Parallel implementation of primal-dual interior-point methods for semidefinite programs

Masakazu Kojima, Kazuhide Nakata

Katsuki Fujisawa and Makoto Yamashita

3rd Annual McMaster Optimization Conference:
Theory and Applications
(MOPTA 03)

July 30 - August 1, 2003

Hamilton, Ontario

1. **SDP (semidefinite program) and its dual**
2. **Existing numerical methods for SDPs**
3. **Outline of our parallel implementation, SDPARA and SDPARA-C**
4. **Computation of search directions in SDPARA**
5. **Numerical results on SDPARA**
6. **Positive definite matrix Completion used in SDPARA-C**
7. **Numerical results on SDPARA-C**
8. **Conclusions**

1. **SDP (semidefinite program) and its dual**
2. **Existing numerical methods for SDPs**
3. **Outline of our parallel implementation, SDPARA and SDPARA-C**
4. **Computation of search directions in SDPARA**
5. **Numerical results on SDPARA**
6. **Positive definite matrix completion used in SDPARA-C**
7. **Numerical results on SDPARA-C**
8. **Conclusions**

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

$$X, \ S \in \mathcal{S}^n, \ y_p \in R \ (1 \le p \le m) \ : \ \text{variables}$$

$$A_0, \ A_p \in \mathcal{S}^n, \ b_p \in R \ (1 \le p \le m) \ : \ \text{given data}$$

$$\mathcal{S}^n \ : \ \text{the set of } n \times n \text{ symmetric matrices}$$

$$U \bullet V = \sum_{i=1}^n \sum_{j=1}^n U_{ij} V_{ij} \ \text{ for every } U, \ V \in R^{n \times n}$$

$$X \succeq O \Leftrightarrow X \in \mathcal{S}^n \ \text{ is positive semidefinite}$$

$$X \succ O \Leftrightarrow X \in \mathcal{S}^n \ \text{ is positive definite}$$

$$\begin{array}{lll} \mathcal{P}: \min & A_0 \bullet X & \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O \\ \mathcal{D}: \max & \sum_{p=1}^m b_p y_p & \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O \end{array}$$

Important features of SDPs

- $n \times n$ matrix variables $X, \ S \in \mathcal{S}^n$, each of which involves $n(n+1)/2$ real variables; for example, $n = 2000 \Rightarrow n(n+1)/2 \approx 2$ million.
- m linear equality constraints in $\mathcal{P}$, where m can be large up to $n(n+1)/n$.
- SDPs can be large scale easily $\Rightarrow$ Solving large scale SDPs (accurately) is a challenging subject.

$$
\begin{array}{llll}
\mathcal{P}: \min & A_0 \bullet X & \text{sub.to} & A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O \\
\mathcal{D}: \max & \sum_{p=1}^m b_p y_p & \text{sub.to} & \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O
\end{array}
$$

Important features of SDPs

- $n \times n$ matrix variables $X, \ S \in \mathcal{S}^n$, each of which involves $n(n+1)/2$ real variables; for example, $n = 2000 \Rightarrow n(n+1)/2 \approx 2$ million.
- m linear equality constraints in $\mathcal{P}$, where m can be large up to $n(n+1)/n$.
- SDPs can be large scale easily
  $\Rightarrow$ Solving large scale SDPs (accurately) is a challenging subject.

$$\Downarrow$$

$\Diamond$ Special techniques for exploiting structure and sparsity
$\Diamond$ Enormous computational power — parallel computation

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m),\ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0,\ \mathcal{S}^n \ni S \succeq O$$

1. SDP (semidefinite program) and its dual
2. **Existing numerical methods for SDPs**
3. Outline of our parallel implementation, SDPARA and SDPARA-C
4. Computation of search directions in SDPARA
5. Numerical results on SDPARA
6. Positive definite matrix completion used in SDPARA-C
7. Numerical results on SDPARA-C
8. Conclusions

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Some existing numerical methods for SDPs

- **IPMs (Interior-point methods)**
  - **Primal-dual scaling, CSDP(Borchers), SDPA(Fujisawa-K-Nakata), SDPT3(Todd-Toh-Tutuncu), SeDuMi(F.Sturm)**
  - **Dual scaling, DSDP(Benson-Ye)**
- **Nonlinear programming approaches**
  - **Spectral bundle method(Helmberg-Kiwiel)**
  - **Gradient-based log-barrier method(Burer-Monteiro-Zhang)**
  - **PENON(M. Kocvara) — Generalized augmented Lagrangian method**

These methods are competing to each other.

