Parallel Computing for Semidefinite Programming

Masakazu Kojima, Tokyo Institute of Technology

2005 SIAM Conference on Optimization

May 16, 2005

Stockholm, Sweden

This presentation is based on a joint work "SDPA project" with

Katsuki	Fujisawa,	Tokyo Denki University
Mituhiro	Fukuda,	Tokyo Institute of Technology
Yoshiaki	Futakata,	University of Virginia
Masakazu	Kojima,	Tokyo Institute of Technology
Kazuhide	Nakata,	Tokyo Institute of Technology
Makoto	Yamashita,	Kanagawa University

1. SDP (semidefinite program) and its dual.

2. Existing numerical methods for SDPs.

3. A primal-dual IPM and outline of its parallel implementation.

4. Numerical results on SDPARA.

5. Positive definite matrix completion used in SDPARA-C.

6. Numerical results on SDPARA-C.

7. Comparison between SDPARA-C and SDPARA.

8. Concluding remarks

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

$$\begin{aligned}
\mathcal{S}^n &: \text{ the set of } n \times n \text{ symmetric matrices} \\
X, \ S \in \mathcal{S}^n, \ y_p \in R \ (1 \leq p \leq m) &: \text{ variables} \\
A_0, \ A_p \in \mathcal{S}^n, \ b_p \in R \ (1 \leq p \leq m) &: \text{ given data}
\end{aligned}$$

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

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

Important features — SDP can be large-scale easily

- $n \times n$ 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 $\mathcal{P}$ or m A_p's $\in \mathcal{S}^n$.

$\Downarrow$

◊ Exploit structured sparsity.
◊ Enormous computational power $\Rightarrow$ parallel computation.

$\mathcal{P}$: 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 $\sum_{p=1}^m A_p y_p + S = A_0$, $\mathcal{S}^n \ni S \succeq O$

Structured sparsity

The aggregate sparsity pattern $\widehat{A}$: a symbolic $n \times n$ matrix:

$$\widehat{A}_{ij} = \begin{cases} \star & \text{if the } (i,j)\text{th element of } A_p \text{ is nonzero for } \exists p = 0,\ldots,m, \\ 0 & \text{otherwise,} \end{cases}$$

where $\star$ denotes a nonzero number.

Example: $m = 1$

$$A_0 = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \ A_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 3 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 \end{pmatrix} \Rightarrow \widehat{A} = \begin{pmatrix} \star & \star & 0 & \star \\ \star & \star & \star & 0 \\ 0 & \star & \star & 0 \\ \star & 0 & 0 & \star \end{pmatrix}.$$

Next — three types of structured sparsity

$\mathcal{P}$: 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 $\sum_{p=1}^m A_p y_p + S = A_0$, $\mathcal{S}^n \ni S \succeq O$

Structured sparsity

The aggregate sparsity pattern $\widehat{A}$: a symbolic $n \times n$ matrix:

$$\widehat{A}_{ij} = \begin{cases} \star & \text{if the } (i,j)\text{th element of } A_p \text{ is nonzero for } \exists p = 0, \ldots, m, \\ 0 & \text{otherwise,} \end{cases}$$

where $\star$ denotes a nonzero number.

Structured sparsity-1 : $\widehat{A}$ is block-diagonal.

Then X, S has the same diagonal block structure as $\widehat{A}$.

$$\widehat{A} = \begin{pmatrix} B_1 & O & O \\ O & B_2 & O \\ O & O & B_3 \end{pmatrix}, \ B_i : \text{symmetric.}$$

Example: CH_3N : an SDP from quantum chemistry, Fukuda et al. 2005.

$m = 20,709$, $n = 12,802$, "the number of blocks in $\widehat{A}$" $= 22$,

the largest bl.size $= 3,211 \times 3,211$, the average bl.size $= 583 \times 583$.

$\mathcal{P}$: 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 $\sum_{p=1}^m A_p y_p + S = A_0$, $\mathcal{S}^n \ni S \succeq O$

Structured sparsity

The aggregate sparsity pattern $\widehat{A}$: a symbolic $n \times n$ matrix:

$$\widehat{A}_{ij} = \begin{cases} \star \text{ if the } (i,j)\text{th element of } A_p \text{ is nonzero for } \exists p = 0, \ldots, m, \\ 0 \text{ otherwise,} \end{cases}$$

where $\star$ denotes a nonzero number.

Structured sparsity-2 : $\widehat{A}$ has a sparse Cholesky factorization.

