SEMIDEFINITE PROGRAMMING*

LIEVEN VANDENBERGHE† AND STEPHEN BOYD‡

Abstract. In semidefinite programming one minimizes a linear function subject to the constraint that an affine combination of symmetric matrices is positive semidefinite. Such a constraint is nonlinear and nonsmooth, but convex, so semidefinite programs are convex optimization problems. Semidefinite programming unifies several standard problems (*e.g.*, linear and quadratic programming) and finds many applications in engineering and combinatorial optimization.

Although semidefinite programs are much more general than linear programs, they are not much harder to solve. Most interior-point methods for linear programming have been generalized to semidefinite programs. As in linear programming, these methods have polynomial worst-case complexity, and perform very well in practice.

This paper gives a survey of the theory and applications of semidefinite programs, and an introduction to primal-dual interior-point methods for their solution.

Key words. Semidefinite programming, convex optimization, interior-point methods, eigenvalue optimization, combinatorial optimization, system and control theory

AMS subject classifications. 65K05, 49M45, 93B51, 90C25, 90C27, 90C90, 15A18

1. Introduction.

1.1. Semidefinite programming. We consider the problem of minimizing a linear function of a variable $x \in \mathbf{R}^m$ subject to a matrix inequality:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F(x) \geq 0 \end{array} \tag{1}$$

where

$$F(x) \stackrel{\Delta}{=} F_0 + \sum_{i=1}^{m} x_i F_i.$$

The problem data are the vector $c \in \mathbf{R}^m$ and $m+1$ symmetric matrices $F_0, \ldots, F_m \in \mathbf{R}^{n \times n}$. The inequality sign in $F(x) \geq 0$ means that $F(x)$ is positive semidefinite, *i.e.*, $z^T F(x) z \geq 0$ for all $z \in \mathbf{R}^n$.

We call the inequality $F(x) \geq 0$ a *linear matrix inequality* and the problem (1) a *semidefinite program.* A semidefinite program is a convex optimization problem since its objective and constraint are convex: if $F(x) \geq 0$ and $F(y) \geq 0$, then, for all λ, $0 \leq \lambda \leq 1$,

$$F(\lambda x + (1-\lambda y)) = \lambda F(x) + (1-\lambda) F(y) \geq 0.$$

Figure 1 depicts a simple example with $x \in \mathbf{R}^2$ and $F_i \in \mathbf{R}^{7 \times 7}$. Our goal here is to give the reader a generic picture that shows some of the features of semidefinite

[*] TO APPEAR IN **SIAM REVIEW**, MARCH 1996.

[†] Katholieke Universiteit Leuven, Departement Elektrotechniek, Afdeling ESAT, K. Mercierlaan 94, 3001 Leuven, Belgium (`Lieven.Vandenberghe@esat.kuleuven.ac.be`). Supported by the Belgian National Fund for Scientific Research (NFWO), and by the Belgian program on interuniversity attraction poles (under IUAP 17 and 50).

[‡] Information Systems Laboratory, Electrical Engineering Department, Stanford University, Stanford CA 94305 (`boyd@isl.stanford.edu`). Supported in part by AFOSR (under F49620-92-J-0013), NSF (under ECS-9222391 and EEC-9420565), and ARPA (under F49620-93-1-0085).

programs, so the specific values of the data are not relevant. The boundary of the feasible region is shown as the dark curve. The feasible region, *i.e.*, $\{x | F(x) \geq 0\}$, consists of this boundary curve along with the region it encloses. Very roughly speaking the semidefinite programming problem is to move as far as possible in the direction $-c$, while staying in the feasible region. For this semidefinite program there is one optimal point, x_{opt}.

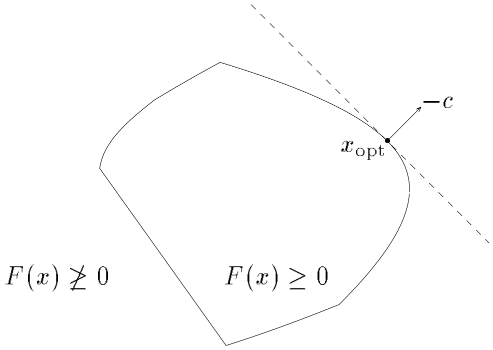

FIG. 1. *A simple semidefinite program with* $x \in \mathbf{R}^2$, $F(x) \in \mathbf{R}^{7\times 7}$.

This simple example demonstrates several general features of semidefinite programs. We have already mentioned that the feasible set is convex. Note that the optimal solution x_{opt} is on the boundary of the feasible set, *i.e.*, $F(x_{\text{opt}})$ is singular; in the general case there is always an optimal point on the boundary (provided the problem is feasible). In this example, the boundary of the feasible set is not smooth. It is piecewise smooth: it consists of two line segments and two smooth curved segments. In the general case the boundary consists of piecewise algebraic surfaces. Skipping some technicalities, the idea is as follows. At a point where the boundary is smooth, it is defined locally by some specific minors of the matrix $F(x)$ vanishing. Thus the boundary is locally the zero set of some polynomials in $x_1, \ldots, x_m$, *i.e.*, an algebraic surface.

Although the semidefinite program (1) may appear quite specialized, we will see that it includes many important optimization problems as special cases. For instance, consider the linear program (LP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax + b \geq 0 \end{array} \tag{2}$$

in which the inequality denotes *componentwise* inequality.[1] Since a vector $v \geq 0$ (componentwise) if and only if the matrix $\mathbf{diag}(v)$ (*i.e.*, the diagonal matrix with the components of v on its diagonal) is positive semidefinite, we can express the LP (2) as a semidefinite program with $F(x) = \mathbf{diag}(Ax + b)$, *i.e.*,

$$F_0 = \mathbf{diag}(b), \qquad F_i = \mathbf{diag}(a_i), \quad i = 1, \ldots, m,$$

where $A = [a_1 \ldots a_m] \in \mathbf{R}^{n\times m}$. In this case, of course, the feasible set is polyhedral; the boundary cannot be curved as in the general semidefinite program or the example shown in Figure 1.

[1] Thus $x \leq y$ denotes componentwise inequality when x and y are vectors, and matrix inequality when x and y are (symmetric) matrices. In this paper, the context will always make it clear which is meant.

Semidefinite programming can be regarded as an extension of linear programming where the componentwise inequalities between vectors are replaced by matrix inequalities, or, equivalently, the first orthant is replaced by the cone of positive semidefinite matrices. We can also view the semidefinite program (1) as a semi-infinite linear program, since the matrix inequality $F(x) \geq 0$ is equivalent to an infinite set of linear constraints on x, *i.e.*, $z^T F(x) z \geq 0$ for each $z \in \mathbf{R}^n$. It is therefore not surprising that the theory of semidefinite programming closely parallels the theory of linear programming, or that many algorithms for solving linear programs should have generalizations that handle semidefinite programs. There are some important differences, however: Duality results are weaker for semidefinite programs than for LPs, and there is no straightforward or practical simplex method for semidefinite programs.[2]

Before proceeding further we give a simple example of a *nonlinear* (convex) optimization problem that can be cast as a semidefinite program, but not as an LP. Consider the problem

$$\begin{array}{ll} \text{minimize} & \dfrac{(c^T x)^2}{d^T x} \\ \text{subject to} & Ax + b \geq 0 \end{array} \tag{3}$$

where we assume that $d^T x > 0$ whenever $Ax+b \geq 0$. We start with the standard trick of introducing an auxiliary variable t that serves as an upper bound on the objective:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & Ax + b \geq 0 \\ & \dfrac{(c^T x)^2}{d^T x} \leq t. \end{array} \tag{4}$$

In this formulation, the objective is a linear function of the variables x and t; the nonlinear (convex) objective in (3) shows up as a nonlinear (convex) constraint in (4). These constraints, in turn, can be expressed as a linear matrix inequality in the variables x and t:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{ccc} \mathbf{diag}(Ax+b) & 0 & 0 \\ 0 & t & c^T x \\ 0 & c^T x & d^T x \end{array} \right] \geq 0. \end{array} \tag{5}$$

Thus we have reformulated the nonlinear (convex) problem (3) as the semidefinite program (5).

The linear matrix inequality in the constraint of the semidefinite program (5) demonstrates two standard tricks: representing multiple linear matrix inequalities as one block-diagonal matrix inequality, and using *Schur complements* to represent a nonlinear convex constraint as a linear matrix inequality. Of course the 2×2 matrix inequality

$$\left[\begin{array}{cc} t & c^T x \\ c^T x & d^T x \end{array} \right] \geq 0 \tag{6}$$

[2] See however Anderson and Nash [9] for simplex-like methods in semi-infinite linear progamming, and Pataki [86] and Lasserre [62] for extensions of the simplex method to semidefinite programming.

is equivalent to $d^Tx \geq 0$ and $t - (c^Tx)^2/d^Tx \geq 0$ (with $t \geq 0$, $c^Tx = 0$ if $d^Tx = 0$). Since we have assumed that $Ax + b \geq 0$ implies $d^Tx > 0$, we see that the constraints in (4) are equivalent to the matrix inequality in (5). The expression $t - (c^Tx)^2/d^Tx$ is called the *Schur complement* of d^Tx in the matrix inequality (6). Recognizing Schur complements in nonlinear expressions is often the key step in reformulating nonlinear convex optimization problems as semidefinite programs.

There are good reasons for studying semidefinite programming. First, positive semidefinite (or definite) constraints arise directly in a number of important applications. Secondly, many convex optimization problems, *e.g.*, linear programming and (convex) quadratically constrained quadratic programming, can be cast as semidefinite programs, so semidefinite programming offers a unified way to study the properties of and derive algorithms for a wide variety of convex optimization problems. Most importantly, however, *semidefinite programs can be solved very efficiently, both in theory and in practice.*

Theoretical tractability follows from convexity, along with the observation that we can construct, in polynomial time, a cutting plane for the constraint set through any given infeasible point (see, *e.g.*, [17, §2.3], or [101]). One can therefore apply the ellipsoid method of Yudin and Nemirovsky, and Shor (see [115, 99]) to solve problem (1) in polynomial time. In practice, however, the ellipsoid method is slow.

Some general methods for nondifferentiable convex optimization are described by Shor [100], Kiwiel [57], and Hiriart-Urruty and Lemaréchal [47]. These methods are more efficient in practice than the ellipsoid method, and can be used to solve semidefinite programs.

In this paper we consider recently developed interior-point methods for semidefinite programming. These methods enjoy several properties that make them especially attractive.

- *Practical efficiency.* It is now generally accepted that interior-point methods for LPs are competitive with the simplex method and even faster for problems with more than $10,000$ variables or constraints (see, *e.g.*, [66]). Similarly, our experience with system and control applications suggests that interior-point methods for semidefinite programs are competitive with other methods for small problems, and substantially faster for medium and large-scale problems. As a very rough rule-of-thumb, interior-point methods solve semidefinite programs in about 5-50 iterations; each iteration is basically a least-squares problem of the same size as the original problem.
- *Theoretical efficiency.* A worst-case analysis of interior-point methods for semidefinite programming shows that the effort required to solve a semidefinite program to a given accuracy grows no faster than a polynomial of the problem size.
- *Ability to exploit problem structure.* Most of the computational effort in an interior-point method for semidefinite programming is in the least-squares problems that must be solved at each iteration. These least-squares problems can be solved by iterative methods such as conjugate-gradients, which can take advantage of problem structure. Sparsity is one well-known example of structure; in engineering applications many other types arise (*e.g.*, Toeplitz structure).

1.2. Historical overview. An early paper on the theoretical properties of semidefinite programs is Bellman and Fan [13]. Other references discussing optimality conditions are Craven and Mond [20], Shapiro [97], Fletcher [31], Allwright [7], Wolkowicz

[112], and Kojima, Kojima and Hara [58].

Many researchers have worked on the problem of minimizing the maximum eigenvalue of a symmetric matrix, which can be cast as a semidefinite program (see §2). See, for instance, Cullum, Donath and Wolfe [21], Goh and Teo [41], Panier [84], Allwright [8], Overton [78, 79], Overton and Womersley [81, 80], Ringertz [93], Fan and Nekooie [27], Fan [26], Hiriart-Urruty and Ye [48], Shapiro and Fan [98], and Pataki [85].

Interior-point methods for LPs were introduced by Karmarkar in 1984 [55], although many of the underlying principles are older (see, *e.g.*, Fiacco and McCormick [29], Lieu and Huard [63], and Dikin [23]). Karmarkar's algorithm, and the interior-point methods developed afterwards, combine a very low, polynomial, worst-case complexity with excellent behavior in practice. Karmarkar's paper has had an enormous impact, and several variants of his method have been developed (see, *e.g.*, the survey by Gonzaga [42]). Interior-point methods were subsequently extended to handle convex quadratic programming, and to certain linear complementarity problems (see, *e.g.*, Kojima, Megiddo, Noma and Yoshise [59]).

An important breakthrough was achieved by Nesterov and Nemirovsky in 1988 [71, 73, 72, 74]. They showed that interior-point methods for linear programming can, in principle, be generalized to all convex optimization problems. The key element is the knowledge of a barrier function with a certain property: self-concordance. To be useful in practice, the barrier (or really, its first and second derivatives) must also be computable. Nesterov and Nemirovsky show that a self-concordant barrier function exists for every convex set, but unfortunately their universal self-concordant barrier is not readily computable.

Semidefinite programs are an important class of convex optimization problems for which readily computable self-concordant barrier functions are known, and, therefore, interior-point methods are applicable. At the same time, they offer a simple conceptual framework and make possible a self-contained treatment of interior-point methods for many convex optimization problems.

Independently of Nesterov and Nemirovsky, Alizadeh [3] and Kamath and Karmarkar [52, 53] generalized interior-point methods from linear programming to semidefinite programming. Other recent articles on interior-point methods for semidefinite programming are Jarre [50], Vandenberghe and Boyd [108], Rendl, Vanderbei and Wolkowicz [92], Alizadeh, Haeberly and Overton [5], Kojima, Shindoh and Hara [61], Faybusovich [28], Gahinet and Nemirovsky [36], and Freund [34]. An excellent reference on interior-point methods for general convex problems is Den Hertog [22].

1.3. Outline. In §2 we describe several applications of semidefinite programming. Section §3 covers duality theory for semidefinite programs, emphasizing the similarities and differences with linear programming duality. In §4 we introduce a barrier function for linear matrix inequalities, and the concepts of central points and central path. In §5 we describe several primal-dual interior-point methods for semidefinite programming. These methods require feasible primal and dual initial points; §6 describes some methods for finding such points, or modifying the algorithms to remove this requirement. In §7 we describe a few extensions of semidefinite programming, including techniques for exploiting problem structure.

In this survey we emphasize primal-dual methods, and do not consider several important and useful algorithms, for instance the projective method of Karmarkar (or, rather, its generalization to semidefinite programs given in [76, §4.3]) and the projective method of Nesterov and Nemirovsky [76, §4.4], [69]. Our motivation for the restriction to primal-dual methods is twofold. Primal-dual methods are generally more efficient in practice, and, secondly, their behavior is often easier to analyze. Finally, all interior-point methods for semidefinite programs are based on similar principles, so we hope that this paper can serve as a tutorial introduction to other interior-point methods not covered here.

2. Examples.

In this section we list a few examples and applications of semidefinite programming. The list is not exhaustive, and our treatment of the two most prominent application areas, combinatorial optimization and control theory, is only cursory. Surveys of semidefinite programming in both fields have already appeared; see [17] for control theory, and [4] for combinatorial optimization. Our purpose is to give an idea of the generality of the problem, to point out the connections between applications in different areas, and to provide references for further reading.

See also Nesterov and Nemirovsky [76, §6.4] for more examples.

Quadratically constrained quadratic programming. A convex quadratic constraint $(Ax+b)^T(Ax+b) - c^Tx - d \le 0$, with $x \in \mathbf{R}^k$, can be written as

$$\left[\begin{array}{cc} I & Ax+b \\ (Ax+b)^T & c^Tx+d \end{array}\right] \ge 0.$$

The left-hand side depends affinely on the vector x: it can be expressed as

$$F(x) = F_0 + x_1F_1 + \cdots + x_kF_k \ge 0,$$

with

$$F_0 = \left[\begin{array}{cc} I & b \\ b^T & d \end{array}\right], \quad F_i = \left[\begin{array}{cc} 0 & a_i \\ a_i^T & c_i \end{array}\right], \quad i = 1, \ldots, k,$$

where $A = [a_1 \ldots a_k]$. Therefore, a general (convex) *quadratically constrained quadratic program* (QCQP)

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \ldots, L \end{array}$$

where each f_i is a convex quadratic function $f_i(x) = (A_ix+b)^T(A_ix+b) - c_i^Tx - d_i$, can be written as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{cc} I & A_0x+b_0 \\ (A_0x+b_0)^T & c_0^Tx+d_0+t \end{array}\right] \ge 0, \\ & \left[\begin{array}{cc} I & A_ix+b_i \\ (A_ix+b_i)^T & c_i^Tx+d_i \end{array}\right] \ge 0, \quad i = 1, \ldots, L, \end{array}$$

which is a semidefinite program with variables $x \in \mathbf{R}^k$ and $t \in \mathbf{R}$. This semidefinite program has dimensions $m = k+1$ and $n = n_0 + \cdots + n_L$, where $A_i \in \mathbf{R}^{n_i \times k}$.

While it is interesting to note that QCQPs can be cast as semidefinite programming problems, it may not be a good idea algorithmically. Indeed, a more efficient interior-point method for QCQPs can be developed using Nesterov and Nemirovsky's formulation as a problem over the second-order cone (see [76, §6.2.3]). The semidefinite programming formulation will be less efficient, especially when the matrices A_i have high rank.

Maximum eigenvalue and matrix norm minimization. Suppose the symmetric matrix $A(x)$ depends affinely on $x \in \mathbf{R}^k$: $A(x) = A_0 + x_1 A_1 + \cdots + x_k A_k$, where $A_i = A_i^T \in \mathbf{R}^{p \times p}$. The problem of minimizing the maximum eigenvalue of the matrix $A(x)$ can be cast as the semidefinite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tI - A(x) \geq 0, \end{array}$$

with variables $x \in \mathbf{R}^k$ and $t \in \mathbf{R}$. Problems of this type arise in control theory, structural optimization, graph theory and combinatorial optimization, and other fields. See Overton [79], Mohar and Poljak [67], and Grötschel, Lovász and Schrijver [44, Chapter 9] for surveys.

Several interesting related problems can be solved using semidefinite programming. As an example, to minimize the sum of the r largest eigenvalues of $A(x)$, one solves the semidefinite program in t, $X = X^T$, and x

$$\begin{array}{ll} \text{minimize} & rt + \mathbf{Tr} X \\ \text{subject to} & tI + X - A(x) \geq 0 \\ & X \geq 0. \end{array} \tag{7}$$

Here $\mathbf{Tr} X$ denotes the trace of a symmetric matrix $X \in \mathbf{R}^{p \times p}$, *i.e.*, $\mathbf{Tr} X = X_{11} + \cdots + X_{pp}$. For a proof, see Nesterov and Nemirovsky [76, p.238], or Alizadeh [1, §2.2]. The semidefinite program (7) has dimensions $m = 1 + k + p(p+1)/2$ and $n = 2p$. These results can also be extended to problems involving absolute values of the eigenvalues, or weighted sums of eigenvalues (see Alizadeh [1, Chapter 2]).

Another related problem is to minimize the (spectral, or maximum singular value) norm $\|A(x)\|$ of a matrix $A(x) = A_0 + x_1 A_1 + \cdots + x_k A_k \in \mathbf{R}^{p \times q}$. (Here the A_i need not be symmetric.) This can be cast as the semidefinite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{cc} tI & A(x) \\ A(x)^T & tI \end{array} \right] \geq 0, \end{array} \tag{8}$$

with variables $x \in \mathbf{R}^k$ and $t \in \mathbf{R}$. The semidefinite program (8) has dimensions $m = k + 1$ and $n = p + q$.

Note that the objectives in these problems, *i.e.*, the maximum eigenvalue, the sum of the largest eigenvalues, and the norm, are nondifferentiable (but of course convex) functions of x.

