PREPRINT ANL/MCS-P692-0997, MATHEMATICS AND COMPUTER SCIENCE DIVISION, ARGONNE NATIONAL LABORATORY

Nonsymmetric Search Directions for Semidefinite Programming

Rongqin Sheng* and Florian A. Potra†

September 1997

Abstract

Two nonsymmetric search directions for semidefinite programming, the XZ and ZX search directions, are proposed. They are derived from a nonsymmetric formulation of the semidefinite programming problem. The XZ direction corresponds to the direct linearization of the central path equation $XZ = \nu I$, while the ZX direction corresponds to $ZX = \nu I$. The XZ and ZX directions are well defined if both X and Z are positive definite matrices, where X may be nonsymmetric. We present an algorithm using the XZ and ZX directions alternately following the Mehrotra predictor-corrector framework. Numerical results show that the XZ/ZX algorithm is, in most cases, faster than the XZ+ZX method of Alizadeh, Overton, and Haeberly (AHO) while achieving similar accuracy.

Key Words: semidefinite programming, nonsymmetric, search direction, interior-point algorithm, high accuracy.
Abbreviated Title: Nonsymmetric directions for SDP.

*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA.
This research was supported by the Mathematical, Information and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.
†Department of Mathematics, The University of Iowa, Iowa City, IA 52242, USA.

1 Introduction

The semidefinite programming (SDP) problem has the standard form

$$(P) \quad \min\{C \bullet X : A_i \bullet X = b_i,\ i = 1, \ldots, m,\ X \in \mathcal{S}^n_+\}, \tag{1.1}$$

and its associated dual problem is

$$(D) \quad \max\{b^T y : \sum_{i=1}^m y_i A_i + Z = C,\ (y, Z) \in \mathbb{R}^m \times \mathcal{S}^n_+\}, \tag{1.2}$$

where $C \in \mathcal{S}^n$, $A_i \in \mathcal{S}^n$, $i = 1, \ldots, m$, $b = (b_1, \ldots, b_m)^T \in \mathbb{R}^m$ are given data. Here $\mathcal{S}^n$ denotes the set of all $n \times n$ symmetric matrices and $\mathcal{S}^n_+$ the set of all $n \times n$ symmetric positive semidefinite matrices. $G \bullet H$ is the trace of $G^T H$. For simplicity we assume that A_i, $i = 1, \ldots, m$, are linearly independent.

Under the assumption that both (1.1) and (1.2) have finite solutions and their optimal values are equal, X^* and (y^*, Z^*) are solutions of (1.1) and (1.2) if and only if they are solutions of the following nonlinear system:

$$A_i \bullet X = b_i,\ i = 1, \ldots, m, \tag{1.3a}$$

$$\sum_{i=1}^m y_i A_i + Z = C, \tag{1.3b}$$

$$XZ = 0, \quad X,\ Z \in \mathcal{S}^n_+. \tag{1.3c}$$

Most primal-dual interior-point methods for semidefinite programming can be interpreted as iterative algorithms for solving the nonlinear system (1.3). The search directions used by those interior-point algorithms are associated with different ways of linearizing the central path equation

$$XZ = \nu I, \tag{1.4}$$

where $\nu \geq 0$ is the central path parameter.

In order to ensure the symmetry of the iterates X^k and Z^k generated by interior-point methods, symmetric reformulations of central path equation (1.4) have been developed. Alizadeh, Haeberly, and Overton [1] considered instead of (1.4) the symmetric equation

$$XZ + ZX = 2\nu I. \tag{1.5}$$

Zhang [11] proposed a generalized symmetrization of the form

$$\frac{1}{2}[P^{-1}XZP + (P^{-1}XZP)^T] = \nu I, \tag{1.6}$$

where P can be any nonsingular matrix. Recently, Monteiro and Tsuchiya [5] considered the symmetric central path equations

$$Z^{1/2}XZ^{1/2} = \nu I, \qquad X^{1/2}ZX^{1/2} = \nu I. \tag{1.7}$$

Linearization of the above symmetric central path equations leads to different search directions. The most commonly used directions are the XZ+ZX or AHO direction [1], the HKM direction [2, 3, 4], and the NT direction [7], obtained from (1.6) by taking P equal to I, $Z^{1/2}$, and $[Z^{1/2}(Z^{1/2}XZ^{1/2})^{-1/2}Z^{1/2}]^{1/2}$ respectively. Among these directions, the AHO direction has been observed to achieve the highest accuracy. We also mention that Monteiro and Zanjácomo [6], and Toh [9] recently reported other search directions that can attain high accuracy.

