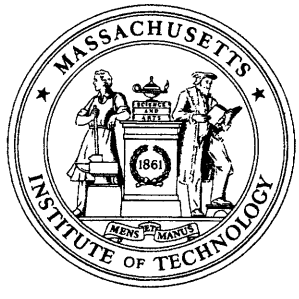
OPERATIONS RESEARCH CENTER

working paper



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Interior Point Methods: Current Status and Future Directions

R.. M. Freund and S. Mizuno

OR 318-96 September 1996

INTERIOR POINT METHODS: CURRENT STATUS AND FUTURE DIRECTIONS

Robert M. Freund[1]
M.I.T.

Shinji Mizuno[2]
The Institute of Statistical Mathematics, Tokyo

September 15, 1996

Abstract

This article provides a synopsis of the major developments in interior point methods for mathematical programming in the last twelve years, and discusses current and future research directions in interior point methods, with a brief selective guide to the research literature.

AMS Subject Classification: 90C, 90C05, 90C60
Keywords: Linear Programming, Newton's Method, Interior Point Methods, Barrier Method, Semidefinite Programming, Self-Concordance

[1]M.I.T. Sloan School of Management, Building E40-149A, Cambridge, MA 02139, USA. email: rfreund@mit.edu

[2]The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106 JAPAN. e-mail: mizuno@ism.ac.jp

1 Introduction and Synopsis

The purpose of this article is twofold: to provide a synopsis of the major developments in interior point methods for mathematical programming in the last twelve years for the researcher who is unfamiliar with interior points, and to discuss current and future research directions in interior point methods for researchers who have some familiarity with interior point methods. Throughout the article, we provide brief selective guides to the most lucid relevant research literature as a means for the uninitiated to become acquainted (but not overwhelmed) with the major developments in interior point methods.

Interior point methods in mathematical programming have been the largest and most dramatic area of research in optimization since the development of the simplex method for linear programming. Over the last twelve years, interior point methods have attracted some of the very best researchers in operations research, applied mathematics, and computer science. Approximately 2,000 papers have been written on the subject following the seminal work of Karmarkar [10] (see for example the `netlib` electronic library initiated by Eberhard Kranich, and the World Wide Web interior point archive `http://www.mcs.anl.gov/home/otc/InteriorPoint/archive.html`). Interior point methods have permanently changed the landscape of mathematical programming theory, practice, and computation. Linear programming is no longer synonymous with the celebrated simplex method, and many researchers now tend to view linear programming more as a special case of nonlinear programming due to these developments.

The pedagogy of interior point methods has lagged the research on interior point methods until quite recently, partly because these methods (i) use more advanced mathematical tools than do pivoting/simplex methods, (ii) their mathematical analysis is typically much more complicated, and (iii) the methods are less amenable to geometric intuition. For most of the last twelve years, educators have struggled with issues of how and where to introduce interior point methods into the curriculum of linear and/or nonlinear programming, and how to cogently exposit interior point methods to students (and fellow researchers). As the research on interior point methods for linear programming has settled down (and the research on interior points for nonlinear programming has heated up), a number of new book projects on linear programming and/or interior point methods have

recently appeared on the scene which promise to surmount these pedagogical difficulties. For example, in the last two years alone, the following new book projects on linear programming have been undertaken which contain substantive and rigorous treatments of interior point methods: *Linear Programming: A Modern Integrated Analysis* by Romesh Saigal (Kluwer, 1995), *Introduction to Linear Optimization* by Dimitris Bertsimas and John Tsitsiklis (Athena Scientific, forthcoming), *Linear Programming: Foundations and Extensions* by Robert Vanderbei (Kluwer, forthcoming), *Interior Point Algorithms: Theory and Analysis* by Yinyu Ye (John Wiley, forthcoming), *Primal-Dual Interior Point Algorithms* by Stephen Wright (SIAM, forthcoming), and *Linear Programming I: An Introduction* by George Dantzig and Mukund Thapa (Springer Verlag, forthcoming).

To begin our synopsis of interior point methods for linear programming, we consider the linear programming problem in standard form:

$$\begin{aligned} P: \quad \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

where x is a vector of n variables, whose standard linear programming dual problem is:

$$\begin{aligned} D: \quad \text{maximize} \quad & b^T y \\ \text{s.t.} \quad & A^T y + s = c \\ & s \geq 0. \end{aligned}$$

Given a feasible solution x of P and a feasible solution (y, s) of D, the duality gap is simply $c^T x - b^T y = x^T s \geq 0$.

We introduce the following notation which will be very convenient for manipulating equations, etc. A feasible solution x of P is *strictly feasible* if $x > 0$, and a feasible solution (y, s) of D is *strictly feasible* if $s > 0$. Let e denote the vector of ones, i.e., $e = (1, \ldots, 1)^T$. Suppose that $x > 0$. Define the matrix X to be the $n \times n$ diagonal matrix whose diagonal entries are precisely the components of x. Then X looks like:

$$\begin{pmatrix} x_1 & 0 & \dots & 0 \\ 0 & x_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & x_n \end{pmatrix}.$$

Notice that $Xe = x$, and $X^{-1}e = (1/x_1, \dots, 1/x_n)^T$. Also, notice that both X and X^{-1} are positive-definite symmetric matrices.

There are many different types of interior point algorithms for linear programming, with certain common mathematical themes having to do with the logarithmic barrier function. In the authors' opinions, most interior point algorithms fall into one of three main categories: affine scaling methods, potential reduction methods, and central trajectory methods. We now briefly summarize these three categories.

