Carnegie Mellon
THE ROBOTICS INSTITUTE

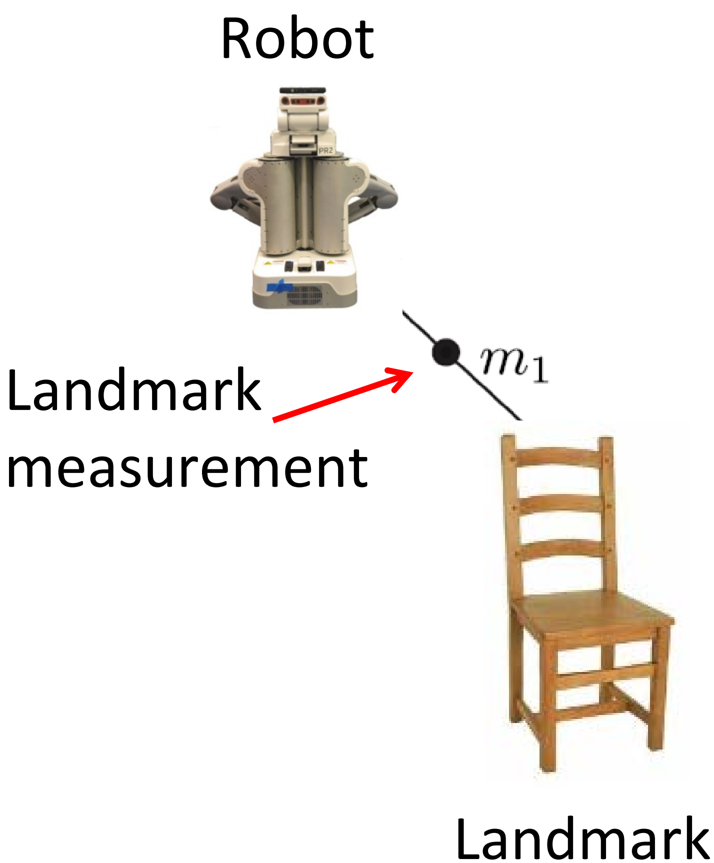
Efficient Incremental Smoothing

SLAM Tutorial @ ICRA 2016

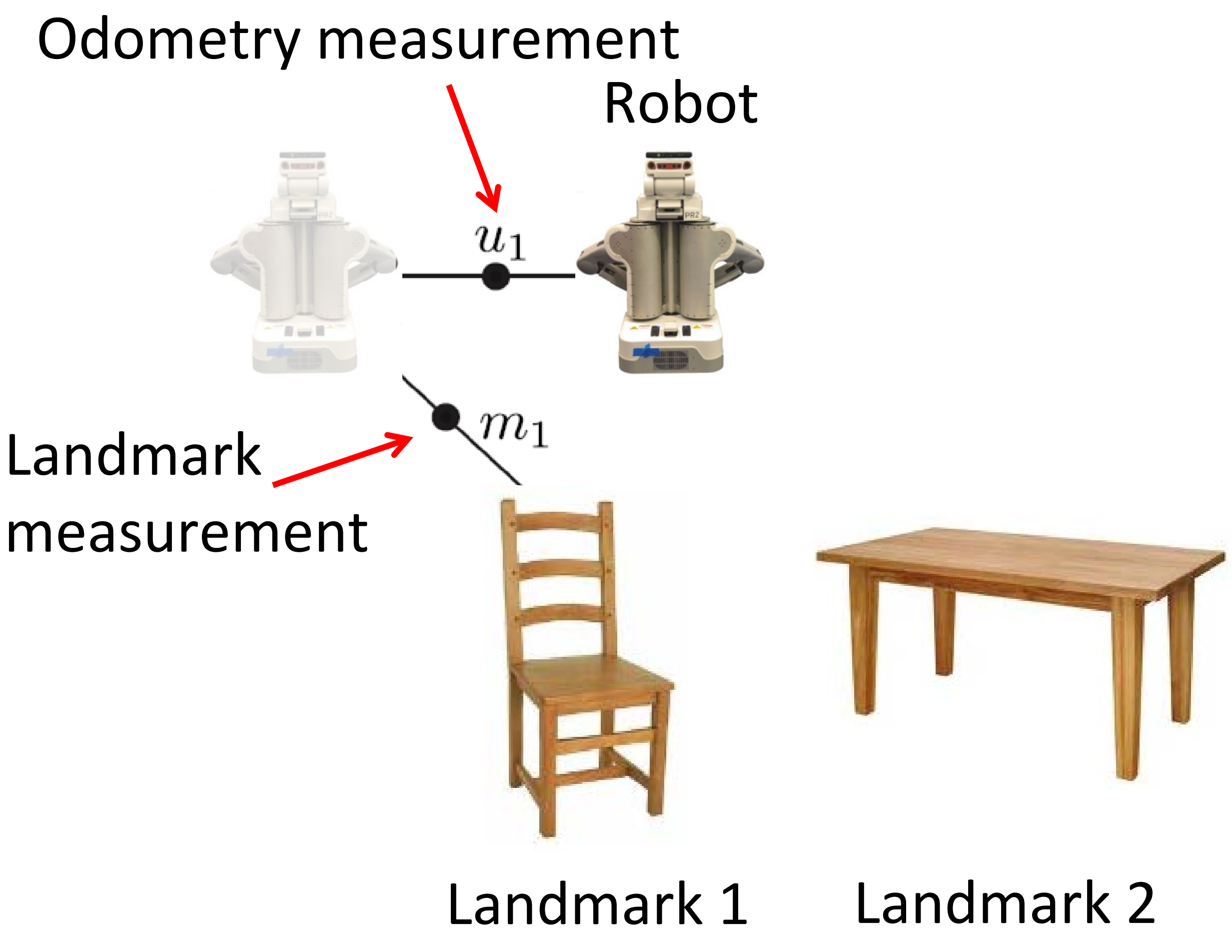
Michael Kaess

May 20, 2016

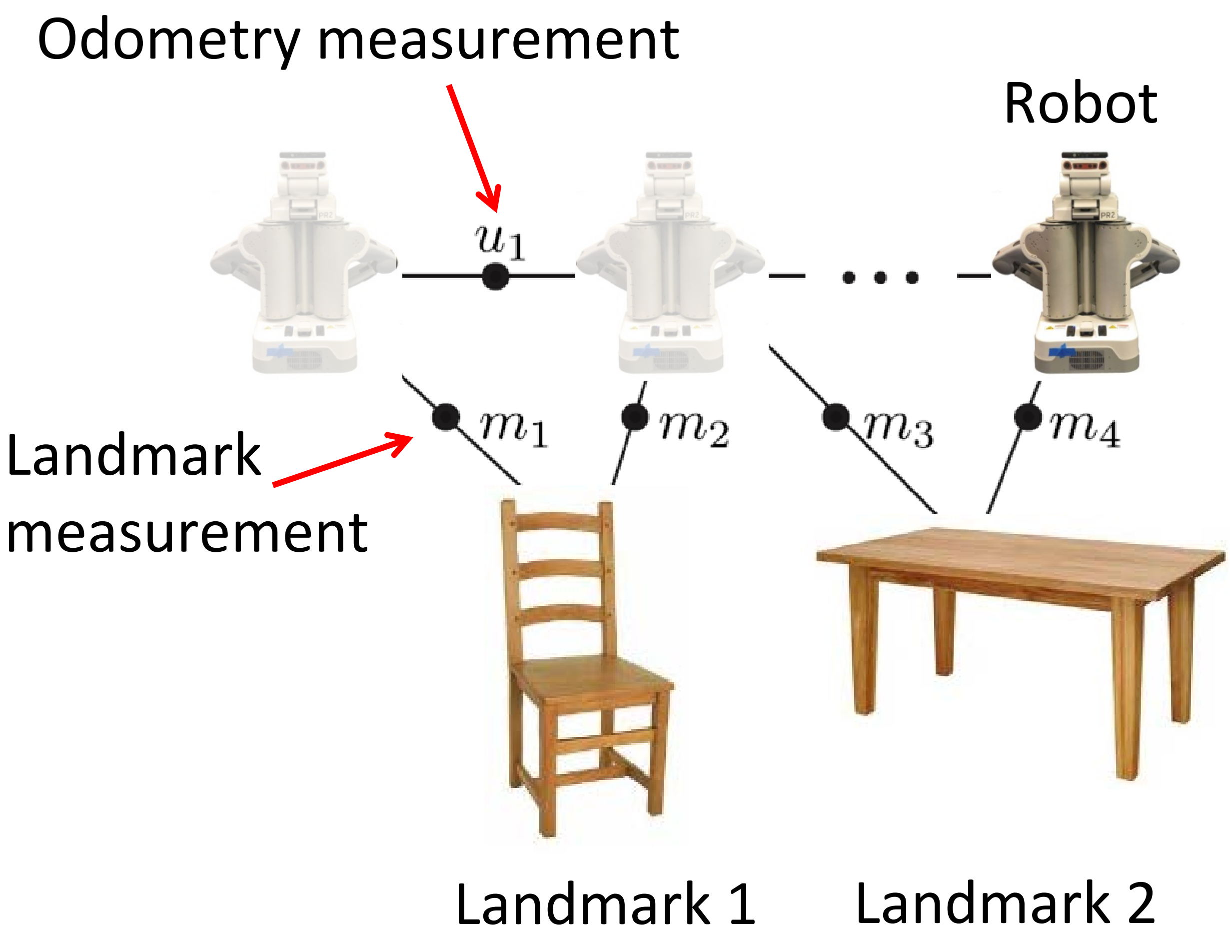
The SLAM Problem (t=0)



The SLAM Problem (t=1)



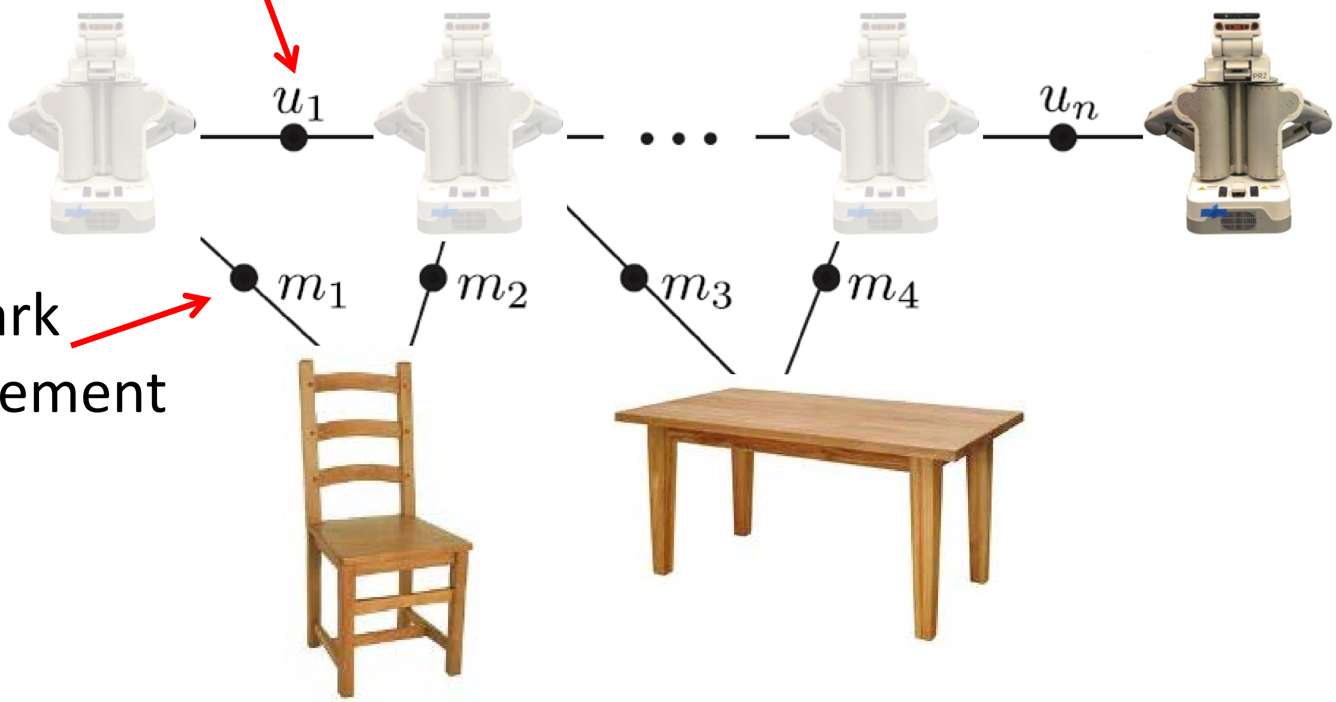
The SLAM Problem (t=n-1)



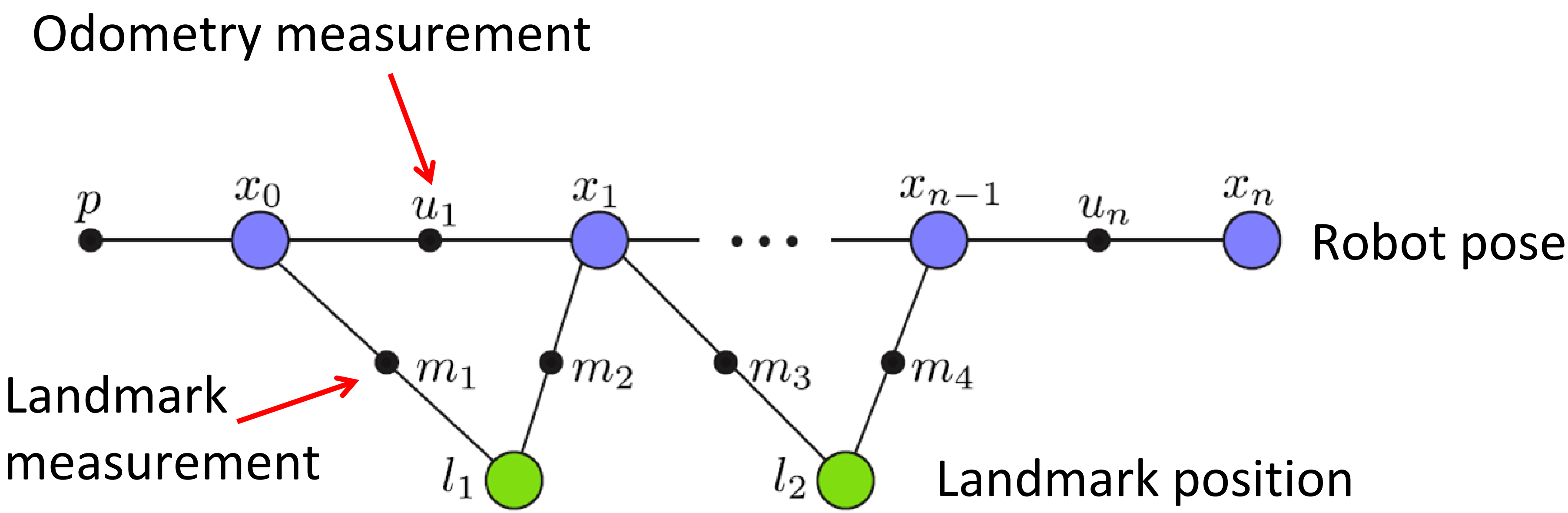
The SLAM Problem (t=n)

