A CONIC INTERIOR POINT DECOMPOSITION APPROACH FOR LARGE SCALE SEMIDEFINITE PROGRAMMING

Kartik Krishnan Sivaramakrishnan
Department of Mathematics
NC State University
kksivara@ncsu.edu
http://www4.ncsu.edu/~kksivara

Operations Research Colloquia
University of North Carolina at Chapel Hill
April 12, 2006

Contents

Conic programming

(P)	min	$c^T x$
	s.t.	$Ax = b$
		$x \in \mathcal{K}$

(D)	max	$b^T y$
	s.t.	$A^T y + s = c$
		$s \in \mathcal{K}$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $\mathcal{K} = \mathcal{K}_1 \times \ldots \times \mathcal{K}_r$

- $r = 1, \mathcal{K} = \mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x \geq 0\}$ **LP**
  Very large LPs ($m, n \leq$ **1,000,000**) solvable by the simplex method and/or IPMs.
- $\mathcal{K}_i = \mathbb{Q}^{n_i}_+ = \{x \in \mathbb{R}^{n_i} : x_1 \geq ||x_{2:n_i}||\}$ **SOCP**
  Large SOCPs ($m, n \leq$ **100,000**) solvable by IPMs.
- $\mathcal{K}_i = \mathcal{S}^{n_i}_+ = \{X \in \mathcal{S}^{n_i} : X \succeq 0\}$ **SDP**
  Medium sized SDPs ($m, n \leq$ **1000**) solvable by IPMs.
  (Beyond 10,000 seems impossible today!)

Motivation

(a) Solve **large scale structured semidefinite programs (SDP)** arising in science and engineering. Typically, these SDPs need not be solved very accurately.

(b) The technique is to **iteratively** solve an SDP between a **mixed conic master problem** over linear, second order, and semidefinite cones; and **distributed subproblems** (smaller SDPs) in a high performance computing environment.

(c) Improve the **scalability** of interior point methods (IPMs) by applying them instead on the smaller master problem, and subproblems (which are solved in parallel!)

This is our **conic interior point decomposition** scheme.

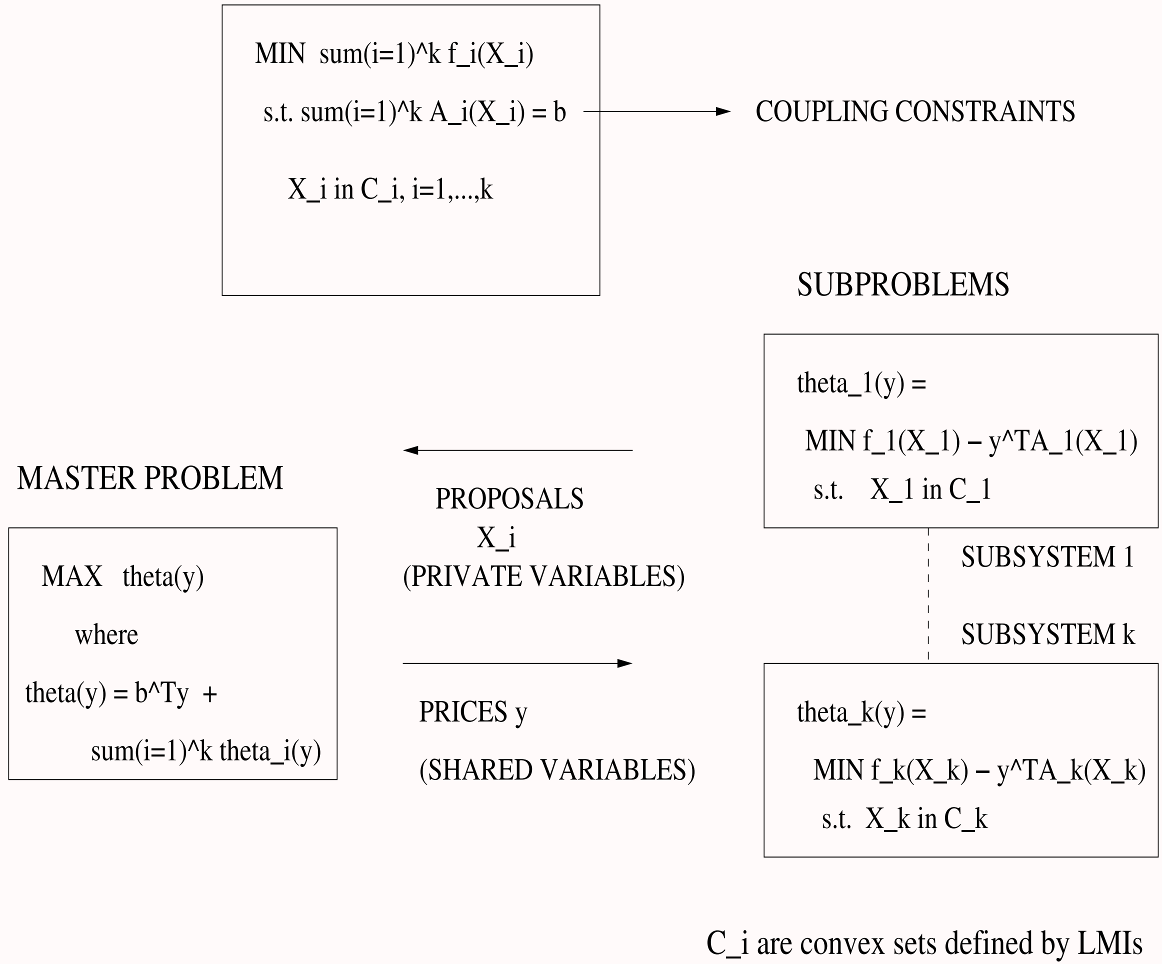


Figure 1: Decomposition by prices

Semidefinite programs with a decomposable structure

1. **Preprocessed SDPs after matrix completion:** Fukuda-Kojima-Nakata-Murota (2000), Lu-Nemirovskii-Monteiro (2004)
2. **Stability analysis of interconnected subsystems:** Langbort-D'Andrea-Xiao-Boyd (2003)
3. **Stochastic semidefinite programs:** Mehrotra-Ozëvin (2005)
4. **Polynomial optimization problems with structured sparsity:** Waki-Kim-Kojima-Muramatsu (2005)
5. **Semidefinite programs arising in lift and project schemes:** Rendl-Sotirov (2003) and Burer-Vandenbussche (2006)