Affine Scaling Methods. The basic strategy of the affine scaling algorithm is as follows: given a strictly feasible solution $\bar{x}$ of P, construct a simple local ellipsoidal approximation of the feasible region of P that is centered at $\bar{x}$. Call this ellipsoid $E_{\bar{x}}$. Then, optimize the objective function $c^T x$ over $E_{\bar{x}}$, and use the resulting direction with a suitable steplength to define a new algorithmic iterate. The specifics of this strategy are as follows. Given a strictly feasible solution $\bar{x}$ of P, the *Dikin Ellipsoid* at $\bar{x}$ is defined as:

$$E_{\bar{x}} = \{x \in \Re^n | Ax = b, (x - \bar{x})^T \bar{X}^{-2} (x - \bar{x}) \leq 1\}.$$

(It is straightforward to show that $E_{\bar{x}}$ is always contained in the feasible region of P whenever $\bar{x}$ is strictly feasible.) The affine scaling direction at $\bar{x}$ is then the solution to the following direction-finding problem:

$$\begin{aligned} (ADFP_{\bar{x}}): \quad \text{minimize} \quad & c^T d \\ \text{s.t.} \quad & Ad = 0 \\ & d^T \bar{X}^{-2} d \leq 1. \end{aligned}$$

Note that $(ADFP_{\bar{x}})$ is a convex program with all linear components except for one convex quadratic constraint. It can be solved directly by forming the associated Karush-Kuhn-Tucker system and then solving an associated linear equation system. One can also write down a closed form solution

algebraically after a little bit of matrix manipulation. Letting $\bar{d}$ denote the solution to problem $(ADFP_{\bar{x}})$, the next iterate of the affine scaling algorithm is obtained by setting $x_{new} = \bar{x} + \alpha\bar{d}$, where the steplength α is chosen by one of several strategies to ensure the strict feasibility of the new iterate x_{new} while maintaining a suitable improvement in the objective function.

The affine scaling algorithm is attractive due to its simplicity and its good performance in practice. (However, its performance is quite sensitive to the starting point of the algorithm.) The proof of convergence of the algorithm in the absence of degeneracy is fairly straightforward, but under degeneracy, such a proof is surprisingly long and difficult. There have not been any results on bounds on the efficiency of the algorithm, but it is suspected (for very good reasons that are beyond the scope of this synopsis) that the algorithm is exponential time in the worst case.

Some variants/extensions on the basic affine scaling algorithm are the dual affine scaling algorithm (designed to work on the dual problem D), as well as various versions that work simultaneously on the primal and on the dual, using a variety of ellipsoidal constructions in the space of the primal and the dual. A comprehensive reference on the affine scaling is the book by Saigal [17].

Potential Reduction Methods. Potential reduction methods typically are designed to find improving solutions to the following optimization problem:

$$
\begin{aligned}
PRP: \quad \text{minimize} \quad & f(x,y,s) = q\ln(c^Tx - b^Ty) - \textstyle\sum_{j=1}^{n}\ln(x_j) \\
\text{s.t.} \quad & Ax = b \\
& x > 0, \\
& A^Ty + s = c \\
& s \geq 0,
\end{aligned}
$$

where the objective function $f(x,y,s)$ is called the *potential function*, and q is a parameter of the potential function. It was this type of problem that Karmarkar introduced in his seminal paper [10]. Notice that the "first part" of the potential function is q times the logarithm of the duality gap, and we would like to drive this part to $-\infty$. The second part of the potential function is the logarithmic barrier function, which is designed to repel feasible solutions from the boundary of the feasible region. The potential function

is a surrogate for the goal of reducing the duality gap to zero, and under some mild assumptions regarding the linear program P, one can easily show that the duality gap is bounded from above by a function of the value of the potential function, i.e.,

$$c^T x - b^T y \leq C_1 e^{f(x,y,s)/q}$$

for a constant C_1 that is problem specific.

Now, suppose that the parameter q has been set. In a typical potential reduction method, we have a current iterate $(\bar{x}, \bar{y}, \bar{s})$ and we seek new iterate values $(x_{new}, y_{new}, s_{new})$ with a suitable decrease in the potential function. There are a number of tools that can be used to accomplish this, such as Newton's method, a "partial" Newton's method that only accounts for the Hessian of the second part of of the potential function, and projective transformation methods combined with projected steepest descent. In a typical potential reduction algorithm, there is a guaranteed decrease in the potential function $f(x, y, s)$ of at least an amount δ at each iteration, where $\delta > 0$. Then, from the above, the duality gap is therefore decreased by a fixed proportion in at most q/δ iterations. This reasoning is then used to establish an upper bound on the total number of iterations needed to obtain a near-optimal solution within some optimality tolerance ϵ from some starting point (x^0, y^0, s^0), namely

$$\frac{q}{\delta}\left(\ln\left(\frac{c^T x^0 - b^T y^0}{\epsilon}\right) + C_2\right)$$

iterations, for a constant C_2 that depends on the problem P and on the starting point (x^0, y^0, s^0). This type of logic underlies most potential reduction algorithms.

Although potential reduction methods do not have the simplicity of affine scaling methods, they are more attractive than affine scaling algorithms for at least two reasons: they have a performance guarantee, and they always produce dual information and so allow the user to specify an optimality tolerance to the computer. Also, with a line-search of the potential function, they can be made very efficient in practice.

(Karmarkar's original algorithm [10] used a very specific form of PRP and used the machinery of projective transformations in the algorithm and in the proofs of the algorithm's performance guarantees. Despite their

original mystique, projective transformations are not necessary for potential reduction algorithms to work either in theory or in practice. However, in the authors' opinions, the framework of projective transformations is nevertheless of paramount importance, at least conceptually, in the understanding of interior point methods in general.)

There are numerous types of potential reduction methods, some using the potential function above, others using the so-called Tanabe-Todd-Ye symmetric primal-and-dual potential function

$$g(x, y, s) = q \ln(c^T x - b^T y) - \sum_{j=1}^{n} \ln(x_j) - \sum_{j=1}^{n} \ln(s_j),$$

which has additional desirable properties that go beyond this brief synopsis. In general the potential reduction methods all aim to drive a potential function to $-\infty$ by a variety of primal, dual, or primal-and-dual algorithmic tools. Almost all potential reduction methods enjoy good to excellent performance guarantees, i.e., complexity bounds. Potential reduction methods have not received much attention in terms of computational testing, due perhaps to early difficulties (which have since been overcome) of applying potential reduction methods in a combined Phase I-Phase II environment. For a comprehensive survey of potential reduction methods, see Anstreicher [4] or Todd [18].