Odometry measurement

Landmark measurement

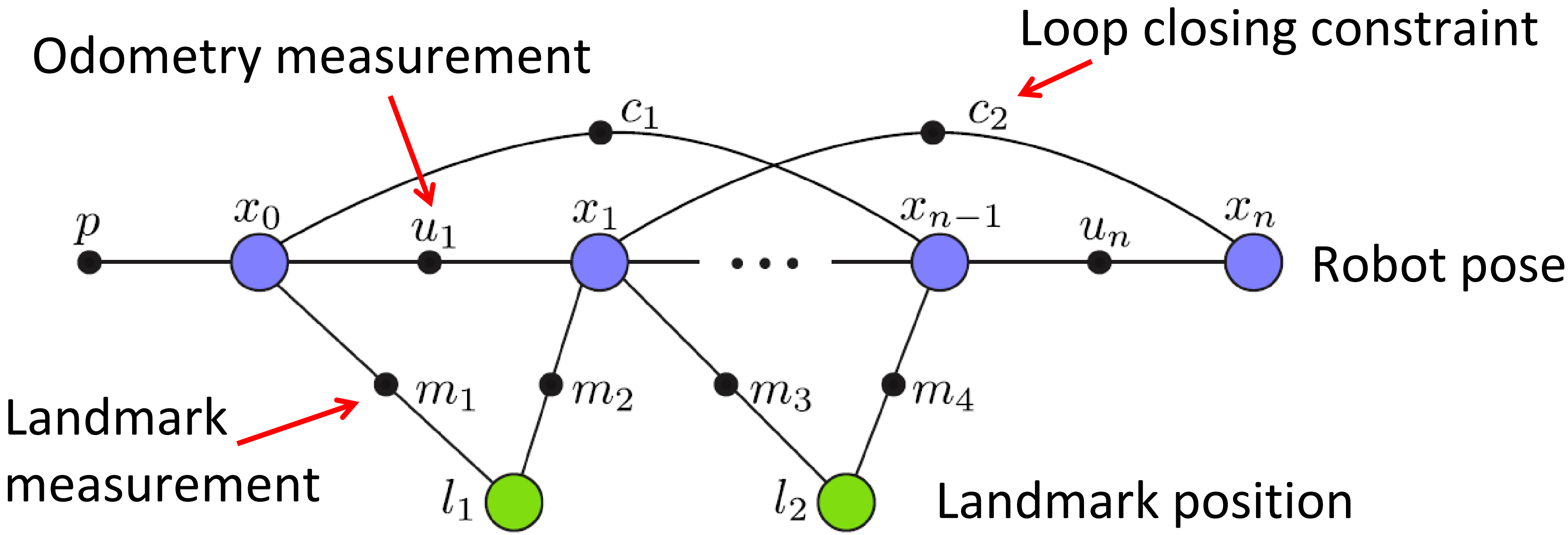


Factor Graph Representation of SLAM

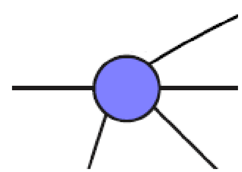


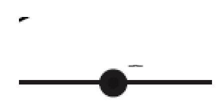
Bipartite graph with ***variable nodes*** and ***factor nodes***

Factor Graph Representation of SLAM



Bipartite graph with ***variable nodes*** and ***factor nodes***



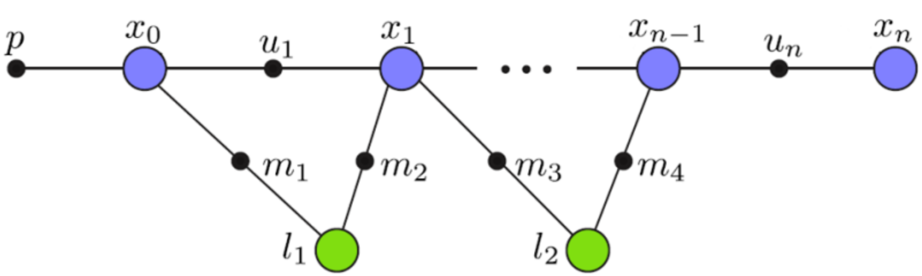


Variables and Measurements

- Variables:

$$\Theta = \{x_0, x_1 \cdots x_n, l_1, l_2\}$$

Might include other quantities such as lines, planes and calibration parameters



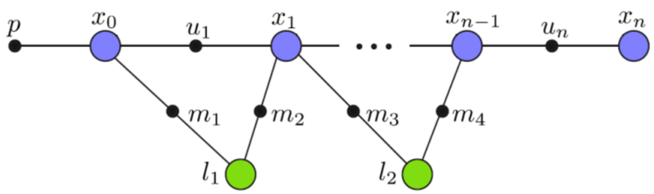
- Measurements:

$$Z = \{p, u_1 \cdots u_n, m_1 \cdots m_4\}$$

p is a prior to fix the gauge freedom (all other measurements are relative!)

[Dellaert & Kaess , IJRR 2006]

SLAM as a Sparse Least-Squares Problem



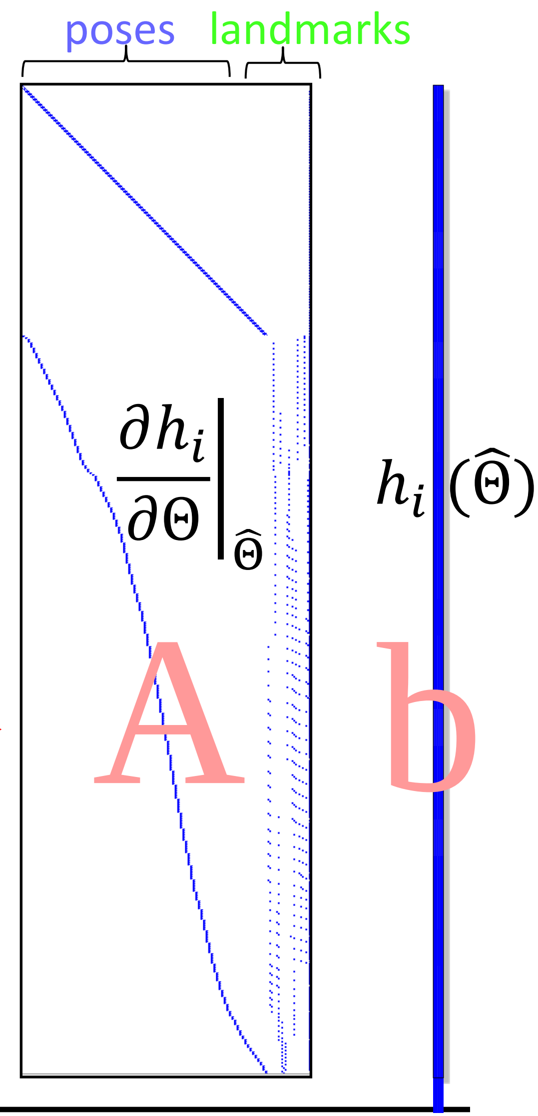
$$\text{argmax}_\Theta \prod_{z \in Z} p\left(z|\Theta\right)$$

Gaussian noise

$$\text{argmin}_\Theta \sum_i \|h_i(\Theta) - z\|^2$$

$$\text{argmin}_\theta \|A\theta - b\|^2$$

Sparse measurement Jacobian



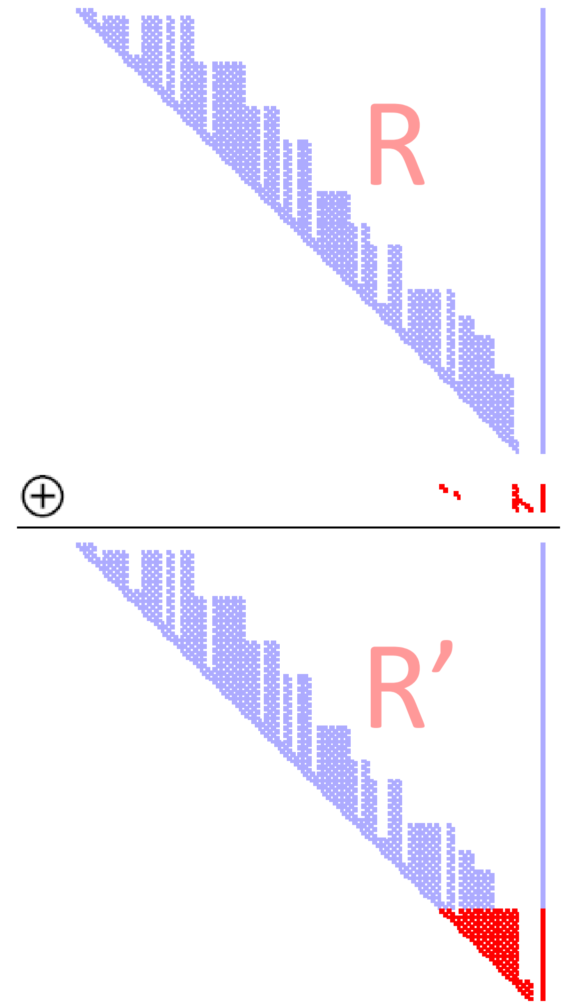
Incremental Smoothing and Mapping (iSAM)

Solving a growing system:

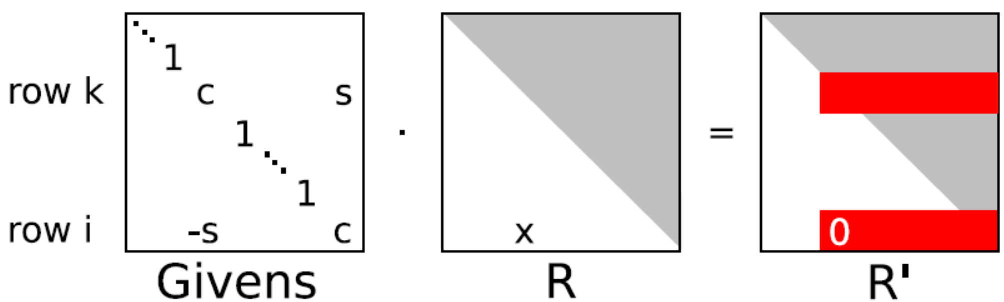
- R factor from previous step
- How do we add new measurements?

Key idea:

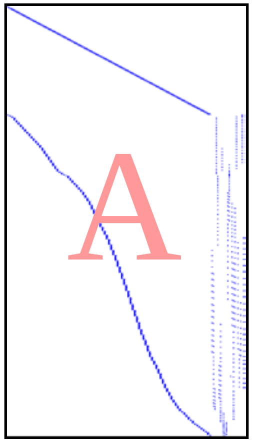
New measurements ->



- Append to existing matrix factorization
- “Repair” using Givens rotations

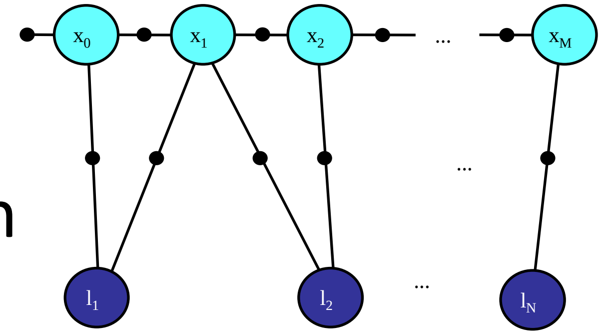


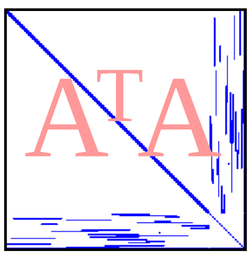
Matrix vs. Graph



Measurement Jacobian

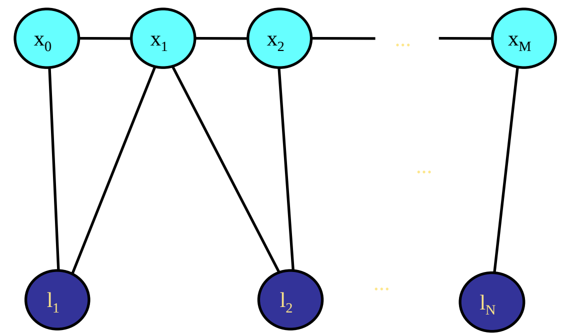
Factor Graph

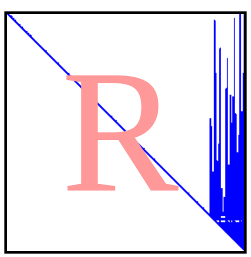




Information Matrix

Markov Random Field

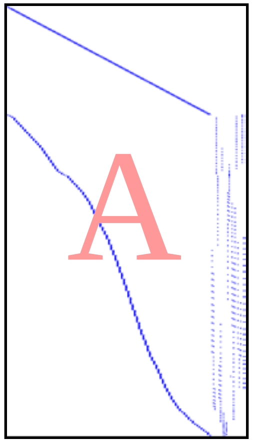




Square Root Inf. Matrix

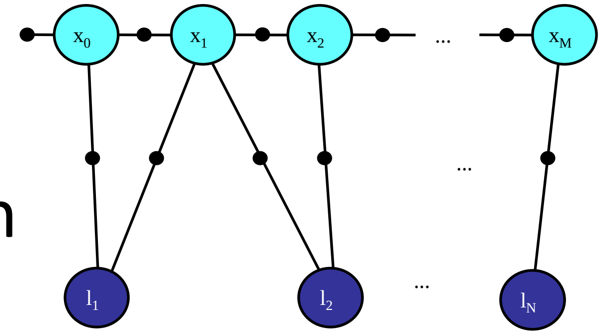
?