Semidefinite programming

(SDP)	min	$C \bullet X$
	s.t.	$\mathcal{A}(X) = b$
		$X \succeq 0$

(SDD)	max	$b^T y$
	s.t.	$\mathcal{A}^T y + S = C$
		$S \succeq 0$

- **Notation**
  - $X, S, C \in \mathcal{S}^n$, $b \in \mathbb{R}^m$
  - $A \bullet B = \text{trace}(AB) = \sum_{i,j=1}^{n} A_{ij} B_{ij}$ (Frobenius inner product)
  - The operator $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ and its adjoint $\mathcal{A}^T : \mathbb{R}^m \to \mathcal{S}^n$ are

$$\mathcal{A}(X) = \begin{pmatrix} A_1 \bullet X \\ \vdots \\ A_m \bullet X \end{pmatrix}, \quad \mathcal{A}^\top y = \sum_{i=1}^{m} y_i A_i$$

where $A_i \in \mathcal{S}^n$, $i = 1, \ldots, m$

Assumptions

(1) $A_1, \ldots, A_m$ are **linearly independent** in $\mathcal{S}^n$.

(2) **Slater condition** for (SDP) and (SDD):

$$\{X \in \mathcal{S}^n : \mathcal{A}(X) = b, X \succ 0\} \neq \emptyset$$

$$\{(y, S) \in \mathbb{R}^m \times \mathcal{S}^n : \mathcal{A}^T y + S = C, S \succ 0\} \neq \emptyset$$

(3) One of the constraints in (SDP) is $I \bullet X = 1$ (trace$(X) = 1$).

(Every SDP with a bounded primal feasible set can be reformulated to satisfy this assumption)

One sample SDP

1. Consider the following integer quadratic program

$$\begin{array}{ll} \text{min} & x^T L x \\ \text{s.t.} & x_i \in \{-1,1\}^n, \; i = 1,\ldots,n \end{array}$$

2. Setting $X = xx^T$, gives an **equivalent** formulation

$$\begin{array}{lrcl} \text{min} & L \bullet X & & \\ \text{s.t.} & X_{ii} & = & 1, \; i = 1,\ldots,n \\ & X & \succeq & 0 \\ & \text{rank}(X) & = & 1 \end{array}$$

3. Dropping the rank constraint gives an SDP **relaxation**

$$\begin{array}{lrcl} \text{min} & L \bullet X & & \\ \text{s.t.} & X_{ii} & = & 1, \; i = 1,\ldots,n \\ & X & \succeq & 0 \end{array}$$

4. Note trace$(X) = n$ in this example.

Conic Dantzig-Wolfe decomposition

- Our semidefinite programs are:

$$\begin{array}{lll} \text{(SDP)} & \min & C \bullet X \\ & \text{s.t.} & \mathcal{A}(X) = b \\ & & \text{trace}(X) = 1 \\ & & X \succeq 0 \end{array}$$

$$\begin{array}{lll} \text{(SDD)} & \max & b^T y + z \\ & \text{s.t.} & \mathcal{A}^T y + zI + S = C \\ & & S \succeq 0 \end{array}$$

- Consider the set:

$$\mathcal{E} = \{X \in \mathcal{S}^n : \text{trace}(X) = 1, X \succeq 0\}$$

- The extreme points of $\mathcal{E}$ are:

$$\{vv^T : v \in \mathbb{R}^n, v^T v = 1\}$$ (Infinite number)

Conic Dantzig-Wolfe decomposition: Primal motivation

Any $X \in \mathcal{E}$ can be written as:

(1) $X = \sum_j \lambda_j d_j d_j^T$ where $\sum_j \lambda_j = 1, \lambda_j \geq 0$

$X \succeq 0$ is replaced by $\lambda_j \geq 0, \forall j$ (Semi-infinite LP formulation)

(2) $X = \sum_j P_j V_j P_j^T$ where $\sum_j$ trace$(V_j) = 1, V_j \in \mathcal{S}_+^2$

$X \succeq 0$ is replaced by $V_j \succeq 0, \forall j$ (Semi-infinite SOCP formulation)

(A 2×2 SDP constraint can be written as a size 3 SOCP constraint)

(3) $X = \sum_j P_j V_j P_j^T$ where $\sum_j$ trace$(V_j) = 1, V_j \in \mathcal{S}_+^{r_j}, r_j \geq 3$

$X \succeq 0$ is replaced by $V_j \succeq 0, \forall j$ (Semi-infinite SDP formulation)

A 2×2 SDP constraint is an SOCP constraint of size 3

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{12} & X_{22} \end{pmatrix} \succeq 0$$

$$\Leftrightarrow$$

$$X_{11} \geq 0, \;\; X_{22} \geq 0, \;\; X_{11}X_{22} - X_{12}^2 \geq 0$$

$$\Leftrightarrow$$

$$\begin{pmatrix} X_{11} + X_{22} \\ 2X_{12} \\ X_{11} - X_{22} \end{pmatrix} \succeq_Q 0$$

Conic Dantzig-Wolfe decomposition: Dual motivation

(1) LP formulation: $$S \succeq 0 \Longleftrightarrow d_j^T S d_j \geq 0, \forall j$$

(if we take a finite number of constraints we obtain an LP relaxation)

(2) SOCP formulation: $$S \succeq 0 \Longleftrightarrow \underbrace{P_j^T S P_j}_{2 \times 2} \succeq 0, \forall j$$

(finite number of constraints gives SOCP relaxation)

(3) SDP formulation: $$S \succeq 0 \Longleftrightarrow \underbrace{P_j^T S P_j}_{r_j \times r_j} \succeq 0, \forall j$$ $(r_j \geq 3, \ \forall j)$

(finite number of constraints gives SDP relaxation)

Conic Dantzig-Wolfe decomposition: Master problem

The **decomposed** conic problem over LP, SOCP and SDP blocks is:

Primal

$$
\begin{array}{llcl}
\min & \sum_{i=1}^{n_l} c_{li} x_{li} + \sum_{j=1}^{n_q} c_{qj}^T x_{qj} + \sum_{k=1}^{n_s} C_{sk} \bullet X_{sk} & & \\
\text{s.t.} & \sum_{i=1}^{n_l} A_{li} x_{li} + \sum_{j=1}^{n_q} A_{qj} x_{qj} + \sum_{k=1}^{n_s} \mathcal{A}_{sk}(X_{sk}) & = & b \\
& \sum_{i=1}^{n_l} x_{li} + \sum_{j=1}^{n_q} x_{qj1} + \sum_{k=1}^{n_s} \text{trace}(X_{sk}) & = & 1 \\
& x_{li} & \geq & 0, \quad i = 1, \ldots, n_l \\
& x_{qj} & \succeq_Q & 0, \quad j = 1, \ldots, n_q \\
& X_{sk} & \succeq & 0, \quad k = 1, \ldots, n_s
\end{array}
$$