Central Trajectory Methods. Interior point methods based on the *central trajectory* are by far the most useful in theory, the most used in practice, and, in our judgement, have the most aesthetic qualities. (In fact, one leading researcher has even referred to the central trajectory as "the most important object in modern optimization theory.") The *central trajectory* of the linear program P is obtained as the solution to an amended version of P, where a parameterized logarithmic barrier term is added to the objective function. Consider the logarithmic barrier problem $BP(\mu)$ parameterized by the positive barrier parameter μ:

$$\begin{aligned} BP(\mu): \quad \text{minimize} \quad & c^T x - \mu \textstyle\sum_{j=1}^{n} \ln(x_j) \\ \text{s.t.} \quad & Ax = b \\ & x > 0. \end{aligned}$$

The Karush-Kuhn-Tucker conditions for $BP(\mu)$ are:

$$\begin{cases} Ax = b, x > 0 \\ \\ c - \mu X^{-1}e = A^T y. \end{cases} \tag{1}$$

If we define $s = \mu X^{-1}e$, then we can rewrite these optimality conditions as:

$$\begin{cases} Ax = b, x > 0 \\ \\ A^T y + s = c, s > 0 \\ \\ XSe - \mu e = 0. \end{cases} \tag{2}$$

Let $(x(\mu), y(\mu), s(\mu))$ denote the solution to system (2) for the given positive parameter μ. Then the set $\Gamma = \{(x(\mu), y(\mu), s(\mu)) | \mu > 0\}$ is defined to be the *central trajectory* (also known as the central path) of the linear program P. From the first two equation systems of (2), we see that a solution (x, y, s) along the central trajectory is strictly feasible for the primal and the dual problem, and that the duality gap on the central trajectory is $x^T s = e^T XSe = \mu e^T e = \mu n$, which follows from the third equation system of (2). Substituting this equation in the third system of (2), we obtain the following equivalent and parameter-free characterization of the central trajectory:

$$\begin{cases} Ax = b, x > 0 \\ \\ A^T y + s = c, s > 0 \\ \\ XSe - (x^T s/n)e = 0. \end{cases} \tag{3}$$

The third equation system in (2) or (3) is precisely where the nonlinearity arises, and in general it is not possible to solve (2) or (3) in closed form except in trivial cases.

The strategy in most central trajectory methods is to solve for approximate solutions along the central trajectory (2) or (3) for a decreasing sequence of the duality gap (or equivalently, of the barrier parameter μ) that tends to zero in the limit. There are a number of ways to carry out this strategy. For example, for a given value of the duality gap or of the barrier parameter μ, one can choose to approximately optimize $BP(\mu)$ or, equivalently, to approximately solve (1), (2), or (3), or to approximately solve

some other equivalent characterization of the central trajectory. Also, one can choose a number of ways to approximately solve the system of nonlinear equations under consideration (Newton's method is one obvious choice, as are predictor-corrector methods and other higher-order methods, preconditioned conjugate gradient methods, etc.). Overlayed with all of this is the way in which the numerical linear algebra is implemented. Furthermore, one needs to decide how to measure "approximate" in the approximate solution. Last of all, there is considerable leeway in developing a strategy for reducing the duality gap (or the barrier parameter μ) at each iteration. (For example, aggressively shrinking the duality gap seems like a good idea, but will also increase the number of iterations of Newton's method (or other method) that is used to re-solve (approximately) the new system of nonlinear equations.)

In terms of theoretical performance guarantees, the best central trajectory methods are guaranteed to reduce the duality gap of the iterates by a fixed proportion in $O(\sqrt{n})$ iterations.

A short summary of central trajectory methods is given in Jansen et al. [9]. More comprehensive treatments of central trajectory methods are given in den Hertog [8], and the forthcoming book by Wright [21].

The rest of this article is devoted to a discussion of important current research topics in interior point methods. We discuss the following topics, in order: infeasible interior point methods, computational aspects of interior point methods, homogeneous and self-dual methods, semidefinite programming, convex programming and self-concordance, linear and nonlinear complementarity problems, and theoretical issues related to interior-point methods.

2 Infeasible Interior Point Methods

By definition, interior point methods naturally work on the interior (or relative interior) of a problem's feasible region, and consequently one obvious issue is how an initial feasible interior point can be obtained. Over the years, a number of techniques for handling the "feasibility" or "Phase I" problem have been proposed, including combined Phase I-Phase II methods, shifted-barrier methods, and homogeneous self-dual methods. In practice, methods based on a variation of a relatively simple algorithm, the "primal-

dual infeasible-interior-point method" proved to be very successful. The basic method attempts to reduce the feasibility and optimality gaps at each iteration by applying Newton's method to the system (2) or (3) from an initial point (x^0, y^0, s^0) which is not necessarily feasible for either P or D, i.e., $Ax^0 \neq b$ and/or $A^T y^0 + s^0 \neq c$, but which is "interior" in that $x^0 > 0$ and $s^0 > 0$. In this sense, the algorithm is a simple variant of the standard central trajectory path-following algorithm, but where the iterates are not necessarily feasible at each iteration. Let $(\bar{x}, \bar{y}, \bar{s})$ be an iterate. Then the Newton direction (d_x, d_y, d_s) for the algorithm is derived from the nonlinear equation system (2) and is the solution of the system

$$\begin{cases} Ad_x = -(A\bar{x} - b) \\ A^T d_y + d_s = -(A^T \bar{y} + \bar{s} - c) \\ \bar{S}d_x + \bar{X}d_s = -(\bar{X}\bar{S}e - \mu e). \end{cases} \tag{4}$$

Of course, if the iterate $(\bar{x}, \bar{y}, \bar{s})$ is feasible, which is the case in a standard central trajectory interior point algorithm, then the right-hand-side of the first and second equations of (4) are 0, and consequently the directions d_x and d_s are orthogonal. As it turns out, the orthogonality of d_x and d_s is essential for an "easy" analysis of Newton's method for solving (2), and is lost when the iterate $(\bar{x}, \bar{y}, \bar{s})$ is infeasible.

