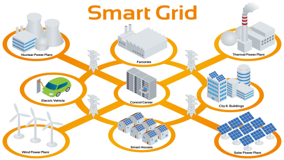
A clique graph based merging strategy for decomposable SDPs

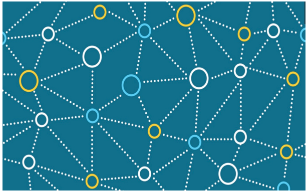
Michael Garstka[1] · Mark Cannon[1] · Paul Goulart[1]

[1]University of Oxford, UK

21st IFAC World Congress, Berlin (virtual)
13th - 17th July 2020

Why are decomposable SDPs useful?



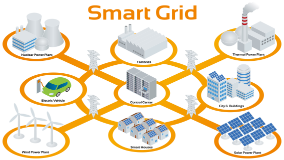


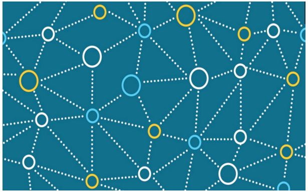
$$
\begin{array}{ll}
\text{minimize} & \langle C, X \rangle \\
\text{subject to} & \langle A_i, X \rangle = b_i,\ i = 1, \dots, m \\
 & X \in \mathbb{S}^n_+
\end{array}
$$

COSMO.jl

Why are decomposable SDPs useful?

$$
\begin{array}{ll}
\text{minimize} & \langle C, X\rangle \\
\text{subject to} & \langle A_i, X\rangle = b_i,\ i = 1,\dots,m \\
 & \boxed{X \in \mathbb{S}^n_+}
\end{array}
$$





COSMO.jl

Overview

Matrix Sparsity and Graphs

Chordal decomposition

Clique merging

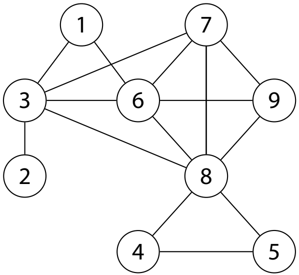
- Clique tree-based merging strategies
- Clique graph-based merging strategy

Benchmarks

Conclusion

Matrix sparsity and graphs

$$
\begin{array}{c}
\\
\begin{array}{c}1\\2\\3\\4\\5\\6\\7\\8\\9\end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc}1&2&3&4&5&6&7&8&9\end{array}\\
\begin{bmatrix}
\bullet & & & & & & & & \\
 & \bullet & & & & & & & \\
\bullet & \bullet & \bullet & & & & & & \\
 & & & \bullet & & & & & \\
 & & & \bullet & \bullet & & & & \\
\bullet & & \bullet & & & \bullet & & & \\
 & & \bullet & & & \bullet & \bullet & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & & & & \bullet & \bullet & \bullet & \bullet
\end{bmatrix}
\end{array}
\xrightarrow[\substack{\text{vertex set } V \\ \text{edge set } E \subseteq V \times V}]{G(V,E)}
$$



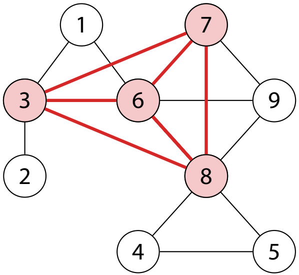
*Survey paper: [VA15]

Matrix sparsity and graphs

$$
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\end{array}
$$

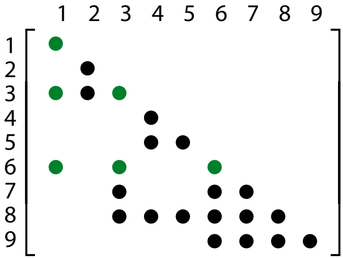
$$
\begin{array}{c|ccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
\hline
1 & \bullet & & & & & & & & \\
2 & & \bullet & & & & & & & \\
3 & \bullet & \bullet & \bullet & & & & & & \\
4 & & & & \bullet & & & & & \\
5 & & & & \bullet & \bullet & & & & \\
6 & \bullet & & \bullet & & & \bullet & & & \\
7 & & & \bullet & & & \bullet & \bullet & & \\
8 & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\
9 & & & & & & \bullet & \bullet & \bullet & \bullet
\end{array}
$$

Complete subgraph

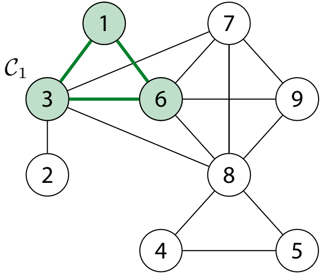


*Survey paper: [VA15]

