A PARALLEL CONIC INTERIOR POINT DECOMPOSITION APPROACH FOR BLOCK-ANGULAR SEMIDEFINITE PROGRAMS

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

INFORMS Annual Meeting
Pittsburgh
November 6, 2006
Contents

- Conic programming
- Motivation for our decomposition approach
- Semidefinite programs with block angular structure
- Preprocessing SDPs into others with block angular structure
- Decomposition framework for block angular SDPs
- Algorithm
- Computational results
- Conclusions
Conic programming

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

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

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.

- \( \mathcal{K}_i = \mathbb{S}^{n_i}_+ = \{ X \in \mathbb{S}^{n_i} : X \succeq 0 \} \) 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.
Semidefinite programs with a decomposable structure

1. Preprocessed SDPs after matrix completion:

2. Control and stability analysis of interconnected subsystems:


4. Polynomial optimization problems with symmetry/sparsity:

5. Exploiting group symmetry in SDPs:
   Kanno et al. (2001), Parrilo-Gatermann (2004), De Klerk et al. (2005)
Semidefinite programming

\begin{align*}
\text{max} & \quad C \cdot X \\
\text{subject to} & \quad A(X) = b \\
& \quad X \succeq 0
\end{align*}

\begin{align*}
\text{min} & \quad b^T y \\
\text{subject to} & \quad A^T y - S = C \\
& \quad S \succeq 0
\end{align*}

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

\[
A(X) = \begin{pmatrix} A_1 \cdot X \\ \vdots \\ A_m \cdot X \end{pmatrix}, \quad A^T y = \sum_{i=1}^{m} y_i A_i
\]

  where \( A_i \in S^n, i = 1, \ldots, m \)
One sample SDP

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 \textbf{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 \textbf{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.
Semidefinite programming 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**

- \( X_i, C_i \in S^{n_i}, b \in \mathbb{R}^m \).
- \( C_i = \{ X_i : B_i(X_i) = d_i, X_i \succeq 0 \} \) are compact semidefinite feasibility sets (described by LMIs).
- The objective function and coupling constraints are **block separable**.
- If the coupling constraints were absent, then we only have to solve \( r \) independent problems of the form

\[
\max_{X_i \in C_i} C_i \cdot X_i
\]
Assumptions

(1) The SDP has a Slater point. Ensures the dual problem attains its solution and there is no duality gap!

(2) The convex sets $C_i$ are compact, i.e., $\text{trace}(X_i) = 1$ for all $X_i \in C_i$.

(3) The number of coupling constraints is small.

(4) The blocks are about the same size.
Preprocessing sparse SDPs into block-angular SDPs: 1

Consider the SDP

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

where $L$ is the adjacency matrix of the graph

**GRAPH**

**CHORDAL EXTENSION OF GRAPH**
Using matrix completion, one can reformulate the earlier SDP as

\[
\begin{align*}
\max & \sum_{k=1}^{4} (L^k \bullet X^k) \\
\text{s.t.} & \quad X^1_{23} - X^4_{13} = 0, \\
& \quad X^2_{34} - X^3_{12} = 0, \\
& \quad X^3_{23} - X^4_{23} = 0, \\
& \quad X^k_{ii} = 1, \quad i = 1, \ldots, |C_k|, \quad k = 1, \ldots, 4, \\
& \quad X^k \succeq 0, \quad k = 1, \ldots, 4,
\end{align*}
\]

which is in block-angular form.
Preprocessing sparse SDPs into block-diagonal SDPs

1. Construct the aggregate sparsity graph $G = (V, E)$ from data matrices $A_0 = C$ and $A_i, i = 1, \ldots, m$ of SDP. We have $V = \{1, \ldots, n\}$ and

$$E = \{(i, j) \in V \times V : \exists k \in \{0, 1, \ldots, m\} \text{ s.t. } (A_k)_{ij} \neq 0\}$$

2. Construct a minimal chordal extension $G' = (V, E')$ of $G = (V, E)$.
3. Find maximal cliques $Cl_i, i = 1, \ldots, k$ in $G' = (V, E')$. We have

$$X \succeq 0 \iff X_{Cl_i, Cl_i} \succeq 0, \; i = 1, \ldots, k$$

4. Each block in the block-diagonal SDP corresponds to a maximal clique.
5. Introduce additional equality constraints for common nodes and edges in the cliques; for instance if cliques $Cl_k$ and $Cl_l$ share a common edge $\{i, j\}$ then $X_{i_j}^k = X_{i_j}^l$ etc.
6. In some special cases, the resulting block-diagonal SDP has a block-angular form.
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 **convex** but **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

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

(these can be computed in parallel!)

