ECE 6504: Advanced Topics in Machine Learning

Probabilistic Graphical Models and Large-Scale Learning

Topics
- Markov Random Fields: Inference
  - Exact+Approximate: BP
  - Exact: Junction Trees

Readings: KF 10.1-10.4, Barber 5

Dhruv Batra

Virginia Tech

Administrativia

- HW1
  - Solutions & Graded copies out next week

Recap of Last Time

Variable Elimination algorithm

- Given a BN and a query $P(\mathbf{Y}|\mathbf{e}) \approx P(\mathbf{Y},\mathbf{e})$
  - "Instantiate Evidence"

IMPORTANT!!!

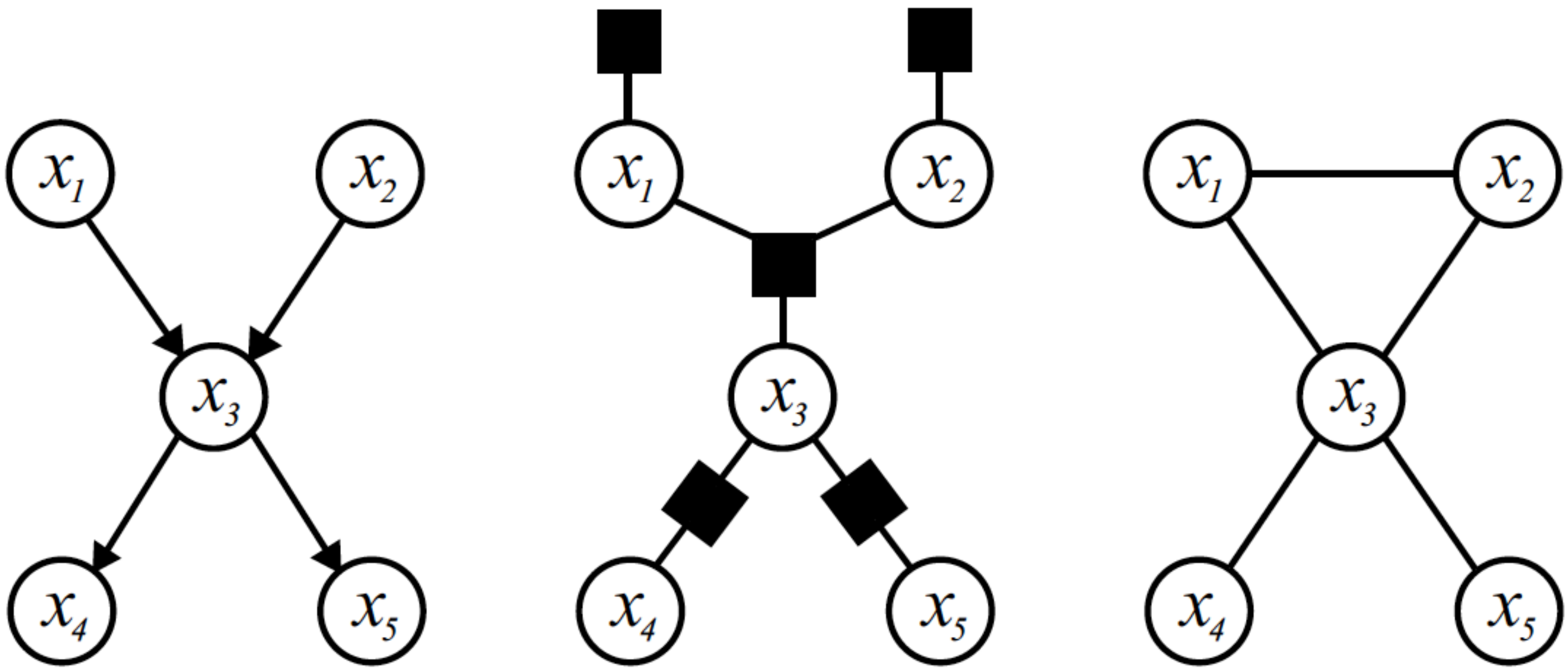
- Choose an ordering on variables, e.g., $X_1, \ldots, X_n$
- For i = 1 to n, If $X_i \notin \{\mathbf{Y},\mathbf{E}\}$
  - Collect factors $f_1,\ldots,f_k$ that include X_i
  - Generate a new factor by eliminating X_i from these factors

$$g = \sum_{X_i} \prod_{j=1}^{k} f_j$$

  - Variable X_i has been eliminated!
- Normalize P(Y,e) to obtain P(Y|e)