Matrix sparsity and graphs

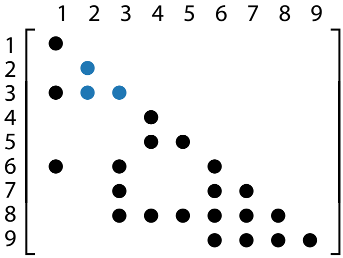


Clique 1

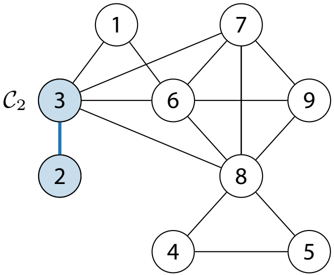


*Survey paper: [VA15]

Matrix sparsity and graphs

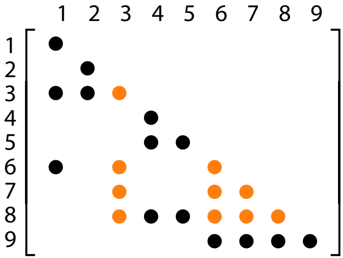


Clique 2

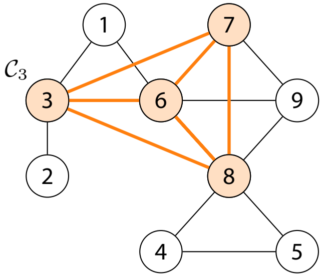


*Survey paper: [VA15]

Matrix sparsity and graphs



Clique 3

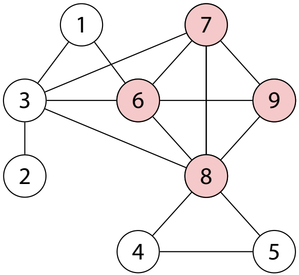


*Survey paper: [VA15]

Matrix sparsity and graphs

$$
\begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8\\9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1&2&3&4&5&6&7&8&9 \end{array}\\
\left[\begin{array}{ccccccccc}
\bullet & & & & & & & & \\
 & \bullet & & & & & & & \\
\bullet & \bullet & \bullet & & & & & & \\
 & & & \bullet & & & & & \\
 & & & \bullet & \bullet & & & & \\
\bullet & & \bullet & & & \bullet & & & \\
 & & \bullet & & & \bullet & \bullet & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & & & & \bullet & \bullet & \bullet & \bullet
\end{array}\right]
\end{array}
$$

Chordal graph

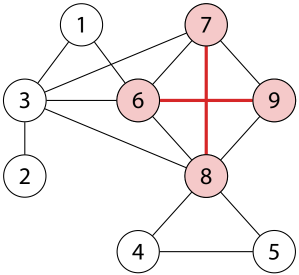


*Survey paper: [VA15]

Matrix sparsity and graphs

$$
\begin{array}{c}
\\
\begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8\\9 \end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}\\
\left[\begin{array}{ccccccccc}
\bullet & & & & & & & & \\
 & \bullet & & & & & & & \\
\bullet & \bullet & \bullet & & & & & & \\
 & & & \bullet & & & & & \\
 & & & \bullet & \bullet & & & & \\
\bullet & & \bullet & & & \bullet & & & \\
 & & \bullet & & & \bullet & \bullet & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & & & & \bullet & \bullet & \bullet & \bullet
\end{array}\right]
\end{array}
$$

Chordal graph



*Survey paper: [VA15]

Chordal decomposition

Dual SDP*

$$
\begin{array}{ll}
\text{maximize} & b^\top y \\
\text{subject to} & \sum_{i=1}^{m} A_i y_i + S = C \\
& \boxed{S \in \mathbb{S}_+^n}
\end{array}
$$

Sparse matrix cones

$$
\mathbb{S}^n(E,0) := \{S \in \mathbb{S}^n \mid S_{ij} = S_{ji} = 0, \text{ if } i \neq j,\ (i,j) \notin E\}
$$

$$
\mathbb{S}_+^n(E,0) := \{S \in \mathbb{S}^n(E,0) \mid S \succeq 0\}
$$

*equivalent decomposition result for primal form SDPs exists

Chordal decomposition

Dual SDP*

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & \sum_{i=1}^m A_i y_i + S = C \\ & \boxed{S \in \mathbb{S}_+^n(E,0)} \end{array}$$

Sparse matrix cones

$$\mathbb{S}^n(E,0) := \{S \in \mathbb{S}^n \mid S_{ij} = S_{ji} = 0, \text{ if } i \neq j,\ (i,j) \notin E\}$$
$$\mathbb{S}_+^n(E,0) := \{S \in \mathbb{S}^n(E,0) \mid S \succeq 0\}$$