All the above-mentioned search directions involve the linearization of a specific symmetric central path equation. In this paper, we show that the nonsymmetric central path equation (1.4) can be directly used without any symmetrization and that the resulting nonsymmetric search direction can be applied for interior-point algorithms. Our approach is based on the following nonsymmetric formulation of SDP whose solution set contains that of (1.3):

$$A_i \bullet X = b_i, \; i = 1, \ldots, m, \tag{1.8a}$$

$$\sum_{i=1}^{m} y_i A_i + Z = C, \tag{1.8b}$$

$$XZ = 0, \;\; 0 \preceq X \in \mathbb{R}^{n\times n}, \; Z \in \mathcal{S}^n_+. \tag{1.8c}$$

In (1.8) the notation $0 \preceq X \in \mathbb{R}^{n\times n}$ means that X is positive semidefinite, but not necessarily symmetric. In Section 2, we will prove that if (X^*, y^*, Z^*) is a solution of (1.8), then $(\mathbf{sym}(X^*), y^*, Z^*)$ is a solution of (1.3), where we define the operator **sym** by

$$\mathbf{sym}(G) = \frac{1}{2}(G + G^T), \quad \text{for any real square matrix } G.$$

The same result holds if (1.8c) is replaced by

$$ZX = 0, \;\; 0 \preceq X \in \mathbb{R}^{n\times n}, \; Z \in \mathcal{S}^n_+. \tag{1.9}$$

The *XZ search direction* $(\Delta X, \Delta y, \Delta Z)$ is defined as the solution of the following linear system:

$$X\Delta Z + \Delta X Z = \sigma\mu I - XZ, \tag{1.10a}$$

$$A_i \bullet \Delta Z = b_i - A_i \bullet X, \;\; i = 1, \ldots, m, \tag{1.10b}$$

$$\sum_{i=1}^{m} \Delta y_i A_i + \Delta Z = C - \sum_{i=1}^{m} y_i A_i - Z, \tag{1.10c}$$

where $\mu = X \bullet Z/n$, and $\sigma \in [0,1]$ is a centering parameter. Thus, the XZ direction can be viewed as the result of the direct linearization of the central path equation $XZ = \nu I$.

Correspondingly the *ZX search direction* is the solution of the linear system (1.10) with (1.10a) replaced by

$$Z\Delta X + \Delta Z X = \sigma\mu I - ZX.$$

We will show that the XZ and ZX direction exist provided X and S are positive definite. Extensive numerical experiments show that interior-point methods based on the XZ or on the ZX direction alone are not very efficient. On the other hand, if these directions are used alternately, the efficiency is highly improved. Such a method is called an XZ/ZX method. Our numerical experiments show that the XZ/ZX method integrated in the Mehrotra predictor-corrector framework is competitive with the corresponding AHO method. The two methods have similar accuracy. Although our method usually takes about three more iterations, the CPU time as well as the number of floating-point operations is less in most cases. This is because our algorithm avoids the Lyapunov equations that the AHO method has to solve at each iteration.

The following notation and terminology are used throughout the paper:
$\mathbb{R}^p$: the p-dimensional Euclidean space;
$\mathbb{R}^p_+$: the nonnegative orthant of $\mathbb{R}^p$;
$\mathbb{R}^p_{++}$: the positive orthant of $\mathbb{R}^p$;
$\mathbb{R}^{p\times q}$: the set of all $p \times q$ matrices with real entries;
$\mathcal{S}^p$: the set of all $p \times p$ symmetric matrices;
$\mathcal{S}^p_+$: the set of all $p \times p$ symmetric positive semidefinite matrices;
$\mathcal{S}^p_{++}$: the set of all $p \times p$ symmetric positive matrices;
$M \succeq 0$: M is positive semidefinite;
$M \succ 0$: M is positive definite;
$\lambda_i(M),\ i = 1, \ldots, n$: the eigenvalues of $M \in \mathcal{S}^n$;
$\lambda_{\max}(M),\ \lambda_{\min}(M)$: the largest, smallest, eigenvalue of $M \in \mathcal{S}^n$;
$G \bullet H \equiv \mathrm{Tr}(G^T H)$;
$\|\cdot\|$: Euclidean norm of a vector and the corresponding norm of a matrix, i.e.,
$\|y\| \equiv \sqrt{\sum_{i=1}^p y_i^2}, \quad \|M\| \equiv \max\{\|My\| : \|y\| = 1\}$;
$\|M\|_F \equiv \sqrt{\sum_{i=1}^p \sum_{j=1}^q [M]_{ij}^2},\ M \in \mathbb{R}^{p\times q}$: Frobenius norm of a matrix;
$\mathbf{sym}(M) \equiv (M + M^T)/2,\ M \in \mathbb{R}^{p\times p}$.