"a small bandwidth" "a small bandwidth + bordered"

$$\widehat{A} = \begin{pmatrix} \star & \star & O & O & O \\ \star & \star & \star & O & O \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ O & O & \star & \star & \star \\ O & O & \cdots & \star & \star \end{pmatrix}, \quad \widehat{A} = \begin{pmatrix} \star & \star & O & O & \star \\ \star & \star & \star & O & \star \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ O & O & \star & \star & \star \\ \star & \star & \cdots & \star & \star \end{pmatrix}, \quad \star : \text{bl.matrix} \neq O$$

- S has the same sparsity pattern as $\widehat{A}$ but X is fully dense in general!

$\Rightarrow$ the positive definite matrix completion in SDPARA-C.

$\mathcal{P}$: 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 $\sum_{p=1}^m A_p y_p + S = A_0, \ \mathcal{S}^n \ni S \succeq O$

Structured sparsity

The aggregate sparsity pattern $\widehat{A}$: a symbolic $n \times n$ matrix:

$$\widehat{A}_{ij} = \begin{cases} \star & \text{if the } (i,j)\text{th element of } A_p \text{ is nonzero for } \exists p = 0, \ldots, m, \\ 0 & \text{otherwise,} \end{cases}$$

where $\star$ denotes a nonzero number.

Structured sparsity-3 : block-diagonal $\widehat{A}$ + blockwise orthogonality,

for most pairs (p, q) $1 \leq p < q \leq m$,

A_p and A_q do not share nonzero blocks; hence $A_p \bullet A_q = 0$.

$\Rightarrow$ the Schur complement matrix used in PDIPM becomes sparse.

$$A_1 = \begin{pmatrix} A_{11} & O & O \\ O & O & O \\ O & O & O \end{pmatrix}, \ A_2 = \begin{pmatrix} O & O & O \\ O & A_{22} & O \\ O & O & O \end{pmatrix}, \ A_3 = \begin{pmatrix} O & O & O \\ O & O & O \\ O & O & A_{33} \end{pmatrix}.$$

- An engineering application, Ben-Tal and Nemirovskii 1999.
- A sparse SDP relaxation of poly. opt. problem, Waki et al. 2005.
- Not incorporated in our software yet.

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

1. SDP (semidefinite program) and its dual.
2. **Existing numerical methods for SDPs.**
3. A primal-dual IPM and outline of its parallel implementation.
4. Numerical results on **SDPARA**.
5. Positive definite matrix completion used in **SDPARA-C**.
6. Numerical results on **SDPARA-C**.
7. Comparison between **SDPARA-C** and **SDPARA**.
8. Concluding Remarks.

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

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
  - Saddle point mirror-prox algorithm (Lu-Nemirovski-Monteiro)

- Medium scale SDPs (e.g. $n, \ m = 1000$) and high accuracy.
- Large scale SDPs (e.g., n=10,000) and low accuracy.

- Parallel implementation of SDPA, DSDP, Spectral bundle method
- SDPARA, SDPARA-C to solve large-scale SDPs with high accuracy.

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

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

- **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

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

1. SDP (semidefinite program) and its dual.
2. Existing numerical methods for SDPs.
3. **A primal-dual IPM and outline of its parallel implementation.**
4. Numerical results on **SDPARA**.
5. Positive definite matrix completion used in **SDPARA-C**.
6. Numerical results on **SDPARA-C**.
7. Comparison between **SDPARA-C** and **SDPARA**.
8. Concluding Remarks.

$\mathcal{P}$: 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 ⇒ SDPA

Step 0: Choose $(X, y, S) = (X^0, y^0, S^0)$; $X^0 \succ O$ and $S^0 \succ O$. $k = 1$.
Step 1: Compute a search direction (dX, dy, dS). ⇒ $B dy = r$
Step 2: Choose α_p and α_d;
$X^{k+1} = X^k + \alpha_p dX \succ O$, $S^{k+1} = S^k + \alpha_d dS \succ O$, $y^{k+1} = y^k + \alpha_d dy$.
Step 3: Let $k = k + 1$. Go to Step 1.

B : $m \times m$ fully dense except special cases, computed from A_p, X, S.