Logarithmic Chebychev approximation. Suppose we wish to solve $Ax \approx b$ approximately, where $A = [a_1 \ldots a_p]^T \in \mathbf{R}^{p \times k}$ and $b \in \mathbf{R}^p$. In Chebychev approximation we minimize the ℓ_∞-norm of the residual, *i.e.*, we solve

$$\text{minimize} \;\; \max_i |a_i^T x - b_i|.$$

This can be cast as a linear program, with x and an auxiliary variable t as variables:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t \leq a_i^T x - b_i \leq t, \quad i = 1, \ldots, p. \end{array}$$

In some applications b_i has the dimension of a power or intensity, and is typically expressed on a logarithmic scale. In such cases the more natural optimization problem is

$$\text{minimize} \;\; \max_i |\log(a_i^T x) - \log(b_i)| \tag{9}$$

(assuming $b_i > 0$, and interpreting $\log(a_i^T x)$ as $-\infty$ when $a_i^T x \leq 0$).

This *logarithmic Chebychev approximation* problem can be cast as a semidefinite program. To see this, note that

$$|\log(a_i^T x) - \log(b_i)| = \log \max(a_i^T x/b_i, b_i/a_i^T x)$$

(assuming $a_i^T x > 0$). Problem (9) is therefore equivalent to

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & 1/t \leq a_i^T x/b_i \leq t, \quad i = 1, \ldots, p, \end{array}$$

or,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{ccc} t - a_i^T x/b_i & 0 & 0 \\ 0 & a_i^T x/b_i & 1 \\ 0 & 1 & t \end{array}\right] \geq 0, \quad i = 1, \ldots, p \end{array}$$

which is a semidefinite program. This example illustrates two important points. It shows that semidefinite programming includes many optimization problems that do not look like (1) at first sight. And secondly, it shows that the problem is much more general than linear programming, despite the close analogy.

Structural optimization. Ben-Tal and Bendsøe in [14] consider the following problem from structural optimization. A structure of k linear elastic bars connects a set of p nodes. The geometry (topology and lengths of the bars) and the material (Young's modulus) are fixed; the task is to size the bars, *i.e.*, determine appropriate cross-sectional areas for the bars. In the simplest version of the problem we consider one fixed set of externally applied nodal forces f_i, $i = 1, \ldots, p$. (More complicated versions consider multiple loading scenarios.) The vector of (small) node displacements resulting from the load forces f will be denoted d. The objective is the elastic stored energy $\frac{1}{2} f^T d$, which is a measure of the inverse of the stiffness of the structure. We also need to take into account constraints on the total volume (or equivalently, weight), and upper and lower bounds on the cross-sectional area of each bar.

The design variables are the cross-sectional areas x_i. The relation between f and d is linear: $f = A(x)d$, where

$$A(x) \triangleq \sum_{i=1}^{k} x_i A_i$$

is called the stiffness matrix. The matrices A_i are all symmetric positive semidefinite and depend only on fixed parameters (Young's modulus, length of the bars, and geometry). The optimization problem then becomes

$$\begin{array}{ll} \text{minimize} & f^T d \\ \text{subject to} & f = A(x)d \\ & \displaystyle\sum_{i=1}^{k} l_i x_i \leq v \\ & \underline{x}_i \leq x_i \leq \overline{x}_i, \quad i = 1, \ldots, k \end{array}$$

where d and x are the variables, v is maximum volume, l_i are the lengths of the bars, and $\underline{x}_i$, $\overline{x}_i$ the lower and upper bounds on the cross-sectional areas. For simplicity, we assume that $\underline{x}_i > 0$, and that $A(x) > 0$ for all positive values of x_i. We can then eliminate d and write

$$\begin{array}{ll} \text{minimize} & f^T A(x)^{-1} f \\ \text{subject to} & \sum_{i=1}^{k} l_i x_i \le v \\ & \underline{x}_i \le x_i \le \overline{x}_i, \quad i = 1, \ldots, k \end{array}$$

or,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} t & f^T \\ f & A(x) \end{bmatrix} \ge 0 \\ & \sum_{i=1}^{k} l_i x_i \le v \\ & \underline{x}_i \le x_i \le \overline{x}_i, \quad i = 1, \ldots, k, \end{array}$$

which is a semidefinite program in x and t. (Note that we have used Schur complements to express $f^T A(x)^{-1} f \le t$ as a linear matrix inequality.)

Pattern separation by ellipsoids. The simplest classifiers in pattern recognition use hyperplanes to separate two sets of points $\{x^1, \ldots, x^K\}$ and $\{y^1, \ldots, y^L\}$ in $\mathbf{R}^p$. The hyperplane defined by $a^T x + b = 0$ separates these two sets if

$$\begin{array}{ll} a^T x^i + b \le 0, & i = 1, \ldots, K, \\ a^T y^j + b \ge 0, & j = 1, \ldots, L. \end{array}$$

This is a set of linear inequalities in $a \in \mathbf{R}^p$ and $b \in \mathbf{R}$ which can be solved by linear programming. If the two sets cannot be separated by a hyperplane, we can try to separate them by a quadratic surface. In other words we seek a quadratic function $f(x) = x^T A x + b^T x + c$ that satisfies

$$(x^i)^T A x^i + b^T x^i + c \le 0, \qquad i = 1, \ldots, K, \tag{10}$$

$$(y^j)^T A y^j + b^T y^j + c \ge 0, \qquad j = 1, \ldots, L. \tag{11}$$

These inequalities are a set of linear inequalities in the variables $A = A^T \in \mathbf{R}^{p \times p}$, $b \in \mathbf{R}^p$, and $c \in \mathbf{R}$. (So the total number of variables is $\frac{p(p+1)}{2} + p + 1$.) Again the problem can be solved using linear programming.

We can put further restrictions on the quadratic surface separating the two sets. As an example, for cluster analysis we might try to find an *ellipsoid* that contains all the points x^i and none of the y^j (see Rosen [95]). This constraint imposes the condition $A > 0$ in addition to the linear inequalities (10) and (11) on the variables A, b, and c. Thus we can find an ellipsoid that contains all the x^i but none of the y^j (or determine that no such ellipsoid exists) by solving a semidefinite programming feasibility problem.

We can optimize the shape and the size of the ellipsoid by adding an objective function and other constraints. For example, we can search for the "most spherical" ellipsoid as follows. The ratio of the largest to the smallest semi-axis length is the square root of the condition number of A. In order to make the ellipsoid as spherical as possible, one can introduce an additional variable γ, add the constraint

$$I \le A \le \gamma I, \tag{12}$$

and minimize γ over (10), (11) and (12). This is a semidefinite program in the variables γ, A, b and c. This semidefinite program is feasible if and only if there is an ellipsoid that contains all the x^i and none of the y^j; its optimum value is one if and only there is a *sphere* that separates the two sets of points. A simple example is shown in Figure 2 below.

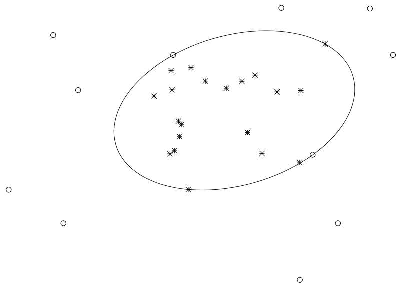

FIG. 2. *Cluster analysis using ellipsoids. The ellipsoid shown minimizes condition number among all ellipsoids containing all the x^i (shown as stars) and none of the y^j (shown as circles). Finding such an ellipsoid can be cast as a semidefinite program, hence efficiently solved.*

Statistics. Semidefinite programs arise in minimum trace factor analysis (see Bentler and Woodward [15], Fletcher [30, 31], Shapiro [96], Watson [110]).

Assume $x \in \mathbf{R}^p$ is a random vector, with mean $\bar{x}$ and covariance matrix Σ. We take a large number of samples $y = x + n$, where the measurement noise n has zero mean, is uncorrelated with x, and has an unknown but diagonal covariance matrix D. It follows that $\hat{\Sigma} = \Sigma + D$, where $\hat{\Sigma}$ denotes the covariance matrix of y. We assume that we can estimate $\hat{\Sigma}$ with high confidence, *i.e.*, we consider it a known, constant matrix.

We do not know Σ, the covariance of x, or D, the covariance matrix of the measurement noise. But they are both positive semidefinite, so we know that Σ lies in the convex set

$$\mathbf{\Sigma} \stackrel{\Delta}{=} \{ \; \hat{\Sigma} - D \mid \hat{\Sigma} - D \geq 0, \; D > 0, \; D \text{ diagonal } \}.$$

This fact allows us to derive bounds on linear functions of Σ, by solving semidefinite programming problems over the set $\mathbf{\Sigma}$.

As an example, consider the sum of the components of x, *i.e.*, $e^T x$ where e is the vector with all components equal to one. The variance of $e^T x$ is given by

$$e^T \Sigma e = e^T \hat{\Sigma} e - \mathbf{Tr} D.$$

We do not know Σ, and so cannot say exactly what $e^T \Sigma e$ is. But by determining the maximum and minimum of $e^T \Sigma e$ over the set $\mathbf{\Sigma}$, we can determine an interval in which it lies. In other words, we can compute upper and lower bounds on $e^T \Sigma e$.

It is easy to see that the upper bound is $e^T \hat{\Sigma} e$. A lower bound can be computed by solving the semidefinite program

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{p} d_i \\ \text{subject to} & \hat{\Sigma} - \mathbf{diag}(d) \geq 0 \\ & d \geq 0. \end{array} \tag{13}$$

Fletcher [30] calls (13) the *educational testing problem.* The vector y gives the scores of a random student on a series of p tests, and $e^T y$ gives the total score. One considers the test to be reliable if the variance of the measured total scores $e^T y$ is close to the variance of $e^T x$ over the entire population. The quantity

$$\rho = \frac{e^T \Sigma e}{e^T \hat{\Sigma} e}$$

is called the *reliability* of the test. By solving the semidefinite program (13) one can compute a lower bound for ρ.

Semidefinite programming also has applications in optimal experiment design (see Pukelsheim [88]).

Geometrical problems involving quadratic forms. Many geometrical problems involving quadratic functions can be expressed as semidefinite programs. We will give one simple example. Suppose we are given k ellipsoids $\mathcal{E}_1, \ldots, \mathcal{E}_k$ described as the sublevel sets of the quadratic functions

$$f_i(x) = x^T A_i x + 2b_i^T x + c_i, \quad i = 1, \ldots, k,$$

i.e., $\mathcal{E}_i = \{x | f_i(x) \leq 0\}$. The goal is to find the smallest ball that contains all k of these ellipsoids (or equivalently, contains the convex hull of their union).

The condition that one ellipsoid contain another can be expressed in terms of a matrix inequality. Suppose that the ellipsoids $\mathcal{E} = \{x | f(x) \leq 0\}$ and $\tilde{\mathcal{E}} = \{x | \tilde{f}(x) \leq 0\}$, with

$$f(x) = x^T A x + 2b^T x + c, \qquad \tilde{f}(x) = x^T \tilde{A} x + 2\tilde{b}^T x + \tilde{c},$$

have nonempty interior. Then it can be shown that $\mathcal{E}$ contains $\tilde{\mathcal{E}}$ if and only if there is a $\tau \geq 0$ such that

$$\begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \leq \tau \begin{bmatrix} \tilde{A} & \tilde{b} \\ \tilde{b}^T & \tilde{c} \end{bmatrix}.$$

(The 'if' part is trivial; the 'only if' part is less trivial. See [17, 105]).

Returning to our problem, consider the ball $\mathcal{S}$ represented by $f(x) = x^T x - 2x_c^T x + \gamma \leq 0$. $\mathcal{S}$ contains the ellipsoids $\mathcal{E}_1, \ldots, \mathcal{E}_k$ if and only if there are nonnegative $\tau_1, \ldots, \tau_k$ such that

$$\begin{bmatrix} I & -x_c \\ -x_c^T & \gamma \end{bmatrix} \leq \tau_i \begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix}, \quad i = 1, \ldots, k.$$

Note that these conditions can be considered one large linear matrix inequality in the variables x_c, γ, and $\tau_1, \ldots, \tau_k$.

Our goal is to minimize the radius of the ball $\mathcal{S}$, which is $r = \sqrt{x_c^T x_c - \gamma}$. To do this we express the condition $r^2 \leq t$ as the matrix inequality

$$\begin{bmatrix} I & x_c \\ x_c^T & t + \gamma \end{bmatrix} \geq 0$$

and minimize the variable t.

Putting it all together we see that we can find the smallest ball containing the ellipsoids $\mathcal{E}_1, \ldots, \mathcal{E}_k$ by solving the semidefinite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{cc} I & -x_c \\ -x_c^T & \gamma \end{array}\right] \leq \tau_i \left[\begin{array}{cc} A_i & b_i \\ b_i^T & c_i \end{array}\right], \quad i = 1, \ldots, k, \\ & \tau_i \geq 0, \quad i = 1, \ldots, k, \\ & \left[\begin{array}{cc} I & x_c \\ x_c^T & t + \gamma \end{array}\right] \geq 0. \end{array}$$

The variables here are x_c, $\tau_1, \ldots, \tau_k$, γ, and t.

This example demonstrates once again the breadth of problems that can be reformulated as semidefinite programs. It also demonstrates that the task of this reformulation can be nontrivial.

A simple example is illustrated in Figure 3 below.

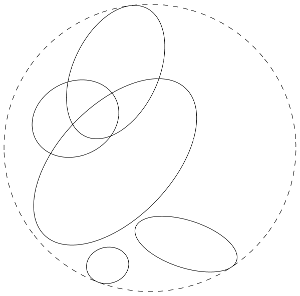

FIG. 3. *Smallest ball containing five ellipsoids. Finding such a ball can be cast as a semidefinite program, hence efficiently solved.*

Combinatorial and non-convex optimization. Semidefinite programs play a very useful role in non-convex or combinatorial optimization. Consider, for example, the quadratic optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \ i = 1, \ldots, L \end{array} \tag{14}$$

where $f_i(x) = x^T A_i x + 2b_i^T x + c_i$, $i = 0, 1, \ldots, L$. The matrices A_i can be indefinite, and therefore problem (14) is a very hard, non-convex optimization problem. For example, it includes all optimization problems with polynomial objective function and polynomial constraints (see [76, §6.4.4],[101]).

For practical purposes, *e.g.*, in branch-and-bound algorithms, it is important to have good and cheaply computable lower bounds on the optimal value of (14). Shor and others have proposed to compute such lower bounds by solving the semidefinite program (with variables t and τ_i)

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & \left[\begin{array}{cc} A_0 & b_0 \\ b_0^T & c_0 - t \end{array}\right] + \tau_1 \left[\begin{array}{cc} A_1 & b_1 \\ b_1^T & c_1 \end{array}\right] + \cdots + \tau_L \left[\begin{array}{cc} A_L & b_L \\ b_L^T & c_L \end{array}\right] \geq 0 \\ & \tau_i \geq 0, \quad i = 1, \ldots, L. \end{array} \tag{15}$$

One can easily verify that this semidefinite program yields lower bounds for (14). Suppose x satisfies the constraints in the nonconvex problem (14), *i.e.*,

$$f_i(x) = \begin{bmatrix} x \\ 1 \end{bmatrix}^T \begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} \leq 0$$

for $i = 1, \ldots, L$, and t, τ_1, $\ldots$, τ_L satisfy the constraints in the semidefinite program (15). Then

$$\begin{aligned} 0 &\leq \begin{bmatrix} x \\ 1 \end{bmatrix}^T \left(\begin{bmatrix} A_0 & b_0 \\ b_0^T & c_0 - t \end{bmatrix} + \tau_1 \begin{bmatrix} A_1 & b_1 \\ b_1^T & c_1 \end{bmatrix} + \cdots + \tau_L \begin{bmatrix} A_L & b_L \\ b_L^T & c_L \end{bmatrix} \right) \begin{bmatrix} x \\ 1 \end{bmatrix} \\ &= f_0(x) - t + \tau_1 f_1(x) + \cdots + \tau_L f_L(x) \\ &\leq f_0(x) - t. \end{aligned}$$

Therefore $t \leq f_0(x)$ for every feasible x in (14), as desired. Problem (15) can also be derived via Lagrangian duality; for a deeper discussion, see Shor [101], or Poljak, Rendl, and Wolkowicz [87].

Most semidefinite relaxations of NP-hard combinatorial problems seem to be related to the semidefinite program (15), or the related one,

$$(16) \qquad \begin{array}{ll} \text{minimize} & \mathbf{Tr} X A_0 + 2b_0^T x + c_0 \\ \text{subject to} & \mathbf{Tr} X A_i + 2b_i^T x + c_i \leq 0, \quad i = 1, \ldots, L \\ & \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \geq 0, \end{array}$$

where the variables are $X = X^T \in \mathbf{R}^{k \times k}$ and $x \in \mathbf{R}^k$. We will see in Section 3 that (16) is the *dual* of Shor's relaxation (15); the two problems (15) and (16) yield the same bound.

Note that the constraint

$$(17) \qquad \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \geq 0$$

is equivalent to $X \geq xx^T$. The semidefinite program (16) can therefore be directly interpreted as a relaxation of the original problem (14), which can be written as

$$(18) \qquad \begin{array}{ll} \text{minimize} & \mathbf{Tr} X A_0 + 2b_0^T x + c_0 \\ \text{subject to} & \mathbf{Tr} X A_i + 2b_i^T x + c_i \leq 0, \quad i = 1, \ldots, L \\ & X = xx^T. \end{array}$$

The only difference between (18) and (16) is the replacement of the (nonconvex) constraint $X = xx^T$ with the convex relaxation $X \geq xx^T$. It is also interesting to note that the relaxation (16) becomes the original problem (18) if we add the (nonconvex) constraint that the matrix on the left hand side of (17) is rank one.

As an example, consider the $(-1, 1)$-quadratic program

$$(19) \qquad \begin{array}{ll} \text{minimize} & x^T A x + 2b^T x \\ \text{subject to} & x_i^2 = 1, \quad i = 1, \ldots, k, \end{array}$$

which is NP-hard. The constraint $x_i \in \{-1, 1\}$ can be written as the quadratic equality constraint $x_i^2 = 1$, or, equivalently, as two quadratic inequalities $x_i^2 \leq 1$ and $x_i^2 \geq 1$. Applying (16) we find that the semidefinite program in $X = X^T$ and x

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr} XA + 2b^T x \\ \text{subject to} & X_{ii} = 1, \;\; i = 1, \ldots, k \\ & \left[\begin{array}{cc} X & x \\ x^T & 1 \end{array}\right] \geq 0 \end{array} \tag{20}$$

yields a lower bound for (19). In a recent paper on the MAX-CUT problem, which is a specific case of (19) where $b = 0$ and the diagonal of A is zero, Goemans and Williamson have *proved* that the lower bound from (20) is at most 14% suboptimal (see [39] and [40]). This is much better than any previously known bound. Similar strong results on semidefinite programming relaxations of NP-hard problems have been obtained by Karger, Motwani, and Sudan [54].

The usefulness of semidefinite programming in combinatorial optimization was recognized more than twenty years ago (see, *e.g.*, Donath and Hoffman [24]). Many people seem to have developed similar ideas independently. We should however stress the importance of the work by Grötschel, Lovász, and Schrijver [44, Chapter 9], [64] who have demonstrated the power of semidefinite relaxations on some very hard combinatorial problems. The recent development of efficient interior-point methods has turned these techniques into powerful practical tools; see Alizadeh [3, 1, 2], Kamath and Karmarkar [52, 53], Helmberg, Rendl, Vanderbei and Wolkowicz [46].

For a more detailed survey of semidefinite programming in combinatorial optimization, we refer the reader to the recent paper by Alizadeh [4].

Control and system theory. Semidefinite programming problems arise frequently in control and system theory; Boyd, El Ghaoui, Feron and Balakrishnan catalog many examples in [17]. We will describe one simple example here.