Dual

$$
\begin{array}{llcl}
\max & b^T y + z & & \\
\text{s.t.} & A_{li}^T y + z + s_{li} & = & c_{li}, \quad i = 1, \ldots, n_l \\
& A_{qj}^T y + z \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + s_{qj} & = & c_{qj}, \quad j = 1, \ldots, n_q \\
& \mathcal{A}_{sk}^T y + z I_{r_k} + S_{sk} & = & C_{sk}, \quad k = 1, \ldots, n_s \\
& s_{li} & \geq & 0, \quad i = 1, \ldots, n_l \\
& s_{qj} & \succeq_Q & 0, \quad j = 1, \ldots, n_q \\
& S_{sk} & \succeq & 0, \quad k = 1, \ldots, n_s
\end{array}
$$

Conic Dantzig-Wolfe decomposition: Separation Oracle

Input: (y^*, z^*) a feasible point for dual master problem.

- If $\lambda_{\min}(S^*) \geq 0$, report feasibility STOP.
- Else, solve the following subproblem for X^*:

$$\begin{array}{lrcl} \text{min} & (C - \mathcal{A}^T y^* - z^* I) \bullet X & & \\ \text{s.t.} & I \bullet X & = & 1 \\ & X & \succeq & 0 \end{array}$$

Factorize $X^* = DMD^T$ with $M \succ 0$.
Return the cut $D^T(C - \mathcal{A}^T y - zI)D \succeq 0$ with $D \in \mathbb{R}^{n \times r}$.

– If $r = 1$ is an LP cut.

– If $r = 2$ is an SOCP cut.

– If $r \geq 3$ is an SDP cut of small size.

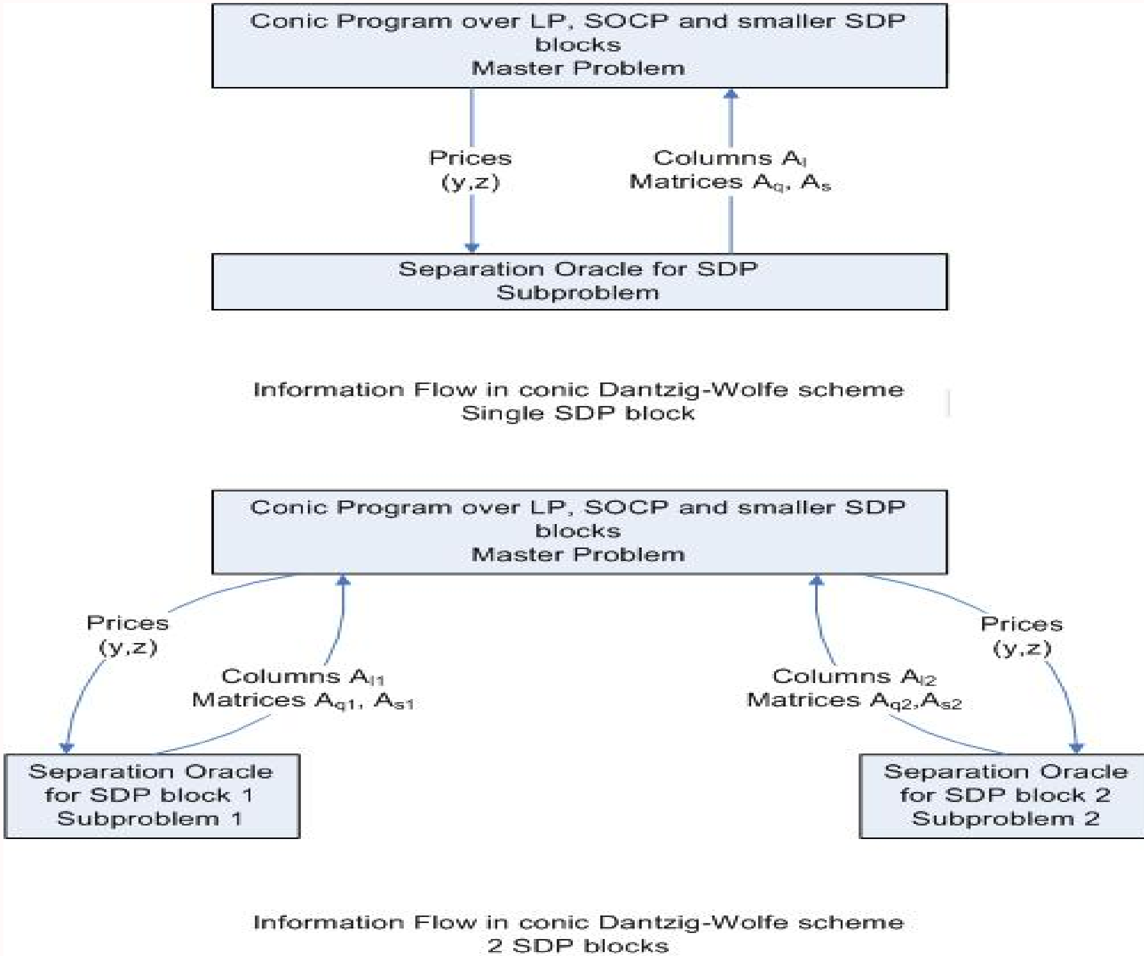


Figure 2: Conic Dantzig-Wolfe Algorithm

Choice of the query point: I

- In the original Dantzig-Wolfe (Kelley) scheme the query point (y, z) is an **optimal solution** to the dual master problem. This scheme has a **poor** rate of convergence.
- Better to solve the master problem approximately initially, and gradually tighten this tolerance as the algorithm proceeds.
  - Initially, the master problem is a poor approximation to the SDP problem.

$$\begin{aligned}\text{Weak tolerance} &\Rightarrow \text{Central dual iterates } (y, z) \\ &\Rightarrow \text{Oracle returns better cuts}\end{aligned}$$

  - As the algorithm proceeds, the master problem approximations get tighter.

