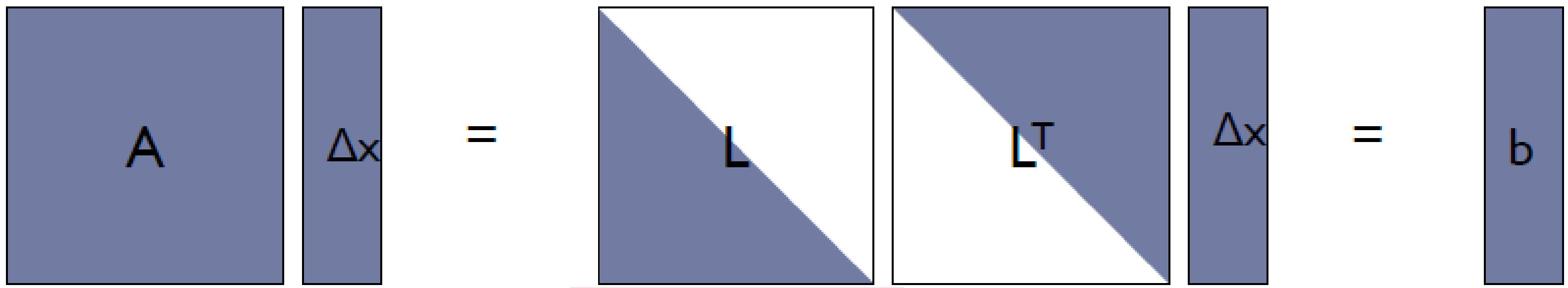
Variable reordering strategies for SLAM

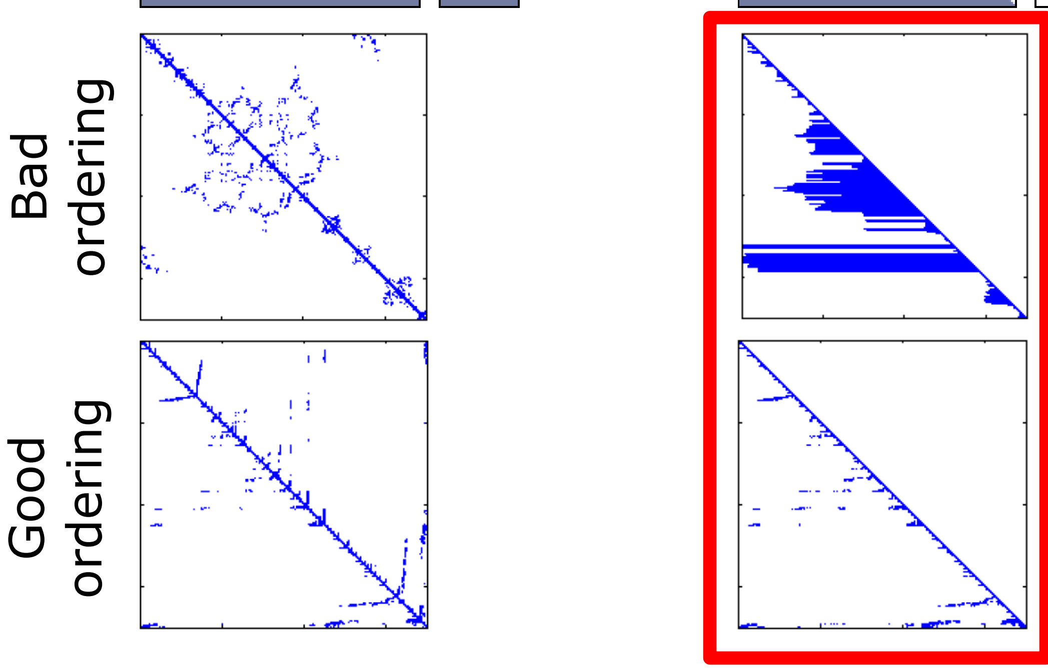
Pratik Agarwal and Edwin Olson

University of Freiburg and University of Michigan

Graph based SLAM

