INTERIOR POINT METHODS IN SEMIDEFINITE PROGRAMMING WITH APPLICATIONS TO COMBINATORIAL OPTIMIZATION *

FARID ALIZADEH†

Abstract. We study the *semidefinite programming problem* (SDP), i.e the optimization problem of a linear function of a symmetric matrix subject to linear equality constraints and the additional condition that the matrix be positive semidefinite. First we review the classical cone duality as is specialized to SDP. Next we present an interior point algorithm which converges to the optimal solution in polynomial time. The approach is a direct extension of Ye's projective method for linear programming. We also argue that many known interior point methods for linear programs can be transformed in a mechanical way to algorithms for SDP with proofs of convergence and polynomial time complexity also carrying over in a similar fashion. Finally we study the significance of these results in a variety of combinatorial optimization problems including the general 0-1 integer programs, the maximum clique and maximum stable set problems in perfect graphs, the maximum k-partite subgraph problem in graphs, and various graph partitioning and cut problems. As a result, we present barrier oracles for certain combinatorial optimization problems (in particular, clique and stable set problem for perfect graphs) whose linear programming formulation requires exponentially many inequalities. Existence of such barrier oracles refutes the commonly believed notion that in order to solve a combinatorial optimization problem with interior point methods, one needs its linear programming formulation explicitly.

Key words. semidefinite programming, interior point methods, eigenvalue optimization, combinatorial optimization, maximum cliques, perfect graphs, graph partitioning

AMS subject classifications. 90C10, 90C25, 90C27, 15A18, 68R10

1. Introduction.

Consider the following optimization problem which we call the standard *semidefinite programming problem* (SDP):

$$\min\{C \bullet X : \; A_i \bullet X = b_i \;\text{ for } i = 1, \cdots, m \;\text{ and } X \succeq 0\} \tag{1.1}$$

where C, A_i's and X are $n \times n$ matrices, and X is symmetric; the "$\bullet$" operation is the inner product of matrices: $A \bullet B := \sum_{i,j} A_{ij} B_{ij} = \operatorname{trace} A^T B$; and the "inequality" constraint $\succeq$ indicates the *Löwner* partial order, that is, for real symmetric matrices A and B, $A \succeq B$ (respectively $A \succ B$), whenever $A - B$ is positive semidefinite (respectively positive definite.)

The semidefinite programming problem is an extension of linear programming (LP). Specifically if the condition that X is a diagonal matrix is added to the constraint set then (1.1) reduces to linear programming. Semidefinite programs arise in a wide variety of applications from control theory (see [63] and [20]) to combinatorial optimization (see section 5 below) and even structural computational complexity theory (see [21]). The oldest form of semidefinite programming is the evaluation of eigenvalues of a symmetric matrix. In fact, one can reformulate the classical theorems of Rayleigh-Ritz for the largest eigenvalue, and of Fan for the sum of the first few eigenvalues of a symmetric matrix, as semidefinite programs, see [54, 53] and section 4 below. However, for these special cases, techniques of this paper do not seem to be appropriate as better algorithms from both theoretical and pragmatic points of view

* Received by the editors October 25, 1991; accepted for publication (in revised form) August 30, 1993. This research supported in part by NSF grant no. CDA-9211106, by the Air Force Office of Scientific Research grant AFOSR-87-0127, the National Science Foundation grant DCR-8420935 and the Minnesota Supercomputer Institute.

† Rutgers University, RUTCOR, PO Box 5062, New Brunswick NJ 08903-5062. E-Mail: `alizadeh@rutcor.rutgers.edu`

already exist. Most nontrivial semidefinite programs (those that are not equivalent to evaluation of eigenvalues of a symmetric matrix by a simple transformation) arise in the form of minimizing the largest, or sum of the first few largest eigenvalues of the matrix X subject to some linear constraints on X. An early example of such problems were studied by Donath and Hoffman in connection with graph bisection and graph partitioning problems [17, 18]; see section 5 below. Cullum, Donath and Wolfe studied the problem of minimizing the sum of the first few eigenvalues of a linearly constrained matrix in [15]. They analyzed this problem from the point of view of nonsmooth optimization. Also Fletcher studied a similar problem and derived expressions for the subgradients of the sum of the first few eigenvalues of a symmetric matrix and formulated optimality conditions for this problem. In the same spirit as Fletcher, Overton [51] studied the largest eigenvalue of a symmetric matrix as a convex, but nondifferentiable function. Based on earlier work [24], in [51] he derived a quadratically convergent algorithm for the problem of minimizing the largest eigenvalue of an affinely constrained matrix. This work was further extended in [52] where both second order methods based on sequential quadratic programming, and first order methods based on sequential linear programming for large scale problems were developed.

The algorithms contained in the above works are in the same spirit as the simplex method for linear programming in that they are all active set methods and traverse the boundary of the feasible set to converge to the optimal solution. For that reason their worst case computational complexity is likely to be at least as bad as that of the simplex method, though in practice they may be quite good.

Semidefinite programs, however, are polynomial time solvable if an *a priori* bound on the size of their solution is known. This point was implicit in [41] for a special instance of the SDP problem. It was proved in the work of Grötschel, Lovàsz and Schrijver, [30]. Polynomial time solvability of SDP is a direct consequence of the general results based on the ellipsoid method for convex programming. The main point essentially is that optimization of a linear function over a convex set endowed with a separation oracle and an *a priori* bound on the objective can be achieved in polynomial time using the ellipsoid method. For the SDP problem, the separation oracle is to determine whether a given symmetric matrix is positive semidefinite and if not provide a separating hyperplane. Cholesky factorization or eigenvalue and eigenvector evaluations easily provide polynomial time oracles for this task. See [32] for a thorough treatment.

The ellipsoid method, however, has not proven practical in most applications, including SDP. A more recent development is the possibility of using interior point methods to obtain polynomial time algorithms for semidefinite programs. The earliest work in this direction to our knowledge is that of Nesterov and Nemirovskii [48]. In this important work the authors develop a general approach for using interior point methods for solving convex programming problems which is based on the concept of *p–selfconcordant* barrier functions. See the more recent [50] for a complete treatment of this subject. Nesterov and Nemirovskii show that for any convex set K that is endowed with a p–selfconcordant barrier function, there is an interior point algorithm which optimizes a linear function on K. Furthermore, every $O(\sqrt{p})$ iterations of this algorithm results in an interior point with half the distance to the optimal solution. As a special case, Nesterov and Nemirovskii show that linear programs with p inequality constraints, quadratic programs with p convex quadratic constraints and semidefinite programs over $p \times p$ matrices all admit p-selfconcordant barriers. Therefore, the authors extend the revolutionary result of Karmarkar [36] to a rather general class of convex programs.

In this article we study interior point methods for semidefinite programs from an alternative point of view. Our work [1] started somewhat later than, and independent of [48]. Nesterov and Nemirovskii obtain their complexity theorems by specializing their general results to SDP. We, on the other hand, take a specific interior point algorithm for linear programming (i.e Ye's projective potential reduction method [66]) and extend it to SDP. Furthermore, we argue that many known interior point linear programming algorithms can also be transformed into an algorithm for SDP in a *mechanical way*; proofs of convergence and polynomial time computability extend in a similar fashion. Later Jarre in [35] and Vandenberghe and Boyd in [63] developed similar interior point algorithms for special forms of SDP.

Polynomial time interior point methods for SDP have some interesting consequences for combinatorial optimization problems. In order to solve such a problem by the ellipsoid method, an explicit listing of all of the inequalities in its linear programming formulation is not needed. Rather, one only needs a *separation oracle* and an initial ellipsoid containing its feasible region to start the process. However, it is generally believed that in order to apply interior point methods to the same combinatorial optimization problem one needs to have the explicit listing of all of the inequalities in the LP formulation, see [32] and [27]. For instance, Goldfarb and Todd in their survey article on linear programming write:

> ..., it appears that its [Karmarkar's new algorithm] theoretical implications are far more limited than those of the ellipsoid method. Indeed, Karmarkar's algorithm requires the linear programming problem to be given explicitly with all its constraints and variables listed, and does not appear directly susceptible to column or constraint generation. Thus it cannot be used to provide polynomial algorithms for several combinatorial optimization problems that have been successfully analyzed by the ellipsoid method.

With the aid of hindsight we show that this common belief is not completely accurate. Specifically, in this article we present examples of combinatorial optimization problems whose LP formulations require exponentially many inequalities, and yet one can design interior point algorithms which solve them in polynomial time. In fact, we should emphasize that the general results of Nesterov and Nemirovskii imply that in principle one can apply interior point methods to solve combinatorial optimization problems without explicit knowledge of their LP formulation. All that is required is a polynomial time computable *self-concordant barrier oracle* with a polynomially bounded parameter. The most interesting example is the clique and stable set problem in a class of graphs known as *perfect graphs.* In section 5 we construct such a barrier indirectly by an SDP formulation of the problem due to Lovàsz. This example is particularly interesting because presently no linear programming formulation of the stable set and clique problems for perfect graphs with polynomially bounded number of inequalities is known.

Linear programming interior point methods have been used by Goldberg et al [26] to derive *sublinear* time *parallel* algorithms for the bounded weight assignment problem. We show that maximum stable sets for perfect graphs can be computed in randomized sublinear parallel time. Furthermore, based on the work of Lovàsz and Schrijver [42], we argue that in a branch and bound scheme for 0-1 programs interior point SDP algorithms may efficiently yield much sharper bounds than possible from linear programming relaxations of such problems.

In section 2 we review the so called cone duality theory as specialized to semidefinite programs. This theory, though quite classical, is somewhat forgotten in optimization literature. It turns out that at least for SDP, cone duality, which is a generalization of linear programming duality, is most appropriate for interior point methods (this point of view is also expressed in the latest edition of Nesterov and Nemirovskii [50]). In section 3 we develop an interior point algorithm which, as we mentioned, is a direct extension of Ye's projective potential reduction method. Furthermore, we propose a recipe to extend *mechanically* many known interior point algorithms for LP into similar algorithms for SDP. In this section we also go over some differences between SDP and LP as far as interior point methods and polynomial time algorithms in general are concerned. In section 4 we build on the results of Overton and Womersley [54, 53] and derive semidefinite programming formulation for various eigenvalue optimization problems. We also state complementary slackness results for these problems. Finally, in section 5 we study some applications of SDP interior point methods to various combinatorial optimization problems. These include 0-1 integer programs of [42], maximum clique and maximum stable set problems in graphs, and various partitioning and cut problems in graphs.

Notation and terminology. Unless otherwise stated the following convention and terminology is used throughout this article.

The **semidefinite programming problem** (SDP) refers to any optimization problem with any mixture of (symmetric) matrix and scalar–valued variables which has a linear objective function and any combination of linear equality or (either component–wise '$\geq$' or Löwner '$\succeq$') inequality constraints.

We use lower case boldface letters to name column vectors, and upper case letters to name matrices. In particular, $\mathbf{1}$ and $\mathbf{0}$ denote vector of all ones and the zero vector, respectively, and I and 0 denote the identity and zero matrices, respectively. Also, $\mathrm{Diag}(\mathbf{x})$ denotes the diagonal matrix made up of the vector $\mathbf{x}$; $\mathbf{diag}(X)$ is the vector made up of diagonal entries of matrix X. For a vector $\mathbf{x}$, x_j is its j^{th} coordinate; similarly X_{ij} is the i, j entry of matrix X. We sometimes refer to members of $\Re^n$ as n-vectors. $\Re^{\frac{n \times n}{2}}$ is the set of symmetric $n \times n$ matrices.

The i^{th} largest eigenvalue of a symmetric matrix X is $\lambda_i(X)$ (or sometimes another lower case Greek letter, e.g $\omega_i(X)$); its i^{th} largest eigenvalue *absolute-value-wise* is $\lambda^i(X)$ or $\omega^i(X)$. The *Löwner* partial order $\succeq$ and the dot product "$\bullet$" were defined above; the symbol "$\geq$" is used for component-wise comparison between two matrices or two vectors.

For matrices, $\|X\|$ and $\|X\|_2$ are the Frobenius and the spectral norms of X, respectively. Recall that in case of symmetric matrices $\|X\|_2$ equals the spectral radius $\rho(X) = |\lambda^1(X)|$ and

$$\|X\| = \left(\sum_{i,j} X_{ij}^2 \right)^{1/2} = \left(\sum_i \lambda_i(X)^2 \right)^{1/2}.$$

For vectors, $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_\infty$ are the Euclidean and the maximum norms of $\mathbf{x}$; also $\|\mathbf{x}\|_p := (\sum |x_i^p|)^{1/p}$ is the p–norm of $\mathbf{x}$.

If A is a $p \times q$ matrix then $\mathbf{vec}\, A$ is a pq column vector made up of columns of A stacked on each other. If $\mathbf{v}$ is a pq-vector then $\mathrm{Mat}_{pq}\, \mathbf{v}$ is a $p \times q$ matrix whose i^{th} column is made up of the entries at $(i-1)p + 1$ through ip in $\mathbf{v}$; if p and q are clear from the context we drop them from the subscript. For instance the set of relations $A_i \bullet X = b_i$, for $i = 1, \cdots, m$ may be written as $\mathcal{A}\mathbf{vec}X = \mathbf{b}$, where $\mathcal{A} \in \Re^{m\times n^2}$, that is row i of $\mathcal{A}$ is $\mathbf{vec}^T(A_i)$. Also, $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) = \sum y_i A_i$.

$A \otimes B$ is the Kronecker product of matrices: if $A \in \Re^{m\times n}$ and $B \in \Re^{p\times q}$ then $A \otimes B \in \Re^{np\times mq}$ is an $m \times n$ block matrix whose i,j block is $a_{ij}B$. We use the following facts occasionally:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \text{ and } \mathbf{vec}(ABC) = (C^T \otimes A)\mathbf{vec}(B).$$

See Graham's text [29] for definitions and properties of Kronocker products.

If I and J are subsets of integers from 1 to p and from 1 to q, respectively, then $A_{I,J}$ is the submatrix of A whose rows are taken from those rows of A indexed by I, and whose columns are indexed by J. $A_{I.}$ and $A_{.J}$ indicate rows indexed by I and columns indexed by J, respectively. Also if $A \in \Re^{m\times p}$ and $B \in \Re^{m\times q}$ then $[A|B]$ is an $m \times (p+q)$ matrix whose columns are made up of columns of A followed by columns of B.

We use := to define or name the left hand side in terms of the right hand side; in algorithms := is used for assignment.

For any convex cone $\mathcal{K}$, its polar cone $\mathcal{K}^*$ is the set

$$\mathcal{K}^* := \{\mathbf{x} : \text{ for all } \mathbf{a} \in \mathcal{K},\ \mathbf{a}^T\mathbf{x} \geq 0\}.$$

Unless otherwise stated, we use $\mathcal{P}$ for the cone of positive semidefinite matrices. Note that $\mathcal{P}^* = \mathcal{P}$ (this fact is direct consequence of Fejer's theorem in [33]).

$G = (V, E)$ is a simple undirected graph without loops or multiple edges. A *stable set* S in G is a subset of vertices which are mutually nonadjacent. A *clique* K in G is a subset of vertices that are all mutually adjacent. A k-partite graph is one whose vertices can be partitioned into k subsets V_j, for $j = 1, \cdots, k$, where each V_j is a stable set. A *clique covering* of G is a collection K_j, $j = 1, \cdots, k$ of sets of vertices, where each K_j is a clique, and $\cup_j K_j = V$.

2. Duality theory. A duality theory quite similar to that of linear programming may be constructed for the semidefinite programming problem. In this section we state the theory for the standard form SDP problem. We also include proofs of basic results in order to make the paper self–contained. Duality theory for more general forms of SDP follows exactly as in linear programming.

We should mention that LP duality has been extended to optimization problems over convex cones in many works. It is easy to see that any cone $\mathcal{K} \subseteq \Re^n$, which is closed, pointed (that is $\mathcal{K} \cap (-\mathcal{K}) = \{0\}$) and convex, induces a partial order $\geq_{\mathcal{K}}$ on $\Re^n$: $\mathbf{x} \geq_{\mathcal{K}} \mathbf{y}$ iff $\mathbf{x} - \mathbf{y} \in \mathcal{K}$. For instance, the nonnegative orthant and the positive semidefinite matrices induce the component–wise "$\geq$" and the Löwner "$\succeq$" partial orders, respectively. The duality theory in linear programming can be extended to generalized linear programming problems where "$\geq_{\mathcal{K}}$" replaces "$\geq$" in the primal problem and "$\geq_{\mathcal{K}^*}$" replaces "$\geq$" in the dual problem.

Duffin in [19] was the first one to study such generalized duality theories. Later Hurwicz [34], Ben-Israel, Charnes and Kortanek [9], Borwein and Wolkowicz [11, 12], and Wolkowicz [64] among others developed alternative formulations of the duality theory. For a comprehensive treatment of generalized duality theory from the point of view of infinite dimensional linear programs, see the text of Anderson and Nash [3] and for alternative extensions refer to [11, 12]. It is worth mentioning that Anderson and Nash in [3] study the duality theory from the point of view of basic feasible solutions and extend the "tableau based" proofs of LP duality. The latest version of Nesterov and Nemirovskii's text [50] also treats *cone duality* for the general convex programs. Papers of Overton and Womersley [53] and Fletcher [23] treat duality theory for the eigenvalue optimization problem from the point of view of subdifferentials. Such an approach is related to the Kuhn-Tucker duality theory and relies on derivatives or subgradients. Also Lovász in [41], Grötschel, Lovász and Schrijver [30, 31, 32], and Shapiro in [61] study more or less the same duality theory as we do, but their treatment is restricted to a special form of SDP.

We now proceed to state and prove duality for the semidefinite programming problem. However observe that the following development–in particular the weak duality lemma 2.1, lemma 2.2, the extended Farkas' lemma 2.3 and the strong duality theorem 2.8 actually apply to generalized linear programming over any closed, pointed convex cone $\mathcal{K}$ (in the primal) and $\mathcal{K}^*$ (in the dual.)