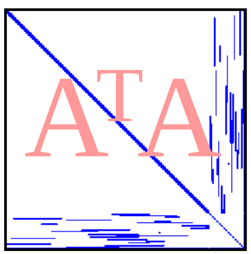
Matrix vs. Graph



Measurement Jacobian

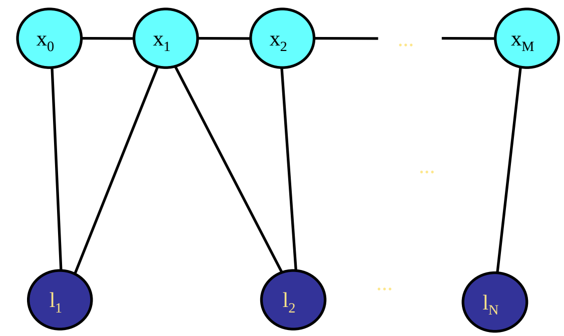
Factor Graph

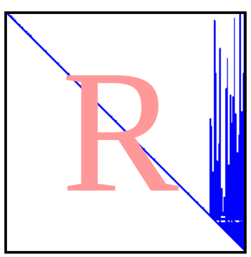




Information Matrix

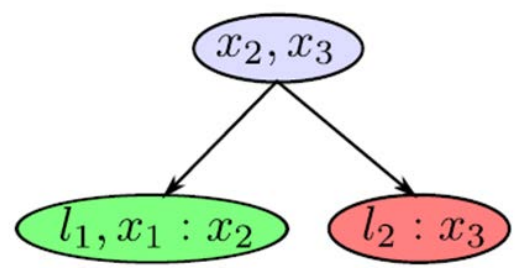
Markov Random Field



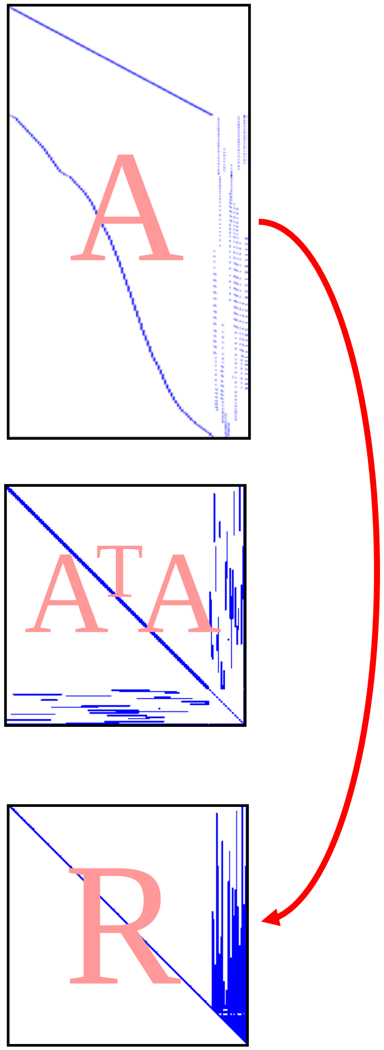


Square Root Inf. Matrix

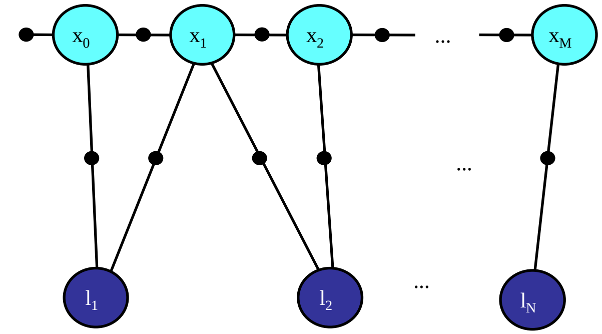
Bayes Tree

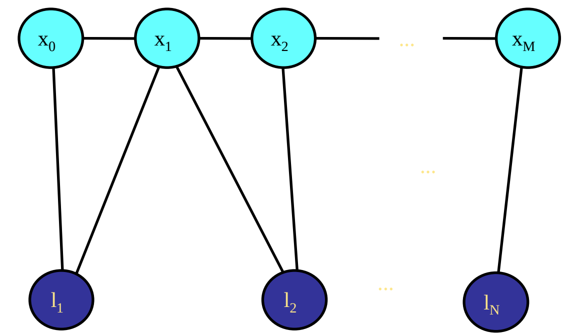


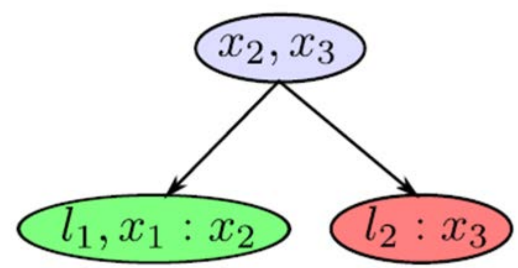
Matrix vs. Graph



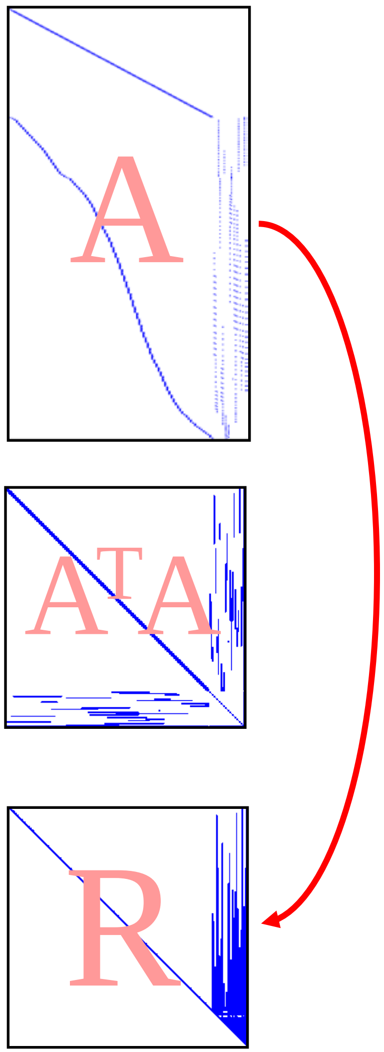
Matrix factorization





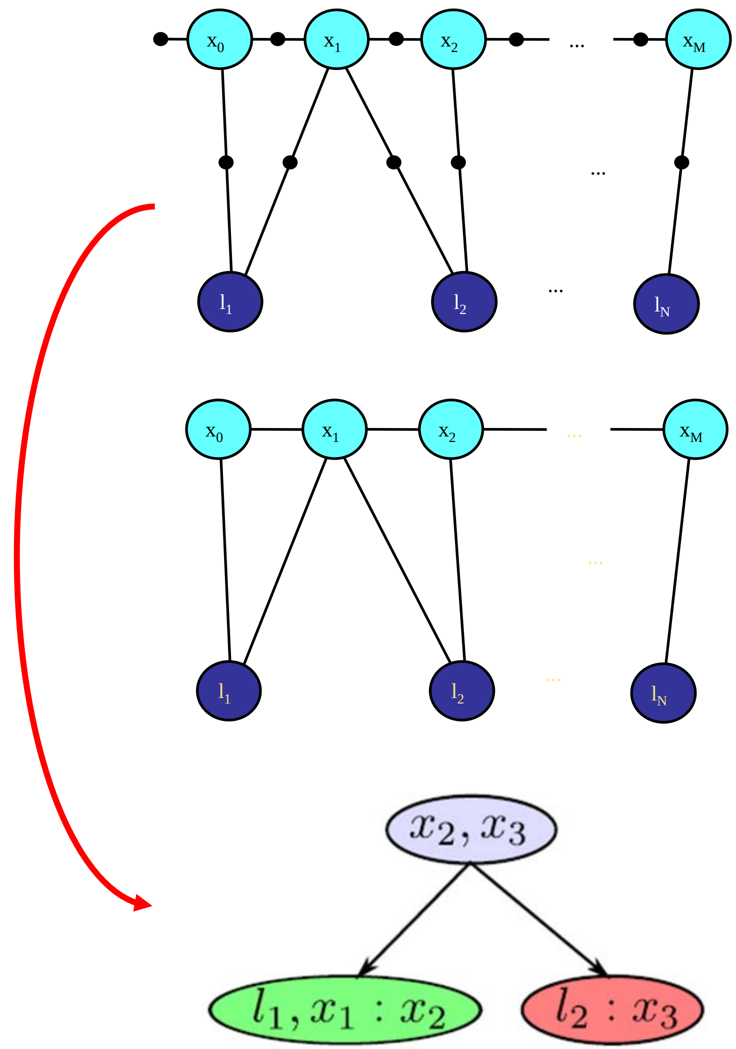


Matrix vs. Graph



Matrix factorization

Variable elimination



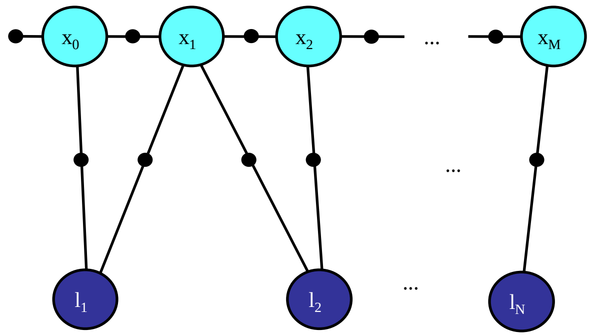
iSAM2: Bayes Tree

Goal: Convert factor graph to tree structure

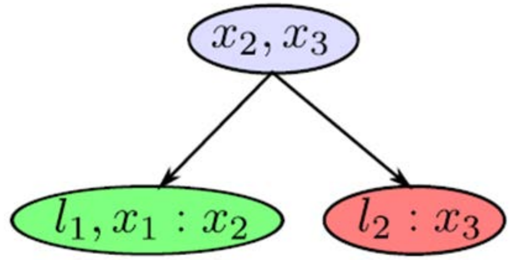
Why? Inference in tree structure is easy!

Two stage process:

1. Variable elimination converts factor graph to Bayes net



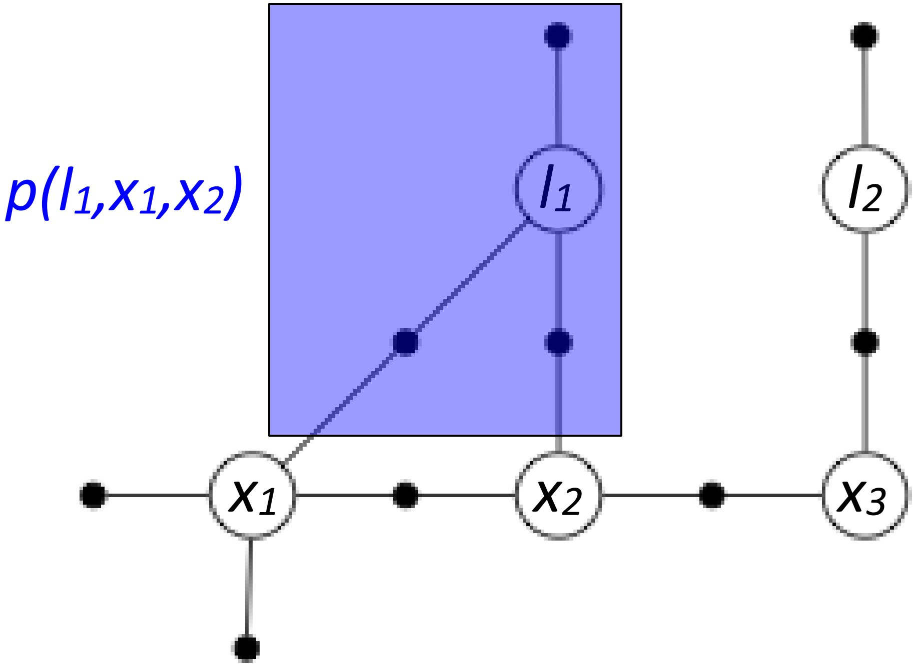
2. Discovering cliques provides Bayes tree

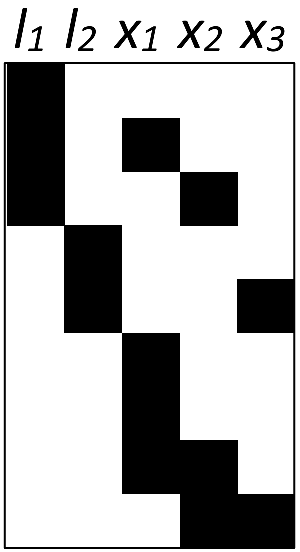


"iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree" M. Kaess et al., IJRR 2012