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Some existing numerical methods for SDPs

- IPMs (Interior-point methods)
  - Primal-dual scaling, CSDP(Borchers), SDPA(Fujisawa-K-Nakata), SDPT3(Todd-Toh-Tutuncu), SeDuMi(F.Sturm)
  - Dual scaling, DSDP(Benson-Ye)
- Nonlinear programming approaches
  - Spectral bundle method(Helmberg-Kiwiel)
  - Gradient-based log-barrier method(Burer-Monteiro-Zhang)
  - PENON(M. Kocvara) — Generalized augmented Lagrangian method

These methods are competing to each other.

- Large scale SDPs (e.g., n=10,000) and low accuracy ⇒ Spectral bundle, Gradient-based log-barrier or IPMs using CG
- Medium scale SDPs (e.g. n, $m = 1000$) and high accuracy ⇒ IPMs

Parallel implementation of SDPA, DSDP, Spectral bundle method

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \leq p \leq m),\ \mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0,\ \mathcal{S}^n \ni S \succeq O$

Advantages of primal-dual IPMs (interior-point methods)

- **Based on deep and strong theory (Nesterov-Nemirovskii)**
- **Highly accurate solutions.** *cf* **S.bundle and Gradient-based methods**
- **The number of iterations is small;**
  usually 20 — 100 iterations in practice, independent of sizes of SDPs.

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Advantages of primal-dual IPMs (interior-point methods)

- **Based on deep and strong theory (Nesterov-Nemirovskii)**
- **Highly accurate solutions.** *cf* **S.bundle and Gradient-based methods**
- **The number of iterations is small;**
  usually 20 — 100 iterations in practice, independent of sizes of SDPs.

Disadvantage of primal-dual IPMs (interior-point methods)

- **Heavy computation in each iteration**

$\Downarrow$

Parallel execution of heavy computation in each iteration

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. **Outline of our parallel implementation, SDPARA and SDPARA-C**
4. Computation of search directions in SDPARA
5. Numerical results on SDPARA
6. Positive definite matrix completion used in SDPARA-C
7. Numerical results on SDPARA-C
8. Conclusions

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Generic primal-dual IPM on a single CPU ⇒ SDPA

Step 0: Choose $(X, y, S) = (X^0, y^0, S^0)$; $X^0 \succ O$ and $S^0 \succ O$.

Step 1: Compute a search direction (dX, dy, dS).

Step 2: Choose α_p and α_d; $X + \alpha_p dX \succ O$ and $S + \alpha_d dS \succ O$. Let $X = X + \alpha_p dX$, $(y, S) = (y, S) + \alpha_d(dy, dS)$.

Step 3: Go to Step 1.

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$

Generic primal-dual IPM on a single CPU ⇒ SDPA

Step 0: Choose $(X, y, S) = (X^0, y^0, S^0)$; $X^0 \succ O$ and $S^0 \succ O$.

Step 1: Compute a search direction (dX, dy, dS). ⇒ matrix B

Step 2: Choose α_p and α_d; $X + \alpha_p dX \succ O$ and $S + \alpha_d dS \succ O$. Let $X = X + \alpha_p dX, \ (y, S) = (y, S) + \alpha_d (dy, dS)$.

Step 3: Go to Step 1.

SDPA $\overset{\text{parallel}}{\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow}$ SDPARA for large $m \le 30,000$ but small $n \le 1,000$

- Computation of Schur complement matrix B
- Cholesky factorization of B

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$

Generic primal-dual IPM on a single CPU $\Rightarrow$ SDPA

Step 0: Choose $(X, y, S) = (X^0, y^0, S^0)$; $X^0 \succ O$ and $S^0 \succ O$.

Step 1: Compute a search direction (dX, dy, dS).

Step 2: Choose α_p and α_d; $X + \alpha_p dX \succ O$ and $S + \alpha_d dS \succ O$. Let $X = X + \alpha_p dX, \ (y, S) = (y, S) + \alpha_d (dy, dS)$.

Step 3: Go to Step 1.

SDPA $\xrightarrow{\text{parallel}}$ SDPARA for large $m \leq 30,000$ but small $n \leq 1,000$

$\Downarrow$

• Positive definite matrix completion technique

$\Downarrow$

SDPARA-C for larger n

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Generic primal-dual IPM on a single CPU ⇒ SDPA

Step 0: Choose $(X, y, S) = (X^0, y^0, S^0)$; $X^0 \succ O$ and $S^0 \succ O$.

Step 1: Compute a search direction (dX, dy, dS).

Step 2: Choose α_p and α_d; $X + \alpha_p dX \succ O$ and $S + \alpha_d dS \succ O$. Let $X = X + \alpha_p dX$, $(y, S) = (y, S) + \alpha_d(dy, dS)$.