Consider the *differential inclusion*

$$\frac{dx}{dt} = Ax(t) + Bu(t), \qquad y(t) = Cx(t), \qquad |u_i(t)| \leq |y_i(t)|, \;\; i = 1, \ldots, p \tag{21}$$

where $x(t) \in \mathbf{R}^l$, $u(t) \in \mathbf{R}^p$, and $y(t) \in \mathbf{R}^p$. In the terminology of control theory, this is described as a linear system with uncertain, time-varying, unity-bounded, diagonal feedback.

We seek an invariant ellipsoid, *i.e.*, an ellipsoid $\mathcal{E}$ such that for any x and u that satisfy (21), $x(T) \in \mathcal{E}$ implies $x(t) \in \mathcal{E}$ for all $t \geq T$. The existence of such an ellipsoid implies, for example, that all solutions of the differential inclusion (21) are bounded.

The ellipsoid $\mathcal{E} = \{x \mid x^T P x \leq 1\}$, where $P = P^T > 0$, is invariant if and only if the function $V(t) = x(t)^T P x(t)$ is nonincreasing for any x and u that satisfy (21). In this case we say that V is a quadratic Lyapunov function that proves stability of the differential inclusion (21).

We can express the derivative of V as a quadratic form in $x(t)$ and $u(t)$:

$$\frac{d}{dt} V(x(t)) = \left[\begin{array}{c} x(t) \\ u(t) \end{array}\right]^T \left[\begin{array}{cc} A^T P + PA & PB \\ B^T P & 0 \end{array}\right] \left[\begin{array}{c} x(t) \\ u(t) \end{array}\right] \tag{22}$$

We can express the conditions $|u_i(t)| \leq |y_i(t)|$ as the quadratic inequalities

$$u_i^2(t) - y_i^2(t) = \left[\begin{array}{c} x(t) \\ u(t) \end{array}\right]^T \left[\begin{array}{cc} -c_i^T c_i & 0 \\ 0 & E_{ii} \end{array}\right] \left[\begin{array}{c} x(t) \\ u(t) \end{array}\right] \leq 0, \qquad i = 1, \ldots, p, \tag{23}$$

where c_i is the ith row of C, and E_{ii} is the matrix with all entries zero except the ii entry, which is 1.

Putting it all together we find that $\mathcal{E}$ is invariant if and only if (22) holds whenever (23) holds. Thus the condition is that one quadratic inequality should hold whenever some other quadratic inequalities hold, *i.e.*:

$$\text{for all } z \in \mathbf{R}^{l+p}, \quad z^T T_i z \leq 0, \ i = 1, \ldots, p \implies z^T T_0 z \leq 0 \tag{24}$$

where

$$T_0 = \begin{bmatrix} A^T P + PA & PB \\ B^T P & 0 \end{bmatrix}, \qquad T_i = \begin{bmatrix} -c_i^T c_i & 0 \\ 0 & E_{ii} \end{bmatrix}, \ i = 1, \ldots, p.$$

In the general case, simply verifying that (24) holds for a given P is very difficult. But an obvious sufficient condition is

$$\text{there exist } \ \tau_1 \geq 0, \ldots, \tau_p \geq 0 \ \text{ such that } \ T_0 \leq \tau_1 T_1 + \cdots + \tau_p T_p. \tag{25}$$

Replacing the condition (24) with the stronger condition (25) is called the $\mathcal{S}$-procedure in the Soviet literature on control theory, and dates back at least to 1944 (see [17, p.33], [33], [65]). Note the similarity between Shor's bound (see (14) and (15)) and the $\mathcal{S}$-procedure ((24) and (25)). Indeed Shor's bound is readily derived from the $\mathcal{S}$-procedure, and vice versa.

Returning to our example, we apply the $\mathcal{S}$-procedure to obtain a sufficient condition for invariance of the ellipsoid $\mathcal{E}$: for some $D = \mathbf{diag}(\tau_1, \ldots, \tau_p)$,

$$\begin{bmatrix} A^T P + PA + C^T DC & PB \\ B^T P & -D \end{bmatrix} \leq 0. \tag{26}$$

This is a linear matrix inequality in the (matrix) variables $P = P^T$ and (diagonal) D. Hence, by solving a semidefinite feasibility problem we can find an invariant ellipsoid (if the problem is feasible). One can also optimize various quantities over the feasible set; see [17]. Note that (26) is really a convex relaxation of the condition that $\mathcal{E}$ be invariant, obtained via the $\mathcal{S}$-procedure.

The close connections between the $\mathcal{S}$-procedure, used in control theory to form semidefinite programming relaxations of hard control problems, and the various semidefinite relaxations used in combinatorial optimization, do not appear to be well known.

3. Duality.

3.1. The dual semidefinite program. The *dual* problem associated with the semidefinite program (1) is

$$\begin{array}{ll} \text{maximize} & -\mathbf{Tr} F_0 Z \\ \text{subject to} & \mathbf{Tr} F_i Z = c_i,\ i = 1, \ldots, m \\ & Z \geq 0. \end{array} \tag{27}$$

Here the variable is the matrix $Z = Z^T \in \mathbf{R}^{n \times n}$, which is subject to m equality constraints and the matrix nonnegativity condition. Note that the objective function in (27) is a linear function of Z. We will say that $Z = Z^T \in \mathbf{R}^{n \times n}$ is *dual feasible* if $\mathbf{Tr} Z F_i = c_i$, $i = 1, \ldots, m$, and $Z \geq 0$. We will also refer to the original semidefinite program (1) as the *primal* problem.

The dual problem (27) is also a semidefinite program, *i.e.*, it can be put in the same form as the primal problem (1). Let us assume for simplicity that the matrices $F_1, \ldots, F_m$ are linearly independent. Then we can express the affine set

$$\left\{ Z \;\middle|\; Z = Z^T \in \mathbf{R}^{n \times n},\ \mathbf{Tr} F_i Z = c_i,\ \ i = 1, \ldots, m \right\}$$

in the form

$$\left\{ G(y) = G_0 + y_1 G_1 + \cdots + y_p G_p \;\middle|\; y \in \mathbf{R}^p \right\}$$

where $p = \frac{n(n+1)}{2} - m$ and the G_i are appropriate matrices. We define $d \in \mathbf{R}^p$ by $d_i = \mathbf{Tr} F_0 G_i$, so that $d^T y = \mathbf{Tr} F_0 (G(y) - G_0)$. Then the dual problem becomes (except for a constant term in the objective and a change of sign to transform maximization into minimization)

$$\begin{array}{ll} \text{minimize} & d^T y \\ \text{subject to} & G(y) \geq 0 \end{array}$$

which is a semidefinite program. It is possible to use notation that, unlike ours, emphasizes the complete symmetry between the primal and dual semidefinite programs (see, *e.g.*, Nesterov and Nemirovsky [76, §4.2]). Our notation is chosen to make the primal problem as "explicit" as possible, with x denoting a "free" variable.

As an illustration, let us apply the definition of the dual semidefinite program to the linear program (2), *i.e.*, take $F_0 = \mathbf{diag}(b)$ and $F_i = \mathbf{diag}(a_i)$. The dual semidefinite program (27) becomes

$$\begin{array}{ll} \text{maximize} & -\mathbf{Tr}\ \mathbf{diag}(b) Z \\ \text{subject to} & \mathbf{Tr}\ \mathbf{diag}(a_i) Z = c_i,\ \ i = 1, \ldots, m \\ & Z \geq 0. \end{array} \tag{28}$$

This semidefinite program can be simplified. The objective function and the equality constraints only involve the diagonal elements of Z. Moreover, if $Z \geq 0$ we can always replace the off-diagonal elements of Z by zeros, and still retain a positive semidefinite matrix. Instead of optimizing over all symmetric $n \times n$ matrices Z, we can therefore limit ourselves to diagonal matrices $Z = \mathbf{diag}(z)$. Problem (28) then reduces to

$$\begin{array}{ll} \text{maximize} & -b^T z \\ \text{subject to} & z \geq 0 \\ & a_i^T z = c_i,\ \ i = 1, \ldots, m, \end{array} \tag{29}$$

which is the familiar dual of the LP (2).

In this example the diagonal structure of the semidefinite program allows us to reduce the dual problem to one with fewer variables. In general, it is often the case that the dual problem can be simplified when the matrices F_i are structured. For example, if the matrix $F(x)$ is block diagonal, the dual variables Z can be assumed to have the same block diagonal structure.

Linear programming duality is very strong as a result of the polyhedral character of the feasible set: The optimum values of (2) and (29) are always equal, except in the pathological case where both problems are infeasible. (We adopt the standard convention that the optimum value of (2) is $+\infty$ if the problem is infeasible, and the optimum value of (29) is $-\infty$ if the dual problem is infeasible.) Duality results for general semidefinite programs are weaker, however, as we will see below.

Let us return to our discussion of the dual semidefinite program. The key property of the dual semidefinite program is that it yields bounds on the optimal value of the primal semidefinite program (and vice versa). Suppose that Z is dual feasible, and x is primal feasible. Then

$$c^T x + \mathbf{Tr} Z F_0 = \sum_{i=1}^{m} \mathbf{Tr} Z F_i x_i + \mathbf{Tr} Z F_0 = \mathbf{Tr} Z F(x) \geq 0, \tag{30}$$

in which we use the fact that $\mathbf{Tr} AB \geq 0$ when $A = A^T \geq 0$ and $B = B^T \geq 0$. Thus we have:

$$-\mathbf{Tr} F_0 Z \leq c^T x, \tag{31}$$

i.e., the dual objective value of any dual feasible point Z is smaller than or equal to the primal objective value of any primal feasible point x. We refer to the difference as the *duality gap* η associated with x and Z:

$$\eta \overset{\Delta}{=} c^T x + \mathbf{Tr} F_0 Z = \mathbf{Tr} F(x) Z. \tag{32}$$

Note that the duality gap is a *linear* function of x and Z, even though the second expression in (32) looks bilinear.

Let p^* denote the optimal value of the semidefinite program (1), *i.e.*,

$$p^* \overset{\Delta}{=} \inf \left\{ \; c^T x \;\middle|\; F(x) \geq 0 \; \right\},$$

and let Z be dual feasible. Since (31) holds for *any* feasible x, we conclude that $-\mathbf{Tr} Z F_0 \leq p^*$. In other words: dual feasible matrices yield lower bounds for the primal problem.

Similarly, primal feasible points yield upper bounds for the dual problem: $d^* \leq c^T x$, where d^* is the optimal value of the dual semidefinite program (1),

$$d^* \overset{\Delta}{=} \sup \left\{ \; -\mathbf{Tr} F_0 Z \;\middle|\; Z = Z^T \geq 0, \; \mathbf{Tr} F_i Z = c_i, \; i = 1, \ldots, m \; \right\}.$$

It follows that $d^* \leq p^*$, *i.e.*, the optimal value of the dual problem is less than or equal to the optimal value of the primal problem. In fact equality usually obtains. Let $\mathbf{X}_{\mathrm{opt}}$ and $\mathbf{Z}_{\mathrm{opt}}$ denote the primal and dual optimal sets, *i.e.*,

$$\begin{aligned}
\mathbf{X}_{\mathrm{opt}} &\overset{\Delta}{=} \left\{ x \;\middle|\; F(x) \geq 0 \text{ and } c^T x = p^* \right\}, \\
\mathbf{Z}_{\mathrm{opt}} &\overset{\Delta}{=} \left\{ Z \;\middle|\; Z \geq 0, \; \mathbf{Tr} F_i Z = c_i, \;\; i = 1, \ldots, m, \text{ and } -\mathbf{Tr} F_0 Z = d^* \right\}.
\end{aligned}$$

Note that $\mathbf{X}_{\rm opt}$ (or $\mathbf{Z}_{\rm opt}$) can be empty, even if p^* (or d^*) is finite, as in the semidefinite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} x & 1 \\ 1 & t \end{bmatrix} \geq 0. \end{array}$$

THEOREM 3.1. *We have $p^* = d^*$ if either of the following conditions holds.*

1. *The primal problem (1) is strictly feasible,* i.e., *there exists an x with $F(x) > 0$.*
2. *The dual problem (27) is strictly feasible,* i.e., *there exists a Z with $Z = Z^T > 0$, $\mathbf{Tr} F_i Z = c_i$, $i = 1, \ldots, m$.*

If both conditions hold, the optimal sets $\mathbf{X}_{\rm opt}$ and $\mathbf{Z}_{\rm opt}$ are nonempty.

For a proof, see Nesterov and Nemirovsky [76, §4.2], or Rockafellar [94, §30]. Theorem 3.1 is an application of standard duality in convex analysis, so the constraint qualification is not surprising or unusual. Wolkowicz [112], and Ramana [89, 90, 91] have formulated two different approaches to a duality theory for semidefinite programming that does not require strict feasibility. For our present purposes, the standard duality outlined above will be sufficient.

Assume the optimal sets are nonempty, *i.e.*, there exist feasible x and Z with

$$c^T x = -\mathbf{Tr} F_0 Z = p^* = d^*.$$

From (30), we have $\mathbf{Tr} F(x) Z = 0$. Since $F(x) \geq 0$ and $Z \geq 0$ we conclude that $Z F(x) = 0$. (Here we use the fact that if A and B are symmetric positive semidefinite and $\mathbf{Tr} AB = 0$, then $AB = 0$.) The condition $ZF(x) = 0$ is the *complementary slackness* condition; it says that the ranges of the symmetric matrices Z and $F(x)$ are orthogonal. This generalizes the familiar complementary slackness condition for the LP (2) and its dual (29). Taking $Z = \mathbf{diag}(z)$, and $F(x) = \mathbf{diag}(Ax + b)$, we see that $ZF(x) = 0$ if and only if $z_i(Ax + b)_i = 0$ for $i = 1, \ldots, n$, *i.e.*, the zero patterns in z and $Ax + b$ are complementary. (See [6] for a detailed analysis of complementarity in semidefinite programming.)

Theorem 3.1 gives us optimality conditions for the semidefinite program (1) if we assume strict primal and dual feasibility: x is optimal if and only if there is a Z such that

$$\begin{array}{l} F(x) \geq 0 \\ Z \geq 0, \quad \mathbf{Tr} F_i Z = c_i, \quad i = 1, \ldots, m \\ Z F(x) = 0. \end{array} \tag{33}$$

Example. We first consider a simple example where $p^* \neq d^*$:

$$\begin{array}{ll} \text{minimize} & x_1 \\ \text{subject to} & \begin{bmatrix} 0 & x_1 & 0 \\ x_1 & x_2 & 0 \\ 0 & 0 & x_1 + 1 \end{bmatrix} \geq 0. \end{array}$$

The feasible set is $\left\{ [x_1\ x_2]^T \mid x_1 = 0, x_2 \geq 0 \right\}$, and therefore $p^* = 0$. The dual problem can be simplified to

$$\begin{array}{ll} \text{maximize} & -z_2 \\ \text{subject to} & \begin{bmatrix} z_1 & (1 - z_2)/2 & 0 \\ (1 - z_2)/2 & 0 & 0 \\ 0 & 0 & z_2 \end{bmatrix} \geq 0, \end{array}$$

which has feasible set $\left\{[z_1\ z_2]^T \mid z_1 \geq 0, z_2 = 1\right\}$. The dual problem therefore has optimal value $d^* = -1$. Of course, this semidefinite program violates both conditions in Theorem 3.1. Both problems are feasible, but not strictly feasible. Note also the contrast with linear programming, where it is impossible to have a finite nonzero duality gap at the optimum.

Example. The educational testing problem (13) can be written as

$$\begin{array}{ll} \text{minimize} & -e^T d \\ \text{subject to} & \left[\begin{array}{cc} \hat{\Sigma} - \mathbf{diag}(d) & 0 \\ 0 & \mathbf{diag}(d) \end{array}\right] \geq 0 \end{array}$$

where e is a vector with all components equal to one. In other words, problem (13) is a semidefinite program with $c = -e$,

$$F_0 = \left[\begin{array}{cc} \hat{\Sigma} & 0 \\ 0 & 0 \end{array}\right] \text{ and } F_i = \left[\begin{array}{cc} -\mathbf{diag}(e^i) & 0 \\ 0 & \mathbf{diag}(e^i) \end{array}\right], \quad i = 1, \ldots, p$$

(e^i stands for the ith unit vector). Applying (27), we find as the dual problem

$$\begin{array}{ll} \text{maximize} & -\mathbf{Tr}\hat{\Sigma} Z_{11} \\ \text{subject to} & (-Z_{11} + Z_{22})_{ii} = -1 \\ & \left[\begin{array}{cc} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{array}\right] \geq 0. \end{array}$$

Without loss of generality, we can assume the off-diagonal block Z_{12} is zero, and Z_{22} is diagonal. These simplifications result in a more compact form of the dual:

$$\begin{array}{ll} \text{minimize} & \mathbf{Tr}\hat{\Sigma} Q \\ \text{subject to} & Q = Q^T \geq 0 \\ & Q_{ii} \geq 1, \quad i = 1, \ldots, p. \end{array}$$

Example. Consider the matrix norm minimization problem mentioned in §2:

$$\begin{array}{ll} \text{minimize} & \|A(x)\| \\ x \in \mathbf{R}^k & \end{array} \tag{34}$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_k A_k$. We remind the reader that $\|A(x)\|$ is the maximum singular value of $A(x)$.

The problem (34) is (an instance of) a basic problem that arises in the study of normed vector spaces; its optimum value is the norm of (the image of) A_0 in the quotient space of $\mathbf{R}^{p \times q}$ modulo span$\{A_1, \ldots, A_k\}$. In this context we encounter the following dual of (34):

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr} A_0^T Q \\ \text{subject to} & \mathbf{Tr} A_i^T Q = 0, \quad i = 1, \ldots, k \\ & \|Q\|_* \leq 1. \end{array} \tag{35}$$

Here $\|Q\|_*$ is the norm dual to the maximum singular value norm, *i.e.*,

$$\|Q\|_* = \sup\left\{ \mathbf{Tr} Y Q \mid \|Y\| \leq 1 \right\}.$$

It can be shown that $\|Q\|_*$ is the sum of the singular values of Q. It is also known that the optimal values of (34) and (35) are always equal.

Let us verify that this (normed vector space) notion of duality coincides with semidefinite programming duality. The dual semidefinite program of problem (8) is

$$\begin{array}{ll} \text{maximize} & -2\mathbf{Tr}A_0^T Z_{12} \\ \text{subject to} & \mathbf{Tr}A_i^T Z_{12} = 0, \quad i = 1, \ldots, k \\ & \mathbf{Tr}Z_{11} + \mathbf{Tr}Z_{22} = 1 \\ & \left[\begin{array}{cc} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{array}\right] \geq 0. \end{array} \tag{36}$$

This can be simplified. We have the following property: given a matrix Z_{12}, there exist Z_{11}, Z_{22} such that

$$\mathbf{Tr}Z_{11} + \mathbf{Tr}Z_{22} = 1, \quad \left[\begin{array}{cc} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{array}\right] \geq 0 \tag{37}$$

if and only if $\|Z_{12}\|_* \leq 1/2$.

To see this, let $Z_{12} = U\Sigma V^T$ be the singular value decomposition of Z_{12}. The matrix Σ is square, diagonal, and of size $\min\{p, q\}$, and its trace $\mathbf{Tr}\Sigma$ is equal to $\|Z_{12}\|_*$.

First assume that Z_{11} and Z_{22} satisfy (37). Then

$$\mathbf{Tr}\left[\begin{array}{cc} UU^T & -UV^T \\ -VU^T & VV^T \end{array}\right]\left[\begin{array}{cc} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{array}\right] \geq 0$$

because the trace of the product of two positive semidefinite matrices is always nonnegative. As a consequence,

$$\begin{array}{rcl} 0 & \leq & -2\mathbf{Tr}UV^T Z_{12}^T + \mathbf{Tr}UU^T Z_{11} + \mathbf{Tr}VV^T Z_{22} \\ & \leq & -2\mathbf{Tr}\Sigma + \mathbf{Tr}Z_{11} + \mathbf{Tr}Z_{22} \\ & = & -2\,\|Z_{12}\|_* + \mathbf{Tr}Z_{11} + \mathbf{Tr}Z_{22} \\ & = & -2\,\|Z_{12}\|_* + 1. \end{array}$$

So $\|Z_{12}\|_* \leq 1/2$.

