A PARALLEL TWO-STAGE INTERIOR POINT DECOMPOSITION ALGORITHM FOR LARGE SCALE STRUCTURED SEMIDEFINITE PROGRAMS

Kartik K. Sivaramakrishnan
Research Division
Axioma, Inc., Atlanta
kksivara@axiomainc.com
http://www4.ncsu.edu/~kksivara

DOS Seminar
Industrial and Systems Engineering
Georgia Tech
November 18, 2008
Contents

• Semidefinite programs (SDP)
• Motivation for our two stage decomposition algorithm
• Two representative applications of SDP
• Block angular SDPs
• Phase 1: Preprocessing SDPs into block-angular SDPs
• Phase 2: Decomposition algorithm for block-angular SDPs
• Results on the distributed Henry2 cluster at NCSU
• Comparisons with the OpenMP version of CSDP
• Conclusions
Conic programming

(P) \[ \begin{align*} & \text{max} \quad c^T x \\ & \text{s.t.} \quad Ax = b \\ & \quad x \in \mathcal{K} \end{align*} \]

(D) \[ \begin{align*} & \text{min} \quad b^T y \\ & \text{s.t.} \quad A^T y - s = c \\ & \quad s \in \mathcal{K} \end{align*} \]

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

- \( r = 1, \ \mathcal{K} = \mathbb{R}^n_+ = \{ x \in \mathbb{R}^n : x \geq 0 \} \) \ L P
  Very large LPs \( (m, n \leq 1,000,000) \) solvable by the simplex method and/or IPMs.

- \( \mathcal{K}_i = \mathbb{Q}^{n_i}_+ = \{ x \in \mathbb{R}^{n_i} : x_1 \geq \|x_{2:n_i}\|\} \) \ S OC P
  Large SOCPs \( (m, n \leq 100,000) \) solvable by IPMs.

- \( \mathcal{K}_i = \mathbb{S}^{n_i}_+ = \{ X \in \mathbb{S}^{n_i} : X \succeq 0 \} \) \ S D P
  Medium sized SDPs \( (n \leq 5000 \) and \( m \leq 10,000) \) solvable by IPMs.
Semidefinite programming

\[
\begin{array}{l}
\text{(SDP)} \quad \max \ C \bullet X \\
\text{s.t.} \quad \mathcal{A}(X) = b \quad X \succeq 0
\end{array}
\quad \begin{array}{l}
\text{(SDD)} \quad \min \ b^T y \\
\text{s.t.} \quad \mathcal{A}^T y - S = C \quad S \succeq 0
\end{array}
\]

• Notation

- \( X, S, C \in S^n, b \in \mathbb{R}^m \)
- \( A \bullet B = \text{trace}(AB) = \sum_{i,j=1}^{n} A_{ij}B_{ij} \) (Frobenius inner product)
- The operator \( \mathcal{A} : S^n \to \mathbb{R}^m \) and its adjoint \( \mathcal{A}^T : \mathbb{R}^m \to S^n \) are
  \[
  \mathcal{A}(X) = \begin{pmatrix} A_1 \bullet X \\ \vdots \\ A_m \bullet X \end{pmatrix}, \quad \mathcal{A}^T y = \sum_{i=1}^{m} y_i A_i
  \]
  where \( A_i \in S^n, i = 1, \ldots, m \)
- The matrices \( X \) and \( S \) are required to be positive semidefinite.

-3-
Motivation

(a) Exploit \textit{symmetry/sparsity} in underlying SDP and preprocess it into an equivalent SDP having a \textit{block-angular} structure.

(c) Solve block-angular SDP \textit{iteratively} between a \textit{master problem} (quadratic program) and \textit{decomposed and distributed subproblems} (smaller SDPs) in a parallel computing environment.

(d) Improve the \textit{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 \textit{parallel interior point decomposition} algorithm.
Primal SDPs with block angular structure

\[
\begin{align*}
\max \quad & \sum_{i=1}^{r} C_i \cdot X_i \\
\text{s.t.} \quad & \sum_{i=1}^{r} A_i(X_i) = b \\
\quad & X_i \in C_i, \quad i = 1, \ldots, r
\end{align*}
\]

• Notes
- \( X_i, C_i \in S_{n_i}, \quad b \in \mathbb{R}^m, \quad \text{and} \quad A_i : S_{n_i} \to \mathbb{R}^m. \)
- \( C_i = \{ X_i : B_i(X_i) = d_i, X_i \succeq 0 \} \) - compact semidefinite feasibility sets.
- The objective function and coupling constraints are block separable.
- In the absence of coupling constraints, solve \( r \) independent problems

\[
\max_{X_i \in C_i} C_i \cdot X_i
\]
Dual SDPs with block angular structure

\[
\max \sum_{i=1}^{r} c_i^T y_i \\
\text{s.t. } \sum_{i=1}^{r} A_i y_i = b \\
y_i \in C_i, \ i = 1, \ldots, r
\]

- Notes

- \( y_i, c_i \in \mathbb{R}^{n_i}, b \in \mathbb{R}^m \), and \( A_i \in \mathbb{R}^{m \times n_i} \).
- \( C_i = \{ y_i : D_i - B_i^T y_i \succeq 0 \} \) - compact semidefinite feasibility sets where \( D_i \in \mathbb{S}^{m_i} \) and \( B_i^T : \mathbb{R}^{n_i} \to \mathbb{S}^{m_i} \).
- The objective function and coupling constraints are block separable.
- In the absence of coupling constraints, solve \( r \) independent problems

\[
\max_{y_i \in C_i} c_i^T y_i
\]
Application 1: Combinatorial Optimization

1. Consider the following integer quadratic program

\[
\begin{align*}
\text{max} & \quad x^T L x \\
\text{s.t.} & \quad x_i \in \{-1, 1\}^n, \ i = 1, \ldots, n
\end{align*}
\]