*equivalent decomposition result for primal form SDPs exists

Agler's theorem

Agler's theorem*

Let $G(V,E)$ be a chordal graph with a set of maximal cliques $\{\mathcal{C}_1,\dots,\mathcal{C}_p\}$. Then $S \in \mathbb{S}^n_+(E,0)$ if and only if there exist matrices $S_\ell \in \mathbb{S}^{|\mathcal{C}_\ell|}_+$ for $\ell = 1,\dots,p$ such that

$$S = \sum_{\ell=1}^{p} T_\ell^\top S_\ell T_\ell.$$

*[AHMR88], Grone's theorem for primal form SDPs [GJSW84]

Example: Chordal Decomposition

maximize $\quad b^\top y$

subject to $\quad \sum_{i=1}^{m} A_i y_i + S = C$

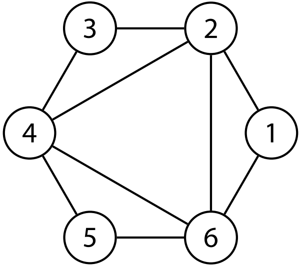
$\quad S \in \mathbb{S}_+^n(E, 0)$

$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$

Example: Chordal Decomposition

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & \sum_{i=1}^m A_i y_i + S = C \\ & S \in \mathbb{S}_+^n(E, 0) \end{array}$$

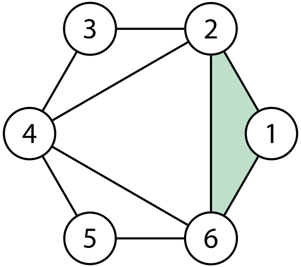
$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$



Example: Chordal Decomposition

$$
\begin{array}{ll}
\text{maximize} & b^\top y \\
\text{subject to} & \sum_{i=1}^m A_i y_i + S = C \\
& S \in \mathbb{S}^n_+(E, 0)
\end{array}
$$

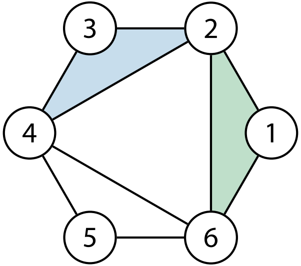
$$
\begin{bmatrix}
S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\
S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\
0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\
0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\
0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\
S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66}
\end{bmatrix}
$$



Example: Chordal Decomposition

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & \sum_{i=1}^m A_i y_i + S = C \\ & S \in \mathbb{S}_+^n(E, 0) \end{array}$$

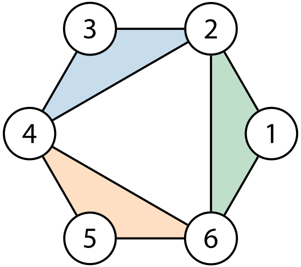
$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$



Example: Chordal Decomposition

$$
\begin{array}{ll}
\text{maximize} & b^\top y \\
\text{subject to} & \sum_{i=1}^m A_i y_i + S = C \\
& S \in \mathbb{S}_+^n(E, 0)
\end{array}
$$

$$
\begin{bmatrix}
S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\
S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\
0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\
0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\
0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\
S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66}
\end{bmatrix}
$$

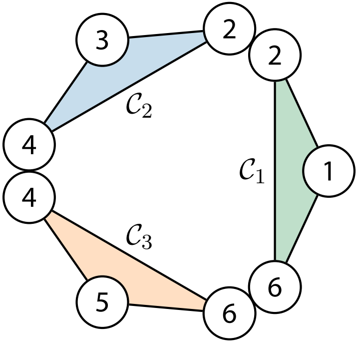


Example: Chordal Decomposition

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & \sum_{i=1}^m A_i y_i + S = C \\ & S \in \mathbb{S}_+^n(E, 0) \end{array}$$

$$\begin{array}{ll} \text{maximize} & b^\top y \\ \text{subject to} & \sum_{i=1}^m A_i y_i + \sum_{\ell=1}^3 T_\ell^\top S_\ell T_\ell = C \\ & S_1 \in \mathbb{S}_+^{|\mathcal{C}_1|},\ S_2 \in \mathbb{S}_+^{|\mathcal{C}_2|},\ S_3 \in \mathbb{S}_+^{|\mathcal{C}_3|} \end{array}$$