Although quite successful computationally, the primal-dual infeasible-interior-point method long defied any reasonable theoretical analysis, nor even a proof of convergence, until a few years ago, when satisfactory analyses by several researchers emerged. One of the difficulties in the analysis of the algorithm was the lack of foreknowledge of the existence of the feasible solutions (interior or not), the existence of the central trajectory, or the existence of optimal LP solutions. In the case when the either P or D is infeasible, it is theoretically possible to detect the infeasibility of the problems P and/or D, but such detection mechanisms do not necessarily work well in practice.

To overcome these difficulties, another type of infeasible-interior-point algorithm has been developed. The basic algorithm uses the following

variant of the system (3):

$$\begin{cases} Ax = b + \theta(Ax^0 - b), x > 0 \\ A^T y + s = c + \theta(A^T y^0 + s^0 - c), s > 0 \\ XSe - \theta\left((x^0)^T s^0/n\right)e = 0. \end{cases} \tag{5}$$

where (x^0, y^0, s^0) is the initiating point of the algorithm (where $x^0 > 0$ and $s^0 > 0$, but quite possibly $Ax^0 \neq b$ and $A^T y^0 + s^0 \neq c$), and $\theta \in (0, 1]$. Here, the goal is to use Newton's method to solve (5) for a decreasing sequence of values of θ tending to 0. If $\|X^0 S^0 e - \left((x^0)^T s^0/n\right)e\|$ is small enough, the point (x^0, y^0, s^0) is a good approximate solution of the system (5) for $\theta = 1$. The set of solutions to the system (5) forms a path parameterized by θ, which does not lie in the feasible region of the problem unless the initial point is feasible. Nevertheless, if P and D are feasible, the path leads to optimal primal and dual solutions as θ goes to 0. If either P and/or D is infeasible, there exists a positive lower bound on θ for which the system (5) has a solution, and the path diverges to infinity as θ approaches this lower bound. By exploiting these and other features of the path, one can develop an infeasible interior point path-following algorithm which either solves P and D or detects infeasibility in a polynomial number of iterations.

The former type of algorithms based on the Newton system (4) are preferred in practice, probably because they more effective than the latter method (based on (5)) when the linear program is feasible. The authors believe, however, that the latter type of algorithm is most likely to outperform the former when the underlying linear program is either infeasible or is close to being infeasible. Affine scaling methods and potential reduction methods starting from an infeasible starting point have also been developed, but practical versions of these algorithms have not received very much attention. For a comprehensive summary of infeasible interior point algorithms, see Mizuno [14].

3 Computational Aspects of Interior Point Methods for Linear Programming

Much of the initial excitement about interior point methods stemmed from the rather remarkable computational promise of the method, as articulated by Karmarkar and others. Twelve years later, computational aspects of interior point methods are still of paramount importance. Although neither author is particularly qualified to comment on computational issues, it is only fair to briefly discuss key aspects of computation nevertheless.

After much effort in designing and testing various interior point methods for linear programming in the 1980's and early 1990's, the computational picture of interior point methods for linear programming has somewhat settled down. While a suitably customized simplex method implementation enjoys superior performance over interior point methods for most routine applications of linear programming, interior point algorithms are superior to the simplex method for certain important classes of problems. The simplex method tends to perform poorly on large massively-degenerate problems, whereas interior point methods are immune to degeneracy (and are aided by it in certain ways), and so one can expect an interior point algorithm to outperform the simplex method to the extent that the underlying linear program has massive degeneracy. This is the case in large scheduling problems, for instance, which give rise to LP relaxations of binary integer programs. Also, because the linear-algebra engine of an interior point iteration works with a Cholesky factorization of the matrix $A\bar{X}^2A^T$, interior point algorithms will outperform the simplex method to the extent that the matrix A is conducive to producing relatively sparse Cholesky factors of the matrix $A\bar{X}^2A^T$. Such is the case in large staircase multi-period linear programs, for example. Other than these two general problem types, there are not many other ways to predict which method will be more efficient. The "state-of-the-art" of interior point computation as of the early 1990's was described in the article of Lustig, Marsten, and Shanno [13]; more recent advances (using higher-order methods that are up to 25% faster) are described in Andersen et al. [3]. For a comprehensive treatment of computational issues in interior point methods, we recommend the forthcoming book by S. Wright [21].

There are a number of software codes for interior point methods

for linear programming, including PCx (by Czyzyk, Mehrotra, and Wright), HOPDM (by Gondzio et al.), BPMPD (by Mészáros), OSL (IBM), CPLEX/Barrier (CPLEX Optimization Inc.), XPRESS-MP (Dash Associates), LOQO (by R. Vanderbei), and LIPSOL (by Y. Zhang). Information on these and other interior point method codes is updated regularly on the World Wide Web page `http://www.mcs.anl.gov/home/wright/IPPD/`. Some of these codes are free to the research community, others are solely commercial, and others are a hybrid.

Computational issues for nonlinear optimization is deferred to the sections on semidefinite programming (Section 4) and on convex optimization (Section 5).

4 Semidefinite Programming

In the opinion of the authors, semidefinite programming (SDP) is the most exciting development in mathematical programming in the 1990's. SDP has applications in traditional convex constrained optimization, as well as in such diverse domains as control theory and combinatorial optimization. Because SDP is solvable via interior point methods, there is the promise that these applications can be solved efficiently in practice as well as in theory. Before defining a semidefinite program, we need to amend our notation. Let S^n denote the set of symmetric $n \times n$ matrices, and let S^n_+ denote the set of positive semidefinite (psd) $n \times n$ matrices. Then S^n_+ is a closed convex cone in $\Re^{n^2}$ of dimension $n \times (n-1)/2$. We write "$X \succeq 0$" to denote that X is symmetric and positive semidefinite, and we write "$X \succeq Y$" to denote that $X - Y \succeq 0$ ("$\succeq$" is the Löwner partial ordering on S^n). Here, X is any symmetric matrix, not necessarily a diagonal matrix as denoted earlier. We write "$X \succ 0$" to denote that X is symmetric and positive definite, etc. Let $X \in S^n$. A linear function $M(X)$ can be written as $M \bullet X$, where $M \bullet X = \sum_{i=1}^n \sum_{j=1}^n M_{ij} X_{ij}$. Then a semidefinite program (SDP) is an optimization problem of the form:

$$
\begin{aligned}
SDP: \quad \text{minimize} \quad & C \bullet X \\
\text{s.t.} \quad & A_i \bullet X = b_i, i = 1, \ldots, m \\
& X \succeq 0,
\end{aligned}
$$

where X is an $n \times n$ matrix, and the data for the problem are the m symmetric matrices $A_1, \ldots, A_m$, the symmetric matrix C, and the $m-$vector b. Notice that SDP has a linear objective function and linear equality constraints, just like a linear program. However, the standard LP constraint that x is nonnegative is replaced by the constraint that the variable X is symmetric and positive semidefinite. (It is helpful to think of $X \succeq 0$ as stating that the *vector of eigenvalues* of X is nonnegative.) The Lagrange dual of SDP is derived as:

$$\begin{aligned} SDD: \quad \text{maximize} \quad & b^T y \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^m y_i A_i + S = C \\ & S \succeq 0. \end{aligned}$$