2. Setting \( X = xx^T \), gives an equivalent formulation

\[
\begin{align*}
\text{max} & \quad L \cdot X \\
\text{s.t.} & \quad X_{ii} = 1, \ i = 1, \ldots, n \\
& \quad X \succeq 0 \\
& \quad \text{rank}(X) = 1
\end{align*}
\]

3. Dropping the rank constraint gives an SDP relaxation

\[
\begin{align*}
\text{max} & \quad L \cdot X \\
\text{s.t.} & \quad X_{ii} = 1, \ i = 1, \ldots, n \\
& \quad X \succeq 0
\end{align*}
\]

4. Goemans and Williamson developed an 0.878 approximation algorithm for the maxcut problem which uses this SDP relaxation.
Application 2: Polynomial Optimization

1. Consider the POP

\[
\begin{align*}
\min & \quad -x_2 \\
\text{s.t.} & \quad 3 + 2x_2 - x_1^2 - x_2^2 \geq 0 \\
& \quad -x_1 - x_2 - x_1x_2 \geq 0 \\
& \quad 1 + x_1x_2 \geq 0
\end{align*}
\]

Global optimal solution is \( x^* = (-0.6180, 1.6180) \) with value \(-1.6180\).

2. The first SDP relaxation in Lasserre hierarchy (of order 1) is

\[
\begin{align*}
\min & \quad -y_{01} \\
\text{s.t.} & \quad \begin{bmatrix}
1 & y_{10} & y_{01} \\
y_{10} & y_{20} & y_{11} \\
y_{01} & y_{11} & y_{02}
\end{bmatrix} \succeq 0 \\
& \quad 3 + 2y_{01} - y_{20} - y_{02} \geq 0 \\
& \quad -y_{10} - y_{01} - y_{11} \geq 0 \\
& \quad 1 + y_{11} \geq 0.
\end{align*}
\]

Optimal objective value of the SDP is \(-2\).
From Henrion & Lasserre (IEEE Control Magazine)
Exploiting sparsity to get block-angular SDPs

Consider the SDP

$$\max \ L \bullet X$$

s.t. $$X_{ii} = 1, \ i = 1, \ldots, n,$$

$$X \succeq 0,$$

where $L$ is the adjacency matrix of the graph

GRAPH

CHORDAL EXTENSION OF GRAPH
Exploiting sparsity to get block-angular SDPs

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

\[
\max \sum_{k=1}^{4} (L^k \bullet X^k)
\]

subject to:

\[
X_{23}^1 - X_{13}^4 = 0,
\]

\[
X_{34}^2 - X_{12}^3 = 0,
\]

\[
X_{23}^3 - X_{23}^4 = 0,
\]

\[
X_{ii}^k = 1, \quad i = 1, \ldots, |C_k|, \quad k = 1, \ldots, 4,
\]

\[
X^k \succeq 0, \quad k = 1, \ldots, 4,
\]

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

-11-
Preprocessing algorithm - Phase I

1. Construct minimal chordal extension $G' = (V, E')$ and its graph $A(E')$ from $G = (V, E)$. Find maximal cliques $Cl_i$, $i = 1, \ldots, r$ in $G'$.

2. Construct block-angular SDP from clique information as follows:

(a) For each nonzero element $(i, j) \in A(E')$, let $\hat{r}(i, j) = \min\{s : (i, j) \in Cl_s \times Cl_s\}$ and let $i$ and $j$ be elements $p$ and $q$ in $Cl_{\hat{r}(i,j)}$. Set $C_{\hat{r}(i,j)}^{pq} = C_{ij}$ and $[A_k]_{\hat{r}(i,j)}^{pq} = [A_k]_{ij}$, $k = 1, \ldots, m_{org}$.

(b) Set $EC_{ij} = 1$, $i, j = 1, \ldots, n$. Repeat (b) for all pairs of cliques: Suppose $Cl_k$ and $Cl_l$ share an element $(i, j); and i and j are elements $p$ and $q$ in $Cl_k$; and elements $s$ and $t$ in $Cl_l$.

- If $(i, j) \in I$, add $X_{pq}^k = B_{ij}$ and $X_{st}^l = B_{ij}$ to new SDP. Update $EC_{ij} = EC_{ij} + 1$.
- If $(i, j) \in J$, add $X_{pq}^k - X_{st}^l = 0$ and $X_{pq}^k \leq E_{ij}$ and $X_{st}^l \leq E_{ij}$ to new SDP. Update $EC_{ij} = EC_{ij} + 1$.
- If $(i, j) \notin I \cup J$, add $X_{pq}^k - X_{st}^l = 0$ to new SDP.

(c) Set $C_{pq}^k = \frac{C_{ij}}{EC_{ij}}$, $p, q = 1, \ldots, |Cl_k|$, where $p$ and $q$ in $Cl_k$ are nodes $i$ and $j$ in $G' = (V, E')$. 
Preprocessing SDPs into block-angular form

1. Consider the SDP

\[
\begin{align*}
\text{max} & \quad C \cdot X \\
\text{s.t.} & \quad X_{ij} = B_{ij}, \quad (i, j) \in I, \\
& \quad X_{pq} \leq F_{pq}, \quad (p, q) \in J, \\
& \quad A_k \cdot X = b_k, \quad k = 1, \ldots, m_{\text{org}}, \\
& \quad X \succeq 0.
\end{align*}
\]

2. \( I \) and \( J \) are disjoint sets that include \( D = \{(1, 1), \ldots, (n, n)\} \).

