_+(b - A\bar{x}_+) = 0, \quad \nabla F_*(s) + \nabla^2 F_*(u)\Delta s + t_+\bar{x}_+ = 0, \quad A^T\Delta y + \Delta s = 0.$$

Writing $\overline{\Delta x} := \bar{x}_+ - x$, we obtain

$$(DQPOC) \qquad \begin{array}{rrrl} & A^T\Delta y + & \Delta s &= 0, \\ A\overline{\Delta x} & & & = 0, \\ \overline{\Delta x} & + t_+^{-1}\nabla^2 F_*(u)\Delta s & &= -x - t_+^{-1}\nabla F_*(s). \end{array}$$

We note that this system can also be viewed as a Newton-like system for a modified form of (3.6), where $t_+ x + \nabla F_*(s) = 0$ replaces $ts + \nabla F(x) = 0$ as the final equation. From this viewpoint, a natural way to adapt the methods to the case where x or (y, s) is not a strictly feasible solution of (P) or (D) is apparent. As long as $x \in \operatorname{int} \mathbf{K}$ and $s \in \operatorname{int} \mathbf{K}_*$, we can define search directions using $(PQPOC)$ or $(DQPOC)$ where the zero right-hand sides in the first two equations are replaced by the appropriate residuals in the equality constraints. These so-called *infeasible-interior-point methods* are simple and much used in practice, although their analysis is hard. Polynomial-time complexity for linear programming was established by Zhang (1994) and Mizuno (1994). The other possibility to deal with infeasible iterates is to use a self-dual embedding: see the references in Section 2.3.

There is clearly a strong similarity between the conditions $(PQPOC)$ and $(DQPOC)$, but they will only define the same directions ($\overline{\Delta x} = \Delta x$ and

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

$(\overline{\Delta y}, \overline{\Delta s}) = (\Delta y, \Delta s)$) under rather strong hypotheses, for example, if

$$t_+^{-1}\nabla^2 F_*(u) = [t_+^{-1}\nabla^2 F(v)]^{-1}, \tag{3.7}$$

$$t_+^{-1}\nabla^2 F(v)(-x - t_+^{-1}\nabla F_*(s)) = -s - t_+^{-1}\nabla F(x). \tag{3.8}$$

Using Proposition 3.2, this holds if $v = x = x_*(t_+)$ and $u = s = s_*(t_+)$, but in this case all the directions are zero and it is pointless to solve the systems! (It also holds if $(t_+/t)^{1/2}v = x = x_*(t)$ and $(t_+/t)^{1/2}u = s = s_*(t)$, again a very special situation.) In the next subsection, we will describe situations where the equations above hold for any x and s by suitable choice of u and v.

The solution to $(PQPOC)$ can be obtained by solving for $\overline{\Delta s}$ in terms of $\overline{\Delta y}$ and then Δx in terms of $\overline{\Delta s}$. Substituting in the equation $A\Delta x = 0$, we see that we need to solve

$$(A[\nabla^2 F(v)]^{-1}A^T)\overline{\Delta y} = A[\nabla^2 F(v)]^{-1}(s + t_+^{-1}\nabla F(x). \tag{3.9}$$

Let us examine the form of these equations in the cases of linear and semidefinite programming. (The analysis for the second-order cone is also straightforward, but the formulae are rather cumbersome.) In the first case, $\nabla^2 F(v)$ for the usual log barrier function becomes $[\mathrm{Diag}(v)]^{-2}$ and $\nabla F(x)$ becomes $-[\mathrm{Diag}(x)]^{-1}e$, with e a vector of ones. Hence (3.9) can be written

$$(A[\mathrm{Diag}(v)]^2 A^T)\overline{\Delta y} = A[\mathrm{Diag}(v)]^2 s - t_+^{-1}A[\mathrm{Diag}(v)]^2[\mathrm{Diag}(x)]^{-1}e.$$

In the large sparse case, the coefficient matrix in the equation above can be formed fairly cheaply and usually retains some of the sparsity of A; its Cholesky factorization can be obtained somewhat cheaply. The typically very low number of iterations required then compensates to a large extent for the iterations being considerably more expensive than pivots in the simplex method. (Indeed, for the primal–dual algorithms of the next subsection, 10 to 50 iterations almost always provide 8 digits of accuracy, even for very large LP problems.)

In the case of semidefinite programming, A can be thought of as an operator from symmetric matrices to $\mathbb{R}^m$ and A^T as the adjoint operator from $\mathbb{R}^m$ to the space of symmetric matrices; see the discussion preceding (3.3). With the usual log determinant barrier function, $\nabla^2 F(V)$ maps a symmetric matrix Z to $V^{-1}ZV^{-1}$ and $\nabla F(X)$ is $-X^{-1}$, so (3.9) becomes

$$A_i \bullet \sum_j (VA_jV)\overline{\Delta y}_j = A_i \bullet (VSV - t_+^{-1}VX^{-1}V), \quad i = 1, \ldots, m.$$

If we take $V = X$, as seems natural, then there is a large cost in even forming the $m \times m$ matrix with ijth entry $A_i \bullet (XA_jX)$: the A_is may well be sparse, but X is frequently not, and then we must compute the Cholesky factorization of the resulting usually dense matrix.

Let us return to the general case. Computing Δx in this way using $v = x$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

gives the primal path-following algorithm. We could also use $\overline{\Delta y}$ and $\overline{\Delta s}$ to update the dual solution, but it is easily seen that in fact Δx is independent of y and s as long as $A^T y + s = c$, so that the 'true' iterates are in x-space. However, updating the dual solution (if feasible) does give an easy way to determine the quality of the primal points generated. If the Newton decrement for t (*i.e.*, $\sqrt{\Delta x^T \nabla^2 F(x) \Delta x}$, where Δx is computed with $t_+ = t$) is small, then updating t_+ as in (2.14) and then using a damped Newton step will yield an updated primal point x_+ at which the Newton decrement for t_+ is also small (and the updated dual solution will be feasible). In practice, heuristics may be used to choose much longer steps and accept points whose Newton decrement is much larger.

