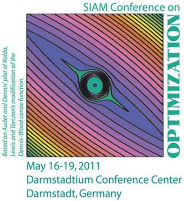
A Robust Algorithm for Semidefinite Programming

Xuan Vinh Doan, Serge Kruk, Henry Wolkowicz

Dept. of Combinatorics and Optimization
University of Waterloo



Motivation: Avoiding ill-conditioning in SDP methods

- Current SDP methods are based on primal-dual interior-point approaches; these involve a
  (i) symmetrization step to apply Newton's method,
  (ii) block elimination to reduce size of Newton equation.
- Both these steps create ill-conditioning in the Newton equation and singularity of the Jacobian at optimality
- We avoid the ill-conditioning and singularity using preprocessing and a backwards stable primal-dual interior-point approach based on an inexact Gauss-Newton method with a preconditioned (matrix-free) iterative method for finding the search direction
- We get improved accuracy and a iteration reduction

Outline: Robust Algorithm/Semidefinite Programming

1. Background: SDP/Algorithms
2. Matrix-free, path following, P-D Inter-exter. method
   - Preprocessing for bilinear optimality conditions
   - Local Quadratic Convergence and Crossover
   - Presolve and Preconditioning
3. Lovász Theta Function Problem
4. Numerics
   - Random Well-Conditioned Instances
   - Random Ill-Conditioned Instances
   - Random Instances of Lovász Theta Function Problem
5. Summary

Central Problem

Primal-dual SDP pair

$$\text{(PSDP)} \qquad \begin{array}{rll} p^* := \min & C \cdot X & \\ \text{s.t.} & \mathcal{A}(X) = b & \quad (\operatorname{trace} A_i X = b_i) \\ & X \succeq 0 & \end{array}$$

$$\text{(DSDP)} \qquad \begin{array}{rll} d^* := \max & b^T y & \\ \text{s.t.} & \mathcal{A}^*(y) + Z = C & \\ & Z \succeq 0 & \quad (Z = C - \sum_i y_i A_i) \end{array}$$

- $C, X, Z \in \mathcal{S}^n$, $n \times n$ real symmetric matrices
- $C \cdot X = \operatorname{trace}(CX)$, trace inner product
- $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ linear transformation; $\mathcal{A}^*$ adjoint

Optimality Paradigm

Perturbed overdetermined optimality conditions:
(i) dual/(ii) primal feas., (iii) pert. compl. slack. $:\to \mathbb{R}^{n\times n}$

$$F_\mu(X,y,Z) := \begin{pmatrix} (i) & \mathcal{A}^*(y)+Z-C \\ (ii) & \mathcal{A}(X)-b \\ (iii) & ZX-\mu I \end{pmatrix} = 0, \quad \mu > 0$$

Characterization of optimality

Under constraint qualification, the primal-dual solution (X,y,Z) with $X,Z \succeq 0$ is optimal iff $F_0(X,y,Z)=0$

P-d i-p methods: maintain $X,Z \succ 0$; with $\mu \downarrow 0$,

These methods are based on path following; the perturbed system is overdetermined; under nondegeneracy, Jacobian is full rank at optimality, e.g., AHO97[1].

Overdetermined: Symmetrize for Newton's method

Two most popular symmetrizations

(i) HRVW/KSM/M96 [5, 6, 8], and
(ii) NT97 [9, 10]

But: algorithms are not backwards stable! precond? sparsity?

- The linearizations (Jacobian) are singular at optimum e.g., T99[12]
- and the block elimination schemes result in increased ill-conditioning e.g.,GWW04[4],W01[14]
- finding reasonable preconditioners is difficult (impossible?)
- block eliminations make it difficult to exploit sparsity

Notation

$A_i \in \mathcal{S}^n, \forall i$

- $\mathcal{A}(X) = b \Leftrightarrow A_i \cdot X = b_i, \quad \forall i = 1, \ldots, m,$ ($\mathcal{A}$ full rank m)
- $\mathcal{A}^*(y) = \sum_{i=1}^{m} y_i A_i, \forall y \in \mathbb{R}^m$
- $\mathrm{vec} : \mathcal{M}^n \to \mathbb{R}^{n^2}$ (columnwise) $\quad \mathrm{Mat} := \mathrm{vec}^{-1}$
- triangular number, $t(n) = n(n+1)/2$
- $\mathrm{svec} : \mathcal{S}^n \to \mathbb{R}^{t(n)}$ (isometry)
  $\mathrm{sMat} = \mathrm{svec}^{-1}$ (note $\mathrm{sMat}^* = \mathrm{svec}$)

Set $A \in \mathbb{R}^{m \times t(n)}$ with rows $A_{i,:} = \mathrm{svec}(A_i), \forall i = 1, \ldots, m$

$$\mathcal{A}(X) = b \Leftrightarrow A\mathrm{svec}(X) = b.$$

Single bilinear optimality conditions

Null space representation of primal feas.; dual feas.