To prove the converse, suppose that $\|Z_{12}\|_* \leq 1/2$. Then one can verify that

$$Z_{11} = U\Sigma U^T + \gamma I, \quad Z_{22} = V\Sigma V^T + \gamma I,$$

with $\gamma = (1 - 2\|Z_{12}\|_*)/(p+q)$, satisfy (37).

Problem (36) therefore reduces to

$$\begin{array}{ll} \text{maximize} & -2\mathbf{Tr}A_0^T Z_{12} \\ \text{subject to} & \mathbf{Tr}A_i^T Z_{12} = 0, \quad i = 1, \ldots, k \\ & \|Z_{12}\|_* \leq 1/2, \end{array}$$

which is the same as (35) with $Q = 2Z_{12}$.

Problem (8) is always strictly feasible; it suffices to choose $x = 0$ and $t > \|A_0\|$. Applying Theorem 3.1, we conclude that the optimal duality gap is always zero.

We refer the reader to the papers by Zietak [116, 117] and Watson [111] for more details and additional references on duality in matrix norm minimization.

Example. In §2 we claimed that the sum of the r largest eigenvalues of a matrix $A(x)$ can be minimized by solving the semidefinite program (7). This formulation is due to Alizadeh [3, §2.2] and Nesterov and Nemirovsky [76, §6.4.3]. Duality provides an elegant way to derive this result.

It is well known that the sum of the r largest eigenvalues of a matrix $A = A^T \in \mathbf{R}^{p\times p}$ can be expressed as the optimal value of the problem

$$\begin{array}{ll} \text{maximum} & \mathbf{Tr} W^T A W \\ \text{subject to} & W \in \mathbf{R}^{p\times r} \\ & W^T W = I. \end{array} \tag{38}$$

This result is attributed to Ky Fan [25]. Overton and Womersley [80] have observed that (38) can be expressed as the semidefinite program

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr} A Z_{11} \\ \text{subject to} & \mathbf{Tr} Z_{11} = r \\ & Z_{11} + Z_{22} = I \\ & \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{bmatrix} \geq 0. \end{array} \tag{39}$$

The equivalence can be seen as follows. The matrix Z_{12} can be assumed to be zero without loss of generality. The matrix Z_{22} acts as slack variable, and (39) simplifies to

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr} A Z_{11} \\ \text{subject to} & \mathbf{Tr} Z_{11} = r \\ & 0 \leq Z_{11} \leq I. \end{array} \tag{40}$$

Overton and Womersley have shown that the extreme points of the feasible set of (40) are precisely the matrices that have the form $Z_{11} = WW^T$ for some $W \in \mathbf{R}^{p\times r}$ with $W^T W = I$. The solution of (40) will be at one of those extreme points, and therefore (39) and (38) are equivalent.

The semidefinite program (39) is in the dual form (27). The dimensions are $n = 2p$ and $m = 1 + p(p+1)/2$. After a calculation we obtain that the corresponding primal problem is

$$\begin{array}{ll} \text{minimize} & rt + \mathbf{Tr} X \\ \text{subject to} & tI + X - A \geq 0 \\ & X \geq 0, \end{array}$$

which is precisely the expression used in (7).

3.2. Primal-dual problem formulation. In most semidefinite program problems that we encounter the hypothesis of Theorem 3.1 holds, so that $d^* = p^*$. (When the conditions do not hold it is possible, at least in principle, to reduce the original problem to one for which one of the conditions holds; see [17, §2.5]. See also the recent paper by Freund [34] for an interior-point method that does not require strict feasibility.)

Primal-dual methods for semidefinite programs, which we describe in detail in §5, generate a sequence of primal and dual feasible points $x^{(k)}$ and $Z^{(k)}$, where $k = 0, 1, \ldots$ denotes iteration number. We can interpret $x^{(k)}$ as a *suboptimal point* that gives the upper bound $p^* \leq c^T x^{(k)}$ and $Z^{(k)}$ as a *certificate* that proves the lower bound

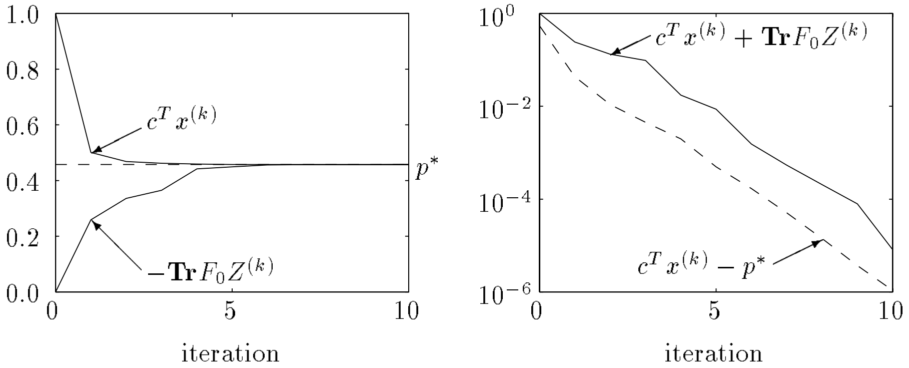

FIG. 4. *Convergence of a primal-dual algorithm. The problem is a matrix norm minimization problem (10 matrices in $\mathbf{R}^{10\times 10}$), and the algorithm is described in §5.3. The plot on the left shows how the primal and dual objectives converge to the optimal value. The solid curve in the right plot is the duality gap,* i.e., *the difference between the primal and dual objectives. The dashed line is the difference between the current (primal) objective and the optimal value. At the kth iteration, we know the value of the duality gap (*i.e., *the solid curve); we do not know the value of the actual error (*i.e., *the dashed curve).*

$p^* \geq -\mathbf{Tr} F_0 Z^{(k)}$. We can bound how suboptimal our current point $x^{(k)}$ is in terms of the duality gap $\eta^{(k)}$:

$$c^T x^{(k)} - p^* \leq \eta^{(k)} = c^T x^{(k)} + \mathbf{Tr} F_0 Z^{(k)}.$$

Therefore the stopping criterion

$$c^T x^{(k)} + \mathbf{Tr} F_0 Z^{(k)} \leq \epsilon$$

where $\epsilon > 0$ is some pre-specified tolerance, *guarantees* ϵ-suboptimality on exit. Indeed, the algorithm produces not only an ϵ-suboptimal point $\hat{x}$, but also a certificate (*i.e.*, a dual feasible $\hat{Z}$) that proves $\hat{x}$ is ϵ-suboptimal.

This idea is illustrated in the left plot of Figure 4, which shows the values of the primal and dual objectives of the iterates of a primal-dual algorithm as a function of iteration number. The optimal value is shown as the dashed line. The particular semidefinite program is a matrix norm minimization problem; the algorithm used will be explained later (in §5.3), but is not relevant here.

From this plot we can draw some interesting conclusions that illustrate some of the basic ideas behind primal-dual algorithms. After one iteration, we have found a point $x^{(1)}$ with objective value $c^T x^{(1)} = 0.5$. In fact this point is only 0.04-suboptimal, but we don't know this after one iteration: all we know is that the optimal value exceeds $-\mathbf{Tr} F_0 Z^{(1)} = 0.26$. After three iterations our primal point $x^{(3)}$, with objective value 0.46, is very nearly optimal, but we don't yet know it. All we can guarantee after three iterations is that the optimal value exceeds 0.36. Over the next few iterations our dual point steadily improves; by the *fifth* iteration we can now conclude that our first (primal) iterate was at most 10% suboptimal! This example illustrates the important distinction between converging to a given tolerance and *knowing* (*i.e.*, guaranteeing) convergence to a given tolerance. The possibility of terminating an optimization algorithm with a guaranteed accuracy of, say, 10%, is very useful in engineering applications.

The duality gap $\eta^{(k)}$ measures the width of our "uncertainty interval" for the optimal value p^* at iteration k. It is plotted at right in Figure 4 as the solid curve, along with the actual difference between the current (primal) objective value and the optimal value, shown as the dotted curve. The important point is that after k iterations we *know* the duality gap, which is an *upper bound* on $c^T x^{(k)} - p^*$, which of course we don't know.

Primal-dual methods can also be interpreted as solving the *primal-dual optimization problem*

$$\begin{array}{ll} \text{minimize} & c^T x + \mathbf{Tr} F_0 Z \\ \text{subject to} & F(x) \geq 0, \quad Z \geq 0 \\ & \mathbf{Tr} F_i Z = c_i, \quad i = 1, \ldots, m. \end{array} \tag{41}$$

Here we minimize the duality gap $c^T x + \mathbf{Tr} F_0 Z$ over all primal and dual feasible points; the optimal value is known in advance to be zero. The duality gap is a linear function of x and Z, and therefore problem (41) is a semidefinite program in x and Z.

At first glance there seems to be no benefit in considering the primal-dual optimization problem (41). Since the variables x and Z are independent (*i.e.*, the feasible set is the Cartesian product of the primal and dual feasible sets) and the objective in (41) is the sum of the primal and dual objectives, we can just as well solve the primal and dual problems separately. We shall see later that there is, however, a benefit: in each step we use dual information (*i.e.*, $Z^{(k)}$) to help find a good update for the primal variable $x^{(k)}$, and vice-versa.

4. The central path.

From now on we assume strict primal and dual feasibility, *i.e.*, we assume there exists an x with $F(x) > 0$, and a $Z = Z^T > 0$ with $\mathbf{Tr} F_i Z = c_i$, $i = 1, \ldots, m$. We will also assume that the matrices F_i, $i = 1, \ldots, m$ are linearly independent.

4.1. Barrier function for a linear matrix inequality.

The function

$$\phi(x) \stackrel{\Delta}{=} \begin{cases} \log\det F(x)^{-1} & \text{if } F(x) > 0 \\ +\infty & \text{otherwise} \end{cases} \tag{42}$$

is a *barrier function* for $\mathbf{X} \stackrel{\Delta}{=} \{x \mid F(x) > 0\}$, *i.e.*, $\phi(x)$ is finite if and only if $F(x) > 0$, and becomes infinite as x approaches the boundary of $\mathbf{X}$. There are many other barrier functions for $\mathbf{X}$ (for example, $\mathbf{Tr} F(x)^{-1}$ can be substituted for $\log\det F(x)^{-1}$ in (42)), but this one enjoys many special properties. In particular, when $F(x) > 0$, it is analytic, strictly convex, and self-concordant (see [76]).

Figure 5 shows the contour lines of the barrier function for the semidefinite pro-

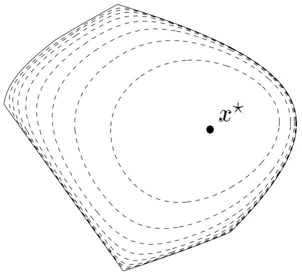

FIG. 5. *Contour lines of the barrier function (incremented in unit steps). $x^\star$ is the minimizer of the barrier function,* i.e., *the analytic center of the linear matrix inequality (see §4.2).*

gram of Figure 1. It illustrates that the function is quite flat in the interior of the feasible set and sharply increases towards the boundary.

The gradient $\nabla\phi(x)$ and the Hessian $\nabla^2\phi(x)$ of ϕ at x are given by

$$(\nabla\phi(x))_i = -\mathbf{Tr}F(x)^{-1}F_i = -\mathbf{Tr}F(x)^{-1/2}F_iF(x)^{-1/2}, \tag{43}$$

and

$$\begin{aligned}(\nabla^2\phi(x))_{ij} &= \mathbf{Tr}F(x)^{-1}F_iF(x)^{-1}F_j \\ &= \mathbf{Tr}\left(F(x)^{-1/2}F_iF(x)^{-1/2}\right)\left(F(x)^{-1/2}F_jF(x)^{-1/2}\right),\end{aligned} \tag{44}$$

for $i,j=1,\ldots,m$. Here $F(x)^{1/2}$ denotes the symmetric square root of $F(x)$; similar formulas that use Cholesky factors of $F(x)$ are also easily derived. Expressions (43) and (44) follow from the second order expansion of $\log\det X^{-1}$: If X and Y are symmetric with $X>0$, then for small Y

$$\begin{aligned}&\log\det(X+Y)^{-1} = \\ &\qquad \log\det X^{-1} - \mathbf{Tr}X^{-1}Y + \tfrac{1}{2}\mathbf{Tr}X^{-1}YX^{-1}Y + o\left(\|Y\|^2\right).\end{aligned} \tag{45}$$

For a set of linear inequalities $a_i^Tx+b_i\geq 0$, $i=1,\ldots,n$, the barrier function ϕ reduces to the familiar logarithmic barrier function used in interior-point linear programming:

$$\phi(x) = \begin{cases} -\displaystyle\sum_{i=1}^{n}\log(a_i^Tx+b_i) & \text{if } a_i^Tx+b_i>0,\ i=1,\ldots,n \\ +\infty & \text{otherwise.}\end{cases}$$

In this case we can interpret ϕ as the potential function associated with a repelling force from each constraint plane, directed away from the plane and inversely proportional to distance from the plane. To see this we simply note that

$$\nabla\phi(x) = \sum_{i=1}^{n} -\frac{1}{a_i^Tx+b_i}a_i = \sum_{i=1}^{n}\frac{-1}{r_i\|a_i\|}a_i$$

where r_i is the distance from x to the ith constraint plane. Thus the contribution from the ith constraint is a force pointing away from the ith constraint plane (*i.e.*, in the direction $-a_i/\|a_i\|$) with magnitude $1/r_i$.

4.2. Analytic center of a linear matrix inequality. In this section and in §4.3 we suppose that $\mathbf{X}$ is bounded. Since ϕ is strictly convex, it has a unique minimizer, which we denote

$$x^\star \overset{\Delta}{=} \text{argmin } \phi(x). \tag{46}$$

We will refer to $x^\star$ as the *analytic center* of the linear matrix inequality $F(x)\geq 0$.

It is important to note that the analytic center depends on the matrix inequality, and not the (strict) solution set $\mathbf{X}$: The same set $\mathbf{X}$ can be represented by different matrix inequalities, which have different analytic centers. Simply adding redundant constraints to the semidefinite program depicted in Figure 5, for example, will move $x^\star$ to another position.

From (43) we see that $x^\star$ is characterized by

$$\mathbf{Tr} F(x^\star)^{-1} F_i = 0, \quad i = 1, \ldots, m. \tag{47}$$

Thus, $F(x^\star)^{-1}$ is orthogonal to the span of $F_1, \ldots, F_m$. Note the similarity of the condition (47) and the equality constraints $\mathbf{Tr} F_i Z = c_i$, $i = 1, \ldots, m$ arising in the dual semidefinite program (27). We will soon see a close connection between analytic centers and dual feasibility.

In the case of a set of linear inequalities, the definition (46) coincides with Sonnevend's definition [102, 103], *i.e.*,

$$\begin{array}{rll} x^\star = & \text{argmax} & \prod_{i=1}^{n} (a_i^T x + b_i) \\ & \text{subject to} & a_i^T x + b_i \geq 0, \quad i = 1, \ldots, n. \end{array}$$

From this formula we see a simple geometric interpretation of the analytic center of a set of linear inequalities: $x^\star$ is the feasible point that maximizes the product of the distances to the constraint planes (*i.e.*, the planes defined by $a_i^T x + b_i = 0$). Of course we can also interpret the analytic center of a set of linear inequalities as the equilibrium point for the inverse-distance force field mentioned in §4.1.

4.3. Computing the analytic center. In this section we consider the problem of computing the analytic center of a linear matrix inequality. We do this for several reasons. First, the analysis we will encounter here will give us a good interpretation of a quantity that will play a key role in the algorithms we will later describe. And second, the algorithm described here foreshadows the primal-dual interior-point algorithms we will see later.

Newton's method with line search can be used to efficiently compute the analytic center, given an initial strictly feasible point, *i.e.*, x such that $F(x) > 0$. By definition, the Newton direction δx^N at x is the vector that minimizes the second-order expansion of $\phi(x+v) - \phi(x)$ over all $v \in \mathbf{R}^m$. From (45) we obtain (with $F \stackrel{\Delta}{=} F(x)$)

$$\begin{aligned} \delta x^N &= \underset{v \in \mathbf{R}^m}{\text{argmin}} \left(-\mathbf{Tr} F^{-1} \left(\sum_{i=1}^m v_i F_i \right) \right. \\ &\qquad \left. + \tfrac{1}{2} \mathbf{Tr} \left(\sum_{i=1}^m v_i F_i \right) F^{-1} \left(\sum_{j=1}^m v_j F_j \right) F^{-1} \right) \\ &= \underset{v \in \mathbf{R}^m}{\text{argmin}} \left\| -I + \sum_{i=1}^m v_i F^{-1/2} F_i F^{-1/2} \right\|_F . \end{aligned} \tag{48}$$

The norm used in equation (48) is the Frobenius-norm, *i.e.*, $\|A\|_F = (\mathbf{Tr} A^T A)^{1/2} = \left(\sum_{ij} A_{ij}^2 \right)^{1/2}$. Thus, the Newton direction δx^N is found by solving (48), which is a least-squares problem with m variables and $n(n+1)/2$ equations.

A line search is used to determine the length of the step to be made in the direction δx^N. We compute a step-length that (approximately) minimizes $\phi(x + p\delta x^N)$ over all $p \in \mathbf{R}$, which can be done efficiently using standard methods such as bisection. A simple precomputation makes the line search quite efficient. With $\delta F \stackrel{\Delta}{=} \sum_{i=1}^m \delta x_i^N F_i$, we have

$$\phi(x + p\delta x^N) = \phi(x) - \log\det\left(I + pF^{-1/2}\delta F F^{-1/2}\right) = \phi(x) - \sum_{i=1}^{n} \log(1 + p\mu_i),$$

where μ_i are the eigenvalues of $F^{-1/2}\delta F F^{-1/2}$. The last expression shows that once we have computed the eigenvalues μ_i, the derivatives of $\phi(x + p\delta x^N)$ can be computed in $O(n)$ operations. This idea will come up again in §5.5.

The algorithm is thus:

```
Newton method for computing the analytic center
given strictly feasible $x$.
repeat
   1. Compute the Newton direction $\delta x^N$ by solving the least-squares problem (48).
   2. Find $\hat{p} = \mathrm{argmin}\ \phi(x + p\delta x^N)$.
   3. Update: $x := x + \hat{p}\delta x^N$.
```

Of course it is well known that the asymptotic convergence will be quadratic. Nesterov and Nemirovsky in [76, §2.2] give a complete analysis of the *global* speed of convergence of this algorithm:

THEOREM 4.1. *Let $x^{(k)}$ denote the value of x in the algorithm above after the kth iteration, and assume $0 < \epsilon \leq 0.5$. For*

$$k \geq 11(\phi(x^{(0)}) - \phi(x^\star)) + \log_2\log_2(1/\epsilon) \tag{49}$$

we have $\phi(x^{(k)}) - \phi(x^\star) \leq \epsilon$.

Note that the right-hand side of (49) does not depend on the problem size (*i.e.*, m or n) at all, and only depends on the problem data (*i.e.*, $F_0, \ldots, F_m$) through the difference between the value of the barrier function at the initial point and the analytic center. For all practical purposes the term $\log_2\log_2(1/\epsilon)$ can be considered a constant, say, five (which guarantees an accuracy of $\epsilon = 2^{-32}$).

We should mention two points. First, Theorem 4.1 holds for an 'implementable' version of the algorithm as well, in which an appropriate *approximate* line search is used instead of the exact line search. Second, Nesterov and Nemirovsky give an explicit, readily computable stopping criterion that guarantees $\phi(x^{(k)}) - \phi(x^\star) \leq \epsilon$.