In semidefinite programming it is convenient to assume that C and A_i are symmetric. There is no loss of generality in this assumption. If C is not symmetric, since $C^T \bullet X = C \bullet X$, we can replace C by $1/2(C + C^T)$. The same argument holds for the A_i's. These assumptions of symmetry allow us to formulate the pair of primal and dual standard SDP problems:

$$\begin{array}{lllll} & \underline{\textbf{Primal}} & & \underline{\textbf{Dual}} & \\ & \min & C \bullet X & \max & \mathbf{b}^T\mathbf{y} \\ (2.1) & \text{s.t.} & A_i \bullet X = b_i \ \text{ for } i = 1, \cdots, m & \text{s.t.} & C - \sum_{i=1}^m y_i A_i \succeq 0. \\ & & X \succeq 0 & & \end{array}$$

Notice the similarity of primal and dual SDP pair to the corresponding linear programming pair. First we state the weak duality lemma.

LEMMA 2.1. *Let X be any feasible matrix for primal and $\mathbf{y}$ any feasible vector for dual. Then $C \bullet X \geq \mathbf{b}^T\mathbf{y}$.*

Proof. We have:

$$\begin{aligned} C \bullet X - \sum_{i=1}^m b_i y_i &= C \bullet X - \sum_{i=1}^m (A_i \bullet X) y_i \\ &= (C - \sum_{i=1}^m y_i A_i) \bullet X \\ &\geq 0. \end{aligned}$$

The last inequality is true because the inner product of two positive semidefinite matrices is nonnegative due to self–polarity of the positive semidefinite cone. □

We now state generalizations of Farkas' lemma. Such generalizations for arbitrary convex cones have been studied as early as 1958 by Hurwicz, [34]. See [3] for references on the history and various extensions of Farkas' lemma to nonpolyhedral cones. Here we study the relevant forms of this lemma in the special case of SDP.

It is not possible to generalize classical Farkas' lemma to nonpolyhedral cones without additional qualifications. The difficulty arises from the fact that affine transformations of closed cones are not necessarily closed, and therefore the appropriate strong forms of separation theorems cannot be invoked. (For polyhedral cones however closedness is preserved under affine transformation.) For our purposes we need to have that the set

$$K_1 := \mathcal{A}(\mathcal{P}) = \{\mathcal{A}\mathbf{vec}X : X \succeq 0\}$$

is closed[1]. One class of sufficient conditions for closedness of K_1 is based on assuming that certain sets associated with $\mathcal{P}$ have nonempty interiors. Such conditions are sometimes referred to as *Slater type constraint qualifications.* Though these conditions are not the weakest possible, they are sufficient for the purposes of this paper. We need in any case to assume nonemptyness of the interior for both primal and dual problems so that we have a valid interior point algorithm. Furthermore, in section 3 we show how any pair of primal and dual semidefinite programs may be transformed into an equivalent pair whose K_1 has nonempty interior–as long as an *a priori* bound on the size of primal and dual feasible sets are known. Here is a lemma of Slater type constraint qualifications:

LEMMA 2.2. *If* $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succ 0$ *for some* $\mathbf{y} \in \Re^m$*, then* K_1 *is closed.*
(Recall that $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) = \sum y_i A_i$.)

Proof. Let $\mathcal{L}$:=$\{\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) : \mathbf{y} \in \Re^m\}$. The condition in the lemma says that

$$\mathcal{L} \cap \mathrm{Int}\,\mathcal{P} \neq \emptyset.$$

Thus any translate of the linear subspace $\mathcal{L}$ also intersects $\mathcal{P}$ and its interior. This is equivalent to saying that every symmetric $n \times n$ matrix can be written as sum of two matrices, one of which is positive semidefinite and the other belongs to $\mathcal{L}$. Therefore, $\Re^{\frac{n \times n}{2}} = \mathcal{P} + \mathcal{L}$. Taking the polar we have

$$\{0\} = \mathcal{P} \cap \mathcal{L}^{\perp}.$$

Here $\mathcal{L}^{\perp}$ is the set $\{X : \mathcal{A}\,\mathbf{vec}X = \mathbf{0}\}$. Hence we have that $X = 0$ is the only solution of the system $\mathcal{A}\,\mathbf{vec}X = 0$, and $X \succeq 0$ and by Theorem 9.1 of [59], p. 73 we conclude that K_1 is closed. □

Now we state the most common form of Farkas' lemma as given in Schrijver's text [60], and as extended to the positive semidefinite cone:

LEMMA 2.3. *(***Extended Farkas' lemma***) Let* $\mathbf{b} \in \Re^m$ *and* $\mathcal{A} \in \Re^{m \times n^2}$ *be a matrix such that its rows* $\mathcal{A}_{i.}^T = \mathbf{vec}A_i$ *where* A_i *are symmetric for* $i = 1, \cdots, m$*. Furthermore, let there be an* m*–vector* $\mathbf{y}$ *such that* $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succ 0$*. Then there exists a symmetric matrix* $X \succeq 0$*, with* $\mathcal{A}\,\mathbf{vec}X = \mathbf{b}$ *if and only if* $\mathbf{y}^T\mathbf{b} \geq 0$ *for all* $\mathbf{y}$ *for which* $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0$.

Proof. For the only if part we have,

$$\mathbf{b}^T\mathbf{y} = (\mathcal{A}\,\mathbf{vec}X)^T\mathbf{y} = \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \bullet X \geq 0.$$

(the last inequality is due to self-polarity of the positive semidefinite cone.) To prove the if part, Suppose that the system $\mathcal{A}\,\mathbf{vec}X = \mathbf{b}$, and $X \succeq 0$ is infeasible. Then $\mathbf{b} \notin K_1 = \{\mathcal{A}\,\mathbf{vec}X : X \succeq 0\}$. By lemma 2.2 K_1 is a closed cone and thus there must exist a hyperplane, specifically a linear half-space, that separates $\mathbf{b}$ and K_1, i.e. there exists some vector $\mathbf{y}$ such that $\mathbf{b}^T\mathbf{y} < 0$ and $(\mathcal{A}\,\mathbf{vec}X)^T\mathbf{y} \geq 0$ for all $X \succeq 0$, (see [59], Theorem 11.7, p. 100). But this means that $X \bullet \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \geq 0$ for all $X \succeq 0$, which is equivalent to $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0$, and therefore the if part of the theorem is proved. □

We may formulate and prove several other variants of Farkas' lemma in a similar vein, all of which are extensions of lemmas for the component-wise inequalities, as given for example in Schrijver's text [60]. Related extensions for infinite programs have been studied in [34] and [13], and in the case of matrix variables in [14]. In all

[1] Alternative extensions without closedness assumption are treated in [11, 12, 64]

of these extensions we need to assume either some closedness criteria, or the lemma must be modified by using cones other than $\mathcal{P}$ (as in [64], for instance.) We mention a few more:

LEMMA 2.4. *Let $\mathcal{A} \in \Re^{n^2 \times m}$ be a matrix whose columns are linearly independent and are of the form $\mathbf{vec}A_i$ for symmetric A_i, and $B \in \Re^{\frac{n \times n}{2}}$. Assume that there exists some symmetric matrix $Y \succ 0$ such that $(\mathbf{vec}Y)^T\mathcal{A} = \mathbf{0}$. Then $\mathrm{Mat}(\mathcal{A}\mathbf{x}) \preceq B$ has a solution in $\mathbf{x}$ if and only if $B \bullet Y \geq 0$ for all $Y \succeq 0$ for which $(\mathbf{vec}Y)^T\mathcal{A} = \mathbf{0}$.*

LEMMA 2.5. *Let $A \in \Re^{m\times n}$ and $B \in \Re^{m\times m}$. Suppose there exist some matrix Y such that $A^TYA \succ 0$. Then the system $AXA^T = B$ and $X \succeq 0$ has a solution iff for all symmetric matrices Y, $A^TYA \succeq 0$ implies that $B \bullet Y \geq 0$.*

LEMMA 2.6. *Let $A \in \Re^{m\times n}$ and $B \in \Re^{m\times m}$. Suppose there exist some matrix Y such that $A^TYA = 0$ and $Y \succ 0$. Then the system $AXA^T \preceq B$ has a solution iff for all symmetric matrices $Y \succeq 0$ and $A^TYA = 0$ implies that $B \bullet Y \geq 0$.*

LEMMA 2.7. *Let $A \in \Re^{m\times n}$ and $B \in \Re^{m\times m}$. Suppose there exist some matrix Y such that $A^TYA \succ 0$. Then the system $AXA^T \succeq B$ and $X \succeq 0$ has a solution iff for all symmetric matrices $Y \succeq 0$ and $A^TYA \succeq 0$ implies that $B \bullet Y \geq 0$.*

A strong duality theorem similar to linear programming holds for SDP. We say the primal problem in (2.1) is *feasible* if the set $\{X \in \Re^{\frac{n \times n}{2}} : \mathcal{A}\mathbf{vec}X = \mathbf{b},\ \text{and}\ X \succeq 0\}$ is nonempty, otherwise we say it is infeasible. Feasibility is defined similarly for the dual in (2.1). Recall that infimum over the empty set is by definition $+\infty$ and similarly supremum over the empty set is $-\infty$. Furthermore, the primal (respectively dual) problem in (2.1) is *unbounded* if the infimum (respectively supremum) over the feasible set is $-\infty$ (respectively $+\infty$).

THEOREM 2.8. *Let*

$$\begin{aligned} z_1 &:= \inf\{C \bullet X : \mathcal{A}\mathbf{vec}X = \mathbf{b},\ \text{and}\ X \succeq 0\}\ \textit{and} \\ z_2 &:= \sup\{\mathbf{b}^T\mathbf{y} : C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0\}. \end{aligned}$$

Assume that there is an m–vector $\mathbf{y}$ such that $\mathcal{A}^T\mathbf{y} \succ 0$. Then $z_2 = z_1$.

Proof. Notice that the dual problem is always feasible, because in the proof of lemma 2.2 we showed that $\Re^{\frac{n \times n}{2}} = \mathcal{P} + \mathcal{L}$, and in particular there are some $\mathbf{y}$ and $S \succeq 0$ such that $\mathrm{Mat}(\mathcal{A}^T\mathbf{y}) + S = C$. If $z_1 = -\infty$ (i.e the primal problem is unbounded) then by the weak duality lemma $z_2 = -\infty$, and the dual problem is infeasible, which is a contradiction. If $z_2 = +\infty$ (i.e the dual problem is unbounded) then by the weak duality lemma 2.1 $z_1 = +\infty$ (i.e the primal is infeasible) and the theorem is proved. Conversely, if $z_1 = +\infty$, then the primal problem is infeasible and the extended Farkas' lemma 2.3 implies that for some vector $\mathbf{y}_1$ we have

$$\mathrm{Mat}\mathcal{A}^T\mathbf{y}_1 \preceq 0 \text{ and } \mathbf{b}^T\mathbf{y}_1 > 0. \tag{2.2}$$

But (2.2) implies that the dual problem is unbounded since to any dual-feasible pair $\mathbf{y}$ one can add an arbitrarily large positive multiple of $\mathbf{y}_1$ and obtain another feasible solution with larger objective function value. Therefore, $z_2 = z_1 = +\infty$. Thus, we may assume that both z_1 and z_2 are finite. Suppose $z_2 < z_1$. Then the system

$$\begin{aligned} &C \bullet X = z_2 \\ &\mathcal{A}\mathbf{vec}X = \mathbf{b} \\ &X \succeq 0 \end{aligned}$$

is infeasible. Therefore, by the extended Farkas' lemma 2.3, there exists a scalar y_0

and m-vector $\mathbf{y}$ such that

$$y_0 C + \sum_{i=1}^{m} y_i A_i \succeq 0 \text{ and } z_2 y_0 + \mathbf{b}^T \mathbf{y} < 0. \tag{2.3}$$

where $\mathbf{vec} A_i$ is the i^{th} row of $\mathcal{A}$. Now:

1. If $y_0 = 0$, (2.3) is equivalent to

$$\mathrm{Mat}(\mathcal{A}^T \mathbf{y}) \succeq 0 \text{ and } \mathbf{b}^T \mathbf{y} < 0,$$

which by extended Farkas' lemma implies that $\mathcal{A}\mathbf{vec}X = \mathbf{b}$ and $X \succeq 0$ is infeasible and thus $z_1 = \infty$.

2. If $y_0 > 0$, then dividing both relations in (2.3) by y_0 we get

$$C - \mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0)) \succeq 0 \text{ and } z_2 - \mathbf{b}^T(-\mathbf{y}/y_0) < 0$$

which means z_2 is not an optimal solution of the dual problem.

3. If $y_0 < 0$, then dividing both relations in (2.3) by $-y_0$ we get

$$-C + \mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0)) \succeq 0 \text{ and } -z_2 + \mathbf{b}^T(-\mathbf{y}/y_0) < 0$$

In fact, since we have strict inequality, we must have

$$-C + \mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0)) \succeq 0 \text{ and } -z_2 + \mathbf{b}^T(-\mathbf{y}/y_0) < -\epsilon$$

for some $\epsilon > 0$. But also, by optimality of z_2 there must exist a $\mathbf{y}^*$ such that

$$C - \mathrm{Mat}(\mathcal{A}^T \mathbf{y}^*) \succeq 0 \text{ and } z_2 - \mathbf{b}^T \mathbf{y}^* < \epsilon.$$

Adding the last two sets of relations we get

$$\mathrm{Mat}(\mathcal{A}^T(-\mathbf{y}/y_0 - \mathbf{y}^*) \succeq 0 \text{ and } \mathbf{b}^T(-\mathbf{y}/y_0 - \mathbf{y}^*) < 0$$

which again by extended Farkas' lemma implies that the primal problem is infeasible and $z_1 = \infty$.

Hence the assumption $z_2 < z_1$ results in contradiction. Since by weak duality lemma we have that $z_2 \leq z_1$ we conclude that $z_2 = z_1$. □

It is also possible to derive a "complementary slackness" theorem. In fact, Grötschel, Lovász and Schrijver in [31] and Shapiro in [61] mention the complementary slackness theorem for a more restricted form of SDP. Note that when the strong duality theorem is true and both primal and dual problems are bounded and feasible then the duality gap $X \bullet S$ vanishes. However, in SDP, as in linear programming, a stronger form of complementary slackness results from this observation. First note the following easy lemma:

LEMMA 2.9. *Let A and B be symmetric $n \times n$ matrices. If $A \succeq 0$, $B \succeq 0$, then $A \bullet B = 0$ if and only if $AB = 0$.*

Proof. Let $B = U\Omega U^T$ be the eigenvalue decomposition of B, with $\Omega = \mathrm{Diag}(\omega_i)$ and $\omega_i \geq 0$ for $i = 1, \cdots, n$. Set $C{:=}U^T AU$, thus $C \succeq 0$, and in particular, its diagonal elements $C_{ii} \geq 0$. We only need to show that $C\Omega = 0$. From $A \bullet B = 0$ we have $C \bullet \Omega = 0$ and therefore, $\sum_{i=1}^n C_{ii}\omega_i = 0$. Since all the summands are nonnegative, it follows that they are all zero. Thus we have:

(i) If $\omega_i > 0$ then $C_{ii} = 0$, and by $C \succeq 0$, the entire row and column i is zero.

(ii) If $C_{ii} > 0$, then $\omega_i = 0$.

Now suppose $(C\Omega)_{ij} \neq 0$ for some i, j. Then $C_{ij}\omega_j \neq 0$, which by (i) above we must have that the entire column j is zero, and so $C_{ij} = 0$, a contradiction.[2] □

Now the complementary slackness theorem is immediate:

THEOREM 2.10. *Let X^* be a feasible matrix for the primal, and $\mathbf{y}^*$ a feasible vector for the dual in* (2.1). *Define $S^*{:=}C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}^*)$. Then X^* and $\mathbf{y}^*$ are primal and dual optimal, respectively, if and only if*

$$X^*S^* = 0. \tag{2.4}$$

Notice that, in contrast with linear programming, component-wise multiplication in the complementary slackness theorem is replaced by the ordinary matrix multiplication. The complementary slackness theorem for SDP can be restated in the following way which makes it quite similar to the LP variant:

COROLLARY 2.11. *Let X^* be a feasible matrix for the primal problem in (2.1) with eigenvalues $\lambda_1, \cdots, \lambda_n$; and $S^*{:=}C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}^*)$ be feasible for the dual problem with eigenvalues $\omega_1, \cdots, \omega_n$. Then X^* and S^* are primal and dual optimal, respectively, if and only if they commute and there is a permutation π of eigenvalues of S^* such that*

$$\lambda_i \omega_{\pi_i} = 0 \textit{ for } i = 1, \cdots, n.$$

Recall our convention that λ_i and ω_i are the i^{th} largest eigenvalues of X and S, respectively; this point necessitates the permutation π in the statement of the corollary.

Proof. X^* and S^* are optimal if and only if $X^*S^* = 0$. Thus, X^* and S^* commute with each other and therefore, they share a system of eigenvectors. Let columns of U be a joint system of orthonormal eigenvectors of X^* and S^*, i.e

$$X^* = U\mathrm{Diag}(\lambda_1, \cdots, \lambda_n)U^T \text{ and } S^* = U\mathrm{Diag}(\omega_{\pi_1}, \cdots, \omega_{\pi_n})U^T$$

for some permutation π. The corollary follows immediately by multiplying the right hand sides of these two identities. □

One can extend the notion of *strict complementarity* in linear programming to SDP. This can be stated by saying that in the preceding corollary *exactly* one of λ_i or ω_{π_i} corresponding to eigenvector $\mathbf{u}_i$ be zero for each $i = 1, \cdots, n$. Equivalently we may require that $\mathrm{Rank}(X^*) + \mathrm{Rank}(S^*) = n$. However, unlike standard linear programming, where in the absence of nondegeneracy one could say that precisely m components of the optimal solution $\mathbf{x}^*$ is nonzero, it is not clear in general how to predict $\mathrm{Rank}(X^*)$ or $\mathrm{Rank}(S^*)$ before solving the SDP problem. All we can say is that $\mathrm{Rank}(X^*) < n$ as the optimum of the primal SDP problem is attained on the boundary of the semidefinite cone. In section 4 we encounter another negative effect of the unpredictability of the rank of the optimal solution in the context of interior point methods. We should also mention the paper of G. Pataki which studies facial structure of feasible sets of SDP's and characterizes "degeneracy" in semidefinite programs [55]. Similar to linear programming, the complementary slackness theorem 2.10 may be used as a basis for primal-dual algorithms. Indeed in this paper, our interior point algorithm is a primal-dual method which maintains a primal feasible

[2] D. E. Knuth and an anonymous referee suggested the following slightly shorter proof: $0 = A \bullet B = \text{trace } A^{1/2}BA^{1/2}$. Since $A^{1/2}BA^{1/2} \succeq 0$ and its trace is zero, the matrix product itself must equal zero, and therefore $AB = 0$. It was pointed out to me that this lemma appears as an exercise in the text of Lancaster and Tismenetsky [39].

