A Reinforcement-Learning Framework for Symmetric Reordering in Sparse Cholesky Factorization

Maxime Gasse, Defeng Liu, Andrea Lodi, Mathieu Tanneau

CERC DS4DM, GERAD

June 28, 2018

How I will attempt to solve a hard problem and then try to understand it

aka: Solve first, then think

Maxime Gasse, Defeng Liu, Andrea Lodi, Mathieu Tanneau

CERC DS4DM, GERAD

June 28, 2018

GERAD
GROUP FOR RESEARCH IN DECISION ANALYSIS

CANADA EXCELLENCE RESEARCH CHAIR
DATA SCIENCE FOR REAL-TIME DECISION-MAKING

POLYTECHNIQUE MONTRÉAL

1. Introduction
2. RL framework
3. Conclusion

1 Introduction

2 RL framework

3 Conclusion

From Interior-Point method [Wright, 1997]:

- Solve (sparse) **symmetric linear system**

$$\Phi \Delta y = \xi, \quad \text{with } \Phi = (A \Theta A^T) \succ 0$$

- Compute (sparse) **Cholesky factorization**

$$\Phi = L \times L^T, \quad \text{with } L \text{ lower triangular}$$

- **BUT** even if Φ is sparse, L may not be!

$\Rightarrow$ re-order rows and columns of Φ before computing L

Ordering matters

$$\Phi = \begin{pmatrix} \times & & & & \times \\ & \times & & & \times \\ & & \times & & \times \\ & & & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} \rightarrow L = \begin{pmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ \times & \times & \times & \times & \times \end{pmatrix}$$ **Good**

Ordering matters

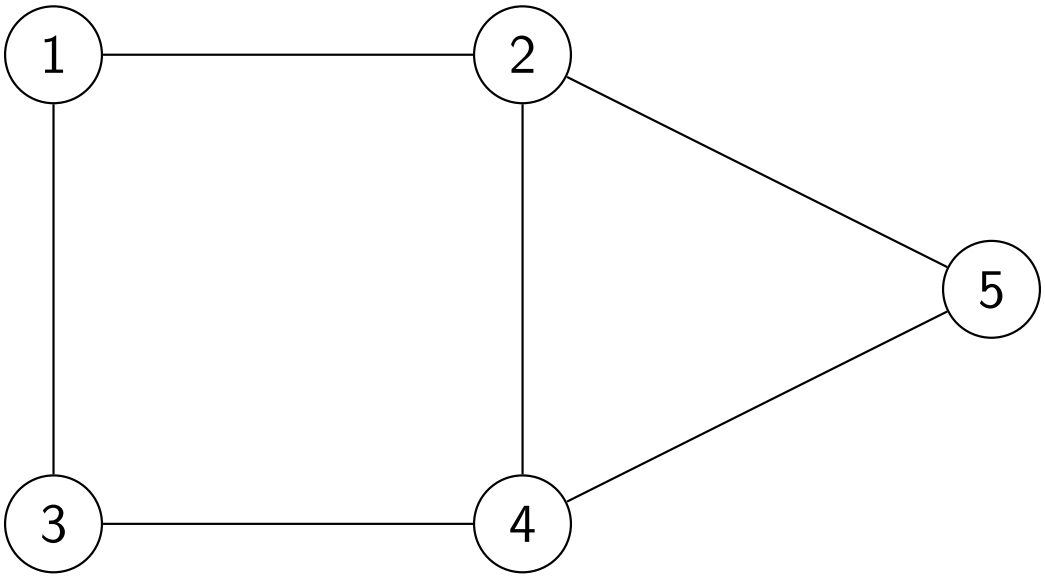
$$\Phi = \begin{pmatrix} \times & & & & \times \\ & \times & & & \times \\ & & \times & & \times \\ & & & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} \rightarrow L = \begin{pmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ \times & \times & \times & \times & \times \end{pmatrix} \quad \textbf{Good}$$

$$\Phi = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & & & \\ \times & & \times & & \\ \times & & & \times & \\ \times & & & & \times \end{pmatrix} \rightarrow L = \begin{pmatrix} \times & & & & \\ \times & \times & & & \\ \times & + & \times & & \\ \times & + & + & \times & \\ \times & + & + & + & \times \end{pmatrix} \quad \textbf{Bad}$$