4.4. The central path: Objective parametrization. Let us return to the primal semidefinite program (1). Consider the linear matrix inequality

$$\begin{array}{c} F(x) > 0 \\ c^T x = \gamma \end{array} \tag{50}$$

where $p^* < \gamma < \overline{p} \overset{\Delta}{=} \sup\{c^T x \mid F(x) > 0\}$. It can be shown that the solution set to (50) is nonempty and bounded under our assumption that the semidefinite program (1) is strictly primal and dual feasible. Therefore, the analytic center of (50), defined as

$$x^\star(\gamma) \overset{\Delta}{=} \begin{array}{ll} \text{argmin} & \log\det F(x)^{-1} \\ \text{subject to} & F(x) > 0 \\ & c^T x = \gamma \end{array} \tag{51}$$

exists for $p^* < \gamma < \overline{p}$. The curve described by $x^\star(\gamma)$ is called the *central path* for the semidefinite program (1). The central path passes through the analytic center of the constraint $F(x) \geq 0$; as γ approaches p^* from above, the central point $x^\star(\gamma)$ converges to an optimal point; as γ approaches $\overline{p}$ from below, it converges to a maximizer of c^Tx subject to $F(x) \geq 0$. This is illustrated in Figure 6, which shows the central path for the semidefinite program of Figure 1.

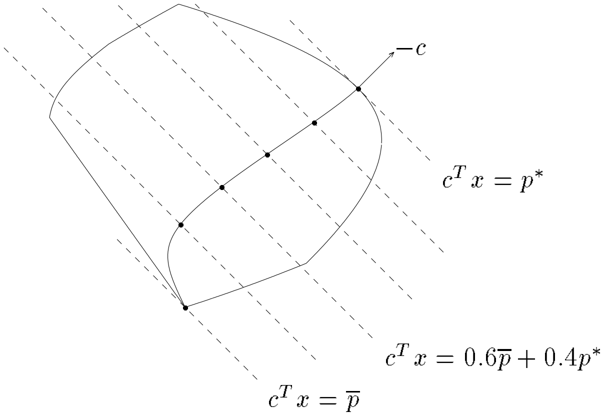

FIG. 6. *The central path for the semidefinite program of Figure 1. The dashed lines represent level sets $c^Tx = \gamma$, for six values γ in the interval $[p^*, \overline{p}]$. The heavy dots are the analytic centers $x^\star(\gamma)$ of the linear matrix inequality (50). The central path is formed by the analytic centers $x^\star(\gamma)$ when γ varies between p^* and $\overline{p}$.*

Writing out the optimality conditions for (51), we find that $x^\star(\gamma)$ satisfies

$$\mathbf{Tr}\, F(x^\star(\gamma))^{-1}F_i = \lambda c_i, \quad i = 1, \ldots, m, \tag{52}$$

where λ is a Lagrange multiplier. It can be shown that λ is positive on the part of the central path between the analytic center and the optimal point the path of centers converges to. From (52) we see that the matrix $F(x^\star(\gamma))^{-1}/\lambda$ is dual feasible when $\lambda > 0$. Thus, points on the primal central path yield dual feasible matrices.

The duality gap associated with the primal-dual feasible pair $x = x^\star(\gamma)$, $Z = F(x^\star(\gamma))^{-1}/\lambda$ is

$$\eta = \mathbf{Tr}\, F(x)Z = \mathbf{Tr}\, F(x^\star(\gamma))F(x^\star(\gamma))^{-1}/\lambda = n/\lambda.$$

Thus the Lagrange multiplier λ appearing in (52) is simply related to the duality gap of the point on the path of centers and the associated dual feasible point.

In fact, the matrix $F(x^\star(\gamma))^{-1}/\lambda$ is not only dual feasible, but is itself on the central path for the dual semidefinite program, *i.e.*, it solves

$$\begin{array}{ll} \text{minimize} & \log\det Z^{-1} \\ \text{subject to} & \mathbf{Tr}\, F_iZ = c_i, \quad i = 1, \ldots, m \\ & Z > 0 \\ & -\mathbf{Tr}\, F_0Z = \gamma - n/\lambda. \end{array}$$

In other words, among all dual feasible Z with dual objective value $\gamma - n/\lambda$, the matrix $F(x^\star(\gamma))^{-1}/\lambda$ minimizes the barrier function $\log\det Z^{-1}$. Thus we have a natural pairing between points on the primal central path and points on the dual central path; moreover for these primal-dual central pairs x, Z, the matrices $F(x)$ and Z are inverses of each other, up to a scale factor.

Almost all interior-point methods approach the optimal point by following the central path, either literally, by returning to the central path periodically, or by keeping some measure for the deviation from the central path below a certain bound. The most natural measure has already been hinted at in §4.3. For every strictly feasible x, we define the deviation from the central path $\psi(x)$ as

$$\psi(x) \overset{\Delta}{=} \log\det F(x)^{-1} - \log\det F(x^\star(c^T x))^{-1}.$$

$\psi(x)$ is the difference between the value of the barrier function at the point x and the minimum of the barrier function over all points with the same value of cost function as x. Figure 7 shows the contour lines of $\psi(x)$ for our example semidefinite program. The central path, on which $\psi(x) = 0$, is shown as a solid curve.

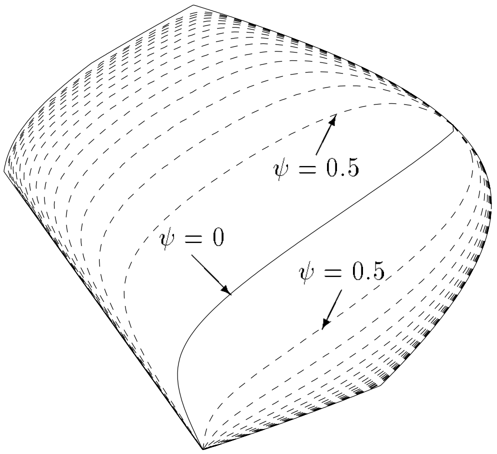

FIG. 7. *Contour lines of the deviation from centrality $\psi(x)$, in increments of 0.5. The solid line is the central path, on which $\psi(x)$ is zero.*

From §4.3, we have the following interpretation. The deviation from centrality $\psi(x)$ bounds above the number of Newton steps needed to compute the point $x^\star(c^T x)$, starting at x (to an accuracy exceeding 2^{-32}):

$$\begin{array}{rcl}\#\text{Newton steps} & \leq & 5 + 11(\log\det F(x)^{-1} - \log\det F(x^\star(c^T x))^{-1}),\\ & = & 5 + 11\psi(x).\end{array}$$

Thus, $\psi(x)$ bounds the effort required to center the point x. On Figure 7 the two curves on which $\psi = 0.5$ define a wide region surrounding the central path. For any point in this region, no more than eleven Newton steps are required to compute a central point with the same objective value.

4.5. The central path: Duality gap parametrization. In the section above we parametrized the central path by the primal objective value γ. We found that the dual central path is also indirectly parametrized by γ as $F(x^\star(\gamma))^{-1}/\lambda$. It turns out that both central paths can be very conveniently parametrized by the duality gap. This will be more convenient when describing primal-dual algorithms.

The primal-dual parametrization of the central path $(x^\star(\eta), Z^\star(\eta))$ is defined by

$$(x^\star(\eta), Z^\star(\eta)) \overset{\Delta}{=} \begin{array}{ll} \text{argmin} & -\log\det F(x) - \log\det Z \\ \text{subject to} & F(x) > 0,\ Z > 0 \\ & \mathbf{Tr} F_i Z = c_i, i = 1, \ldots, m \\ & c^T x + \mathbf{Tr} F_0 Z = \eta \end{array} \tag{53}$$

for $\eta \geq 0$. Thus, among all feasible pairs x, Z with the duality gap η, the pair $(x^\star(\eta), Z^\star(\eta))$ minimizes the primal-dual barrier function $\log\det F(x)^{-1} + \log\det Z^{-1}$.

It can be shown that the pair $(x^\star(\eta), Z^\star(\eta))$ is characterized by

$$\begin{array}{l} F(x^\star(\eta)) \geq 0 \\ Z^\star(\eta) \geq 0, \ \ \mathbf{Tr} F_i Z^\star(\eta) = c_i, \ \ i = 1, \ldots, m \\ Z^\star(\eta) F(x^\star(\eta)) = (\eta/n) I. \end{array} \tag{54}$$

Comparing this to (33), we can interpret the central path as defining a homotopy, with the duality gap η as homotopy parameter. The homotopy perturbs the optimality condition $ZF(x) = 0$ to the condition $ZF(x) = (\eta/n)I$. The pair $(x^\star(\eta), Z^\star(\eta))$ converges to a primal and dual optimal pair as $\eta \to 0$. This interpretation is well-known for LP.

Now consider a feasible pair (x, Z), and define $\eta = c^T x + \mathbf{Tr} F_0 Z = \mathbf{Tr} F(x) Z$. Then $(x^\star(\eta), Z^\star(\eta))$ is the central pair with the same duality gap as x, Z. Therefore

$$-\log\det F(x^\star(\eta)) Z^\star(\eta) = -n \log(\eta/n) = n \log n - n \log \mathbf{Tr} F(x) Z.$$

As in §4.4 we can say that the difference

$$\begin{aligned} \psi(x, Z) &\overset{\Delta}{=} -\log\det F(x) Z + \log\det F(x^\star(\eta)) Z^\star(\eta) \\ &= -\log\det F(x) Z + n \log \mathbf{Tr} F(x) Z - n \log n \end{aligned}$$

is a measure of the deviation of (x, Z) from centrality: $\psi(x, Z)$ is, up to a constant, an upper bound on the computational effort required to 'center' (x, Z) (meaning, compute the central pair with the same duality gap). Of course $\psi(x, Z)$ is nonnegative for all primal and dual feasible x, Z; it is zero only when x and Z are central, *i.e.*, $F(x)$ and Z are inverses of each other, up to a scale factor.

It is interesting to note that the deviation from centrality, $\psi(x, Z)$, can be evaluated for any primal feasible x and dual feasible Z, without computing the central pair with the same duality gap, *i.e.*, $(x^\star(\eta), Z^\star(\eta))$ where $\eta = \mathbf{Tr} F(x) Z$.

The function ψ is not convex or quasiconvex (except of course when restricted to $\mathbf{Tr} F(x) Z$ constant). We also note that ψ depends only on the eigenvalues $\lambda_1, \ldots, \lambda_n$ of $F(x) Z$:

$$\psi(x, Z) = n \log \frac{\left(\sum_{i=1}^n \lambda_i\right)/n}{\left(\prod_{i=1}^n \lambda_i\right)^{1/n}}.$$

Thus $\psi(x, Z)$ is n times the logarithm of the ratio of the arithmetic to the geometric mean of the eigenvalues of $F(x) Z$. (From which we see again that ψ is nonnegative, and zero only when $F(x) Z$ is a multiple of the identity.) We can also think of ψ as a smooth measure of condition number of the matrix $F(x) Z$ since

$$\log \kappa - 2 \log 2 \leq \psi(x, Z) \leq (n - 1) \log \kappa$$

where $\kappa = \lambda_{\max}/\lambda_{\min}$ is the condition number of $F(x)Z$ (see also [11, p.576]).

We should mention that several other measures of deviation from centrality have been used, especially in the linear programming literature, for analyzing and constructing interior-point algorithms. One possible choice is $\|\Lambda - (\eta/n)I\|_F$ where $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_n)$. An advantage of the measure ψ is the simplicity of the (global) analysis of algorithms based on it.

5. Primal-dual potential reduction methods.

5.1. General description. Potential reduction methods are based on the potential function

$$\begin{aligned}(55)\ \varphi(x,Z) &\overset{\Delta}{=} \nu\sqrt{n}\log\left(\mathbf{Tr}F(x)Z\right) + \psi(x,Z),\\ &= (n+\nu\sqrt{n})\log\left(\mathbf{Tr}F(x)Z\right) - \log\det F(x) - \log\det Z - n\log n,\end{aligned}$$

which combines the duality gap of the pair x, Z with the deviation from centrality of the pair x, Z. The constant $\nu \geq 1$ is a parameter which sets the relative weight of the term involving duality gap and the term which is the deviation from centrality.

Since $\psi(x,Z) \geq 0$ for all primal and dual feasible x, Z, we have

$$\eta \leq \exp\left(\varphi/(\nu\sqrt{n})\right).$$

Therefore, if the potential function is small, the duality gap must be small.

Potential reduction methods start at a pair of strictly feasible points $x^{(0)}$, $Z^{(0)}$, and reduce φ by at least a fixed amount in every step:

$$\varphi(x^{(k+1)}, Z^{(k+1)}) \leq \varphi(x^{(k)}, Z^{(k)}) - \delta, \tag{56}$$

where δ is an absolute constant. As a consequence, the iterates remain feasible, and converge to the optimum. Moreover, the convergence is polynomial, in a sense that is made precise in the following theorem.

THEOREM 5.1. *Assume that (56) holds with some $\delta > 0$ that does not depend on n or ϵ, where $0 < \epsilon < 1$. Then for*

$$k \geq \frac{\nu\sqrt{n}\log(1/\epsilon) + \psi(x^{(0)}, Z^{(0)})}{\delta}$$

we have $\mathbf{Tr}F(x^{(k)})Z^{(k)} < \epsilon\mathbf{Tr}F(x^{(0)})Z^{(0)}$.

Roughly speaking, we have convergence in $O(\sqrt{n})$ steps, provided the initial pair is sufficiently centered.

A general outline of a potential reduction method is as follows.

Potential reduction algorithm

```
given strictly feasible $x$ and $Z$.
repeat
    1. Find a suitable direction $\delta x$, and a suitable dual feasible direction $\delta Z$.
    2. Find $p$, $q \in \mathbf{R}$ that minimize $\varphi(x + p\delta x, Z + q\delta Z)$.
    3. Update: $x := x + p\delta x$ and $Z := Z + q\delta Z$.
until duality gap $\leq \epsilon$.
```

By dual feasible direction, we mean a $\delta Z = \delta Z^T$ that satisfies $\mathbf{Tr} F_i \delta Z = 0$, $i = 1, \ldots, m$, so that $Z + q\delta Z$ satisfies the dual equality constraints for any $q \in \mathbf{R}$.

We refer to the second step as the *plane search* since we are minimizing the potential function over the plane defined by the (current) points x, Z and the (current) search directions δx, δZ. We will see in §5.5 that the plane search can be carried out very efficiently.

There are several possibilities for generating suitable descent directions δx and δZ; each choice leads to a different algorithm. The basic computations are all very similar, however. The search directions δx and δZ are obtained from a set of linear equations of the form

$$\begin{aligned} S\delta Z S + \sum_{i=1}^{m} \delta x_i F_i &= -D \\ \mathbf{Tr} F_j \delta Z = 0, \;\; j &= 1, \ldots, m. \end{aligned} \tag{57}$$

The matrices $D = D^T$ and $S = S^T > 0$ depend on the particular algorithm, and change in every iteration. Problem (57) is a set of $m + n(n+1)/2$ equations in $m + n(n+1)/2$ variables. If the linear matrix inequality $F(x)$ is block-diagonal, with L blocks of size n_i, $i = 1, \ldots, L$, then we only have $m + \sum_{i=1}^{L} n_i(n_i+1)/2$ equations and variables.

Equations (57) arise as the optimality conditions of two quadratic minimization problems:

$$\begin{aligned} \delta x \;=\; & \underset{v \in \mathbf{R}^m}{\text{argmin}} \left(\mathbf{Tr} D S^{-1} \left(\sum_{i=1}^{m} v_i F_i \right) S^{-1} \right. \\ & \qquad \left. + \tfrac{1}{2} \mathbf{Tr} \left(\sum_{i=1}^{m} v_i F_i \right) S^{-1} \left(\sum_{j=1}^{m} v_j F_j \right) S^{-1} \right) \end{aligned} \tag{58}$$

$$\begin{aligned} \delta Z \;=\; & \text{argmin} && \mathbf{Tr} D V + \tfrac{1}{2} \mathbf{Tr} V S V S \\ & \text{subject to} && V = V^T \\ & && \mathbf{Tr} F_i V = 0, \; i = 1, \ldots, m. \end{aligned} \tag{59}$$

Problem (57) can be solved in several ways depending on how much structure in the matrices F_i one wishes to exploit. We will briefly discuss different possibilities in Section 7.6. If the matrices F_i are dense or unstructured, then (57) can be solved efficiently via a least-squares problem

$$\delta x = \underset{v \in \mathbf{R}^m}{\text{argmin}} \left\| S^{-1/2} \left(D + \sum_{i=1}^{m} v_i F_i \right) S^{-1/2} \right\|_F . \tag{60}$$

This can be shown by eliminating δZ from the two equations in (57). After a simplification we obtain

$$\begin{aligned} \sum_{i=1}^{m} \delta x_i \mathbf{Tr} \left(S^{-1/2} F_j S^{-1/2} \right) \left(S^{-1/2} F_i S^{-1/2} \right) = & \\ - \mathbf{Tr} \left(S^{-1/2} F_j S^{-1/2} \right) \left(S^{-1/2} D S^{-1/2} \right), & \end{aligned} \tag{61}$$

for $j = 1, \ldots, m$. These equations are precisely the normal equations for (60). Once δx is known from (60), the matrix δZ follows from the first equation in (57).

Let us consider the LP (2) as an example, *i.e.*, assume $F(x) = \mathbf{diag}(Ax+b)$. In this case, all matrices in (57) are diagonal. If we write $D = \mathbf{diag}(d)$, and $\delta Z = \mathbf{diag}(\delta z)$, then (57) reduces to

$$\begin{bmatrix} S^2 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \delta z \\ \delta x \end{bmatrix} = \begin{bmatrix} -d \\ 0 \end{bmatrix}. \tag{62}$$

5.2. Potential reduction method 1. An obvious way to compute search directions δx and δZ is to apply Newton's method to φ. The potential φ is not a convex function, however: The first term, $(n+\nu\sqrt{n})\log(c^Tx+\mathbf{Tr}F_0Z)$, is concave in x and Z, and hence contributes a negative semidefinite term to the Hessian of φ. One simple modification of Newton's method is to ignore the second derivative of this concave term.

Assume the current iterates are x, Z, and set $F \stackrel{\Delta}{=} F(x)$ for simplicity. As in Newton's method, we choose directions δx and δZ that minimize a quadratic approximation of $\varphi(x+v, Z+V)$ over all $v \in \mathbf{R}^m$ and all $V = V^T$, $\mathbf{Tr}F_iV = 0$, $i = 1, \ldots, m$.

The primal direction δx^p is computed as

$$\begin{aligned} \delta x^p &= \underset{v \in \mathbf{R}^m}{\text{argmin}} \left(\rho c^T v - \mathbf{Tr}F^{-1}\left(\sum_{i=1}^m v_iF_i\right) \right. \\ &\qquad \left. + \tfrac{1}{2}\mathbf{Tr}\left(\sum_{i=1}^m v_iF_i\right)F^{-1}\left(\sum_{j=1}^m v_jF_j\right)F^{-1}\right) \\ &= \underset{v \in \mathbf{R}^m}{\text{argmin}} \left(\mathbf{Tr}\left(\rho Z - F^{-1}\right)\left(\sum_{i=1}^m v_iF_i\right) \right. \\ &\qquad \left. + \tfrac{1}{2}\mathbf{Tr}\left(\sum_{i=1}^m v_iF_i\right)F^{-1}\left(\sum_{j=1}^m v_jF_j\right)F^{-1}\right) \end{aligned} \tag{63}$$

with $\rho \stackrel{\Delta}{=} (n+\nu\sqrt{n}) \big/ \left(c^Tx+\mathbf{Tr}F_0Z\right)$.

The quadratic function in (63) is the second order expansion of $\log\det F(x+v)^{-1}$ plus a linear approximation of the concave term $(n+\nu\sqrt{n})\log(c^T(x+v)+\mathbf{Tr}F_0Z)$. Thus, δx^p is the minimizer of a local quadratic approximation to $\varphi(x, Z)$. It is not the exact Newton direction, however, because the second derivative of $\log(c^Tx+\mathbf{Tr}F_0Z)$ is ignored.