2 On the Nonsymmetric Formulation of SDP

The following result is well known.

Lemma 2.1 *Let $X,\ Z \in \mathcal{S}^n_+$. Then $X \bullet Z = 0$ if and only if $XZ = ZX = 0$.*

The next lemma shows that (1.8c) can be replaced by (1.9).

Lemma 2.2 *Let $0 \preceq X \in \mathbb{R}^{n\times n},\ Z \in \mathcal{S}^n_+$. Then $XZ = 0$ if and only if $ZX = 0$.*

Proof. ($\Rightarrow$). $XZ = 0$ implies $(X + X^T) \bullet Z = 2X \bullet Z = 0$. Then from Lemma 2.1, we obtain $(X + X^T)Z = 0$, which yields $X^TZ = -XZ = 0$ and hence $ZX = (X^TZ)^T = 0$.
($\Leftarrow$). Similar. ∎

Theorem 2.3

(a) Every solution of (1.3) is also a solution of (1.8).

(b) If (X^*, y^*, Z^*) is a solution of (1.8), then $(\mathbf{sym}(X^*), y^*, Z^*)$ is a solution of (1.3).

Proof. Part (a) follows directly from the definition of (1.8). To prove (b), we need to show only that $(\mathbf{sym}(X^*), y^*, Z^*)$ satisfies (1.3a) and (1.3c). Since $A_i, i = 1, \ldots, m$ are symmetric, we have

$$A_i \bullet \mathbf{sym}(X^*) = A_i \bullet X^* = b_i, i = 1, \ldots, m.$$

From $X^*Z^* = 0$ and Lemma 2.3 we obtain $Z^*X^* = 0$. Therefore,

$$\mathbf{sym}(X^*)Z^* = \frac{1}{2}[X^*Z^* + (Z^*X^*)^T] = 0.$$

∎

3 On the XZ and ZX Search Directions

The linear system (1.10) for the XZ search direction can be written in the following matrix form.

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ Z \otimes I & 0 & I \otimes X \end{pmatrix} \begin{pmatrix} \mathbf{vec}(\Delta X) \\ \Delta y \\ \mathbf{vec}(\Delta Z) \end{pmatrix} = \begin{pmatrix} \mathbf{vec}(R_d) \\ r_p \\ \mathbf{vec}(R_c) \end{pmatrix}, \tag{3.1}$$

where

$$A^T = [\mathbf{vec}(A_1), \mathbf{vec}(A_2), ..., \mathbf{vec}(A_m)],$$

$$r_i = b_i - A_i \bullet X, \; i = 1, \ldots, m, \tag{3.2a}$$

$$R_d = C - \sum_{i=1}^{m} y_i A_i - Z, \tag{3.2b}$$

$$r_p^T = [r_1, r_2, ..., r_m],$$

$$R_c = \sigma\mu I - XZ.$$

Here $\otimes$ denotes the Kronecker product. For any $n \times n$ matrix M, $\mathbf{vec}(M)$ denotes the vector obtained by stacking the columns of M, that is,

$$\mathbf{vec}(M) = (m_{11}, m_{21}, \ldots, m_{1n}, \ldots, m_{nn})^T.$$

The linear system (1.10) can be solved by the following procedure:

- Compute Δy by solving the linear system

$$M\Delta y = h, \tag{3.3}$$

  where

$$M = A(Z^{-1} \otimes X)A^T$$

  and

$$h = r_p + A[\mathbf{vec}(XR_dZ^{-1}) - \mathbf{vec}(R_cZ^{-1})].$$

- Compute ΔZ, ΔX as follows:

$$\begin{aligned} \Delta Z &= R_d - \sum_{i=1}^{m} \Delta y_i A_i, \\ \Delta X &= R_c Z^{-1} - X\Delta Z Z^{-1}. \end{aligned}$$