A similar analysis for (*DQPOC*) leads to the equations

$$(A\nabla^2 F_*(u)A^T)\Delta y = t_+ Ax + A\nabla F_*(s).$$

Here it is more apparent that the dual direction $(\Delta y, \Delta s)$ is independent of x as long as $Ax = b$, so this is a pure dual path-following method, although again primal iterates can be carried along to assess the quality of the dual iterates. In the case of linear programming, the coefficient matrix takes the form $A[\mathrm{Diag}(u)]^{-2}A^T$, while for semidefinite programming it becomes $(A_i \bullet (U^{-1}A_j U^{-1}))_{i,j=1}^m$.

In the next subsection we consider the case that leads to a symmetric primal–dual path-following algorithm. This requires the notion of self-scaled barrier introduced by Nesterov; further details can be found in Nesterov and Todd (1997, 1998).

3.3. Self-scaled barriers and cones and symmetric primal–dual algorithms

Let us now consider barriers that satisfy a further property: a ϑ-LHSCB F for $\mathbf{K}$ is called a *ϑ-self-scaled barrier* (ϑ-SSB) if, for all $v \in \operatorname{int}\mathbf{K}$, $\nabla^2 F(v)$ maps $\operatorname{int}\mathbf{K}$ to $\operatorname{int}\mathbf{K}_*$ and

$$(\forall v, x \in \operatorname{int}\mathbf{K})\, F_*(\nabla^2 F(v)x) = F(x) - 2F(v) - \vartheta. \tag{3.10}$$

If a cone admits such an SSB, we call it a self-scaled cone. It is easy to check that the three barriers we introduced above for the non-negative, Lorentz, and semidefinite cones are all self-scaled, and so these cones are self-scaled. Moreover, in these examples, v can be chosen so that $\nabla^2 F(v)$ is the identity, so (as we saw) F_* differs from F by a constant.

The condition above implies many other strong properties: the dual barrier F_* is also self-scaled; for all $v \in \operatorname{int}\mathbf{K}$, $\nabla^2 F(v)$ maps $\operatorname{int}\mathbf{K}$ *onto* $\operatorname{int}\mathbf{K}_*$; and we have the following result.

Theorem 3.2. If F is a ϑ-SSB for $\mathbf{K}$, then for every $x \in \operatorname{int}\mathbf{K}$ and $s \in \operatorname{int}\mathbf{K}_*$, there is a unique $w \in \operatorname{int}\mathbf{K}$ such that

$$\nabla^2 F(w)x = s.$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

Moreover,

$$\nabla^2 F(w)\nabla F_*(s) = \nabla F(x) \quad \text{and} \quad \nabla^2 F(w)\nabla^2 F_*(s)\nabla^2 F(w) = \nabla^2 F(x).$$

We call w the *scaling point* for x and s. Clearly, $-\nabla F(w)$ is the scaling point (using F_*) for s and x. Tunçel (1998) found a more symmetric form of the equation (3.10) defining self-scaled barriers: if $\nabla^2 F(v)x = -\nabla F(z)$ for $v, x, z \in \operatorname{int} \mathbf{K}$, then $F(v) = (F(x) + F(z))/2$, and by the result above we also have $\nabla^2 F(v)z = -\nabla F(x)$.

The properties above imply that the cone $\mathbf{K}$ is *symmetric*: it is *self-dual*, since $\mathbf{K}$ and $\mathbf{K}_*$ are isomorphic by the non-singular linear mapping $\nabla^2 F(v)$ for any $v \in \operatorname{int} \mathbf{K}$; and it is *homogeneous*, since there is an automorphism of $\mathbf{K}$ taking any point x_1 of $\operatorname{int} \mathbf{K}$ into any other such point x_2. Indeed, we can choose the automorphism $[\nabla^2 F(w_2)]^{-1}\nabla^2 F(w_1)$, where w_i is the scaling point for x_i and some fixed $s \in \operatorname{int} \mathbf{K}_*$, $i = 1, 2$. Symmetric cones have been much studied and even characterized: see the comprehensive book of Faraut and Koranyi (1994). They also coincide with cones of squares in Euclidean Jordan algebras. These connections were established by Güler (1996). Because of this connection, we know that self-scaled cones do not extend far beyond the cones we have considered: non-negative, Lorentz, and semidefinite cones, and Cartesian products of these.

Let us now return to the conditions (3.7) and (3.8) for $(PQPOC)$ and $(DQPOC)$ to define identical directions. If we set $\bar{u} := t_+^{1/2} u$ and $\bar{v} := t_+^{1/2} v$, these can be rewritten as

$$\nabla^2 F_*(\bar{u}) = [\nabla^2 F(\bar{v})]^{-1}, \quad \nabla^2 F(\bar{v})(-x - t_+^{-1}\nabla F_*(s)) = -s - t_+^{-1}\nabla F(x).$$

When F is self-scaled, these conditions can be satisfied by setting $\bar{v}$ to be the scaling point for x and s, and $\bar{u}$ (equal to $-\nabla F(\bar{v})$) to be the scaling point (for F_*) for s and x. (Notice that, if $(x, s) = (x_*(t), s_*(t))$, then these scaling points are $t^{-1/2}x$ and $t^{-1/2}s$ respectively, and, except for a scalar multiple, we come back to the primal (or dual) direction.)

Let us describe the resulting symmetric primal–dual short-step path-following algorithm. We need a symmetric measure of proximity to the central path. Hence, for x and (y, s) strictly feasible solutions to (P) and (D), define

$$t := t(x, s) := \frac{\vartheta}{s^T x} \quad \text{and} \quad \lambda_2(x, s) := \|ts + \nabla F(x)\|_{F,x}.$$

It can be shown (Nesterov and Todd 1998, Section 3) that $\lambda_2(x, s) = \|tx + \nabla F_*(s)\|_{F_*,s}$ also. Suppose x and s are such that

$$\lambda_2(x, s) \leq 0.1,$$

and we choose

$$t_+ := (1 + 0.06\vartheta^{-1/2})t.$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

We compute the scaling point w for x and s, and let Δx, Δy, and Δs be the solution to $(PQPOC)$ with $v := t_+^{-1/2} w$ (or equivalently to $(DQPOC)$ with $u := -t_+^{-1/2} \nabla F(w)$). Finally, we set $x_+ := x + \Delta x$ and $(y_+, s_+) := (y + \Delta y, s + \Delta s)$. It can be shown (Nesterov and Todd 1998, Section 6) that

$$t(x_+, s_+) = t_+ \quad \text{and} \quad \lambda_2(x_+, s_+) \leq 0.1,$$

so we can continue the process.