Major time consumption (second) on a single cpu implemention.

parallel comp.	part	control11	theta6	maxG51
SDPARA ⇐	Elements of B	463.2	78.3	1.5
SDPARA ⇐	Cholesky fact. of B	31.7	209.8	3.0
SDPARA-C ⇐	dX	1.8	1.8	47.3
SDPARA-C ⇐	Other dense mat. comp.	1.0	4.1	86.5
	Others	7.2	5.13	1.8
	Total	505.2	292.3	140.2

$\mathcal{P}$: 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 $S^0 \succ O$. $k = 1$.
Step 1: Compute a search direction (dX, dy, dS). $\Rightarrow$ $Bdy = r$
Step 2: Choose α_p and α_d;
$X^{k+1} = X^k + \alpha_p dX \succ O$, $S^{k+1} = S^k + \alpha_d dS \succ O$, $y^{k+1} = y^k + \alpha_d dy$.
Step 3: Let $k = k + 1$. Go to Step 1.

SDPA $\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow$ (parallel) SDPARA for large $m \leq 30,000$ but small $n \leq 2,000$

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

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

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 $S^0 \succ O$. $k = 1$.

Step 1: Compute a search direction (dX, dy, dS). $\Rightarrow$ $Bdy = r$

Step 2: Choose α_p and α_d;
$X^{k+1} = X^k + \alpha_p dX \succ O$, $S^{k+1} = S^k + \alpha_d dS \succ O$, $y^{k+1} = y^k + \alpha_d dy$.

Step 3: Let $k = k + 1$. Go to Step 1.

SDPA $\Longrightarrow\Longrightarrow\Longrightarrow$ (parallel) $\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow$ **SDPARA for large $m \leq 30,000$ but small $n \leq 2,000$**

$\Downarrow$

$\Downarrow$ • **positive definite matrix completion**

$\Downarrow$

$\Downarrow$

SDPARA-C for larger n

SDPARA + p.d. matrix completion = SDPARA-C.

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

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. A primal-dual IPM and outline of its parallel implementation.
4. **Numerical results on SDPARA**
5. Positive definite matrix completion used in SDPARA-C.
6. Numerical results on SDPARA-C.
7. Comparison between SDPARA-C and SDPARA.
8. Concluding Remarks.

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

Numerical results on SDPARA

Hardware:

- PC cluster: 2×Opteron 2GHz cpus with 6GB memory in each node.
- MPI (Message Passing Interface) for communication between CPUs.
- Myrinet-2000 between nodes, 2 times faster than gigabit ethernet.

Software:

- ScaLAPACK for parallel Cholesky factorization.
- All data A_p, b_p are distributed to every node.
- Iterates $\{(X^k, y^k, S^k)\}$ are stored and updated in each node.
- Some heavy computations (B and its Cholesky factization to solve $Bdy = r$) are done in parallel and their results are distributed to all nodes, but all other computations are done individually and independently in each node.
- Primal and dual feasibilities, relative duality gaps $\le 1.0e^{-6}$.

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

SDPs from quantum chemistry, Fukuda et al. 2005.

atoms/molecules	m	n	#blocks	the sizes of largest blocks
O	7230	5990	22	[1450, 1450, 450, . . .]
HF	15018	10146	22	[2520, 2520, 792, . . .]
CH_3N	20709	12802	22	[3211, 3211, 1014, . . .]

number of processors		16	64	128	256
O	elements of B	10100.3	2720.4	1205.9	694.2
	Chol.fact. of B	218.2	87.3	68.2	106.2
	total	14250.6	4453.3	3281.1	2951.6
HF	elements of B	*	*	13076.1	6833.0
	Chol.fact. of B	*	*	520.2	671.0
	total	*	*	26797.1	20780.7
CH_3N	elements of B	*	*	34188.9	18003.3
	Chol.fact. of B	*	*	1008.9	1309.9
	total	*	*	57034.8	45488.9

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

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. A primal-dual IPM and outline of its parallel implementation.
4. Numerical results on SDPARA.
5. **Positive definite matrix completion used in SDPARA-C.**
6. Numerical results on SDPARA-C.
7. Comparison between SDPARA-C and SDPARA.
8. Concluding Remarks.

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

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

The aggregate sparsity pattern matrix $\widehat{A}$: a symbolic $n \times n$ matrix:

$$\widehat{A}_{ij} = \begin{cases} \star & \text{if the } (i,j)\text{th element of } A_p \text{ is nonzero for } \exists p = 0, \ldots, m, \\ 0 & \text{otherwise.} \end{cases}$$

— the structured sparsity $\Longleftrightarrow$ a sparse Cholesky factorization

- $S = A_0 - \sum_{p=1}^m A_p y_p$ has the sparsity as $\widehat{A}$, but X does not.
- Using the positive matrix completion, we can make X^{-1} sparse!