$$\begin{aligned}\text{Tight tolerance} &\Rightarrow \text{More emphasis on the objective function} \\ &\Rightarrow \text{Convergence to an optimal solution}\end{aligned}$$

Choice of the query point : 2

Our adaptive strategy:

- Solve the master problem to a tolerance TOL for (x^*, y^*, z^*, s^*). Compute the following parameters:

$$\begin{aligned} \text{GAPTOL}(x^*, s^*) &= \frac{{x^*}^T s^*}{\max\{1, \frac{1}{2}(c^T x^* + b^T y^*)\}} \\ \text{INF}(y^*, z^*) &= \frac{\lambda_{\min}(C - \mathcal{A}^T y^* - z^* I)}{\max\{1, \frac{1}{2}(c^T x^* + b^T y^*)\}} \\ \text{OPT}(x^*, y^*, z^*, s^*) &= \max\{\text{GAPTOL}(x^*, s^*), \text{INF}(y^*, z^*)\} \end{aligned}$$

- If $\text{OPT}(x^*, y^*, z^*, s^*) < \text{TOL}$, we lower TOL by a constant factor. More precisely, $\text{TOL} = \mu \times \text{OPT}(x^*, y^*, z^*, s^*)$ with $0 < \mu < 1$. Else, TOL remains unchanged.