- columns of $Q \in \mathbb{R}^{t(n)\times(t(n)-m)}$ are taken from $\{q_1,\ldots,q_{t(n)-m}\}$ orthonormal basis of $\mathcal{N}(A)$
- $\hat{X} \succ 0$ primal feasible solution; $\hat{x} = \mathrm{svec}\,(\hat{X})$, $v \in \mathbb{R}^{t(n)-m}$
  $\mathcal{A}(X) = b \Leftrightarrow \mathrm{svec}\,(X) = \hat{x} + Qv$
- $c = \mathrm{svec}\,(C)$; dual feasibility:
  $\mathcal{A}^*(y) + Z = C \quad \Leftrightarrow \quad \mathrm{svec}\,(Z) = c - A^T y$

Equivalent (overdetermined) optimality conditions:

$$\boxed{G_\mu(v,y) := \underbrace{\mathrm{sMat}\,(c - A^T y)}_{Z}\ \underbrace{\mathrm{sMat}\,(\hat{x} + Qv)}_{X}\ \underbrace{-\mu I}_{-\mu I} = 0}$$

Apply Gauss-Newton

$$G_\mu(v,y) = \mathrm{sMat}\,(c - A^T y)\mathrm{sMat}\,(\hat{x} + Qv) - \mu I = 0$$

- single bilinear overdetermined system;
  $t(n)$ variables; n^2 equations
- apply Newton's method to $\min \frac{1}{2}\|G_\mu(v,y)\|_2^2$ and discard the second order terms, i.e. GN method
- For each $\mu > 0$, there exists unique p-d solution (X_μ, y_μ, Z_μ) with $X_\mu \succ 0, Z_\mu \succ 0$:
  lies on (defines) central path
  corresponding $\mathrm{svec}\,(Z_\mu) = c - A^T y_\mu$ and
  $\mathrm{svec}\,(X_\mu) = \hat{x} + Qv_\mu$, for appropriate v_μ

$G_\mu(v,y) = \mathrm{sMat}\,(c - A^T y)\mathrm{sMat}\,(\hat{x} + Qv) - \mu I = 0$

Gauss-Newton search direction; least-squares solution

$$-G_\mu(v,y) = G'_\mu(v,y) \begin{pmatrix} \Delta v \\ \Delta y \end{pmatrix},$$

where

- Jacobian $J = G'_\mu(v,y) : \mathbb{R}^{t(n)-m} \times \mathbb{R}^m \to \mathcal{M}^n$
- $\begin{cases} Z := C - \mathrm{sMat}\,(A^T y), \quad X := \hat{X} + \mathrm{sMat}\,(Qv), \\ \mathcal{Z}(v) := Z\mathrm{sMat}\,(Qv), \quad \mathcal{X}(y) := -\mathrm{sMat}\,(A^T y)X \\ J = [\mathcal{Z} \mid \mathcal{X}] \end{cases}$

(overdetermined) Gauss-Newton equation

$$\begin{aligned} -G_\mu(v,y) &= \mathcal{Z}(\Delta v) + \mathcal{X}(\Delta y) \\ &= Z\mathrm{sMat}\,(Q\Delta v) - \mathrm{sMat}\,(A^T \Delta y)X \end{aligned}$$

Adjoint Jacobian, J^*; normal equations

$$J = \begin{bmatrix} \mathcal{Z} & \mathcal{X} \end{bmatrix}; \quad J^* = \begin{bmatrix} \mathcal{Z}^* \\ \mathcal{X}^* \end{bmatrix}$$