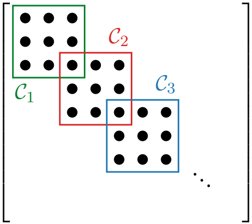
$$\begin{bmatrix} S_{11} & S_{12} & 0 & 0 & 0 & S_{16} \\ S_{21} & S_{22} & S_{23} & S_{24} & 0 & S_{26} \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & S_{42} & S_{43} & S_{44} & S_{45} & S_{46} \\ 0 & 0 & 0 & S_{54} & S_{55} & S_{56} \\ S_{61} & S_{62} & 0 & S_{64} & S_{65} & S_{66} \end{bmatrix}$$

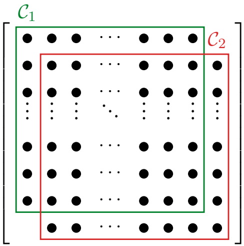


Clique merging

- Combine cliques by introducing new edges in the graph
- One merge operation:
  - replaces two PSD constraints by one larger PSD constraint
  - removes equality constraints

  $\rightarrow$ trade-off depends on the employed solver algorithm
- Obvious cases:



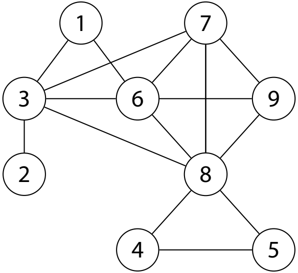


Algorithm: First-order solver

Factor constraint matrix;
while not converged:
    [...]
    Eigenvalue decomposition
    of PSD decision variables;

Clique tree-based merging strategies

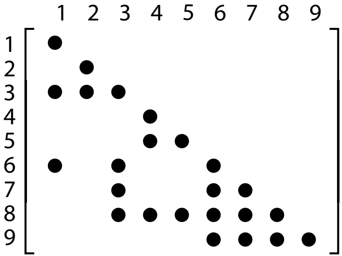
$$
\begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & \bullet & & & & & & & & \\
2 & & \bullet & & & & & & & \\
3 & \bullet & \bullet & \bullet & & & & & & \\
4 & & & & \bullet & & & & & \\
5 & & & & \bullet & \bullet & & & & \\
6 & \bullet & & \bullet & & & \bullet & & & \\
7 & & & \bullet & & & \bullet & \bullet & & \\
8 & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\
9 & & & & & & \bullet & \bullet & \bullet & \bullet
\end{array}
$$

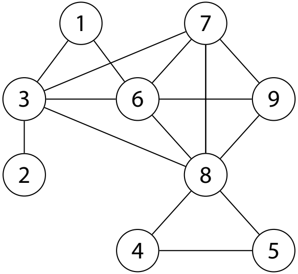


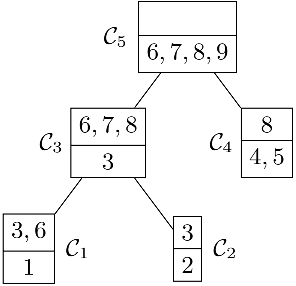
Algorithm: Clique tree-based merging

*Available packages: `SparseCoLO` [FKK+09], `Chompack` [AV15]

Clique tree-based merging strategies





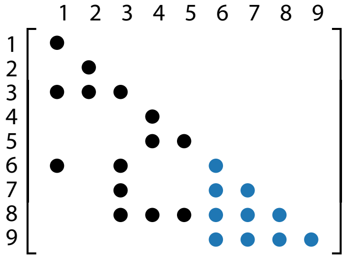


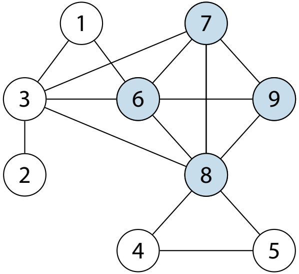
Algorithm: Clique tree-based merging

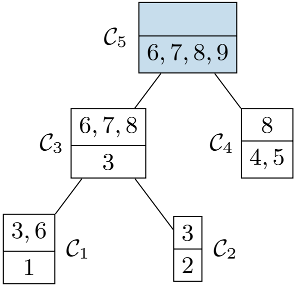
Compute clique tree;

*Available packages: `SparseCoLO` [FKK$^+$09], `Chompack` [AV15]