Theorem 3.3. Suppose (P) and (D) have strictly feasible solutions, and we have a ϑ-SSB F for $\mathbf{K}$. Suppose further we have a strictly feasible pair x_0, (y_0, s_0) for (P) and (D) with $\lambda_2(x_0, s_0) \leq 0.1$. Then the algorithm described above (with x_{k+1} and (y_{k+1}, s_{k+1}) derived from x_k and (y_k, s_k) as are x_+ and (y_+, s_+) from x and (y, s)) is well-defined (all iterates are strictly feasible), maintains closeness to the path ($\lambda_2(x_k, s_k) \leq 0.1$ for all k) and has the efficiency estimate

$$c^T x_k - b^T y_k = s_k^T x_k = \frac{\vartheta}{t_k} \leq s_0^T x_0 \exp\left\{-\frac{0.05}{\sqrt{\vartheta}} k\right\}.$$

Hence, for every $\epsilon > 0$, it takes at most

$$O(1)\sqrt{\vartheta} \ln\left(\frac{s_0^T x_0}{\epsilon}\right)$$

iterations to obtain strictly feasible solutions with duality gap at most ϵ.

Thus we have obtained an algorithm with complexity bounds of the same order as those for the primal path-following method in Theorem 2.1. In fact, the constants are a little worse than those for the primal method. However, it is important to realize that these are worst-case bounds, and that the primal–dual framework is much more conducive to allowing adaptive algorithms that can give much better results in practice: see, *e.g.*, Algorithms 6.2 and 6.3 in Nesterov and Todd (1998). Part of the reason that long-step algorithms are possible in this context is that approximations of F and of $\nabla^2 F$ hold for much larger perturbations of a point $x \in \operatorname{int} \mathbf{K}$. Indeed, results like (2.5) hold true for any perturbation h with $x \pm h \in \operatorname{int} \mathbf{K}$: see Theorems 4.1 and 4.2 of Nesterov and Todd (1997).

There are also symmetric primal–dual potential-reduction algorithms, using the Tanabe–Todd–Ye function (2.21). Note that

$$\nabla_x p(x, s) = \frac{\vartheta + \sqrt{\vartheta}}{s^T x} s + \nabla F(x), \quad \nabla_s p(x, s) = \frac{\vartheta + \sqrt{\vartheta}}{s^T x} x + \nabla F_*(s),$$

and the coefficient of s (or x) is $t_+ := (1 + 1/\sqrt{\vartheta}) t(x, s)$. Thus Newton-like steps to decrease the potential function (where the Hessian is replaced

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

by $t_+\nabla^2 F(w)$) lead to exactly the same search directions as in the path-following algorithm above. Performing a line search on p in those directions leads to a guaranteed decrease of at least 0.24 (for details, see Section 8 of Nesterov and Todd (1997)), and again, this leads to an $O(\sqrt{\vartheta}\ln(s_0^T x_0/\epsilon))$-iteration algorithm from a well-centred initial pair to achieve an ϵ-optimal pair. The big advantage is that now there is no necessity to stay close to the central path, and indeed, the initial pair does not have to be well-centred – the only change is that the complexity bound is modified appropriately.

We now discuss how the scaling point w for x and s can be computed in the case of the non-negative orthant and the semidefinite cone; for the Lorentz cone, the computation is again straightforward but cumbersome. For the non-negative orthant $\mathbb{R}^n_+$, we have $\nabla^2 F(w) = [\mathrm{Diag}(w)]^{-2}$, so we find the scaling point w for positive vectors x and s is given by

$$w = \left(\sqrt{x_j/s_j}\right)_{j=1}^n,$$

so that the equation to be solved for Δy is

$$A\,\mathrm{Diag}(x)\,[\mathrm{Diag}(s)]^{-1}A^T\Delta y = A(x - t_+^{-1}[\mathrm{Diag}(s)]^{-1}e), \tag{3.11}$$

leading to the usual LP primal–dual symmetric search direction. The computation required is of the same order as that for the primal or dual methods.

For the semidefinite cone $\mathbf{S}^p_+$, the defining relation $\nabla^2 F(w)x = s$ becomes $W^{-1}XW^{-1} = S$, or $WSW = X$, for positive definite X and S, from which we find

$$W = S^{-1/2}(S^{1/2}XS^{1/2})^{1/2}S^{-1/2},$$

where $V^{1/2}$ denotes the positive semidefinite square root of a positive semidefinite matrix V. Todd, Toh and Tütüncü (1998) show that W can be computed using two Cholesky factorizations ($X = L_X L_X^T$ and $S = R_S R_S^T$) and one eigenvalue (of $L_X^T S L_X$) or singular value (of $R_S^T L_X$) decomposition. (After W is obtained, Δy (and hence ΔS and ΔX) can be computed using a system like that for the primal or dual barrier method, but with W replacing V or U^{-1}.)

The need for an eigenvalue or singular value decomposition makes each iteration of a (path-following or potential-reduction) interior-point algorithm using the scaling point W quite expensive. While linear and second-order cone programming problems with hundreds of thousands of variables and constraints (with favourable sparsity patterns) can be solved in under 5 minutes on a fairly modest PC, semidefinite programming problems with matrices of order a thousand, even with very favourable structure, can take up to half an hour. When the matrices are of order two thousand, the times increase to an hour even for the simplest such problems.