Variable Elimination – Example

- Choose ordering: l_1, l_2, x_1, x_2, x_3
- Eliminate one node at a time

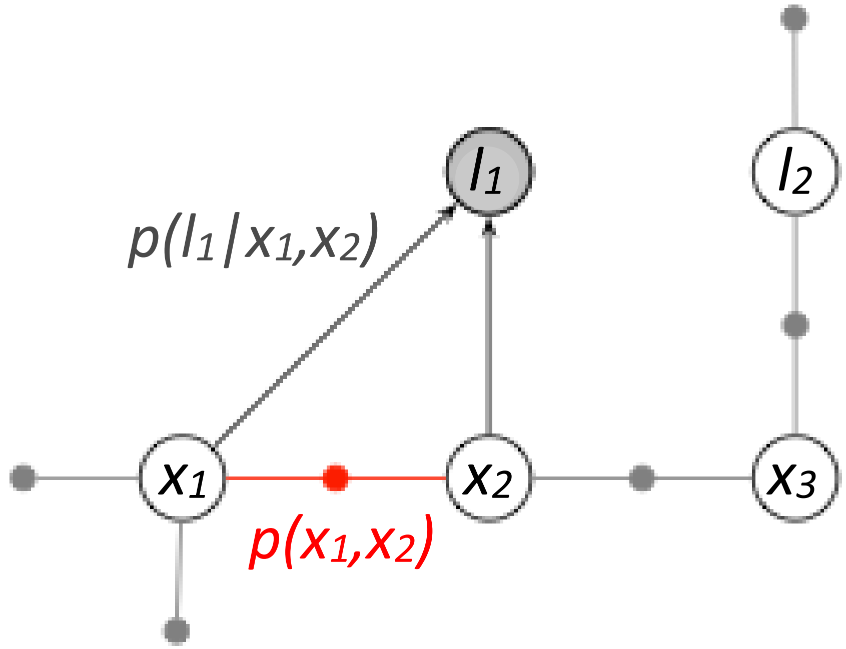


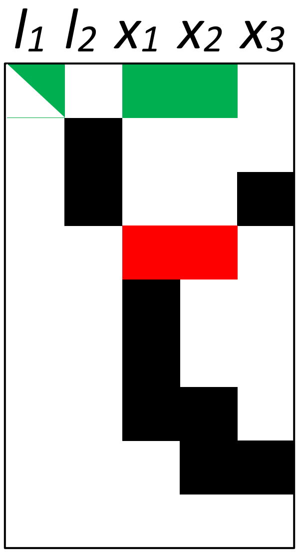


$$p(l_1,x_1,x_2) = p(l_1|x_1,x_2)\, p(x_1,x_2)$$

Variable Elimination – Example

- Choose ordering: l_1, l_2, x_1, x_2, x_3
- Eliminate one node at a time

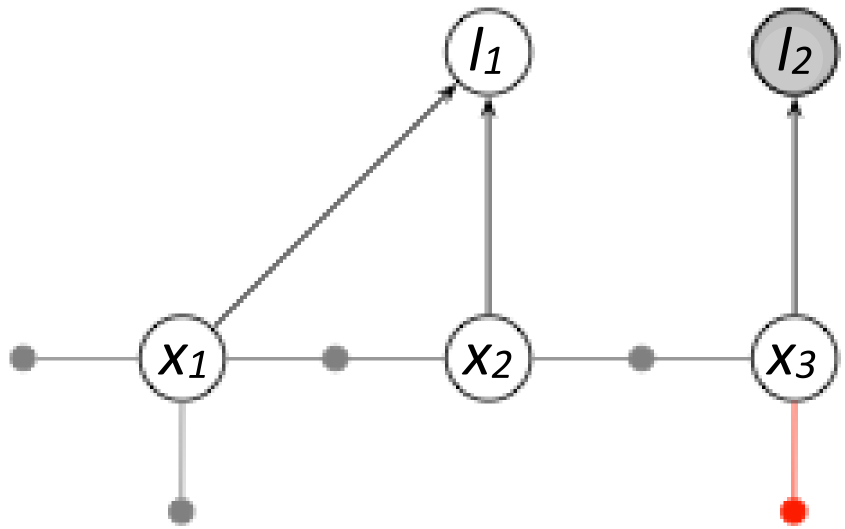


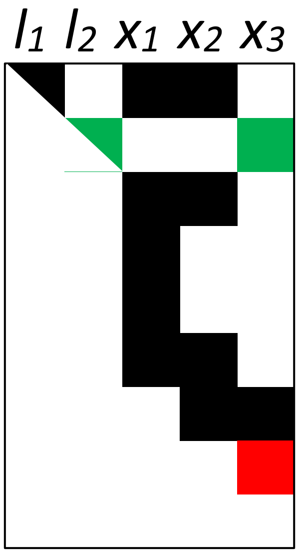


$p(l_1,x_1,x_2) = p(l_1|x_1,x_2)\ p(x_1,x_2)$

Variable Elimination – Example

- Choose ordering: l_1, l_2, x_1, x_2, x_3
- Eliminate one node at a time



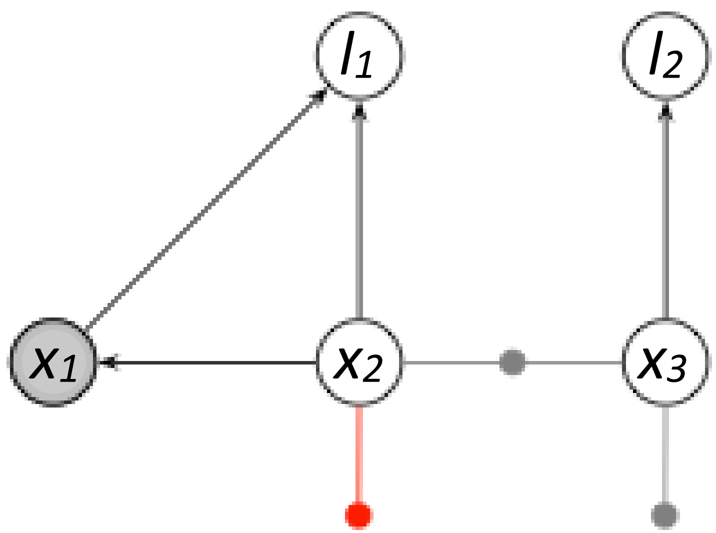


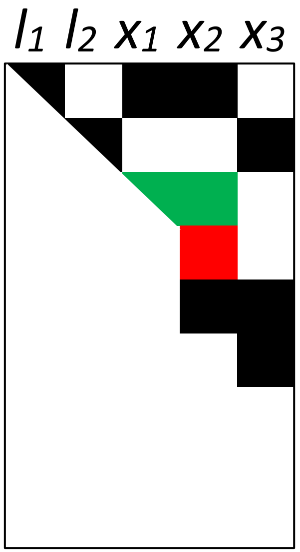
$p(l_2, x_3) = p(l_2 | x_3)\, p(x_3)$

[Kaess et al., IJRR 12]

Variable Elimination – Example

- Choose ordering: l_1, l_2, x_1, x_2, x_3
- Eliminate one node at a time

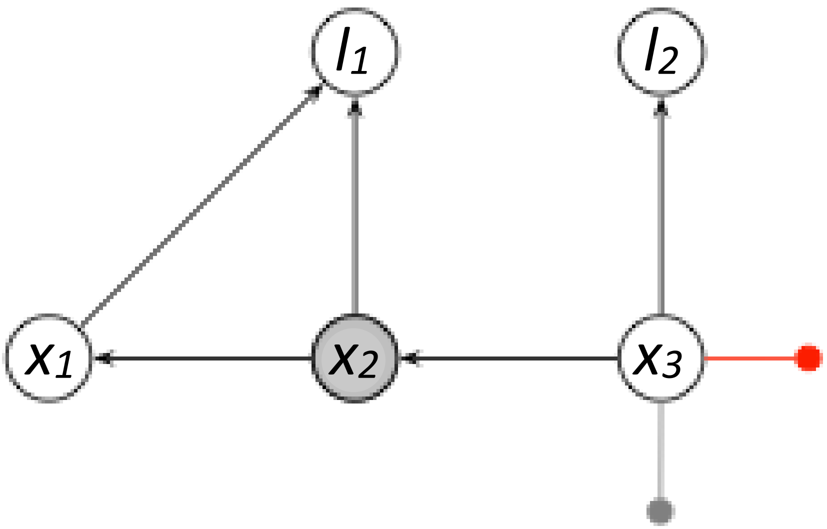


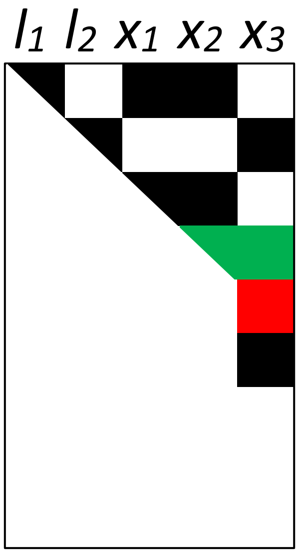


$p(x_1,x_2) = p(x_1|x_2)\, p(x_2)$

Variable Elimination – Example

- Choose ordering: l_1, l_2, x_1, x_2, x_3
- Eliminate one node at a time

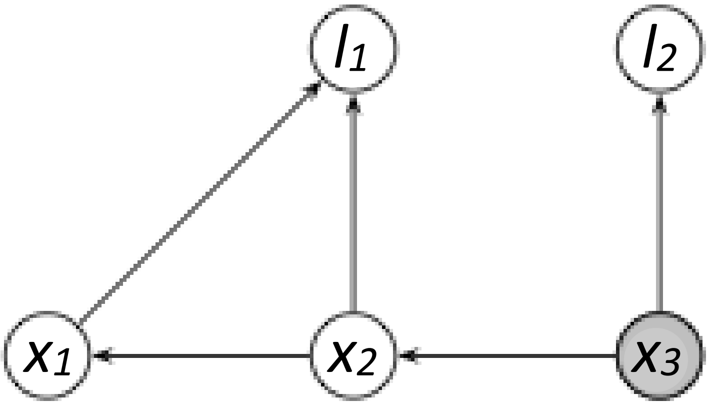


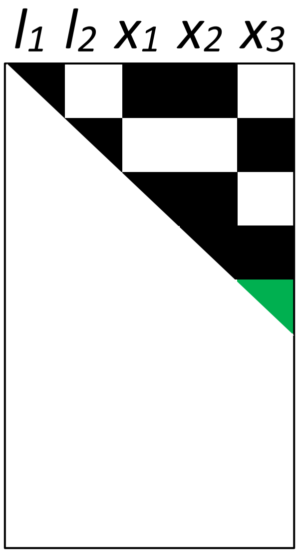


$p(x_2,x_3) = p(x_2|x_3)\, p(x_3)$

Variable Elimination – Example

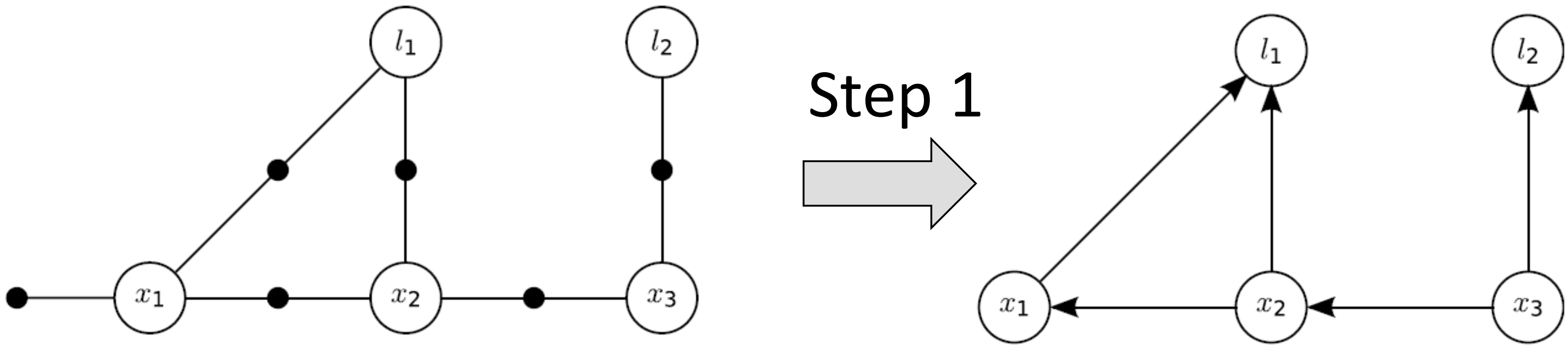
- Choose ordering: l_1, l_2, x_1, x_2, x_3
- Eliminate one node at a time





$p(x_3)$

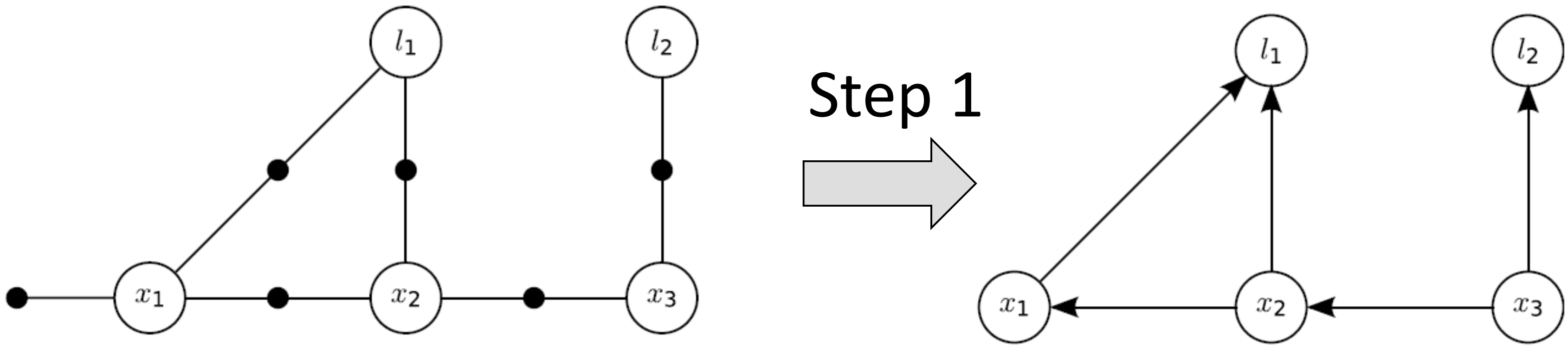
Bayes Tree Data Structure



The Bayes net has a special property: its undirected equivalent is chordal by construction