Given a feasible solution X of SDP and a feasible solution (y, S) of SDD, the duality gap is simply $C \bullet X - b^T y = X \bullet S \geq 0$.

As stated above, SDP has very wide applications in convex optimization. The types of constraints that can be modeled in the SDP framework include: linear inequalities, convex quadratic inequalities, lower bounds on matrix norms, lower bounds on determinants of symmetric positive semidefinite matrices, lower bounds on the geometric mean of a nonnegative vector, plus many others. Using these and other constructions, the following problems (among many others) can be cast in the form of a semidefinite program: linear programming, optimizing a convex quadratic form subject to convex quadratic inequality constraints, minimizing the volume of an ellipsoid that covers a given set of points and ellipsoids, maximizing the volume of an ellipsoid that is contained in a given polytope, plus a variety of maximum eigenvalue and minimum eigenvalue problems.

SDP also has wide applicability in combinatorial optimization as well. A number of $NP-$hard combinatorial optimization problems have convex relaxations that are semidefinite programs. In many instances, the SDP relaxation is very tight in practice, and in certain instances in particular, the SDP relaxation is provably quite tight and can be converted to provably very good feasible solutions with provably good bounds on optimal objective values. Last of all, SDP has wide applications in control theory, where a variety of control and system problems can be cast and solved as instances of SDP.

As it turns out, virtually *all* of the mathematics, constructions, and

even the notation of interior point methods for linear programming extends directly to SDP. This is truly remarkable. (The extension of interior point methods to SDP was developed independently by Alizadeh [1] and Nesterov and Nemirovskii [15] using different frameworks.) For example, the analogous parameterized logarithmic barrier problem BP_μ for linear programming extends to SDP as:

$$\begin{aligned} BSDP(\mu): \quad \text{minimize} \quad & C \bullet X - \mu \ln(det(X)) \\ \text{s.t.} \quad & A_i \bullet X = b_i, i = 1, \ldots, m \\ & X \succ 0, \end{aligned}$$

where notice that $\ln(det(X))$ replaces the logarithmic barrier function $\sum_{j=1}^{n} \ln(x_j)$. The optimality conditions for this problem can be written as:

$$\begin{cases} A_i \bullet X = b_i, i = 1, \ldots, m, X \succ 0 \\ \sum_{i=1}^{n} y_i A_i + S = C, S \succ 0 \\ XS - \mu I = 0, \end{cases} \tag{6}$$

which should be compared with (2). The third equation system of (6) can alternatively be represented in many different equivalent ways, including for example, $(XS + SX)/2 - \mu I = 0$, resulting in many different non-equivalent Newton directions for solving (6).

In terms of theoretical performance guarantees, the best central trajectory methods for SDP are guaranteed to reduce the duality gap of the iterates by a fixed proportion in $O(\sqrt{n})$ iterations. This is identical to the theoretical performance guarantee for linear programming, even though the dimension of the variables in SDP is much larger ($n \times (n-1)/2$ as opposed to n for linear programming).

There are many very active research areas in semidefinite programming. In the arena of theory, there is research on the geometry and the boundary structure of SDP feasible regions (including notions of degeneracy). There is research related to the computational complexity of SDP, such as decidability questions, certificates of infeasibility, and duality theory. There is active research on SDP relaxations of combinatorial optimization in theory and in practice. As regards interior point methods, there are a

host of research issues, mostly involving the development of different interior point algorithms and their properties, including rates of convergence, performance guarantees, etc.

Because SDP has so many applications, and because interior point methods show so much promise, perhaps the most exciting area of research on SDP has to do with computation and implementation of interior point algorithms. Researchers are quite optimistic that interior point methods for SDP will become practical, efficient, and competitive. However, in the research to date, computational issues have arisen that are much more complex than those for linear programming, see for example Alizadeh, Haeberly, and Overton [2]. These computational issues are only beginning to be well-understood. They probably stem from a variety of factors, including the fact that the nonlinear conditions in the third equation system of (6) can be represented in very many different ways, resulting in many different Newton directions, see above. Other contributing issues might be the fact that SDP is not guaranteed to have strictly complementary optimal solutions (as is the case in linear programming), and the fact that the Jacobian of the KKT system defining the Newton step can be much more poorly conditioned than is the typical case in linear programming. Furthermore, there are challenges in developing the linear algebra (both symbolic and numeric) for solving SDP by interior point algorithms. Finally, because SDP is such a new field, there is no representative suite of practical problems on which to test algorithms, i.e., there is no equivalent version of the **netlib** suite of industrial linear programming problems.

A comprehensive survey of semidefinite programming is the article by Vandenberghe and Boyd [19].