- Linearized system of constraints $Ax = b$

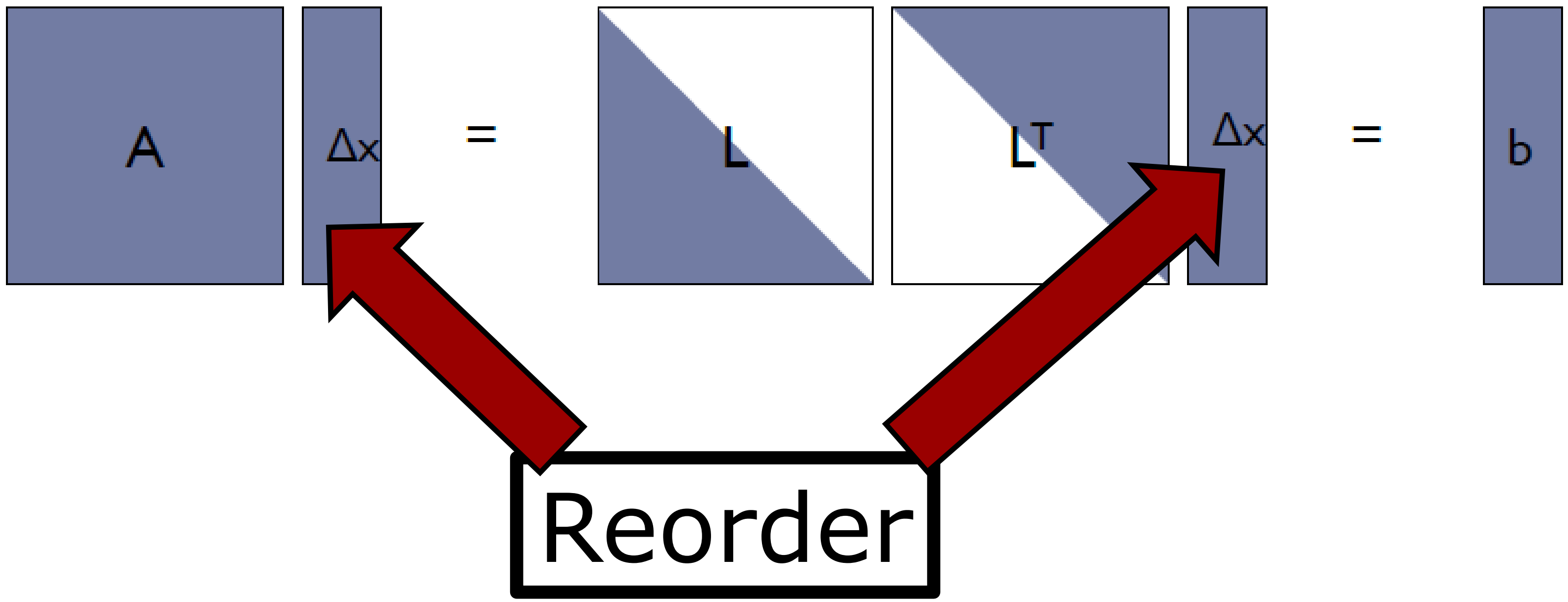




24 times more zeros

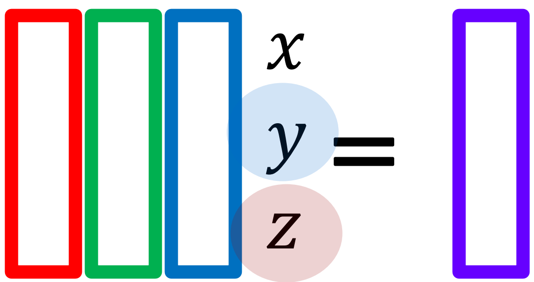
Graph based SLAM

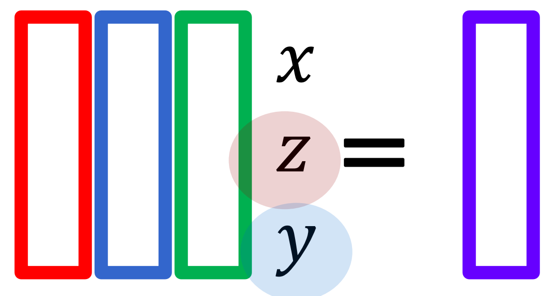
- Linearized system of constraints $Ax = b$