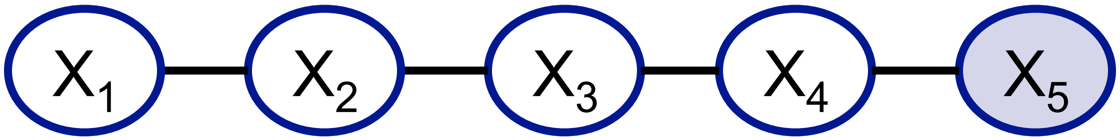
VE for MRF

- Exactly the same algorithm works!
  - Factors are no longer CPTs
  - But VE doesn't care



Example

- Chain MRF



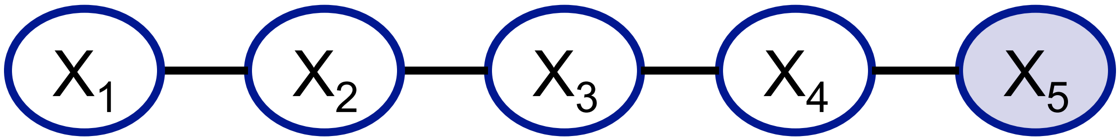
Compute:

$$P(X_1 \mid X_5 = x_5)$$

- VE steps on board

Example

- Chain MRF



Compute:

$$P(X_i \mid X_5 = x_5)$$
$$\forall i \in \{1, 2, 3, 4\}$$

Variable elimination for every i, what's the complexity?

Can we do better by caching intermediate results?

Yes! via Junction-Trees
But let's look at BP first

New Topic: Belief Propagation

Message Passing

- Variables/Factors "talk" to each other via messages:

"I (variable X_3) think that you (variable X_2):

belong to state 1 with confidence 0.4
belong to state 2 with confidence 10
belong to state 3 with confidence 1.5"

Image Credit: James Coughlan

Overview of BP

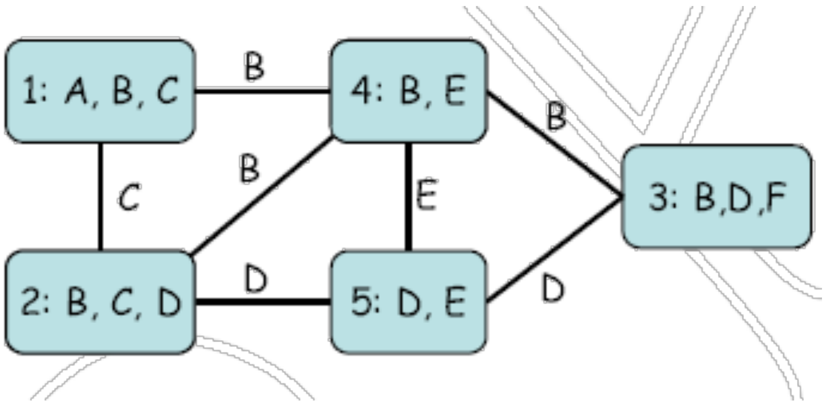
- Pick a graph to pass messages on
  - Cluster Graph

- Pick an ordering of edges
  - Round-robin
  - Leaves-Root-Leaves on a tree
  - Asynchonous

- Till convergence or exhaustion:
  - Pass messages on edges

- At vertices on graph compute *psuedo-marginals*

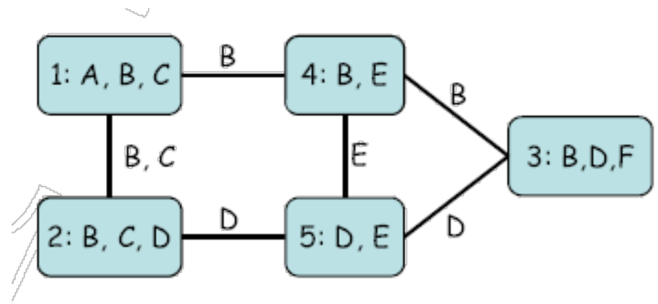
Cluster graph



- **Cluster Graph**:
  For set of factors *F*
  - Undirected graph
  - Each node i associated with a cluster $\mathbf{C}_i$
  - Each edge i – j is associated with a *separator set* of variables $\mathbf{S}_{ij} \subseteq \mathbf{C}_i \cap \mathbf{C}_j$

Generalized BP

- Initialization:
  - Assign each factor ϕ to a cluster $\alpha(\phi)$, Scope[ϕ]$\subseteq \mathbf{C}_{\alpha(\phi)}$
  - Initialize cluster: $\psi_i^0(\mathbf{C}_i) \propto \prod_{\phi:\alpha(\phi)=i} \phi$
  - Initialize messages: $\delta_{j\to i} = 1$