-12-
Solving the Lagrangian dual

1. Construct a model \( \theta^k(y) \) an **underestimate** for \( \theta(y) \)

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

from the function values and subgradient information.

2. The **regularized** master problem then is

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

where \( u^k \geq 0 \) and \( x^k \) is our current center (best iterate so far!)

3. The dual to this quadratic program (much easier to solve!) is

\[
\max \ -\frac{1}{2u^k} \left\| \sum_{i=1}^{r} \sum_{j=1}^{J^k(i)} A_i(X^j_i)\lambda^j_i - b \right\|^2 + \sum_{i=1}^{r} \sum_{j=1}^{J^k(i)} ((C_i - A_i^T x^k) \cdot X^j_i)\lambda^j_i
\]

s.t. \[
\sum_{j=1}^{J^k(i)} \lambda^j_i = 1, \ i = 1, \ldots, r
\]

\[
\lambda^j_i \geq 0, \ i = 1, \ldots, r, \ j = 1, \ldots, J^k(i).
\]
Figure 1: Solving the Lagrangian dual problem
The complete algorithm

1. **Initialize:** Set $k = 1$, $J^1(i) = \phi$, $z^1_i = -\infty$, $i = 1, \ldots, r$. Choose $u^1 > 0$, $y^1 \in \mathbb{R}^m$; and set $x^1 = y^1$, and $\theta^1(y^1) = -\infty$.

2. **Solve subproblems in parallel:** For $i = 1, \ldots, r$, solve $i$th subproblem with $y = y^k$ for optimal objective value $\theta_i(y^k)$.

3. **Update model function:** If $\theta_i(y^k) > z^k_i$, update the $i$th model function $\theta_i^{k+1}(y)$ and set $J^{k+1}(i) = J^k(i) \cup \{k\}$. Else, $J^{k+1}(i) = J^k(i)$.

4. **Update the center $x^k$:** If $k = 1$ or if 

$$\theta(y^k) \leq (1 - \gamma)\theta(x^{k-1}) + \gamma \theta^k(y^k)$$

then set $x^k = y^k$ (serious step); otherwise set $x^k = x^{k-1}$ (null step).

5. **Solve master problem:** Solve dual master QP for $\lambda^j_i$ and $z^{k+1} = (z^{k+1}_1, \ldots, z^{k+1}_r)$ (dual variables). Compute $y^{k+1}$ using

$$y^{k+1} = x^k + \frac{1}{u^k}(\sum_{i=1}^r \sum_{j=1}^{J^k(i)} A_i X_i^j \lambda^j_i - b)$$

and let $\theta^{k+1}(y^{k+1}) = b^T y^{k+1} + \sum_{i=1}^r z^{k+1}_i$. If $\theta^{k+1}(y^{k+1}) = \theta(x^k)$, we are done! Else, set $k = k + 1$ and return to Step 2.
\[
\begin{align*}
\text{MAX} & \quad \sum_{i=1}^{r} \text{trace}(C_iX_i) \\
\text{subject to} & \quad \sum_{i=1}^{r} A_i(X_i) = b \\
& \quad X_i \in C_i, \quad i=1,\ldots,r
\end{align*}
\]

\[
\theta_r(y) = \text{trace}(C_r - A_r^Ty)X_r
\]

\[
\begin{align*}
\text{MAX} & \quad \sum_{i=1}^{r} \theta_i(y) \\
\text{subject to} & \quad X_1 \in C_1 \\
& \quad X_r \in C_r
\end{align*}
\]

\[
\theta(y) = b^Ty + \theta_1(y)
\]

**Figure 2: Decomposition by prices**
General case: Conic master problem 1

1. For some blocks, $C_i = \{ X_i : \text{trace}(X_i) = 1, \quad X_i \succeq 0 \}$. We will assume that the first $s$ of the $r$ blocks are of this type.