- $J^* \circ J = \begin{bmatrix} \mathcal{Z}^* \circ \mathcal{Z} & \mathcal{Z}^* \circ \mathcal{X} \\ \mathcal{X}^* \circ \mathcal{Z} & \mathcal{X}^* \circ \mathcal{X} \end{bmatrix},$
- normal equations $\quad J^* \circ J \begin{pmatrix} \Delta v \\ \Delta y \end{pmatrix} = -J^* \circ G_\mu(v, y)$
- $\mathcal{X}^*(M) = -\frac{1}{2}A\mathrm{svec}\,(XM^T + MX).$
  $\mathcal{Z}^*(M) = \frac{1}{2}Q^T\mathrm{svec}\,(M^TZ + ZM).$

Infeasible start

Residuals: r_d, r_p, R_c

$$F_\mu(v,y,x,z) := \begin{pmatrix} z - c + A^T y \\ x - \hat{x} - Qv \\ \mathrm{sMat}\,(z)\mathrm{sMat}\,(x) - \mu I \end{pmatrix} =: \begin{pmatrix} r_d \\ r_x \\ R_c \end{pmatrix} = 0.$$

- $\Delta x = Q\Delta v - r_x$, $\Delta z = -A^T\Delta y - r_d$
- $G'_\mu(v,y)\begin{pmatrix} \Delta v \\ \Delta y \end{pmatrix} := Z\mathrm{sMat}\,(Q\Delta v - r_x) - \mathrm{sMat}\,(A^T\Delta y + r_d)X = -R_c$
  equivalently:
  $Z\mathrm{sMat}\,(Q\Delta v) - \mathrm{sMat}\,(A^T\Delta y)X = Z\mathrm{sMat}\,(r_x) + \mathrm{sMat}\,(r_d)X - R_c.$
- with step sizes α_p, α_d, new residuals;
  ($\alpha = 1$, exact feasibility maintained)!:
  $r_d \leftarrow (z - \alpha_d(A^T\Delta y + r_d)) - c + A^T(y + \alpha_d\Delta y) = (1-\alpha_d)r_d,$
  $r_x \leftarrow x + \alpha_p(Q\Delta v - r_x) - \hat{x} - Q(v + \alpha_p\Delta v) = (1-\alpha_p)r_p.$

Local quadratic convergence

Consider the following general problem

$F(x,y) = \begin{pmatrix} L(x,y) \\ R(x) \end{pmatrix} = 0, \quad L(x,y) = x - \hat{x} - Ty, T$ full rank

linearization

$$\left\{F'(x,y)\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -F(x,y)\right\} \Leftrightarrow \left\{ \begin{array}{l} \Delta x - T\Delta y = -L(x,y) \\ R'(x)\Delta x = -R(x) \end{array} \right\}.$$

with $\Delta x = T\Delta y - L(x,y)$ get:

$R'(x)T\Delta y = -R(x) + R'(x)L(x,y)$ assumed overdetermined

Local convergence (Kantorovitch type) theorem

$J_x(x) := R'(x)$; $J_y(x) := J_x(x)T$; $R(x) \in \mathcal{C}^2$ in open convex set D; $J_x(x) \in \text{Lip}_\gamma(D)$; $||J_x(x)|| \leq \alpha, \forall x \in D$;
$\lambda = \lambda_{\min}(J_y(x_*)^T J_y(x_*))$; $\exists x_* \in D, y_*$ with $\boxed{L(x_*, y_*) = 0, J_x(x_*)^T R(x_*) = 0}$;
$\left|\left|(J_x(x) - J_x(x_*))^T R(x_*)\right|\right|_2 \leq \sigma \,||x - x_*||_2\,, \quad \forall x \in D.$

Define iteration:

$\Delta x_k = T \Delta y_k - L(x_k, y_k)$, $\boxed{x_{k+1} = x_k + \Delta x_k}$;
$\Delta y_k = (J_y(x_k)^T J_y(x_k))^{-1} J_y(x_k)^T (-R(x_k) + J_x(x_k) L(x_k, y_k))$,
$\boxed{y_{k+1} = y_k + \Delta y_k}$.