Alternative methods greatly improve the computational time per iteration. The Jordan algebra approach (Faybusovich 1997, Schmieta and

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

Alizadeh 2001) replaces the last equation in (3.6) by one exhibiting more primal–dual symmetry. For linear programming, this is $x \circ s = t^{-1}e$, where $\circ$ denotes the Hadamard or componentwise product. A Newton step for this leads to the same direction as the self-scaled method. For semidefinite programming, it gives $XS + SX = 2t^{-1}I$. Unfortunately, linearizing this equation to get

$$(\Delta X S + S\Delta X) + (X\Delta S + \Delta S X) = -XS - SX + 2t_+^{-1}I,$$

as proposed by Alizadeh, Haeberly and Overton (1997), leads to a system that requires even more computation than the self-scaled approach, and does not enjoy scale-invariance properties (see, *e.g.*, Todd *et al.* (1998)). Suppose instead the iterates are first scaled (X by pre- and postmultiplying by $S^{1/2}$, and S by pre- and postmultiplying by $S^{-1/2}$) so that the current iterates are transformed into $\tilde{X} = S^{1/2}XS^{1/2}$ and $\tilde{S} = S^{-1/2}SS^{-1/2} = I$. If the Alizadeh–Haeberly–Overton approach is followed in the transformed space, the linearization becomes

$$2\Delta\tilde{X} + (\tilde{X}\Delta\tilde{S} + \Delta\tilde{S}\tilde{X}) = -2\tilde{X} + 2t_+^{-1}I,$$

or in terms of the original variables after transforming back,

$$\Delta X + \frac{1}{2}(X\Delta S S^{-1} + S^{-1}\Delta S X) = -X + t_+^{-1}S^{-1}.$$

Then the search directions can be obtained after solving the $m \times m$ system

$$A_i \bullet \left(\sum_j (XA_jS^{-1})\Delta y_j\right) = A_i \bullet (X - t_+^{-1}S^{-1}), \quad i = 1, \ldots, m.$$

This method was developed independently by Helmberg, Rendl, Vanderbei and Wolkowicz (1996) and Kojima, Shindoh and Hara (1997), and later derived from a different viewpoint by Monteiro (1997). This approach permits the solution of certain problems with matrices of order two thousand (and favourable structure) in under twenty minutes. A pure dual barrier method can also be used successfully on problems of this size with even faster results, but on some problems it seems not as successful as primal–dual methods.

For truly large semidefinite programming problems, either non-interior-point methods need to be used (see, *e.g.*, Section 6.3 in Todd (2001)), or iterative techniques employed to solve approximately the linear systems arising at each iteration (see, *e.g.*, Toh (2007) and Chai and Toh (2007)). For more information on semidefinite programming, the reader can consult Helmberg's page www-user.tu-chemnitz.de/~helmberg/semidef.html; software for linear, second-order cone, and semidefinite programming can be found at the NEOS solvers site neos.mcs.anl.gov/neos/solvers/index.html.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

3.4. Recent developments

In this final subsection, we describe some recent developments in interior-point methods for conic optimization. We concentrate on classes of cones that are more general than self-scaled cones, but that have some structure that may help in developing efficient interior-point algorithms.

The first class of such cones consists of *hyperbolicity cones*. These cones arise in connection with hyperbolic polynomials: a homogeneous polynomial p on $\mathbb{R}^n$ is hyperbolic in direction $d \in \mathbb{R}^n$ if the univariate polynomial $t \mapsto p(x - td)$ has only real roots for every $x \in \mathbb{R}^n$. The associated hyperbolicity cone $\mathbf{K}(p, d)$ is the set of those x for which all these roots are non-negative. These objects were first studied in the context of PDEs, but were introduced to the interior-point community by Güler (1997) because of their generality and nice properties.

The polynomial $p(x) = x_1 x_2 \cdots x_n$ is hyperbolic in direction d for any positive vector $d \in \mathbb{R}^n$, and the associated hyperbolicity cone is the non-negative orthant. The Lorentz cone arises from $x_1^2 - \sum_{j=2}^n x_j^2$, hyperbolic in the direction $d = (1, 0, \ldots, 0)^T$. Finally, if $n = p(p+1)/2$ and we associate $\mathbb{R}^n$ with $\mathbf{S}^p$, the polynomial $\det(X)$ is hyperbolic in the direction of the identity and gives rise to the semidefinite cone. However, the range of hyperbolicity cones is much larger: Güler (1997) shows, for example, that it includes (properly) all homogeneous cones.

The significance of this class of cones for interior-point methods is that $F(x) := -\ln p(x)$ is an m-LHSCB for the cone $\mathbf{K}(p, d)$, where m is the degree of homogeneity of p. This function has very good properties: for any x, $\nabla^2 F(x)$ takes $\operatorname{int} \mathbf{K}(p, d)$ into (but not necessarily onto) the interior of its dual cone; there is a unique scaling point for each $x \in \operatorname{int} \mathbf{K}(p, d)$ and s in the interior of its dual; and F has good 'long-step properties' like those hinted at below Theorem 3.3 for self-scaled barriers. These results were obtained by Güler (1997), who showed that long-step primal potential-reduction algorithms could be extended from self-scaled cones to hyperbolicity cones. However, the dual barrier of a hyperbolic barrier of this kind is itself a hyperbolic barrier only if the original barrier was self-scaled. Hence it seems unlikely that the primal–dual methods of the previous subsection can be extended to general hyperbolicity cones.

Bauschke, Güler, Lewis and Sendov (2001) study hyperbolic polynomials from the viewpoint of convex analysis and hence rederive some of Güler's results. Of more interest in optimization, Renegar (2006) makes use of an important property of hyperbolic polynomials, namely, that if p is hyperbolic in direction d, then so is the directional derivative $d^T \nabla p$, and the hyperbolicity cone of the latter contains that of p. In this way a hierarchy of relaxations of a hyperbolicity cone programming problem can be defined; Renegar suggests a homotopy method to solve the original problem

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

by considering solutions to these relaxed problems. At present, there is no complexity analysis for this approach, but it seems promising.

The second class of cones we wish to mention arises in global polynomial optimization: that is, one seeks the global minimizer p_* of a polynomial function p of n variables, possibly subject to polynomial inequality constraints. Here the functions involved need not be convex, and the problem is NP-hard even for degree-four polynomials, but we would still like to be able to solve (even approximately) small-scale problems. We describe here briefly an approach, introduced by Parrilo (2003) and Lasserre (2001), that uses semidefinite programming problems as approximations.