Reordering

$$
\begin{aligned}
..x + ..y + ..z &= ..\\
..x + ..y + ..z &= ..\\
..x + ..y + ..z &= ..
\end{aligned}
$$



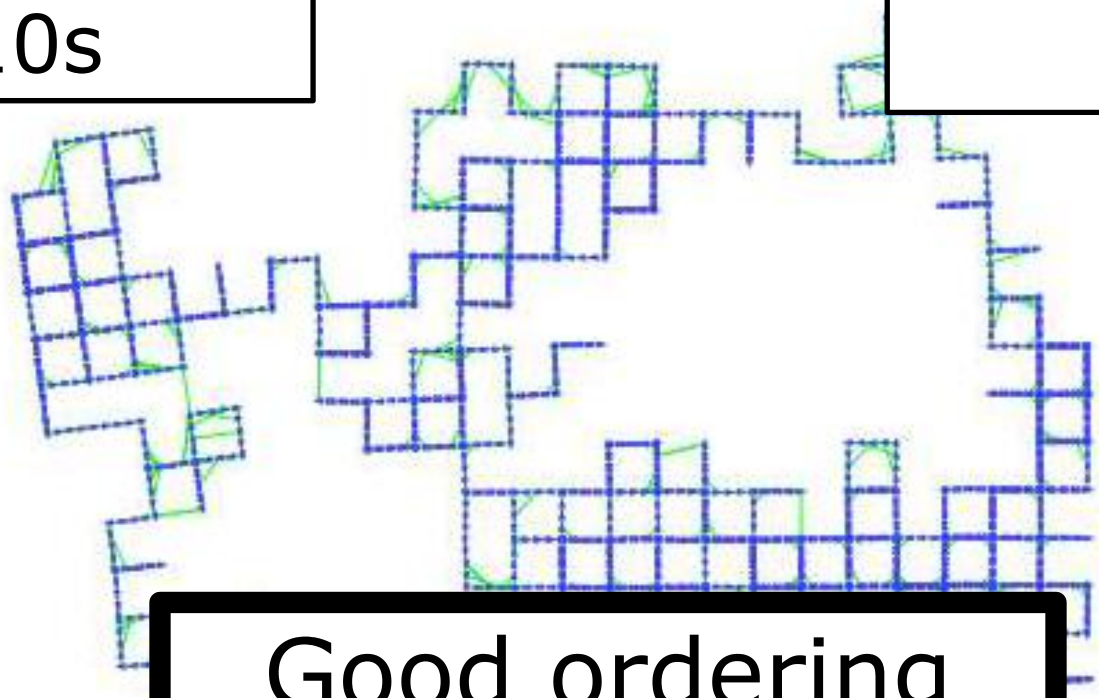


Reordering the variables x, y, z

Reordering is essential for SLAM

Bad ordering
10s

Good ordering
0.1s



Good ordering
100 times faster

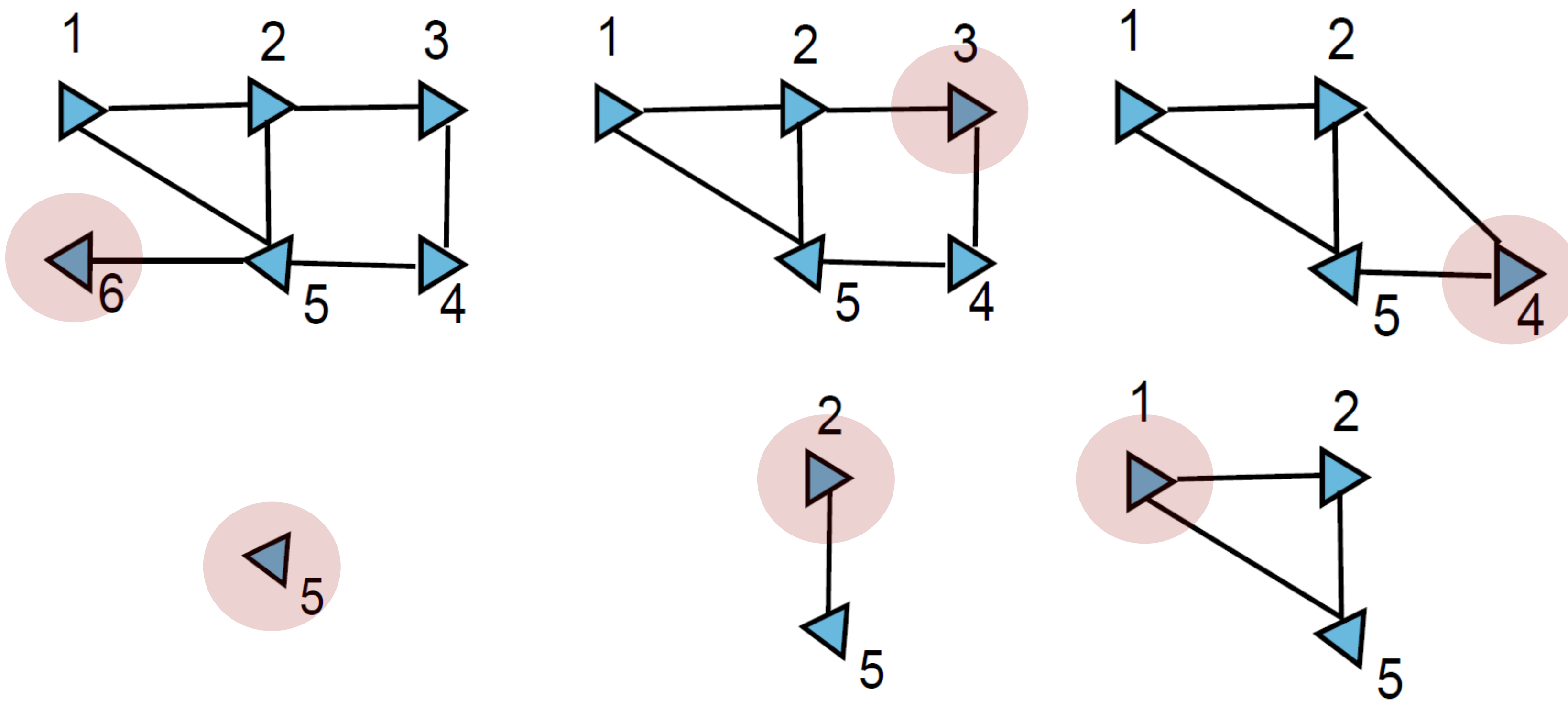