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

The aggregate sparsity pattern matrix $\widehat{A}$: a symbolic $n \times n$ matrix:

$$\widehat{A}_{ij} = \begin{cases} \star & \text{if the } (i,j)\text{th element of } A_p \text{ is nonzero for } \exists p = 0, \ldots, m, \\ 0 & \text{otherwise.} \end{cases}$$

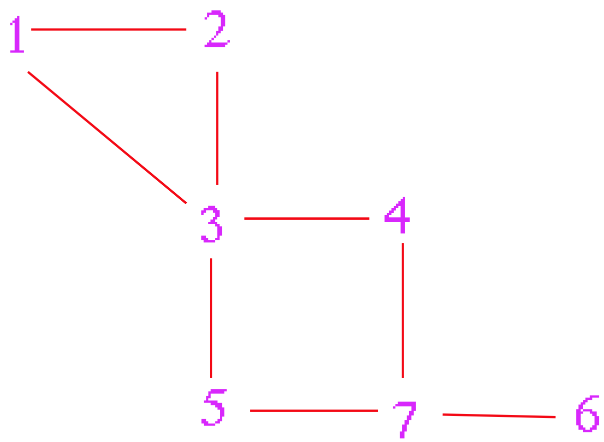
— **the structured sparsity $\Longleftrightarrow$ a sparse Cholesky factorization**

The aggregate sparsity pattern graph $G(N, E)$, where

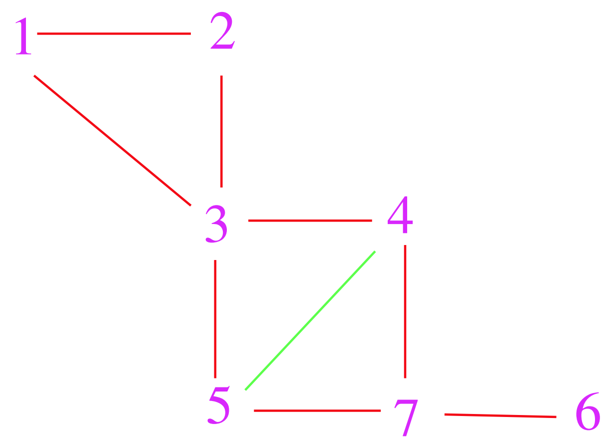
$$N = \{1, 2, \ldots, n\} \quad \text{and } E \equiv \{(i,j) : \widehat{A}_{ij} = \star\}$$

— **the structured sparsity $\Longleftrightarrow$ $\exists$ a sparse chordal extension**
($\forall$ minimal cycle has at most 3 edges)

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



$G(N, E)$: not chordal



$G(N, E')$: chordal

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 $E = \{(i, j)$ in Red $\}$
- X_{ij} $(i, j) \notin E$ are unnecessary to evaluate the objective function and the equality constraints, but necessary for $X \succeq O$.

pd matrix completion: Suppose $X_{ij} \in \mathbb{R}$ $((i, j) \in S)$ are given.

(a) $\exists X_{ij} \in \mathbb{R}$ $((i, j) \notin E)$; a completed matrix X is positive definite iff

$$\begin{pmatrix} X_{11} & X_{14} \\ X_{41} & X_{44} \end{pmatrix}, \begin{pmatrix} X_{11} & X_{24} \\ X_{42} & X_{44} \end{pmatrix}, \begin{pmatrix} X_{11} & X_{34} \\ X_{43} & X_{44} \end{pmatrix}$$ are positive definite.

(b) We can compute such a completed matrix X with the property that X^{-1} has the same sparsity pattern as E.

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 pd 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 above; $G(N, E)$ forms a chordal graph.
- In general, we need to extend $G(N, E)$ to a chordal graph.

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

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. A primal-dual IPM and outline of its parallel implementation.
4. Numerical results on **SDPARA**.
5. Positive definite matrix completion used in **SDPARA-C**.
6. **Numerical results on SDPARA-C.**
7. Comparison between **SDPARA-C** and **SDPARA**.
8. Concluding Remarks.

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

Numerical results: large-size SDPs by SDPARA-C (64 CPUs)

Hardware:

- **PC cluster: Athlon 1900+ (1.6 GHz) cpu with 768 MB memory in each node**
- **MPI (Message Passing Interface) for communication between CPUs.**
- **Myrinet-2000 between nodes, 2 times faster than gigabit ethernet.**

Software:

- **SDPARA + the positive definite matrix completion.**
- **Primal and dual feasibilities, relative duality gaps $\leq 1.0e^{-7}$.**

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

Large-size SDPs by SDPARA-C (64 CPUs)

3 types of test Problems:

(a) SDP relaxations of randomly generated max. cut problems on lattice graphs with size 10×1000, 10×2000 and 10×4000.