Clique tree-based merging strategies





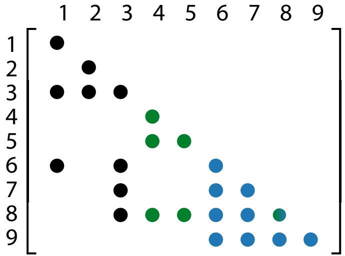


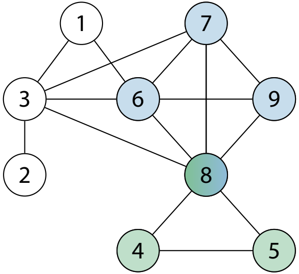
Algorithm: Clique tree-based merging

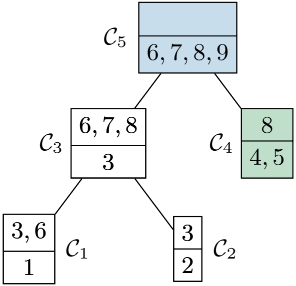
Compute clique tree;
Traverse tree depth-first: $\mathcal{C}_i$:

* Available packages: `SparseCoLO` [FKK+09], `Chompack` [AV15]

Clique tree-based merging strategies





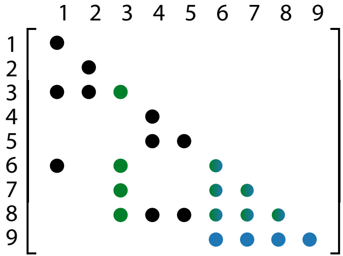


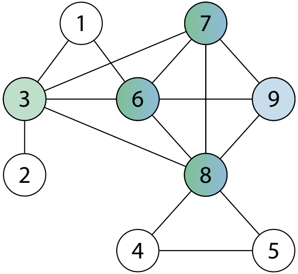
Algorithm: Clique tree-based merging

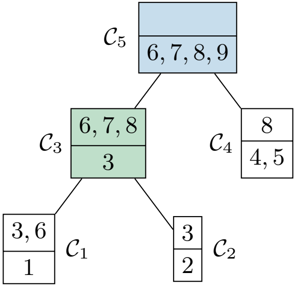
Compute clique tree;
Traverse tree depth-first: $\mathcal{C}_i$:
  Find child node: $\mathcal{C}_j$;
  if heuristic condition $f(\mathcal{C}_i, \mathcal{C}_j) \geq \gamma$ holds:

* Available packages: `SparseCoLO` [FKK+09], `Chompack` [AV15]

Clique tree-based merging strategies





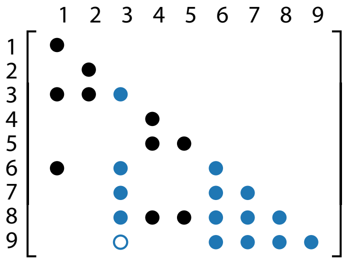


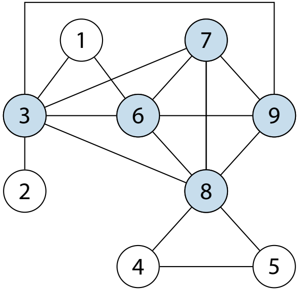
Algorithm: Clique tree-based merging

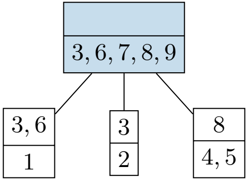
Compute clique tree;
Traverse tree depth-first: $\mathcal{C}_i$:
  Find child node: $\mathcal{C}_j$;
  if heuristic condition $f(\mathcal{C}_i, \mathcal{C}_j) \geq \gamma$ holds:

* Available packages: `SparseCoLO` [FKK+09], `Chompack` [AV15]

Clique tree-based merging strategies





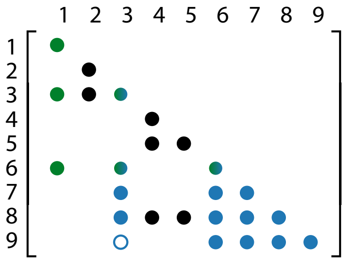


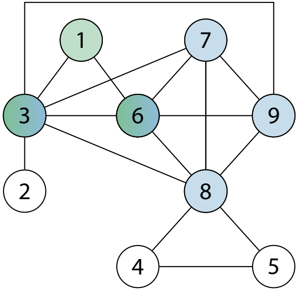
Algorithm: Clique tree-based merging

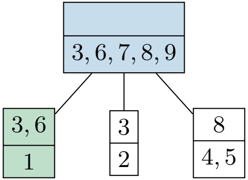
Compute clique tree;
Traverse tree depth-first: $\mathcal{C}_i$:
  Find child node: $\mathcal{C}_j$;
  if heuristic condition $f(\mathcal{C}_i, \mathcal{C}_j) \geq \gamma$ holds:
    $\mathcal{C}_m \leftarrow \mathcal{C}_i \cup \mathcal{C}_j$