	MIN			MAX
	matrix or scalar, ≥ 0	$\longleftrightarrow$	C	matrix or scalar, $\leq$
V	matrix or scalar, ≤ 0	$\longleftrightarrow$	O	matrix or scalar, $\geq$
A	matrix, $\succeq 0$	$\longleftrightarrow$	N	matrix $\preceq$
R	matrix, $\preceq 0$	$\longleftrightarrow$	S	matrix, $\succeq$
	matrix or scalar, unrestricted	$\longleftrightarrow$	T	matrix or scalar, $=$
C	matrix or scalar, $\geq$	$\longleftrightarrow$		matrix or scalar, ≥ 0
O	matrix or scalar, $\leq$	$\longleftrightarrow$	V	matrix or scalar, ≤ 0
N	matrix, $\succeq$	$\longleftrightarrow$	A	matrix, $\succeq 0$
S	matrix, $\preceq$	$\longleftrightarrow$	R	matrix, $\preceq 0$
T	matrix or scalar, $=$	$\longleftrightarrow$		matrix or scalar, unrestricted

FIG. 2.1. *Rules for taking dual of a mixed SDP program. Variables in one program gives rise to constraints in another and vice-versa.*

X_k and dual feasible S_k and each iteration moves $X_k S_k$ closer to the zero matrix. The norm $\|X_k S_k\|$ is an indication of how close our current solution is to the optimum. In general the set of equations:

$$\begin{array}{lcl} \mathcal{A}\mathbf{vec}X & = & \mathbf{b} \\ \mathcal{A}^T\mathbf{y} + S & = & C \\ XS & = & 0 \end{array} \tag{2.5}$$

is a system of $n(n+1)+m$ equations in the same number of unknowns[3]. In the absence of degeneracy one can apply, for instance, Newton's method, or some quasi-Newton method to solve this system. Since SDP is a convex program, the real solutions of this system are global optima of the corresponding SDP problem.

As in linear programming, semidefinite programs may arise in a variety of forms; the standard form (2.1) is just one type. Sometimes we may have positive semidefinite constraints imposed on linear combinations of matrices (as in the dual problem in (2.1), for example). Sometimes we may have component-wise inequalities "$\geq$" on scalar or matrix variables in addition to Löwner inequalities. We may have several matrix expressions constrained to be positive semidefinite. Finally, we may have some or all of these. Of course, as in linear programming, it is possible to convert all such problems to the standard form, usually by introducing new scalar and matrix variables and new constraints. However, it is more convenient to apply duality directly, as with linear programs in general form. It is easy to show that the rules for obtaining the dual are a straightforward extension of these rules for the linear programming problem. The main addition is that constraints that involve semidefinite relations on matrix–valued expressions give rise to matrix–valued dual variables with semidefinite constraints. These rules are summarized in the table in figure 2.1; this table is a direct generalization of a similar table in the text of Bazaraa, Jarvis and Sherali [8].

3. An interior point algorithm. In this section we develop a potential reduction method for solving the primal problem so that, within $O(\sqrt{n}|\log\epsilon|)$ iterations, we get an approximate solution with at least ϵ relative accuracy, if ϵ is sufficiently small. Our development closely follows Ye's projective algorithm for linear programming [66]. Ye's complexity analysis is also extended to semidefinite programs.

3.1. Potential functions and projective transformations. First, recall that the interior of the cone of positive semidefinite matrices is the set of positive-definite

[3] Actually one can reduce the number of unknowns by writing $X = U\mathrm{Diag}(\mathbf{x})U^T$ and $S = U\mathrm{Diag}(\mathbf{s})U^T$ and requiring U to be an orthogonal matrix.

matrices; therefore, all interior points are nonsingular. The boundary of the cone consists of singular semidefinite matrices and so, some of the eigenvalues of the boundary matrices are zero. In particular, optimal solutions of the primal problem in (2.1) are singular.

We assume that there is a positive definite matrix X feasible for the primal and a positive definite matrix S feasible for the dual. Therefore the optimal solution for both primal and dual are finite and is attained on some feasible point. Later, in section 3.4, we show how to transform any primal-dual pair to an equivalent one where an initial interior primal–dual solution is available.

Let $q > 0$, and $\underline{z}$ be a given constant known to be a lower bound on the optimal value z^* of the primal problem in (2.1). Let X be an interior primal feasible matrix, $\mathbf{y}$ an interior dual feasible vector, and $S{:=}C - \sum_{i=1}^m y_i A_i$; thus, $X \succ 0$ and $S \succ 0$. Define the *primal potential function*:

$$\phi(X, \underline{z}) = q \ln(C \bullet X - \underline{z}) - \ln \det X, \tag{3.1}$$

and the *primal-dual potential function*:

$$\psi(X, S) = q \ln(X \bullet S) - \ln \det(XS). \tag{3.2}$$

For motivation, one may think of semidefinite constraints $X \succeq 0$ expressed as $\lambda_i(X) \geq 0$ for $i = 1, \cdots, n$. When the standard logarithmic barrier is applied to these constraints we get: $\sum_{i=1}^n \ln \lambda_i(X) = \ln \det X$.

The strategy of the algorithm is to generate a sequence of interior primal feasible matrices X_k, and a sequence of interior dual vector–matrix pairs $(\mathbf{y}_k, S_k)$, such that the sequence $\psi(X_k, S_k)$ decreases at least like an arithmetic progression. With an appropriate choice of q, this would imply that the duality gap $C \bullet X_k - \mathbf{b}^T \mathbf{y}_k$ decreases at least like a geometric progression with k; in particular it becomes a constant fraction of the original gap after $O(\sqrt{n})$ iterations.

Before describing the algorithm we state the following lemma which is a direct generalization of a similar lemma that appears in the analysis of most interior point linear programming methods. (Recall that $\rho(X)$ is the spectral radius of matrix X, which equals its largest eigenvalue when X is positive semidefinite.)

LEMMA 3.1. *Let X be a symmetric $n \times n$ matrix. If $0 \prec X \prec I$, then*

$$\ln \det X \geq \text{trace } X - n - \frac{\text{trace } (X - I)^2}{2[1 - \rho(X - I)]}.$$

Proof. In most interior-point linear programming algorithms it is shown that if $\|\mathbf{x} - \mathbf{1}\|_\infty < 1$ then

$$\sum_{j=1}^n \ln x_j \geq (\mathbf{1}^T \mathbf{x} - n) - \frac{\|\mathbf{x} - \mathbf{1}\|^2}{2(1 - \|\mathbf{x} - \mathbf{1}\|_\infty)}$$

which is easily proved by expanding $\ln x$, (see for example, Karmarkar [36] or Ye [67].) Now to prove the lemma simply substitute $\lambda_j(X)$ for x_j. □

We use a projective transformation to bring the current iterate to the center, except that the center here is the identity matrix (in contrast with linear programming in which the center is $\mathbf{1}$). An important point is that the transformation should map the set of symmetric matrices to itself. This is needed so that the transformed problem remains a meaningful SDP problem. Let $X_0 \succ 0$ be our current interior primal feasible point. To find a symmetry preserving projective transformation that maps X_0 to the identity matrix I, let L_0 be any $n \times n$ matrix such that $L_0 L_0^T = X_0$. There are infinitely many choices for L_0. For instance, it could be a Cholesky factor of X_0, or it could be its square root, $X_0^{1/2}$. We shall see shortly that it does not matter how we select L_0 as it will not affect the algorithm's behavior and performance. Fix integer r. Define $\mathcal{T} : \Re^{\frac{n \times n}{2}} \rightarrow \Re^{\frac{n \times n}{2}} \times \Re^r$, such that $\mathcal{T}(X) = (\overline{X}, \overline{\mathbf{x}})$. Then:

$$\overline{X} := \frac{(n+r) L_0^{-1} X L_0^{-T}}{r + X_0^{-1} \bullet X} \quad \text{and} \quad \overline{\mathbf{x}} := \left(\frac{n+r}{r + X_0^{-1} \bullet X} \right) \mathbf{1}. \tag{3.3}$$

Also, the inverse transformations is given by:

$$X = \mathcal{T}^{-1}(\overline{X}, \overline{\mathbf{x}}) := \frac{L_0 \overline{X} L_0^T}{\sum \overline{x}_j / r}. \tag{3.4}$$

Under $\mathcal{T}$, the primal SDP problem is transformed into

$$\begin{array}{ll} \min & \overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}} \\ \text{s.t.} & \overline{\mathcal{A}} \mathbf{vec}\, \overline{X} + \overline{A} \overline{\mathbf{x}} = \mathbf{0} \\ & \text{trace}\, \overline{X} + \mathbf{1}^T \overline{\mathbf{x}} = n + r \\ & \overline{X} \succeq 0 \\ & \overline{\mathbf{x}} \geq \mathbf{0}, \end{array} \tag{3.5}$$

where

$$\overline{C} := L_0^T C L_0 \tag{3.6}$$

$$\overline{\mathbf{c}}(\underline{z}) := -\, (\underline{z}/r)\, \mathbf{1} \tag{3.7}$$

$$\overline{A}_i := L_0^T A_i L_0 \tag{3.8}$$

$$\overline{\mathcal{A}} := \mathcal{A}(L_0 \otimes L_0) \tag{3.9}$$

$$\overline{A} := (-1/r)\, \mathbf{b} \mathbf{1}^T. \tag{3.10}$$

Note that $\overline{A}$ is an $m \times r$ rank one matrix. The transformed problem may be viewed as a mixed linear and semidefinite program. We may define the following primal potential function for the transformed problem:

$$\overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) := q \ln \left[\overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}} \right] - \ln \det \overline{X} - \sum_{j=1}^{r} \ln \overline{x}_j. \tag{3.11}$$

The following invariant property holds for the potential functions under projective transformations:

LEMMA 3.2. *If* $\overline{x}_1 = \cdots = \overline{x}_r$, *and* $q = n + r$ *and* $X := \mathcal{T}^{-1}(\overline{X}, \overline{\mathbf{x}})$ *then*

$$\phi(X, \underline{z}) - \phi(X_0, \underline{z}) = \overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) - \overline{\phi}(I, \mathbf{1}, \underline{z}). \tag{3.12}$$

Also, the following result is easily proved by expanding $\overline{\phi}$ and applying lemma 3.1; later we use it to prove the reduction in the primal-dual potential function.

COROLLARY 3.3. *For* $q = n + r$ *we have*

$$\overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) - \overline{\phi}(I, \mathbf{1}, \underline{z}) \leq (n+r) \ln \left(\frac{\overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}}}{\text{trace}\, \overline{C} + \overline{\mathbf{c}}(\underline{z})^T \mathbf{1}} \right) + \frac{\|\overline{X} - I\|^2 + \|\overline{\mathbf{x}} - \mathbf{1}\|^2}{2(1 - \|\overline{X} - I\| + \|\overline{\mathbf{x}} - \mathbf{1}\|)}.$$

3.2. A potential reduction algorithm. Similar to linear programming, in (3.5) we replace the inequality constraints $\overline{X} \succeq 0$ and $\overline{\mathbf{x}} \geq \mathbf{0}$ by an inscribed "ball" constraint, except that for the SDP problem the ball is centered at $(I, \mathbf{1})$. Therefore, (3.5) is replaced by the "ball optimization" problem:

$$\begin{array}{ll} \min & \overline{C} \bullet \overline{X} + \overline{\mathbf{c}}(\underline{z})^T \overline{\mathbf{x}} \\ \text{s.t.} & \overline{\mathcal{A}} \mathbf{vec}\, \overline{X} + \overline{A}\overline{\mathbf{x}} = \mathbf{0} \\ & \operatorname{trace} \overline{X} + \mathbf{1}^T \overline{\mathbf{x}} = n + r \\ & \|X - I\|^2 + \|\mathbf{x} - \mathbf{1}\|^2 \leq \beta^2 < 1 \end{array} \tag{3.13}$$

where β is a fixed constant between 0 and 1 to be determined shortly. Once we solve this problem and map the result back to the original space, we get a point that serves as a candidate for the next iterate. The solution of (3.13) is given by

$$\begin{pmatrix} \mathbf{vec}\overline{X}_1 \\ \overline{\mathbf{x}}_1 \end{pmatrix} := \begin{pmatrix} \mathbf{vec} I \\ \mathbf{1} \end{pmatrix} - \beta \frac{P(\underline{z})}{\|P(\underline{z})\|}, \tag{3.14}$$

and the candidate for the new primal iterate is given by:

$$X(\underline{z}) := \mathcal{T}^{-1}(\overline{X}_1, \overline{\mathbf{x}}_1), \tag{3.15}$$

where

$$P(\underline{z}) := \mathcal{P}_{\mathcal{A}'} \begin{pmatrix} \mathbf{vec}\overline{C} \\ \overline{\mathbf{c}}(\underline{z}) \end{pmatrix}, \quad \mathcal{A}' := \begin{pmatrix} \overline{\mathcal{A}} & \overline{A} \\ (\mathbf{vec}\, I)^T & \mathbf{1}^T \end{pmatrix}, \tag{3.16}$$

and $\mathcal{P}_{\mathcal{A}'}(\mathbf{u})$ is the projection of the $(n^2 + r)$-vector $\mathbf{u}$ to the null space of $\mathcal{A}'$. After expansion the projection $\mathcal{P}_{\mathcal{A}'}$ in (3.16) becomes:

$$P(\underline{z}) = \left(I - \frac{[\mathbf{vec}^T I | \mathbf{1}^T][\mathbf{vec}^T I | \mathbf{1}^T]^T}{n + r} \right) \left(I - [\overline{\mathcal{A}} | \overline{A}]^T ([\overline{\mathcal{A}} | \overline{A}][\overline{\mathcal{A}} | \overline{A}]^T)^{-1} [\overline{\mathcal{A}} | \overline{A}] \right) \begin{pmatrix} \mathbf{vec}\overline{C} \\ \overline{\mathbf{c}}(\underline{z}) \end{pmatrix}. \tag{3.17}$$

Define:

$$\begin{array}{rcl} \mathbf{y}(\underline{z}) & := & \left([\overline{\mathcal{A}}|\overline{A}][\overline{\mathcal{A}}|\overline{A}]^T\right)^{-1} [\overline{\mathcal{A}}|\overline{A}] \begin{pmatrix} \mathbf{vec}\overline{C} \\ \overline{\mathbf{c}}(\underline{z}) \end{pmatrix} \\ & = & \left(\mathcal{A}(X_0 \otimes X_0)\mathcal{A}^T + (1/r)\,\mathbf{b}\mathbf{b}^T\right)^{-1} [\mathcal{A}\mathbf{vec}(X_0 C X_0) + (\underline{z}/r)\,\mathbf{b}] \end{array} \tag{3.18}$$

and

$$S(\underline{z}) := C - \operatorname{Mat}(\mathcal{A}^T \mathbf{y}(\underline{z})). \tag{3.19}$$

$S(\underline{z})$ and $\mathbf{y}(\underline{z})$ serve as candidates for the new dual iterates. In terms of these quantities $P(\underline{z})$ may be written as:

$$P(\underline{z}) = \begin{pmatrix} \mathbf{vec}(L_0^T S(\underline{z}) L_0) \\ \frac{\mathbf{b}^T \mathbf{y}(\underline{z}) - \underline{z}}{r} \mathbf{1} \end{pmatrix} - \frac{C \bullet X_0 - \underline{z}}{n + r} \begin{pmatrix} \mathbf{vec} I \\ \mathbf{1} \end{pmatrix}. \tag{3.20}$$

Observe that $X(\underline{z})$, $S(\underline{z})$ and $\mathbf{y}(\underline{z})$ are all independent of L_0; in fact in actual computation we do not need to have L_0 explicitly.