Lemma 3.1 *If $X \in \mathbb{R}^{n\times n}$ and $Z \in \mathcal{S}^n$ are positive definite, then the linear system (1.10) has a unique solution $(\Delta X, \Delta y, \Delta Z) \in \mathbb{R}^{n\times n} \times \mathbb{R}^m \times \mathcal{S}^n$.*

Proof. If the solution $(\Delta X, \Delta y, \Delta Z)$ of (1.10) exists, then the symmetry of ΔZ is automatic from (1.10c). Therefore, it is sufficient to prove that the Schur matrix $A(Z^{-1} \otimes X)A^T$ is nonsingular. From the symmetry of Z^{-1}, we have

$$\begin{aligned} & A(Z^{-1} \otimes X)A^T + [A(Z^{-1} \otimes X)A^T]^T \\ = \; & A[Z^{-1} \otimes (X + X^T)]A^T \; . \end{aligned}$$

The right-hand side of the above equation is positive definite because A has full rank and both Z and $X + X^T$ are positive definite. Therefore, $A(Z^{-1} \otimes X)A^T$ is positive definite and hence nonsingular. ∎

Remark 3.2 In the Schur complement equation (3.3) the Schur matrix M and the right side h can be computed by

$$m_{i,j} = A_i \bullet (XA_jZ^{-1}) \tag{3.4}$$

and

$$h_i = r_i + A_i \bullet [(XR_dZ^{-1}) + R_cZ^{-1}].$$

Let us consider the complexity of the computation of the XZ direction. Assume that the matrices A_i are not sparse. Then the major computational effort consists in forming the Schur matrix M. If formula (3.4) is used, the XZ direction can be computed in $4mn^3 + 2m^2n^2 + O(max\{m,n\}^3)$ flops, since $2m$ matrix multiplications and m^2 inner products are involved. Therefore, the complexity of computing the XZ direction by using formula (3.4) is

$$4mn^3 + 2m^2n^2 + O(max\{m,n\}^3).$$

Remark 3.3 The complexity of computing most commonly used search directions for SDP is of the form

$$\alpha mn^3 + \beta m^2n^2 + O(max\{m,n\}^3), \tag{3.5}$$

where α and β are two positive constants (see [6, 9]). We note that the third term in (3.5) cannot be neglected because sometimes it may contribute significantly to the complexity, especially when extra matrix factorizations are used. We also note that the computation of the XZ direction needs the least number of matrix factorizations. This feature is also shared by the HKM direction.

Remark 3.4 Theoretically, the complexity of computing the XZ direction can be reduced to

$$3mn^3 + 2m^2n^2 + O(max\{m,n\}^3),$$

by using the Cholesky factorization $Z = LL^T$ and the following formula, which is equivalent to (3.4):

$$m_{i,j} = (L^{-1}A_i) \bullet (L^{-1}A_iX^T). \tag{3.6}$$

Since L^{-1} is triangular, the computation of $L^{-1}A_i,\ i = 1, \ldots, m$ takes $mn^3 + O(\max\{m,n\}^3)$ flops. After $L^{-1}A_i,\ i = 1, \ldots, m$, is obtained, the computation of $L^{-1}A_iX^T$ involves m matrix multiplications, and thus needs $2mn^3 + O(\max\{m,n\}^3)$ flops. Finally, m^2 inner products are needed, thus accounting for $2m^2n^2 + O(\max\{m,n\}^3)$ flops. Therefore, the computation of the XZ direction with formula (3.6) takes $3mn^3 + 2m^2n^2 + O(max\{m,n\}^3$ flops. However, in our Matlab implementation, we use (3.4) instead of (3.6) because the CPU time often increases when (3.6) is applied. A similar observation was made by Toh [9]. Nevertheless, (3.6) may be useful in other computational environments.

Remark 3.5 The computation of the ZX direction is similar to that of the XZ direction. Actually, $(\Delta X, \Delta y, \Delta Z)$ is an XZ direction at (X, y, Z) if and only if $(\Delta X^T, \Delta y, \Delta Z)$ is a ZX direction at (X^T, y, Z).

4 The XZ/ZX Method

The algorithm described below is an XZ/ZX method because it uses the XZ and ZX search directions alternately. It follows the Mehrotra predictor-corrector algorithmic framework of Todd, Toh, and Tütüncü [8].