Let $\tau := ||T||_2$, $c \in (1, \lambda/(\tau^2\sigma))$. If $\tau^2\sigma < \lambda$, then $\exists \epsilon > 0$ with $\forall x_0 \in N(x_*, \epsilon)$, y_0 arbitrary, we have that:
the sequences $\{x_k\}, \{y_k\}$ converge to (x_*, y_*) with
$L(x_k, y_k) = 0, \forall k \geq 1$; and

$$\boxed{||x_{k+1} - x_*|| \leq \frac{\tau^2 c}{\lambda}\left(\sigma\,||x_k - x_*|| + \frac{\alpha\gamma}{2}\,||x_k - x_*||^2\right),}$$

$$\boxed{||x_{k+1} - x_*|| \leq \frac{\lambda + \tau^2 c\sigma}{2\lambda}\,||x_k - x_*|| < ||x_k - x_*||\,.}$$

Applying Theorem to Gauss-Newton equation

Key: lower bound to smallest singular value of Jacobian

$L(z,x,y,v) = \begin{pmatrix} z - c + A^T y \\ x - \hat{x} - Qv \end{pmatrix}$, with $T = \begin{pmatrix} -A^T & 0 \\ 0 & Q \end{pmatrix}$;

$R_\mu(z,x) = \mathrm{sMat}\,(z)\mathrm{sMat}\,(x) - \mu I$; $R_\mu(z_*,x_*) = 0$. $R'(z,x)\begin{pmatrix} \Delta z \\ \Delta x \end{pmatrix} = Z\mathrm{sMat}\,(\Delta x) + \mathrm{sMat}\,(\Delta z)X$ implies $||R'(z,x)|| \leq \sqrt{||Z||^2 + ||X||^2} = ||(z,x)||$. Therefore the constants are:

$$\boxed{\sigma = 0, \tau = ||T|| = \sqrt{||A||^2 + ||Q||^2}, \gamma = 1, \alpha = \sup_{(z,x)\in D} ||(z,x)||}$$

Poly. time global convergence

But it is difficult to estimate $\sigma_{\min}(J_y(x_*))$; which is needed for global poly. time convergence.
(a global polynomial convergence result for a scaled Gauss-Newton method is given in KPRT99[2].)

Crossover technique

Set (Gauss) Newton free: affine scaling $\mu = 0$, step length 1

(under nondegeneracy) Theorem indicates that there is a neighborhood around optimal solution where free Gauss-Newton method converges quadratically.

- affine scaling: set barrier parameter $\mu = 0$
- set G-N free: take full step lengths $= 1$; no nonnegativity checks/no backtracking
- region of quadratic convergence: build heuristic rule based on the infeasibility of the current solution (z, x)

Presolve and Preconditioning

Well-conditioned, sparse bases for range and nullspace of A

- motivation: apply a matrix-free Krylov type method for search direction and exploit sparsity.
- Given A sparse, find permutations $P_r A P_c = [S \mid E]$, with $S \in \mathbb{R}^{m \times m}$ well-conditioned, (approx) triangular
- columns of $Q = \begin{bmatrix} -S^{-1}E \\ I \end{bmatrix}$ gives basis of nullspace of A

ω measure for $X \succ 0$: $\omega(X) := \frac{\operatorname{trace}(X)/n}{\det(X)^{\frac{1}{n}}}$

Use ω-optimal diagonal and ω-optimal block diagonal preconditioners.

Lovász Theta Function Problem

Primal TN

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ undirected graph; $n = |\mathcal{V}|$ nodes; $m = |\mathcal{E}|$ edges
E matrix all ones, $E_{ij} = (e_i e_j^T + e_j e_i^T)/\sqrt{2}$ ij-th unit matrix; e_i i-th unit vector

$$(\text{TN}) \qquad \begin{array}{rll} \vartheta(\mathcal{G}) := p^* := & \max & E \cdot X \\ & \text{s.t.} & I \cdot X = 1 \\ & & E_{ij} \cdot X = 0, \quad \forall (i,j) \in \mathcal{E} \\ & & X \succeq 0 \end{array}$$

Dual DTN

$$(\text{DTN}) \qquad \begin{array}{rll} d^* := & \min & z \\ & \text{s.t.} & zI + \sum_{(i,j)\in\mathcal{E}} y_{ij} E_{ij} - Z = E, \\ & & Z \succeq 0, \end{array}$$

Gauss-Newton for Lovász theta function problem

Let $V := \begin{pmatrix} -e^T \\ I \end{pmatrix} \in \mathbb{R}^{n\times(n-1)}$, (nullspace V is $e^\perp$)