Note that (63) is of the form (58) with $D = \rho FZF - F$ and $S = F$. Applying (57), we see that δx^p can be computed from

$$\begin{gathered} F\delta Z^pF + \sum_{i=1}^m \delta x_i^p F_i = -\rho FZF + F \\ \mathbf{Tr}F_j\delta Z^p = 0, \quad j = 1, \ldots, m. \end{gathered} \tag{64}$$

In a similar way, δZ^d is computed as the minimizer of the second-order approximation of $\log\det(Z+V)^{-1}$ plus a linear approximation of $(n+\nu\sqrt{n})\log(c^Tx+\mathbf{Tr}F_0(Z+V))$:

$$\begin{array}{rcll} \delta Z^d & = & \text{argmin} & \rho\mathbf{Tr}F_0V - \mathbf{Tr}Z^{-1}V + \frac{1}{2}\mathbf{Tr}Z^{-1}VZ^{-1}V \\ & & \text{subject to} & V = V^T \\ & & & \mathbf{Tr}F_iV = 0,\ i = 1,\ldots,m \\ & = & \text{argmin} & \rho\mathbf{Tr}FV - \mathbf{Tr}Z^{-1}V + \frac{1}{2}\mathbf{Tr}Z^{-1}VZ^{-1}V \\ & & \text{subject to} & V = V^T \\ & & & \mathbf{Tr}F_iV = 0,\ i = 1,\ldots,m. \end{array} \tag{65}$$

The second formulation follows because $\mathbf{Tr}F_0V = \mathbf{Tr}FV$ if $\mathbf{Tr}F_iV = 0$, $i = 1,\ldots,m$. Problem (65) is of the form (59) with $S = Z^{-1}$ and $D = \rho F - Z^{-1}$. From (57), we see that δZ^d can be computed from

$$\begin{gathered} Z^{-1}\delta Z^d Z^{-1} + \sum_{i=1}^m \delta x_i^d F_i = -\rho F + Z^{-1} \\ \mathbf{Tr}F_j\delta Z^d = 0,\quad j = 1,\ldots,m. \end{gathered} \tag{66}$$

The first potential reduction method follows the general outline given in §5.1, with the pair δx^p, δZ^d as search directions. Using these directions, it is always possible to reduce φ by at least a fixed amount.

THEOREM 5.2. *Let $x^{(k)}$ and $Z^{(k)}$ denote the values of x and Z after the kth iteration of the potential reduction algorithm with search directions δx^p, δZ^d. We have*

$$\varphi(x^{(k+1)}, Z^{(k+1)}) \le \varphi(x^{(k)}, Z^{(k)}) - 0.78.$$

From Theorem 5.1 it follows that the algorithm has a polynomial worst-case complexity. For a proof of Theorem 5.2 see Vandenberghe and Boyd [108]. The method is a generalization of Gonzaga and Todd's method for linear programming [43]. We should mention that the theorem holds for an implementable version of the algorithm, in which an appropriate approximate plane search is used to determine the step lengths.

Let us consider the LP (2) as an illustration. The linear systems (64) and (66) reduce to

$$\begin{bmatrix} F^2 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \delta z^p \\ \delta x^p \end{bmatrix} = \begin{bmatrix} -\rho F^2 z + Fe \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} Z^{-2} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \delta z^d \\ \delta x^d \end{bmatrix} = \begin{bmatrix} -\rho Fe + Z^{-1}e \\ 0 \end{bmatrix}.$$

5.3. Potential reduction method 2. The algorithm of the previous section has the disadvantage of requiring the solution of two systems, (64) and (66), per iteration. It turns out that a complete primal-dual algorithm can be based on the primal system only, by choosing δZ^p as dual search direction. In linear programming this primal-dual method is due to Ye [114]; the extension to semidefinite programs is due to Nesterov and Nemirovsky [76] and Alizadeh [1].

Again it is possible to reduce φ by at least a fixed amount.

THEOREM 5.3. *Let $x^{(k)}$ and $Z^{(k)}$ denote the values of x and Z after the kth iteration of the potential reduction algorithm with search directions δx^p, δZ^p. We have*

$$\varphi(x^{(k+1)}, Z^{(k+1)}) \leq \varphi(x^{(k)}, Z^{(k)}) - 0.05.$$

Several comments are in order. First, the value of the guaranteed reduction in potential per iteration, *i.e.*, 0.05, has no practical significance. Although this bound is more than 25 times smaller than the bound given in Theorem 5.2, this second potential reduction method seems to perform better in practice than the first one. Second, the theorem holds for an implementable version of the algorithm, in which an appropriate approximate plane search is used to determine the step lengths. A slight variation of Theorem 5.3 is proved by Nesterov and Nemirovsky [76, §4.5.3].

These considerations can be repeated for the dual problem (66). A complete primal-dual algorithm can be based on δx^d and δZ^d. We will call this method potential reduction method $2^\star$. Polynomial complexity follows from Theorem 5.3 by duality.

THEOREM $5.3^\star$. *Let $x^{(k)}$ and $Z^{(k)}$ denote the values of x and Z after the kth iteration of the potential reduction algorithm with search directions δx^d, δZ^d. We have*

$$\varphi(x^{(k+1)}, Z^{(k+1)}) \leq \varphi(x^{(k)}, Z^{(k)}) - 0.05.$$

5.4. Potential reduction method 3. The first potential method treats the primal and dual semidefinite program symmetrically, but requires solving two linear systems per iteration, one for δx^p and one for δZ^d. The second method is not symmetrical (we had a primal and a dual variant) but computes primal and dual directions from a single linear system which is a great advantage in practice.

Nesterov and Todd have recently proposed another variation, which preserves the primal-dual symmetry, yet avoids solving two systems per iteration. In their method primal and dual search directions are computed from

$$\begin{gathered} RR^T \delta Z^{\mathrm{sym}} RR^T + \sum_{i=1}^m \delta x_i^{\mathrm{sym}} F_i = -\rho F + Z^{-1} \\ \mathbf{Tr} F_j \delta Z^{\mathrm{sym}} = 0, \;\; j = 1, \ldots, m. \end{gathered}$$

The matrix R satisfies

$$R^T F^{-1} R = \Lambda^{-1/2} \text{ and } R^T Z R = \Lambda^{1/2},$$

and can be constructed as $R = F^{1/2} U \Lambda^{-1/4}$, where $F^{1/2} Z F^{1/2} = U \Lambda U^T$ is the eigenvalue decomposition of $F^{1/2} Z F^{1/2}$. If F and Z are a central pair, *i.e.*, if $F^{1/2} Z F^{1/2} = (\eta/n) I$, then Λ is a multiple of the identity, $\Lambda = (\eta/n) I$.

Nesterov and Todd [77] have shown that the worst-case complexity of this algorithm is polynomial. They prove the following theorem.

THEOREM 5.4. *Let $x^{(k)}$ and $Z^{(k)}$ denote the values of x and Z after the kth iteration of the potential reduction algorithm with search directions δx^{sym}, δZ^{sym}. We have*

$$\varphi(x^{(k+1)}, Z^{(k+1)}) \leq \varphi(x^{(k)}, Z^{(k)}) - 0.24.$$

Once again, the theorem holds for an algorithm that uses an appropriate approximate plane search. In the case of an LP, with $F = \mathbf{diag}(Ax+b)$ and $Z = \mathbf{diag}(z)$, this symmetric scaling coincides with the primal-dual symmetric scaling used in Kojima, Mizuno and Yoshise [60], for example, where search directions are computed from

$$\begin{bmatrix} FZ^{-1} & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \delta z^{\mathrm{sym}} \\ \delta x^{\mathrm{sym}} \end{bmatrix} = \begin{bmatrix} -\rho F e + Z^{-1} e \\ 0 \end{bmatrix}. \tag{67}$$

The three algorithms we discussed so far differ only in the scaling matrices S used in (57). In linear programming, the equivalent of method 3 is usually preferred, since it is more efficient and has better numerical properties (see, *e.g.*, Wright [113]).

We should however mention two other possibilities that generalize (67). Alizadeh, Haeberly, and Overton [5] have pointed out the potential numerical difficulties in (64) and (66) and proposed to compute δx and δZ from

$$\begin{gathered} F\delta Z + \delta Z F + Z\left(\sum_{i=1}^m \delta x_i F_i\right) + \left(\sum_{i=1}^m \delta x_i F_i\right) Z = -\rho(FZ + ZF) + 2I \\ \mathbf{Tr} F_j \delta Z = 0, \quad j = 1, \ldots, m. \end{gathered}$$

Helmberg, Rendl, Vanderbei, and Wolkowicz [46], and Kojima, Shindoh, and Hara [61] have proposed to solve

$$\begin{gathered} F\delta Z Z^{-1} + \sum_{i=1}^m \delta x_i F_i = -\rho F + Z^{-1} \\ \mathbf{Tr} F_j \delta Z = 0, \quad j = 1, \ldots, m, \end{gathered}$$

and to replace the resulting, nonsymmetric, matrix δZ by its symmetric part.

5.5. Plane search. Once we have selected primal and dual directions δx and δZ, the problem is reduced to a two-dimensional problem, *i.e.*, the selection of lengths of the steps made in the directions δx and δZ. In this section we show that the computational effort of this plane search can be greatly reduced by first diagonalizing the matrices involved. The cost of this diagonalization and subsequent plane search is usually small compared to the cost of computing the search directions themselves, so in these cases the plane search accounts for a small, often negligible, fraction of the total computational effort.

In the plane defined by the directions δx and δZ, the potential function can be written as

$$\begin{aligned} \varphi(x + p\delta x, Z + q\delta Z) &= \varphi(x, Z) + (n + \nu\sqrt{n}) \log(1 + c_1 p + c_2 q) \\ &\quad - \log\det(I + pF^{-1/2}\delta F F^{-1/2}) - \log\det(I + qZ^{-1/2}\delta Z Z^{-1/2}) \end{aligned} \tag{68}$$

where $F \overset{\Delta}{=} F(x)$, $\delta F \overset{\Delta}{=} \sum_{i=1}^m \delta x_i F_i$, and

$$c_1 = \frac{c^T \delta x}{\mathbf{Tr} F(x) Z}, \quad c_2 = \frac{\mathbf{Tr} F_0 \delta Z}{\mathbf{Tr} F(x) Z}.$$

Equation (68) can be expressed in terms of the eigenvalues $\mu_1, \ldots, \mu_n$ of the matrix $F^{-1/2}\delta F F^{-1/2}$ and the eigenvalues $\nu_1, \ldots, \nu_n$ of $Z^{-1/2}\delta Z Z^{-1/2}$ (*i.e.*, the generalized eigenvalues of the matrix pairs $(\delta F, F)$ and $(\delta Z, Z)$):

$$\begin{array}{l}\varphi(x + p\delta x, Z + q\delta Z) = \varphi(x, Z) + \\ \qquad (n + \nu\sqrt{n}) \log(1 + c_1 p + c_2 q) - \sum_{i=1}^{n} \log(1 + p\mu_i) - \sum_{i=1}^{n} \log(1 + q\nu_i).\end{array}$$

The set of feasible p, q is the rectangle defined by $p_{\min} \leq p \leq p_{\max}$, $q_{\min} \leq q \leq q_{\max}$, where

$$\begin{array}{rcl} p_{\min} & = & \max\{-1/\mu_i \mid \mu_i > 0\}, \\ p_{\max} & = & \min\{-1/\mu_i \mid \mu_i < 0\}, \\ q_{\min} & = & \max\{-1/\nu_i \mid \nu_i > 0\}, \\ q_{\max} & = & \min\{-1/\nu_i \mid \nu_i < 0\}. \end{array}$$

Thus, once we have computed the eigenvalues μ_i, ν_i and the constants c_1, c_2, the plane search problem becomes:

$$\begin{array}{ll} \text{minimize} & (n + \nu\sqrt{n}) \log(1 + c_1 p + c_2 q) - \sum_{i=1}^{n} \log(1 + p\mu_i) - \sum_{i=1}^{n} \log(1 + q\nu_i) \\ \text{subject to} & p_{\min} \leq p \leq p_{\max}, \quad q_{\min} \leq q \leq q_{\max} \end{array} \tag{69}$$

It can be shown (see, *e.g.*, [49]) that the objective, *i.e.*, $\varphi(x + p\delta x, Z + q\delta Z)$, is a *quasi-convex* function of p and q; in particular it has a unique local minimum in the feasible rectangle which is the global minimum. Therefore the problem (69) can be solved using standard methods, *e.g.*, a guarded Newton method. Note that the objective and its derivatives with respect to p and q can be computed in $O(n)$ operations.

We can also mention that once we have computed the constants c_1, c_2, $p_{\min}$, $p_{\max}$, $q_{\min}$, and $q_{\max}$, it is trivial to minimize the duality gap over the feasible plane. The solution of course lies at one of the corners of the rectangle. The value of the gap at this corner will be smaller than the value corresponding to the solution at the minimizer of the potential function over the rectangle, *i.e.*, at the next iterates of the primal-dual algorithm. It is possible to terminate the entire algorithm at this point if the gap at the minimum-gap corner is smaller than the required tolerance.

An example of a plane search is illustrated in Figure 8, which shows the contour lines of φ in the p, q plane. Note that its sublevel sets are convex, which is a consequence of the quasiconvexity of $\varphi(x + p\delta x, Z + q\delta Z)$.

We should mention one more point about general plane searches. Instead of diagonalizing the matrices $F^{-1/2}\delta F F^{-1/2}$ and $Z^{-1/2}\delta Z Z^{-1/2}$, we can instead tridiagonalize them. With this preprocessing, we can still compute the derivatives for the reduced two-dimensional problem in $O(n)$ operations. In practice diagonalization and tridiagonalization do not differ too much since the bulk of the effort of computing the eigenvalues is the initial tridiagonalization. In a careful complexity analysis, tridiagonalization has the advantage of requiring only a finite number of exact arithmetic steps.

5.6. Numerical examples. We consider the matrix norm minimization problem described in §2. We take a specific problem instance involving 10 matrices in $\mathbf{R}^{10\times 10}$, so the semidefinite program has dimensions $m = 11$ and $n = 20$. We use potential reduction method 2. Experimentation with other problems (not shown here) shows that the results for this particular problem instance and this algorithm are quite typical.

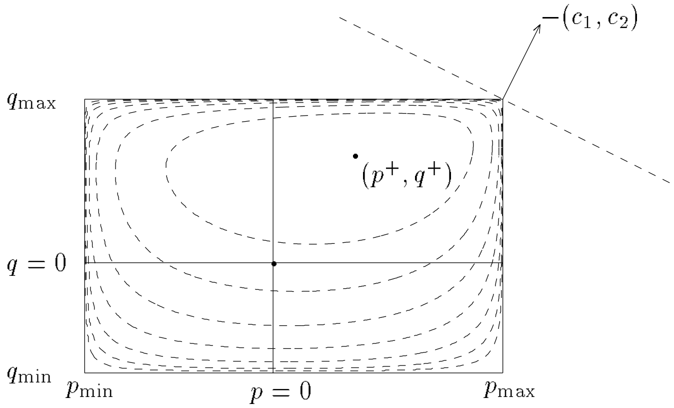

FIG. 8. *Example of a plane search. The intersection of the feasible set with a plane* $(x + p\delta x, Z + q\delta Z)$ *is a rectangle. The dashed lines are the contour lines of the potential function* φ*. The plane search replaces the current* (x, Z) *by* $(x + p^+\delta x, Z + q^+\delta Z)$*, where* (p^+, q^+) *minimizes the potential function in this plane. The upper right corner minimizes the duality gap over all points* $(x + p\delta x, Z + q\delta Z)$*.*

In Figure 9 we compare the actual potential function with the upper bound guaranteed by the theory, for the two parameter values $\nu = 1$ and $\nu = 5$. Note that, especially for the larger value of ν, the actual reduction in potential per iteration is much larger than the lower bound of 0.05 given by Theorem 5.2. The nearly regular and linear decrease of the potential function is typical.

The right plot shows the duality gap during the iteration. We see that the duality gap decreases at a very regular, linear rate. The number of iterations required depends on the value of ν. For $\nu = 1$, 28 iterations are needed to reduce the duality gap from the initial value of 1 to 0.0001; for $\nu = 5$, the number of iterations is 10. These numbers are typical, and, as we will see later, very insensitive to the problem size.

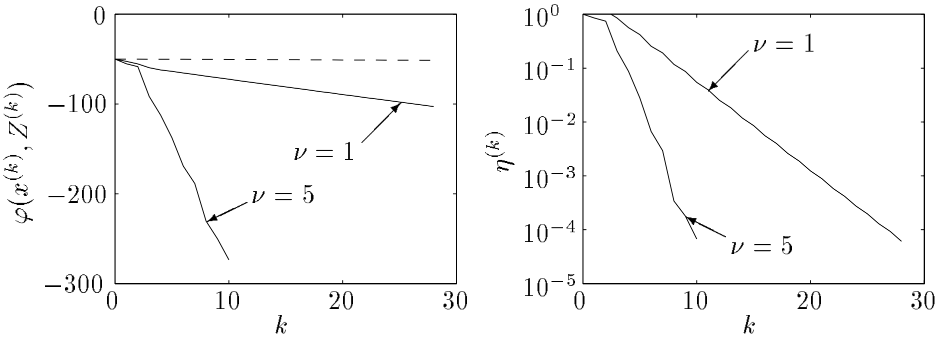

FIG. 9. *The potential function* φ *and the duality gap* η *versus the iteration number* k*, for two values of* ν*. The problem is a matrix norm minimization problem with 10 matrices in* $\mathbf{R}^{10\times10}$*. The dashed line in the left-hand plot shows the upper bound given by Theorem 5.3,* i.e., *a reduction of 0.05 per iteration.*

Another view of the algorithm is given in Figure 10, which shows the trajectories of the duality gap η and the deviation from centrality ψ on a two-dimensional plot. The left plot shows the trajectories for $\nu = 1$; in the right plot we have $\nu = 5$. Each plot shows the trajectories for two different starting points: one on the central path ($\psi = 0$), and one at $\psi = 10$ (The starting point at $\psi = 10$ was also used in Figure 9).

The central path is the horizontal line $\psi = 0$. The dashed lines are the level curves of $\varphi = \nu\sqrt{n}\log\eta + \psi$. Since the duality gap is plotted on a logarithmic scale, these level curves appear as straight lines, with slope determined by ν.

Several features can be seen in the plots. After a few iterations the trajectories that started at poorly centered points (*i.e.*, $\psi(x, Z) = 10$) have been centered to the same rough level as the trajectories that started from the central path. Thereafter the deviation from centrality remains roughly constant, with the constant depending on the value of ν, which sets the relative weight between deviation from centrality and duality gap. For example, with $\nu = 5$ the iterates remain at a deviation of approximately $\psi \approx 2.5$. Recall that this means that the iterates could be centered in no more than about 33 Newton steps. One consequence of ψ remaining approximately constant is that the reduction in potential at each iteration is all due to duality gap reduction.

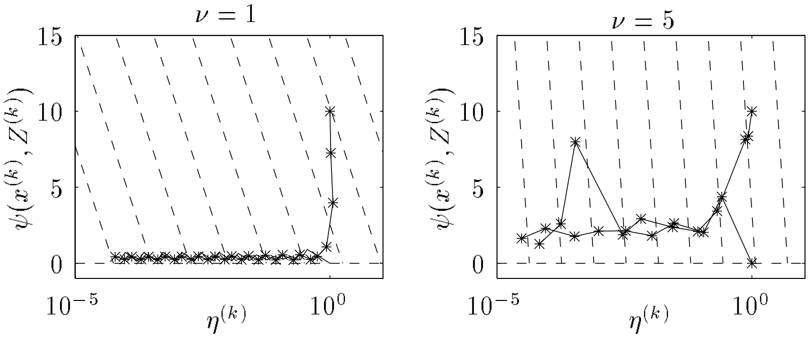

FIG. 10. *Trajectories of the duality gap η and the deviation from centrality ψ, for $\nu = 1$ (at left) and $\nu = 5$ (at right), with two different starting points. The first starting point lies on the central path ($\psi = 0$); the second point lies at $\psi = 10$. The dashed lines are level curves of the primal-dual potential function φ.*

5.7. Dependence on problem size. A natural question is: What is the computational effort required to solve a semidefinite program using the methods described above, and more specifically, how does it grow with problem size? In terms of iterations required, all the methods we have described have the same worst-case complexity: The number of iterations to solve a semidefinite program to a given accuracy grows with problem size as $O(n^{1/2})$. In practice as well, the algorithms behave very similarly, and much better than predicted by the worst-case complexity analyses. It has been observed by many researchers that the number of iterations required grows much more slowly than $n^{1/2}$, perhaps like $\log n$ or $n^{1/4}$, and can often be assumed to be almost constant (see Nesterov and Nemirovsky [76, §6.4.4], or Gonzaga and Todd [43] for comments on the average behavior). For a very wide variety of problems, and a large range of problem sizes, the methods described above typically require between 5 and 50 iterations.