(b) SDP relaxations of randomly generated max. clique problems on lattice graphs with size 10×500, 10×1000 and 10×2000.

(c) Randomly generated norm minimization problems

$$\min. \left\| F_0 - \sum_{i=1}^{10} F_i y_i \right\| \ \text{sub.to} \ y_i \in \mathbb{R} \ (i = 1, 2, \ldots, 10)$$

where $F_i : 10 \times 9990, \ 10 \times 19990$ or 10×39990 and $\|G\|$ = the square root of the max. eigenvalue of $G^T G$.

In all cases, the aggregate sparsity pattern consists of one block and is very sparse.

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

Large-size SDPs by SDPARA-C (64 CPUs)

	Problem	n	m	time (s)	memory (MB)
	Cut(10×1000)	10000	10000	274.3	126
(a)	Cut(10×2000)	20000	20000	1328.2	276
	Cut(10×4000)	40000	40000	7462.0	720
	Clique(10×500)	5000	9491	639.5	119
(b)	Clique(10×1000)	10000	18991	3033.2	259
	Clique(10×2000)	20000	37991	15329.0	669
	Norm(10×9990)	10000	11	409.5	164
(c)	Norm(10×19990)	20000	11	1800.9	304
	Norm(10×39990)	40000	11	7706.0	583

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

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. A primal-dual IPM and outline of its parallel implementation.
4. Numerical results on SDPARA.
5. Positive definite matrix completion used in SDPARA-C.
6. Numerical results on SDPARA-C.
7. **Comparison between SDPARA-C and SDPARA.**
8. Concluding Remarks.

Two types of sparse problems

- $m \leq n$, $\widehat{A}$: very sparse $\Longrightarrow$ SDPARA-C
- $m > n \Longrightarrow$ SDPARA

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

Comparison between SDPARAC and SDPARA where $m \leq n$.

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

#cpu	SDPARA-C	SDPARA
1	545	175
4	195	176
16	75	174
64	62	176

qpG51, sdplib
m=1000, n=2000

#cpu	SDPARA-C	SDPARA
1	2034	M(970MB)
4	575	M
16	196	M
64	108	M

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

SDPARA-C	SDPARA
10387	M(920MB)
3099	M
989	M
530	M

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

Comparison between SDPARAC and SDPARA where $m > n$.

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

#cpu	SDPARA-C	SDPARA
1	27437	429
4	7488	128
16	2308	43
64	1036	22

theta6, m=4375, n=300

#cpu	SDPARA-C	SDPARA
1	2650	694
4	695	147
16	221	65
64	100	37

In pd matrix completion: Heavy overheads to compute

$$B_{ij} = A_i X A_j \bullet S^{-1} \ (1 \le i, j \le m).$$

- X and S^{-1} are not stored because they are dense.
- Sparse Cholesky factorizations of X^{-1} and S are stored and used to compute B_{ij}.

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

1. SDP (semidefinite program) and its dual
2. Existing numerical methods for SDPs
3. A primal-dual IPM and outline of its parallel implementation.
4. Numerical results on SDPARA.
5. Positive definite matrix completion used in SDPARA-C.
6. Numerical results on SDPARA-C.
7. Comparison between SDPARA-C and SDPARA.
8. Concluding Remarks.

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

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

- **Parallel implementation SDPARA of SDPA, suitable for large m and smaller n.**

	m	n (22 blocks)	16cpu	64cpu	128cpu	256cpu
O	7230	5990 (1450,...)	14251	4453	3281	2952
HF	15018	10146 (2520, ...)	*	*	26798	20781
CH_3N	20709	12802 (3211, ...)	*	*	57035	45489

- **SDPARA-C = SDPARA + pd matrix completion, suitable for larger n and $\widehat{A}$: very sparse.**

	m	n	1cpu	4cpu	16cpu	64cpu
cut(10×4000)	40,000	40,000				7,462
norm(10×39,990)	11	40,000				7,706
torusg3-15	3,375	3,375	10378	3099	989	530

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

(b) Future research

- **Parallel sparse Cholesky fact. of the Schur complement mat. B. This is necessary for sparse SDP relaxations of polynomial optimization problems.**
- **Distribution of the data matrices $A_0, \ldots, A_m$ among cpus to solve huge-scale SDPs.**