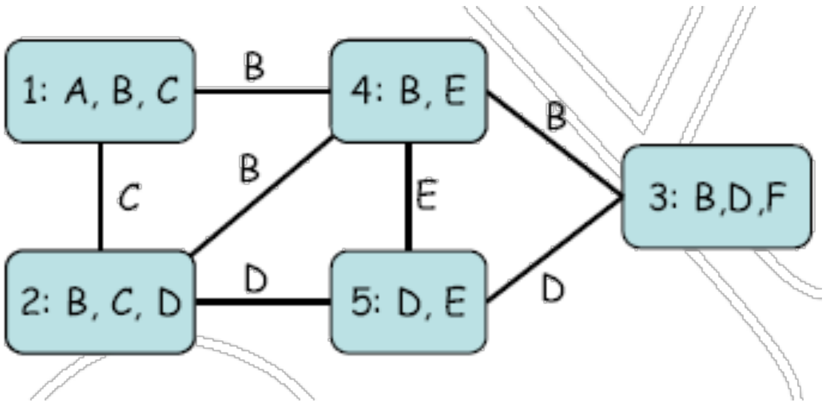


- While not converged, send messages:

$$\delta_{i\to j}(\mathbf{S}_{ij}) \propto \sum_{\mathbf{C}_i - \mathbf{S}_{ij}} \psi_i^0(\mathbf{C}_i) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(\mathbf{S}_{ik})$$

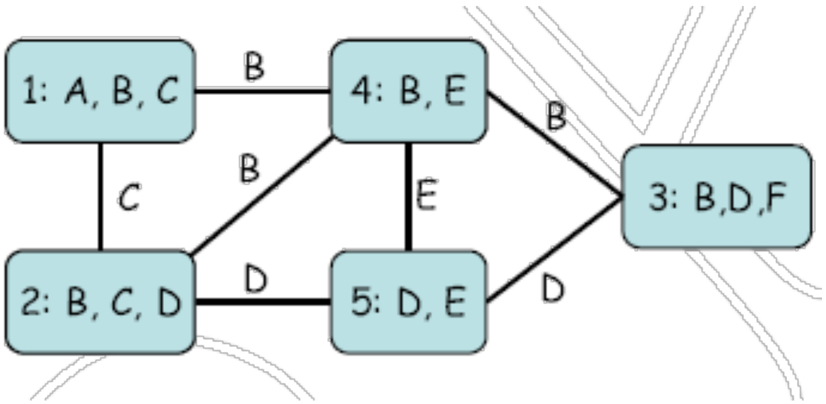
- Belief:
  - On board

Properties of Cluster Graphs



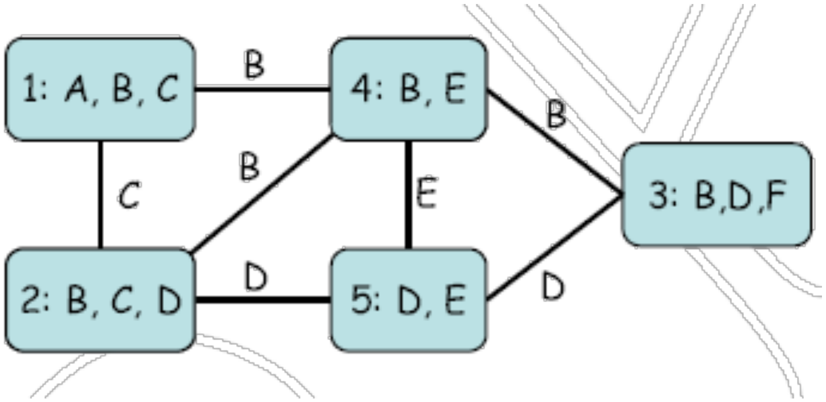
- **Family preserving**:
  For set of factors *F*
  - for each factor $f_j \in F$, $\exists$ node i such that scope[f_i] $\subseteq \mathbf{C}_i$