* Available packages: `SparseCoLO` [FKK+09], `Chompack` [AV15]

Clique tree-based merging strategies





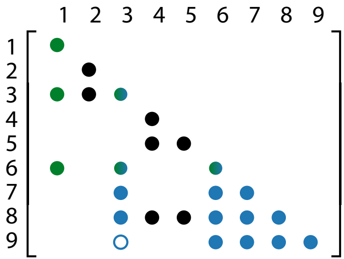


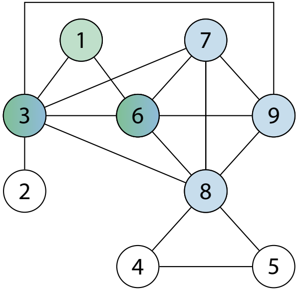
Algorithm: Clique tree-based merging

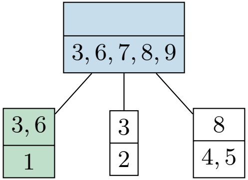
Compute clique tree;
Traverse tree depth-first: $\mathcal{C}_i$:
  Find child node: $\mathcal{C}_j$;
  if heuristic condition $f(\mathcal{C}_i, \mathcal{C}_j) \geq \gamma$ holds:
    $\mathcal{C}_m \leftarrow \mathcal{C}_i \cup \mathcal{C}_j$

* Available packages: `SparseCoLO` [FKK+09], `Chompack` [AV15]

Clique tree-based merging strategies







- Designed for interior-point solvers

+ Clique tree cheap to compute and evaluate

– Disregards distant merge candidates

– Relies on heuristic parameters

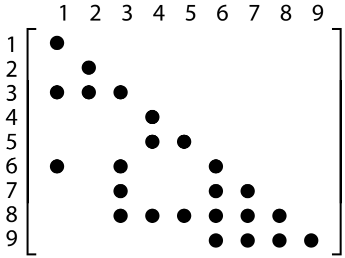
*Available packages: `SparseCoLO` [FKK+09], `Chompack` [AV15]

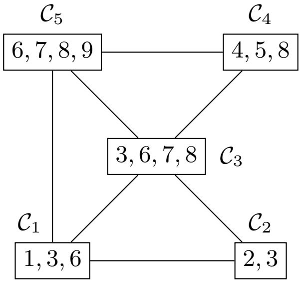
A new clique graph-based merging strategy

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} &
\begin{bmatrix}
\bullet & & & & & & & & \\
 & \bullet & & & & & & & \\
\bullet & \bullet & \bullet & & & & & & \\
 & & & \bullet & & & & & \\
 & & & \bullet & \bullet & & & & \\
\bullet & & \bullet & & & \bullet & & & \\
 & & \bullet & & & \bullet & \bullet & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & & & & \bullet & \bullet & \bullet & \bullet
\end{bmatrix}
\end{array}
$$

Algorithm: Clique graph-based merging

A new clique graph-based merging strategy

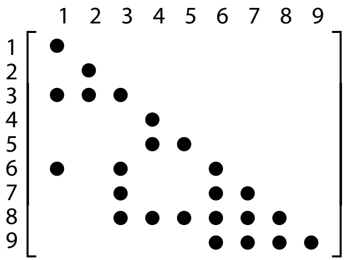


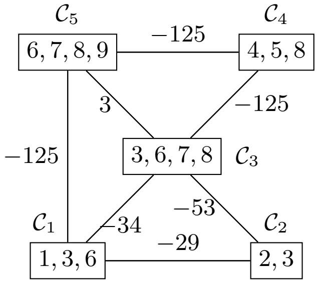


Algorithm: Clique graph-based merging

Compute clique intersection graph;

A new clique graph-based merging strategy



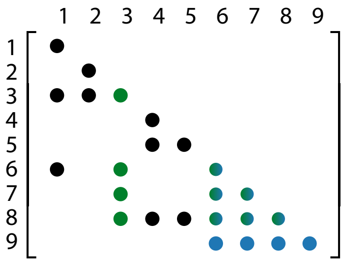


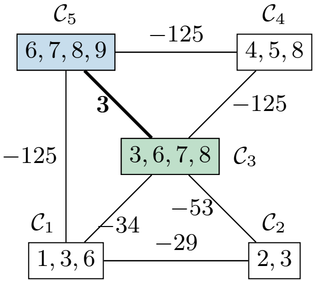
in this example:
$e(\mathcal{C}_i, \mathcal{C}_j) = |\mathcal{C}_i|^3 + |\mathcal{C}_j|^3 - |\mathcal{C}_i \cup \mathcal{C}_j|^3$