Primal-Dual feasibility: X, Z

$$\begin{array}{llcl} X \succeq 0 & \text{primal feas.} & \iff & X = \frac{1}{n}I + \mathrm{Diag}\,(Vw) + \mathrm{sMat}_{\mathcal{E}^c}(v) \\ Z \succeq 0 & \text{dual feasible} & \iff & Z = -E + zI + \mathrm{sMat}_{\mathcal{E}}(y) \end{array}$$

solve the perturbed optimality condition $ZX - \mu I = 0$

Jacobian

$J := G'_\mu = [\mathcal{Z} \mid \mathcal{X}]$
where: $\mathcal{Z}(w,v) = Z\left(\mathrm{Diag}\,(Vw) + \mathrm{sMat}_{\mathcal{E}^c}(v)\right)$
$\mathcal{X}(z,y) = (zI + \mathrm{sMat}_{\mathcal{E}}(y))\,X$

Numerics

Three versions of the Gauss-Newton algorithm

- general version for full matrices
- sparse version for sparse matrices
- specialized version for the Lovász theta function problem

Three different preconditioners

- diagonal
- two block diagonal
- multiple block diagonal

Random Well-Conditioned Instances

Sparse version

Data inputs are $C \in \mathcal{S}^n$, $A_i \in \mathcal{S}^n$, $i = 1, \dots, m$, $b \in \mathbb{R}^m$; coded in MATLAB; G-N directions computed using LSMR by Fong-Saunders10[3] and LSQR Paige-Saunders82[11].
$n = 10 : 100$, $m = \lceil n(n+1)/4 \rceil$; sparseness density $1/(n+m)$; tolerance $\epsilon = 10^{-12}$; crossover is used.

different versions of the code with different preconditioners

without preconditioner (**SRSD**$_{\mathrm{o}}$);
diagonal preconditioner (**SRSD**$_{\mathrm{Diag}}$);
block preconditioner (**SRSD**$_{\mathrm{BDiag}}$).

Random well conditioned instances

Reduction in iterations; tradeoff – diagonal precond.

n	10	20	30	40	50	60	70	80	90	100
(**SRSD**$_{\mathrm{Diag}}$)	0.67	0.55	0.55	0.39	0.42	0.42	0.47	0.36	0.34	0.32
(**SRSD**$_{\mathrm{BDiag}}$)	0.30	0.16	0.13	0.09	0.09	0.08	0.08	0.07	0.07	0.06

Table: Ratios of numbers of LSMR iterations of (**SRSD**$_{\mathrm{Diag}}$) and (**SRSD**$_{\mathrm{BDiag}}$) to (**SRSD**$_{\mathrm{o}}$)

n	10	20	30	40	50	60	70	80	90	100
(**SRSD**$_{\mathrm{Diag}}$)	0.63	0.58	0.59	0.45	0.49	0.51	0.62	0.45	0.48	0.44
(**SRSD**$_{\mathrm{BDiag}}$)	0.33	0.27	0.46	0.57	1.05	1.44	2.20	2.26	3.50	3.26

Table: Ratios of total computational time of (**SRSD**$_{\mathrm{Diag}}$) and (**SRSD**$_{\mathrm{BDiag}}$) to (**SRSD**$_{\mathrm{o}}$)

Accuracy/Random sparse instances

SRSD$_{\mathrm{Diag}}$ with crossover

$n = 100, m = 100$, ($N = 1000$ instances, log-average) tolerance 10^{-12}

$$\text{RelZXnorm} = \frac{\|ZX\|_F}{|C \cdot X| + 1}, \qquad \text{Relmineig} = \frac{\min\{\lambda_{\min}(X), \lambda_{\min}(Z)\}}{|C \cdot X| + 1}$$

Accuracy measures for random sparse instances

	Iteration	RelZXnorm	Relmineig
Average	18.66	2.62×10^{-15}	-8.79×10^{-16}
Best	14.00	2.78×10^{-16}	-3.86×10^{-17}
Worst	26.00	8.36×10^{-13}	-1.97×10^{-13}

Accuracy comparisons

20 random instances; four SDP solvers

SeDuMi 1.3, CSDP 6.1.0, SDPA 7.3.1, SDPT3 4.0.