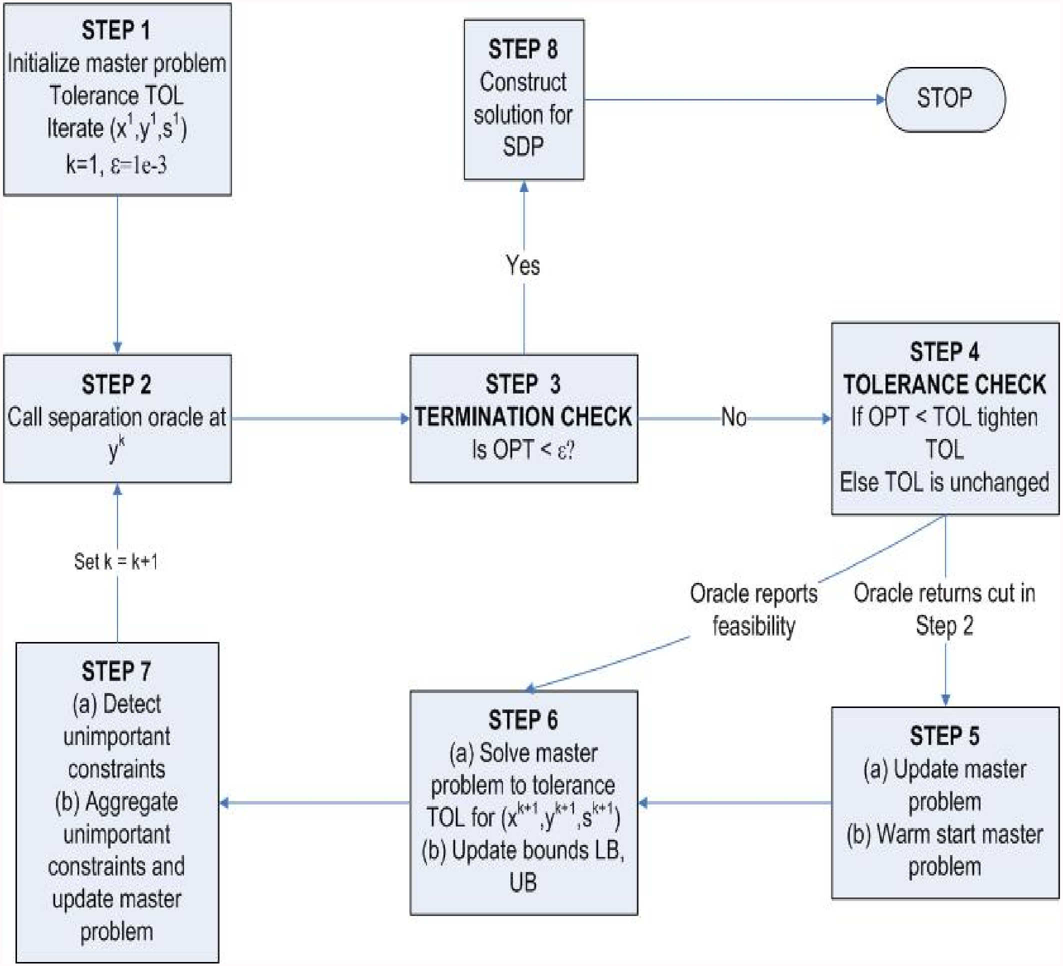


Figure 3: The complete algorithm

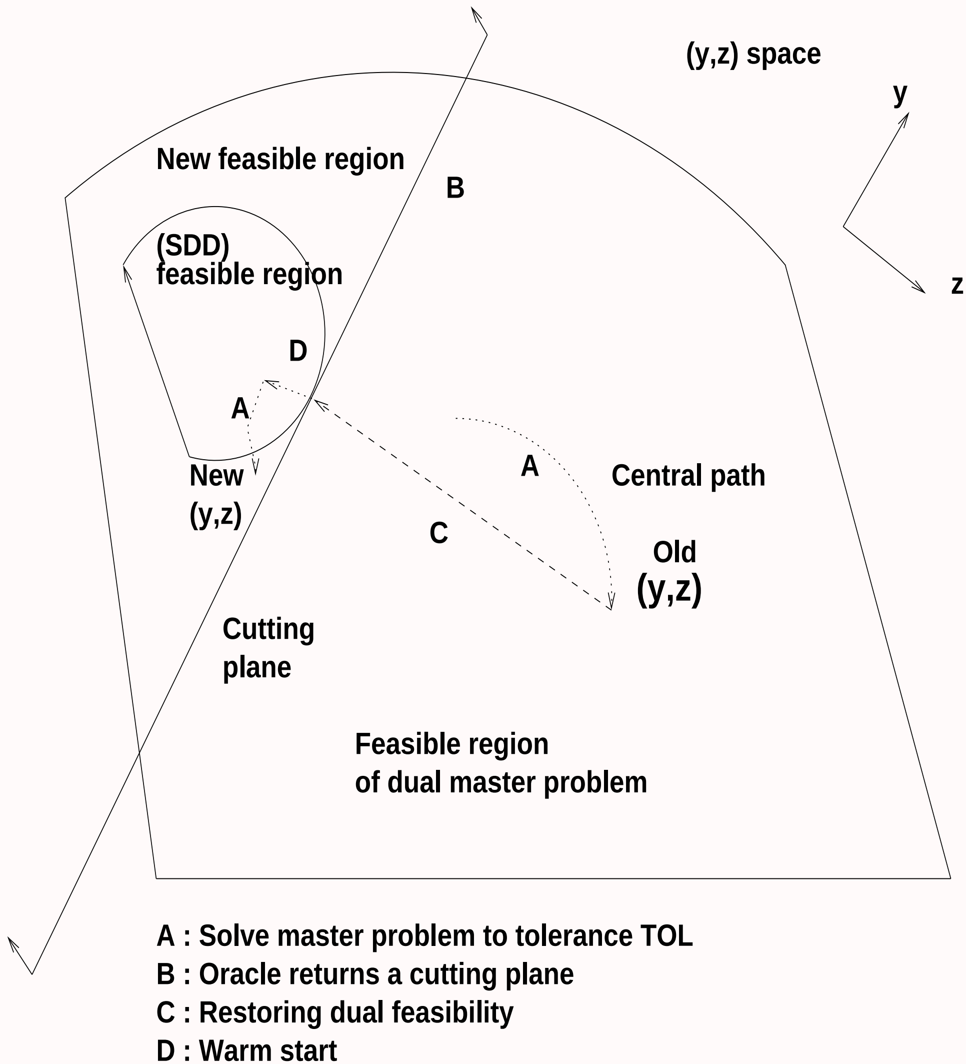


Figure 4: One iteration of algorithm

Upper and lower bounds

- The objective value of the primal master problem in every iteration gives an upper bound on the SDP objective value. This is the objective value of the dual master problem plus the duality gap.
- Given a solution (y^*, z^*) to the dual master problem in every iteration, a lower bound is computed as follows:
  - Compute $\lambda^* = \lambda_{\min}(S^*)$, where $S^* = (C - \mathcal{A}^T y^* - z^* I)$.
  - Set $y^{lb} = y^*$ and $z^{lb} = z^* + \lambda^*$.
  - A lower bound on the SDP objective value is then $b^t y^{lb} + z^{lb}$
    Note: (y^{lb}, z^{lb}) is a feasible point in (SDD).
- We could also terminate if the difference between these bounds is small.

Warm start after adding a column: 1

- The solution to the old master problem is $(x_l^*, x_s^*, y^*, s_l^*, s_s^*)$. Consider one linear cut $a_l^T y \le d$ in the dual master problem. The new master problem is :

$$\begin{array}{ll} \text{min} & c_l^T x_l + c_s^T x_s + d^T \beta \\ \text{s.t.} & A_l x_l + A_s x_s + a_l \beta = b \\ & x_l \ge 0, \quad \beta \ge 0 \\ & x_s \succeq 0. \end{array}$$

$$\begin{array}{ll} \text{max} & b^T y \\ \text{s.t.} & s_l = c_l - A_l^T y \ge 0 \\ & s_s = c_s - A_s^T y \succeq_Q 0 \\ & \gamma = d - a_l^T y \ge 0 \end{array}$$

- We have $\gamma^* < 0$. We can perturb y^* (lower bounding) to generate y^{lb} so that $\gamma^{lb} = 0$.
- The perturbed point $(x_l^*, x_s^*, \beta^*, y^{lb}, s_l^{lb}, s_s^{lb}, \gamma^{lb})$ is feasible in the new master problem with $\beta^* = \gamma^{lb} = 0$, but NOT STRICTLY!.
- We want to increase β^* and γ^{lb} from their current zero values, while limiting the variation in the other variables.

Warm start after adding a column: 2

- We solve the following problems

$$
\begin{array}{lll}
\text{(WSP)} & \max & \log\beta \\
& \text{s.t.} & A_l\Delta x_l + A_s\Delta x_s + a_l\beta = 0 \\
& & \sqrt{||D_l^{-1}\Delta x_l||^2 + ||D_s^{-1}\Delta x_s||^2} \leq 1 \\
& & \beta \geq 0.
\end{array}
$$

$$
\begin{array}{lll}
\text{(WSD)} & \max & \log\gamma \\
& \text{s.t.} & a_l^T\Delta y + \gamma = 0 \\
& & \sqrt{||D_l A_l^T\Delta y||^2 + ||D_s A_s^T\Delta y||^2} \leq 1 \\
& & \gamma \geq 0.
\end{array}
$$

  for $(\Delta x_l, \Delta x_s, \beta)$ and $(\Delta y, \gamma)$ respectively. Here D_l and D_s are appropriate primal-dual scaling matrices for linear and semidefinite blocks at $(x_l^*, x_s^*, s_l^{lb}, s_s^{lb})$.

- Compute $(\Delta s_l, \Delta s_s) = (-A_l^T\Delta y, -A_s^T\Delta y)$.

Warm start after adding a column: 3

- The solutions to (WSP) and (WSD) are given by

$$\begin{array}{rcl} \Delta x_l &=& -D_l^2 A_l^T (A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_l \beta \\ \Delta x_s &=& -D_q^2 A_q^T (A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_l \beta \\ \Delta y &=& -(A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_l \beta \\ \gamma &=& V \beta \end{array}$$

  where $\beta \in \mathbb{R}$ is the solution to

$$\min\{\tfrac{1}{2}\beta^T V \beta - \log \beta : \beta \geq 0\}$$

  where $V = a_l^T (A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_l$.

- Finally set

$$(x_l^{st}, x_s^{st}, \beta^{st}) = (x_l^* + \kappa \alpha_{max}^p \Delta x_l, x_s^* + \kappa \alpha_{max}^p \Delta x_s, \kappa \alpha_{max}^p \beta)$$
$$y^{st} = y_l^{lb} + \kappa \alpha_{max}^d \Delta y$$
$$(s_l^{st}, s_s^{st}, \gamma^{st}) = (s_l^{lb} + \kappa \alpha_{max}^d \Delta s_l, s_s^{lb} + \kappa \alpha_{max}^d \Delta s_s, \kappa \alpha_{max}^d \gamma)$$

  where $\kappa \in (0, 1)$ and α_{max}^p, α_{max}^d are the maximal primal and dual step lengths respectively.