Contribution

- Find best state-of-art method
- A simple easy alternative

All methods compute identical x

Exact minimum degree (EMD)

- Remove node with min edges
- Connect neighbors



Our method: bucket heap AMD

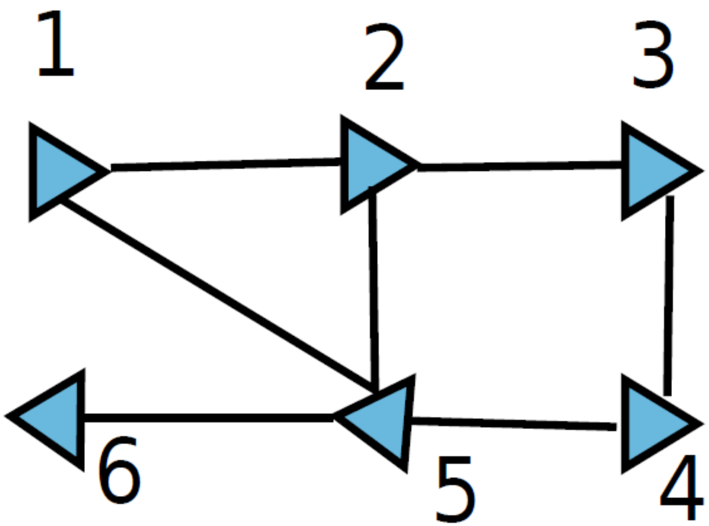
Advantage over EMD:

- Single query - multiple vertex elimination

How?