3. Examples include SDP relaxations of maxcut, stable set, and box QPs.

4. Such SDPs can be processed into an equivalent SDP in primal block-angular form using our preprocessing scheme (Sivaramakrishnan 2007).
Exploiting symmetry to get block-angular SDPs

1. Consider the SDP where

\[
\begin{align*}
\max & \quad C \cdot X \\
\text{s.t.} & \quad A_i \cdot X = b_i, \quad i = 1, \ldots, 3 \\
& \quad X \succeq 0
\end{align*}
\]

where \( b_3 = 1 \), \( A_3 \) is the \( 3 \times 3 \) identity matrix, and

\[
C = \begin{pmatrix}
5 & 3 & 3 & 1 \\
3 & 5 & 3 & 1 \\
3 & 3 & 5 & 1 \\
1 & 1 & 1 & 5
\end{pmatrix}, \quad A_1 = \begin{pmatrix}
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{pmatrix}, \quad A_2 = \begin{pmatrix}
0 & 2 & 2 & 0 \\
2 & 0 & 2 & 0 \\
2 & 2 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}.
\]

2. The data matrices are unchanged under any permutation of the first 3 rows and columns. The solution matrix \( X \) also inherits this symmetry (Gatermann & Parrilo 2003).

3. Our SDP is invariant under the action of the permutation group \( S_3 \).
Exploiting symmetry to get block-angular SDPs

1. Consider \( a : (1, 2, 3) \rightarrow (2, 3, 1) \) in \( S_3 \) that can be represented as

\[
P(a) = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\] or \( P_1(a) = 1 \) or \( P_2(a) = \begin{pmatrix}
-\frac{1}{2} & \frac{\sqrt{3}}{2} \\
\frac{\sqrt{3}}{2} & -\frac{1}{2}
\end{pmatrix}
\)

where \( P_1(a) \) and \( P_2(a) \) are irreducible (dimensions 1 and 2).

2. Note \( P(a)^T C P(a) = C \), \( P(a)^T A_i P(a) = A_i \), and \( P(a)^T X P(a) = X \).

3. Compute a symmetry adapted basis \( T \) such that

\[
T^T P(a) T = 2 P_1(a) \oplus P_2(a) \quad \text{where} \quad T = \begin{pmatrix}
\frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} & 0 \\
\frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \\
0 & 1 & 0 & 0
\end{pmatrix}.
\]

(multiplicities of \( P_1(a) \) and \( P_2(a) \) are 2 and 1).
Exploiting symmetry to get block-angular SDPs

1. Replacing the matrices $C, A_i$ and $X$ in the original SDP with $T^TCT$, $T^TA_iT$ and $T^TXT$ we get an equivalent block-angular SDP

$$\max \begin{pmatrix} 11 & \sqrt{3} \\ \sqrt{3} & 5 \end{pmatrix} \bullet X^1 + 4X^2$$

$$\text{s.t.} \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix} \bullet X^1 - 4X^2 = b_2$$

$$X^i \in C_i, \; i = 1, 2 \text{ where}$$

$$X^1 = \begin{pmatrix} X_{11} + 2X_{12} & \sqrt{3}X_{14} \\ \sqrt{3}X_{14} & X_{44} \end{pmatrix}$$

$$X^2 = X_{11} - X_{12}$$

$$C_1 = \{ X^1 \in S^2 : 2\sqrt{3}X_{12} = b_1, \; I \bullet X^1 \leq 1, \; X^1 \succeq 0 \}$$

$$C_2 = \{ X^2 \in IR : 0 \leq X^2 \leq 0.5 \}$$

(size and the number of copies of the $i$th block correspond to the multiplicity and dimension of $i$th irreducible representation)
Generating block-angular SDP relaxations for POPs

1. Consider following POP whose feasible region is contained in the unit cube
   \[
   \begin{align*}
   \min & \quad x_1 x_2 + x_1 x_3 - x_1 - 0.5x_2 - 0.5x_3 \\
   \text{s.t.} & \quad (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.25 \\
   & \quad (x_1 - 0.5)^2 + (x_3 - 0.5)^2 \leq 0.25.
   \end{align*}
   \]

2. Set \( F_k \) contain variables that appear in the \( k \)th monomial of \( p(x) \).
3. Sets \( H_j \) contain all variables appearing in constraint polynomial \( g_j(x) \).
4. Construct CSP graph from POP, where \( V = \{1, 2, \ldots, n\} \) and
   \[
   E = \{(i, j) : i, j \in F_k \text{ for some } k \text{ or } i, j \in H_l \text{ for some } l\}
   \]
5. Let \( I_i, i = 1, \ldots, r \) be a set of maximal cliques of CSP graph.
   (find chordal extension of graph and then maximal cliques in extension)
   In our example, \( I_1 = \{1, 2\} \) and \( I_2 = \{1, 3\} \).
6. Let \( J_i, i = 1, \ldots, r \) be an index set of \( g_j(x) \) whose list of variables is entirely in \( I_i \).
   In our example, \( J_1 = \{1\} \) and \( J_2 = \{2\} \).
Generating block-angular SDP relaxations for POPs