The main result to be proved is that first, at least one of the following holds:

1. Either $X(\underline{z}) \succ 0$ and thus primal feasible, or

2. $S(\underline{z}) \succ 0$ and therefore $(\mathbf{y}(\underline{z}), S(\underline{z}))$ is dual feasible

Furthermore choosing either one of the feasible candidates reduces the value of the primal-dual potential function ψ by a constant amount. First observe that $\mathcal{P}_{\mathcal{A}'}$ is a projector, that is $\mathcal{P}_{\mathcal{A}'}^2 = \mathcal{P}_{\mathcal{A}'}$. Therefore, from (3.14) we get:

$$\overline{C} \bullet (\overline{X}_1 - I) + \overline{\mathbf{c}}(\underline{z})^T(\overline{\mathbf{x}} - \mathbf{1}) = -\beta\|P(\underline{z})\|$$

Hence, noting that $\ln(1+x) \le x$, for nonnegative x, corollary 3.3 implies:

COROLLARY 3.4. *Let $q = n + r$ and $\overline{X}_1$ and $\overline{\mathbf{x}}$ be as in* (3.14). *Then*

$$\overline{\phi}(\overline{X}, \overline{\mathbf{x}}, \underline{z}) - \overline{\phi}(I, \mathbf{1}, \underline{z}) \le -(n+r)\beta \frac{\|P(\underline{z})\|}{\mathbf{c}(\underline{z})^T\mathbf{1} + \operatorname{trace} \overline{C}} + \frac{\beta^2}{2(1-\beta)}.$$

Let Δ_0 be the size of the duality gap in the current iterate, that is

$$\Delta_0 := C \bullet X_0 - \underline{z}$$

and let

$$\Delta_1 := S(\underline{z}) \bullet X_0 = C \bullet X_0 - \mathbf{b}^T\mathbf{y}(\underline{z}).$$

Thus Δ_1 should be interpreted as the value of the duality gap if we choose $\mathbf{y}(\underline{z})$ as our new dual iterate. Before deriving the amount of reduction in the potential function we prove the following lemma:

LEMMA 3.5. *If there is some real number α with $0 < \alpha < 1$, such that*

$$\|P(\underline{z})\| \le \alpha \frac{\Delta_0}{n+r}$$

then $S(\underline{z}) \succ 0$, and $\mathbf{b}^T\mathbf{y}(\underline{z}) > \underline{z}$. Furthermore,

$$\left\| L_0^T S(\underline{z}) L_0 - \frac{\Delta_1}{n} I \right\| \le \frac{\Delta_1}{n}\alpha\sqrt{\frac{n + n^2/r}{n + n^2/r - \alpha^2}}, \tag{3.21}$$

and

$$\left| \frac{n+r}{n}\frac{\Delta_1}{\Delta_0} - 1 \right| \le \frac{\alpha}{\sqrt{n + n^2/r}}. \tag{3.22}$$

Proof. Suppose $S(\underline{z}) \not\succ 0$. Then $L_0^T S(\underline{z}) L_0$ is not positive definite and so some of its eigenvalues are less than or equal to 0. Thus, from (3.20) we have

$$\|P(\underline{z})\| \ge \rho\left(\frac{\Delta_0}{n+r} I - L_0^T S(\underline{z}) L_0\right) \ge \frac{\Delta_0}{n+r},$$

a contradiction. Also, If $\mathbf{b}^T\mathbf{y}(\underline{z}) \le \underline{z}$ then from (3.20) we have

$$\|P(\underline{z})\| \ge \frac{\Delta_0}{n+r} - \frac{\mathbf{b}^T\mathbf{y}(\underline{z}) - \underline{z}}{r} \ge \frac{\Delta_0}{n+r},$$

which is again a contradiction. Now from (3.20) we have

$$P(\underline{z}) = \begin{pmatrix} \left[\mathbf{vec}(L_0^T S(\underline{z}) L_0) - \frac{\Delta_1}{n} I\right] - \left[\frac{\Delta_0}{n+r} - \frac{\Delta_1}{n}\right] \mathbf{vec} I \\ \left[\frac{\Delta_0 - \Delta_1}{r} - \frac{\Delta_0}{n+r}\right] \mathbf{1} \end{pmatrix}.$$

Since $I \bullet [(L_0^T S(\underline{z}) L_0) - (\Delta_1/n)I] = 0$, we have

$$\begin{aligned} \|P(\underline{z})\|^2 &= \left\| L_0^T S(\underline{z}) L_0 - \frac{\Delta_1}{n} I \right\|^2 + n\left(\frac{\Delta_0}{n+r} - \frac{\Delta_1}{n}\right)^2 + r\left(\frac{\Delta_0 - \Delta_1}{r} - \frac{\Delta_0}{n+r}\right)^2 \\ &= \left\| L_0^T S(\underline{z}) L_0 - \frac{\Delta_1}{n} I \right\|^2 + \left(n + \frac{n^2}{r}\right)\left(\frac{\Delta_1}{n} - \frac{\Delta_0}{n+r}\right)^2. \qquad (3.23) \end{aligned}$$

If (3.21) is false then from (3.23) we have

$$\begin{aligned} \|P(\underline{z})\|^2 &> \left(\frac{\Delta_1}{n}\right)^2 \alpha^2 \frac{n + n^2/r}{n + n^2/r - \alpha^2} + \left(n + \frac{n^2}{r}\right)\left(\frac{\Delta_1}{n} - \frac{\Delta_0}{n+r}\right)^2 \\ &\geq \alpha^2 \left(\frac{\Delta_0}{n+r}\right)^2. \qquad (3.24) \end{aligned}$$

(The last inequality is proved by taking the right hand side of the first inequality as a quadratic function in Δ_1/n and minimizing it.) But (3.24) contradicts the assumption of the lemma, so (3.21) must be true. Finally, since (3.24) is false, we have:

$$\left(n + \frac{n^2}{r}\right)\left(\frac{\Delta_1}{n} - \frac{\Delta_0}{n+r}\right)^2 \leq \alpha^2 \left(\frac{\Delta_0}{n+r}\right)^2,$$

from which (3.22) follows. □

Now we may prove the potential reduction theorem.

THEOREM 3.6. *Let X_0 be any interior feasible matrix for the primal problem (2.1) and $\mathbf{y}_0$ interior feasible for the dual. Let also, $r{:=}\lceil\sqrt{n}\,\rceil$ and $q{:=}n+r$, $S_0{:=}C - \sum_{i=1}^m y_i A_i$, $\underline{z}_0{:=}\mathbf{b}^T\mathbf{y}_0$, $X(\underline{z}){:=}\mathcal{T}^{-1}(\overline{X}_1, \overline{\mathbf{x}}_1)$, as in (3.4), $\mathbf{y}_1{:=}\mathbf{y}(\underline{z}_0)$, and $S_1{:=}S(\underline{z}_0)$. Then there exist an absolute constant δ such that either*

$$\psi(X(\underline{z}), S_0) \leq \psi(X_0, S_0) - \delta,$$

or

$$\psi(X_0, S_1) \leq \psi(X_0, S_0) - \delta,$$

Furthermore, if we set $\alpha = 0.55$ and $\beta{:=}0.3$, then $\delta > 0.1$.

Proof. If for some constant $0 < \alpha < 1$

$$\|P(\underline{z})\| \geq \alpha \frac{\Delta_0}{n+r}$$

then

$$\begin{aligned} \psi(X(\underline{z}), S_0) - \psi(X_0, S_0) &= \phi(X(\underline{z}), \underline{z}_0) - \phi(X_0, \underline{z}_0) \\ &= \overline{\phi}(\overline{X}(\underline{z}), \overline{\mathbf{x}}_1, \underline{z}_0) - \overline{\phi}(I, \mathbf{1}, \underline{z}_0) \\ &\leq -\beta\alpha + \frac{\beta^2}{2(1-\beta)} \end{aligned}$$

(the last inequality is true by corollary 3.4). Otherwise, the conditions of lemma 3.5 are satisfied. Also applying lemma 3.1 to $(n/\Delta_1)L_0^T S_1 L_0$, and setting $\gamma{:=}\alpha\sqrt{\frac{n+n^2/r}{n+n^2/r-\alpha^2}}$

we have:

$$\begin{aligned} n \ln X_0 \bullet S_1 - \ln\det X_0 S_1 &= n \ln\left(\frac{nX_0 \bullet S_1}{\Delta_1}\right) - \ln\det \frac{nX_0S_1}{\Delta_1} \\ &= n\ln n - \ln\det\frac{nX_0S_1}{\Delta_1} \\ &\leq n\ln n + \frac{\|nL_0^T S_1 L_0/\Delta_1 - I\|^2}{2(1-\|nL_0^T S_1 L_0/\Delta_1 - I\|)} \\ &\leq n\ln X_0 \bullet S_0 - \ln\det X_0S_0 + \frac{\gamma^2}{2(1-\gamma)}, \end{aligned} \tag{3.25}$$

where the last relation results from applying the arithmetic-geometric mean inequality to the eigenvalues of X_0S_0 (which are all real.) By (3.22) of lemma 3.5 we have

$$\Delta_1 < \left(1 - \frac{r}{n+r} - \frac{n}{n+r}\frac{\alpha}{\sqrt{n+n^2/r}}\right)\Delta_0.$$

Thus,

$$r\ln\frac{X_0 \bullet S_1}{X_0 \bullet S_0} = r\ln\frac{\Delta_1}{\Delta_0} \leq \frac{r^2}{n+r}\left(-1 + \frac{\alpha}{\sqrt{r+r^2/n}}\right). \tag{3.26}$$

Adding (3.25) and (3.26) we get

$$\psi(X_0,S_1) - \psi(X_0,S_0) \leq \frac{r^2}{n+r}\left(-1 + \frac{\alpha}{\sqrt{r+r^2/n}}\right) + \frac{\gamma^2}{2(1-\gamma)}. \tag{3.27}$$

It is easily verified that choice of $\alpha = 0.55$, $\beta = 0.3$ and $\delta = 0.1$ is consistent with all the conditions of the theorem. □

Based on this result we present the projective version of the algorithm displayed in figure 3.1. Note that in this algorithm β^* and $\underline{z}^*$ are obtained by line search on the potential function. We justify this in the next subsection. Also it should be realized that this algorithm is only a prototype and in a practical implementation one must apply substantial simplifications to eliminate redundant use of storage and algebraic operations, especially regarding symmetric matrices.

The following theorem now shows that using algorithm 3.1, one can get the duality gap to less than ϵ in a number of iterations k which is dependent on $|\ln\epsilon|$, $\sqrt{n}$, and the value of the potential function at the initial solution.

THEOREM 3.7. *Let X_0, $\mathbf{y}_0$ and $S_0{:=}C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0)$ be given initial interior points for the primal and dual semidefinite programming problems in (2.1). Let also that $r = \lceil\sqrt{n}\,\rceil$ and $q = n + r$ in the primal-dual potential function ψ, and assume that $\psi(X_0,S_0) \leq O(\sqrt{n}E)$ for some constant E. If an algorithm generates a sequence of interior primal and dual points X_j, $\mathbf{y}_j$ (and thus S_j) such that $\psi(X_j,S_j) \geq \psi(X_{j+1},S_{j+1}) + \delta$ for some fixed number δ then, after $k = O(\sqrt{n}|\log\epsilon|)$ iterations, for primal and dual solutions X_k, $\mathbf{y}_k$ and S_k we have*

$$C \bullet X_k - \mathbf{b}^T\mathbf{y}_k < 2^E\epsilon.$$

Algorithm SDP:
Input:
An $n \times n$ matrix X_0, interior feasible for the primal problem in (2.1);
an m-vector $\mathbf{y}_0$ interior feasible for the dual problem;
a constant $\epsilon > 0$.
Output:
A primal feasible solution X and dual feasible solution $\mathbf{y}$ such that
$C \bullet X - \mathbf{b}^T\mathbf{y} < \epsilon$.
Method:
1) Set $k = 0$ and $\alpha = 0.55$.
2) Set $\underline{z}_0 = \mathbf{b}^T\mathbf{y}_0$.
3) Set $S_k := C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0)$.
4) While $C \bullet X_k - \mathbf{b}^T\mathbf{y}_k \geq \epsilon$ do
begin
Compute $S(\underline{z}_k)$ from (3.19) and $P(\underline{z}_k)$ from (3.16).
If $\|P(\underline{z}_k)\| \geq \alpha(C \bullet X_k - \underline{z}_k)/(n+r)$ then
a)Find $\beta^* := \mathrm{argmin}_{0\leq\beta\leq 1}\psi(X_k - \beta L_k P(\underline{z}_k)L_k^T, S_k)$,
using a line search procedure.
b) Set $(\overline{X}_{k+1}, \overline{\mathbf{x}}_{k+1}) = (I, \mathbf{1}) - \beta^* P(\underline{z})$,
and set $X_{k+1} := \mathcal{T}^{-1}(\overline{X}_{k+1}, \overline{\mathbf{x}}_{k+1})$.
c) Set $S_{k+1} := S_k$, and $\underline{z}_{k+1} := \underline{z}_k$.
Else
d) Find $\underline{z}^* := \mathrm{argmin}_{\underline{z}\leq\underline{z}_k}\psi(X_k, S(\underline{z}))$ by a line search.
e) Set $S_{k+1} = S(\underline{z}^*)$.
f) Set $X_{k+1} = X_k$, and $\underline{z}_{k+1} = \mathbf{b}^T\mathbf{y}(\underline{z}^*)$.
Set $k = k+1$.
end.

Fig. 3.1. *A projective potential reduction algorithm.*

Proof. Each iteration reduces the potential function by at least δ. Thus, if $\psi(X_0, S_0) < O(\sqrt{n}E)$ then after $O(\sqrt{n}|\log\epsilon|)$ iterations we have:

$$\begin{aligned}\psi(X_k, S_k) &< \left(\sqrt{n}E - \sqrt{n}|\log\epsilon|\right)\\ &= (\sqrt{n}[\log 2^E - |\log\epsilon|]\\ &\leq \sqrt{n}|\log(2^E\epsilon)|.\end{aligned}$$

Therefore,

$$\begin{aligned}\sqrt{n}\ln X_k \bullet S_k &< -n\ln X_k \bullet S_k + \ln\det X_k S_k + \sqrt{n}|\log(2^E\epsilon)|\\ &< -n\ln n + \sqrt{n}|\ln(2^E\epsilon)|.\end{aligned}$$

The last inequality comes from applying the arithmetic-geometric inequality to the eigenvalues of $X_k S_k$, which are real, as both matrices are positive definite. Thus, $\ln X_k \bullet S_k < |\log(2^E\epsilon)|$, and since $X_k \bullet S_k = C \bullet X_k - \mathbf{b}^T\mathbf{y}_k$, the theorem follows. □

In other words if we start our potential reduction algorithm at a pair $(X_0, \mathbf{y}_0)$ with $\psi(X_0, S_0) = \sqrt{n}E)$, then after $O(\sqrt{n}(E + |\log\epsilon|))$ iterations we will have a solution with duality gap less than ϵ. Therefore for all $\epsilon < 2^{-E}$ the term $|\log\epsilon|$ dominates E and so the number of iterations is bounded by $O(\sqrt{n}|\log\epsilon|)$. Also observe that this proof solely depends on the reduction of the potential function ψ. We have to guarantee a reduction of at least δ in each iteration; but larger reductions may speed up the algorithm without affecting its worst case complexity. Therefore, in steps 4a and 4d of the algorithm in figure 3.1 we allow a line search to find a step length β^* and $\underline{z}^*$ which maximizes the reduction in the potential function.

3.3. Feasibility, boundedness and polynomial–time computability. To complete our analysis we must study feasibility of the SDP problem and bounds on the norms of the optimal primal and dual solutions. The situation is somewhat different from linear programming. First, let us assume that all entries in the primal and dual problems (2.1) are integers. In contrast with linear programming, the optimal solution of (2.1) is not necessarily a rational number. Therefore we need to specify an error tolerance, ϵ, and ask for a pair of primal and dual solutions X and S such that the duality gap $X \bullet S \leq \epsilon$.[4] If ϵ is also a rational number, define L, the *size* of the SDP problem, as the number of bits in the binary representation of ϵ and entries of C, $\mathcal{A}$, and $\mathbf{b}$, see [32] for complete definition of "size" of a problem. One might expect that the interior point method developed in the previous sections lead to an algorithm which runs in time polynomial in m, n and L. However, this is not true in general as the solution itself may be exponentially large. To see this consider the optimization problem:

$$\min\{x_n : \ x_1 \geq 2, \ \text{and } x_i \geq x_{i-1}^2 \text{ for } i = 2, \cdots, n\} \tag{3.28}$$

Clearly, $x_i = 2^{2^{i-1}}$ for $i = 1, \cdots, n$ is the solution of this problem which requires exponential number of bits. Now (3.28) can be written as the following semidefinite program:

$$\begin{array}{ll} \min & x_n \\ \text{s.t.} & x_1 \geq 2 \\ & \begin{pmatrix} x_i & x_{i-1} \\ x_{i-1} & 1 \end{pmatrix} \succeq 0 \quad \text{for } i = 2, \cdots, n. \end{array}$$

This SDP problem can be easily turned into a standard form SDP whose input size (taking $\epsilon = 1$, say) is polynomial in n and whose output requires exponential number of bits. Therefore no algorithm can solve it in polynomial time[5].

In many cases, including all of the combinatorial optimization problems described below, one may be able to put an *a priori* bound on the norms of the optimal solutions. For instance we may be able to prove that the feasible sets of both the primal and dual solutions are inside balls of radius, say R (a rational number) centered at the origin. In such cases we can show that the interior point algorithm developed earlier can produce in polynomial time (in $|\ln \epsilon|$, L and $\ln R$) primal and dual solutions whose duality gap is smaller than ϵ. Notice that in the ellipsoid method such an *a priori* bound is assumed by requiring that the initial ellipsoid be a ball of radius R centered at the origin. Let L' be the number of bits in the binary expansion of some R. Then, similar to linear programming, one can always transform the pair of primal and dual problems (2.1) to another pair for which initial interior feasible points are readily available. We extend the construction suggested by Kojima, Mizuno and Yoshise in [38] which in turn is based on Megiddo's [43].

---

[4] Since X, S and $\mathbf{y}$ are solution of the algebraic system of equations: $XS = 0, \mathcal{A}\mathbf{vec}X = \mathbf{b}$ and $\mathcal{A}^T\mathbf{y} + S = C$, there are algebraic solutions among all optimal solutions of an SDP problem with integral input.

[5] I am indebted to Motakuri Ramana for bringing to my attention an error in [1, 2] where I had claimed that the norm of the solution to any SDP problem is bounded by 2^L. Ramana essentially provided this counter example.

Consider the following pair of primal and dual problems:

$$(3.29)\qquad \begin{array}{ll} \min & C \bullet X + M x_1 \\ \text{s.t.} & \mathcal{A}\mathbf{vec}(X) + [\mathbf{b} - \mathcal{A}\mathbf{vec}(X_0)]x_1 = \mathbf{b} \\ & [\mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0) + S_0 - C] \bullet X + x_2 = N \\ & X \succeq 0 \\ & x_1, x_2 \geq 0 \end{array}$$

and,

$$(3.30)\qquad \begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} - N y_1 \\ \text{s.t.} & \mathrm{Mat}(\mathcal{A}^T\mathbf{y}) + S + [C - \mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0) - S_0]y_1 = C \\ & [\mathbf{b} - \mathcal{A}\mathbf{vec}(X_0)]^T\mathbf{y} + y_2 = M \\ & S \succeq 0 \\ & y_1, y_2 \geq 0 \end{array}$$

where X_0 and S_0 are arbitrary positive definite $n \times n$ matrices, $\mathbf{y}_0$ an arbitrary m-vector, and M and N are large enough positive numbers to ensure that $y_2 \geq 0$ and $x_2 \geq 0$. Clearly $X{:=}X_0$, $x_1{:=}1$ and $x_2{:=}N - (\mathrm{Mat}(\mathcal{A}^T\mathbf{y}_0) + S_0 - C) \bullet X_0$ are interior feasible for the primal (3.29) (with large enough N); and $S{:=}S_0$, $\mathbf{y}{:=}\mathbf{y}_0$, $y_1{:=}1$, and $y_2{:=}M - (\mathbf{b} - \mathcal{A}\mathbf{vec}(X_0))^T\mathbf{y}_0$ are interior feasible for the dual problem in (3.30) (for large enough M). By choosing $X_0 = S_0 = I$, $x_1 = y_1 = 1$, $\mathbf{y}_0 = \mathbf{0}$, it suffices to choose M and N such that

$$\begin{aligned} N &> \max\left(n - \sum_i C_{ii},\ \sum_i X^*_{ii} - C \bullet X^*\right), \\ M &> \max\left(0,\ \mathbf{b}^T\mathbf{y}^* - \sum_i y^*_i \mathrm{trace}\ (A_i)\right). \end{aligned}$$

For instance we may set $N = M = 2^{L+L'}$. It is easy to see that if the optimal value of x_1 is not zero, then the original primal is infeasible (the proof is exactly like the one given in Kojima et al. in [38]). Similarly if the optimal value of y_1 is not zero, then the original dual is infeasible. Otherwise, the optimal X^* and $\mathbf{y}^*$ are also optimal for the original primal and dual problems, respectively. Furthermore, It is easily verified that the value of the primal-dual potential function ψ at the initial point is bounded by $O(\sqrt{n}(L + L'))$. So, for the general SDP problem, any algorithm that reduces the primal-dual potential function ψ by a constant amount may find, in $O(\sqrt{n}\max(L, L', |\log\epsilon|))$ iterations, a pair of primal and dual feasible solutions whose duality gap is less than ϵ; if $\epsilon < 2^{-L-L'}$, then the number of iterations is bounded by $O(\sqrt{n}|\ln\epsilon|)$.