Algorithm: Clique graph-based merging

Compute clique intersection graph;
Compute edge weights $w_{ij} = e(\mathcal{C}_i, \mathcal{C}_j)$;

A new clique graph-based merging strategy



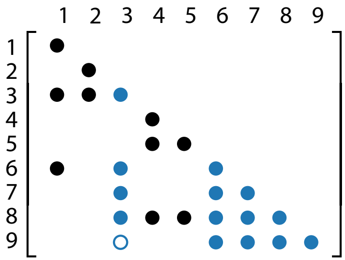


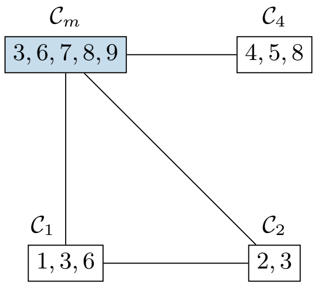
in this example:
$e(\mathcal{C}_i, \mathcal{C}_j) = |\mathcal{C}_i|^3 + |\mathcal{C}_j|^3 - |\mathcal{C}_i \cup \mathcal{C}_j|^3$

Algorithm: Clique graph-based merging

Compute clique intersection graph;
Compute edge weights $w_{ij} = e(\mathcal{C}_i, \mathcal{C}_j)$;
while $w_{ij} > 0$ exists:

A new clique graph-based merging strategy



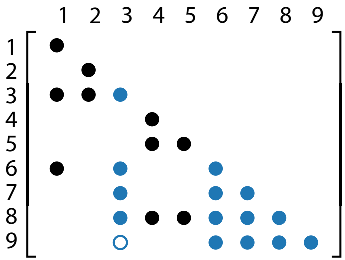


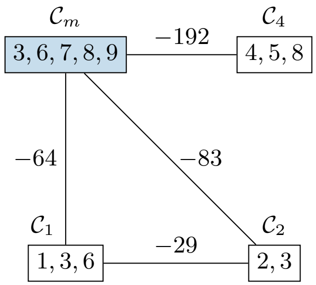
in this example:
$e(\mathcal{C}_i, \mathcal{C}_j) = |\mathcal{C}_i|^3 + |\mathcal{C}_j|^3 - |\mathcal{C}_i \cup \mathcal{C}_j|^3$

Algorithm: Clique graph-based merging

Compute clique intersection graph;
Compute edge weights $w_{ij} = e(\mathcal{C}_i, \mathcal{C}_j)$;
while $w_{ij} > 0$ exists:
  Merge $\mathcal{C}_i, \mathcal{C}_j$ with max weight $\rightarrow \mathcal{C}_m$;

A new clique graph-based merging strategy





in this example:
$e(\mathcal{C}_i, \mathcal{C}_j) = |\mathcal{C}_i|^3 + |\mathcal{C}_j|^3 - |\mathcal{C}_i \cup \mathcal{C}_j|^3$

Algorithm: Clique graph-based merging

Compute clique intersection graph;
Compute edge weights $w_{ij} = e(\mathcal{C}_i, \mathcal{C}_j)$;
while $w_{ij} > 0$ exists:
  Merge $\mathcal{C}_i, \mathcal{C}_j$ with max weight $\rightarrow \mathcal{C}_m$;
  Update edge weights connected to $\mathcal{C}_m$;