The first SDP relaxation in the Waki et al. (2006) hierarchy is

\[
\begin{align*}
\min & \quad y_4^1 + y_4^2 - 0.5y_1^1 - 0.5y_2^1 - 0.5y_2^2 \\
\text{s.t.} & \quad \begin{bmatrix} 1 & y_1^1 & y_2^1 \\ y_1^1 & y_3^1 & y_4^1 \\ y_2^1 & y_4^1 & y_5^1 \\ 1 & y_1^2 & y_2^2 \\ y_1^2 & y_3^2 & y_4^2 \\ y_2^2 & y_4^2 & y_5^2 \end{bmatrix} \succeq 0 \\
& \quad y_1^1 + y_2^1 - y_3^1 - y_5^1 \geq 0.25 \\
& \quad y_1^2 + y_2^2 - y_3^2 - y_5^2 \geq 0.25 \\
& \quad y_1^1 - y_1^2 = 0 \\
& \quad y_1^1 - y_1^2 = 0 \\
& \quad y_j^i \geq 0, \quad i = 1, \ldots, 5, \quad j = 1, \ldots, 2 \\
& \quad y_j^i \leq 1, \quad i = 1, \ldots, 5, \quad j = 1, \ldots, 2.
\end{align*}
\]

which is in the dual block-angular form (see Sivaramakrishnan 2008).
Polynomial optimization

1. Consider the POP

$$\begin{align*}
\min & \quad p(x) \\
\text{s.t.} & \quad g_i(x) \geq 0, \; i = 1, \ldots, m
\end{align*}$$

whose feasible region $S$ is contained in the unit hypercube. Let $2d_0$ or $2d_0 - 1$ denote degree of $p(x) = \sum_{\alpha} p_{\alpha} x^{\alpha}$ where $\alpha \in \mathbb{Z}_+^n$, and $2d_i$ or $2d_i - 1$ denote degrees of $g_i(x) = \sum_{\alpha} (g_i)_{\alpha} x^{\alpha}$.

2. $\mu$ is a probability measure, $y_{\alpha} = \int x^{\alpha} d\mu$ is its sequence of moments, and $S^r = \{ \alpha \in \mathbb{Z}_+^n : \sum_i \alpha_i \leq r \}$.

3. Let $\alpha, \beta \in S^r$. $M_r(y)(\alpha, \beta) = y_{\alpha + \beta}$ is the truncated moment matrix, and $M_r(g \ast y)(\alpha, \beta) = \sum_{\gamma} g_{\gamma} y_{\alpha + \beta + \gamma}$ is a localizing matrix. The dimension $M_r(y)$ is $n+r \choose r$ and it contains $n+2r \choose 2r$ distinct variables.

4. Let $r \geq \max_{i=0,\ldots,m} d_i$. The $r$th Lasserre relaxation for POP is

$$\begin{align*}
\min & \quad \sum_{\alpha} p_{\alpha} y_{\alpha} \\
\text{s.t.} & \quad M_r(y) \succeq 0 \\
& \quad M_{r-d_j}(g_j \ast y) \succeq 0, \; j = 1, \ldots, m.
\end{align*}$$
Generating block-angular SDP relaxations for POPs

1. Let $I_i$, $i = 1, \ldots, r$ be maximal cliques in the CSP graph and $\mathcal{I}_i$ is the set of all subsets in $I_i$.

2. $S^r_{\mathcal{I}} = \{ \alpha \in \mathbb{Z}^n_+ : \text{supp}(\alpha) \in \mathcal{I}, \sum_i \alpha_i \leq r \}$.

3. $M_r(y, \mathcal{I})$ is a sparse truncated moment matrix and $M_r(g \ast y, \mathcal{I})$ is a sparse localizing matrix with rows indexed by $\alpha \in S^r_{\mathcal{I}}$.

4. Our contribution (Sivaramakrishnan 2008) is to write the $r$th sparse SDP relaxation in the Waki et al. (2006) hierarchy as

$$\begin{align*}
\min & \quad \sum_{i=1}^{r} f_i^T y_i \\
\text{s.t.} & \quad M_r(y_i, \mathcal{I}_i) \succeq 0, \ i = 1, \ldots, r \\
& \quad M_{r-ri_j}(g_{ij} \ast y_i, \mathcal{I}_i) \succeq 0, \ i = 1, \ldots, r, \ j \in J_i \\
& \quad \sum_{i=1}^{r} G_i y_i = 0 \\
& \quad y_i \geq 0, \ i = 1, \ldots, r \\
& \quad y_i \leq e, \ i = 1, \ldots, r.
\end{align*}$$

which is in the dual block-angular format.
The Lagrangian dual problem

- The Lagrangian dual problem is

\[
\min_y \, \theta(y) = b^T y + \sum_{i=1}^{r} \theta_i(y)
\]

where

\[
\theta_i(y) = \max_{X_i \in C_i} (C_i - A_i^T y) \cdot X_i
\]

- Dual is an unconstrained \textbf{convex} but \textbf{nonsmooth} problem.
- Given \(y^k\), we have \(\theta(y^k) = b^T y^k + \sum_{i=1}^{r} (C_i - A_i^T y^k) \cdot X_i^k\)

and a subgradient \(g(y^k) = (b - \sum_{i=1}^{r} A_i(X_i^k))\) where

\[
X_i^k = \arg\max_{X_i \in C_i} (C - A_i^T y^k) \cdot X_i
\]

(these can be computed in parallel!)