Chordal: No cycle greater than 3 that has no shortcut

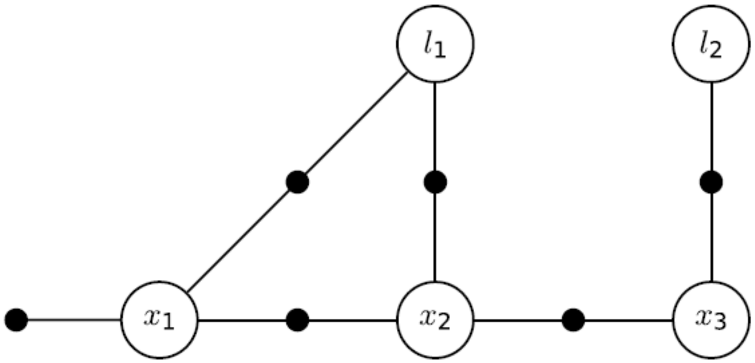
Bayes Tree Data Structure



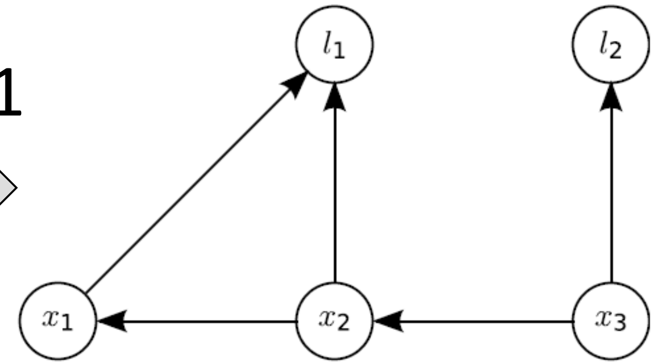
Step 2: Find cliques in reverse elimination order:

[Kaess et al., IJRR 12]

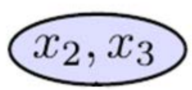
Bayes Tree Data Structure



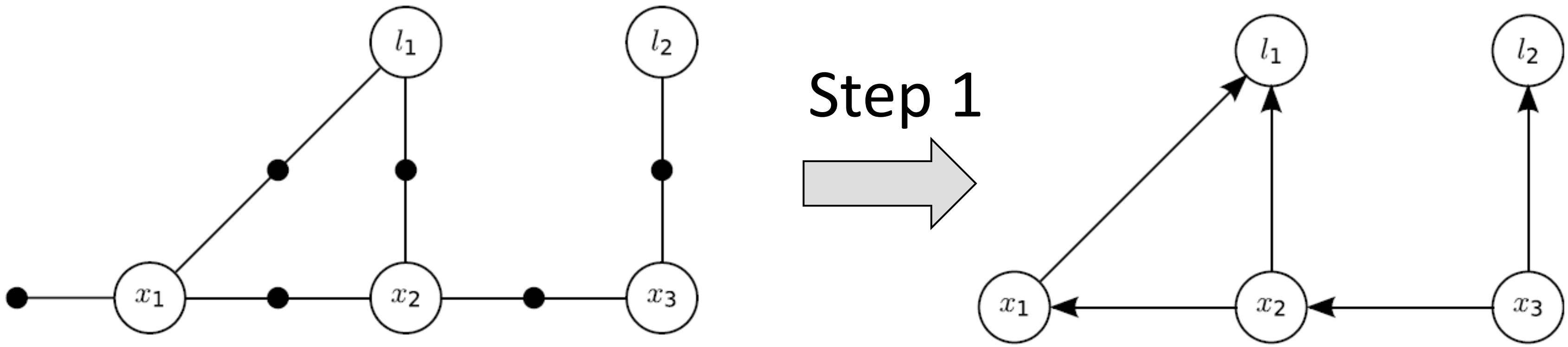
Step 1



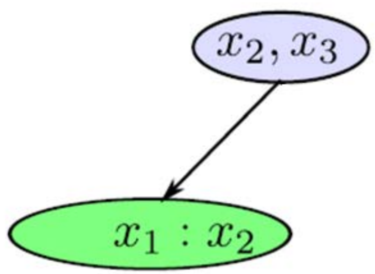
Step 2: Find cliques in reverse elimination order:



Bayes Tree Data Structure

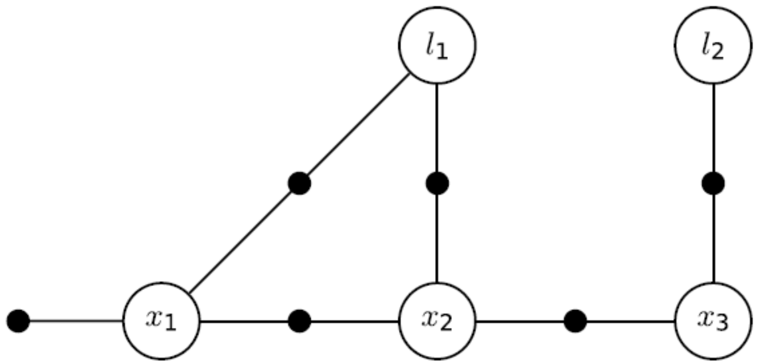


Step 2: Find cliques in reverse elimination order:

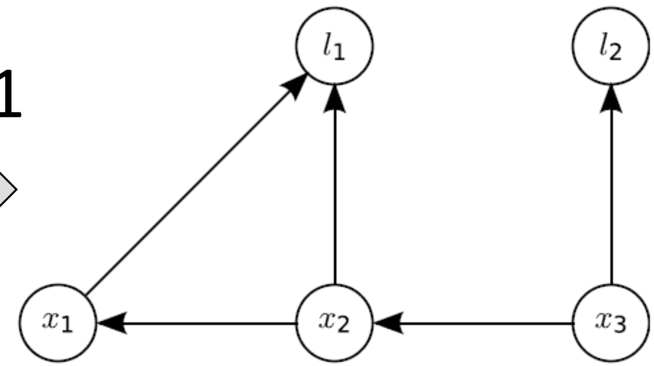


[Kaess et al., IJRR 12]

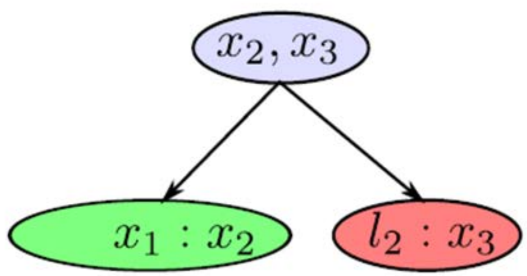
Bayes Tree Data Structure



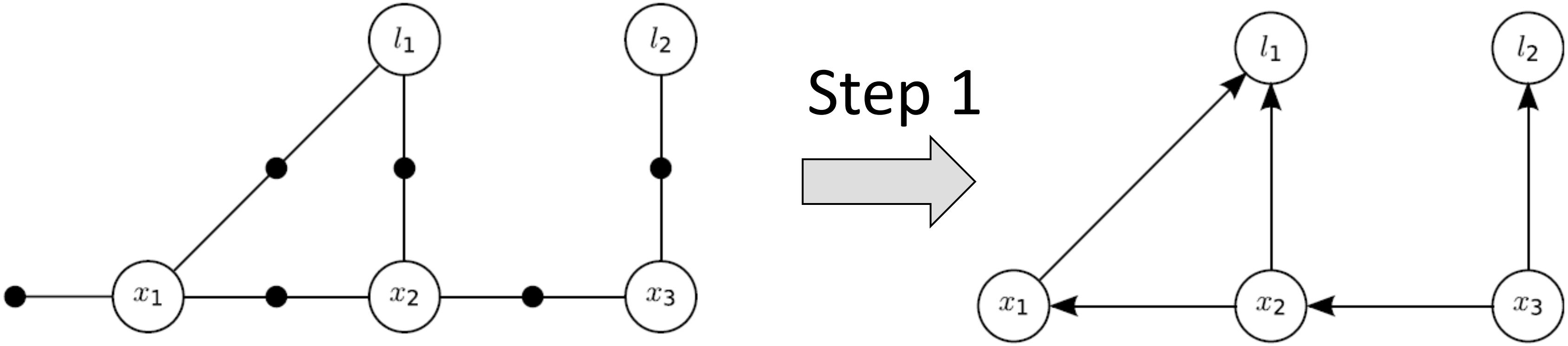
Step 1



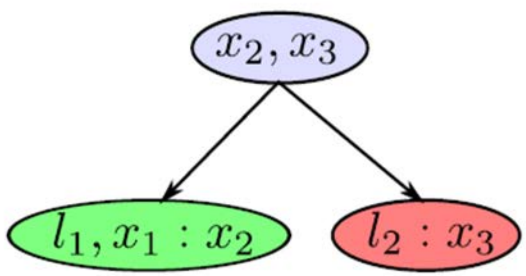
Step 2: Find cliques in reverse elimination order:



Bayes Tree Data Structure

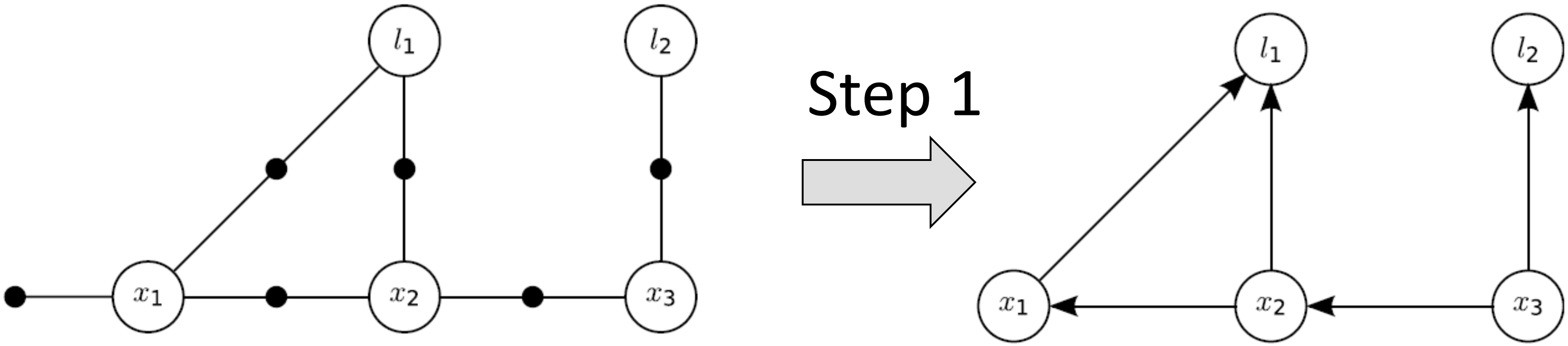


Step 2: Find cliques in reverse elimination order:

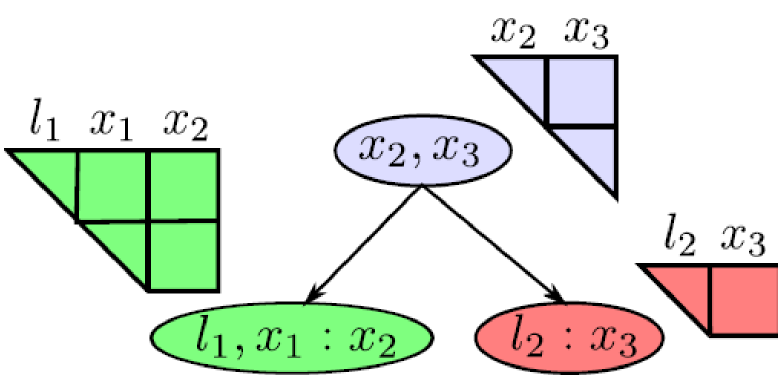


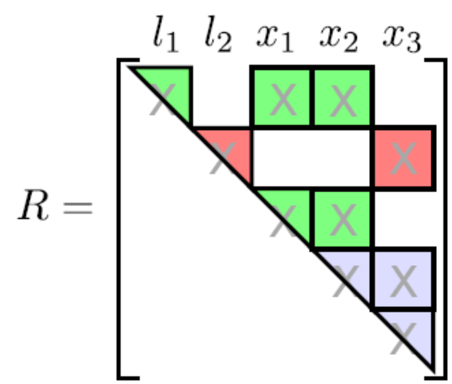
[Kaess et al., IJRR 12]

Bayes Tree Data Structure



Step 2: Find cliques in reverse elimination order:

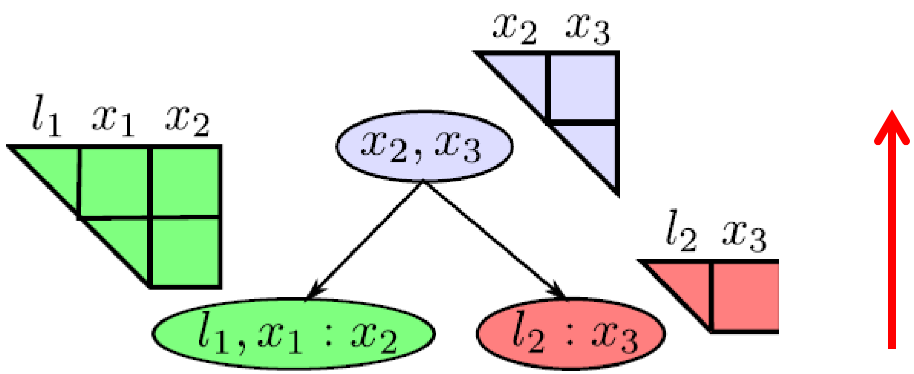




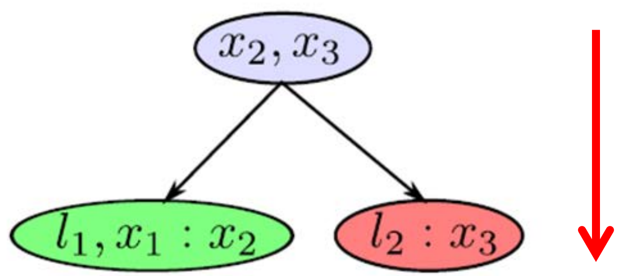
$$P(x_j|S_j) \propto \exp\left\{-\frac{1}{2\sigma^2}(x_j + rS_j - d)^2\right\}$$

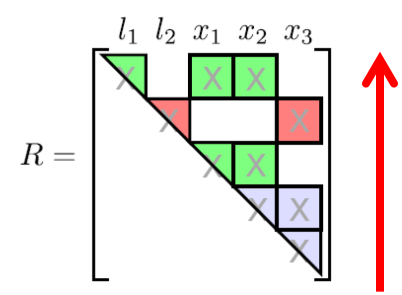
Backsubstitution in the Graph

- Inference is a two step process:
  - Elimination starts at leaves and proceeds to the root