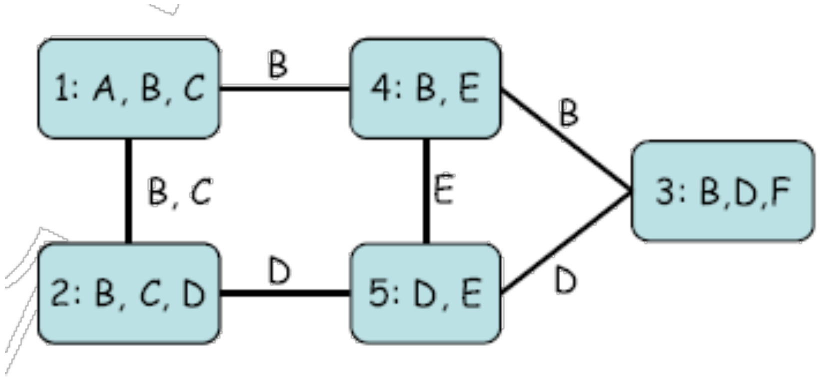
Properties of Cluster Graphs



- **Running intersection property (RIP)**
  - If $X \in \mathbf{C}_i$ and $X \in \mathbf{C}_j$ then $\exists$ *one and only one path* from $\mathbf{C}_i$ to $\mathbf{C}_j$ where $X \in \mathbf{S}_{uv}$ for every edge (u,v) in the path

Two cluster graph satisfying RIP with different edge sets





Overview of BP

- Pick a graph to pass messages on
  - Cluster Graph

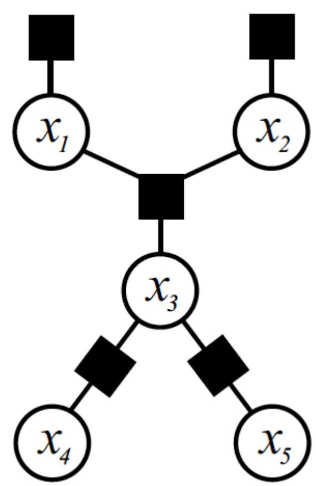
- Pick an ordering of edges
  - Round-robin
  - Leaves-Root-Leaves on a tree
  - Asynchonous

- Till convergence or exhaustion:
  - Pass messages on edges

- At vertices on graph compute *psuedo-marginals*

Cluster Graph for Loopy BP

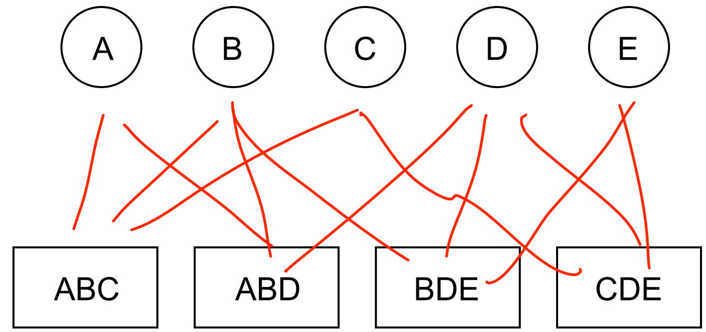
- Bethe Cluster Graph
  - Set of Clusters = Factors F $\cup$ $\{X_i\}$
  - Sometimes also called "Running BP on Factor Graphs"
  - Example on board
- Does the Bethe Cluster Graph satisfy properties?



Loopy BP in Factor graphs

- From node *i* to factor *j*:
  - *F*(i) factors whose scope includes X_i

$$\delta_{i \to j}(X_i) \propto \prod_{k \in \mathcal{F}(i) - j} \delta_{k \to i}(X_i)$$



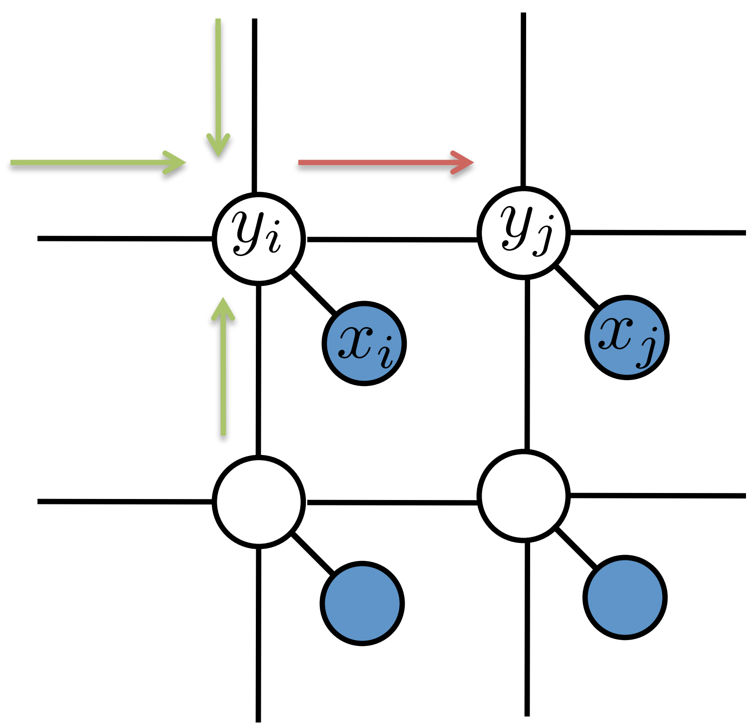
- From factor *j* to node *i*:
  - Scope[ϕ_j] = **Y** $\cup$ {X_i}