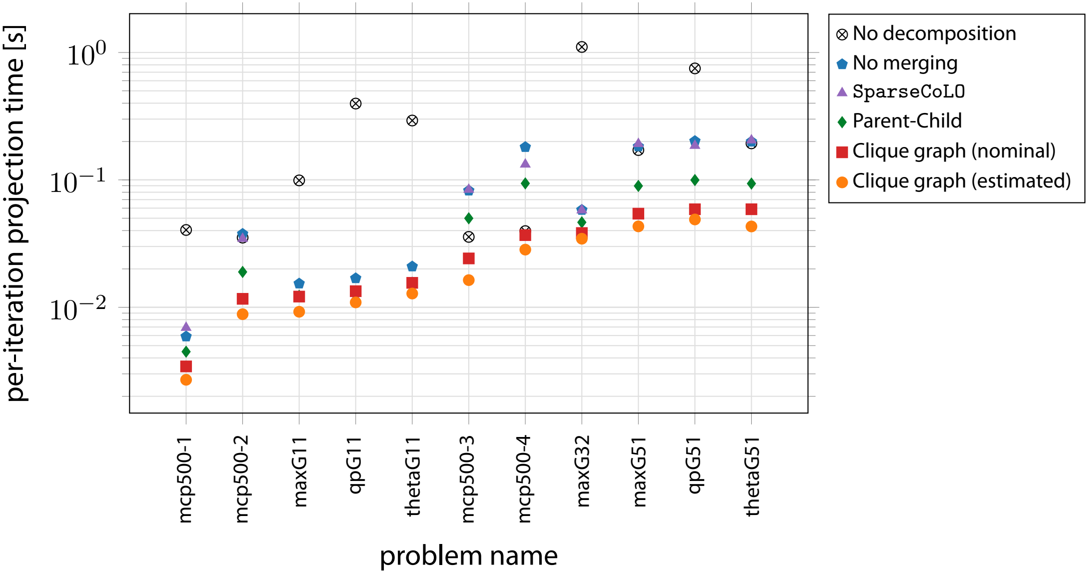
Benchmarks

- **Goal:** Reduce the projection time of our first-order ADMM solver `COSMO`
- **Problem set:** Large, chordal SDPs from the SDPLib collection
- **Setup:** Compare different merge strategies with our solver

a) No decomposition
b) No merging
c) `SparseCoLO` merging
d) Parent-child merging
e) Clique graph merging (nominal)
f) Clique graph merging (estimated)

* SDPLib [Bor99]

Benchmark results



Benchmark results

Table: Solve time for different merging strategies (s).

problem	NoDe[1]	NoMer[2]	SpCo[3]	ParCh[4]	CG1[5]	CG2[6]
maxG11	29.7	4.11	7.9	3.69	**2.72**	2.82
maxG32	320.98	21.12	27.08	13.09	**12.47**	15.79
maxG51	29.12	28.04	19.86	9.59	**5.67**	8.25
mcp500-1	10.28	1.04	1.19	0.78	0.47	**0.37**
mcp500-2	8.9	10.25	7.61	5.97	2.08	**1.95**
mcp500-3	7.66	22.69	30.45	15.76	5.41	**4.35**
mcp500-4	11.63	51.37	60.52	21.92	**5.32**	8.74
qpG11	173.81	6.05	6.48	7.65	4.14	**3.87**
qpG51	607.61	138.38	155.04	150.14	113.87	**85.19**
thetaG11	225.89	8.28	37.16	10.24	9.01	**5.95**
thetaG51	505.33	82.48	587.79	103.47	**28.28**	78.08

[1] No decomposition
[2] No merging
[3] `SparseCoLO`
[4] Parent-Child
[5] Clique graph (nominal)
[6] Clique graph (estimated)

Conclusion:

- Novel clique graph based merging strategy
- Considers all pair-wise merges
- Customisable to solver algorithm and hardware used
- Reduced per-iteration time of our solver

Conclusion:

- Novel clique graph based merging strategy
- Considers all pair-wise merges
- Customisable to solver algorithm and hardware used
- Reduced per-iteration time of our solver

Future work:

- Test with interior-point solver
- Extend strategy to consider multithreading

Conclusion:

- Novel clique graph based merging strategy
- Considers all pair-wise merges
- Customisable to solver algorithm and hardware used
- Reduced per-iteration time of our solver

Future work:

- Test with interior-point solver
- Extend strategy to consider multithreading

Solver package available:

`https://github.com/oxfordcontrol/COSMO.jl`

Conclusion:

- Novel clique graph based merging strategy
- Considers all pair-wise merges
- Customisable to solver algorithm and hardware used
- Reduced per-iteration time of our solver

Future work:

- Test with interior-point solver
- Extend strategy to consider multithreading

Solver package available:

`https://github.com/oxfordcontrol/COSMO.jl`

Questions? michael.garstka[at]eng.ox.ac.uk

Bibliography

J. Agler, W. Helton, S. McCullough, and L. Rodman.
Positive semidefinite matrices with a given sparsity pattern.
Linear Algebra and its Applications, 107:101–149, 1988.

M. S. Andersen and L Vandenberghe.
CHOMPACK: A Python package for chordal matrix computations, 2015.

B. Borchers.
SDPLIB 1.2, A library of semidefinite programming test problems.
Optimization Methods and Software, 11(1-4):683–690, 1999.

K. Fujisawa, S. Kim, M. Kojima, Y. Okamoto, and M. Yamashita.
User's manual for SparseCoLO: Conversion methods for sparse conic-form linear optimization problems.
Report B-453, Dept. of Math. and Comp. Sci. Japan, Tech. Rep., pages 152–8552, 2009.

R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz.
Positive definite completions of partial Hermitian matrices.