2. The function $\theta(y)$ is

$$\theta(y) = b^T y + \sum_{i=1}^{s} \lambda_{\max}(C_i - A_i^T y) + \sum_{i=s+1}^{r} \max_{X_i \in C_i} (C_i - A_i^T y) \bullet X_i$$

3. For the first $s$ blocks, the subproblem is **Lanczos** solver that computes the maximum eigenvalue of $(C_i - A_i^T y)$ and its associated eigenspace (fast and very easy to implement!).

4. Our model function $\theta^k(y)$ is then

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

where $P^j_i$ contains the eigenspace associated with $\lambda_{\max}(C_i - A_i^T y^j)$.

5. This gives rise to a master problem which is a quadratic conic problem over linear, and smaller dimensional semidefinite cones.
General case: Conic master problem 2

The master problem is

$$\max - \frac{1}{2w_k^k} \| \sum_{i=1}^{r} A_i(W_i) - b \|^2 + \sum_{i=1}^{r} (C_i - A_i^T x^k) \cdot W_i$$

s.t. \( W_i = \sum_{j=1}^{J^k(i)} P_i^j V_i^j P_i^j^T, \) \( i = 1, \ldots, s \)

\( W_i = \sum_{j=1}^{J^k(i)} \lambda_i^j X_i^j, \) \( i = s + 1, \ldots, r \)

\( \sum_{j=1}^{J^k(i)} \text{trace}(V_i^j) = 1, \) \( i = 1, \ldots, s \)

\( \sum_{j=1}^{J^k(i)} \lambda_i^j = 1, \) \( i = s + 1, \ldots, r \)

\( V_i^j \geq 0, \) \( i = 1, \ldots, s, \) \( j = 1, \ldots, J^k(i) \)

\( \lambda_i^j \geq 0, \) \( i = s + 1, \ldots, r, \) \( j = 1, \ldots, J^k(i). \)
Convergence of the algorithm

Theorem 1 Suppose, \( x^k \) is not an optimal solution of the Lagrangian dual problem. Then there exists \( m > k \) such that \( \theta(x^m) < \theta(x^k) \).

(in other words the algorithm is not stuck at a suboptimal point and the number of null steps is finite!)

Theorem 2 The decomposition scheme generates a sequence \( \{x^k\} \) that converges to an optimal solution of the Lagrangian dual problem.
Computational Results: 1

- Results on the IBM Blade Center Linux Cluster (Henry2) at NC State.
- Each of the 175 nodes is a 2.8-3.2 GHz processor with 4 GB of memory.
- Our code is in C and uses MPI for interprocessor communication.
- CPLEX 9.0 was used to solve the master problem and CSDP was used to solve the subproblems.
- 2-3 digits of accuracy or an upper limit of 2 hours in computations.
- Chose a constant weight parameter $u = 10$ in the computations.
### Computational Results: 3 digits of accuracy

<table>
<thead>
<tr>
<th>Prob</th>
<th>n</th>
<th>$n_p$</th>
<th>m</th>
<th>Our bound</th>
<th>Our h:m:s (p)</th>
<th>SDPT3 bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxG11</td>
<td>800</td>
<td>216(4)</td>
<td>1208(360)</td>
<td>629.51</td>
<td>1:23(4)</td>
<td>629.16</td>
</tr>
<tr>
<td>maxG32</td>
<td>2000</td>
<td>1022(3)</td>
<td>2820(786)</td>
<td>1568.78</td>
<td>8:31(3)</td>
<td>1567.64</td>
</tr>
<tr>
<td>qpG11</td>
<td>1600</td>
<td>432(4)</td>
<td>1256(408)</td>
<td>2450.32</td>
<td>2:48(4)</td>
<td>2448.66</td>
</tr>
</tbody>
</table>

1. The original SDPs have 800, 2000, and 800 constraints respectively.

2. $n_p = 216(4)$ indicates that the block angular SDP has 4 blocks and the size of the largest block is 216.

3. $m = 1208(360)$ indicates that the block angular SDP has 1208 constraints of which 360 are coupling constraints.