$$\delta_{j \to i}(X_i) \propto \sum_{\mathbf{y}} \phi_j(X_i, \mathbf{y}) \prod_{X_k \in \text{Scope}[\phi_j] - X_i} \delta_{k \to j}(x_k)$$

- Belief:
  - Node:
  - Factor:

Loopy BP on Pairwise Markov Nets

$$\delta_{i \to j}(y_j) = \sum_{y_i} \phi_i(y_i)\phi_{ij}(y_i, y_j) \prod_{k \in \mathcal{N}(i)-j} \delta_{k \to i}(y_i)$$

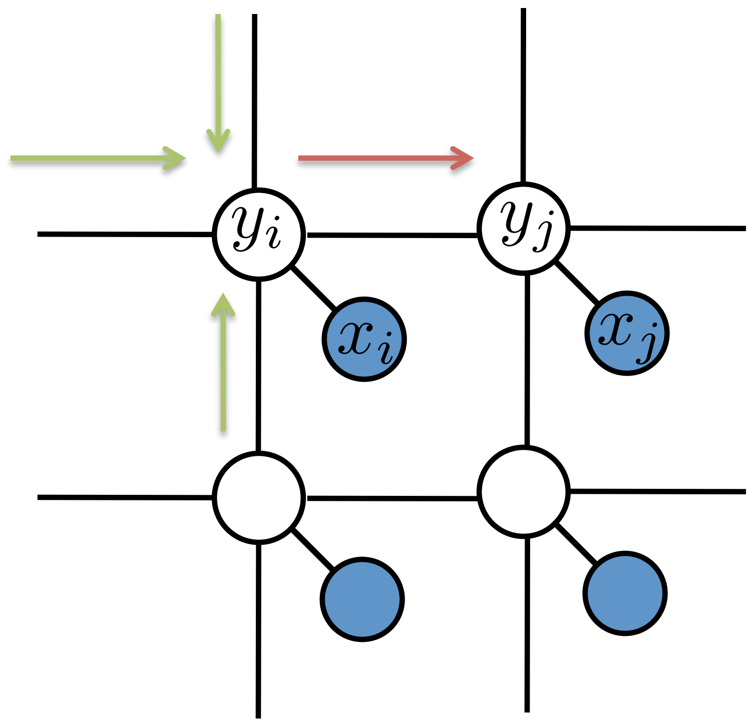


Plan for today

- MRF Inference
  - Approximate Inference
    - Bethe Cluster Graph
    - Loopy BP
  - Exact Inference
    - Junction Tree
    - BP on Junction Trees
  - Message-Passing as Variational Inference

Loopy BP on Pairwise Markov Nets

$$\overrightarrow{\delta_{i\to j}}(y_j) = \sum_{y_i} \phi_i(y_i)\phi_{ij}(y_i, y_j) \prod_{k\in\mathcal{N}(i)-j} \overrightarrow{\delta_{k\to i}}(y_i)$$



Calibration

- Cluster Graphs are *calibrated*
  - when adjacent clusters agree in beliefs about sep-sets

Convergence

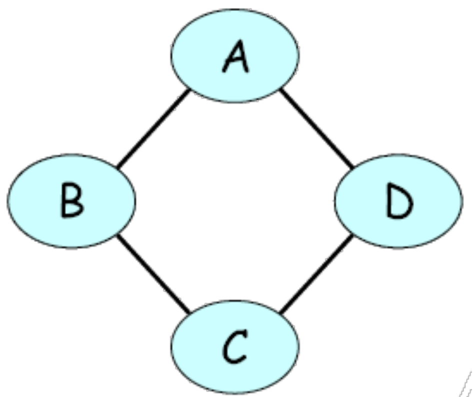
$$\delta_{i\to j}(\mathbf{S}_{ij}) \propto \sum_{\mathbf{C}_i - \mathbf{S}_{ij}} \psi_i^0(\mathbf{C}_i) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(\mathbf{S}_{ik})$$