Algorithm 4.1 *Select a starting point $(X^0, y^0, Z^0) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n \times \mathcal{S}^n$ such that X and Z are positive definite. Choose an exponent ω and a constant $\gamma \in (0, 1)$.*

Repeat for $k = 0, 1, 2, \ldots$:
[For simplicity, let $(X, y, Z) = (X^k, y^k, Z^k)$ and $(X^+, y^+, Z^+) = (X^{k+1}, y^{k+1}, Z^{k+1})$.]

(Predictor step)

- *Compute the predicted direction $(\delta X, \delta y, \delta Z)$ by solving the linear system (1.10) with $\sigma = 0$.*

- *Determine the parameter σ:*

$$\sigma := \left(\frac{(X + \psi\delta X) \bullet (Z + \phi\delta Z)}{X \bullet Z} \right)^{\omega}, \tag{4.1}$$

where

$$\psi := \frac{-\gamma}{\min(-\gamma,\ \lambda_{\min}(\mathbf{sym}(X)^{-1}\mathbf{sym}(\delta X))}, \tag{4.2a}$$

$$\phi := \frac{-\gamma}{\min(-\gamma,\ \lambda_{\min}(Z^{-1}\delta Z))}. \tag{4.2b}$$

(Corrector step)

- *Compute the corrected direction* $(\Delta X, \Delta y, \Delta Z)$ *by solving linear system (1.10) with* σ *defined by (4.1) and the right side of (1.10a) modified as*

$$\sigma\mu I - XZ + \delta X \delta Z.$$

- *Compute* ψ *and* ϕ *from (4.2) with* $\delta X, \delta Z$ *replaced by* $\Delta X, \Delta Z$.
- *Update* $(X^+, y^+, Z^+) = (X^T, y, Z) + (\psi \Delta X^T, \phi \Delta y, \phi \Delta Z)$.

In our numerical implementation, we choose $\gamma = 0.98$ and set ω equal to 2 for the AHO method and 1 for others.

Remark 4.2 In Algorithm 4.1, through the updating

$$X^{k+1} := [X^k + \psi_k \Delta X^k]^T,$$

we actually use the XZ and ZX directions alternately. More specifically, Algorithm 4.1 is equivalent to an algorithm using the XZ and ZX directions alternately with the iteration sequence $\{(\tilde{X}^k,\ y^k,\ Z^k)\}$, where $\tilde{X}^k = X^k$ for $k = 2p-1$, and $\tilde{X}^k = (X^k)^T$ for $k = 2p,\ p \geq 1$. This property can be verified by a simple linear algebra manipulation.

5 Numerical Results

We thank Toh, Todd, and Tütüncü for making their Matlab code SDPT3 [10] available to us. We used their code for running the Mehrotra algorithm using the AHO, HKM, and NT search directions. We tested the following problems:

1. random SDP problem with $n = 100,\ m = 50$,
2. random SDP problem with $n = 50,\ m = 100$,

3. random SDP problem with $n = 100,\ m = 100$,

4. the matrix norm minimization problem with $n = 100,\ m = 30$,

5. the problem of computing the Chebyshev polynomial of a matrix with $n = 100\ m = 31$,

6. the Max-Cut problem with $n = 200,\ m = 200$,

7. the Educational Testing Problem (ETP) with $n = 110,\ m = 55$, and

8. the logarithmic Chebyshev approximation problem with $n = 300, m = 50$.

All the problems are taken from [8] and [10]. The reader is referred to [8] and [10] for details on the problems and the computation of the AHO, HKM and NT search directions. We performed our numerical experiment using Matlab 5.0. The computations were carried out on the IBM RS/6000 SP system at Argonne National Laboratory.

We tested ten random instances for each problem. We stopped the computation when either no progress was made (due to numerical instability) or the number of iterations reached 50. The average results are given in Tables 5.1 and 5.2.

From the results displayed in the two tables we observe the following:

- The XZ/ZX method and the AHO method achieve higher accuracy than the other methods.

- In most cases the XZ/ZX method is faster than the AHO method.

- The XZ/ZX method takes about three more iterations than the AHO method, with the exception of the ETP problem where the XZ/ZX method takes significantly more iterations.

- With the exception of the ETP problem the XZ/ZX method requires significantly fewer flops per iteration than the AHO method and only slightly more flops than the HKM method which requires the fewest flops per iteration of the methods tested.

Table 5.1: Computational results for varying classes of SDP. Ten random instances of each class are tested. Note that the infeasibility of all problems is reduced to a level of 10^{-13}, except for the last problem, where 10^{-12} is attained.