Warm start after adding a matrix: 1

- The solution to the old master problem is $(x_l^*, x_s^*, y^*, s_l^*, s_s^*)$.
  Consider adding a semidefinite cut $a_s^T y \preceq d$ in the dual master problem.
  The new master problem is :

$$\begin{array}{ll} \text{min} & c_l^T x_l + c_s^T x_s + d^T \beta \\ \text{s.t.} & A_l x_l + A_s x_s + a_s \beta = b \\ & x_l \geq 0, \quad \beta \succeq 0 \\ & x_s \succeq 0. \end{array}$$

$$\begin{array}{ll} \text{max} & b^T y \\ \text{s.t.} & s_l = c_l - A_l^T y \geq 0 \\ & s_s = c_s - A_s^T y \succeq 0 \\ & \gamma = d - a_s^T y \succeq 0 \end{array}$$

- We have $\gamma^* \not\succeq 0$. We can perturb y^* (lower bounding) to generate y^{lb} so that $\gamma^{lb} \succeq 0$.
- The perturbed point $(x_l^*, x_s^*, \beta^*, y^{lb}, s_l^{lb}, s_s^{lb}, \gamma^{lb})$ is feasible in the new master problem with $\beta^* \succeq 0$ and $\gamma^{lb} \succeq 0$, but NOT STRICTLY!.
- We want to increase β^* and γ^{lb}, making them strictly feasible, while limiting the variation in the other variables.

Warm start after adding a matrix: 2

- We solve the following problems

$$
\text{(WSP)} \quad \begin{array}{ll} \max & \log\det \beta \\ \text{s.t.} & A_l \Delta x_l + A_s \Delta x_s + a_s \beta = 0 \\ & \sqrt{||D_l^{-1}\Delta x_l||^2 + ||D_s^{-1}\Delta x_s||^2} \leq 1 \\ & \beta \succeq 0. \end{array}
$$

$$
\text{(WSD)} \quad \begin{array}{ll} \max & \log\det \gamma \\ \text{s.t.} & a_s^T \Delta y + \gamma = 0 \\ & \sqrt{||D_l A_l^T \Delta y||^2 + ||D_s A_s^T \Delta y||^2} \leq 1 \\ & \gamma \succeq 0. \end{array}
$$

for $(\Delta x_l, \Delta x_s, \beta)$ and $(\Delta y, \gamma)$ respectively. Here D_l and D_s are appropriate primal-dual scaling matrices for LP and SDP blocks at $(x_l^*, x_s^*, s_l^{lb}, s_s^{lb})$.

- Compute $(\Delta s_l, \Delta s_s) = (-A_l^T \Delta y, -A_s^T \Delta y)$.

Warm start after adding a matrix: 3

- Compute

$$\begin{array}{rcl} \Delta x_l & = & -D_l^2 A_l^T (A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_s \beta \\ \Delta x_s & = & -D_s^2 A_q^T (A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_s \beta \\ \Delta y & = & -(A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_s \beta \\ \gamma & = & V \beta \end{array}$$

where $\beta \in \mathcal{S}^p$ is the solution to

$$\min\{\tfrac{p}{2}\beta^T V \beta - \log\det\beta : \beta \succeq 0\}$$

where $V = a_s^T (A_l D_l^2 A_l^T + A_s D_s^2 A_s^T)^{-1} a_s$.

- Finally set

$$(x_l^{st}, x_s^{st}, \beta^{st}) = (x_l^* + \kappa\alpha_{max}^p \Delta x_l, x_s^* + \kappa\alpha_{max}^p \Delta x_s, \kappa\alpha_{max}^p \beta)$$
$$y^{st} = y_l^{lb} + \kappa\alpha_{max}^d \Delta y$$
$$(s_l^{st}, s_s^{st}, \gamma^{st}) = (s_l^{lb} + \kappa\alpha_{max}^d \Delta s_l, s_s^{lb} + \kappa\alpha_{max}^d \Delta s_s, \kappa\alpha_{max}^d \gamma)$$

where $\kappa \in (0,1)$ and $\alpha_{max}^p, \alpha_{max}^d$ are the maximal primal and dual step lengths respectively.

Aggregating unimportant blocks

- Use a primal-dual scaling measure to evaluate blocks:
  - For the ith linear block the measure is $\frac{x_i}{s_i}$.
  - For the jth semidefinite block the measure is $\frac{\text{trace}(x_j)}{\text{trace}(s_j)}$.
- Blocks with small measure are aggregated in an LP block.
- The resulting master problem is

$$
\begin{array}{ll}
\text{min} & c_{agg}x_{agg} + c_l^T x_l + c_s^T x_s \\
\text{s.t.} & A_{agg}x_{agg} + A_l x_l + A_s x_s = b \\
 & x_l \geq 0, \quad x_{agg} \geq 0 \\
 & x_s \succeq 0.
\end{array}
$$

$$
\begin{array}{ll}
\text{max} & b^T y \\
\text{s.t.} & s_l = c_l - A_l^T y \geq 0 \\
 & s_s = c_s - A_s^T y \succeq 0 \\
 & s_{agg} = c_{agg} - A_{agg}^T y \geq 0
\end{array}
$$

Computational Results

- Matlab code with the following features:
  - Master problem contains linear and semidefinite blocks only.
  - **SDPT3** is used as conic solver for master problem with H.K.M. scaling
  - Oracle is implemented using MATLAB's Lanczos solver (**eigs**), which computes the $r = \lfloor \frac{\sqrt{m}}{2} \rfloor$ most negative eigenvalues and eigenvectors of S in each iteration.
  - Solve the problem for m iterations, or until TOL $= 1e-3$, whichever comes earlier.
- All computational results on a 2.4 GHz processor with 1 GB of memory.