Step 3: Go to Step 1.

Outline of our parallel implementation

- MPI (Message Passing Interface) for communication between CPUs.
- Myrinet-2000 between nodes, higher transmission than Gigabit Ethernet.
- ScaLAPACK (Scalable Linear Algebra PACKage) for parallel Cholesky factorization.

$$\mathcal{P}: \min \; A_0 \bullet X \quad \text{sub.to} \; A_p \bullet X = b_p \; (1 \le p \le m), \; \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \; \sum_{p=1}^m b_p y_p \; \text{sub.to} \; \sum_{p=1}^m A_p y_p + S = A_0, \; \mathcal{S}^n \ni S \succeq O$$

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. Outline of our parallel implementation, SDPARA and SDPARA-C
4. **Computation of search directions in SDPARA**
5. Numerical results on SDPARA
6. Positive definite matrix completion used in SDPARA-C
7. Numerical results on SDPARA-C
8. Conclusions

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Computing HRVW/KSH/M search direction in SDPARA

(X, y, S) ; $X \succ O$ and $S \succ O$ — the current point

(dX, dy, dS) — HRVW/KSH/M search direction

$$Bdy = r, \ \text{where } B \in \mathcal{S}^m_{++}, \ r \in \mathbb{R}^m \Leftarrow \ (X, y, S), \ A_p, \ b_p$$

$$dS = D - \sum_{j=1}^m dy_j, \ \widehat{dX} = (K - X\,dS)S^{-1}, \ dX = \left(\widehat{dX} + \widehat{dX}^T\right)/2$$

where $D \in \mathcal{S}^n, \ K \in \mathbb{R}^{n \times n} \ \Leftarrow \ (X, y, S), \ A_p, \ b_p$

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \le p \le m)$, $\mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0$, $\mathcal{S}^n \ni S \succeq O$

Computing HRVW/KSH/M search direction in SDPARA

(X, y, S) ; $X \succ O$ and $S \succ O$ — the current point

(dX, dy, dS) — HRVW/KSH/M search direction

$Bdy = r$, where $B \in \mathcal{S}^m_{++}$, $r \in \mathbb{R}^m \Leftarrow (X, y, S)$, A_p, b_p

$dS = D - \sum_{j=1}^m dy_j$, $\widehat{dX} = (K - X dS)S^{-1}$, $dX = \left(\widehat{dX} + \widehat{dX}^T\right)/2$

where $D \in \mathcal{S}^n$, $K \in \mathbb{R}^{n \times n} \Leftarrow (X, y, S)$, A_p, b_p

The computation of B requires $O(m^2 n^2)$ arithmetic operations, and the solution of $Bdy = r$ $O(m^3)$ arithmetic operations $\Rightarrow$ SDPARA.

The computation of $\widehat{dX} = (K - X dS)S^{-1}$ and $dX = \left(\widehat{dX} + \widehat{dX}^T\right)/2$ is expensive when n is large $\Rightarrow$ SDPARA-C.

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

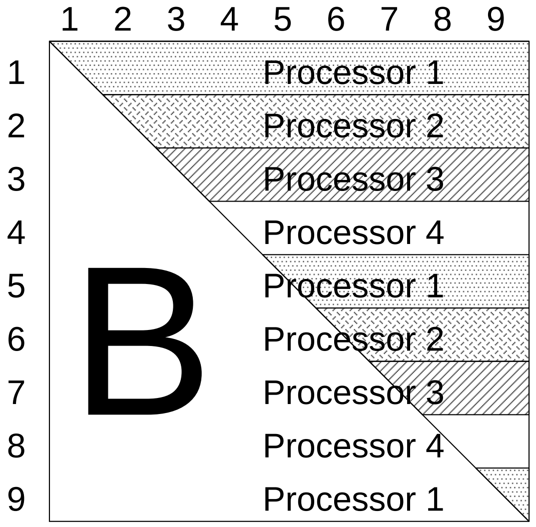
$$Bdy = r, \ \text{where } B \in \mathcal{S}^m_{++}, \ B_{pq} = \text{Trace } S^{-1} A_p X A_q \ (p, q = 1, \ldots, m)$$

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \leq p \leq m)$, $\mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0$, $\mathcal{S}^n \ni S \succeq O$