Let us follow Lasserre in describing a convex formulation of such a polynomial optimization problem. Suppose p is a polynomial of degree $2m$ in n variables. Using the notation $x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and $|\alpha| := \sum_j \alpha_j$, where α is a non-negative integer n-vector, we can associate p with its vector of coefficients $(p_\alpha)_{|\alpha| \le 2m}$, where

$$p(x) = \sum_{|\alpha| \le 2m} p_\alpha x^\alpha.$$

The key idea is to replace an optimization problem over the n-vector x with one over probability measures μ on $\mathbb{R}^n$. Then minimizing p over $\mathbb{R}^n$ can be replaced by minimizing $\int p(x)\,\mathrm{d}\mu(x)$, which is a convex (even linear!) function of the infinite-dimensional variable μ. Moreover, since p is a polynomial, we have

$$\int p(x)\,\mathrm{d}\mu(x) = \sum_{|\alpha| \le 2m} p_\alpha y_\alpha,$$

where y_α is the α-moment of μ, $\int x^\alpha\,\mathrm{d}\mu(x)$. We now have a linear optimization problem over the finite-dimensional vector $(y_\alpha)_{|\alpha| \le 2m}$, with the constraint that this vector be the vector of moments of some probability measure. The constraint can be separated: we need y to be the vector of moments of a Borel measure (this defines a convex cone, the *moment cone*), and $y_0 = 1$ (this requires the measure to be a probability measure).

Unfortunately (as we would expect from the NP-hardness result), this convex cone is hard to deal with: in particular, it is very unlikely that a computationally tractable barrier function for it exists. We would therefore like to approximate it. Here is one necessary condition, based on a large matrix whose entries are the components of y. Let us enumerate all $\binom{m+n}{n}$ monomials x^β with $|\beta| \le m$ and use them to index the rows and columns of a matrix. Let $M_m(y)$ denote the symmetric matrix whose entry in the row corresponding to x^β and column corresponding to x^γ is $y_{\beta+\gamma}$. Then

$$M_m(y) \succeq 0.$$

Indeed, if $(q_\alpha)_{|\alpha| \le m}$ is the vector of coefficients of a polynomial $q(x)$ of

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

degree m, then $q^T M_m(y) q$ is $\int (q(x))^2 \, \mathrm{d}\mu(x)$, which is non-negative. We can then minimize the linear function $\sum p_\alpha y_\alpha$ subject to $y_0 = 1$ and this semidefinite constraint. This is a relaxation of the original polynomial optimization problem and will provide a lower bound.

It turns out that this lower bound is tight exactly when $p(x) - p_*$ (a polynomial that is non-negative everywhere) can be written as a sum of squares. Indeed, finding the smallest $\bar{p}$ such that $p(x) - \bar{p}$ is a sum of squares can be formulated as a semidefinite programming problem, and it is precisely the dual of the problem above. The complication is that, except in very special cases, the set of non-negative polynomials is larger than the set of sums of squares (this is related to Hilbert's 17th problem), but there are results in semi-algebraic geometry that provide ways to attack the problem. Without going into details, we merely note that a sequence of semidefinite programming problems can be formulated, whose optimal values approach p_*, and frequently the value is attained in a finite (and small) number of steps. The disadvantage is that the sizes of these semidefinite problems grow very fast, so that only small-scale problems can be solved. Lasserre (2001) gives results for (constrained) problems with degree up to four and up to 10 variables; Parrilo and Sturmfels (2003) solve (unconstrained) degree four problems in 13 variables and degree six problems in 7 variables in under half an hour. A MATLAB package for solving sum of squares optimization problems using semidefinite programming is available at www.cds.caltech.edu/sostools/.

We described above the polynomial minimization problem as that of minimizing $\sum_{|\alpha| \leq 2m} p_\alpha y_\alpha$, subject to $y_0 = 1$ and $(y_\alpha)_{|\alpha| \leq 2m}$ belonging to the cone of moments (up to degree $2m$) of a Borel measure. It is not hard to see that the corresponding dual cone consists of the coefficients $(q_\alpha)_{|\alpha| \leq 2m}$ of polynomials q of degree at most $2m$ that are non-negative everywhere. These are two dual convex cones, easy to describe, but hard to deal with computationally, that are important in applications. Another such pair of cones arises in copositive programming.

Suppose we wish to minimize the quadratic function $x^T Q x$ over the standard simplex $\{x \in \mathbb{R}^n_+ : e^T x = 1\}$, where $e \in \mathbb{R}^n$ is the vector of ones. This *standard quadratic programming problem* includes the problem of computing a maximum stable set in a graph and can arise in general quadratic optimization as a test for global optimality (see Bomze (1998)). In fact, the standard quadratic programming problem can be written as the conic optimization problem of minimizing $Q \bullet X$ subject to $E \bullet X = 1$ and X lying in the cone of *completely positive* symmetric matrices: those that can be written as JJ^T for a non-negative (entrywise) matrix J. Here $E := ee^T$, the $n \times n$ matrix of ones. This equivalence can be seen by characterizing the extreme solutions of the latter problem, as in Bomze, de Klerk, Roos, Quist and Terlaky (2000). The dual of the completely positive cone is easily shown to be the cone of copositive matrices, *i.e.*, those that are positive

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

semidefinite on the non-negative orthant. In turn, these are related to non-negative quartics: P is copositive if and only if the quartic $\sum_{i,j} p_{ij} z_i^2 z_j^2$ is everywhere non-negative. Hence copositive programming (and so also the standard quadratic programming problem) can be attacked using the techniques discussed above, introduced by Parrilo and Lasserre. This is a topic of considerable recent interest: see Bomze and de Klerk (2002) and the references therein.

4. IPMs for non-convex programming

In this short final section, we sketch the algorithms that have been proposed for general, not necessarily convex, nonlinear programming. For further details, see the survey papers of Forsgren *et al.* (2002) and Gould *et al.* (2005); the issues are also nicely treated in Nocedal and Wright (2006). These methods were inspired by the great success of interior-point methods for specially structured convex problems, and differ in many respects from the earlier barrier methods of the 1960s and 1970s. However, since they are designed for general problems, the motivating concerns are very different from those for convex optimization: global convergence (possibly to an infeasible point which is a local minimizer of some measure of infeasibility) replaces complexity analysis; superlinear convergence, and the resulting careful control of the parameter t, is of considerable interest; step-size control usually involves a merit function; and modifications to Newton systems are often employed to avoid convergence to stationary points that are not local minimizers. There are two families of interior-point methods for nonlinear programming: those based on line searches and those based on trust regions. Here we restrict ourselves to line-search methods as they are closer to what we have discussed for convex problems.

For simplicity, we concentrate on the inequality-constrained problem

$$(NLP) \quad \min f(y), \quad g(y) \leq 0,$$

where $f : \mathbb{R}^m \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}^n$ are twice continuously differentiable functions. Other forms of problem are discussed by many of the authors of the papers cited below, but the main ideas can be illustrated in this framework. The somewhat unconventional notation is chosen to facilitate comparison with the dual linear programming problem, where $f(y) = -b^T y$ and $g(y) = A^T y - c$.