Computational results for Max-Cut

Problem	n	Opt value	LP cones	SDP cones	Lower bound	ϵ	Time (h:m:s)
mcp100	100	-226.16	101	152(31)	-226.32	1.6e-3	16
mcp124-1	124	-141.99	125	187(48)	-141.87	1.5e-3	34
mcp124-2	124	-269.88	125	173(36)	-270.05	1.4e-3	25
mcp124-3	124	-467.75	125	174(35)	-468.07	1.5e-3	22
mcp124-4	124	-864.41	125	189(41)	-864.99	1.5e-3	21
mcp250-1	250	-317.26	251	339(84)	-317.46	1.3e-3	2:03
mcp250-2	250	-531.93	251	343(51)	-532.35	1.7e-3	1:05
mcp250-3	250	-981.17	251	307(44)	-981.86	1.6e-3	49
mcp250-4	250	-1681.96	251	301(43)	-1683.1	1.6e-3	47
mcp500-1	500	-598.15	501	735(115)	-598.56	1.6e-3	12:29
mcp500-2	500	-1070.06	501	634(70)	-1070.70	1.5e-3	6:18
mcp500-3	500	-1847.97	501	570(57)	-1849.20	1.6e-3	4:33
mcp500-4	500	-3566.73	501	523(49)	-3569.10	1.6e-3	2:51
toruspm3-8-50	512	-527.80	513	681(84)	-528.10	1.4e-3	9:28
torusg3-8	512	-457.36	513	605(89)	-457.60	1.4e-3	13.11
maxG11	800	-629.16	801	734(84)	-629.39	1.4e-3	1:49:55
maxG51	1000	-4003.81	1001	790(54)	-4008.90	1.7e-3	32:24

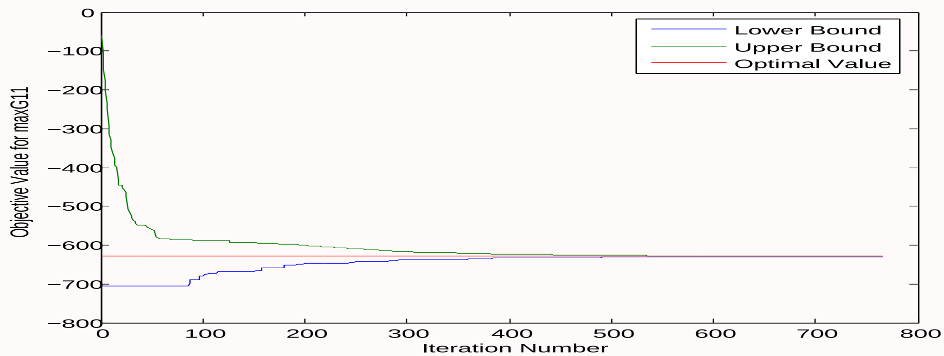


Figure 5: Variation of bounds for maxG11 problem

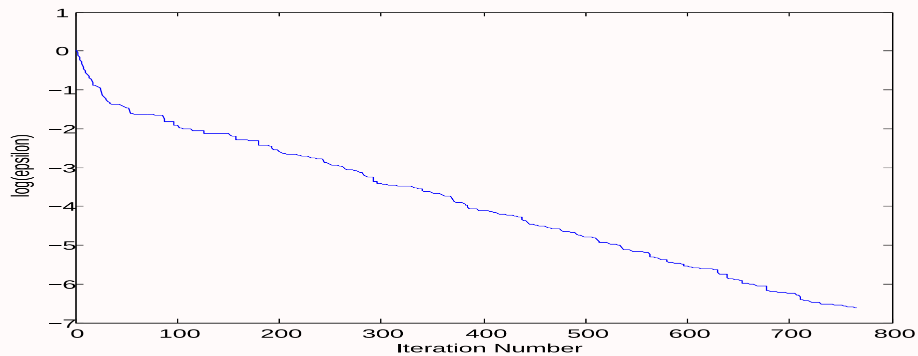


Figure 6: Variation of error for maxG11 problem

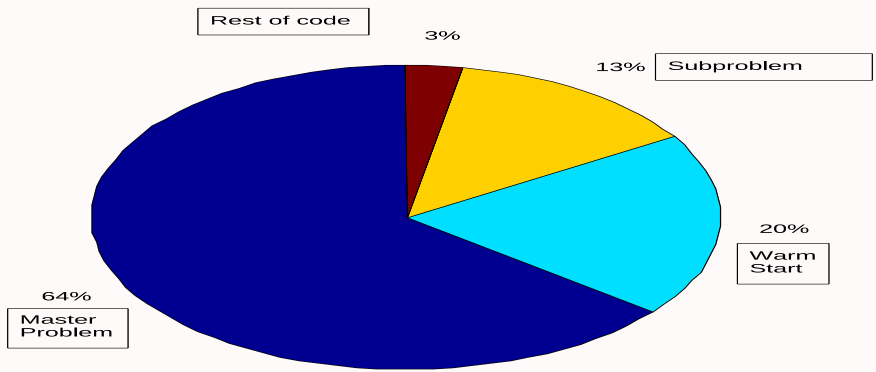


Figure 7: Distribution of times for maxG11 problem

Computational results for graph partitioning

Problem	n	Opt value	LP cones	SDP cones	Lower bound	ϵ	Time (h:m:s)
gpp100	100	39.91	103	80(23)	39.89	2e-3	50
gpp124-1	124	5.31	127	187(52)	5.26	2.2e-2	1:35
gpp124-2	124	42.35	127	121(39)	42.33	2.1e-3	1:24
gpp124-3	124	141.36	127	189(42)	141.29	1.4e-3	50
gpp124-4	124	387.7	127	188(39)	387.43	1.4e-3	43
gpp250-1	250	12.25	253	231(66)	12.18	1.7e-2	10:24
gpp250-2	250	73.33	253	182(49)	73.28	2.5e-3	10:11
gpp250-3	250	278.72	253	321(50)	278.53	1.6e-3	3:55
gpp250-4	250	695.83	251	358(53)	695.42	1.3e-3	3:49
gpp500-1	500	18.78	503	299(92)	18.67	2e-2	1:48:23
gpp500-2	500	138.98	503	277(65)	138.89	2.5e-3	2:11:08
gpp500-3	500	476.96	503	522(63)	476.72	1.3e-3	40:21
gpp500-4	500	1467.92	503	592(56)	1466.90	1.6e-3	26:22
bm1	882	17.65	885	1272(339)	17.47	4.3e-2	26:31:13

Computational results for Lovasz theta

Problem	n	m	Opt value	LP blocks	SDP blocks	Lower bound	ϵ	Time (h:m:s)
theta-1	50	104	-23	208	141(35)	-23.01	1.7e-3	1:09
theta-2	100	498	-32.87	996	396(37)	-32.90	1.5e-3	38:20
theta-3	150	1106	-42.17	2212	608(40)	-42.23	3e-3	5:15:52

Computational results on random semidefinite programs

$$\begin{array}{ll} \max & b^T y + z \\ \text{s.t.} & S = (C - \mathcal{A}^T y - zI) \succeq 0 \\ & l \le y \le u. \end{array}$$