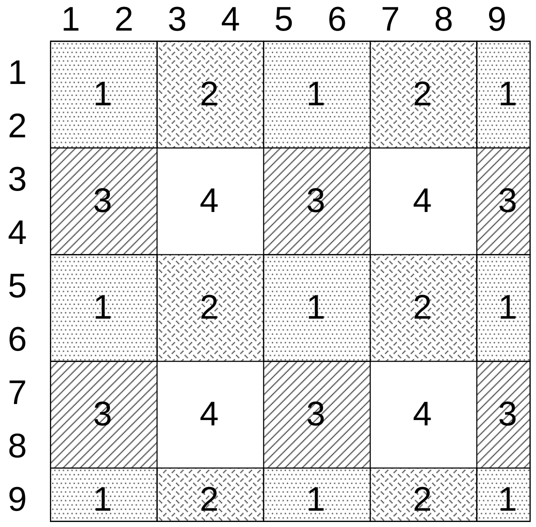
$Bdy = r$, where $B \in \mathcal{S}^m_{++}$, $B_{pq} = \text{Trace } S^{-1} A_p X A_q \ (p, q = 1, \ldots, m)$

In SDPARA:

- Compute each row of B by one cpu; $S^{-1}A_pX$ is common in pth row.
- Redistribute the elements of B in 2-dim. block-cyclic distribution.
- Apply the parallel Cholesky factorization to B for solving $Bdy = r$.



Computation of 9×9 B by 4cpu's



2-dimensional block-cyclic dist.

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. Outline of our parallel implementation, SDPARA and SDPARA-C
6. Computation of search directions in SDPARA
5. Numerical results on SDPARA
6. Positive definite matrix completion used in SDPARA-C
7. Numerical results on SDPARA-C
8. Conclusions

$$\mathcal{P}: \min \ A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \ \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Numerical results on SDPARA

- **PC cluster; 1.6GHz cpu and 768 MB memory in each node.**
- **Myrinet-2000 communication between the nodes.**

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \ \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Numerical results on SDPARA

- PC cluster; 1.6GHz cpu and 768 MB memory in each node.
- Myrinet-2000 communication between the nodes.
- All data A_p, b_p are distributed to every node.
- Iterates $\{(X^k, y^k, S^k)\}$ are stored and updated in each nodes.
- Some heavy computations are done in parallel and their results are distributed to all node, but all other computations are done individually and independently in each node.
- Primal and dual feasibilities, relative duality gaps $\leq 1.0e^{-6}$.

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Numerical results on SDPARA

Test problem	m >>	size of A_p (diag. blocks) – small
control11 from sdplib	1596	(100, 55)
theta6 from sdplib	4375	300
thetaG51 from sdplib	6910	1001
sdp from quantum chemistry 1	15,313	(120,120,256)
sdp from quantam chemistry 2	24,503	(153,153,324)

Test probem	m	size of A_p – large
maxG51 from sdplib	1000	1000
qpG11 from sdplib	800	1600
qpG51 from sdplib	1000	2000
torusg3-15 from dimacs	3,375	3,375
norm min.	10	40000

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \ \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Numerical results on SDPARA

			comp.	time	in	sec
Test probem	m	size of A_p	1cpu	4cpu	16cpu	64cpu
control11	1596	(110,55)	685	195	67	32
theta6	4375	300	600	168	68	38
thetaG51	6970	1001	M	1345	627	447
sdp from q.c 1	15,313	(120,120,256)	M	M	2077	733
sdp from q.c 2	24,503	(153,153,324)	M	M	6370	1985

M: lack of memory

Test probem	m	size of A_p	1cpu	4cpu	16cpu	64cpu
maxG51	1000	1000	176	177	184	190
qpG11	800	1600	638	651	650	652
qpG51	1000	2000	M	M	M	M
torusg3-15	3,375	3,375	M	M	M	M
norm min.	10	40000	M	M	M	M

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Numerical results on SDPARA

			comp.	time	in	sec
Test probem	m	size of A_p	1cpu	4cpu	16cpu	64cpu
control11	1596	(110,55)	685	195	67	32
theta6	4375	300	600	168	68	38
thetaG51	6970	1001	M	1345	627	447
sdp from q.c 1	15,313	(120,120,256)	M	M	2077	733
sdp from q.c 2	24,503	(153,153,324)	M	M	6370	1985

M: lack of memory

Test probem	m	size of A_p	1cpu	4cpu	16cpu	64cpu
maxG51	1000	1000	176	177	184	190
qpG11	800	1600	638	651	650	652
qpG51	1000	2000	M	M	M	M
torusg3-15	3,375	3,375	M	M	M	M
norm min.	10	40000	M	M	M	M

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^{m} b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^{m} A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. Outline of our parallel implementation, SDPARA and SDPARA-C
4. Computation of search directions in SDPARA
5. Numerical results on SDPARA
6. **Positive definite matrix completion used in SDPARA-C**
7. Numerical results on SDPARA-C
8. Conclusions