This phenomenon is illustrated in Figure 11, which shows duality gap versus iterations for three instances of the matrix norm minimization problem, using potential reduction method 2, with $\nu = 10$. The smallest problem involves 10 matrices of size 10×10 (*i.e.*, $m = 11$, $n = 20$); another problem involves 10 matrices of size 70×70 ($m = 11$, $n = 140$); and the last problem involves 100 matrices of size 20×20 ($m = 101$, $n = 40$). The total size of the problem data for the two larger problems is about 50 times larger than for the smaller problem. Nevertheless the plots look remarkably similar: In all three cases, we observe the steady, nearly linear convergence of the duality gap that we have observed before. In fact this behavior is typical for general semidefinite programs, and not just matrix norm problems.

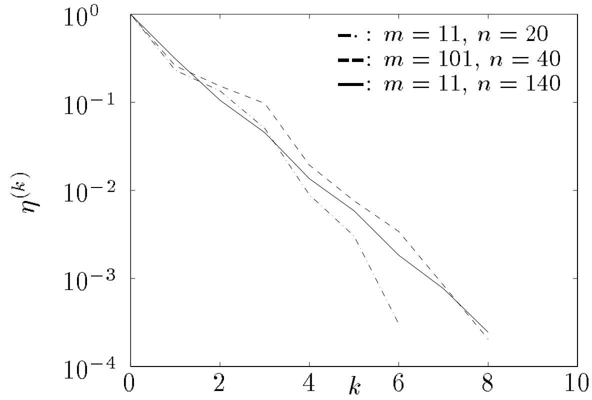

FIG. 11. *Duality gap versus iteration number k for three instances of the matrix norm minimization problem with different dimensions m and n, using potential reduction method 2. Although the (total) problem sizes vary over a $50:1$ range, the convergence is quite similar. The stopping criterion is 0.1% relative accuracy.*

The stopping criterion is 0.1% relative accuracy, *i.e.*, the algorithm was terminated when the duality gap became smaller than 0.1% of the primal objective value. (A stopping criterion based on relative accuracy is natural for matrix norm minimization. For other problems, one may prefer an absolute, or a combination of an absolute and a relative criterion.)

With this stopping criterion, the smaller problem required only 6 iterations, and the larger problems only 8 iterations. We should note that while the number of iterations required to solve the three problems varied only from 6 to 8, the total solution time varied by a factor exceeding 500:1, due to the range in size of the least-squares problems solved in each iteration.

To give a more convincing illustration of the regularity of the convergence and insensitivity to problem size, we generated and solved 340 matrix norm problem instances. The matrices A_i were chosen from a normal distribution, and then scaled so that $A_0 = 0.5$. As starting point, we take $t = 1$, $x = 0$, and $Z = (1/2p)I$. As in the examples above, we use the method of §5.3 with $\nu = 10$. The stopping criterion is a relative duality gap less than 0.1%. In one experiment, we take a fixed number of matrices, 10, and vary the size of A_i from 10×10 to 70×70. In the other experiment, the size of the matrices is fixed at 20×20, and we vary the number of matrices from 10 to 100. For each combination of sizes we generate and solve 20 problem instances.

Figure 12 shows the average number of iterations required as a function of the dimension, along with bars indicating the standard deviation. Over the 340 problem instances solved, the algorithm never needed less than six or more than 10 iterations.

Since the number of iterations required is quite insensitive to problem size, the next natural question is, what is the work required per iteration? Unfortunately (or perhaps, fortunately) there is no simple answer to this question since it depends very much on the amount of structure in the matrices F_i that the user will exploit. We will come back to this question in §7.6.

While the methods described above perform quite similarly in practice, we can still

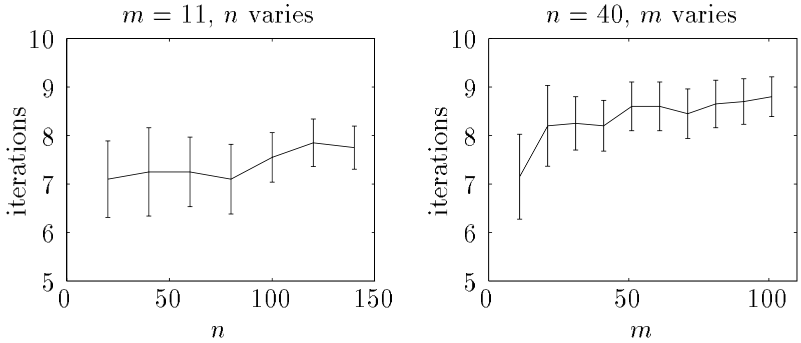

FIG. 12. *The average number of iterations to solve the matrix norm minimization problem with k matrices in $\mathbf{R}^{p\times p}$, which yields a semidefinite program of dimension $m = k+1$, $n = 2p$. In the left plot $k = 10$ ($m = 11$) is fixed and we vary p (n). In the right plot $p = 20$ ($n = 40$) is fixed and we vary k (m). Each point is the average of 20 random instances. The error bars show the standard deviation.*

make a few comments comparing them. Our experience, mostly with problems arising in control theory, suggests that the potential reduction method 1 often takes a few more iterations than the methods 2, $2^\star$, and 3, and also requires the solution of two sets of equations per iteration. The methods 2, $2^\star$, and 3 appear to be quite comparable, and have some practical advantage over method 1. (Moreover, the distinction between method 2 and $2^\star$ is just convention, since we could just as well refer to the dual problem as the primal, and vice versa.)

Finally we note that since the algorithms all reduce the same potential function, we can switch arbitrarily between them. For example, we can use method 2 for the even iterations and method $2^\star$ for the odd iterations. Although the possibility of switching is interesting, we do not know whether it yields any practical advantage.

6. Phase I and combined Phase I–Phase II methods. We have assumed so far that initial strictly feasible primal and dual points are known. That is often the case, as in the minimum matrix norm problem of §2. This section describes what to do when an initial primal strictly feasible or dual strictly feasible point is not known.

6.1. Big-M method. The 'big-M' method is standard in nonlinear programming; see, *e.g.*, Bazaraa, Sherali and Shetty [12], or Anstreicher [10]. We distinguish three cases.

Case 1. A strictly feasible x is known, but no strictly feasible Z.

Case 2. A strictly feasible Z is known, but no strictly feasible x.

Case 3. Neither a strictly feasible x nor a strictly feasible Z is known.

Case 1. Assume that a strictly feasible $x^{(0)}$ is given, but no strictly feasible dual point. In this case one can modify the primal problem by introducing an upper bound on the trace of $F(x)$:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F(x) \geq 0 \\ & \mathbf{Tr}\, F(x) \leq M. \end{array} \tag{70}$$

If M is big enough, this entails no loss of generality: the solutions of (70) and the original semidefinite program (1) are the same (assuming $p^* > -\infty$). The initial point $x^{(0)}$ will still be strictly feasible if $\mathbf{Tr}\, F(x^{(0)}) < M$.

The dual of the modified problem (70) is

$$\begin{array}{ll} \text{maximize} & -\mathbf{Tr}F_0(Z-zI)-Mz \\ \text{subject to} & \mathbf{Tr}F_i(Z-zI)=c_i,\ i=1,\ldots,m \\ & Z\geq 0, z\geq 0 \end{array} \tag{71}$$

where z is a scalar variable that did not appear in the dual (27) of the original semidefinite program. It is easy to compute strictly feasible points for problem (71). First compute any solution $U=U^T$ to the underdetermined set of equations

$$\mathbf{Tr}F_iU=c_i,\ i=1,\ldots,m.$$

Take $z^{(0)}>-\min\{\lambda_{\min}(U),0\}$, and then set $Z^{(0)}=U+z^{(0)}I$. One can verify that $Z^{(0)}$, $z^{(0)}$ are strictly feasible for (71). Any of the primal-dual methods described above can now be used, starting at the initial points $x^{(0)}$ and $Z^{(0)}$, $z^{(0)}$.

The difficulty with this scheme is the choice of M. Sometimes it is possible to (analytically) determine an appropriate value for M from the problem data. In other cases we need to check that at the solution of the modified problem (70), the extra constraint $\mathbf{Tr}F(x)\leq M$ is not active, *i.e.*, we have $\mathbf{Tr}F(x)<M$. If this is the case then we have actually solved the original semidefinite program (1); if not, we can increase M and solve the modified problem (70) again.

Note that M serves as an upper bound on $F(x)$ (*e.g.*, it implies $\|F(x)\|\leq M$) which in turn bounds the (primal) feasible set. As a very rough rule of thumb, simple bounds on the primal variables often lead to easy identification of strictly feasible dual points.

Case 2. This is dual to the previous case. Assume we have a dual strictly feasible point $Z^{(0)}$, but no primal strictly feasible point. One can then add 'big M'-terms to the primal problem:

$$\begin{array}{ll} \text{minimize} & c^Tx+Mt \\ \text{subject to} & F(x)+tI\geq 0 \\ & t\geq 0. \end{array} \tag{72}$$

To obtain a strictly feasible solution to (72), choose any $x^{(0)}$, and take

$$t^{(0)}>-\min\{\lambda_{\min}(F(x^{(0)})),0\}.$$

The dual of problem (72) is

$$\begin{array}{ll} \text{maximize} & -\mathbf{Tr}F_0Z \\ \text{subject to} & \mathbf{Tr}F_iZ=c_i,\ i=1,\ldots,m \\ & \mathbf{Tr}Z+z=M \\ & Z\geq 0, z\geq 0, \end{array}$$

or, if we eliminate the slack variable z,

$$\begin{array}{ll} \text{maximize} & -\mathbf{Tr}F_0Z \\ \text{subject to} & \mathbf{Tr}F_iZ=c_i,\ i=1,\ldots,m \\ & \mathbf{Tr}Z\leq M. \end{array}$$

From this we see that we have modified the dual semidefinite program (27) by adding an upper bound on the trace of Z.

Case 3. When no primal or dual strictly feasible points are known, one can combine the two cases above and introduce two coefficients, M_1 and M_2. The primal problem becomes

$$\begin{array}{ll} \text{minimize} & c^T x + M_1 t \\ \text{subject to} & F(x) + tI \geq 0 \\ & \mathbf{Tr} F(x) \leq M_2 \\ & t \geq 0, \end{array}$$

and the dual

$$\begin{array}{ll} \text{maximize} & -\mathbf{Tr} F_0(Z - zI) - M_2 z \\ \text{subject to} & \mathbf{Tr} F_i(Z - zI) = c_i, \ i = 1, \ldots, m \\ & \mathbf{Tr} Z \leq M_1 \\ & Z \geq 0, z \geq 0. \end{array}$$

6.2. Other methods. Several methods are known that can start at infeasible points, and do not require big-M terms. Examples are Nesterov and Nemirovsky's projective method [76], and the primal-dual methods described by Helmberg, Rendl, Vanderbei and Wolkowicz [46], Alizadeh, Haeberly and Overton [5], Kojima, Shindoh and Hara [61], and Nesterov [70].

7. Some extensions.

We mention a few interesting extensions of, and variations on, the semidefinite programming problem.

7.1. Generalized linear-fractional programming. In §2 we saw that the problem of minimizing the maximum eigenvalue of a symmetric matrix $A(x)$ can be cast as a semidefinite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tI - A(x) \geq 0. \end{array}$$

Suppose now we have a pair of matrices $(A(x), B(x))$, both affinely dependent on x. In order to minimize their maximum generalized eigenvalue, we can solve the optimization problem

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tB(x) - A(x) \geq 0 \\ & B(x) \geq 0. \end{array} \tag{73}$$

This is called a generalized linear-fractional problem. It includes the linear fractional problem

$$\begin{array}{ll} \text{minimize} & \dfrac{c^T x + d}{e^T x + f} \\ \text{subject to} & Ax + b \geq 0, \quad e^T x + f > 0 \end{array}$$

as a special case.

Problem (73) is not a semidefinite program, however, because of the bilinear term $tB(x)$. It is a quasi-convex problem, and can still be efficiently solved. See Boyd and El Ghaoui [16], Haeberly and Overton [45], and Nesterov and Nemirovsky [75, 68] for details.

7.2. Determinant maximization. In §4.3 we discussed the problem of minimizing the barrier function $-\log\det F(x)$, or, equivalently, maximizing the determinant of $F(x)$, over all x such that $F(x) > 0$. This problem often arises in combination with additional linear matrix inequality constraints:

$$\begin{array}{ll} \text{minimize} & \log\det F(x)^{-1} \\ \text{subject to} & F(x) > 0 \\ & C(x) \geq 0 \end{array} \tag{74}$$

where $C(x) \overset{\Delta}{=} C_0 + x_1C_1 + \cdots + x_mC_m$. This problem is a convex programming problem, and can be solved very efficiently. In fact, Nesterov and Nemirovsky [76, §6.4.3] have showed that (74) can be cast as a semidefinite program, although it is more efficient to solve it directly. An important application is the computation of the maximum volume ellipsoid contained in a polytope; see Nesterov and Nemirovsky [76, §6.5] or Khachiyan and Todd [56] for interior-point methods to solve this problem.

7.3. Rank minimization. If the semidefinite program (1) is solvable, its solution lies on the boundary of the feasible set, *i.e.*, at a point where $F(x)$ is singular. This observation motivates the second extension: minimizing the rank of a positive semidefinite symmetric matrix,

$$\begin{array}{ll} \text{minimize} & \mathbf{rank}\ B(x) \\ \text{subject to} & A(x) \geq 0,\ B(x) \geq 0, \end{array}$$

where A and B are symmetric matrices that dependly affinely on x. Many problems in control theory, statistics, and other fields can be reduced to this problem.

In contrast with semidefinite programs, generalized fractional problems, and determinant maximization problems, however, this problem is hard. That general rank constraints can greatly increase the complexity of the problem is to be expected; we saw in §2 that the difference between the NP-hard indefinite quadratic problem (14) and the semidefinite relaxation (16) is exactly the constraint that a matrix have rank one.

As another example, we can formulate Boolean linear programming as a rank minimization problem. Consider the problem of determining whether there is an $x \in \mathbf{R}^m$ such that $Cx + d \geq 0$ and $x_i \in \{0, 1\}$, which is NP-hard. It can be formulated as the rank minimization problem with

$$A(x) = \mathbf{diag}(Cx + d), \qquad B(x) = \mathbf{diag}(x_1, \ldots, x_m, 1 - x_1, \ldots, 1 - x_m).$$

Here the rank of B is always at least m, and is m only when $x_i \in \{0, 1\}$.

Some special problems involving rank contraints can be solved efficiently; see [104].

7.4. General conic formulation. Semidefinite programming can be considered as an extension of linear programming in which the positive orthant is replaced by the cone of positive definite matrices. Semidefinite programming, in turn, can be further generalized to the case of a general, pointed cone. This general conic formulation is discussed in Wolkowicz [112] and Nesterov and Nemirovsky [76]. The methods described here can be extended to the general conic formulation; see chapters 4–6 of [76].

7.5. More efficient barriers. One can also replace the barrier function by several others that result in better worst-case complexity estimates. Nesterov and

Nemirovsky [76, §5.5] have generalized Vaidya's volumetric and combined volumetric barriers to the cone of positive semidefinite matrices. We do not know of any experimental results that indicate whether these improved barrier functions are better in practice than the standard barrier $\log\det F(x)^{-1}$.

7.6. Exploiting problem structure. It is possible to modify the semidefinite program methods described above to exploit problem structure. The dominant part in every iteration is the solution of a linear system of the form (57), or a least-squares problem of the form (60). Problem (60) has m variables and $n(n+1)/2$ equations, and can be solved in $O(m^2n^2)$ operations using direct methods. Important savings are possible when the matrices F_i are structured. The easiest type of structure to exploit is block-diagonal structure. Assume $F(x)$ consists of L diagonal blocks of size n_i, $i = 1,\ldots,L$. Then the number of equations in (60) is $\sum_{i=1}^{L} n_i(n_i+1)/2$, which is often an order less than $n(n+1)/2$. For instance, in the LP case (diagonal matrix $F(x)$), the number of variables is n, and solving the least-squares problem requires only $O(m^2n)$ operations.

Usually much more can be gained by exploiting the internal structure (*e.g.*, sparse, Toeplitz, etc.) of the diagonal blocks in F_i. In this section we give an overview of several techniques that have been used for exploiting structure in LPs, and point out the parallels and differences with semidefinite programming.

As in linear programming, we can distinguish between direct and iterative methods.

Direct sparse-matrix techniques. Several strategies have been proposed to solve systems (62) when the matrix A is large and sparse.

The most popular and fastest approach consists in reducing the system to

$$A^T S^{-2} A \delta x = -A^T S^{-2} d. \tag{75}$$

Since S is diagonal, the product in (75) is usually still sparse. This depends on the sparsity pattern of A, however. Dense rows in A, for example, have a catastrophic effect on sparsity. Equation (75) can be solved using a sparse Cholesky decomposition [66].

The second strategy is to solve the sparse system (62) directly. Several researchers have argued that this method has better numerical properties (See Fourer and Mehrotra [32], Gill, Murray, Ponceleon, and Saunders [38], and Vanderbei and Carpenter [109]). Moreover, directly solving (62) avoids the loss of sparsity caused by squaring A.

Neither of these techniques works for semidefinite programs unfortunately, because they lead to systems with large dense blocks, even if the matrices F_i are sparse.

A third possibility that avoids this difficulty is to introduce new variables $W \in \mathbf{R}^{n\times n}$ and to write (57) as

$$\begin{array}{rcl} -W^T + \delta Z S &=& 0 \\ \frac{1}{2}(WS + SW^T) + \sum_{i=1}^{m} \delta x_i F_i &=& -D \\ \mathbf{Tr} F_j \delta Z &=& 0,\ \ j = 1,\ldots,m. \end{array} \tag{76}$$

This is a sparse, symmetric indefinite system that can be solved using sparse-matrix techniques.

Iterative techniques. A second group of methods solves the equations (61), (57) or (76) iteratively.

For (61) or (60) the conjugate gradients method or the LSQR algorithm of Paige and Saunders [83] appear to be very well suited. In exact arithmetic, these algorithms solve (60) in $m+1$ iterations, where each iteration requires an evaluation of the two (adjoint) linear mappings

$$(v_1,\ldots,v_m)\mapsto\sum_{i=1}^{m} v_iF_i, \quad \text{and} \quad W\mapsto(\mathbf{Tr}F_1W,\ldots,\mathbf{Tr}F_mW) \tag{77}$$

for some vector v and matrix $W=W^T$. When the matrices F_i are unstructured, these two operations take mn^2 operations. Hence, the cost of solving (60) using LSQR is $O(n^2m^2)$, and nothing is gained over direct methods.

In most cases, however, the two operations (77) are much cheaper than mn^2 because of the special structure of the matrices F_i. The equations are often dense, but still highly structured in the sense that the two linear functions (77) are easy to evaluate. References [18, 108] discuss iterative methods for exploiting structure in semidefinite programs arising in engineering.

One can also consider solving the symmetric systems (57) or (76) iteratively, using Paige and Saunders' SYMMLQ method [82], or Freund and Nachtigal's symmetric QMR method [35]. Working on (57) or (76) has the advantage of allowing more freedom in the selection of preconditioners [38].

In practice, *i.e.*, with roundoff error, convergence of these methods can be slow and the number of iterations can be much higher than $m+1$. There are techniques to improve the practical performance, but the implementation is very problem specific, and falls outside the scope of this paper.