tolerances set small; six DIMACS error measures [7]

$$
\begin{aligned}
\epsilon_1(X,y,Z) &= \frac{\|\mathcal{A}(X)-b\|_2}{1+\|b\|_\infty} \\
\epsilon_2(X,y,Z) &= \max\left\{0, \frac{-\lambda_{\min}(X)}{1+\|b\|_\infty}\right\} \\
\epsilon_3(X,y,Z) &= \frac{\|\mathcal{A}^{\mathrm{adj}}(y)+Z-C\|_F}{1+\|C\|_{\max}} \\
\epsilon_4(X,y,Z) &= \max\left\{0, \frac{-\lambda_{\min}(Z)}{1+\|C\|_{\max}}\right\} \\
\epsilon_5(X,y,Z) &= \frac{C\bullet X - b^T y}{1+|C\bullet X|+|b^T y|} \\
\epsilon_6(X,y,Z) &= \frac{Z\bullet X}{1+|C\bullet X|+|b^T y|}
\end{aligned}
$$

High accuracy in comparison with 4 solvers

Performance measures for random sparse instances

	($\mathbf{SRSD}_{\mathrm{Diag}}$)	SeDuMi	CSDP	SDPA	SDPT3
Iteration	23.45	19.15	16.50	16.30	24.50
RelZXnorm	7.18×10^{-15}	1.50×10^{-07}	8.92×10^{-08}	5.86×10^{-08}	1.01×10^{-10}
Relmineig	-1.12×10^{-15}	-8.72×10^{-15}	3.28×10^{-13}	1.46×10^{-13}	5.99×10^{-17}
DIMACS1	1.28×10^{-15}	7.93×10^{-11}	1.24×10^{-13}	2.48×10^{-13}	3.19×10^{-13}
DIMACS2	2.28×10^{-14}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
DIMACS3	9.39×10^{-16}	3.53×10^{-16}	1.64×10^{-08}	1.44×10^{-15}	8.72×10^{-14}
DIMACS4	3.22×10^{-15}	3.42×10^{-13}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
DIMACS5	1.19×10^{-15}	9.72×10^{-13}	1.54×10^{-09}	4.96×10^{-10}	1.10×10^{-13}
DIMACS6	4.65×10^{-16}	2.84×10^{-14}	1.16×10^{-09}	4.96×10^{-10}	2.01×10^{-13}
Time	61.89	1.23	0.60	0.74	1.13

Increasing number constraints

$n = 100$, $m = 500 : 5000$; sparseness density $1/(4n)$;
block diagonal preconditioner

Random instances; varying number constraints

	($\mathbf{SRSD}_{\mathrm{BDiag}}$)	SeDuMi	CSDP	SDPA	SDPT3
Iteration	14.82	15.73	15.36	16.00	21.09
RelZXnorm	6.93×10^{-15}	1.99×10^{-07}	1.91×10^{-07}	1.77×10^{-08}	2.27×10^{-09}
Relmineig	-3.69×10^{-16}	-4.94×10^{-15}	4.06×10^{-13}	9.06×10^{-14}	2.31×10^{-16}

Time ratios relative to ($\mathbf{SRSD}_{\mathrm{BDiag}}$); varying number constraints

m	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
SeDuMi	168.74	49.99	21.47	4.18	1.92	1.89	1.09	0.66	0.68	0.32
CSDP	144.02	39.62	21.64	7.58	5.89	5.56	3.53	2.40	2.88	1.62
SDPA	488.28	203.06	134.49	50.99	37.09	33.37	20.98	15.55	17.83	9.41
SDPT3	87.49	27.96	16.10	6.00	3.80	4.42	2.11	2.30	2.89	1.81

Varying sparsity

$n = 100, m = 2500$; varying density $1/(4sn), s = 1, \ldots, 10$

Average measures for random instances with different sparsity

	$(\mathbf{SRSD}_{\mathrm{Diag}})/2$	SeDuMi	CSDP	SDPA	SDPT3
Iteration	13.80	16.80	16.20	16.20	24.30
RelZXnorm	4.47×10^{-15}	5.59×10^{-08}	1.30×10^{-07}	3.80×10^{-09}	3.65×10^{-11}
Relmineig	-7.59×10^{-16}	-2.24×10^{-16}	4.96×10^{-13}	8.71×10^{-14}	8.51×10^{-18}