3.4. A correspondence between proofs in linear and semidefinite programming. The remarkable similarity between the algorithm presented here and Ye's LP algorithm in [66] suggests that other LP interior point methods may also be extended to SDP problems. Proofs of convergence and polynomial-time complexity may be extended as well. The correspondence is summarized in figure 3.2. Given an interior point algorithm for linear programming we may construct, in a mechanical way, an algorithm for the SDP problem by replacing any references to the entries under the LP column, with the corresponding entry under the SDP column. Proofs of convergence or polynomial time complexity may also be extended mechanically in the

LP	SDP
unknown vector: $\mathbf{x}$	unknown symmetric matrix: X
inequality constraints: $\geq$	Löwner constraints: $\succeq$
dual variable: $\mathbf{y}$	dual variable: $\mathbf{y}$
dual slack vector: $\mathbf{s}$	dual slack symmetric matrix: S
$\mathbf{1}$	I
linear scaling: $\mathbf{x} \rightarrow (x_i/(\mathbf{x}_0)_i)_{i=1}^n = [\mathrm{Diag}(\mathbf{x}_0)]^{-1}\mathbf{x}$	linear scaling: $X \rightarrow L_0^{-1}XL_0^{-T} = \mathrm{Mat}[(L_0^{-1} \otimes L_0^{-1})\mathbf{vec}(X)]$
projective scaling: $\mathbf{x} \rightarrow \frac{c_1[\mathrm{Diag}(\mathbf{x}_0)]^{-1}\mathbf{x}}{c_2+\mathbf{1}^T[\mathrm{Diag}(\mathbf{x}_0)]^{-1}\mathbf{x}}$	projective scaling: $X \rightarrow \frac{c_1 L_0^{-1}XL_0^{-T}}{c_2+\mathrm{trace}\ L_0^{-1}XL_0^{-T}}$
barrier function: $\sum \ln x_i$	barrier function: $\ln\det X$
norms: $\|\mathbf{x}\|$, $\|\mathbf{x}\|_\infty$, $\|\mathbf{x}\|_p$	norms: $\|X\|$, $\|X\|_2$, $(\sum \lvert\lambda_i(X)\rvert^p)^{1/p}$

FIG. 3.2. *Correspondence between linear programming and semidefinite programming*

same manner. We have already verified this claim on the approaches of Gonzaga [28] and Ye [67] (see [2]). Extension to primal-dual methods such as Monteiro and Adler [44] is more challenging. This table itself may be summarized by the following rule: In any linear programming algorithm, replace any implicit or explicit reference to x_i (or s_i) by a reference to $\lambda_i(X)$ (or $\lambda_i(S)$). Furthermore, in any scaling, replace affine or projective transformations by corresponding *symmetry preserving* transformation on matrices. Notice that these same rules were implicitly used to derive various duality and complementary slackness theorems for SDP from the corresponding theorems for LP.

Similar techniques may be applied to more general problems. For instance one can define a semidefinite analog of convex quadratic programming or, more generally, a semidefinite analog of the linear complementarity problem. Also, one can study a semidefinite analog of linear fractional programs. For the linear version of all these problems interior point methods have been published (see [45], [37], and [4], for example) and one can apply the conversion rules mentioned above to obtain interior point methods for their semidefinite variants. Details are omitted here.

3.5. Differences between SDP and LP interior point algorithms. Thus far, we have emphasized the similarity of linear and semidefinite interior point methods. There are however, important distinctions and some favorable circumstances in LP do not extend to SDP. We have already seen the differences between LP and SDP when we studied irrationality and *a priori* bounds on the number of bits in the optimal solutions. We list other distinctions which must be studied carefully before a serious practical implementation of interior point SDP algorithms is attempted.

1. In the absence of degeneracy one can predict that precisely m entries of the optimal vector $\mathbf{x}^*$ are nonzero in the standard linear program with coefficient matrix $A \in \Re^{m\times n}$. Recall that in each iteration of a primal interior point algorithm, the main computational effort is in obtaining $(A\ \mathrm{Diag}(\mathbf{x})^2\ A^T)^{-1}\mathbf{v}$, where $\mathbf{v}$ is some vector. Therefore, if A is of rank m and reasonably well-conditioned, this computation is fairly straightforward and typically no numerical difficulties should arise. In SDP however, even if we assume strict complementarity, (i.e $\mathrm{Rank}(X^*) + \mathrm{Rank}(S^*) = n$,) we still do not know what $\mathrm{Rank}(X^*)$ is going to be before solving the SDP problem. Furthermore, let $\mathrm{Rank}(X^*) = r$. Since the main computational work in SDP interior

point methods is computing $(\mathcal{A}(X \otimes X)\mathcal{A}^T)^{-1}\mathbf{v}$, even if $\mathcal{A}$ is full rank and reasonably well-conditioned, $\mathcal{A}(X \otimes X)\mathcal{A}^T$ may converge to a singular matrix unless $m \leq r^2$, which is not guaranteed. The same issue arises if we use dual or primal-dual interior point algorithms.

2. The main reason that interior point methods in linear programming are practically competitive–aside from the small number of iterations–is that if the matrix AA^T is sparse, so is ADA^T for any diagonal matrix D; in fact, ADA^T and AA^T have precisely the same nonzero structure. Therefore, once a good order of elimination is obtained for AA^T, the same order works for all subsequent iterations of the interior point algorithm. This is is not the case for SDP. In general even if $\mathcal{A}\mathcal{A}^T$ is sparse the matrix $\mathcal{A}(X \otimes X)\mathcal{A}^T$ may not be sparse at all. It is not clear how factorization of $\mathcal{A}(X_k \otimes X_k)\mathcal{A}^T$ could be of any use in factoring $\mathcal{A}(X_{k+1} \otimes X_{k+1})\mathcal{A}^T$.

3. Karmarkar in [36] gives a nice amortized method for updating factors of ADA^T. He develops a technique where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ differ only in j_k entries where $\sum j_k$ over all iterations is bounded by $O(\sqrt{n})$. From this observation he manages to reduce the overall number of operations by a factor of $\sqrt{n}$. It is not clear how one extends Karmarkar's amortized scheme to SDP interior point algorithms. (See [49] for some progress in this direction.)

4. Eigenvalues as semidefinite programs. In many cases semidefinite programs arise in the form of minimizing or maximizing an appropriate linear combination of eigenvalues of a symmetric matrix subject to linear constraints on the matrix. In this section we study problems of this form, and show that under proper assumptions they are indeed special case semidefinite programs. We give primal and dual characterization of each problem and examine the complementary slackness theorem as specialized to that problem.

4.1. Minimizing sum of the first few eigenvalues. First we consider minimizing sum of the first k eigenvalues of a symmetric matrix subject to linear constraints on the matrix. We consider two variations, namely

$$\min\{\lambda_1(X) + \cdots + \lambda_k(X) : \mathcal{A}\mathbf{vec}X = \mathbf{b}\} \tag{4.1}$$

and

$$\min \sum_{i=1}^{k} \lambda_i(A(\mathbf{x})) \text{ where } A(\mathbf{x}) = A_0 + \sum_{i=1}^{m} x_i A_i. \tag{4.2}$$

To show that these problems are indeed semidefinite programs, we use the following elegant characterization by Overton and Womersley [54, 53].

THEOREM 4.1. *For the sum of the first k eigenvalues of a symmetric matrix A the following semidefinite programming characterization holds:*

$$\begin{array}{rcll} \lambda_1(A) + \cdots + \lambda_k(A) & = & \max & A \bullet U \\ & & \text{s.t.} & \operatorname{trace} U = k \\ & & & 0 \preceq U \preceq I. \end{array} \tag{4.3}$$

It is worth mentioning that this result is based on a beautiful convex hull characterization which was known at least as early as 1971, see [22], but unfortunately has remained somewhat obscure. Here is the statement of this result:

LEMMA 4.2. *Let*

$$S_1 {:=} \{YY^T : Y \in \Re^{n\times k},\ Y^TY = I\}$$

and

$$S_2 := \{W : \; W = W^T, \; \text{trace}\; W = k, \; 0 \preceq W \preceq I\}.$$

Then

$$\text{conv}\; S_1 = S_2,$$

and S_1 is exactly the set of extreme points of S_2.

For an historical account of this result, its connection to the well-known, but computationally less useful theorem of K. Fan, and interesting connections to the theorem of Birkhoff and Von Neumann concerning the convex hull of doubly stochastic matrices, refer to Overton and Womersley [53].

In order to express (4.1) as a semidefinite program we first derive a dual characterization of sum of the first k eigenvalues of A, by finding a dual version of Theorem 4.1. Let us see how applying the rules of figure (2.1) to (4.3) can aid us in finding such dual characterization. The constraint $U \preceq I$ gives rise to dual variable V, which by the 3rd line of the table in figure (2.1) must satisfy $V \succeq 0$. The variable $U \succeq 0$, by the eight line of the table (2.1), gives rise to the constraint $zI + V \succeq A$. Thus we have:

THEOREM 4.3. *For the sum of the first k eigenvalues of a symmetric matrix A the following semidefinite programming characterization holds:*

$$\begin{array}{rcll} \lambda_1(A) + \cdots + \lambda_k(A) & = & \min & kz + \text{trace}\; V \\ & & \text{s.t.} & zI + V \succeq A \\ & & & V \succeq 0. \end{array} \tag{4.4}$$

Now, it is easy to incorporate the equality constraints into (4.4) by replacing A with X. So the optimization problem (4.1) is equivalent to

$$\begin{array}{ll} \min & kz + \text{trace}\; V \\ \text{s.t.} & \mathcal{A}\mathbf{vec}X = \mathbf{b} \\ & zI + V - X \succeq 0 \\ & V \succeq 0, \end{array} \tag{4.5}$$

and taking the dual again we have the following dual characterization:

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & U = \text{Mat}(\mathcal{A}^T\mathbf{y}) \\ & \text{trace}\; U = k \\ & 0 \preceq U \preceq I. \end{array} \tag{4.6}$$

The complementary slackness result for primal feasible z^*, X^*, and V^*, and dual feasible U^* states that these are optimal if and only if

$$(z^*I + V^* - X^*)U^* = (I - U^*)V^* = 0$$

Similarly (4.2) may be expressed by the following primal and dual pair:

$$\begin{array}{llll} \min & kz + \text{trace}\; V & \max & A_0 \bullet Y \\ \text{s.t.} & zI + V - \sum x_i A_i \succeq A_0 & \text{s.t.} & \text{trace}\; Y = k \\ & V \succeq 0 & & A_i \bullet Y = 0 \text{ for } i = 1, \cdots, m \\ & & & 0 \preceq Y \preceq I. \end{array} \tag{4.7}$$

When $k = 1$, these characterizations become simpler, because in that case the constraint $Y \preceq I$ (and thus variable V) are redundant. Therefore, the problem

$$\min\{\lambda_1(X) : \mathcal{A}\mathbf{vec}X = \mathbf{b}\}$$

may be expressed as the solution of the primal and dual SDP pair:

$$\begin{array}{llll} \min & z & \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & zI - X \succeq 0 & \text{s.t.} & \text{trace } \operatorname{Mat}(\mathcal{A}^T\mathbf{y}) = 1 \\ & \mathcal{A}\mathbf{vec}X = \mathbf{b} & & \operatorname{Mat}(\mathcal{A}^T\mathbf{y}) \succeq 0 \end{array} \tag{4.8}$$

and the complementary slackness theorem indicates that for X^* and $\mathbf{y}^*$ to be primal and dual optimum solution for (4.8), in addition to being primal and dual feasible they must satisfy:

$$\operatorname{Mat}(\mathcal{A}^T\mathbf{y}^*)(\lambda_1(X^*)I - X^*) = 0.$$

4.2. Minimizing weighted sums of eigenvalues. In this section we consider the weighted sum of eigenvalues of a matrix. Let $m_1 \geq m_2 \geq \cdots \geq m_k > m_{k+1} = 0$ be a set of fixed real numbers. We are interested in the following problem:

$$\min\{m_1\lambda_1(X) + \cdots + m_k\lambda_k(X) : \mathcal{A}\mathbf{vec}X = \mathbf{b}\}. \tag{4.9}$$

Note that without the condition $m_1 \geq m_2 \geq \cdots \geq m_k > 0$ (4.9) is not necessarily a convex program. To formulate this problem as a semidefinite program, we use a technique originally employed by Donath and Hoffman in [18]. They rewrote the sum as follows:

$$\begin{aligned} m_1\lambda_1(A) + m_2\lambda_2(A) + \cdots + m_k\lambda_k(A) \;=\; & (m_1 - m_2)\lambda_1(A) + \\ & (m_2 - m_3)[\lambda_1(A) + \lambda_2(A)] + \cdots + \\ & (m_{k-1} - m_k)[\lambda_1(A) + \cdots + \lambda_{k-1}(A)] + \\ & m_k[\lambda_1(A) + \cdots + \lambda_k(A)] \end{aligned} \tag{4.10}$$

and observed that the right hand side of (4.10) is a nonnegative combination of convex functions, and therefore, itself is convex. This formulation also allows us to write (4.9) as a semidefinite programming problem. For each of the partial sums of eigenvalues in (4.10) we may use the SDP formulation of the last subsection and obtain the primal:

$$\begin{array}{ll} \min & \sum_{i=1}^k iz_i + \sum_{i=1}^k \text{trace } V_i \\ \text{s.t.} & z_iI + V_i - (m_i - m_{i+1})X \succeq 0 \;\text{ for } i = 1, \cdots, k \\ & \mathcal{A}\mathbf{vec}X = \mathbf{b} \\ & V_i \succeq 0 \;\text{ for } i = 1, \cdots, k \end{array} \tag{4.11}$$

and the dual

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \mathcal{A}^T\mathbf{y} - \sum_{i=1}^k (m_i - m_{i+1})U_i = 0 \\ & \text{trace } U_i = i \;\text{ for } i = 1, \cdots, k \\ & 0 \preceq U_i \preceq I \;\text{ for } i = 1, \cdots, k \end{array} \tag{4.12}$$

equivalents of (4.9).

The complementary slackness condition for feasible X^*, z_i^*, V_i^*, $\mathbf{y}^*$, and U_i^* for $i = 1, \cdots, k$ to be optimal may be stated as:

$$(z_i^* I + V_i - (m_i - m_{i+1})X^*)U_i^* = (I - U_i^*)V_i^* = 0 \quad \text{for } i = 1, \cdots, k.$$

Notice that the primal and dual characterizations (4.11) and (4.12) contain $2k$ semidefinite constraints each involving $n \times n$ matrices, and therefore, the interior point methods discussed earlier require $O(\sqrt{kn})$ iterations for each new significant digit of accuracy. It would be interesting to improve this complexity to $O(\sqrt{n})$.

4.3. Minimizing sums of absolute-value-wise largest eigenvalues. The results of the two preceding subsections may be extended to the sum of the k *absolute-value-wise* largest eigenvalues as well. Overton and Womersley derived the max characterization similar to (4.3); applying duality to their result we obtain:

THEOREM 4.4. *For a symmetric matrix A the sum $|\lambda^1(A)| + \cdots + |\lambda^k(A)|$ is equal to optimal solution of the pair of primal and dual semidefinite programs:*

$$\begin{array}{llll} \max & A \bullet Y - A \bullet W & \min & kz + \text{trace } V + \text{trace } U \\ \text{s.t.} & \text{trace } (Y + W) = k & \text{s.t.} & zI + V - A \succeq 0 \\ & 0 \preceq Y \preceq I & & zI + U + A \succeq 0 \\ & 0 \preceq W \preceq I & & U \succeq 0 \\ & & & V \succeq 0. \end{array} \tag{4.13}$$

(Recall that $\lambda^i(X)$ is the i^{th} largest eigenvalue of X in the absolute-value sense.)