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Weak point in primal-dual interior-point methods

- **The primal X is dense in general even when all A_p's are sparse.**
- **The dual $S = A_0 - \sum_{p=1}^m A_p y_p$ inherits the sparsity of A_p's.**

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \leq p \leq m)$, $\mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0$, $\mathcal{S}^n \ni S \succeq O$

Weak point in primal-dual interior-point methods

- **The primal X is dense in general even when all A_p's are sparse.**
- **The dual $S = A_0 - \sum_{p=1}^m A_p y_p$ inherits the sparsity of A_p's.**

This difference causes a critical disadvantage of primal-dual interior-point methods compared to dual interior-point methods for large scale SDPs

To overcome this disadvantage, we exploit

$$E \equiv \{(i,j) : [A_p]_{ij} \neq 0 \text{ for } \exists p\}$$

("the aggregate sparsity pattern" over all A_p's)

based on some fundamental results about positive matrix completion. Fukuda-K-Murota-Nakata '00.

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \ \text{sub.to} \ \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Example: $m = 2$, $n = 4$.

$$\min \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 3 & 3 \\ 1 & 2 & 3 & 9 \end{pmatrix} \bullet \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{pmatrix}$$

Remember! $C \bullet X = \sum_{i,j} C_{ij} X_{ij}$

$$\text{sub.to} \begin{pmatrix} 9 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 7 & 3 \\ 1 & 2 & 3 & 5 \end{pmatrix} \bullet X = 6, \ \begin{pmatrix} 2 & 0 & 0 & 6 \\ 0 & 8 & 0 & 8 \\ 0 & 0 & 0 & 0 \\ 6 & 8 & 0 & 4 \end{pmatrix} \bullet X = 5, \ X \succeq O$$

- "the aggregate sparsity pattern" over all A_p's $E = \{(i, j) \ \text{ in Red } \}$
- X_{ij} $(i, j) \notin E$ are unnecessary to evaluate the objective function and the equality constraints, but necessary for $X \succeq O$.

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \ \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Example: $m = 2$, $n = 4$.

$$\min \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 3 & 3 \\ 1 & 2 & 3 & 9 \end{pmatrix} \bullet \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{pmatrix}$$

Remember!
$$C \bullet X = \sum_{i,j} C_{ij} X_{ij}$$

$$\text{sub.to} \begin{pmatrix} 9 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 7 & 3 \\ 1 & 2 & 3 & 5 \end{pmatrix} \bullet X = 6, \quad \begin{pmatrix} 2 & 0 & 0 & 6 \\ 0 & 8 & 0 & 8 \\ 0 & 0 & 0 & 0 \\ 6 & 8 & 0 & 4 \end{pmatrix} \bullet X = 5, \quad X \succeq O$$

- "the aggregate sparsity pattern" over all A_p's $E = \{(i, j)$ in Red $\}$
- X_{ij} $(i, j) \notin E$ are unnecessary to evaluate the objective function and the equality constraints, but necessary for $X \succeq O$.
- Using positive definite matrix completion, we can generate each iterate (X, y, S) such that both X^{-1} and S have the same sparsity pattern as E when E is "nicely sparse" as in above example.
- In general, we need to expand E to a "nicely sparse" E'.

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \le p \le m)$, $\mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0$, $\mathcal{S}^n \ni S \succeq O$

Technical details of the pd matrix completion

$$E \equiv \{(i,j) : [A_p]_{ij} \neq 0 \text{ for } \exists p\} \text{(the aggregate sparsity pattern)}$$

Each iteration of IPM, we need to

(I) compute a step length α_p such that $X + \alpha_p dX \succ O$,

(II) compute a search direction (dX, dy, dS).

By applying the positive definite matrix completion technique, we can execute (I) and (II) using only X_{ij}, $(i,j) \in E'$ for some "nice and small" $E' \supset E$ (a chordal extension of E).

Positive definite matrix completion problem:

Given $X_{ij} = \bar{X}_{ij}$, $(i,j) \in E'$, assign values to the other elements X_{ij}, $(i,j) \notin E'$ so that $X \succ O$.