Time ratios relative to $(\mathbf{SRSD}_{\mathrm{Diag}})/2$ for different sparsity

s	1	2	3	4	5	6	7	8	9	10
SeDuMi	2.84	2.74	1.26	1.15	1.49	0.62	0.53	0.49	0.34	0.30
CSDP	7.33	12.17	6.07	6.77	8.06	4.14	3.21	3.39	2.49	2.17
SDPA	43.35	48.63	22.13	20.95	25.47	12.33	9.25	9.82	7.08	6.04
SDPT3	4.60	8.82	5.79	5.74	7.50	3.94	2.86	2.75	2.27	2.09

Random Ill-Conditioned Instances

Strict complementarity fails

code for problem generation WW06 [13]; $n = 50, m = 1000$;
block preconditioner without crossover; tolerance 10^{-14}.

GN-method significant accuracy increase $\|ZX\|$, RelZXnorm

	($\mathbf{GRSD}_{\mathrm{BDiag}}$)	SeDuMi	CSDP	SDPA	SDPT3
Iteration	29.20	13.60	12.10	12.00	22.70
RelZXnorm	1.32×10^{-12}	4.12×10^{-06}	6.53×10^{-05}	1.91×10^{-07}	2.97×10^{-08}
Relmineig	5.96×10^{-20}	6.20×10^{-14}	1.44×10^{-15}	3.25×10^{-12}	1.21×10^{-18}
DIMACS1	2.16×10^{-12}	2.01×10^{-07}	3.50×10^{-10}	3.79×10^{-07}	4.62×10^{-10}
DIMACS2	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
DIMACS3	1.81×10^{-11}	6.34×10^{-15}	2.10×10^{-08}	2.09×10^{-14}	1.54×10^{-11}
DIMACS4	0.00×10^{00}	1.48×10^{-14}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
DIMACS5	2.07×10^{-12}	9.23×10^{-09}	1.23×10^{-08}	1.58×10^{-07}	5.70×10^{-11}
DIMACS6	1.86×10^{-12}	4.99×10^{-09}	4.25×10^{-08}	3.50×10^{-07}	1.06×10^{-10}
Time	281.45	64.02	69.50	27.31	60.25

Table: Performance measures; random instances; strict complementarity fails

Random Ill-Conditioned Instances

dual Slater's CQ (almost) fails

	(**GRSD**$_{\mathrm{BDiag}}$)	SeDuMi	CSDP	SDPA	SDPT3
Iteration	26.00	17.10	13.80	15.20	20.40
RelZXnorm	9.90×10^{-16}	6.14×10^{-07}	4.61×10^{-07}	2.05×10^{-07}	2.13×10^{-08}
Relmineig	1.05×10^{-17}	1.06×10^{-15}	1.60×10^{-12}	1.06×10^{-10}	1.47×10^{-15}
DIMACS1	2.78×10^{-13}	1.07×10^{-10}	1.10×10^{-12}	1.17×10^{-13}	8.41×10^{-11}
DIMACS2	0.00×10^{00} (*)	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
DIMACS3	1.42×10^{-12}	3.56×10^{-14}	5.21×10^{-09}	6.83×10^{-08}	2.59×10^{-13}
DIMACS4	0.00×10^{00} (*)	1.48×10^{-14}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}
DIMACS5	1.09×10^{-14}	2.77×10^{-12}	1.38×10^{-09}	2.02×10^{-14}	1.26×10^{-11}
DIMACS6	3.44×10^{-15}	3.16×10^{-12}	8.54×10^{-10}	4.43×10^{-08}	1.82×10^{-12}
Time	58.04	75.02	76.42	32.32	48.98

Random Instances; Lovász Theta Function

random graph with $n = 100$ nodes; $n(n-1)/4$ (approx) edges

	Iteration	RelZXnorm	Relmineig
Average	18.31	4.85×10^{-15}	2.47×10^{-13}
Best	16.00	6.42×10^{-17}	-1.28×10^{-15}
Worst	23.00	9.74×10^{-13}	-3.58×10^{-11}