Now to solve the optimization problem

$$\min\{|\lambda^1(X)| + \cdots + |\lambda^k(X)| : \mathcal{A}\mathbf{vec}X = \mathbf{b}\} \tag{4.14}$$

we may simply add the equality constraints to the min formulation in (4.13) and then take its dual and we get the following pair of primal and dual semidefinite programs:

$$\begin{array}{llll} \min & kz + \text{trace } V + \text{trace } U & \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \mathcal{A}\mathbf{vec}X = \mathbf{b} & \text{s.t.} & \mathcal{A}^T\mathbf{y} = Y - W \\ & zI + V - X \succeq 0 & & \text{trace } (Y + W) = k \\ & zI + U + X \succeq 0 & & 0 \preceq Y \preceq I \\ & U \succeq 0 & & 0 \preceq W \preceq I. \\ & V \succeq 0 & & \end{array} \tag{4.15}$$

The complementary slackness theorem indicates that primal feasible z^*, V^*, and U^*, and dual feasible Y^*, and W^* are optimal if and only if

$$(z^* I + V^* - X^*)Y^* = (z^* I + U^* + X^*)W^* = (I - Y^*)U^* = (I - W^*)V^* = 0.$$

Again these results may be generalized to the weighted sums of absolute-value-wise largest eigenvalues. In other words, the problem

$$\min\{m_1|\lambda^1(X)| + \cdots + m_k|\lambda^k(X)| : \mathcal{A}\mathbf{vec}X = \mathbf{b}\} \tag{4.16}$$

may be expressed by a primal and dual pair of semidefinite programs. First, let us ignore the equality constraints $\mathcal{A}\mathbf{vec}X = \mathbf{b}$, and assume that X is a fixed matrix A. Then, we have

THEOREM 4.5. *The sum* $m_1|\lambda^1(A)| + \cdots + m_k|\lambda^k(A)|$, *where* A *is a symmetric matrix equals the optimal solution of the primal program:*

$$\begin{array}{ll} \min & \sum_{i=1}^k iz_i + \sum_{i=1}^k \text{trace } (U_i + V_i) \\ \text{s.t.} & z_i I + U_i - (m_i - m_{i+1})A \succeq 0 \ \text{ for } i = 1, \cdots, k \\ & z_i I + V_i + (m_i - m_{i+1})A \succeq 0 \ \text{ for } i = 1, \cdots, k \\ & U_i \succeq 0 \ \text{ for } i = 1, \cdots, k \\ & V_i \succeq 0 \ \text{ for } i = 1, \cdots, k \end{array} \tag{4.17}$$

and the dual program:

$$\begin{array}{ll} \max & \sum_{i=1}^k (m_i - m_{i+1})(A \bullet Y_i - A \bullet W_i) \\ \text{s.t.} & \text{trace } (Y_i + W_i) = i \ \text{ for } i = 1, \cdots, k \\ & 0 \preceq Y_i \preceq I \ \text{ for } i = 1, \cdots, k \\ & 0 \preceq W_i \preceq I \ \text{ for } i = 1, \cdots, k. \end{array} \tag{4.18}$$

Now we may replace A by X and impose the equality constraints on the min characterization in (4.17). After taking the dual we have the following pair of primal and dual SDP equivalents of (4.16):

$$\begin{array}{ll} \min & \sum_{i=1}^k iz_i + \sum_{i=1}^k \text{trace } (U_i + V_i) \\ \text{s.t.} & \mathcal{A}\mathbf{vec}X = \mathbf{b} \\ & z_i I + U_i - (m_i - m_{i+1})X \succeq 0 \ \text{ for } i = 1, \cdots, k \\ & z_i I + V_i + (m_i - m_{i+1})X \succeq 0 \ \text{ for } i = 1, \cdots, k \\ & U_i \succeq 0 \ \text{ for } i = 1, \cdots, k \\ & V_i \succeq 0 \ \text{ for } i = 1, \cdots, k \end{array} \tag{4.19}$$

and,

$$\begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & \mathcal{A}^T\mathbf{y} = \sum_{i=1}^k (m_i - m_{i+1})(Y_i - W_i) \\ & \text{trace } (Y_i + W_i) = i \ \text{ for } i = 1, \cdots, k \\ & 0 \preceq Y_i \preceq I \ \text{ for } i = 1, \cdots, k \\ & 0 \preceq W_i \preceq I \ \text{ for } i = 1, \cdots, k. \end{array} \tag{4.20}$$

Finally, the complementary slackness theorem for problem (4.16) states that primal (4.19) feasible z_i^*, V_i^*, and U_i^*, and dual (4.20) feasible Y_i^*, and W_i^*, for $i = 1, \cdots, k$ are optimal if and only if

$$(z_i^* I + V_i^* - (m_i - m_{i+1})X_i^*)Y_i^* = (z_i^* I + U_i^* + (m_i - m_{i+1})X_i^*)W_i^* = (I - Y_i^*)U_i^* = (I - W_i^*)V_i^* = 0$$

for $i = 1, \cdots, k$.

The characterization (4.3), and the max part of (4.13) were given in Overton and Womersley [54]. Also, Fletcher in [23] derives a closely related result to (4.3) but the result was incorrect (Fletcher had $0 \preceq S$ rather than $0 \preceq S \preceq I$.) The min characterizations as well as the primal and dual formulation of the variants with equality constraints, we believe are new.

In a similar manner, primal and dual SDP formulations can be derived for maximizing (weighted) sums of the smallest eigenvalues of symmetric matrices or (weighted) sums of the largest singular values of arbitrary matrices; we omit these straightforward formulations here (see [62] for the study of singular values). However, maximizing the last few smallest eigenvalues of a symmetric matrix *absolute-value-wise*, or sum of the last few smallest singular values of an arbitrary matrix cannot be formulated as SDP because these problems are not convex programs.

5. Applications in combinatorial optimization. The semidefinite programming problem studied in the previous sections has applications in combinatorial optimization, especially in graph theory. The connection usually is the spectral properties of graphs. Semidefinite programs may arise in two different roles. Their more common role is to provide an approximation–an upper or lower bound–on an NP-hard combinatorial optimization problem. In such role one hopes that the SDP bound give rise to much sharper bounds than the more common LP bounds. Remarkably SDP relaxations have been shown to give rise to approximation algorithms whose guaranteed performance is superior to any known combinatorial or LP approximation technique, see for instance [25]. The second role is to give *exact* characterization to some special cases of combinatorial optimization problems. An example of such application is the SDP formulation of maximum clique and maximum stable set problem in perfect graphs.

In the following sections we first examine a general approach of Lovász and Schrijver which applies semidefinite programming to zero-one integer programming problems. Then we study other applications such as the maximum stable set, the maximum induced k-partite subgraph, and graph partitioning (in particular, graph bisection) and the maximum cut problems.

5.1. Nonlinear relaxations of 0-1 programming. Consider the integer programming problem

$$\max\{\bar{\mathbf{c}}^T\bar{\mathbf{x}} : \bar{A}\bar{\mathbf{x}} \geq \mathbf{b} \text{ and } \bar{x}_i \in \{0,1\}\}. \tag{5.1}$$

The LP relaxation of (5.1) results from replacing $\bar{x}_i \in \{0,1\}$ with $0 \leq \bar{x}_i \leq 1$. This relaxation serves as a first approximation of the solution of (5.1). In general, this first approximation may be nonintegral and far from the actual solutions. Most effective methods of integer programming consist of adding new "cutting planes" to the LP relaxation and then using some branch and bound technique to the resulting problem. It seems however, that little work has been done in generating "nonlinear" but convex cuts in the feasible region of the LP relaxation. Generally such cuts may produce far better approximations than linear cuts. An ingenious approach for creating a class of nonlinear cuts has been proposed by Lovász and Schrijver in [42]. The idea is to "lift" the space from vectors in $\Re^n$ to $n \times n$ symmetric matrices[6]. In essence, they provide a convex set which contains the feasible region of (5.1) and is contained in the feasible region of its LP relaxation. Furthermore, this convex relaxation may be expressed as a projection of the feasible set of a semidefinite program, and therefore itself may be represented as an SDP. Here is a summary of Lovász and Schrijver technique.

First, it is convenient to homogenize the integer program (5.1) by introducing a new variable x_0 as a multiple of $\mathbf{b}$ and then imposing the constraint $x_0 = 1$. After this transformation the homogenized integer programming problem and its linear

[6] The presentation here is more restrictive than given in [42]. Lovász and Schrijver consider optimization problems over a cone $\mathcal{K}$ endowed with a separation oracle, and derive nonliner cuts for the subcone generated by 0-1 vectors in $\mathcal{K}$.

programming relaxation can be written as:

$$
\begin{array}{llll}
\underline{\textbf{IP}} & & & \\
\max & \mathbf{c}^T\mathbf{x} & & \\
\text{s.t.} & \mathbf{a}_i^T\mathbf{x} \geq 0 & \text{for } i = 1, \cdots, m \\
 & x_i \in \{0,1\} & \text{for } i = 0, \cdots, n \\
 & x_0 = 1 & &
\end{array}
\qquad
\begin{array}{llll}
\underline{\textbf{LP}} & & & \\
\max & \mathbf{c}^T\mathbf{x} & & \\
\text{s.t.} & \mathbf{a}_i^T\mathbf{x} \geq 0 & \text{for } i = 1, \cdots, m \\
 & 0 \leq x_i \leq x_0 & \text{for } i = 0, \cdots, n \\
 & x_0 = 1. & &
\end{array}
$$

Let P be the convex cone which is the feasible region of the LP relaxation *without* the constraint $x_0 = 1$, and $\Im(P)$ its integer hull (that is, $\Im(P)$ is the convex cone generated by 0-1 vectors in **LP** with $x_0 = 1$.) First, we decompose the set of constraints into two sets (with possible overlap); then we multiply each inequality in the first set by each inequality in the second set to obtain quadratic constraints, then replace each occurrence of $x_i x_j$ by a new variable x_{ij} to get linear constraints again; finally we impose on the matrix $X = (x_{ij})$ the positive semidefinite constraint. If P_1 and P_2 are the cones defined by the first and second sets of constraints, then $P = P_1 \cap P_2$, and the space of matrices just defined is denoted by $M_+(P_1, P_2)$. More formally, let J_1 and J_2 be two subsets that cover the index set of the inequality constraints in **LP**. Define $A_1{:=}A_{J_1}$, and $A_2{:=}A_{J_2}$, and P_i the set $\{\mathbf{x} : A_i\mathbf{x} \geq \mathbf{0}\}$ for $i = 1, 2$. We require that constraints $0 \leq x_i \leq x_0$ be in both subsets. Then

$$M_+(P_1, P_2){:=}\{X \in \Re^{\frac{n \times n}{2}} : X \succeq 0,\ X\mathbf{e}_0 = \mathbf{diag}(X),\ \text{and } (A_1 \otimes A_2)\mathbf{vec}(X) \geq \mathbf{0}\}$$

where $\mathbf{e}_0 = (1, 0, \cdots, 0)^T$. Also, let $N_+(P_1, P_2)$ be the set of n-vectors made up of diagonals of matrices in $M_+(P_1, P_2)$, that is

$$N_+(P_1, P_2){:=}\{\mathbf{diag}(X) : X \in M_+(P_1, P_2)\}.$$

The main result of Lovász and Schrijver–for the purposes of our discussion–is that:

$$\Im(P) \subseteq N_+(P_1, P_2) \subseteq P.$$

It is clear that optimizing a linear function over $N_+(P_1, P_2)$ is a semidefinite programming problem, and interior point techniques may be applied (as long as P is given by an explicit system of inequalities.) The process just described may be quite powerful in certain combinatorial optimization problems. For instance in a general branch and bound algorithm, one may use interior point algorithms to solve the optimization problem

$$\max\{\mathbf{c}^T\mathbf{x} : \mathbf{x} \in N_+(P_1, P_2)\}.$$

The solution then may be used as a bound and the resulting $\mathbf{x}$ necessarily satisfies $\mathbf{0} \leq \mathbf{x} \leq \mathbf{1}$. Now if for some coordinate i we have $0 < x_i < 1$ then we branch by solving the two SDP subproblems with additional constraints, respectively $x_i = 0$ and $x_i = 1$. From a practical point of view such subproblems are all polynomial time solvable by the interior point methods, though they are computationally more expensive than the classical branch and bound approach based on linear programming relaxations. The advantage however is that the bounds are sharper (hopefully much sharper) than the corresponding LP bounds, and therefore the total number of subproblems solved may be considerably smaller.

Lovász and Schrijver show that applying the N_+ operator to the LP relaxation of the stable set polytope of a graph $G = (V, E)$ gives bounds that are already stronger than a combination of several well-known classes of linear cuts. Recall that a stable set in a graph $G = (V, E)$ is a subset of vertices S where each pair of vertices i and j in S are nonadjacent. Let $\mathbf{w}$ be a weight vector on the vertices of G, such that w_i is the weight of vertex i. The weighted maximum stable set problem in graphs can now be formulated as the following 0-1 program:

$$\begin{array}{lll} \max & \mathbf{w}^T\mathbf{x} & \\ \text{s.t.} & x_i + x_j \leq 1 & \text{for all } \{i,j\} \in E \\ & x_i \in \{0,1\} & \text{for all } i \in V. \end{array} \tag{5.2}$$

Now we homogenize (5.2) by adding a new variable x_0, then apply the N_+ operator with P_1 begin the entire feasible set of (5.2) and P_2 induced by $0 \leq x_i \leq x_0$:

$$P_1{:=}P = \{\mathbf{x} : \ x_i + x_j \leq x_0 \text{ for all } i,j \in E, \ \text{ and } 0 \leq x_i \leq x_0 \text{ for all } i \in V\},$$

and

$$P_2{:=}\{\mathbf{x} : \ x_0 - x_i \geq 0, \ \text{ and } x_i \geq 0\}.$$

Let the resulting set be $N_+(\text{STAB } G)$. Optimization over this set is a semidefinite program and can be done in polynomial time using interior point methods (Lovász and Schrijver use the ellipsoid method to establish polynomiality). Furthermore, it is clear that

$$\text{STAB } G \subseteq N_+(\text{STAB } G) \subseteq E{-}\text{STAB } G$$

where STAB G is the convex hull of all 0-1 vectors that characterize some stable set of G, and E–STAB G is the polytope associated with the LP relaxation of (5.2) (that is the polytope obtain by replacing constraints $x_i \in \{0,1\}$ by $0 \leq x_i \leq 1$.) The set $N_+(\text{STAB } G)$ is convex, but generally nonpolyhedral. However, Lovász and Schrijver show that the set of points in $N_+(\text{STAB } G)$ already satisfy the following classes of well-known valid inequalities for STAB G:

1. **Clique constraints.** Let K be a clique in G, that is a subset of vertices every pair of which is adjacent. Let S be a stable set in G. Then clearly $|S \cap K| \leq 1$. This observation implies that for all cliques in G the inequality

$$\mathbf{1}_K^T\,\mathbf{x} \leq 0 \tag{5.3}$$

(where $\mathbf{1}_K$ is the characteristic vector of clique K) is valid for STAB G. Define

$$Q{-}\text{STAB } G := \{\mathbf{x} : \ x_i \geq 0 \text{ and } \mathbf{1}_K^T\,\mathbf{x} \leq 0 \text{ for each clique } K\}$$

2. **Odd hole constraints.** For every cycle (hole) C with $2k+1$ edges and every stable set S we know that $|C \cap S| \leq k$. Thus, for all odd cycles C in G the constraint

$$\mathbf{1}_C^T\mathbf{x} \leq k \tag{5.4}$$

is valid for STAB G. Define

$$C{-}\ \text{STAB } G := \{\mathbf{x} : \ x_i \geq 0 \text{ and } \mathbf{1}_C^T\mathbf{x} \leq k \text{ for each odd cycle } C\}$$

3. **Odd anti-hole constraints.** Let $\bar{C}$ be a graph whose edge complement set is an odd cycle. Then the maximum stable set in G has two vertices and therefore, $|\bar{C} \cap S| \leq 2$ for all stable sets S. Therefore, for all odd antiholes $\bar{C}$ in G every inequality

$$\mathbf{1}_{\bar{C}}^T \mathbf{x} \leq 2 \tag{5.5}$$

is valid for STAB G. Define

$$\bar{C}-\text{STAB } G := \{\mathbf{x} : \ x_i \geq 0 \text{ and } \mathbf{1}_{\bar{C}}^T \mathbf{x} \leq 2 \text{ for each odd antihole } \bar{C}\}$$

4. **Odd wheel constraints.** Let W be a graph with $2k$ vertices such that vertices $v_1, v_2, \cdots, v_{2k-1}$ induce a cycle and vertex v_{2k} is adjacent to all other vertices. Then W is called an odd wheel. It can be shown (see [32]) that for all wheels W in G, the inequality

$$\sum_{i=1}^{2k-1} x_{v_i} + (k-1) x_{v_{2k}} \leq k-1 \tag{5.6}$$

is valid for STAB G. Define

$$W\text{–STAB } G := \{\mathbf{x} : \ x_i \geq 0 \text{ and } \sum_{i=1}^{2k-1} x_{v_i} + (k-1) x_{v_{2k}} \leq k-1 \text{ for each odd wheel } W\}$$

It turns out that (see [42])

$$\begin{aligned} &\text{STAB } G \subseteq N_+(\text{STAB } G) \\ &\subseteq Q-\text{STAB } G \cap C-\text{STAB } G \cap \bar{C}-\text{STAB } G \cap W-\text{STAB } G \subseteq E-\text{STAB } G \end{aligned}$$

and $N_+(\text{STAB } G)$ already provides sharper relaxation of STAB G than any of the polytopes defined above. Yet optimization over $N_+(\text{STAB } G)$ is an SDP problem and the interior point methods developed in this paper may yield practical ways of achieving strong bounds on the maximum stable set problem.

Remark: Barriers for polytopes with exponentially many facets. A strong property of the ellipsoid method for combinatorial optimization problems is that generally one does not need to have the linear programming formulation of the problem *explicitly*. All that is required is existence of a separation oracle and an initial ellipsoid to start the process. For instance, for certain classes of graphs the stable set polytope may be characterized completely by $C-$STAB G (such graphs are called t-perfect). Other classes may have their stable set polytope characterized by $Q-$STAB G (perfect graphs), or by $C-$STAB $G \cap Q-$STAB G (h-perfect graphs), or in general any combination of the polytopes mentioned in items 1 through 4 above. The stable set polytopes of such graphs have in general exponentially many facets. However, in [32, 42] it is shown that one can construct polynomial time computable separation oracles for these polytopes and thus find the maximum stable set for the corresponding graphs in polynomial time.