The first step is to introduce slack variables to convert the inequality constraints to the form $g(y) + s = 0$, $s \geq 0$. A barrier method then tries to find approximate solutions to problems of the form

$$(NLB_t) \quad \min tf(y) - \sum_j \ln s_j, \quad g(y) + s = 0 \quad (s > 0),$$

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

for positive parameters t increasing to ∞. If we associate Lagrange multipliers $\lambda \in \mathbb{R}^n$ to the constraints, and then define $x := t^{-1}\lambda$, we find that the optimality conditions for (NLB_t) can be written as

$$\begin{aligned} \nabla f(y) \; + \; \nabla g(y)x \qquad\qquad &= 0, \\ g(y) \qquad\qquad + \; s \; &= 0, \\ \mathrm{Diag}(x)\mathrm{Diag}(s)e \quad &= t^{-1}e. \end{aligned} \tag{4.1}$$

Given a trial solution (y, x, s) with x and s positive, a Newton step towards a solution of (4.1) will move in the direction $(\Delta y, \Delta x, \Delta s)$ satisfying

$$\begin{bmatrix} K & \nabla g(y) & 0 \\ \nabla g(y)^T & 0 & I \\ 0 & \mathrm{Diag}(s) & \mathrm{Diag}(x) \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta x \\ \Delta s \end{pmatrix} = \begin{pmatrix} -\nabla f(y) - \nabla g(y)x \\ -g(y) - s \\ t^{-1}e - \mathrm{Diag}(x)\mathrm{Diag}(s)e \end{pmatrix}, \tag{4.2}$$

where K denotes the Hessian of the Lagrangian function $L(y, x, s) := f(y) + x^T(g(y) + s)$ with respect to y. Using the last set of equations to solve for Δs, we arrive at

$$\begin{bmatrix} K & \nabla g(y) \\ \nabla g(y)^T & -[\mathrm{Diag}(x)]^{-1}\mathrm{Diag}(s) \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta x \end{pmatrix} = \begin{pmatrix} -\nabla f(y) - \nabla g(y)x \\ -g(y) - t^{-1}[\mathrm{Diag}(x)]^{-1}e \end{pmatrix}; \tag{4.3}$$

if we further eliminate Δx, we reach

$$\begin{aligned} &[K + \nabla g(y)\,\mathrm{Diag}(x)[\mathrm{Diag}(s)]^{-1}\nabla g(y)^T]\Delta y \\ &\qquad = -\nabla f(y) - \nabla g(y)[x + t^{-1}s + \mathrm{Diag}(x)[\mathrm{Diag}(s)]^{-1}g(y)]. \end{aligned} \tag{4.4}$$

This reduces to the primal–dual system (3.11) when (NLP) reduces to the linear programming problem $\min\{-b^T y : A^T y - c \leq 0\}$ and when $s = c - A^T y$.

Primal–dual line-search methods start by solving one of the three linear systems above. If the coefficient matrix in (4.4) is positive definite (this is guaranteed when sufficiently close to a local minimizer of (NLP) satisfying the strong second-order sufficient conditions), the resulting solution $(\Delta y, \Delta x, \Delta s)$ is taken as the search direction. Otherwise, most methods modify the system in some way: either a multiple of the identity matrix of order m is added to K, or a multiple of the identity matrix of order n is subtracted from the $(2, 2)$ block in (4.3), for example. The resulting direction Δy can then be shown to be a descent direction for a merit function such as

$$f(y) - t^{-1}\sum_j \ln s_j + \rho\|g(y) + s\|, \tag{4.5}$$

possibly after increasing the positive penalty parameter ρ. A step is then taken along the direction $(\Delta y, \Delta x, \Delta s)$ to ensure 'sufficient' decrease in the merit function.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

In feasible methods (called quasi-feasible methods if there are also equality constraints present that may not be satisfied exactly), s is reset after each iteration to $-g(y)$, so that $s > 0$ forces $g(y) < 0$ for all iterates. This requirement complicates and restricts the line search, but can avoid some undesirable convergence behaviour. Such methods include those of Gay, Overton and Wright (1998), Forsgren *et al.* (2002) (except in their Section 6.4), and the quasi-feasible method of Chen and Goldfarb (2006). The more common infeasible methods allow $g(y) + s$ to be non-zero, and control it implicitly through the merit function: see, *e.g.*, Vanderbei and Shanno (1999) (LOQO), Waltz, Morales, Nocedal and Orban (2006) (KNITRO/DIRECT), Wächter and Biegler (2006) (IPOPT), and the infeasible method of Chen and Goldfarb (2006). Important practical issues such as how the linear systems are modified and solved, how the line searches are performed, how the parameter t is adjusted, and what – if any – back-up techniques are employed if poor convergence is observed, are discussed further in these papers. For example, KNITRO (Byrd *et al.* 2006) reverts to a trust-region interior-point subproblem to ensure global convergence if negative curvature or slow convergence is detected, and IPOPT uses a filter approach instead of a traditional line search with a merit function, and also includes a feasibility restoration phase. Chen and Goldfarb (2006) modify the $(2, 2)$-block of (4.3) to correspond to the Newton system for moving to a local minimizer of the merit function in (4.5) and may also modify the $(1, 1)$-block; they prove strong global convergence properties for both quasi-feasible and infeasible algorithms.

Overall, these methods have proved strongly competitive for general nonlinear programming problems, and research remains very active. Our treatment has only scratched the surface; for details, consult the references cited and the comprehensive survey articles of Forsgren *et al.* (2002) and Gould *et al.* (2005). The software systems mentioned are available (sometimes free) from the NEOS solvers website neos.mcs.anl.gov/neos/solvers/index.html.

5. Summary

Interior-point methods have changed the way we look at optimization problems over the last twenty years. In this paper we have concentrated on convex problems, and in particular on the classes of structured convex problems for which interior-point methods provide provably efficient algorithms. We have highlighted the theory and motivation for these methods and their domains of applicability, and also pointed out new topics of research. Finally, we have sketched very briefly interior-point methods for general nonlinear programming.