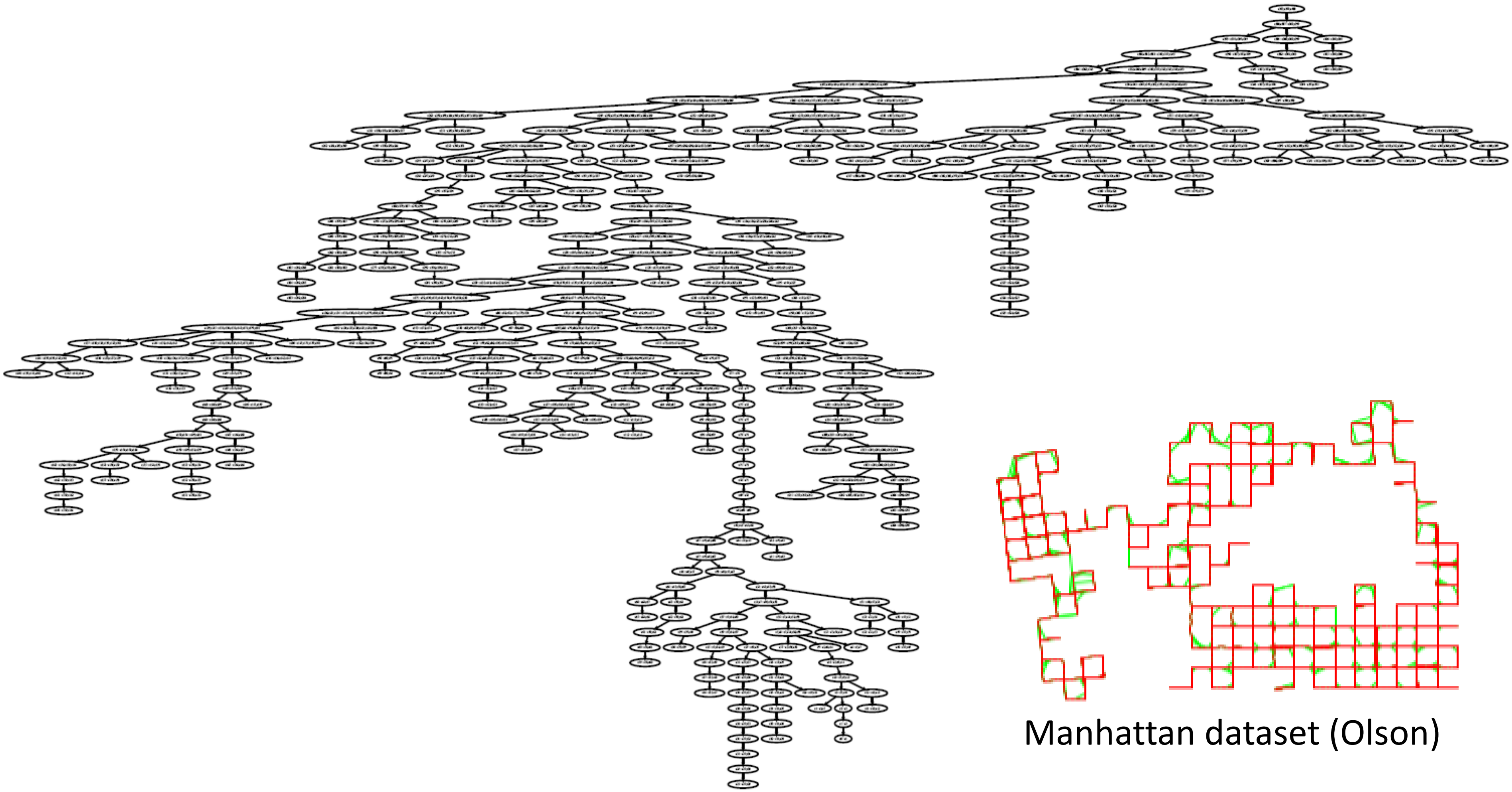
  - Solving starts at root and proceeds to the leaves





[Kaess et al., IJRR 12]

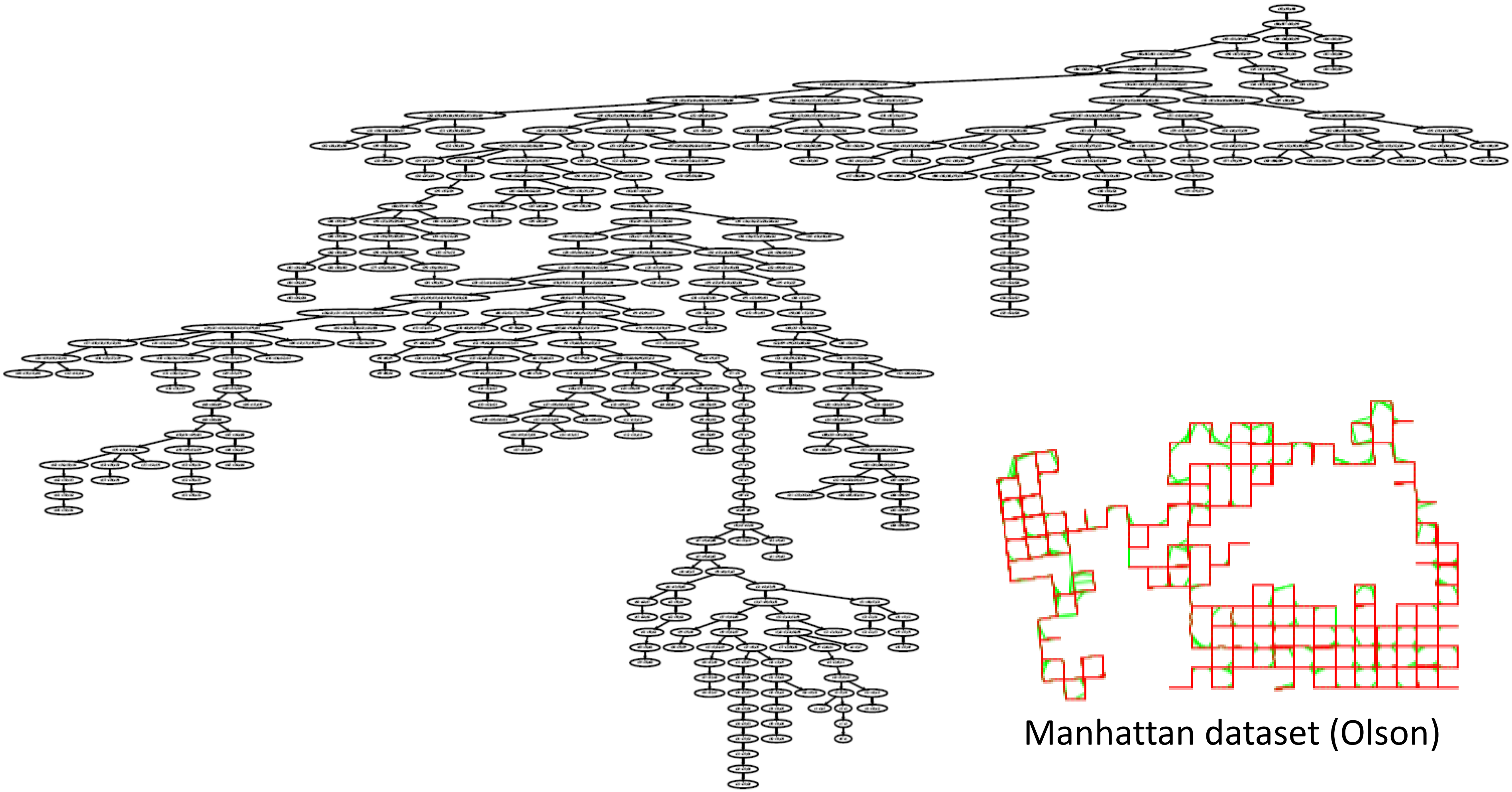
iSAM2: Bayes Tree Example



Manhattan dataset (Olson)

Complexity depends on the size of the largest clique

iSAM2: Bayes Tree Example

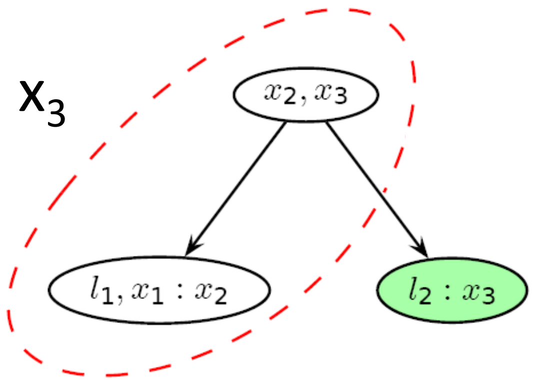


Manhattan dataset (Olson)

How to update with new measurements / add variables?