It is common belief that in contrast to the ellipsoid method, interior point methods require explicit knowledge of the facets of the polytope on which we wish to optimize, see for instance [32] and the quotation from [27] in the introduction. However, we can use polynomial time interior point methods to optimize over STAB G in the special cases mentioned above, *even though* the number of facets in such polytopes may be exponentially large. In fact, the ground breaking work of Nesterov and Nemirovskii implies that–at least in principle–a listing of all inequality constraints in the LP formulation is not necessary. One needs–instead of a separation oracle as is required in the ellipsoid method–a polynomial time computable *barrier oracle* with a polynomially bounded *self-concordance* parameter. For instance, as was indicated, we can optimize over $N_+(\text{STAB } G)$ in polynomial time, and $N_+(\text{STAB } G) = \text{STAB } G$ for the classes of graphs mentioned above. The general results of Nesterov and Nemirovskii imply that one can directly compute a barrier function for $N_+(\text{STAB } G)$:

THEOREM 5.1. *Let $b : \text{Int}N_+(\text{STAB } G) \to \Re$ be the function defined by:*

$$b(\mathbf{x}) := \min\{-\ln\det X : \mathbf{diag}(X) = \mathbf{x}, X \in M_+(\text{STAB } G)\}. \tag{5.7}$$

Then there is an interior point algorithm which uses $b(\mathbf{x})$ as its barrier and finds $\max\{\mathbf{w}^T\mathbf{x} : \mathbf{x} \in N_+(\text{STAB } G)\}$ in $O(\sqrt{n}\max(\|\mathbf{w}\|, |\ln\epsilon|))$ iterations and error at most ϵ.

Proof. Nesterov and Nemirovskii prove that $\ln\det X$ is *n-selfconcordant* for the cone of positive semidefinite $n \times n$ matrices. (See [48] for definitions). They also show that existence of an n-self-concordant barrier for a convex set in general implies that one can optimize a linear function over that set with every $O(\sqrt{n})$ iterations yielding a significant bit. Furthermore, in Proposition 1.5, p. 121 of [50] they show that if a convex set $K \subseteq \Re^n$ is endowed with an n-self-concordant barrier b, and $\mathcal{A} : \Re^n \to \Re^m$ is an affine transformation mapping K on to $\mathcal{A}(K)$ then the following function is n-self-concordant for $\mathcal{A}(K)$:

$$b^+(\mathbf{y}) := \inf\{b(\mathbf{x}) : \mathbf{x} \in \mathcal{A}^{-1}(\mathbf{y}) \cap \text{Int } K\}.$$

Now the theorem follows immediately from the definition of $N_+(\text{STAB } G)$ as given in [42] with the affine transformation $\mathcal{A}$ replaced by projection of elements of $M_+(\text{STAB } G)$ onto their diagonals. □

Notice that each iteration itself requires evaluating function ($b(\mathbf{x})$ which involves another optimization problems. Nevertheless the result above shows that if a convex set K in $\Re^n$ can be represented as a projection of another convex set $K' \in \Re^N$ with $N > n$, such that K' is endowed with a polynomial time computable p-self-concordant barrier, then there is a polynomial time computable p-self-concordant barrier for K. In combinatorial optimization there are many examples of polytopes with exponentially many facets which anyhow can be represented as a projection of polytopes in higher dimensions but with fewer (polynomially many) facets. For all such polytopes one can apply interior point methods and optimize over them in polynomial time. For a thorough discussion of liftings of polyhedra associated with combinatorial optimization problems consult [65, 42] and the references cited in them.

It is an interesting problem to look for easily computable (for instance NC–computable or at least polynomial time computable) barriers for combinatorial optimization problems whose linear programming formulation contains exponentially many inequalities. A concrete open problem is to find an easily computable barrier for the matching polytope with the property that a suitable interior point algorithm with such barrier requires $O(\sqrt{m})$ iterations where m is the number of edges in the graph. This problem is especially interesting because Yannakakis has shown that under certain symmetry preserving conditions on the lift operator it is impossible to lift the matching polytope to a higher dimensional polytope with polynomially many facets, [65]. Whether the matching polytope can be represented as a projection of a convex set endowed with an $O(m)$-self-concordant barrier function remains an interesting open problem.

5.2. Maximum cliques in perfect graphs. A particularly nice application of semidefinite programming is to the solution of the maximum clique problem in perfect graphs. A graph $G(V,E)$ is called perfect if for all induced subgraphs G' of G, the size of the maximum clique, $\omega(G')$, equals the size of minimum proper coloring, $\chi(G')$. (A proper coloring of vertices of a graph is an assignment of colors to each vertex such that no two adjacent vertices have the same color.) It is clear that $\omega(G) \le \chi(G)$ for all graphs, as one needs at least $\omega(G)$ colors just to cover the vertices of the maximum clique. Several interesting properties of perfect graphs should be noted. First, the perfect graph theorem of Lovász indicates that a graph is perfect if and only if its complement is perfect, [40]. This statement is equivalent to saying that for all induced subgraphs G' of G, $\alpha(G') = \rho(G')$, where $\alpha(G')$ is the size of the largest stable set in G', and $\rho(G')$ is the size of the smallest number of cliques that cover all vertices of G'. Thus, in effect studying cliques in perfect graphs is equivalent to studying stables sets and any algorithm for one is valid for the other one (by simply applying it to the complementary graph.) As a consequence of the perfect graph theorem one can show that equality of maximum cliques and minimum coloring extends to the weighted graphs. More precisely, let $\mathbf{w} \in N^n$ be an integral weight vector defined on the vertices of G. A proper $\mathbf{w}$-coloring of G is an assignment of colors to the vertices of G such that each vertex has at least w_i colors and for two adjacent vertices, their color sets are disjoint. $\chi(G,\mathbf{w})$ is the minimum number of colors over all proper $\mathbf{w}$-colorings of G. A maximum weighted clique in G is the clique whose sum of weights of vertices is maximum; this sum is denoted by $\omega(G,\mathbf{w})$. A graph is perfect if and only if for all weight vectors $\mathbf{w} \in N^n$, $\omega(G,\mathbf{w}) = \chi(G,\mathbf{w})$. Restating this for the complements of graphs, we have that a graph is perfect if and only if $\alpha(G,\mathbf{w}) = \rho(G,\mathbf{w})$, where, $\alpha(G,\mathbf{w})$ is the weight of the maximum weighted stable set in G, and $\rho(G,\mathbf{w})$ is the minimum number of cliques required to cover vertices of G such that each vertex i is in at least w_i cliques. These results are equivalent to the following statement (See [32]):

THEOREM 5.2. *A graph $G = (V,E)$ is perfect iff* STAB $G = Q -$ STAB G.

Therefore, already the discussion in the preceding subsection imply that computing maximum cliques and maximum independent sets in perfect graphs can be accomplished in polynomial time by interior point methods. However, in this case one can derive a slightly stronger result.

Lovász in [41] discovered an invariant of graphs, $\theta(G,\mathbf{w})$, which has two desirable properties: first it is polynomial time computable, and second it is simultaneously an upper bound for $\omega(G,\mathbf{w})$ and a lower bound for $\chi(G,\mathbf{w})$. This invariant can be defined by a pair of primal and dual semidefinite programs. Let

$$\mathcal{M}{:=}\{X \in \Re^{\frac{n \times n}{2}} : \ X_{ij} = 0 \text{ for all } i,j \in E \text{ or } i = j\}$$

and

$$\mathcal{M}^{\perp}{:=}\{Y \in \Re^{\frac{n \times n}{2}} : \ Y_{ij} = 0 \text{ for all } i,j \notin E\}.$$

Then the *weighted Lovász number* of G is defined by the following primal-dual SDP pair:

$$\begin{aligned} \theta(G,\mathbf{w}) \ &:= \ \min\{\lambda_1(X+W) : \ X \in \mathcal{M}\} \\ &= \ \max\{W \bullet Y : \ Y \in \mathcal{M}^{\perp},\ Y \succeq 0 \text{ and trace } Y = 1\} \end{aligned} \tag{5.8}$$

where $W{:=}\sqrt{\mathbf{w}}\sqrt{\mathbf{w}}^T$ and $\sqrt{\mathbf{w}}$ is an n-vector whose i^{th} component is $\sqrt{w_i}$. This min–max equality is proved directly in [32], and also follows easily from the duality theory stated earlier, see (4.8).

LEMMA 5.3. *For every vertex weighted graph $G = (V, E)$,*

$$\omega(G, \mathbf{w}) \leq \theta(G, \mathbf{w}) \leq \chi(G, \mathbf{w})$$

and

$$\alpha(G, \mathbf{w}) \leq \vartheta(G, \mathbf{w}) := \theta(\bar{G}, \mathbf{w}) \leq \rho(G, \mathbf{w}).$$

See [32] chapter 9 for a thorough treatment of Lovász number of graphs including several other characterization and many interesting properties. We just mention here that $\theta(G, \mathbf{w})$ is a relaxation of $\max\{\mathbf{w}^{\mathbf{x}} : \mathbf{x} \in N_+(\text{STAB } G)\}$. Since in case of perfect graphs we have

$$\omega(G, \mathbf{w}) = \theta(G, \mathbf{w}) = \chi(G, \mathbf{w})$$

and

$$\alpha(G, \mathbf{w}) = \vartheta(G, \mathbf{w}) = \rho(G, \mathbf{w}).$$

we can actually compute the maximum clique and maximum stable set in polynomial time for this class of graphs. In [32] the ellipsoid method was used to establish the polynomial time computability of maximum cliques in perfect graphs. We now show that interior point methods give us a slightly stronger result than the ellipsoid method. More precisely, we show that computing maximum cliques (and maximum stable sets) in perfect graphs can be accomplished in $\tilde{O}(\sqrt{n})$ randomized parallel time using P-RAM model of computation if $\|\mathbf{w}\|_\infty = O(n^c)$ for some constant c[7]. This is straightforward. First recall that we showed a standard SDP problem can be solved in $O(\sqrt{n}\max(L, L', |\ln\epsilon|))$ iterations, if L is the number of bits in the input SDP, L' is an *a priori* bound on the norm of the solution, and ϵ is the accuracy required on the size of the duality gap. In case of perfect graphs we only need to set $\epsilon = 1/3$; in fact, if z_k and Y_k are our current primal and dual estimates where there is only one integer between z_k and $W \bullet Y_k$ then we can stop and declare $\theta(G, \mathbf{w}) = \lceil z_k \rceil = \lfloor W \bullet Y_k \rfloor$. Furthermore, $L = \tilde{O}(1)$ since all coefficients in the primal-dual characterization of $\theta(G, \mathbf{w})$ in (5.8) are either zero or one or $w_i w_j$. Similarly, $L' = \tilde{O}(1)$ because the weight of the maximum clique cannot exceed $\sum w_i$. Thus computing $\theta(G, \mathbf{w})$ requires $\tilde{O}(\sqrt{n})$ iterations. Each iteration essentially involves solving a system of linear equations which is already known to be in complexity class NC, that is requires $\tilde{O}(1)$ time with polynomial number of processors. Therefore, computing $\theta(G, \mathbf{w})$ for polynomially bounded $\mathbf{w}$ requires $\tilde{O}(\sqrt{n})$ operations on a P-RAM model of computation.

It remains to show that computing the maximum clique itself can be accomplished in $\tilde{O}(\sqrt{n})$. We cannot use the self reducibility process here since it may require $O(n)$ time even on a P-RAM machine. However, observe that if the maximum clique is unique then we can compute it in $\tilde{O}(\sqrt{n})$ parallel time. One could remove one vertex i of the graph and compute $\theta(G\backslash i, \mathbf{w})$ for the remaining graph. The vertex i is in the unique maximum clique if and only if $\omega(G\backslash i, \mathbf{w}) < \omega(G, \mathbf{w})$. Therefore, testing this simultaneously for all vertices we get the set of vertices in the maximum clique. In general we do not have uniqueness, but we could use the randomized perturbation scheme of Mulmuley, Vazirani and Vazirani, [46]. First recall their isolating lemma:

[7] $\tilde{O}(\sqrt{n})$ means $O(\sqrt{n}\log^k n)$ for some constant k.

LEMMA 5.4. *Let $S = \{x_1, \cdots, x_n\}$ and F a family of subsets of S, that is $F = \{S_1, \cdots, S_N\}$. Further, let elements of S be assigned integer weights chosen uniformly and independently at random from $[1, 2n]$. Then,*

$$\Pr[\text{There is a unique maximum weight set in } F] \geq \frac{1}{2}.$$

See [46] for proof.

To get a maximum clique in a perfect graph we follow a procedure similar to the one adopted by Mulmuley, Vazirani and Vazirani for constructing the minimum weighted perfect matching in graphs. The idea is to assign weights to vertices randomly so that with high probability the maximum clique with the new weights is unique, but at the same time, this clique is among the maximum cliques with the original weights.

Let $C := \sum_i w_i$. First give a weight of $2C^2 w_i$ to each vertex i so that the weight of maximum weighted cliques is at least $2C^2$ more than the next largest clique weight. Then perturb weight of each vertex i by adding integer u_i uniformly and independently chosen from integers in $[1, 2C]$. So now each vertex has weight $\mathbf{w}_i = 2C^2 w_i + u_i$. Notice that if a clique was not maximum before, then it is impossible for it to become maximum after assigning new weights. Therefore, the maximum clique with respect to new weights is among one of the maximum cliques with respect to the original weights. The isolating lemma implies that this clique is unique with a probability at least $1/2$ and we may use the scheme mentioned earlier in this section to find it in parallel.

We should mention that this scheme, in fact, results in a Las Vegas type randomized algorithm. No randomization is involved in computing the $\omega(G, \mathbf{w})$; only constructing a maximum clique involves probabilistic choices. If the weights generated do not result in a unique maximum weighted clique, our method may return a set which is not even a clique. This can be checked in parallel and the algorithm returns a message of failure; any set returned by the algorithm is a genuine maximum clique with no possibility of error. We summarize these results in the following theorem:

THEOREM 5.5. *Let $G = (V, E)$ be a perfect graph with an integral weight vector $\mathbf{w}$ on its vertices. Let also that $\|\mathbf{w}\|_\infty = O(n^c)$ for some constant c. Then one can compute the maximum weighted clique and the maximum weighted stable set of G in $\tilde{O}(\sqrt{n})$ Las Vegas randomized parallel time using a P-RAM model of computation.*

Finally we remark that presently no representation of the stable set polytope of perfect graphs as projection of a higher dimensional polytope with polynomially many facets is known. Therefore, STAB G for a perfect graph G serves as an example of a polytope with exponentially many facets on which one can optimize a linear function in polynomial time using interior point methods. In fact, as mentioned in the last subsection, one can compute an n-self-concordant barrier for this polytope in polynomial time.

5.3. The maximum induced k-partite subgraph problem. In [47] G. Narasimhan and R. Manber generalized the concept of the Lovász number of graphs as follows: Let $\alpha_k(G)$ be the size of the largest induced k-partite subgraph in G. Recall that $\rho(G)$ is the minimum number of cliques that can cover all vertices of G. Then Narasimhan and Manber show that

$$\alpha_k(G) \leq \vartheta_k(G) := \min_{X \in \mathcal{M}^\perp} \sum_{i=1}^{k} \lambda_i(X + J) \leq k\rho(G) \tag{5.9}$$

where J is the matrix of all 1's. For $k = 1$ ϑ_k reduces to the Lovász number ϑ. It is clear now that computing $\vartheta_k(G)$ is an SDP problem and may be solved by interior point methods. Taking the dual of (5.9) we get

$$\begin{array}{rcll} \vartheta_k(G) & = & \max & J \bullet Y \\ & & \text{s.t.} & \operatorname{trace} Y = k \\ & & & Y \in \mathcal{M} \\ & & & 0 \preceq Y \preceq I. \end{array} \tag{5.10}$$

It is not difficult to extend the bound of Narasimhan and Manber to the weighted case. Let $\mathbf{w}$ be a weight vector over the vertices of G and $\alpha_k(G, \mathbf{w})$ the maximum weight k-colorable induced subgraph of G.

THEOREM 5.6. *Let* $W = (\sqrt{\mathbf{w}})(\sqrt{\mathbf{w}})^T$. *Then* $\alpha_k(G, \mathbf{w}) \le \vartheta_k(G, \mathbf{w})$, *where* $\vartheta_k(G, \mathbf{w})$ *is defined as*

$$\begin{array}{rcl} \vartheta_k(G, \mathbf{w}) & := & \min\{\sum_1^k \lambda_i(X + W) : X \in \mathcal{M}^\perp\} \\ & = & \max\{W \bullet Y : Y \in \mathcal{M} \text{ and } \operatorname{trace} Y = k,\ 0 \preceq Y \preceq I\}. \end{array} \tag{5.11}$$

Proof. (This proof is essentially the same as the one given in [31] for the case $k = 1$.) One can transform a weighted graph G into an unweighted one $G_{\mathbf{w}}$ by replacing each vertex i with w_i mutually nonadjacent vertices and then connecting all w_i vertices arising from vertex i to all w_j vertices arising from vertex j if and only if i and j are adjacent in G. Clearly the size of the unweighted maximum k-partite subgraph of $G_{\mathbf{w}}$ equals $\alpha_k(G, \mathbf{w})$. It suffices to show that $\vartheta_k(G, \mathbf{w}) = \vartheta_k(G_{\mathbf{w}})$. Now, in $G_{\mathbf{w}}$ two vertices i and j (respectively edges uv and kl) are *equivalent* if there is an automorphism of $G_{\mathbf{w}}$ mapping i to j (respectively uv to kl). In particular all w_i vertices arising from vertex i in G are equivalent; so are the all edges arising from uv. It is clear that if two vertices i and j (respectively two edges uv and kl) are equivalent, then in (5.10) the corresponding variables Y_{ii} and Y_{jj} (respectively Y_{uv} and Y_{kl}) are equivalent in the sense that by exchanging these variables (5.10) does not change at all. This in turn implies that among all optimal solutions of (5.10) for graph $G_{\mathbf{w}}$, there are solutions where equivalent vertices (respectively edges) have identical optimal values for their corresponding variables. In other words, among all optimal solutions of (5.10) for $G_{\mathbf{w}}$, there is one solution $Y^*_{\mathbf{w}}$ with the following property: $Y^*_{\mathbf{w}}$ can be partitioned into an $n \times n$ block matrix, such that the i, j block is a $w_i \times w_j$ matrix with all its entries equal to, say, y^*_{ij}. Now, matrix Y^* whose i, j entry is $y^*_{ij}/\sqrt{w_i w_j}$ is feasible for the max problem in the theorem and it is easy to verify that $W \bullet Y^* = J \bullet Y^*_{\mathbf{w}} = \vartheta_k(G_{\mathbf{w}})$ and thus, $\vartheta_k(G_{\mathbf{w}}) \le \vartheta_k(G, \mathbf{w})$. The other direction inequality is also easily verified by reversing the construction given. ◻

Let $\mho(k)$ be the class of graphs for which $\alpha_k(G') = \vartheta_k(G')$ for all induced subgraphs G'. Then the sublinear parallel time algorithm of theorem 5.5 may be extended to solve the largest induced k-partite subgraph problem for graphs in class $\mho(k)$. It remains an interesting open problem to fully characterize $\mho(k)$.