(I) Computation of a step length α_p: Example ($m = 2$, $n = 4$)

$$\min \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 3 & 3 \\ 1 & 2 & 3 & 9 \end{pmatrix} \bullet \begin{pmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{pmatrix}$$

$$\text{sub.to} \begin{pmatrix} 9 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 7 & 3 \\ 1 & 2 & 3 & 5 \end{pmatrix} \bullet X = 6, \begin{pmatrix} 2 & 0 & 0 & 6 \\ 0 & 8 & 0 & 8 \\ 0 & 0 & 0 & 0 \\ 6 & 8 & 0 & 4 \end{pmatrix} \bullet X = 5, \ X \succeq O$$

Given X_{ij}, $(i, j) \in E$, $\exists$ X_{ij}, $(i, j) \notin E$ such that $X \succ O$ iff $\begin{pmatrix} X_{ii} & X_{i4} \\ X_{4i} & X_{44} \end{pmatrix} \succ O$ $(i = 1, 2, 3)$ ($\forall$ maximal principal submatrix in $X'_{ij}s$ is pd.)

Given X_{ij}, dX_{ij} $(i, j) \in E$, a step length α_p is determined such that

$$\begin{pmatrix} X_{ii} & X_{i4} \\ X_{4i} & X_{44} \end{pmatrix} + \alpha_p \begin{pmatrix} dX_{ii} & dX_{i4} \\ dX_{4i} & dX_{44} \end{pmatrix} \succ O \ (i = 1, 2, 3)$$

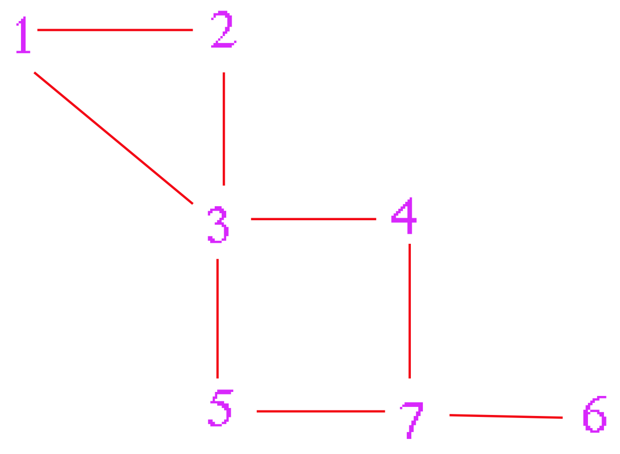
- X_{ij}, dX_{ij} $(i, j) \notin E$ are unnecessary!
- We can generalize this method to any aggregate sparsity pattern E characterized as a chordal graph.

$\mathbf{7 \times 7}$ **example with the aggregate sparsity pattern** $\boldsymbol{E} = \{(\boldsymbol{i}, \boldsymbol{j}) : \boldsymbol{A}_{ij} \neq \mathbf{0}\}$,

$$\boldsymbol{A} = \begin{array}{c|ccccccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{7} \\ \hline \mathbf{1} & * & * & * & 0 & 0 & 0 & 0 \\ \mathbf{2} & * & * & * & 0 & 0 & 0 & 0 \\ \mathbf{3} & * & * & * & * & * & 0 & 0 \\ \mathbf{4} & 0 & 0 & * & * & 0 & 0 & * \\ \mathbf{5} & 0 & 0 & * & 0 & * & 0 & * \\ \mathbf{6} & 0 & 0 & 0 & 0 & 0 & * & * \\ \mathbf{7} & 0 & 0 & 0 & * & * & * & * \end{array}$$

where $*$ **: a nonzero element.**

$\boldsymbol{E}$ **is represented by the graph:**

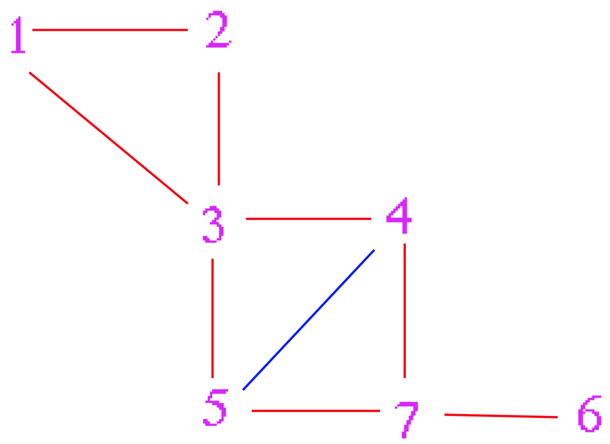


7×7 example with the aggregate sparsity pattern $E = \{(i,j) : A_{ij} \neq 0\}$,

$$A = \begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & * & * & * & 0 & 0 & 0 & 0 \\ 2 & * & * & * & 0 & 0 & 0 & 0 \\ 3 & * & * & * & * & * & 0 & 0 \\ 4 & 0 & 0 & * & * & \mathbf{0} & 0 & * \\ 5 & 0 & 0 & * & \mathbf{0} & * & 0 & * \\ 6 & 0 & 0 & 0 & 0 & 0 & * & * \\ 7 & 0 & 0 & 0 & * & * & * & * \end{array}$$

where $*$: a nonzero element.