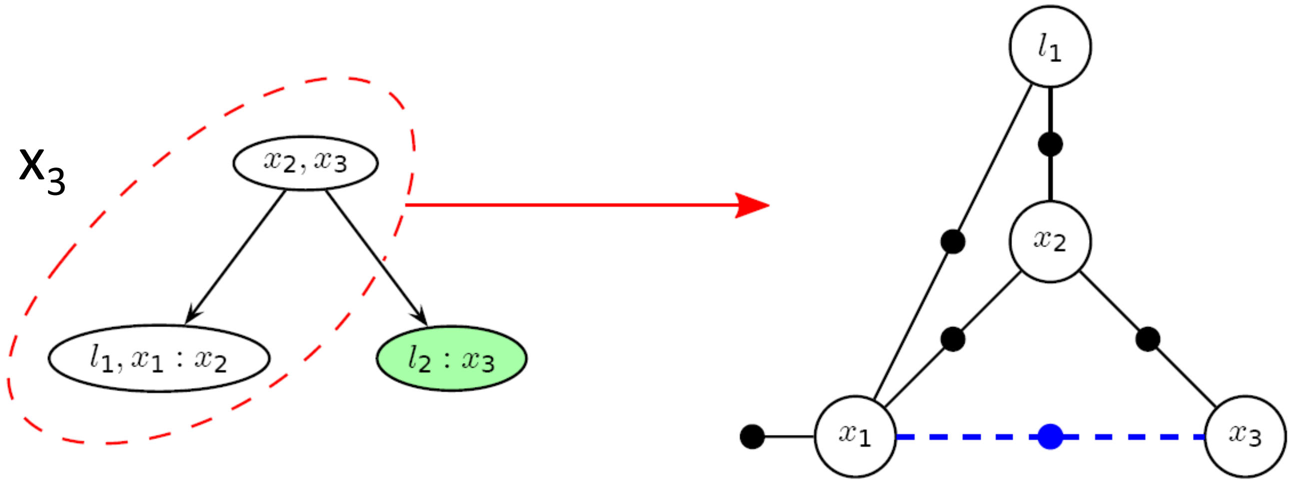
iSAM2: Updating the Bayes Tree

Add new factor between x_1 and x_3



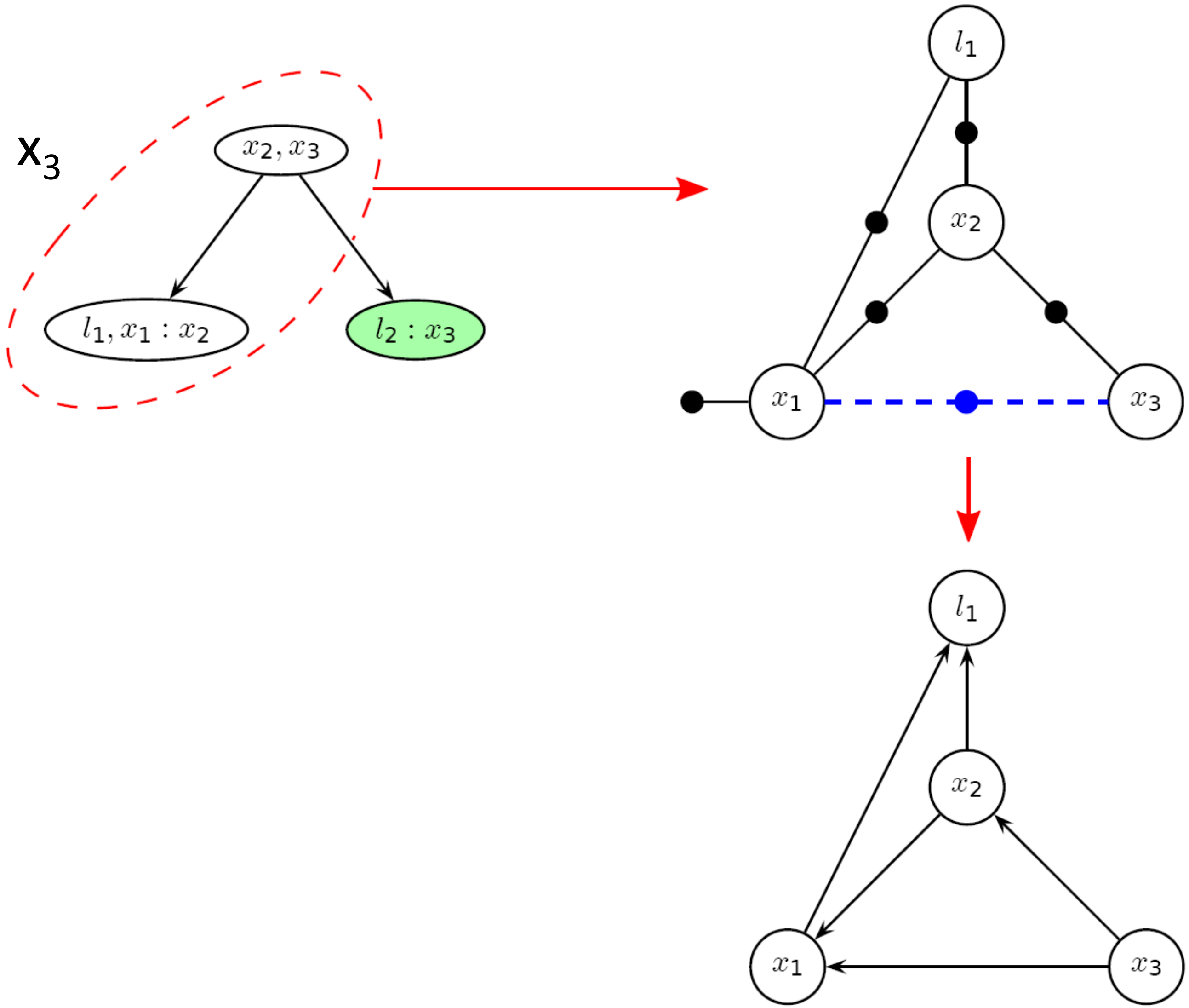
iSAM2: Updating the Bayes Tree

Add new factor between x_1 and x_3



iSAM2: Updating the Bayes Tree

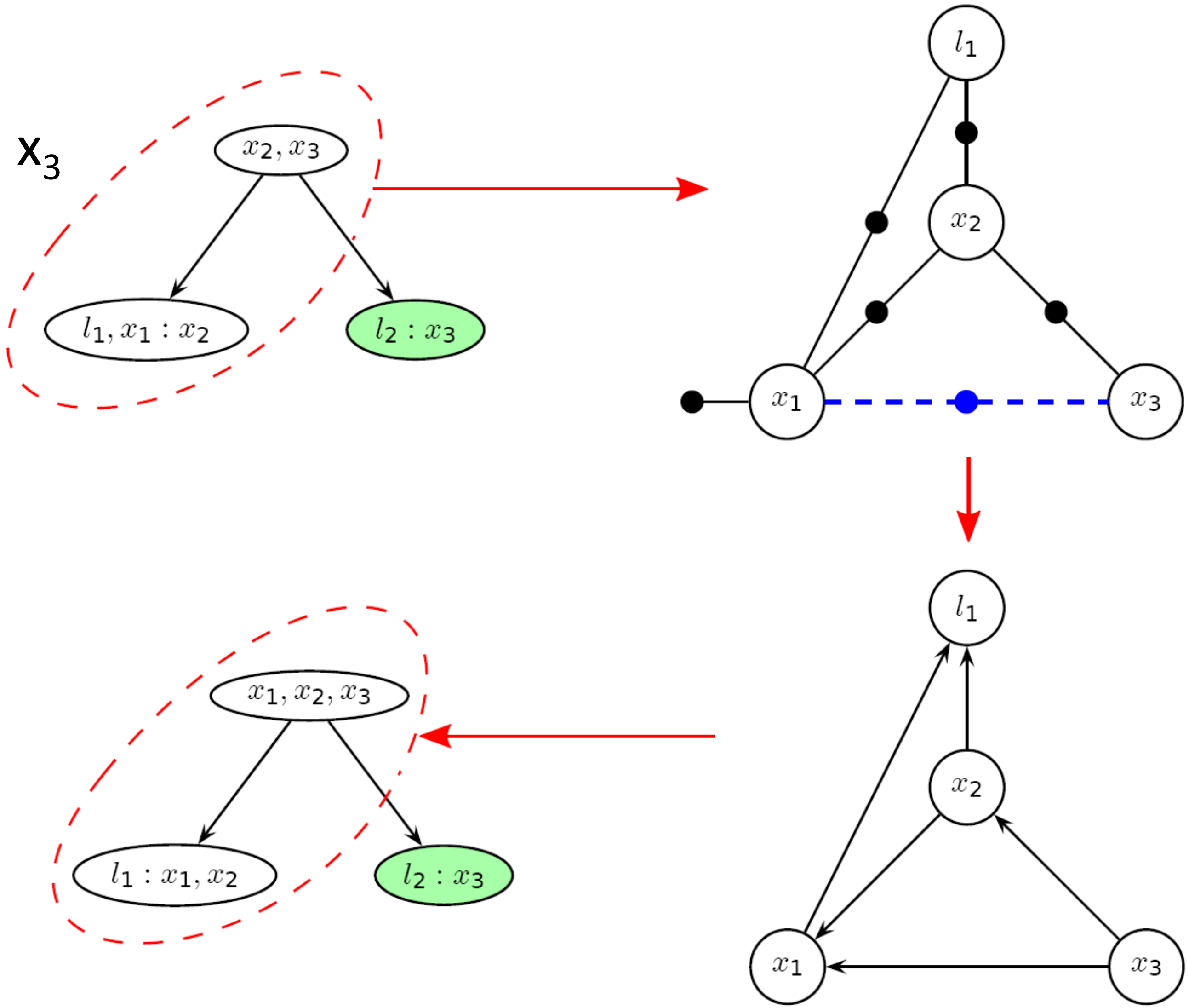
Add new factor between x_1 and x_3

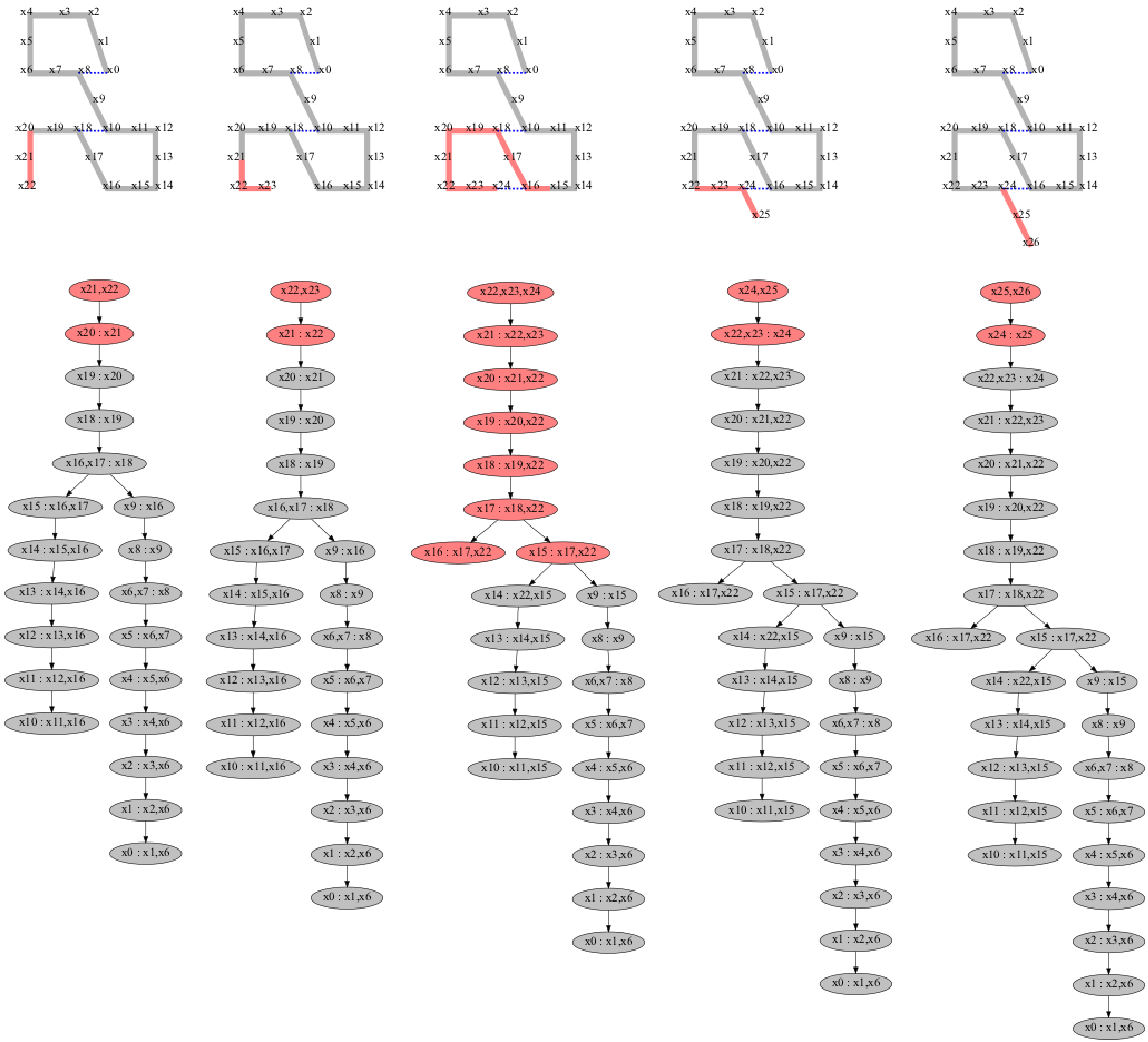


[Kaess et al., IJRR 12]

iSAM2: Updating the Bayes Tree

Add new factor between x_1 and x_3




x4 x3 x2
x5 x1
x6 x7 x8 x0
x9
x20 x19 x18 x10 x11 x12
x21 x17 x13
x22 x16 x15 x14
x21,x22
x20 : x21
x19 : x20
x18 : x19
x16,x17 : x18
x15 : x16,x17
x9 : x16
x14 : x15,x16
x8 : x9
x13 : x14,x16
x6,x7 : x8
x12 : x13,x16
x5 : x6,x7
x11 : x12,x16
x4 : x5,x6
x10 : x11,x16
x3 : x4,x6
x2 : x3,x6
x1 : x2,x6
x0 : x1,x6
x4 x3 x2
x5 x1
x6 x7 x8 x0
x9
x20 x19 x18 x10 x11 x12
x21 x17 x13
x22 x23 x16 x15 x14
x22,x23
x21 : x22
x20 : x21
x19 : x20
x18 : x19
x16,x17 : x18
x15 : x16,x17
x9 : x16
x14 : x15,x16
x8 : x9
x13 : x14,x16
x6,x7 : x8
x12 : x13,x16
x5 : x6,x7
x11 : x12,x16
x4 : x5,x6
x10 : x11,x16
x3 : x4,x6
x2 : x3,x6
x1 : x2,x6
x0 : x1,x6
x4 x3 x2
x5 x1
x6 x7 x8 x0
x9
x20 x19 x18 x10 x11 x12
x21 x17 x13
x22 x23 x24 x16 x15 x14
x22,x23,x24
x21 : x22,x23
x20 : x21,x22
x19 : x20,x22
x18 : x19,x22
x17 : x18,x22
x16 : x17,x22
x15 : x17,x22
x14 : x22,x15
x9 : x15
x13 : x14,x15
x8 : x9
x12 : x13,x15
x6,x7 : x8
x11 : x12,x15
x5 : x6,x7
x10 : x11,x15
x4 : x5,x6
x3 : x4,x6
x2 : x3,x6
x1 : x2,x6
x0 : x1,x6
x4 x3 x2
x5 x1
x6 x7 x8 x0
x9
x20 x19 x18 x10 x11 x12
x21 x17 x13
x22 x23 x24 x16 x15 x14
x25
x24,x25
x22,x23 : x24
x21 : x22,x23
x20 : x21,x22
x19 : x20,x22
x18 : x19,x22
x17 : x18,x22
x16 : x17,x22
x15 : x17,x22
x14 : x22,x15
x9 : x15
x13 : x14,x15
x8 : x9
x12 : x13,x15
x6,x7 : x8
x11 : x12,x15
x5 : x6,x7
x10 : x11,x15
x4 : x5,x6
x3 : x4,x6
x2 : x3,x6
x1 : x2,x6
x0 : x1,x6
x4 x3 x2
x5 x1
x6 x7 x8 x0
x9
x20 x19 x18 x10 x11 x12
x21 x17 x13
x22 x23 x24 x16 x15 x14
x25
x26
x25,x26
x24 : x25
x22,x23 : x24
x21 : x22,x23
x20 : x21,x22
x19 : x20,x22
x18 : x19,x22
x17 : x18,x22
x16 : x17,x22
x15 : x17,x22
x14 : x22,x15
x9 : x15
x13 : x14,x15
x8 : x9
x12 : x13,x15
x6,x7 : x8
x11 : x12,x15
x5 : x6,x7
x10 : x11,x15
x4 : x5,x6
x3 : x4,x6
x2 : x3,x6
x1 : x2,x6
x0 : x1,x6

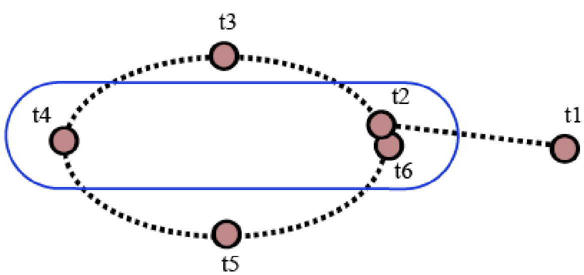
[Kaess et al., IJRR 12]

Incremental Variable Reordering

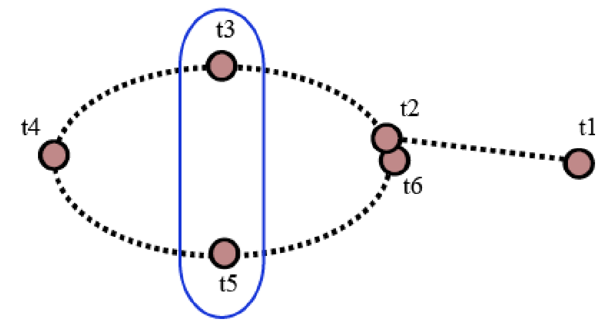
For a small loop, what constitutes a "good" ordering?