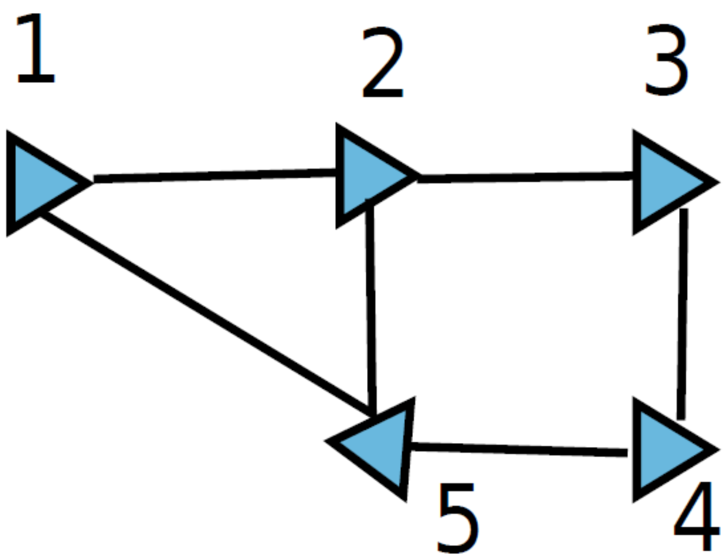
- Lists of vertices with similar #edges

BHAMD in action

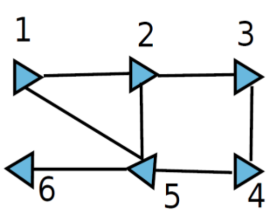


Key	Vertices
1	**6**
2	**1, 3, 4**
3	**2**
4	**5**

BHAMD in action

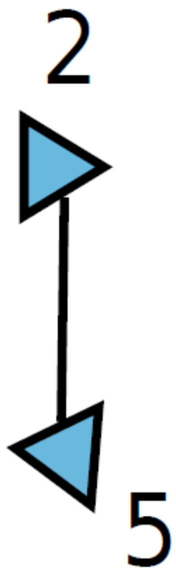


Key	Vertices
2	**1, 3, 4**
3	**2**
4	**5***

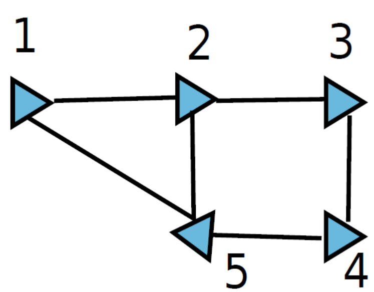


Key	Vertices
1	**6**
2	**1, 3, 4**
3	**2**
4	**5**

BHAMD in action

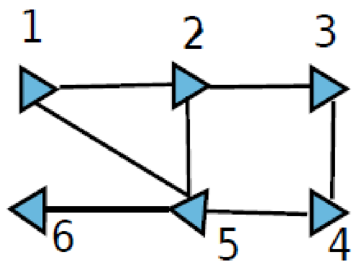


Key	Vertices
3	**2***
4	**5***



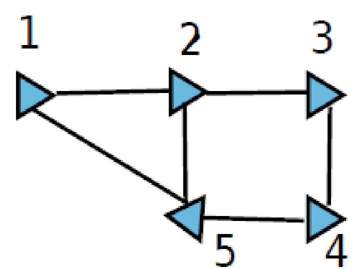
Key	Vertices
2	**1, 3, 4**
3	**2**
4	**5***