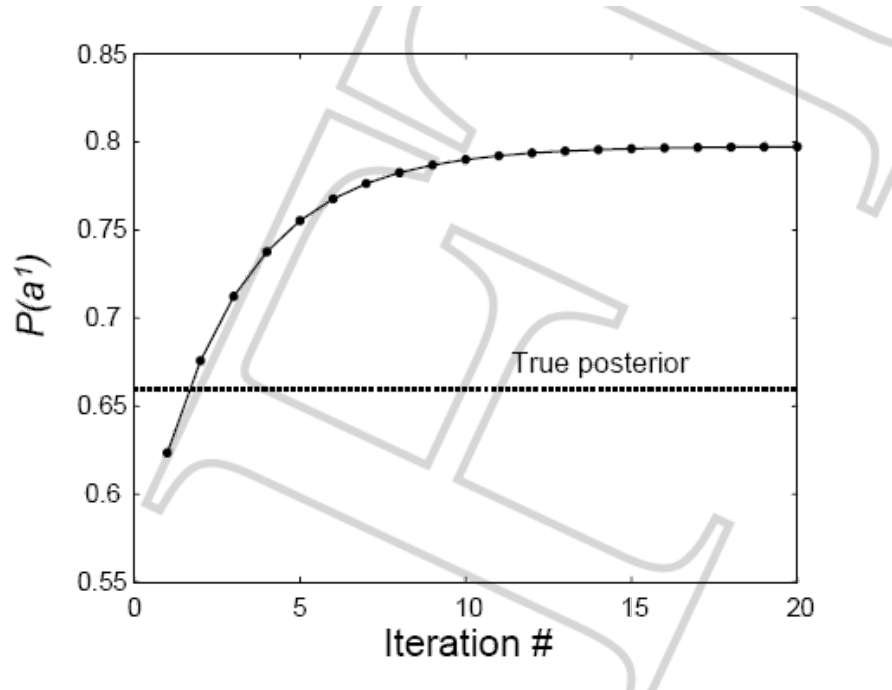
- If you tried to send all messages, and messages haven't changed (in practice by much) → converged
- Convergence of BP => Calibration of Cluster Graph
- Note, this doesn't mean pseudo-marginals are correct!

BP as Reparameterization

- On board

An example of running loopy BP





Loopy BP

$$\delta_{i\to j}(X_j) = \sum_{x_i} \phi_i(x_i)\phi_{ij}(x_i, X_j) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(x_i)$$

- What happened?
  - evidence goes around the loops multiple times
  - may not converge
  - if it converges, usually overconfident about probability values

- But often gives you reasonable, or at least useful answers
  - especially if you just care about the argmax rather than the actual probabilities

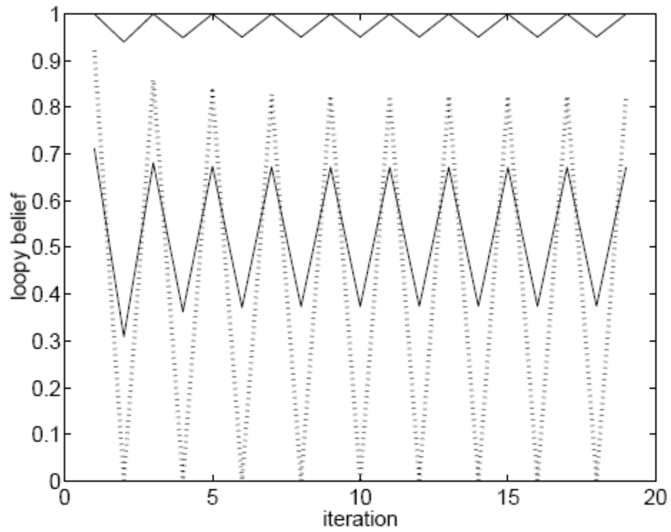
(Non-)Convergence of Loopy BP

- **Loopy BP can oscillate!!!**
  - oscillations can small
  - oscillations can be really bad!