	Average Accuracy Achieved by $\lvert\text{mean}(\log_{10}(X \bullet Z))\rvert$				Average CPU Time (min.) to Attain the Accuracy			
	AHO	XZ/ZX	HKM	NT	AHO	XZ/ZX	HKM	NT
Random $n = 100$ $m = 50$	8.76	8.86	7.41	7.24	4.99	4.28	3.10	2.98
Random $n = 50$ $m = 100$	9.37	9.41	7.61	7.60	4.06	4.36	2.52	2.38
Random $n = 100$ $m = 100$	8.50	8.52	7.49	7.22	14.81	14.23	9.40	8.81
Norm min. $n = 100$ $m = 30$	12.52	12.56	9.64	9.19	2.72	2.26	1.66	1.79
Cheby. Poly. $n = 100$ $m = 31$	14.25	13.99	11.17	10.64	8.34	5.87	4.51	4.77
Maxcut $n = 200$ $m = 200$	11.05	10.50	7.74	7.43	65.20	28.73	21.82	22.84
ETP $n = 110$ $m = 55$	9.16	7.95	7.37	7.11	3.00	3.32	2.27	2.34
Log. Cheby. $n = 300$ $n = 50$	10.63	10.78	10.39	10.43	4.98	3.00	1.94	1.99

Table 5.2: Average results for the number of iterations and the flops per iteration used to achieve the accuracy listed in Table 5.1.

	Average No. of Iterations to Achieve the Accuracy				Average Mflops per Iteration			
	AHO	XZ/ZX	HKM	NT	AHO	XZ/ZX	HKM	NT
Random $n = 100$ $m = 50$	13.4	16.1	15.4	14.2	618	299	275	300
Random $n = 50$ $m = 100$	14.7	17.5	15.8	14.9	188	111	86	89
Random $n = 100$ $m = 100$	14.2	17.0	16.4	15.2	1253	652	553	578
Norm min. $n = 100$ $m = 30$	14.3	17.8	15.9	15.7	387	188	178	205
Cheby. Poly. $n = 100$ $m = 31$	15.6	19.1	16.5	16.2	1848	947	902	988
Maxcut $n = 200$ $m = 200$	15.6	17.7	15.7	15.7	12680	3419	3405	3616
ETP $n = 110$ $m = 55$	20.9	31.5	24.9	23.9	103	167	167	189
Log. Cheby. $n = 300$ $m = 50$	16.0	18.1	17.7	17.6	2.71	2.16	1.45	1.48

Acknowledgment

The authors thank Steve Wright for insightful discussions.

References

[1] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. Technical Report 721, New York University, New York, NY, 1996. To appear in *SIAM Journal on Optimization*.

[2] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2):342–361, 1996.

[3] M. Kojima, S. Shindoh, and S. Hara. Interior-point methods for the monotone linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7:86–125, 1997.

[4] R. D. C. Monteiro. Primal-dual path following algorithms for semidefinite programming. Working paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, September 1995. To appear in *SIAM Journal on Optimization*.

[5] R. D. C. Monteiro and T. Tsuchiya. Polynomiality of primal-dual algorithms for semidefinite linear complementarity problems based on the Kojima-Shindoh-Hara family of directions. Technical report, The Institute of Statistical Mathematics, Tokyo, August 1996.

[6] R. D. C. Monteiro and P. Zanjácomo. Implementation of primal-dual methods for semidefinite programming based on Monteiro and Tsuchiya Newton directions and their variants. Working paper, School of Industrial and Systems Engineering, Georgia Institute of Technology, July 1997. Revised August 1997.

[7] Y. E. Nesterov and M. J. Todd. Primal–dual interior–point methods for self–scaled cones. Technical Report 1125, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, 1995. To appear in *SIAM Journal on Optimization*.

[8] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. Technical Report, School of Operations Research and Industrial

Engineering, Cornell University, Ithaca, New York, 1996. To appear in *SIAM Journal on Optimization*.

[9] K. C. Toh. Search directions for primal-dual interior point methods in semidefinite programming. Technical Report, Department of Mathematics, National University of Singapore, July 1997.

[10] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3 — a Matlab software package for semidefinite programming. Manuscript, December 1996.

[11] Y. Zhang. On extending primal–dual interior–point algorithms from linear programming to semidefinite programming. TR 95-20, Department of Mathematics and Statistics, University of Maryland Baltimore County, October 1995. To appear in *SIAM Journal on Optimization*.