5 Convex Programming and Self-Concordance

Almost immediately after Karmarmkar's work appeared, researchers began to explore extensions of interior point methods to general convex optimization problems. Indeed, the nonlinear nature of interior point methods naturally suggested that such extensions were possible. Throughout the 1980's, a number of papers were written that showed that central trajectory methods and potential reduction methods for LP could be generalized to certain types of convex programs with theoretical performance guarantees, under a

variety of restrictions (such as smoothness conditions) on the convex functions involved. However, there was no unifying theory or analysis. Then, in an incredible *tour de force*, Nesterov and Nemirovskii [15] presented a deep and unified theory of interior point methods for all of convex programming based on the notion of *self-concordant functions*. The intellectual contributions of this one research monograph cannot be overstated, whether it be for its mathematical depth, its implications for the theory of convex optimization and computational complexity, or for its implications for computation. To outline the thrust of this work, consider the following general convex program:

$$CP: \quad \text{minimize} \quad f(x)$$

$$\text{s.t.} \qquad g_i(x) \leq 0, i = 1, \ldots, m,$$

where $g_i(x)$ is convex, $i = 1, \ldots, m$, and the objective function $f(x)$ is linear (if not, add a new variable t and a new constraint $f(x) \leq t$ and declare the new objective function to be "minimize t"). Let $D = \{x | g_i(x) < 0, i = 1, \ldots, m\}$, and suppose we have a (convex) barrier function $B(x)$ that goes to infinity as x goes to the boundary of D. Then the barrier problem associated with CP is:

$$BCP_\mu: \quad \text{minimize} \quad f(x) + \mu B(x)$$

$$\text{s.t.} \qquad x \in D,$$

and the central trajectory of BCP_μ is the set of optimal solutions $x(\mu)$ to BCP_μ parameterized by the barrier parameter μ.

Nesterov and Nemirovskii show that Newton's method is a very efficient tool for solving CP by tracing the central trajectory of BCP_μ, when the barrier function $B(x)$ has the property of *self-concordance*. Their approach is very general: $B(x)$ does not necessarily depend on the way the functions $g_i(x)$ are expressed. It just depends on the interior of the underlying feasible region. One of the central ideas in the understanding of self-concordance (and the interior point algorithms derived from them) is the use of the Hessian of $B(x)$ to induce a local norm at x. Let $H(x)$ be the Hessian of $B(x)$ at x. The *induced norm* at x is defined to be $n_x(v) = \sqrt{v^T H(x) v}$, which is a quadratic norm using the Hessian $H(x)$ as the quadratic form. Roughly speaking, a function $B(x)$ is a *ϑ−self-concordant barrier function* with barrier parameter ϑ if $B(x)$ satisfies the following conditions: (i) local changes in the Hessian of $B(\cdot)$ at two points x and y can be bounded by the induced

norm at x of $(x-y)$, and (ii) the induced norm of the Newton step at x is no larger than $\sqrt{\vartheta}$. (Stated more colloquially, a function is a $\vartheta-$self-concordant barrier if the Hessian of the function is a relevant tool in bounding the size of the Newton step (by $\sqrt{\vartheta}$) and in measuring changes in the Hessian itself.) Nesterov and Nemirovskii show that when a convex program has a $\vartheta-$self-concordant barrier, then Newton's method improves the accuracy of a given solution of (CP) by at least t digits in $O(\sqrt{\vartheta}t)$ Newton steps.

At present, $\vartheta-$self-concordant barriers are known for only a few, but very important, classes of convex programs. These include linear and convex quadratically constrained programs (where $B(x) = -\sum_{i=1}^{m} \ln(b_i - a_i^T x - x^T Q_i x)$ and $\vartheta = m$) and semidefinite programming (where $B(x) = -\ln(det(X))$ for the $n \times n$ matrix X and $\vartheta = n$), as well as convex programs involving the second-order cone $\{x | x^T Q x \leq (c^T x + d)^2, c^T x + d \geq 0\}$, and even epigraphs of matrix norms. However, at least in theory, self-concordant barriers can be used to process *any* convex program efficiently: indeed, Nesterov and Nemirovskii show that every open convex set in $\Re^n$ posesses a $\vartheta-$self-concordant barrier where $\vartheta \leq Cn$ for some universal constant C. The implications of this truly far-reaching result for the complexity of convex programming is now being explored.

Nesterov and Nemirovskii also provide a "barrier calculus" consisting of many simple tools which allow the derivation of self-concordant barriers for complicated convex sets, based on self-concordant barriers for simpler convex sets.

In addition, Nesterov and Nemirovskii also work on the following conic form of convex optimization:

$$\begin{aligned} KP: \quad \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \in K, \end{aligned}$$

where K is a pointed, closed, convex cone with non-empty interior which posesses a $\vartheta-$self-concordant barrier; their algorithms and performance guarantees apply easily to this case. This elegant form allows for better presentation, and also makes it easier to draw parallels (when applicable) among interesting and well studied special cases of CP and KP, such as linear programming (where K is the nonnegative orthant) and semidefinite programming (where K is the cone of symmetric positive semidefinite matrices).

Finally, researchers such as Güler [7] are demonstrating deep connections between the theory of interior point methods using $\vartheta-$self-concordant barriers, and other branches of mathematics including algebra, complex analysis, and partial differential equations.

At present, computational experience with interior point methods for convex programming is rather limited, except as noted in the case of semidefinite programming. However, researchers are optimistic that at least some of the success of interior point methods for linear and semidefinite programming will be realized for more general convex programs.

6 Homogeneous and Self-Dual Methods

A linear programming problem is called *self-dual* if its dual problem is equivalent to the primal problem. Given a linear program P and an initial (possibly infeasible) point (x^0, y^0, s^0) for which $x^0 > 0$ and $s^0 > 0$, a homogeneous and self-dual interior point method constructs the following artificial linear program $HSDP$ which is self-dual and almost homogeneous:

$$
\begin{array}{lrrrrl}
HSDP: \ \text{minimize} & & & & ((x^0)^T s^0 + 1)\theta & \\
\text{s.t.} & Ax & -b\tau & & +\bar{b}\theta & = 0, \\
& -A^T y & +c\tau & & -\bar{c}\theta & \geq 0, \\
& b^T y & -c^T x & & +\bar{z}\theta & \geq 0, \\
& -\bar{b}^T y & +\bar{c}^T x & -\bar{z}\tau & & = -(x^0)^T s^0 - 1, \\
& x \geq 0, & \tau \geq 0, & & &
\end{array}
$$

where

$$\bar{b} = b - Ax^0, \quad \bar{c} = c - A^T y^0 - s^0, \quad \bar{z} = c^T x^0 + 1 - b^T y^0.$$