- Typically,
  - if factors are closer to uniform, loopy does well (converges)
  - if factors are closer to deterministic, loopy doesn't behave well

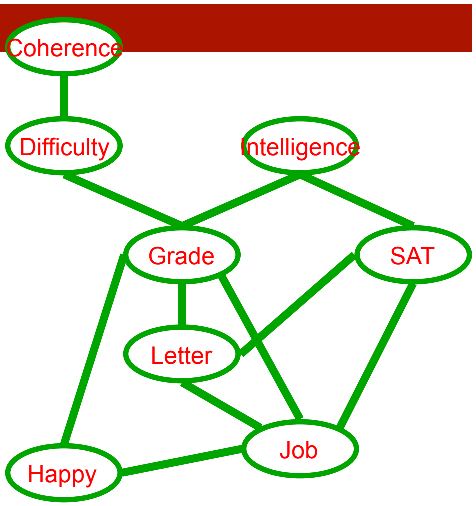
- One approach to help: damping messages
  - new message is average of old message and new one:

  - often better convergence
    - but, when damping is required to get convergence, result often bad



graph from Murphy et al. '99

Loopy BP



- Numerical problem:
  - messages < 1 get multiplied together as we go around the loops
  - numbers can go to zero
  - Work in log-space
  - normalize messages to one:

$$\delta_{i\to j}(X_j) = \frac{1}{Z_{i\to j}} \sum_{x_i} \phi_i(x_i)\phi_{ij}(x_i, X_j) \prod_{k\in\mathcal{N}(i)-j} \delta_{k\to i}(x_i)$$

  - $Z_{i\to j}$ doesn't depend on X_j, so doesn't change the answer

- Computing node pseudo-marginals (estimates of probs.):

$$\widehat{P}(X_i) = \frac{1}{Z_i}\phi_i(X_i) \prod_{k\in\mathcal{N}(i)} \delta_{k\to i}(X_i)$$

How to pass messages?

- Synchronous
  - All messages at once
  - Good for parallelization
  - Bad for convergence

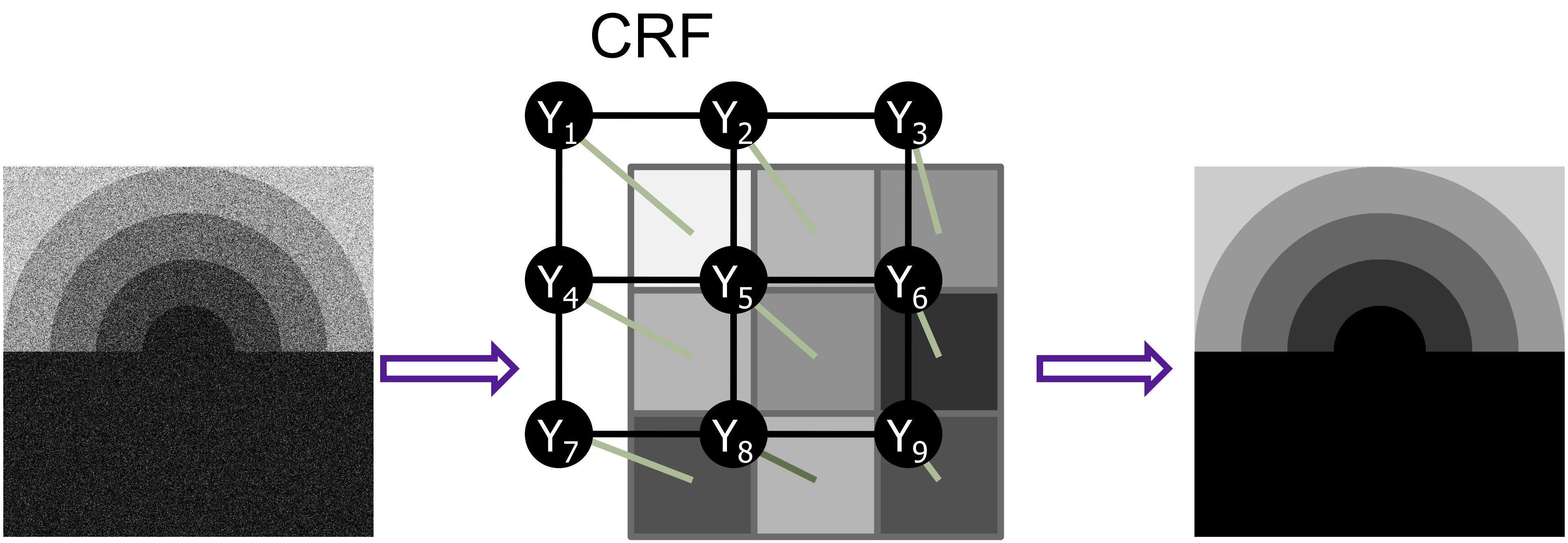
- Asynchronous
  - Sequential according to some priority
  - Bad for parallelization
  - Good for convergence

How to prioritize messages?