5.4. The graph partitioning problem. An important class of combinatorial NP-hard optimization problems which lend themselves to SDP methods for finding upper or lower bounds, arise from graph partitioning and cut problems. In many cases such problems result in semidefinite programs with only $O(n)$ variables. Therefore, the interior point methods may be especially efficient as each iteration requires only solving $n \times n$ systems of equations.

The general graph partitioning problem is defined as follows. Suppose we are given a set of integers $m_1 \geq m_2 \geq \cdots \geq m_k$, with $\sum_j m_j = n$. Denote by $\mathbf{m}$ the k-vector made up of m_j's. Let also that $G = (V, E)$ be a complete edge-weighted graph with n vertices and each edge $\{i, j\}$ with weight w_{ij}. We want to partition the vertices of G into k subsets such that the j^{th} subset has cardinality m_j, and that sum of the weights of those edges whose endpoints are in different subsets is minimized. Let us denote this minimum number by $\pi_{\mathbf{m}}(G)$. Computing $\pi_{\mathbf{m}}(G)$ is of course NP-hard. Donath and Hoffman in [17] and [18] derive the following lower bound on the size of the minimum partition (see also Barnes and Hoffman [5]). Let A be a matrix with $A_{ij} = w_{ij}$ $(A_{ii} = 0)$. Then Donath and Hoffman prove the following relation [18]:

$$\pi_{\mathbf{m}}(G) \geq -\frac{1}{2} \min_{\mathbf{1}^T\mathbf{x}=a} \sum_{j=1}^{k} m_j \lambda_j (A + \operatorname{Diag} \mathbf{x}) \tag{5.12}$$

where $a := -\sum w_{ij}$. Again it is clear that computing this bound is an SDP problem. Using the results from section 4 and after some simplification we get the following pair of primal and dual SDP programs:

$$\begin{array}{lll} \min & \sum_{i=1}^{k} i z_i + \mathbf{1}^T\mathbf{x} + \sum_{i=1}^{k} \operatorname{trace} V_i & \\ \text{s.t.} & z_i I + V_i + (m_i - m_{i+1}) \operatorname{Diag}\mathbf{x} \succeq (m_i - m_{i+1}) A & \text{for } i = 1, \cdots, k \\ & V_i \succeq 0 \quad \text{for } i = 1, \cdots, k & \end{array} \tag{5.13a}$$

and

$$\begin{array}{lll} \max & A \bullet \left(\sum_{i=1}^{k} (m_i - m_{i+1}) U_i \right) & \\ \text{s.t.} & \operatorname{trace} U_i = i & \text{for } i = 1, \cdots, k \\ & \sum_{i=1}^{k} (m_i - m_{i+1}) (U_i)_{jj} = 1 & \text{for } j = 1, \cdots, n \\ & 0 \preceq U_i \preceq I & \text{for } i = 1, \cdots, k. \end{array} \tag{5.13b}$$

Barnes and Hoffman in [5] describe a method that uses eigenvectors associated with the k largest eigenvalues of the optimal matrix $A + \operatorname{Diag}\mathbf{x}^*$ to generate a partition of the nodes of the graph. See also Barnes [6, 7].

An important special case of the graph partitioning problem is the case when all m_i's are equal. In that case the graph partitioning problem simplifies to:

$$\begin{array}{llllll} \min & (k/n)\mathbf{1}^T\mathbf{x} + \operatorname{trace} V & \max & A \bullet Y & \\ \text{s.t.} & V + \operatorname{Diag} \mathbf{x} \succeq A & \text{s.t.} & Y_{ii} = \frac{k}{n} & \text{for } i = 1, \cdots, n \\ & V \succeq 0 & & 0 \preceq Y \preceq I. & \end{array} \tag{5.14}$$

Boppana in [10] considers the graph bisection problem (that is when $k = 2$ and $m_1 = m_2 = n/2$) and derives the following bound on the *bisection width* $\beta(G)$, which is always sharper than (5.14):

$$\beta(G) \geq \frac{1}{4} \max [J \bullet (A + \operatorname{Diag}(\mathbf{x})) - n\lambda_1 (P(A + \operatorname{Diag}(\mathbf{x})P))]$$

where $P := (I - J/n)$ is the projection operator on the linear space $S := \{\mathbf{x} : \mathbf{1}^T\mathbf{x} = 0\}$. This characterization is equivalent to the following primal and dual SDP pair:

$$\begin{array}{ll} \min & nz + \mathbf{1}^T\mathbf{x} \\ \text{s.t.} & zI - \operatorname{Diag}(\mathbf{x}) - \frac{\mathbf{1}\mathbf{x}^T + \mathbf{x}\mathbf{1}^T}{2n} \succeq A + \frac{JA + AJ}{2n} \end{array} \tag{5.15a}$$

and

$$
\text{(5.15b)} \qquad \begin{array}{ll} \max & A(I+J/n) \bullet Y \\ \text{s.t.} & Y_{ii} + (1/n)\sum_{j=1}^{n} Y_{ij} = 1 \quad \text{for } i = 1, \cdots, n \\ & Y \succeq 0. \end{array}
$$

(Boppana had the min characterization only, the max characterization results by simply taking the dual.) To find an actual bisection Boppana uses an eigenvector corresponding to the largest eigenvalue of $\lambda_1(P(A + \mathrm{Diag}(\mathbf{x}^*)P)$ and outputs the bisection that has the $n/2$ largest component of the eigenvector on one side. Using the primal characterization Boppana shows that in the unweighted case (i.e the matrix A is simply the 0-1 adjacency matrix of graph G) one may get the optimal bisection with high probability. The graph bisection problem has important applications in the VLSI routing problem. Combining the SDP formulation of Hoffman and Donath, favorable average case analysis of Boppana, and the interior point technique developed in this paper may result in an effective and practical method for solving this problem. For generalizations of these ideas see [58].

Related to the graph bisection problem is the maximum cut problem: partition the nodes of the graph into two sets such that the number of edges with endpoints on different sets is maximum. Of course one obvious way for finding bounds for this problem is to solve the graph partitioning problem with $k = 2$, $m_1 = i$, and $m_2 = n - i$ for all $i = 1, \cdots \lfloor n/2 \rfloor$ (notice that in graph partitioning problem max and min characterizations are essentially equivalent by simply changing the weights w_i with $\sum w_j - w_i$). In [16, 56] the following SDP bound is proposed:

$$
\text{(5.16)} \qquad \min\{\frac{n}{4}\lambda_1(A + \mathrm{Diag}(\mathbf{x})) : \mathbf{1}^T\mathbf{x} = a\} \geq \mathrm{MC}(G)
$$

where $\mathrm{MC}(G)$ is the size of maximum cut in G. (5.16) is equivalent to:

$$
\text{(5.17)} \qquad \begin{array}{ll} \min & z + (1/n)\mathbf{1}^T\mathbf{x} \\ \text{s.t.} & zI - \mathrm{Diag}(\mathbf{x}) \succeq A \\ & \end{array} \qquad \begin{array}{ll} \max & A \bullet Y \\ \text{s.t.} & Y_{ii} = 1/n \\ & Y \succeq 0 \end{array}
$$

and may be solved by interior point methods. Recently Goemans and Williamson [25] have shown that the solution of (5.16) yields a cut whose size is guaranteed to be to within 0.87 of the optimum; the previous best result only guaranteed 0.5 of the optimum. Thus far, the best approximation algorithm for the maximum cut problem (as well as for the maximum satisfiability problem) are based on SDP relaxations. For related treatment of maximum cut and graph bisection problems see [57].

Acknowledgment. Discussions with Stephen Boyd, Uriel Feige, Don Knuth, László Lovász, Yuri Nesterov, Arkadii Nemirovskii, Motakuri Ramana, Rob Womersley and particularly Michael Overton and Yinyu Ye, have been most useful in preparing this article and removing several errors in the preliminary versions. Also comments of an anonymous referee were instrumental in removing errors in the statements of Farkas' lemma and duality theory in the earlier versions of this paper.

REFERENCES

[1] F. Alizadeh, *Combinatorial Optimization with Interior Point Methods and Semi-Definite Matrices*, PhD thesis, Computer Science Department, University of Minnesota, Minneapolis, Minnesota, 1991.

[2] ———, *Optimization Over Positive Semi-Definite Cone; Interior-Point Methods and Combinatorial Applications*, in Advances in Optimization and Parallel Computing, P. Pardalos, ed., North–Holland, Amsterdam, 1992.
[3] E. Anderson and P. Nash, *Linear Programming in Infinite Dimensional Spaces*, John Wiley & Sons, New York, 1987.
[4] K. M. Anstreicher, *A monotone projective algorithm for fractional linear programming*, Algorithmica, 1 (1986), pp. 483–498.
[5] E. Barnes and A. J. Hoffman, *Partitioning, spectra, and linear programming*, in Progress in Combinatorial Optimization, W. E. Pulleyblank, ed., Academic Press, New York, 1984.
[6] E. R. Barnes, *An algorithm for partitioning the nodes of a graph*, SIAM J. Alg. and Disc. Meth., 3 (1982).
[7] ———, *Partitioning the nodes of a graph*, in Graph Theory with Applications to Algorithms and Computer Science, L. Lesniak, D. R. Lick, and C. E. Wall, eds., John Wiley, 1982.
[8] M. Bazaraa, J. Jarvis, and H. Sherali, *Linear Programming and Network Flows*, John Wiley & Sons, New York, 1990.
[9] A. Ben-Israel, A. Charnes, and K. O. Kortanek, *Duality and Asymptotic Solvability over Cones*, Bull. Amer. Math. Soc., 75 (1969), pp. 318–324.
[10] R. B. Boppana, *Eigenvalues and graph bisection: an average case analysis*, in Proc. 28th IEEE Annual Symposium on Foundations of Computer Science, 1987.
[11] J. M. Borwein and H. Wolkowicz, *Characterization of optimality for the abstract convex program with finite dimensional range*, J. Austral. Math. Soc. Ser. A, 30 (1981), pp. 390–411.
[12] ———, *Facial reduction for a cone-convex programming problem*, J. Austral. Math. Soc. Ser. A, 30 (1981), pp. 369–380.
[13] B. D. Craven and J. J. Koliha, *Generalizations of Farkas' Theorem*, SIAM J. Math. Anal., 8 (1977), pp. 983–997.
[14] B. D. Craven and B. Mond, *Linear Programming with Matrix Variables*, Linear Algebra Appl., 38 (1981), pp. 73–80.
[15] J. Cullum, W. E. Donath, and P. Wolfe, *The minimization of certain nondifferentiable sums of eigenvalue problems*, Math. Programming Stud., 3 (1975), pp. 35–55.
[16] C. Delorme and S. Poljak, *Laplacian eigenvalues and the maximum cut problem*, Math. Programming, 62 (1993), pp. 557,574.
[17] W. E. Donath and A. J. Hoffman, *Algorithms for partitioning graphs and computer logic based on eigenvectors of connection matrices*, IBM Technical Disclosure Bulletin, 15 (1972).
[18] ———, *Lower bounds for the partitioning of graphs*, IBM J. Res. and Devel., 17 (1973).
[19] R. J. Duffin, *Infinite Programs*, in Linear inequalities and Related Systems, H. W. Kuhn and A. W. Tucker, eds., Princeton University Press, Princeton, NJ, 1956, pp. 157–170.
[20] M. Fan, *A quadratically convergent local algorithm on minimizing the largest eigenvalue of a symmetric matrix*, Linear Algebra Appl., 188,189 (1993), pp. 207–230.
[21] U. Feige and L. Lovász, *Two-prover one-round proof systems: their power and their problems*, in Proc. 24th Annual ACM Symposium on Theory of Computing, 1992, pp. 733–744.
[22] P. A. Fillmore and J. P. Williams, *Some convexity theorems for matrices*, Glasgow Math. J., 12 (1971), pp. 110–117.
[23] R. Fletcher, *Semi-definite matrix constraints in optimization*, SIAM J. Control Optim., 23 (1985), pp. 493–513.
[24] S. Friedland, J. Nocedal, and M. L. Overton, *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM J. Numer. Anal., 24 (1987), pp. 634–557.
[25] M. X. Goemans and D. P. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*, J. Assoc. Comput. Mach., (1994). (To appear). A preliminary version appeared in Proc. 26th Annual ACM Symposium on Theory of Computing.
[26] A. V. Goldberg, S. A. Plotkin, D. Shmoys, and E. Tardos, *Interior-Point Methods in Parallel Computation*, SIAM J. Comput., 21 (1991), pp. 140–150.
[27] D. Goldfarb and M. J. Todd, *Linear Programming*, in Optimization, G. L. Nemhauser, A. H. G. R. Kan, and M. J. Todd, eds., Handbooks in Operations Research and Management Sciences, North–Holland, Amsterdam, 1989.
[28] C. C. Gonzaga, *An Algorithm for Solving Linear Programming in $O(n^3L)$ Operations*, in Progress in Mathematical Programming, N. Megiddo, ed., Springer Verlag, Berlin, 1989, pp. 1–28.
[29] A. Graham, *Kronecker Products and Matrix Calculus: with Applications*, Ellis Horwood, London, 1981.
[30] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences*

in combinatorial optimization, Combinatorica, 1 (1981).

[31] ———, *Polynomial algorithms for perfect graphs*, in Perfect Graphs, C. Berge and V. Chvatal, eds., North Holland, Amsterdam, 1984. also Ann. Discrete Math., 21.

[32] ———, *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, 1988.

[33] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[34] L. Hurwicz, *Programming in Linear Spaces*, in Studies in Linear and Non–linear Programming, II, K. J. Arrow, L. Hurwicz, and H. Uzawa, eds., Stanford University Press, Stanford, CA, 1958, pp. 4–102.

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

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

[37] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*, no. 538 in Lecture Notes in Computer Science, Springer–Verlag, New York, 1991.

[38] M. Kojima, S. Mizuni, and A. Yoshise, *A primal-dual interior-point algorithm for linear programming*, in Progress in Mathematical Programming, N. Megiddo, ed., Berlin, 1989, Springer Verlag.

[39] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, New York, 1985.

[40] L. Lovász, *Normal hypergraphs and the weak perfect graph conjecture*, Discrete Math., 2 (1972), pp. 253–267.

[41] ———, *On the shannon capacity of a graph*, IEEE Trans. Inform. Theory, 25 (1979), pp. 1–7.

[42] L. Lovász and A. Schrijver, *Cones of Matrices and Setfunctions, and 0-1 Optimization*, SIAM J. Optim., 1 (1991), pp. 166–190.

[43] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, N. Megiddo, ed., New York, 1989, Springer Verlag, Berlin.

[44] R. C. Monteiro and I. Adler, *Interior path following primal–dual algorithms. Part* I*: Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[45] ———, *Interior path following primal–dual algorithms. Part* II*: Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[46] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani, *Matching is as easy as matrix inversion*, Combinatorica, (1987), pp. 105–131.

[47] G. Narasimhan and R. Manber, *A Generalization of Lovász's ϑ Function*, in Polyhedral Combinatorics, W. Cook, and P. D. Seymour, ed., vol. 1 of DIMACS series in Discrete Mathematics and Theoretical Computer Science, Amer. Math. Soc. and Ass. for Computing Machinery, 1990, pp. 19–27.

[48] Y. Nesterov and A. Nemirovskii, *Self-Concordant Functions and Polynomial Time Methods in Convex Programming*, Central Economical and Mathematical Institute, U.S.S.R Academy of Science, Moscow, 1990.

[49] ———, *Acceleration of the path–following method for optimization over the cone of positive semidefinite matrices.* Unpublished manuscript, 1992.

[50] ———, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[51] M. L. Overton, *On minimizing the maximum eigenvalue of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 256–268.

[52] ———, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

[53] M. L. Overton and R. S. Womersley, *On the sum of the largest eigenvalues of a symmetric matrix*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 41–45.

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

[55] G. Pataki, *Algorithms for linear programs over cones and semi-definite programming.* Manuscript, Graduate School of Industrial Administration, Carnegie-Mellon University, 1993.

[56] S. Poljak and F. Rendl, *Solving the max-cut problem using eigenvalues*, Tech. Report 91735-OR, Forschungsinstitut Für Diskrete Mathematik, Institut Für ökonometrie und Operations Research, Rheinische Friedrich–Wilhelms-Universität, Bonn, November 1991.

[57] ———, *Nonpolyhedral relaxations of graph-bisection problems*, Tech. Report 92-55, DIMACS, November 1992.

[58] F. Rendl and H. Wolkowicz, *A projection technique for partitioning the nodes of a graph*, CORR Report 90-20, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada, 1993.

[59] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[60] A. Schrijver, *Theory of Linear and Integer Programming*, J. Wiley & Sons, New York, 1986.
[61] A. Shapiro, *Extremal problems on the set of nonnegative definite matrices*, Linear Algebra Appl., 67 (1985), pp. 7–18.
[62] S. Subramani, *Sums of Singular Values*, master's thesis, School of Mathematics, University of New South Wales, Kensington, Australia, 1993.
[63] L. Vandenberghe and S. Boyd, *Primal–dual potential reduction method for problems involving matrix inequalities*, tech. report, Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA, January 1993. *Math. Prog.* (to appear).
[64] H. Wolkowicz, *Some applications of optimization in matrix theory*, Linear Algebra Appl., 40 (1981), pp. 101–118.
[65] M. Yannakakis, *Expressing combinatorial optimization problems by linear programs*, J. Comput. Syst. Sci., 43 (1991), pp. 441–466.
[66] Y. Ye, *A class of projective transformations for linear programming*, SIAM J. Comput., 19 (1990), pp. 457–466.
[67] ———, *An $O(n^3L)$ Potential Reduction Algorithm for Linear Programming*, Math. Programming, 50 (1991), pp. 239–258.