4. Our bound is the best upper bound obtained by our code, while SDPT3 bound is the optimal objective value of the semidefinite program.

5. h:m:s is the time taken by the codes in hours:minutes:seconds and $p$ is the number of processors employed in the parallel decomposition code.
## Other Computational Results: 3 digits of accuracy

<table>
<thead>
<tr>
<th>Prob</th>
<th>$n$</th>
<th>$n_p$</th>
<th>$m$</th>
<th>Our bound</th>
<th>Our h:m:s (p)</th>
<th>SDPT3 bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>mcp124-1</td>
<td>124</td>
<td>60(6)</td>
<td>172(43)</td>
<td>142.05</td>
<td>2(6)</td>
<td>141.99</td>
</tr>
<tr>
<td>mcp250-1</td>
<td>250</td>
<td>176(9)</td>
<td>381(124)</td>
<td>317.36</td>
<td>34(9)</td>
<td>317.26</td>
</tr>
<tr>
<td>mcp500-1</td>
<td>500</td>
<td>204(63)</td>
<td>1584(951)</td>
<td>598.71</td>
<td>38(10)</td>
<td>598.15</td>
</tr>
<tr>
<td>toruspm3-8-50</td>
<td>512</td>
<td>474(39)</td>
<td>1310(570)</td>
<td>528.07</td>
<td>9:43(10)</td>
<td>527.81</td>
</tr>
<tr>
<td>torusg3-8</td>
<td>512</td>
<td>402(111)</td>
<td>2822(1650)</td>
<td>457.63</td>
<td>11:3(10)</td>
<td>457.36</td>
</tr>
<tr>
<td>maxG51</td>
<td>1000</td>
<td>971(24)</td>
<td>1677(517)</td>
<td>4008.95</td>
<td>48:5(10)</td>
<td>4003.81</td>
</tr>
<tr>
<td>qpG32</td>
<td>4000</td>
<td>1054(14)</td>
<td>5566(3370)</td>
<td>6229.84</td>
<td>31:28(10)</td>
<td>6226.55</td>
</tr>
</tbody>
</table>
Figure 3: Variation of bounds for preprocessed maxG11 problem
A regularized trust region variant

1. Software is not readily available for quadratic mixed conic problems over linear and semidefinite cones.

2. Employ an infinity norm trust region in lieu of the quadratic penalty term. This gives the following linear conic master problem

\[
\begin{align*}
\min & \quad b^T y + \sum_{i=1}^{r} z_i \\
\text{s.t.} & \quad z_i I \geq (P_i^j)^T (C_i - A_i^T y) P_i^j, \quad i = 1, \ldots, s, \quad j = 1, \ldots, J^k(i) \\
& \quad z_i \geq (C_i - A_i^T y) \bullet X_i^j, \quad i = s + 1, \ldots, r, \quad j = 1, \ldots, J^k(i) \\
& \quad y \geq x^k - \Delta^k \\
& \quad y \leq x^k + \Delta^k.
\end{align*}
\]

which can be solved in CSDP.

3. The radius $\Delta^k$ of the trust region is varied in a similar fashion to the quadratic penalty term $u$. 

-24-
Ongoing and Future work

1. Adaptively changing the weights $u$ for faster convergence.

2. Developing our own code to solve the quadratic conic problem in the general case, with ARPACK solving our specialized subproblems.

3. Aggregating unimportant columns to keep the master problem small.

4. Employing warm-start in the master problem and subproblems.

5. Employing a infinity norm trust constraint $\|y - x^k\| \leq \Delta$ in the master problem (with adjustable $\Delta^k$).
Thank you for your attention!
Questions, Comments, Suggestions?

The slides from this talk are available online at
http://www4.ncsu.edu/~kksivara/publications/kartik-informs06.pdf

A technical report appears at

-26-


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

Bibliography: 2

10. **R.B. Lehoucq, D.C. Sorensen, and C. Yang**, *ARPACK user’s guide*
    http://www.caam.rice.edu/software/ARPACK/


    http://www.stanford.edu/~boyd/distr_contr_cdc04.html

15. **S. Mehrotra and M.G. ¨Ozevin**, *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