Since the field is so active, we conclude by pointing out once more some sources for tracking current research and algorithms: the websites for Opti-

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

mization Online at www.optimization-online.org/ and for the NEOS solvers at neos.mcs.anl.gov/neos/solvers/index.html, and, for semidefinite programming, Helmberg's page at www-user.tu-chemnitz.de/~helmberg/semidef.html.

Acknowledgements

We are very grateful to Renato Monteiro, Jorge Nocedal and Jim Renegar for help with this paper. The second author was partially supported by NSF through grant DMS-0513337 and ONR through grant N00014-02-1-0057.

REFERENCES

F. Alizadeh (1995), 'Interior point methods in semidefinite programming with applications to combinatorial optimization', *SIAM J. Optim.* **5**, 13–51.

F. Alizadeh, J.-P. A. Haeberly and M. L. Overton (1998), 'Primal–dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results', *SIAM J. Optim.* **8**, 746–768.

E. D. Andersen and Y. Ye (1999), 'On a homogeneous algorithm for monotone complementarity system', *Math. Program.* **84**, 375–399.

H. H. Bauschke, O. Güler, A. S. Lewis and H. S. Sendov (2001), 'Hyperbolic polynomials and convex analysis', *Canad. J. Math.* **53**, 470–488.

A. Ben-Tal and A. S. Nemirovski (2001), *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, SIAM, Philadelphia.

R. E. Bixby (2002), 'Solving real-world linear programs: A decade and more of progress', *Oper. Res.* **50**, 3–15.

I. M. Bomze (1988), 'On standard quadratic optimization problems', *J. Global Optim.* **13**, 369–387.

I. M. Bomze and E. de Klerk (2002), 'Solving standard quadratic optimization problems via semidefinite and copositive programming', *J. Global Optim.* **24**, 163–185.

I. M. Bomze, M. Dür, E. de Klerk, C. Roos, A. J. Quist and T. Terlaky (2000), 'On copositive programming and standard quadratic optimization problems', *J. Global Optim.* **18**, 301–320.

S. Boyd and L. Vandenberghe (2004), *Convex Optimization*, Cambridge University Press.

S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan (1994), *Linear Matrix Inequalities in System and Control Theory*, Vol. 15 of *Studies in Applied Mathematics*, SIAM.

R. H. Byrd, J. Nocedal and R. A. Waltz (2006), KNITRO: An integrated package for nonlinear optimization. In *Large-Scale Nonlinear Optimization* (G. di Pillo and M. Roma, eds), Springer, New York, pp. 35–59.

C. W. Carroll (1961), 'The created response surface technique for optimizing nonlinear, restrained systems', *Oper. Res.* **9**, 169–185.

J. S. Chai and K. C. Toh (2007), 'Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming', *Comput. Optim. Appl.* **36**, 221–247.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

L. Chen and D. Goldfarb (2006), 'Interior-point l_2-penalty methods for nonlinear programming with strong convergence properties', *Math. Program.* **108**, 1–36.
R. Courant (1943), 'Variational methods for the solution of problems of equilibrium and vibrations', *Bull. Amer. Math. Soc.* **49**, 1–23.
J. Faraut and A. Koranyi (1994), *Analysis on Symmetric Cones*, Clarendon Press, Oxford.
L. Faybusovich (1997), 'Linear systems in Jordan algebras and primal–dual interior-point algorithms', *J. Comput. Appl. Math.* **86**, 149–175.
A. V. Fiacco and G. P. McCormick (1968), *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley.
A. Forsgren, P. E. Gill and M. H. Wright (2002), 'Interior methods for nonlinear optimization', *SIAM Review* **44**, 525–597.
R. W. Freund, F. Jarre and S. Schaible (1996), 'On self-concordant barrier functions for conic hulls and fractional programming', *Math. Program.* **74**, 237–246.
K. R. Frisch (1955), The logarithmic potential method of convex programming. Memorandum of May 13, University Institute of Economics, Oslo, Norway.
D. M. Gay, M. L. Overton and M. H. Wright (1998), A primal–dual interior method for nonconvex nonlinear programming. In *Advances in Nonlinear Programming* (Y. Yuan, ed.), Kluwer, pp. 31–56.
P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin and M. H. Wright (1986), 'On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method', *Math. Program.* **36**, 183–209.
M. X. Goemans (1997), 'Semidefinite programming in combinatorial optimization', *Math. Program.* **79**, 143–161.
C. C. Gonzaga (1989), An algorithm for solving linear programming problems in $O(n^3L)$ operations. In *Progress in Mathematical Programming: Interior Point and Related Methods* (N. Megiddo, ed.), Springer, New York, pp. 1–28.
N. I. M. Gould, D. Orban and P. L. Toint (2005), Numerical methods for large-scale nonlinear optimization. In *Acta Numerica*, Vol. 14, Cambridge University Press, pp. 299–361.
C. Guéret, C. Prins and M. Sevaux (2002), *Applications of Optimization with XpressMP*, Dash Optimization. Translated and revised by Susanne Heipcke.
O. Güler (1996), 'Barrier functions in interior-point methods', *Math. Oper. Res.* **21**, 860–885.
O. Güler (1997), 'Hyperbolic polynomials and interior-point methods for convex programming', *Math. Oper. Res.* **22**, 350–377.
C. Helmberg, F. Rendl, R. Vanderbei and H. Wolkowicz (1996), 'An interior-point method for semidefinite programming', *SIAM J. Optim.* **6**, 342–361.
N. Karmarkar (1984), 'A new polynomial-time algorithm for linear programming', *Combinatorica* **4**, 373–395.
E. de Klerk, C. Roos and T. Terlaky (1997), 'Initialization in semidefinite programming via a self-dual skew-symmetric embedding', *Oper. Res. Letters* **20**, 213–221.
M. Kojima, S. Mizuno and A. Yoshise (1989), 'A polynomial–time algorithm for a class of linear complementarity problems', *Math. Program.* **44**, 1–26.
M. Kojima, S. Shindoh and S. Hara (1997), 'Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices', *SIAM J. Optim.* **7**, 86–125.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