It is not hard to see that this program is self-dual, because the coefficient matrix is skew-symmetric. Denote the slacks on the second and third set of constraints by s and κ. Then $HDSP$ has a trivial feasible interior point $(x, \tau, \theta, y, s, \kappa) = (x^0, 1, 1, y^0, s^0, 1)$ that can be used to initiate any interior point algorithm for solving $HDSP$. Since the dual is equivalent to the primal, the optimal value of $HDSP$ is zero. By using a path following interior point algorithm, one can compute a strictly self-complementary solution $(x^*, \tau^*, \theta^*, y^*, s^*, \kappa^*)$ such that $\theta^* = 0$, $x^* + s^* > 0$, and $\tau^* + \kappa^* > 0$. If

$\tau^* > 0$, then x^*/τ^* is an optimal solution of P and $(y^*/\tau^*, s^*/\tau^*)$ is an optimal solution of D. Otherwise, by strict complementarity, $\kappa^* > 0$, whereby from the third constraint it follows that either $c^T x^* < 0$ or $-b^T y^* < 0$. The former case implies the infeasibility of the primal problem P, and the latter case implies the infeasibility of the dual problem D.

The homogeneous and self-dual interior point method possesses the following nice features: (i) it solves a linear program P without any assumption concerning the existence of feasible, interior feasible, or optimal solutions, (ii) it can start at any initial point, feasible or not, (iii) each iteration solves a system of linear equations whose size is almost the same as for standard interior-point algorithms, (iv) if P is feasible, it generates a sequence of iterates that approach feasibility and optimality simultaneously, otherwise it correctly detects infeasibility for at least one of the primal and dual problems, and (v) it solves a problem in polynomial time ($O(\sqrt{n}L)$ iterations) without using any "big M" constants. We point out that the infeasible interior point algorithms presented in Section 2 do not posess this last feature. Homogeneous and self-dual interior point methods have the promise to be competitive in practice with standard interior point software, see Xu et al. [22].

Homogeneous and self-dual methods can be extended to more general problems, such as linear and nonlinear complementarity problems. We refer readers to Ye et al. [23] for an initial description.

7 Linear and Nonlinear Complementarity Problems

The standard linear complementarity problem, or LCP, is to find a pair (x, s) of n-dimensional variables that satisfy the linear constraint

$$s = Mx + q$$

and the complementarity conditions

$$(x, s) \geq 0, \quad x_j s_j = 0, \quad j = 1, \ldots, n,$$

where M is an $n \times n$ matrix and $q \in R^n$. The optimality conditions for both linear programming and convex quadratic programming can be cast as an

instance of LCP, and for this reason LCP is often used as a general model in the development and analysis of interior-point algorithms.

While there are several important classes of the LCP, the most important class is the *monotone* LCP, defined to be those instances for which the set of feasible solutions (x, s) are maximal and monotone in R^{2n} (equivalently, for which the matrix M is positive semidefinite). Linear and convex quadratic programming problems fall into this class. More generally, instances of LCP are typically classified according to classes of the matrix M, such as P_0-matrices and $P_*(\kappa)$-matrices (see Kojima et al. [11] for definitions).

Interior point methods for solving LCP have been developed using the following generalization of the central trajectory equation system (2):

$$\begin{cases} s = Mx + q, x > 0, s > 0, \\ \\ XSe - \mu e = 0. \end{cases} \tag{7}$$

If the matrix M is a P_0-matrix and a feasible interior point exists, then the set of solutions to (7) forms a path (central trajectory) parameterized by μ, leading to a solution of LCP as μ goes to 0, and so one can solve the standard LCP with a P_0 matrix by using a path-following interior point algorithm. This approach extends to infeasible interior point methods, and potential reduction methods for solving LCP have also been proposed by researchers. In the case of the monotone LCP, many interior point algorithms have a polynomial time performance guarantee. For $P_*(\kappa)$-matrix LCP, there is also an explicit complexity analysis and performance guarantee. The solution of LCP with P_0-matrices is known to be NP-complete.

The nonlinear complementarity problem, or $NLCP$, is the problem for finding (x, s) such that

$$s = f(x), \quad (x, s) \geq 0, \quad XSe = 0,$$

for a given continuous function $f(\cdot)$. If $f(\cdot)$ is monotone, $NLCP$ is also called monotone. The optimality conditions for convex constrained optimization can be cast as an instance of the monotone $NLCP$. For this class of $NLCP$, the central trajectory system (7) can be suitably generalized, and so can be solved by path-following interior point methods. Interior point methods for more general classes of $NLCP$ are discussed in Kojima et al. [12].

Recently researchers have become interested in the semidefinite complementarity problem, or $SDCP$, which is a special class of $NLCP$ arising in the study of semidefinite programming (see Section 4). Infeasible interior point algorithms have been developed for the monotone instances of $SDCP$, and $SDCP$ is currently a very active research problem.

8 Some Theoretical Issues Related to Interior Point Methods

Recently, theoretical research on the complexity of solving linear programming has focused on developing appropriate measures for adequately representing the "size" of an LP instance, that are more relevant to computation than traditional measures of "size" such as the dimensions m and n or the bit-size L of a binary representation of an LP instance. In this closing section, we discuss two such measures, namely $C(d)$ of an LP data instance $d = (A, b, c)$, and $\bar{\chi}_A$ for the matrix A.