-21-
Solving the Lagrangian dual

1. Construct a model $\theta^k(y)$ for $\theta(y)$

$$
\theta^k(y) = b^Ty + \sum_{i=1}^{r} \max_{j=1,\ldots,J^k(i)} (C_i - A_i^T y) \cdot X^j_i
$$

from the function values and subgradient information.

2. $J^k(i) \subseteq \{1, 2, \ldots, k\}$ is the set of iteration indices when $\theta_i^k(y)$ is updated.

3. $\theta^k(y)$ is an **underestimate** for $\theta(y)$

4. The **regularized** master problem is

$$
\min_y \theta^k(y) + \frac{u^k}{2} ||y - x^k||^2
$$

where $u^k > 0$ (weight) and $x^k$ (current center) that are updated during the course of the algorithm.

-22-
Solving the Lagrangian dual

1. The master problem is the following QP
\[
\begin{align*}
\min & \quad b^T y + \sum_{i=1}^{r} z_i + \frac{u_k^2}{2} ||y - x^k||^2 \\
\text{s.t.} & \quad A_i(X^j_i)^T y + z_i \geq C_i \bullet X^j_i, \quad i = 1, \ldots, r, \quad j \in J^k(i).
\end{align*}
\]

2. The center $x^k$ is updated depending on the relation between $\theta(\boldsymbol{y}^k+1)$ and $\theta^k(\boldsymbol{y}^k+1)$ at the current solution $\boldsymbol{y}^k+1$ to master problem.

3. If these values are close, we set $x^{k+1} = \boldsymbol{y}^{k+1}$ (serious step). Else, $x^{k+1} = x^k$ (null step).

4. If $\theta_i(\boldsymbol{y}^{k+1}) = z^k_i$, then $\boldsymbol{y}^{k+1}$ is optimal in the $i$th subproblem and $\theta^k_i(\boldsymbol{y})$ is not updated in the $(k + 1)$th iteration.

5. Else, the $i$th subproblem adds the following cutting plane
\[
A_i(X^k_i)^T y + z_i \geq C_i \bullet X^k_i
\]
to the master problem.
Upper and lower bounds

1. The function value \( \theta(y^k) \) provides an upper bound on the optimal objective value.

2. The value \( \theta^k(y^{k+1}) \) computed from the solution \( y^{k+1} \) to the master problem provides a lower bound.

3. Our termination check in the \( k \) iteration is

   \[
   \frac{\theta(x^k) - \theta^k(y^{k+1})}{\max\{1, |\theta(x^k)|\}} \leq \epsilon
   \]

   where \( \epsilon > 0 \) is a suitable parameter. In our experiments \( \epsilon = 10^{-2} \) or \( 10^{-3} \).

4. When the termination criteria is satisfied, the solution to the subproblems for \( x^k \) provides an optimal solution to the block-angular SDP.
Decomposition algorithm for block-angular SDPs

1. **Initialize:** Set $k = 1$, $J^0(i) = \emptyset$, $z^0_i = -\infty$, $i = 1, \ldots, r$. Choose $m_L \in (0, 1)$, starting point $y^1 \in \mathbb{R}^m$, and set $x^0 = y^1$.

2. **Solve subproblems in parallel:** $X^k_i$ is solution to $i$th subproblem.

3. **Update master problem:** If $\theta_i(y^k) > z^k_i$, add constraint
   \[ A_i(X^k_i)^T y + z_i \geq C_i \cdot X^k_i \]
   to master problem and set $J^k(i) = J^{k-1}(i) \cup \{k\}$. Else, $J^k(i) = J^{k-1}(i)$.

4. **Update center $x^k$ and weight $u^k$:** If $k = 1$ or if
   \[ \theta(y^k) \leq (1 - m_L)\theta(x^{k-1}) + m_L\theta^{k-1}(y^k) \]
   then set $x^k = y^k$ (serious step). Else, set $x^k = x^{k-1}$ (null step).

5. **Solve the master problem:** Let $(y^{k+1}, z^{k+1})$ and $\lambda^k$ be primal and dual solutions. Let $\theta^k(y^{k+1}) = b^T y^{k+1} + \sum_{i=1}^r z^{k+1}_i$.

6. **Termination Check:** If $\theta^{k+1}(y^{k+1}) = \theta(x^k)$ stop!

7. **Aggregation:** Aggregate constraints in master problem with small $\lambda^k$. Set $k = k + 1$ and return to Step 2.
Figure 1: Decomposition by prices
1. The weight $u^k > 0$ serves as a regularization term in the master problem.

2. Think of $\frac{u^k}{2} ||y - x^k||^2$ in the objective rather as a trust region constraint $||y - x^k|| \leq \sigma^k$, where a large radius $\sigma^k$ corresponds to a small $u^k$ and vice-versa.

3. $u^k$ is dynamically updated during the algorithm to improve its convergence (Kiwiel (1990), Helmberg & Kiwiel (2002), Sivaramakrishnan (2008))

4. The idea is to increase $u^k$ after a null step and reduce it after a serious step. $u^k$ may also remain unchanged during an iteration.

5. Our starting choice is

$$u^1 = \frac{||g(y^1)||}{\sqrt{n}}.$$
Master problem aggregation

1. Enforce an upper bound on the number of constraints in the master problem (bundle size).

2. Aggregate less important constraints in the master problem with small dual multipliers $\gamma$. Let $J^k_{agg}(i)$ denote the set of these constraints for the $i$th account in the $k$th iteration.