J. B. Lasserre (2001), 'Global optimization with polynomials and the problem of moments', *SIAM J. Optim.* **11**, 796–817.
A. S. Lewis and M. L. Overton (1996), Eigenvalue optimization. In *Acta Numerica*, Vol. 5, Cambridge University Press, pp. 149–160.
M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret (1998), 'Applications of second-order cone programming', *Linear Algebra Appl.* **284**, 193–228.
Z.-Q. Luo, J. F. Sturm and S. Zhang (2000), 'Conic convex programming and self-dual embedding', *Optim. Methods Software* **14**, 169–218.
N. Megiddo (1989), Pathways to the optimal set in linear programming. In *Progress in Mathematical Programming: Interior Point and Related Methods* (N. Megiddo, ed.), Springer, New York, pp. 131–158.
S. Mizuno (1994), 'Polynomiality of infeasible-interior-point algorithms for linear programming', *Math. Program.* **67**, 109–119.
R. D. C. Monteiro (1997), 'Primal–dual path-following algorithms for semidefinite programming', *SIAM J. Optim.* **7**, 663–678.
R. D. C. Monteiro and I. Adler (1989), 'Interior path following primal–dual algorithms I: Linear programming', *Math. Program.* **44**, 27–41.
S. G. Nash (1998), 'SUMT (revisited)', *Oper. Res.* **46**, 763–775.
A. Nemirovski (2004), *Interior Point Polynomial Time Methods in Convex Programming*. Lecture Notes: www2.isye.gatech.edu/~nemirovs/Lect_IPM.pdf.
A. Nemirovski and L. Tunçel (2005), ' " Cone-free" primal–dual path-following and potential-reduction polynomial time interior-point methods', *Math. Program.* **102**, 261–295.
Y. Nesterov (1997), 'Long-step strategies in interior-point primal–dual methods', *Math. Program.* **76**, 47–94.
Y. Nesterov (2003), *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer, Dordrecht.
Y. Nesterov and A. Nemirovski (1994), *Interior Point Polynomial Time Methods in Convex Programming*, SIAM, Philadelphia.
Y. E. Nesterov and M. J. Todd (1997), 'Self-scaled barriers and interior-point methods for convex programming', *Math. Oper. Res.* **22**, 1–42.
Y. E. Nesterov and M. J. Todd (1998), 'Primal–dual interior-point methods for self-scaled cones', *SIAM J. Optim.* **8**, 324–364.
Y. Nesterov, M. J. Todd and Y. Ye (1999), 'Infeasible-start primal–dual methods and infeasibility detectors for nonlinear programming problems', *Math. Program.* **84**, 227–267.
J. Nocedal and S. J. Wright (2006), *Numerical Optimization*, Springer, New York.
P. A. Parrilo (2003), 'Semidefinite programming relaxations for semialgebraic problems', *Math. Program.* **96**, 293–320.
P. A. Parrilo and B. Sturmfels (2003), Minimizing polynomials. In *Algorithmic and Quantitative Real Algebraic Geometry*, Vol. 60 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, AMS, pp. 83–99.
F. A. Potra and R. Sheng (1998), 'On homogeneous interior-point algorithms for semidefinite programming', *Optim. Methods Software* **9**, 161–184.
J. Renegar (1988), 'A polynomial-time algorithm, based on Newton's method, for linear programming', *Math. Program.* **40**, 59–93.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press

J. Renegar (2001), *A Mathematical View of Interior-Point Methods in Convex Optimization*, SIAM, Philadelphia.
J. Renegar (2006), 'Hyperbolic programs, and their derivative relaxations', *Found. Comput. Math.* **6**, 59–79.
C. Roos, T. Terlaky and J.-P. Vial (1997), *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, Wiley.
S. H. Schmieta and F. Alizadeh (2001), 'Associative and Jordan algebras, and polynomial-time interior-point algorithms for symmetric cones', *Math. Oper. Res.* **26**, 543–564.
K. Tanabe (1988), Centered Newton method for mathematical programming. In *System Modeling and Optimization*, Springer, New York, pp. 197–206.
M. J. Todd (2001), Semidefinite optimization. In *Acta Numerica*, Vol. 10, Cambridge University Press, pp. 515–560.
M. J. Todd and Y. Ye (1990), 'A centered projective algorithm for linear programming', *Math. Oper. Res.* **15**, 508–529.
M. J. Todd, K.-C. Toh and R. H. Tütüncü (1998), 'On the Nesterov–Todd direction in semidefinite programming', *SIAM J. Optim.* **8**, 769–796.
K. C. Toh (2007), 'An inexact primal–dual path-following algorithm for convex quadratic SDP', *Math. Program.* **112**, 221–254.
L. Tunçel (1998), 'Primal–dual symmetry and scale-invariance of interior-point algorithms for convex programming', *Math. Oper. Res.* **23**, 708–718.
R. J. Vanderbei (2007), *Linear Programming: Foundations and Extensions*, Springer, New York.
R. J. Vanderbei and D. F. Shanno (1999), 'An interior-point algorithm for nonconvex nonlinear programming', *Comput. Optim. Appl.* **13**, 231–252.
R. A. Waltz, J. L. Morales, J. Nocedal and D. Orban (2006), 'An interior algorithm for nonlinear optimization that combines line search and trust region steps', *Math. Program.* **107**, 391–408.
A. Wächter and L. T. Biegler (2006), 'On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming', *Math. Program.* **106**, 25–57.
H. Wolkowicz, R. Saigal and L. Vanderberghe, eds (2000), *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, Kluwer, Boston.
M. H. Wright (1992), Interior methods for constrained optimization. In *Acta Numerica*, Vol. 1, Cambridge University Press, pp. 341–407.
S. J. Wright (1997), *Primal–Dual Interior-Point Methods*, SIAM, Philadelphia.
X. Xu, P. F. Hung and Y. Ye (1996), 'A simplified homogeneous self-dual linear programming algorithm and its implementation', *Ann. Oper. Res.* **62**, 151–171.
Y. Ye (1991), 'An $O(n^3L)$ potential reduction algorithm for linear programming', *Math. Program.* **50**, 239–258.
Y. Ye (1997), *Interior Point Algorithms: Theory and Analysis*, Wiley.
Y. Ye, M. J. Todd and S. Mizuno (1994), 'An $O(\sqrt{nL})$-iteration homogeneous and self-dual linear programming algorithm', *Math. Oper. Res.* **19**, 53–67.
Y. Zhang (1994), 'On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem', *SIAM J. Optim.* **4**, 208–227.

https://doi.org/10.1017/S0962492906370018 Published online by Cambridge University Press