BHAMD in action

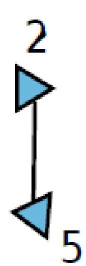


Key	Vertices
1	6
2	1, 3, 4
3	2
4	5

Multiple vertices eliminated

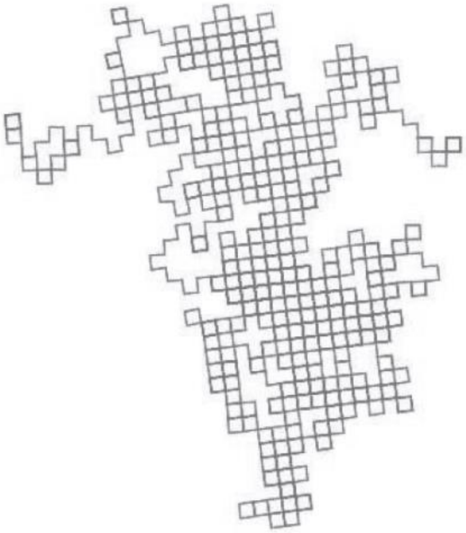


Key	Vertices
2	1, 3, 4
3	2
4	5*



Key	Vertices
3	2*
4	5*

BHAMD on a 10,000 node graph

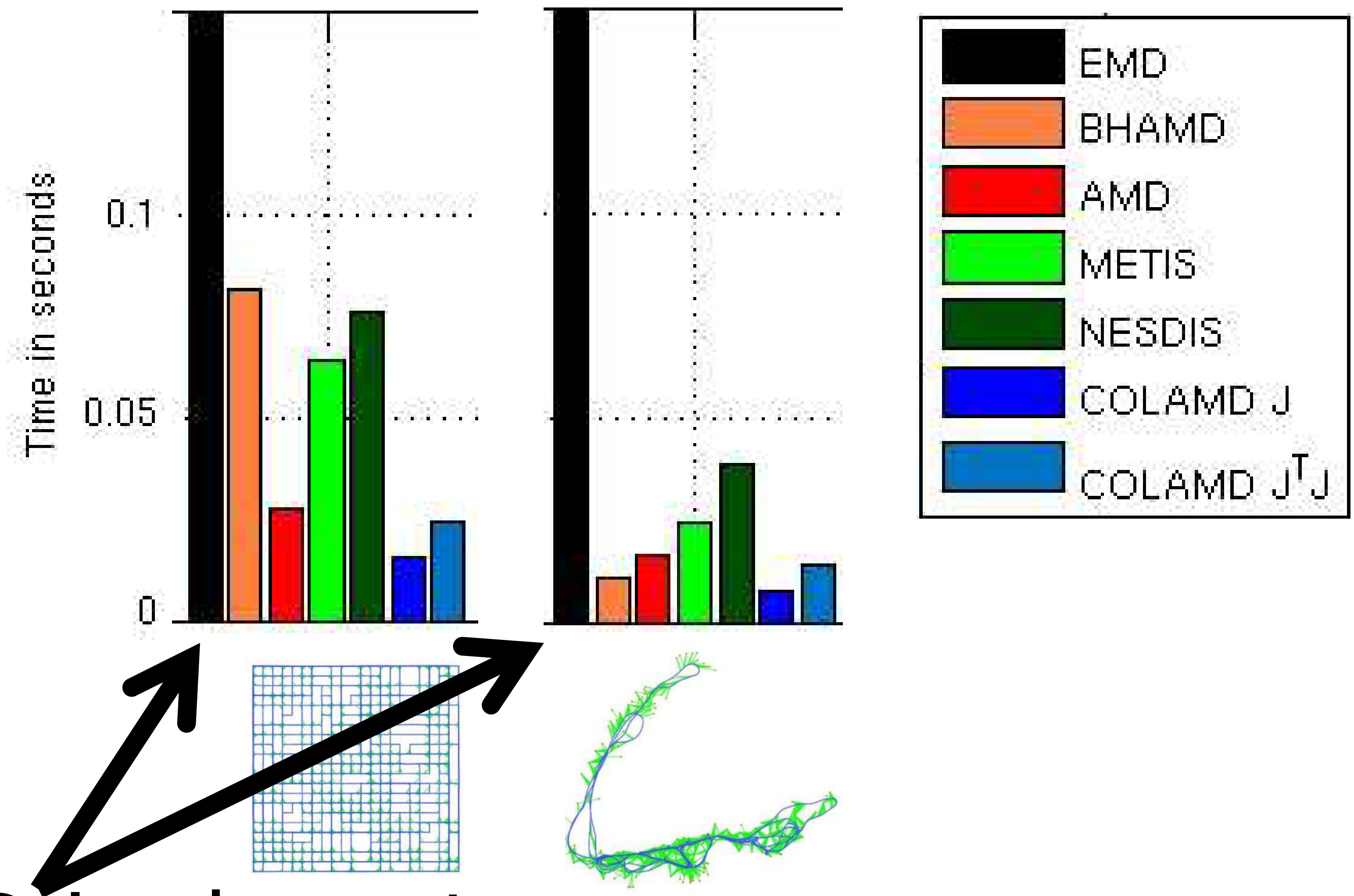