Prob	n	m	Den	LP cones	SDP cones	Our LB	Our (h:m:s)	SDPT3 LB	SDPT3 (h:m:s)
test-1	100	501	0.1	1100	45(9)	424.68	1:06	424.97	43
test-2	100	501	0.9	1100	140(28)	181.91	5:21	182.02	1:32
test-3	500	101	0.1	700	50(25)	33.36	4:33	33.38	4:31
test-4	500	101	0.9	700	116(29)	-37.99	4:49	-37.96	6:35
test-5	1000	51	0.1	1100	24(12)	7.99	18:40	7.99	16:03
test-6	1000	51	0.9	1100	86(19)	-13.33	13:25	-13.32	17.30
test-7	300	300	0.1	900	105(21)	147.51	2:39	147.56	4:13
test-8	300	300	0.9	900	142(18)	22.91	4:52	22.92	8:11

Ongoing and Future work

- Applying the conic interior point decomposition approach for structured block angular SDPs in a parallel computing environment.

  (an ongoing project with Brian Borchers at New Mexico Tech)

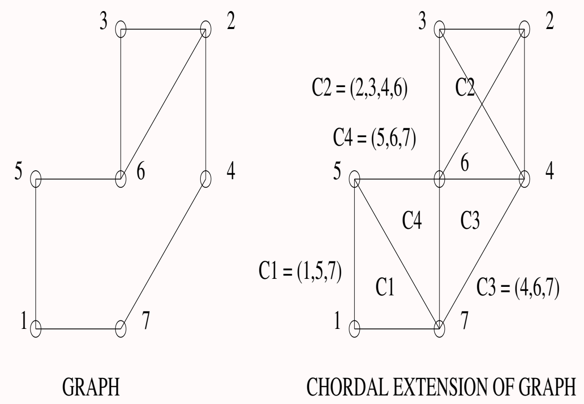
- Incorporating the decomposition scheme in the pricing phase of an SDP based conic branch-cut-price scheme for mixed integer and non-convex problems.

Preprocessing sparse SDPs into block-angular SDPs: 1

Consider the SDP

$$\begin{array}{lrcl} \min & L \bullet X & & \\ \text{s.t.} & X_{ii} & = & 1, \quad i = 1, \ldots, n, \\ & X & \succeq & 0, \end{array}$$

where L is the adjacency matrix of the graph



Preprocessing sparse SDPs into block-angular SDPs: 2

Using **matrix completion**, one can reformulate the earlier SDP as

$$
\begin{array}{lrcl}
\min & \sum_{k=1}^{4} (L^k \bullet X^k) & & \\
\text{s.t.} & X^1_{23} - X^4_{13} & = & 0, \\
 & X^2_{34} - X^3_{12} & = & 0, \\
 & X^3_{23} - X^4_{23} & = & 0, \\
 & X^k_{ii} & = & 1, \;\; i = 1, \ldots, |C_k|, \;\; k = 1, \ldots, 4, \\
 & X^k & \succeq & 0, \;\; k = 1, \ldots, 4,
\end{array}
$$

which is in **block-angular** form.

Thank you for your attention!.

Questions, Comments, Suggestions ?

The slides from this talk are available online at

http://www4.ncsu.edu/∼kksivara/publications/unc-or.pdf

A technical report appears at

http://www4.ncsu.edu/∼kksivara/publications/conic-ipm-decomposition.pdf

Bibliography: 1

1. K. Sivaramakrishnan, *A conic interior point decomposition for large scale structured block angular semidefinite programs*, forthcoming.

2. K. Sivaramakrishnan, G. Plaza, and T. Terlaky, *A conic interior point decomposition approach for semidefinite programming*, submitted.
`http://www4.ncsu.edu/~kksivara/publications/conic-ipm-decomposition.pdf`

3. L. Lasdon, *Optimization Theory for Large Systems*, Dover Publications Inc., 2002.

4. A. Ruszczyński, *Nonlinear Optimization*, Princeton University Press, 2006.

5. J.B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer Verlag, New York, 1993.

6. R.H. Tütüncü, K.C. Toh, and M.J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT3*, Mathematical Programming, 95(2003), pp. 189-217.
`http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`

7. B. Borchers, *SDPLIB 1.2, A library of semidefinite programming test problems*, Optimization Methods and Software, 11, 1999, pp. 683-690. `http://infohost.nmt.edu/~sdplib/`

8. DIMACS 7TH challenge. `http://dimacs.rutgers.edu/Challenges/Seventh/Instances/`

9. M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion I: General framework*, SIAM Journal on Optimization, 11(2000), pp. 647-674.

Bibliography: 2

10. Z. Lu, A. Nemirovskii, and R.D.C. Monteiro, *Large scale semidefinite programming via saddle point mirror-prox algorithm*.
http://www.optimization-online.org/DB_HTML/2004/11/998.html

11. C. Langbort, R. D'Andrea, L. Xiao, and S. Boyd, *A decomposition approach to distributed analysis of networked systems*, Proceedings of the 43rd IEEE Conference on Decision and Control, pp. 3980-3985.
http://www.stanford.edu/∼boyd/distr_contr_cdc04.html

12. S. Mehrotra and M.G. Özevin, *On the implementation of interior point decomposition algorithms for two-stage stochastic conic programs*.
http://www.optimization-online.org/DB_HTML/2005/10/1237.html

13. H. Waki, S. Kim, M. Kojima, and M. Muramatsu, *Sums of squares and semidefinite programming relaxations for polynomial problems with structured sparsity*.
http://www.optimization-online.org/DB_HTML/2004/10/988.html

14. F. Rendl and R. Sotirov, *Bounds for the quadratic assignment problem using the bundle method*. http://www.optimization-online.org/DB_HTML/2003/08/712.html

15. S. Burer and D. Vandenbussche, *Solving lift-and-project relaxations of binary integer programs*, SIAM Journal on Optimization, 16(2006), pp. 726-750.

Bibliography: 3

16. M.R. Oskoorouchi and J.L.Goffin, *A matrix generation approach for eigenvalue optimization*. http://public.csusm.edu/oskoorouchi/accsm.pdf

17. C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM Journal on Optimization, 10(2000), pp. 673-696.

18. K. Sivaramakrishnan and J.E. Mitchell, *A semidefinite programming based polyhedral cut and price algorithm for the maxcut problem*, Computational Optimization and Applications, 33(2006), pp. 51-71.