Consider the very general convex optimization problem cast as follows:

$$\begin{array}{lll} P(d): & \text{maximize} & c^T x \\ & \text{s.t.} & b - Ax \in C_Y \\ & & x \in C_X, \end{array}$$

where C_X and C_Y are closed convex cones, and the data d for the problem is the array $d = (A, b, c)$. Any convex optimization problem can be cast in this form, including LP as a special case. The terminology $P(d)$ emphasizes the dependence of the problem on the data $d = (A, b, c)$. Renegar [16] develops a *condition number* $C(d)$ for $P(d)$ that is intuitively appealing, arises naturally in considering the problem $P(d)$, is an extension of the traditional condition number for systems of linear equations, and posesses many attractive geometric and algebraic characteristics. (For example, if $P(d)$ has a feasible solution, then it must have a feasible solution whose norm is no larger than $C(d)$.) We give a rough description of $C(d)$ as follows. Let $d = (A, b, c)$ be the data for $P(d)$ and let $\Delta d = (\Delta A, \Delta b, \Delta c)$ be a change in the data. Let $\rho(d)$ be the smallest change Δd needed to make the problem $P(d)$ either infeasible or unbounded. Then $C(d)$ is defined to be $\|d\|/\rho(d)$. That is, $C(d)$ is a scale-invariant version of the reciprocal of the smallest

change in the data $d = (A, b, c)$ needed to cause $P(d+\Delta d)$ to be ill-behaved. Roughly speaking, Renegar shows that the complexity of an interior point method for solving $P(d)$ is inherently sensitive only to the condition number $C(d)$ of the underlying problem and to the barrier parameter ϑ of the self-concordant barrier for the cones C_X and C_Y, (see Section 5), and that the complexity bound on the number of iterations is $O(\sqrt{\vartheta}(\ln(C(d) + \ln(1/\epsilon))$ to produce an $\epsilon-$optimal solution of $P(d)$. Therefore, the interior point algorithm is efficient in a well-defined sense. Not surprisingly, the condition number $C(d)$ is intrinsically involved in a variety of special properties of the central trajectory of a linear program (see Freund and Nunez [5]), and we anticipate that the study of the condition number $C(d)$ will yield much new insight into linear and convex programming in the future.

Another very interesting development due to Vavasis and Ye [20] is an interior point algorithm for linear programming whose running time depends only on the dimension n and on a certain measure of the matrix A denoted by $\bar{\chi}_A$. Let (A, b, c) be the data for an LP instance, where the data are *not* restricted to be rational numbers. For the matrix A, define the quantity:

$$\bar{\chi}_A = \sup\{\|A^T(ADA^T)^{-1}AD\| \mid D \text{ is a positive } n \times n \text{ diagonal matrix } \}.$$

Then Vavasis and Ye present an interior point algorithm for solving a linear program in at most $O\left(n^{3.5}\left(\ln(\bar{\chi}_A) + \ln(n) + C\right)\right)$ iterations of Newton's method, where C is a universal constant. The significance of this result derives from the fact that the data b and c play no role in the bound on the number of iterations. Put a different way, the efficiency of their algorithm for linear programming depends only on the dimension and on a certain algebraic property of the matrix A embodied in the quantity $\bar{\chi}_A$. This research improves on earlier work by Tardos, by showing that the dependency of the complexity result on A is true even when the data are not presumed to be integer (or rational) coefficients.

9 Acknowledgement

The authors are grateful to Levent Tunçel and Stephen Wright for their input regarding certain aspects of this article. (Of course, the authors alone take responsibility for all opinions stated herein, as well as for all possible errors.)

References

[1] F. Alizadeh, Interior point methods in semidefinite programming with applications to combinatorial optimization, *SIAM Journal on Optimization* 5 13-51, 1995.

[2] F. Alizadeh, J.P. Haeberly, and M. Overton, Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results, New York University Computer Science Dept Report 721, May 1996.

[3] E. Andersen, J. Gondziio, C. Meszaros, and X. Xu, Implementation of interior point methods for large scale linear programming, in *Interior point methods in mathematical programming*, T. Terlaky, ed., Kluwer Academic Publisher, 1996.

[4] K. Anstreicher, Potential Reduction Methods, in *Interior point methods in mathematical programming*, T. Terlaky, ed., Kluwer Academic Publishers, 1996.

[5] R.M. Freund and M. Nunez, Condition measures and properties of the central trajectory of a linear program, submitted to *Mathematical Programming*, M.I.T. Sloan School working paper 3889-96 MSA, March, 1996.

[6] C. Gonzaga, Path following methods for linear programming, *SIAM Review* 34(2) 167-227, 1992.

[7] O. Güler, Barrier functions in interior point methods, *Mathematics of Operations Research* 21 (1996), to appear.

[8] D. den Hertog, *Interior point approach to linear, quadratic and convex programming, algorithms, and Complexity*, Kluwer Publishers, Dordrecht, The Netherlands, 1994.

[9] B. Jansen, C. Roos, and T. Terlaky, A short survey on ten years interior point methods, Report 95-45, Delft University of Technology, Delft, The Netherlands, 1995.

[10] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* 4 373-395, 1984.

[11] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A unified approach to interior point algorithms for linear complementarity problems*, Lecture Notes in Computer Science 538, Springer-Verlag, Berlin (1991).

[12] M. Kojima, N. Megiddo, and S. Mizuno, A general framework of continuation methods for complementarity problem, *Mathematics of Operations Research* 18 945-963, 1994.

[13] I. Lustig, R. Marsten, and D. Shanno, Interior point methods: computational state of the art, *ORSA Journal on Computing* 6 1-14, 1994.

[14] S. Mizuno, Infeasible-interior-point algorithms, in *Interior point methods in mathematical programming*, T. Terlaky, ed., Kluwer Academic Publishers, 1996.

[15] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, SIAM Publications, Philadelphia, U.S.A., 1994.

[16] J. Renegar, Linear programming, complexity theory and elementary functional analysis, *Mathematical Programming* 70, 279-351, 1995.

[17] R. Saigal, *Linear Programming: A Modern Integrated Analysis*, Kluwer Academic Publishers, Boston, 1995.

[18] M. Todd, Potential reduction methods in mathematical programming, *Mathematical Programming B*, to appear.

[19] L. Vandenberghe and S. Boyd, Semidefinite programming, *SIAM Review* 38 1 49-95, 1996.

[20] S. Vavasis and Y. Ye, A primal-dual interior point method whose running time depends only on the constraint matrix, *Mathematical Programming* 74 79-120, 1996.

[21] S. Wright, *Primal-Dual Interior Point Algorithms*, SIAM Publications, Philadelphia, forthcoming.

[22] X. Xu, P.-F. Hung, and Y. Ye, A simplified homogeneous and self-dual linear programming algorithm and its implementation, *Annals of Operations Research* 62, 151-171, 1996.

[23] Y. Ye, M. Todd, and S. Mizuno, An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm, *Mathematics of Operations Research* 19 53-67, 1994.