3. Take a convex combination of these constraints in aggregate block constraints (one for each account) in the master problem.

4. The aggregated master problem is

$$\min \ b^T y + \sum_{i=1}^{r} z_i + \frac{u_k}{2}||y - x^k||^2$$

s.t. $A_i(X^j_i)^T y + z_i \geq C_i \bullet X^j_i, \ j \in (J^k(i) \cap J^k_{agg}(i))$,  
$A^agg_i y + z_i \geq C^agg_i, \ i = 1, \ldots, r.$
Setup for Computational Experiments

- MPI code is run on IBM Blade Center Linux Cluster (Henry2) at NC State: 175 nodes, each node is a dual 2.8-3.2GHz processor with 4GB of memory (2GB per processor).
- Our code is in C and uses MPI for communication between processors.
- CPLEX 10.0 solves master problem and CSDP 6.0 solves subproblems.
- Accuracy: $\epsilon = \frac{UB - LB}{1 + |UB|} < 10^{-2}$.
- Used procedure in Helmberg-Kiwiel to adjust weight $u$.
- Employed aggregation scheme of Kiwiel to limit size of master problem.
- SDPs are generated from graphs in Gset library. We exploit the sparsity in SDPs to get block-angular SDPs in Phase I of our algorithm.
- OpenMP version of CSDP is run on IBM Power5 shared memory system at NC State with eight 1.9GHz processors and 32GB of shared memory.
## Test Instances

<table>
<thead>
<tr>
<th>Prob</th>
<th>n</th>
<th>m</th>
<th>$n_p$</th>
<th>$m_p$</th>
<th>$n_{total}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxG57</td>
<td>5000</td>
<td>5000</td>
<td>405 (516)</td>
<td>61595 (51883)</td>
<td>9712</td>
</tr>
<tr>
<td>thetaG32</td>
<td>2001</td>
<td>6001</td>
<td>212 (80)</td>
<td>22774 (16762)</td>
<td>3187</td>
</tr>
<tr>
<td>thetaG48</td>
<td>3001</td>
<td>9001</td>
<td>441 (102)</td>
<td>39468 (30632)</td>
<td>4586</td>
</tr>
<tr>
<td>thetaG57</td>
<td>5001</td>
<td>15001</td>
<td>434 (182)</td>
<td>75369 (60711)</td>
<td>9313</td>
</tr>
<tr>
<td>maxG62</td>
<td>7000</td>
<td>7000</td>
<td>515 (692)</td>
<td>100873 (87232)</td>
<td>13641</td>
</tr>
<tr>
<td>maxG65</td>
<td>8000</td>
<td>8000</td>
<td>656 (791)</td>
<td>101300 (85961)</td>
<td>15339</td>
</tr>
<tr>
<td>maxG66</td>
<td>9000</td>
<td>9000</td>
<td>633 (898)</td>
<td>134904 (117344)</td>
<td>17560</td>
</tr>
<tr>
<td>maxG67</td>
<td>10000</td>
<td>10000</td>
<td>903 (942)</td>
<td>131593 (112532)</td>
<td>19061</td>
</tr>
<tr>
<td>maxG77</td>
<td>14000</td>
<td>14000</td>
<td>916 (1395)</td>
<td>219118 (191591)</td>
<td>27527</td>
</tr>
<tr>
<td>thetaG62</td>
<td>7001</td>
<td>21001</td>
<td>506 (540)</td>
<td>118828 (98180)</td>
<td>13496</td>
</tr>
<tr>
<td>thetaG65</td>
<td>8001</td>
<td>24001</td>
<td>687 (555)</td>
<td>134199 (110726)</td>
<td>14974</td>
</tr>
<tr>
<td>thetaG67</td>
<td>10001</td>
<td>30001</td>
<td>853 (694)</td>
<td>201570 (172268)</td>
<td>19028</td>
</tr>
<tr>
<td>thetaG77</td>
<td>14001</td>
<td>42001</td>
<td>782 (933)</td>
<td>263318 (222229)</td>
<td>26315</td>
</tr>
</tbody>
</table>
## Computational Results on instances