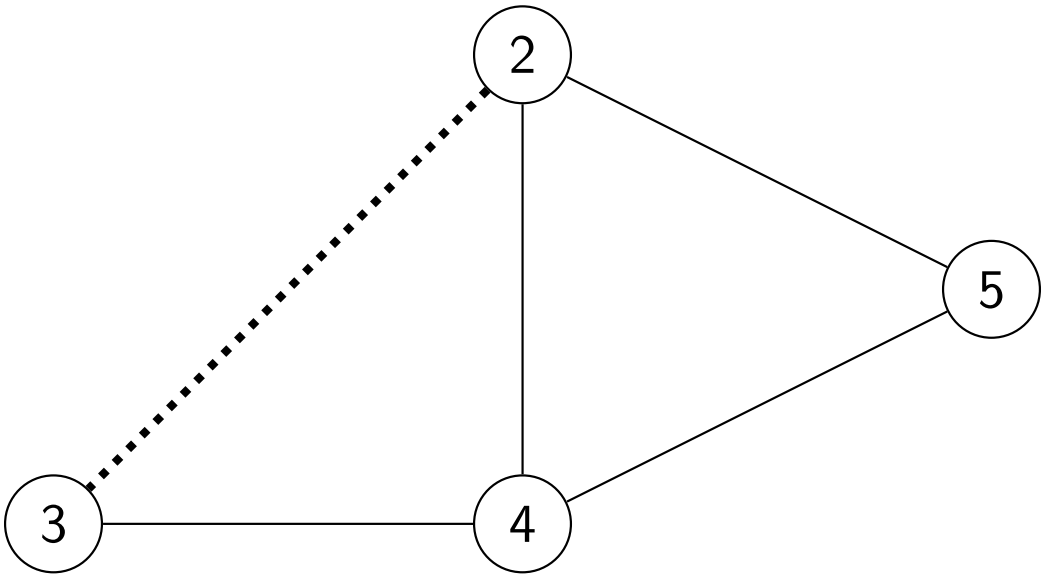
Figure: $\times$ are non-zeros elements of Φ ; $+$ are fill-in (zero in Φ but not in L)

Finding a good ordering reduces to a **Graph elimination problem**:

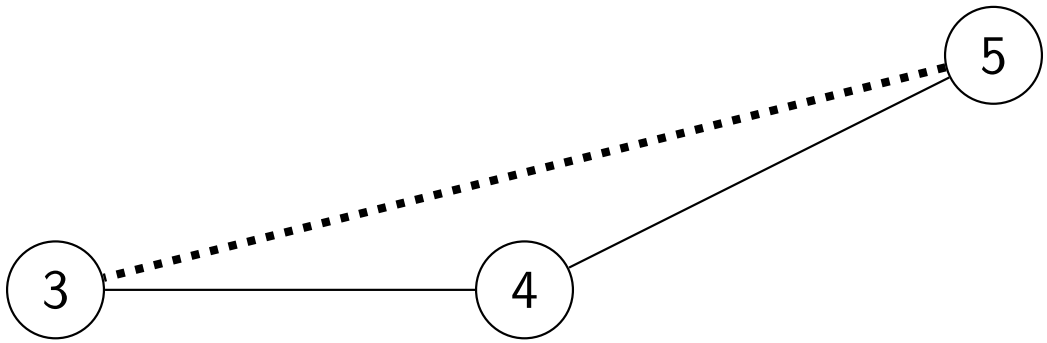
- Input is undirected graph G
- Remove a vertex $v \in G$ and add edges so that neighbours of v form a clique. Repeat until G has no more vertices.
- Goal: find an elimination sequence that minimizes the total number of new edges.



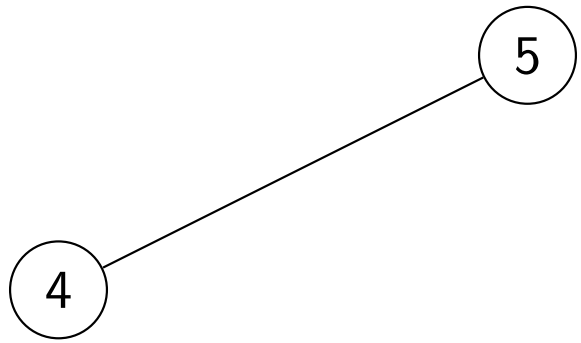
Elimination order:



Elimination order: 1,



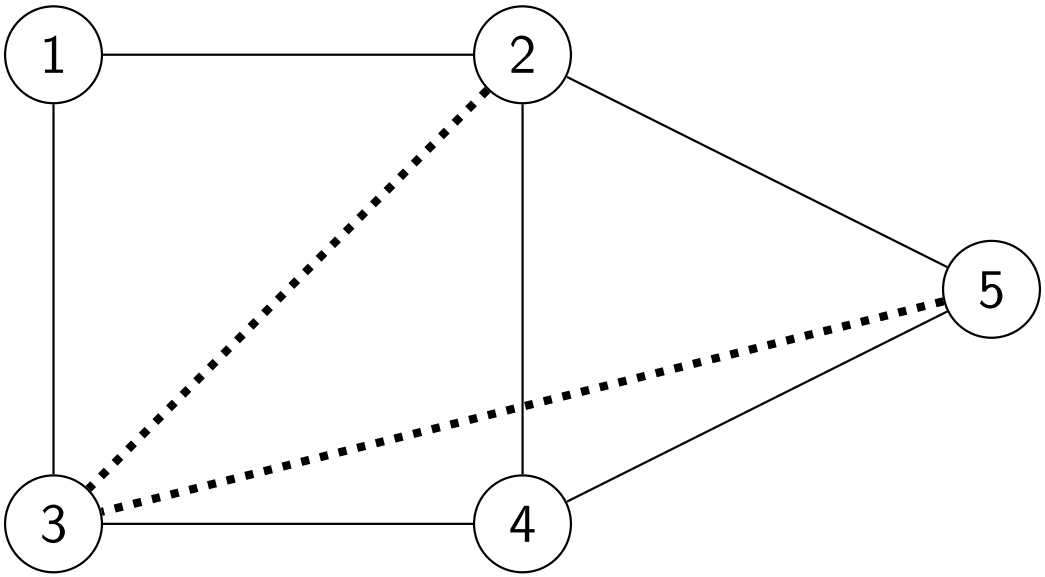
Elimination order: $1, 2,$



Elimination order: $1, 2, 3,$

Elimination order: $1, 2, 3, 4,$

Elimination order: $1, 2, 3, 4, 5$



Elimination order: $1, 2, 3, 4, 5 \Rightarrow 2$ edges added

1 Introduction

2 RL framework

3 Conclusion

Reinforcement Learning

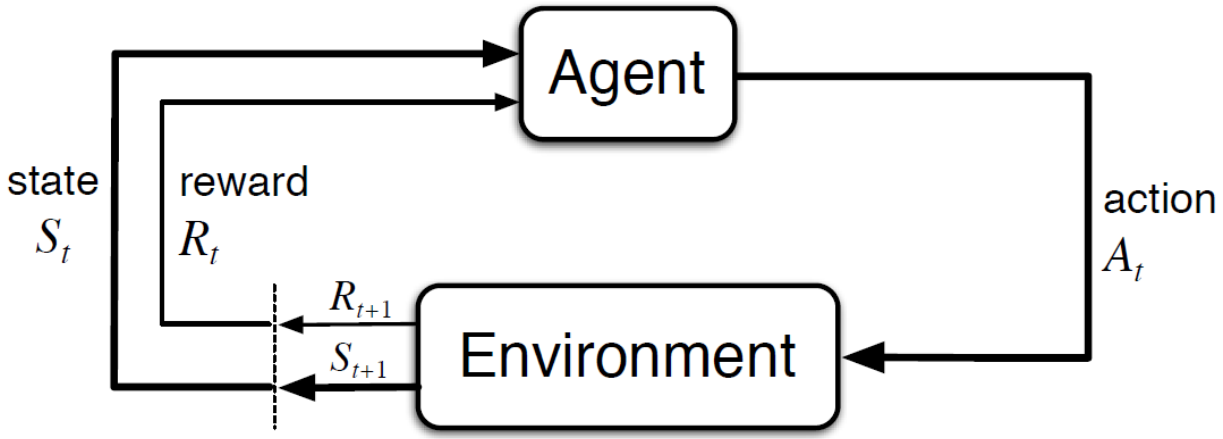
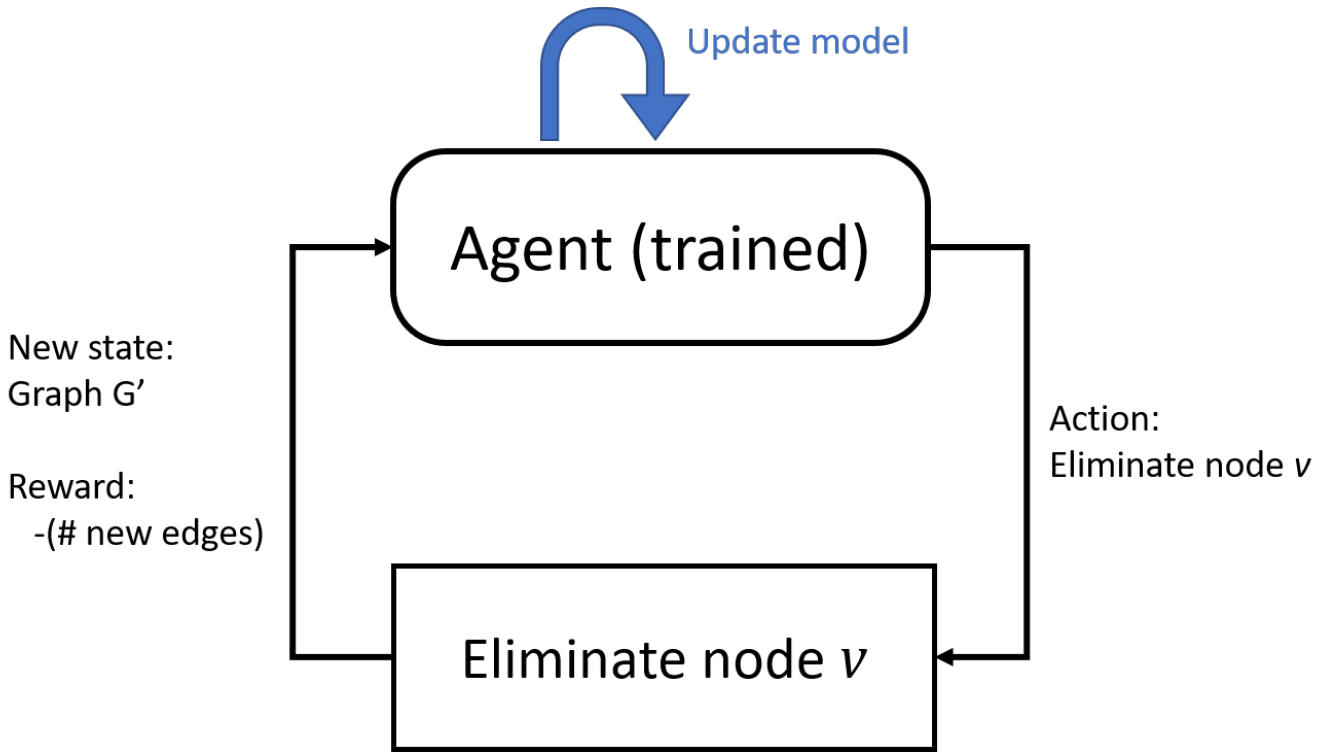


Figure 3.1: The agent–environment interaction in reinforcement learning.

- Maximize long-term cumulative reward
- **Learn a policy** (i.e. which action do I take, given the current state)
- **Theorem:** *there exists an optimal greedy policy*

See *Reinforcement Learning*, Sutton, Barto 2017

Goal: use RL to learn greedy heuristics



Can we learn heuristics...

- ...at all? (likely, yes)
- ...that perform as good as existing ones? (likely, yes)
- ...that also run as fast as existing ones? (likely, no)
- ...for other objectives? (hopefully, yes)

Can we learn heuristics...

- ...at all? (likely, yes)
- ...that perform as good as existing ones? (likely, yes)
- ...that also run as fast as existing ones? (likely, no)
- ...for other objectives? (hopefully, yes)

What can we learn from those (learned) heuristics?

- New heuristics
- Underlying problem structure (by looking at decision rules)

1 Introduction

2 RL framework

3 Conclusion

Take away

- Consider a (combinatorial) optimization problem
- Attempt to solve it using an RL model
- Use that model to infer further knowledge, e.g.: decomposition? parallelism?


Wright, S. (1997).
Primal-Dual Interior-Point Methods.
Society for Industrial and Applied Mathematics.