8. Conclusions.

Semidefinite programming can be considered an extension of linear programming that includes a wide variety of interesting nonlinear convex optimization problems. We have described several primal-dual interior-point methods for semidefinite programs that generalize interior-point methods devised for LPs.

While the details of the primal-dual algorithms are different, they have similar structure and worst-case complexity analysis, and behave similarly in practice:

- *Common structure.* Each iteration involves the solution of one (or two) least-squares problems to determine suitable primal and dual search directions. Suitable step lengths are determined by solving a (smooth, quasiconvex) two dimensional optimization problem that involves only the duality gap and the deviation from centrality. The computational effort of this plane search is greatly reduced by first diagonalizing (or tridiagonalizing) the matrices involved.
- *Worst-case complexity.* Each of the algorithms can be proved to reduce a potential function by at least some fixed amount at each iteration. Hence the number of iterations required to solve a semidefinite program to a given accuracy can grow no faster than the squareroot of the problem size.
- *Practical performance.* In practice the algorithms perform much better than the worst-case bound. The decrease in potential function at each iteration is usually much more than the minimum guaranteed. The convergence of the duality gap is quite regular and nearly linear.

  The number of iterations required appears to grow much more slowly with problem size than the squareroot bound given by the theory. For practical purposes the number of iterations required can be considered almost independent of problem size, ranging between 5 and 50.

In summary, primal-dual algorithms for semidefinite programs share many of the features and characteristics of the corresponding algorithms for LPs. Our final conclusion is therefore: it is not much harder to solve a rather wide class of nonlinear convex optimization problems than it is to solve LPs.

Acknowledgments. We thank Hervé Lebret who helped us with an early draft of this paper, and coded and tested most of the algorithms described here (as well as many others); also Michael Grant, Benoît Lecinq, and Shao-Po Wu who helped with several related software projects. The comments and suggestions of Farid Alizadeh, Istvan Kollar, Claude Lemaréchal, Julia Olkin, and two anonymous reviewers improved the content and presentation of this paper considerably.

REFERENCES

[1] F. Alizadeh, *Combinatorial optimization with interior point methods and semi-definite matrices*, PhD thesis, Univ. of Minnesota, Oct. 1991.

[2] ———, *Combinatorial optimization with semidefinite matrices*, in Proceedings of second annual Integer Programming and Combinatorial Optimization conference, Carnegie-Mellon University, 1992.

[3] ———, *Optimization over the positive-definite cone: interior point methods and combinatorial applications*, in Advances in Optimization and Parallel Computing, P. Pardalos, ed., North-Holland, 1992.

[4] ———, *Interior point methods in semidefinite programming with applications to combinatorial optimization*, SIAM Journal on Optimization, 5 (1995), pp. 13–51.

[5] F. Alizadeh, J.-P. Haeberly, and M. Overton, *Primal-dual interior-point methods for semidefinite programming*. Manuscript, 1994.

[6] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Programming (Series B), (1996). To appear.

[7] J. Allwright, *Positive semidefinite matrices: characterization via conical hulls and least-squares solution of a matrix equation*, SIAM Journal on Control and Optimization, 26 (1988), pp. 537–556.

[8] ———, *On maximizing the minimum eigenvalue of a linear combination of symmetric matrices*, SIAM J. on Matrix Analysis and Applications, 10 (1989), pp. 347–382.

[9] E. J. Anderson and P. Nash, *Linear Programming in Infinite-Dimensional Spaces: Theory and Applications*, John Wiley & Sons, 1987.

[10] K. M. Anstreicher, *A combined phase I – phase II scaled potential algorithm for linear programming*, Mathematical Programming, 52 (1991), pp. 429–439.

[11] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.

[12] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming. Theory and Algorithms.*, Wiley, second ed., 1993.

[13] R. Bellman and K. Fan, *On systems of linear inequalities in Hermitian matrix variables*, in Convexity, V. L. Klee, ed., vol. 7 of Proceedings of Symposia in Pure Mathematics, American Mathematical Society, 1963, pp. 1–11.

[14] A. Ben-Tal and M. P. Bendsøe, *A new method for optimal truss topology design*, SIAM Journal on Optimization, 3 (1993), pp. 322–358.

[15] P. M. Bentler and J. A. Woodward, *Inequalities among lower bounds to reliability: with applications to test construction and factor analysis*, Psychometrika, 45 (1980), pp. 249–267.

[16] S. Boyd and L. El Ghaoui, *Method of centers for minimizing generalized eigenvalues*, Linear Algebra and Applications, special issue on Numerical Linear Algebra Methods in Control, Signals and Systems, 188 (1993), pp. 63–111.

[17] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15 of Studies in Applied Mathematics, SIAM, Philadelphia, PA, June 1994.

[18] S. Boyd, L. Vandenberghe, and M. Grant, *Efficient convex optimization for engineering design*, in Proceedings IFAC Symposium on Robust Control Design, Sept. 1994, pp. 14–23.

[19] S. Boyd and S. Wu, *SDPSOL: A Parser/Solver for Semidefinite Programs with Matrix Structure. User's Guide, Version Alpha.*, Stanford University, Mar. 1995.

[20] B. Craven and B. Mond, *Linear programming with matrix variables*, Linear Algebra and Appl., 38 (1981), pp. 73–80.

[21] J. Cullum, W. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices*, Math. Programming Study, 3 (1975), pp. 35–55.

[22] D. den Hertog, *Interior Point Approach to Linear, Quadratic and Convex Programming*, Kluwer, 1993.

[23] I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Soviet Math. Dokl., 8 (1967), pp. 674–675.

[24] W. E. Donath and A. J. Hoffman, *Lower bounds for the partitioning of graphs*, IBM Journal of Research and Development, 17 (1973), pp. 420–425.

[25] K. Fan, *On a theorem of Weyl concerning eigenvalues of linear transformations. I*, Proceedings of the National Academy of Sciences of the U.S.A., 35 (1949), pp. 652–655.

[26] M. K. H. Fan, *A quadratically convergent algorithm on minimizing the largest eigenvalue of a symmetric matrix*, Linear Algebra and Appl., 188-189 (1993), pp. 231–253.

[27] M. K. H. Fan and B. Nekooie, *On minimizing the largest eigenvalue of a symmetric matrix*, in Proc. IEEE Conf. on Decision and Control, Dec. 1992, pp. 134–139.

[28] L. Faybusovich, *On a matrix generalization of affine-scaling vector fields*, SIAM J. on Matrix Analysis and Applications, 16 (1995), pp. 886–897.

[29] A. Fiacco and G. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques*, Wiley, 1968. Reprinted 1990 in the SIAM Classics in Applied Mathematics series.

[30] R. Fletcher, *A nonlinear programming problem in statistics (educational testing)*, SIAM Journal on Scientific and Statistical Computing, 2 (1981), pp. 257–267.

[31] ———, *Semidefinite matrix constraints in optimization*, SIAM J. Control and Opt., 23 (1985), pp. 493–513.

[32] R. Fourer and S. Mehrotra, *Performance of an augmented system approach for solving least-squares problems in an interior-point method for linear programming*, Mathematical Programming Society. Committee on Algorithms Newsletter, (1991), pp. 26–31.

[33] A. L. Fradkov and V. A. Yakubovich, *The $\mathcal{S}$-procedure and duality relations in nonconvex problems of quadratic programming*, Vestnik Leningrad Univ. Math., 6 (1979), pp. 101–109. In Russian, 1973.

[34] R. Freund, *Complexity of an algorithm for finding an approximate solution of a semi-definite program with no regularity assumption*, Tech. Report OR 302-94, Operations Research Center, MIT, 1994.

[35] R. Freund and N. Nachtigal, *A new Krylov-subspace method for symmetric indefinite linear systems*, Numerical Analysis Manuscript 94-07, AT&T Bell Laboratories, 1994.

[36] P. Gahinet and A. Nemirovskii, *LMI Lab: A Package for Manipulating and Solving LMIs*, INRIA, 1993.

[37] P. Gahinet, A. Nemirovskii, A. J. Laub, and M. Chilali, *The LMI Control Toolbox*, in Proc. IEEE Conf. on Decision and Control, Dec. 1994, pp. 2083–2041.

[38] P. E. Gill, W. Murray, D. B. Ponceleon, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. on Matrix Analysis and Applications, 13 (1992), pp. 292–311.

[39] M. Goemans and D. Williamson, *.878-approximation algorithm for max-cut and max-2sat*, tech. report, MIT, 1994.

[40] ———, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, tech. report, Department of Mathematics, MIT, 1995.

[41] C. Goh and D. Teo, *On minimax eigenvalue problems via constrained optimization*, Journal of Optimization Theory and Applications, 57 (1988), pp. 59–68.

[42] C. C. Gonzaga, *Path-following methods for linear programming*, SIAM Review, 34 (1992), pp. 167–224.

[43] C. C. Gonzaga and M. J. Todd, *An $O(\sqrt{n}L)$-iteration large-step primal-dual affine algorithm for linear programming*, SIAM Journal on Optimization, 2 (1992), pp. 349–359.

[44] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, vol. 2 of Algorithms and Combinatorics, Springer-Verlag, 1988.

[45] J.-P. A. Haeberly and M. A. Overton, *Optimizing eigenvalues of symmetric definite pencils*, in Proceedings of the 1994 American Control Conference. Baltimore, 1994.

[46] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. on Optimization, (1994). To appear.

[47] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*

II: Advanced Theory and Bundle Methods, vol. 306 of Grundlehren der mathematischen Wissenschaften, Springer-Verlag, New York, 1993.

[48] J.-B. Hiriart-Urruty and D. Ye, *Sensitivity analysis of all eigenvalues of a symmetric matrix*, Numerische Mathematik, 70 (1995), pp. 45–72.

[49] M. Iri and H. Imai, *A multiplicative barrier function method for linear programming*, Algorithmica, 1 (1986), pp. 455–482.

[50] F. Jarre, *An interior-point method for minimizing the maximum eigenvalue of a linear combination of matrices*, SIAM J. Control and Opt., 31 (1993), pp. 1360–1377.

[51] F. John, *On symmetric matrices whose eigenvalues satisfy linear inequalities*, in Proceedings of the American Mathematical Society, E. Dyer, W. H. J. Fuchs, A. P. Mattuck, R. C. Buck, F. John, and I. Reiner, eds., Providence, Rhode Island, 1966, American Mathematical Society, pp. 1140–1145.

[52] A. Kamath and N. Karmarkar, *A continuous approach to compute upper bounds in quadratic maximization problems with integer constraints*, in Recent Advances in Global Optimization, C. Floudas and P. Pardalos, eds., Princeton University Press, Oxford, 1992, pp. 125–140.

[53] ———, *An $O(nL)$ iteration algorithm for computing bounds in quadratic optimization problems*, in Complexity in Numerical Optimization, P. M. Pardalos, ed., World Scientific Publishing Co., Singapore, 1993, pp. 254–268.

[54] D. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring by semidefinite programming*, tech. report, Department of Computer Science, Stanford University, 1994.

[55] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[56] L. G. Khachiyan and M. J. Todd, *On the complexity of approximating the maximal inscribed ellipsoid for a polytope*, Mathematical Programming, 61 (1993), pp. 137–159.

[57] K. C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, vol. 1133 of Lecture Notes in Mathematics, Springer-Verlag, 1985.

[58] M. Kojima, S. Kojima, and S. Hara, *Linear algebra for semidefinite programming*, tech. report, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Oh-Okayama, Meguro-ku, Tokyo 152, Japan, Oct. 1994.

[59] 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, Springer-Verlag, 1991.

[60] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Mathematical Programming, 50 (1991), pp. 331–342.

[61] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone linear complementarity problem in symmetric matrices*, tech. report, Department of Information Sciences. Tokyo Institute of Technology, 1994.

[62] J. B. Lasserre, *Linear programming with positive semi-definite matrices*, Tech. Report LAAS-94099, Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, 1995.

[63] B. Lieu and P. Huard, *La méthode des centres dans un espace topologique*, Numerische Mathematik, 8 (1966), pp. 56–67.

[64] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and 0-1 optimization*, SIAM J. on Optimization, 1 (1991), pp. 166–190.

[65] A. I. Lur'e and V. N. Postnikov, *On the theory of stability of control systems*, Applied mathematics and mechanics, 8 (1944). In Russian.

[66] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Interior point methods for linear programming: Computational state of the art*, ORSA Journal on Computing, 6 (1994).

[67] B. Mohar and S. Poljak, *Eigenvalues in combinatorial optimization*, in Combinatorial and Graph-Theoretical Problems in Linear Algebra, R. A. Brualdi, S. Friedland, and V. Klee, eds., Springer-Verlag, New York, 1993, pp. 107–151.

[68] A. Nemirovsky, *Long-step method of analytic centers for fractional problems*, Tech. Report 3/94, Technion, Haifa, Israel, 1994.

[69] A. Nemirovsky and P. Gahinet, *The projective method for solving linear matrix inequalities*, in Proc. American Control Conf., June 1994, pp. 840–844. Submitted to *Math. Programming, Series B.*

[70] Y. Nesterov, *Infeasible start interior point primal-dual methods in nonlinear programming*, tech. report, CORE, 1994.

[71] Y. Nesterov and A. Nemirovsky, *A general approach to polynomial-time algorithms design for convex programming*, tech. report, Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow, USSR, 1988.

[72] ———, *Optimization over positive semidefinite matrices: Mathematical background and user's*

manual, USSR Acad. Sci. Centr. Econ. & Math. Inst., 32 Krasikova St., Moscow 117418 USSR, 1990.

[73] ———, *Self-concordant functions and polynomial time methods in convex programming*, tech. report, Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow, USSR, Apr. 1990.

[74] ———, *Conic formulation of a convex programming problem and duality*, tech. report, Centr. Econ. & Math. Inst., USSR Academy of Sciences, Moscow USSR, 1991.

[75] ———, *An interior point method for generalized linear-fractional programming*, Submitted to Math. Programming, Series B., (1991).

[76] ———, *Interior-point polynomial methods in convex programming*, vol. 13 of Studies in Applied Mathematics, SIAM, Philadelphia, PA, 1994.

[77] Y. E. Nesterov and M. J. Todd, *Self-scaled cones and interior-point methods in nonlinear programming*, Tech. Report 1091, Cornell University, Apr. 1994.

[78] M. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. on Matrix Analysis and Applications, 9 (1988), pp. 256–268.

[79] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optimization, (1992), pp. 88–120.

[80] M. Overton and R. Womersley, *On the sum of the largest eigenvalues of a symmetric matrix*, SIAM J. on Matrix Analysis and Applications, (1992), pp. 41–45.

[81] ———, *Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices*, Mathematical Programming, 62 (1993), pp. 321–357.

[82] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. on Numerical Analysis, 12 (1975), pp. 617–629.

[83] C. C. Paige and M. S. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Transactions on Mathematical Software, 8 (1982), pp. 43–71.

[84] E. Panier, *On the need for special purpose algorithms for minimax eigenvalue problems*, Journal Opt. Theory Appl., 62 (1989), pp. 279–287.

[85] G. Pataki, *On the multiplicity of optimal eigenvalues*, Tech. Report MSRR-604, Carnegie Mellon University. Graduate School of Industrial Administration, 1994.

[86] ———, *Cone-LP's and semi-definite programs: facial structure, basic solutions, and the simplex method*, tech. report, GSIA, Carnegie-Mellon University, 1995.

[87] S. Poljak, F. Rendl, and H. Wolkowicz, *A recipe for best semidefinite relaxation for* $(0,1)$*-quadratic programming*, Tech. Report CORR 94-7, University of Waterloo, 1994.

[88] F. Pukelsheim, *Optimal Design of Experiments*, Wiley, 1993.

[89] M. Ramana, *An Algorithmic Analysis of Multiquadratic and Semidefinite Programming Problems*, PhD thesis, The Johns Hopkins University, 1993.

[90] ———, *An exact duality theory for semidefinite programming and its complexity implications*, DIMACS Technical Report 95-02, RUTCOR. Rutgers University, 1995. To appear in Mathematical Programming.

[91] M. Ramana and A. Goldman, *Some geometric results in semidefinite programming*, Journal of Global Optimization, 7 (1995), pp. 33–50.

[92] F. Rendl, R. Vanderbei, and H. Wolkowicz, *A primal-dual interior-point method for the max-min eigenvalue problem*, tech. report, University of Waterloo, Dept. of Combinatorics and Optimization, Waterloo, Ontario, Canada, 1993.

[93] U. T. Ringertz, *Optimal design of nonlinear shell structures*, Tech. Report FFA TN 1991-18, The Aeronautical Research Institute of Sweden, Stockholm, 1991.

[94] R. T. Rockafellar, *Convex Analysis*, Princeton Univ. Press, Princeton, second ed., 1970.

[95] J. B. Rosen, *Pattern separation by convex programming*, Journal of Mathematical Analysis and Applications, 10 (1965), pp. 123–134.

[96] A. Shapiro, *Weighted minimum trace factor analysis*, Psychometrika, 47 (1982), pp. 243–264.

[97] ———, *Extremal problems on the set of nonnegative definite matrices*, Linear Algebra and Appl., 67 (1985), pp. 7–18.

[98] A. Shapiro and M. K. H. Fan, *On eigenvalue optimization*, SIAM J. on Optimization, (1994). To appear.

[99] N. Z. Shor, *Cut-off method with space extension in convex programming problems*, Cybernetics, 13 (1977), pp. 94–96.

[100] ———, *Minimization Methods for Non-differentiable Functions*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[101] ———, *Quadratic optimization problems*, Soviet Journal of Circuits and Systems Sciences, 25 (1987), pp. 1–11.

[102] G. Sonnevend, *An 'analytical centre' for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, in Lecture Notes in Control and Information Sciences, vol. 84, Springer-Verlag, 1986, pp. 866–878.

[103] ———, *New algorithms in convex programming based on a notion of 'centre' (for systems*

of analytic inequalities) and on rational extrapolation, International Series of Numerical Mathematics, 84 (1988), pp. 311–326.

[104] P. Tarazaga and M. Trosset, *An optimization problem on subsets of the positive-semidefinite matrices*, Journal of Optimization Theory and Applications, 79 (1993), pp. 513–524.

[105] F. Uhlig, *A recurring theorem about pairs of quadratic forms and extensions: A survey*, Linear Algebra and Appl., 25 (1979), pp. 219–237.

[106] L. Vandenberghe and S. Boyd, *A polynomial-time algorithm for determining quadratic Lyapunov functions for nonlinear systems*, in Eur. Conf. Circuit Th. and Design, 1993, pp. 1065–1068.

[107] ———, *SP: Software for Semidefinite Programming. User's Guide, Beta Version.*, K.U. Leuven and Stanford University, Oct. 1994.

[108] ———, *A primal-dual potential reduction method for problems involving matrix inequalities*, Mathematical Programming, 69 (1995), pp. 205–236.

[109] R. J. Vanderbei and T. J. Carpenter, *Symmetric indefinite systems for interior point methods*, Mathematical Programming, 58 (1993), pp. 1–32.

[110] G. A. Watson, *Algorithms for minimum trace factor analysis*, SIAM J. on Matrix Analysis and Applications, 13 (1992), pp. 1039–1053.

[111] ———, *Characterization of the subdifferential of some matrix norms*, Linear Algebra and Appl., 170 (1992), pp. 33–45.

[112] H. Wolkowicz, *Some applications of optimization in matrix theory*, Linear Algebra and Appl., 40 (1981), pp. 101–118.

[113] M. Wright, *Interior methods for constrained optimization*, Acta Numerica, 1 (1992), pp. 341–407.

[114] Y. Ye, *An $O(n^3L)$ potential reduction algorithm for linear programming*, Mathematical Programming, 50 (1991), pp. 239–258.

[115] D. B. Yudin and A. S. Nemirovski, *Informational complexity and efficient methods for solving complex extremal problems*, Matekon, 13 (1977), pp. 25–45.

[116] K. Zietak, *On the characterization of the extremal points of the unit sphere of matrices*, Linear Algebra and Appl., 106 (1988), pp. 57–75.

[117] ———, *Subdifferentials, faces, and dual matrices*, Linear Algebra and Appl., 185 (1993), pp. 125–141.