<table>
<thead>
<tr>
<th>Prob</th>
<th>Opt val</th>
<th>p</th>
<th>UB</th>
<th>Time (h:m:s)</th>
<th>rel_acc</th>
<th>Serious steps</th>
<th>Null steps</th>
<th>Master (h:m:s)</th>
<th>Subproblem (h:m:s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxG57</td>
<td>3875.98</td>
<td>16</td>
<td>3903.62</td>
<td>17:31</td>
<td>7.1e-3</td>
<td>16</td>
<td>46</td>
<td>10:23</td>
<td>7:06</td>
</tr>
<tr>
<td>thetaG32</td>
<td>1000.00</td>
<td>16</td>
<td>1004.93</td>
<td>3:58</td>
<td>4.9e-3</td>
<td>19</td>
<td>48</td>
<td>1:21</td>
<td>2:36</td>
</tr>
<tr>
<td>thetaG48</td>
<td>1500.00</td>
<td>16</td>
<td>1505.49</td>
<td>25:17</td>
<td>3.6e-3</td>
<td>18</td>
<td>70</td>
<td>3:42</td>
<td>21:33</td>
</tr>
<tr>
<td>thetaG57</td>
<td>2500.00</td>
<td>16</td>
<td>2509.78</td>
<td>44:00</td>
<td>3.9e-3</td>
<td>21</td>
<td>83</td>
<td>15:35</td>
<td>28:22</td>
</tr>
<tr>
<td>maxG62</td>
<td>5410.91</td>
<td>16</td>
<td>5452.37</td>
<td>48:40</td>
<td>7.6e-3</td>
<td>17</td>
<td>57</td>
<td>31:58</td>
<td>16:32</td>
</tr>
<tr>
<td>maxG65</td>
<td>6185.04</td>
<td>16</td>
<td>6227.18</td>
<td>42:58</td>
<td>6.8e-3</td>
<td>16</td>
<td>41</td>
<td>17:26</td>
<td>25:24</td>
</tr>
<tr>
<td>maxG66</td>
<td>7097.21</td>
<td>16</td>
<td>7144.64</td>
<td>1:15:39</td>
<td>6.6e-3</td>
<td>17</td>
<td>54</td>
<td>45:28</td>
<td>30:01</td>
</tr>
<tr>
<td>maxG67</td>
<td>7708.93</td>
<td>8</td>
<td>7764.16</td>
<td>1:31:25</td>
<td>7.1e-3</td>
<td>16</td>
<td>42</td>
<td>28:32</td>
<td>1:01:30</td>
</tr>
<tr>
<td>maxG77</td>
<td>11097.67</td>
<td>16</td>
<td>11187.50</td>
<td>2:32:35</td>
<td>8.0e-3</td>
<td>18</td>
<td>52</td>
<td>1:18:34</td>
<td>1:13:41</td>
</tr>
<tr>
<td>thetaG62</td>
<td>3500.00</td>
<td>16</td>
<td>3511.57</td>
<td>1:46:40</td>
<td>3.3e-3</td>
<td>21</td>
<td>104</td>
<td>54:34</td>
<td>51:58</td>
</tr>
<tr>
<td>thetaG65</td>
<td>4000.00</td>
<td>16</td>
<td>4014.76</td>
<td>3:25:14</td>
<td>3.6e-3</td>
<td>20</td>
<td>95</td>
<td>48:35</td>
<td>2:36:31</td>
</tr>
<tr>
<td>thetaG67</td>
<td>5000.00</td>
<td>16</td>
<td>5024.48</td>
<td>6:19:05</td>
<td>4.8e-3</td>
<td>22</td>
<td>118</td>
<td>2:41:26</td>
<td>3:37:25</td>
</tr>
<tr>
<td>thetaG77</td>
<td>7000.00</td>
<td>16</td>
<td>7028.62</td>
<td>6:29:00</td>
<td>4.0e-3</td>
<td>22</td>
<td>119</td>
<td>3:42:30</td>
<td>2:46:05</td>
</tr>
</tbody>
</table>
### Comparison with OpenMP version of CSDP

<table>
<thead>
<tr>
<th>Prob</th>
<th>rel_acc</th>
<th>Time (h:m:s)</th>
<th>Iter</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>p=1</td>
<td>p=2</td>
<td>p=4</td>
</tr>
<tr>
<td>thetaG32</td>
<td>5.0e-3</td>
<td>19:44</td>
<td>14:26</td>
<td>8:40</td>
</tr>
<tr>
<td>thetaG48</td>
<td>3.6e-3</td>
<td>56:39</td>
<td>54:25</td>
<td>48:04</td>
</tr>
<tr>
<td>maxG57</td>
<td>7.1e-3</td>
<td>42:36</td>
<td>37:36</td>
<td>32:18</td>
</tr>
<tr>
<td>thetaG57</td>
<td>3.9e-3</td>
<td>2:42:35</td>
<td>1:29:30</td>
<td>1:18:56</td>
</tr>
<tr>
<td>maxG62</td>
<td>7.7e-3</td>
<td>1:48:54</td>
<td>1:39:15</td>
<td>1:21:34</td>
</tr>
<tr>
<td>maxG65</td>
<td>6.8e-3</td>
<td>2:54:56</td>
<td>1:39:26</td>
<td>1:25:17</td>
</tr>
<tr>
<td>maxG77</td>
<td>8.0e-3</td>
<td>-</td>
<td>4:21:11</td>
<td>3:35:31</td>
</tr>
<tr>
<td>thetaG77</td>
<td>4e-3</td>
<td>-</td>
<td>14:04:37</td>
<td>11:50:07</td>
</tr>
</tbody>
</table>