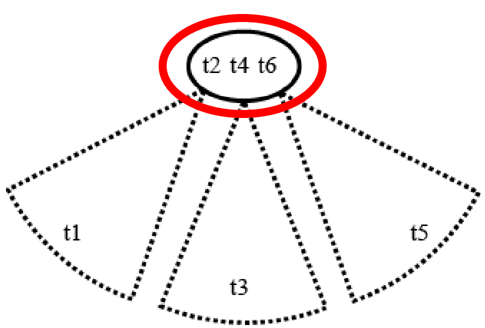
Include loop closing into cut

Loop closing not part of cut



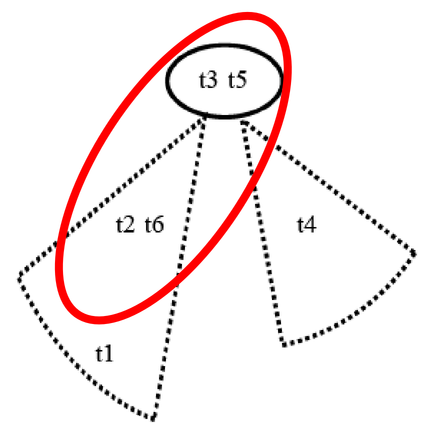
Trajectory





Affected by next update

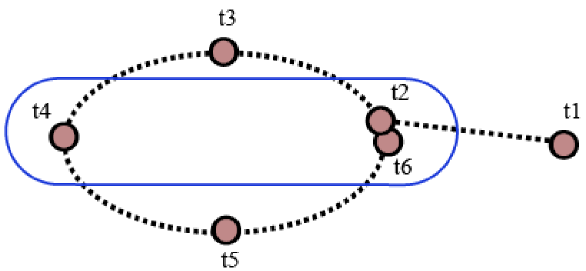
Bayes tree

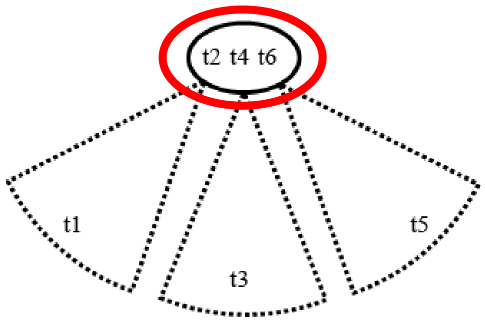


Incremental Variable Reordering

Most recent variable at the end

expected to make future updates cheaper





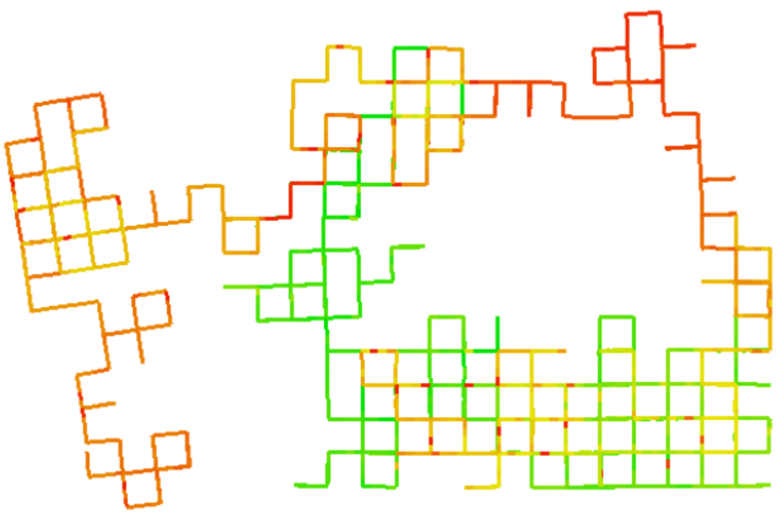
- Force most recent variables to the end
- Find best ordering for remaining variables

Using constrained version of COLAMD algorithm (CCOLAMD)

Variable Reordering – Constrained COLAMD

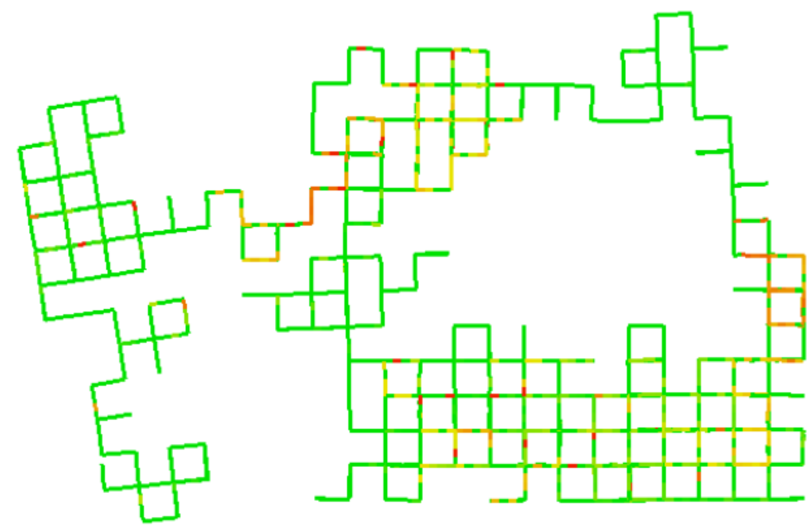
Greedy approach

Arbitrary placement of newest variable



Constrained Ordering

Newest variables forced to the end



Number of affected variables:

low high

Much cheaper!

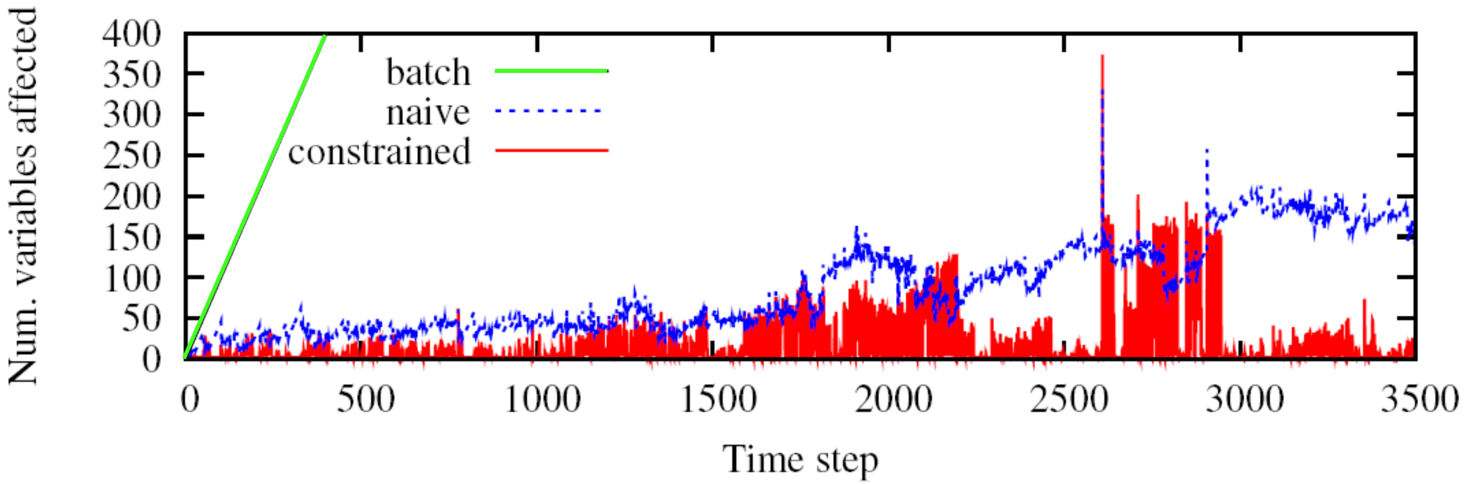
[Kaess et al., IJRR 12]

iSAM2: Incremental Update + Variable Ordering

Variable ordering changes incrementally during update

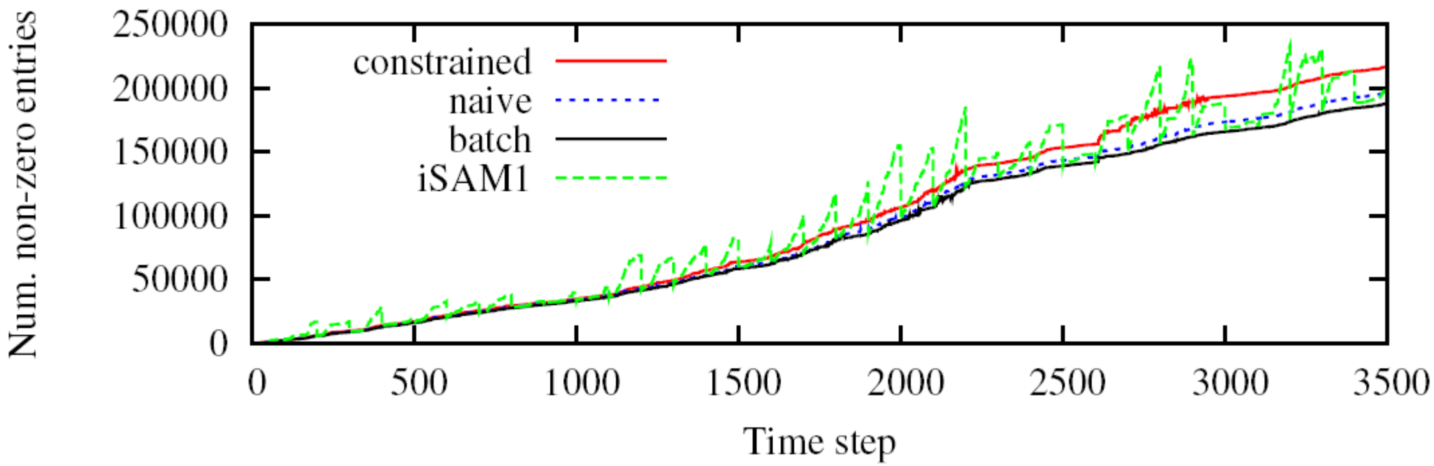
- Not understood in matrix version
- Sparse matrix data structure not suitable

Large savings in computation



Variable Reordering – Fill-in

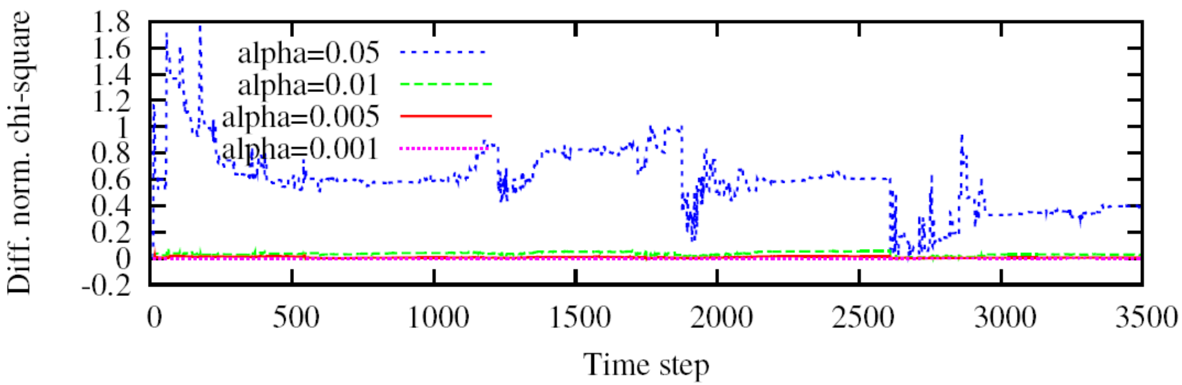
Incremental ordering still yields good overall ordering

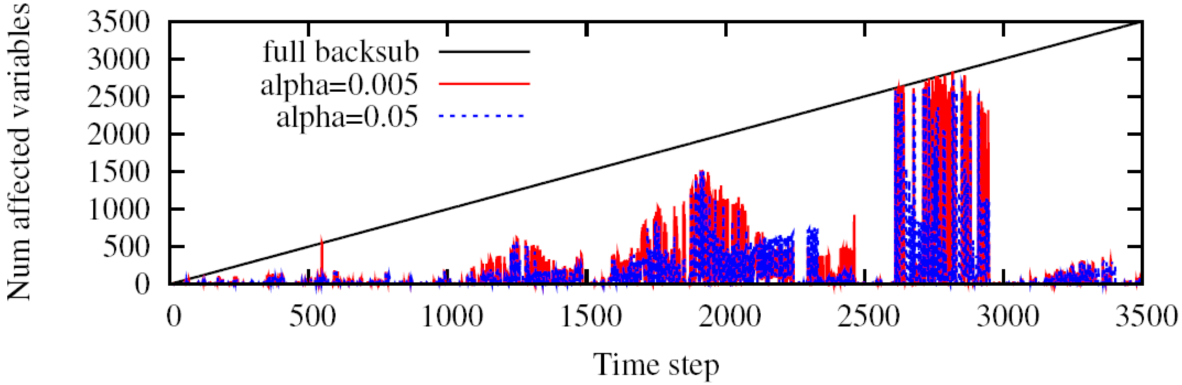


- Only slightly more fill-in than batch COLAMD ordering
- Constrained ordering is worse than naïve/greedy:
  - Suboptimal ordering because of partial constraint, but cheaper to update!

iSAM2: Recovering Only Variables That Change

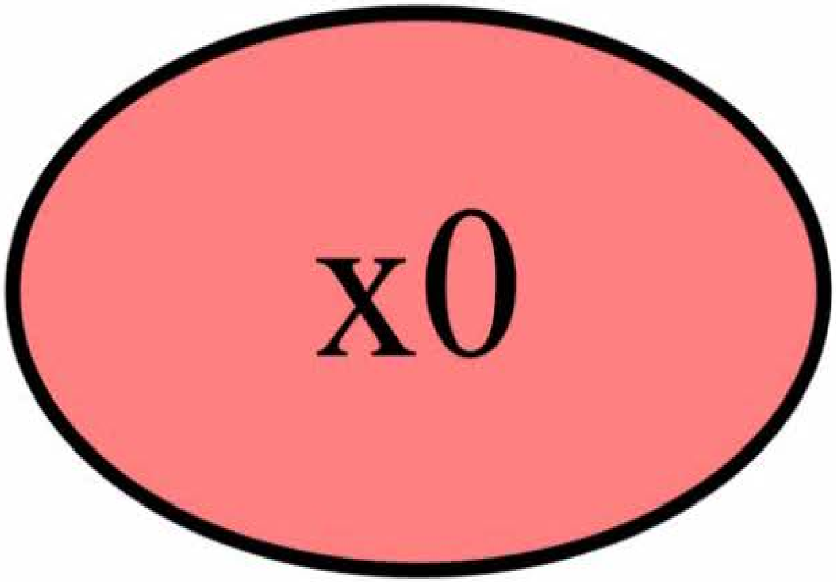
Again good quality and low cost are achievable:





[Kaess et al., IJRR 12]

iSAM2: Bayes Tree for Manhattan Sequence



Conclusion

- Exploit temporal structure
- Efficient incremental nonlinear least-squares solution
- Requirements:
  - Sparse graph
  - Good initial estimates