E is represented by the graph:



Let E' be a chordal extension of E, *i.e.*, $E' \supset E$ such that $\forall$ minimal cycle has no more than 3 edges

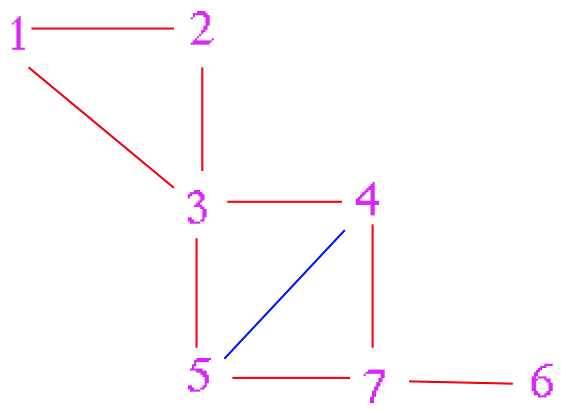
or the aggregate sparsity pattern E' induced by the symbolic Cholesky factorization of A.

$E' = E \cup \{(4,5)\}$: a chordal extension of E

E':

$A =$

	1	2	3	4	5	6	7
1	*	*	*	0	0	0	0
2	*	*	*	0	0	0	0
3	*	*	*	*	*	0	0
4	0	0	*	*	0	0	*
5	0	0	*	0	*	0	*
6	0	0	0	0	0	*	*
7	0	0	0	*	*	*	*



Theorem (Gron-Sá et.al. '84) Given $X_{ij} = \bar{X}_{ij}$, $(i,j) \in E'$, $\exists$ X_{ij}, $(i,j) \notin E'$; $X \succ O$ iff $X_{CC} \succ O$ for $\forall$ maximal clique C of E'

In the example above, the maximal cliques are

$$C_1 = \{1,2,3\}, C_2 = \{3,4,5\}, C_3 = \{4,5,7\}, C_4 = \{6,7\}$$

Hence α_p is chosen; $X_{CC} + \alpha_p dX_{CC} \succ O$ for $\forall$ maximal clique C of E', instead of $X + \alpha_p dX \succ O$.

- **This requires less cpu time and memory.**
- **We need only $X_{ij} = \overline{X}_{ij}$, $dX_{ij} = \overline{dX}_{ij}$, $(i,j) \in E'$.**

(II) Computation of a search direction (dX, dy, dS).

Let $\hat{X} \in \mathcal{S}^n_{++}$ be the matrix that maximizes its determinant among all pd completions of a given partial matrix with $X_{ij} = \bar{X}_{ij},\ (i,j) \in E'$. Then $\hat{X}^{-1}$ and the Cholesky factorization LL^T of $\hat{X}^{-1}$, which we can easily compute, enjoys the same sparsity pattern as E'.

(II) Computation of a search direction (dX, dy, dS).

Let $\hat{X} \in \mathcal{S}^n_{++}$ be the matrix that maximizes its determinant among all pd completions of a given partial matrix with $X_{ij} = \bar{X}_{ij}$, $(i,j) \in E'$. Then $\hat{X}^{-1}$ and the Cholesky factorization LL^T of $\hat{X}^{-1}$, which we can easily compute, enjoys the same sparsity pattern as E'.

Store the Cholesky factorization LL^T of $\hat{X}^{-1}$ instead of $\hat{X}$ itself, and the Cholesky factorization NN^T of S. Then, all A_p's, N and L enjoy the same sparsity pattern as E'. Use $L^{-T}L^{-1}$ and NN^T instead of $\hat{X}$ and S, respectively.

In particular,

$$B_{pq} \equiv \text{Trace } A_p \hat{X} A_q S^{-1} = \text{Trace } A_p L^{-T} L^{-1} A_q N^{-T} N^{-1}.$$

for computing the $m \times m$ coefficient matrix B of the Schur complement equation $B dy = r$ for (dX, dy, dS).

(II) Computation of a search direction (dX, dy, dS).

Let $\hat{X} \in \mathcal{S}^n_{++}$ be the matrix that maximizes its determinant among all pd completions of a given partial matrix with $X_{ij} = \bar{X}_{ij}, \ (i,j) \in E'$. Then $\hat{X}^{-1}$ and the Cholesky factorization LL^T of $\hat{X}^{-1}$, which we can easily compute, enjoys the same sparsity pattern as E'.