**Table 1:** MPI code on preprocessed SDP - Henry2 distributed memory system

-32-
Comparison with OpenMP version of CSDP

<table>
<thead>
<tr>
<th>Prob</th>
<th>dual_gap</th>
<th>p=1</th>
<th>p=2</th>
<th>p=4</th>
<th>p=8</th>
<th>Iter</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>thetaG32</td>
<td>4e-7</td>
<td>12:01</td>
<td>6:54</td>
<td>4:07</td>
<td>2:43</td>
<td>23</td>
<td>0.63</td>
</tr>
<tr>
<td>thetaG48</td>
<td>5e-7</td>
<td>39:42</td>
<td>21:38</td>
<td>12:51</td>
<td>8:09</td>
<td>24</td>
<td>1.39</td>
</tr>
<tr>
<td>maxG57</td>
<td>8.5e-7</td>
<td>1:00:00</td>
<td>32:28</td>
<td>21:41</td>
<td>13:55</td>
<td>16</td>
<td>2.31</td>
</tr>
<tr>
<td>thetaG57</td>
<td>6.4e-7</td>
<td>2:44:10</td>
<td>1:29:17</td>
<td>52:37</td>
<td>36:01</td>
<td>23</td>
<td>3.84</td>
</tr>
<tr>
<td>maxG62</td>
<td>6.3e-7</td>
<td>2:32:10</td>
<td>1:26:08</td>
<td>52:48</td>
<td>36:01</td>
<td>16</td>
<td>4.51</td>
</tr>
<tr>
<td>maxG65</td>
<td>3e-7</td>
<td>3:58:30</td>
<td>2:08:26</td>
<td>1:18:51</td>
<td>51:11</td>
<td>18</td>
<td>5.88</td>
</tr>
<tr>
<td>maxG66</td>
<td>4e-7</td>
<td>5:27:01</td>
<td>2:59:38</td>
<td>1:51:10</td>
<td>1:12:39</td>
<td>17</td>
<td>7.43</td>
</tr>
<tr>
<td>thetaG62</td>
<td>7.6e-7</td>
<td>6:33:02</td>
<td>4:42:50</td>
<td>2:05:16</td>
<td>1:20:50</td>
<td>21</td>
<td>7.51</td>
</tr>
<tr>
<td>maxG67</td>
<td>2.5e-7</td>
<td>7:45:47</td>
<td>4:13:35</td>
<td>2:34:42</td>
<td>1:42:05</td>
<td>18</td>
<td>9.18</td>
</tr>
<tr>
<td>maxG77</td>
<td>7.3e-7</td>
<td>20:01:32</td>
<td>Failed</td>
<td>Failed</td>
<td>4:02:15</td>
<td>17</td>
<td>17.97</td>
</tr>
<tr>
<td>thetaG77</td>
<td>-</td>
<td>MM</td>
<td>MM</td>
<td>MM</td>
<td>MM</td>
<td>-</td>
<td>31.2</td>
</tr>
</tbody>
</table>

Table 2: OpenMP CSDP on original SDP - Power5 shared memory system
Scalability of algorithm on selected problems on up to 16 processors

<table>
<thead>
<tr>
<th>Prob</th>
<th>p=1</th>
<th>p=2</th>
<th>p=4</th>
<th>p=8</th>
<th>p=16</th>
</tr>
</thead>
<tbody>
<tr>
<td>thetaG32</td>
<td>19:44 (65)</td>
<td>14:26 (70)</td>
<td>8:40 (63)</td>
<td>5:35 (70)</td>
<td>3:58 (68)</td>
</tr>
<tr>
<td></td>
<td>18:28 (46)</td>
<td>13:16 (51)</td>
<td>7:32 (44)</td>
<td>4:25 (52)</td>
<td>2:36 (48)</td>
</tr>
<tr>
<td>maxG57</td>
<td>42:36 (65)</td>
<td>37:36 (65)</td>
<td>32:18 (64)</td>
<td>29:01 (71)</td>
<td>17:31 (63)</td>
</tr>
<tr>
<td></td>
<td>31:41 (48)</td>
<td>27:29 (48)</td>
<td>20:35 (47)</td>
<td>17:10 (54)</td>
<td>7:06 (46)</td>
</tr>
<tr>
<td>maxG66</td>
<td>3:31:45 (72)</td>
<td>2:43:22 (70)</td>
<td>2:08:31 (70)</td>
<td>1:31:34 (71)</td>
<td>1:15:39 (72)</td>
</tr>
<tr>
<td></td>
<td>2:24:26 (54)</td>
<td>1:53:17 (53)</td>
<td>1:26:51 (53)</td>
<td>50:56 (53)</td>
<td>30:01 (54)</td>
</tr>
<tr>
<td></td>
<td>3:33:20 (105)</td>
<td>2:26:44 (91)</td>
<td>2:02:30 (98)</td>
<td>1:28:26 (116)</td>
<td>51:58 (104)</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>8:01:01 (138)</td>
<td>6:47:16 (128)</td>
<td>5:34:20 (122)</td>
<td>3:37:25 (118)</td>
</tr>
<tr>
<td>maxG77</td>
<td>-</td>
<td>4:21:11 (72)</td>
<td>3:35:31 (71)</td>
<td>3:02:44 (69)</td>
<td>2:32:35 (71)</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>2:51:05 (53)</td>
<td>2:16:23 (52)</td>
<td>1:49:42 (50)</td>
<td>1:13:41 (52)</td>
</tr>
</tbody>
</table>
Figure 2: Speed ups attained by algorithm on selected problems
Conclusions and Future work

- The decomposition algorithm is much faster and requires considerably less memory storage than the serial version of CSDP on selected SDP problems.

- Our solution times are very competitive with the OpenMP version of CSDP on selected large SDP instances.

- Algorithm attains good parallel scalability as one increases the number of processors.

- We demonstrate that the matrix completion scheme can be modified and used in the solution of much larger SDPs than was previously possible.

- We are developing a hybrid (MPI-OpenMP) version of the decomposition algorithm, where each subproblem is solved in parallel with the OpenMP version of CSDP.

- Large scale parallel solution of structured POPs (more details in Sivaramakrishnan 2008).
Thank you for your attention!
Questions, Comments, Suggestions?

The slides from this talk are available online at

A technical report appears at
1. **K. Sivaramakrishnan**, *A parallel interior point decomposition algorithm for block angular semidefinite programs*, Computational Optimization and Applications (to appear)


8. **H. Waki, S. Kim, M. Kojima, and M. Muramatsu**, *SparsePOP: a sparse semidefinite programming relaxation of polynomial optimization problems*, Available at
   http://www.is.titech.ac.jp/~kojima/SparsePOP/