- EMD - 10,000 steps
- BHAMD - 107 steps
- Max heap size in BHAMD 42

State-of-the-art techniques

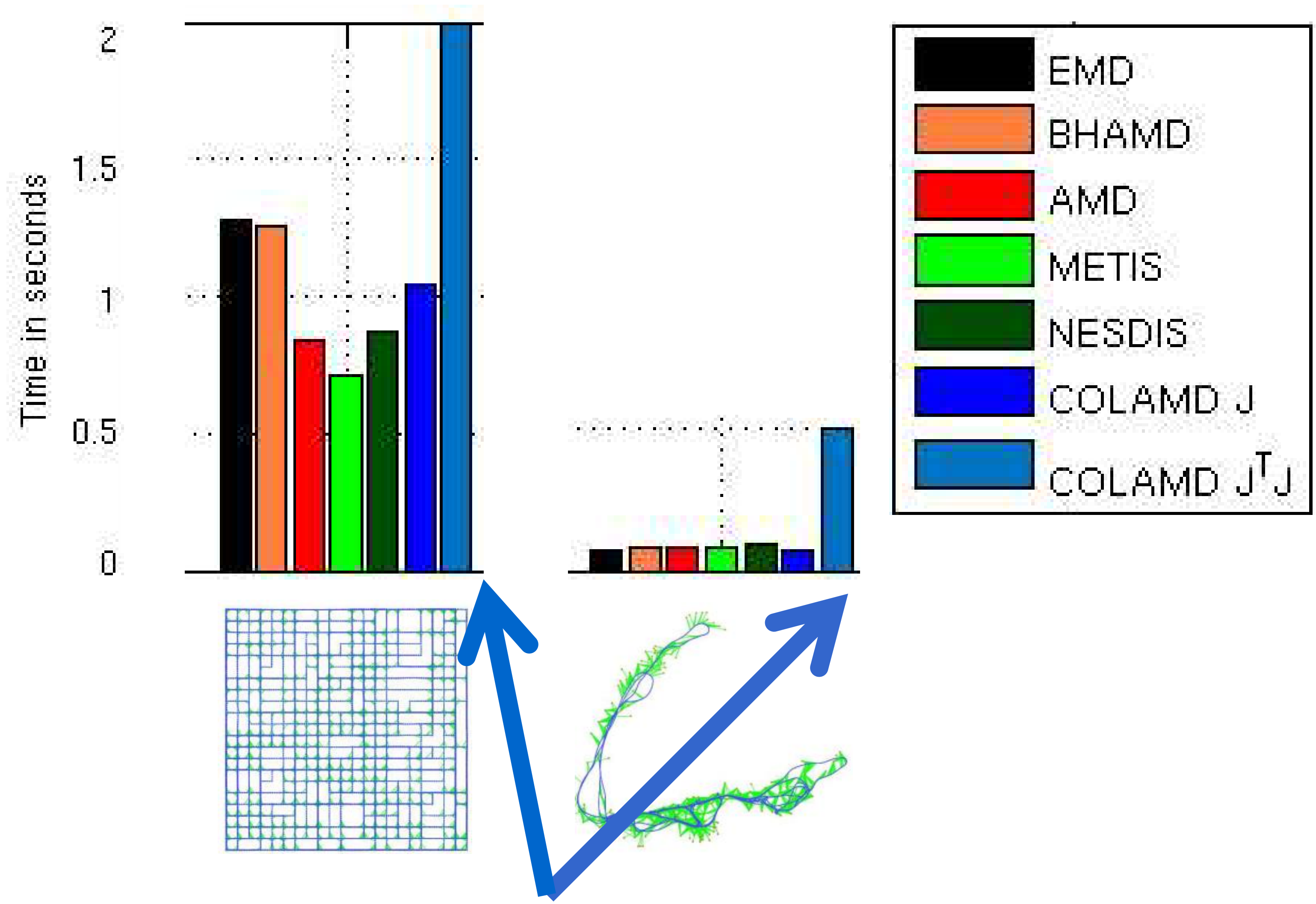
- **AMD** Approximate Minimum Degree
- **COLAMD** Column Approximate Minimum Degree
- **NESDIS** Nested Dissection
- **METIS** Serial Graph Partition