> **Store the Cholesky factorization LL^T of $\hat{X}^{-1}$ instead of $\hat{X}$ itself, and the Cholesky factorization NN^T of S. Then, all A_p's, N and L enjoy the same sparsity pattern as E'. Use $L^{-T}L^{-1}$ and NN^T instead of $\hat{X}$ and S, respectively.**

In particular,

$$B_{pq} \equiv \text{Trace } A_p \hat{X} A_q S^{-1} = \text{Trace } A_p L^{-T} L^{-1} A_q N^{-T} N^{-1}.$$

for computing the $m \times m$ coefficient matrix B of the Schur complement equation $Bdy = r$ for (dX, dy, dS).

- **Considerable savings in memory.**
- **Suitable for parallel computation to reduce computational time.**

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \ \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. Outline of our parallel implementation, SDPARA and SDPARA-C
4. Computation of search directions in SDPARA
5. Numerical results on SDPARA
6. Positive definite matrix completion used in SDPARA-C
7. **Numerical results on SDPARA-C**
8. Conclusions

$\mathcal{P}$: min $A_0 \bullet X$ sub.to $A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$
$\mathcal{D}$: max $\sum_{p=1}^m b_p y_p$ sub.to $\sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$

7. Numerical results on SDPARA-C

Test problem	m	size of A_p	1cpu	4cpu	16cpu	64cpu
theta6	4375	300	2650	695	221	100
thetaG51	6970	1001	M	7910	2218	685

M: lack of memory

Test problem	m	size of A_p	1cpu	4cpu	16cpu	64cpu
maxG51	1000	1000	545	195	75	62
qpG11	800	1600	90	29	11	8
qpG51	1000	2000	2034	575	196	108
torusg3-15	3,375	3,375	10378	3099	989	530
norm min.	10	40000				7706

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \le p \le m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \quad \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Comparison of Real Computational Time (sec) between SDPARAC, SDPARA and PDSDP (Benson).

control10, sdplib, m=1326, n=(100,50)

#CPU	SDPARA-C	SDPARA	PDSDP
1	27437	429	2101
4	7488	128	727
16	2308	43	311
64	1036	22	207

theta6, sdplib, m=4375, n=300

#CPU	SDPARA-C	SDPARA	PDSDP
1	2650	694	1007
4	695	147	557
16	221	65	360
64	100	37	452

$$\mathcal{P}: \min \quad A_0 \bullet X \quad \text{sub.to} \quad A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \quad \sum_{p=1}^m b_p y_p \ \text{sub.to} \quad \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

maxG51, sdplib, max cut m=1000, n=1000

#CPU	SDPARA-C	SDPARA	PDSDP
1	545	175	82
4	195	176	79
16	75	174	84
64	62	176	109

qpG51, sdplib, m=1000, n=2000

#CPU	SDPARA-C	SDPARA	PDSDP
1	2034	M(970MB)	436
4	575	M	424
16	196	M	432
64	108	M	495

torusg3-15 dimacs max cut, m=3,375, n=3,375

#CPU	SDPARA-C	SDPARA	PDSDP
1	10387	M(920MB)	1875
4	3099	M	1752
16	989	M	1748
64	530	M	1871

$$\mathcal{P}: \ \min \ \ A_0 \bullet X \quad \text{sub.to} \ \ A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \ \max \ \ \sum_{p=1}^m b_p y_p \ \text{sub.to} \ \ \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. Outline of our parallel implementation, SDPARA and SDPARA-C
4. Computation of search directions in SDPARA
5. Numerical results on SDPARA
6. Positive definite matrix completion used in SDPARA-C
7. Numerical results on SDPARA-C
8. **Conclusions**

$$\mathcal{P}: \min \ A_0 \bullet X \ \text{sub.to} \ A_p \bullet X = b_p \ (1 \leq p \leq m), \ \mathcal{S}^n \ni X \succeq O$$
$$\mathcal{D}: \max \ \sum_{p=1}^m b_p y_p \ \text{sub.to} \ \sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$$

Two types of parallel primal-dual interior-point methods for SDPs

(a) Parallel implementation SDPARA of SDPA,
suitable for large m and smaller n.
The largest problem solved is

	m	size of A_p	1cpu	4cpu	16cpu	64cpu
sdp from q.c 2	24,503	(153,153,324)	M	M	6370	1985

(b) SDPARA-C = SDPARA + the pd matrix completion,
suitable for larger n.
The largest problems solved are

	m	size of A_p	1cpu	4cpu	16cpu	64cpu
torusg3-15	3,375	3,375	10378	3099	989	530
norm min.	10	40000				7706