Performance measures for random Lovász theta function

	SeDuMi	CSDP	SDPA	SDPT3	(**TRSD**$_{\mathrm{MBDiag}}$)
Iteration	21.00	17.10	16.00	22.60	21.00
RelZXnorm	5.00×10^{-08}	1.56×10^{-08}	2.38×10^{-07}	1.73×10^{-11}	2.67×10^{-14}
Relmineig	3.34×10^{-14}	7.09×10^{-14}	3.49×10^{-12}	2.32×10^{-18}	6.74×10^{-13}
DIMACS1	1.84×10^{-11}	6.07×10^{-14}	5.09×10^{-14}	1.27×10^{-13}	1.25×10^{-16}
DIMACS2	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	3.11×10^{-15}
DIMACS3	9.09×10^{-16}	1.09×10^{-07}	1.52×10^{-14}	8.87×10^{-13}	8.75×10^{-15}
DIMACS4	2.39×10^{-13}	0.00×10^{00}	0.00×10^{00}	0.00×10^{00}	4.82×10^{-12}
DIMACS5	9.09×10^{-12}	4.44×10^{-10}	9.61×10^{-09}	1.47×10^{-13}	5.92×10^{-16}
DIMACS6	9.93×10^{-14}	2.10×10^{-10}	9.61×10^{-09}	7.08×10^{-14}	4.19×10^{-16}
Time	90.46	13.05	6.02	12.91	158.09

Summary

- Current p-d i-p methods for SDP have two basic steps that lead to ill-conditioning: (i) symmetrize to be able to apply Newton's method; (ii) block elimation. (Implies difficulty for: accurate solutions; exploit sparsity.)
- We avoid these two steps using stable block elimination steps and applying Gauss-Newton method
- The numerical tests illustrate that we get improved accuracy and can exploit sparsity.
- Future research: To improve computation time, we need to exploit structure and find further stable eliminations.

F. Alizadeh, J-P.A. Haeberly, and M.L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Math. Programming **77** (1997), 111–128.

E. de Klerk, J. Peng, C. Roos, and T. Terlaky, *A scaled Gauss-Newton primal-dual search direction for semidefinite optimization*, SIAM J. Optim. **11** (2001), no. 4, 870–888 (electronic). MR 2002g:90070

D. C.-L. Fong and M. A. Saunders, *LSMR: An iterative algorithm for sparse least-squares problems*, Report SOL 2010-2, Systems Optimization Laboratory, Stanford University, 2010, to appear, SIAM J. Sci. Comp.

M. Gonzalez-Lima, H. Wei, and H. Wolkowicz, *A stable primal-dual approach for linear programming under nondegeneracy assumptions*, Comput. Optim. Appl. **44** (2009), no. 2, 213–247.

- C. Helmberg, F. Rendl, R.J. Vanderbei, and H. Wolkowicz, *An interior-point method for semidefinite programming*, SIAM J. Optim. **6** (1996), no. 2, 342–361. MR 97f:90086

- M. Kojima, S. Shindoh, and S. HARA, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim. **7** (1997), no. 1, 86–125. MR 97m:90088

- H.D. Mittelmann, *An independent benchmarking of SDP and SOCP solvers*, Math. Program. **95** (2003), no. 2, Ser. B, 407–430, Computational semidefinite and second order cone programming: the state of the art. MR MR1976487 (2004c:90033)

- R.D.C. Monteiro, *Primal-dual path-following algorithms for semidefinite programming*, SIAM J. Optim. **7** (1997), no. 3, 663–678.

Y.E. Nesterov and M.J. Todd, *Self-scaled barriers and interior-point methods for convex programming*, Math. Oper. Res. **22** (1997), no. 1, 1–42.

______, *Primal-dual interior-point methods for self-scaled cones*, SIAM J. Optim. **8** (1998), 324–364.

C.C. Paige and M.A. Saunders, *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software **8** (1982), no. 1, 43–71. MR 83f:65048

M.J. Todd, *A study of search directions in primal-dual interior-point methods for semidefinite programming*, Optim. Methods Softw. **11&12** (1999), 1–46.

H. Wei and H. Wolkowicz, *Generating and solving hard instances in semidefinite programming*, Math. Programming **125** (2010), no. 1, 31–45.

H. Wolkowicz, *Solving semidefinite programs using preconditioned conjugate gradients*, Optim. Methods Softw. **19** (2004), no. 6, 653–672. MR MR2102220 (2005h:90091)

Thanks for your attention!

A Robust Algorithm for Semidefinite Programming

Xuan Vinh Doan, Serge Kruk, Henry Wolkowicz

Dept. of Combinatorics and Optimization
University of Waterloo