Evaluation – reordering time



- EMD is slowest

Evaluation – solve time



- COLAMD on A is slowest

Further room for improvement?

Ideally

- consider all possible orderings

1000 nodes ⟶ $\mathbf{10^{2567}}$ **orderings**

Instead

- Local changes in existing orderings

maximum improvement of 0.5%

(with 1 week compute time)

Conclusion

1. All methods comparable ~~EMD & COLAMD on A~~

2. BHAMD - simple yet competitive

3. Negligible improvement with small changes

Questions

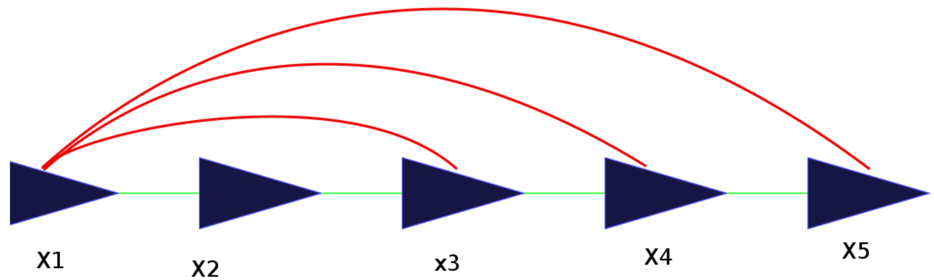
Reordering matrix - P

$$Ax = b$$
$$(PA)x_p = Pb$$
$$x = P^{-1}x_p$$

P^{-1} is easy, since $P^{-1} = P^T$

- Computing the best reordering matrix P is NP hard
- Heuristics work quite well

Toy example



Bad ordering

$$J = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} X & X & & & \\ & X & X & & \\ & & X & X & \\ & & & X & X \\ X & & X & & \\ X & & & X & \\ X & & & & X \end{pmatrix} \end{array} \quad A = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} X & X & X & X & X \\ X & X & X & & \\ X & X & X & X & \\ X & & X & X & X \\ X & & & X & X \end{pmatrix} \end{array} \quad L = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} X & & & & \\ X & X & & & \\ X & X & X & & \\ X & X & X & X & \\ X & X & X & X & X \end{pmatrix} \end{array}$$

Good ordering

$$J_r = \begin{array}{c} \begin{array}{ccccc} 5 & 2 & 3 & 4 & 1 \end{array} \\ \begin{pmatrix} & X & & & X \\ & X & X & & \\ & & X & X & \\ X & & & X & \\ & & X & & X \\ & & & X & X \\ X & & & & X \end{pmatrix} \end{array} \quad A_r = \begin{array}{c} \begin{array}{ccccc} 5 & 2 & 3 & 4 & 1 \end{array} \\ \begin{pmatrix} X & & & X & X \\ & X & X & & X \\ & X & X & X & X \\ X & & X & X & X \\ X & X & X & X & X \end{pmatrix} \end{array} \quad L_r = \begin{array}{c} \begin{array}{ccccc} 5 & 2 & 3 & 4 & 1 \end{array} \\ \begin{pmatrix} X & & & & \\ & X & & & \\ & X & X & & \\ X & & X & X & \\ X & X & X & X & X \end{pmatrix} \end{array}$$

Multiple Min Degree (MMD) vs BHAMD

- Similarity

  multiple elimination

- Difference

  MMD computes and eliminates independent nodes

  MMD is still exact unlike BHAMD

But I use CSparse with COLAMD

- It calls AMD if the matrix is PSD
- Be careful when opening the box

Graph based SLAM

- Linearized system of constraints $Ax = b$