- Residual BP
  - e.g. [Elidan et al., 2006], [Sutton & McCallum, 2007]
  - Pass messages where cliques disagree the most about separators

Asynchronous Belief Propagation

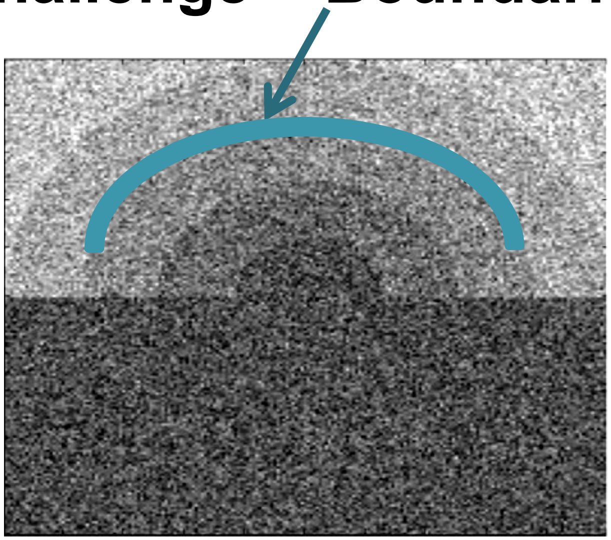
- [Gonzalez et al. AISTATS09]



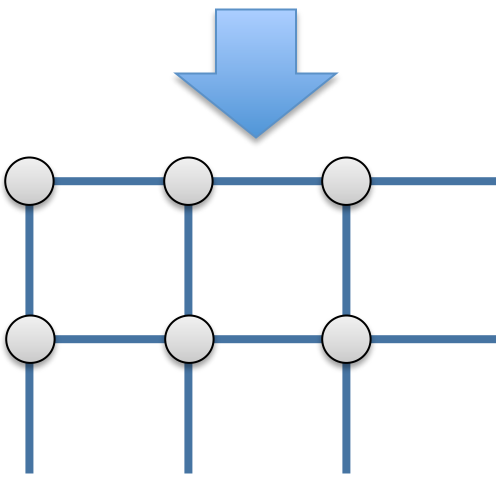
Noisy Image Pixels Predicted Image

Asynchronous Belief Propagation

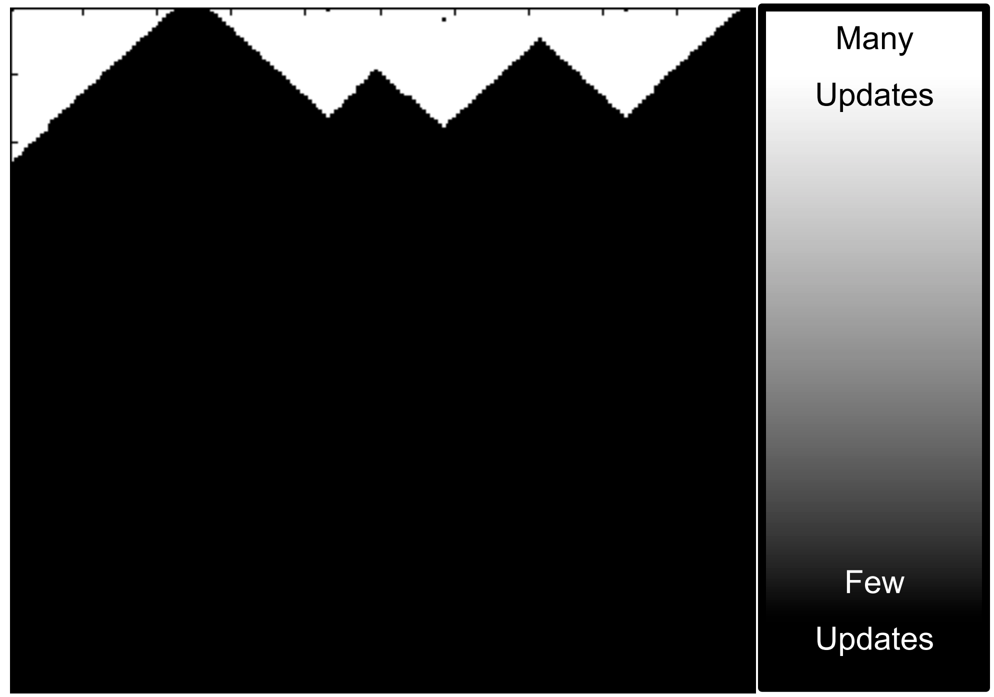
Challenge = Boundaries



Synthetic Noisy Image



Graphical Model



Cumulative Vertex Updates

Algorithm identifies and focuses on hidden sequential structure

How to prioritize messages?

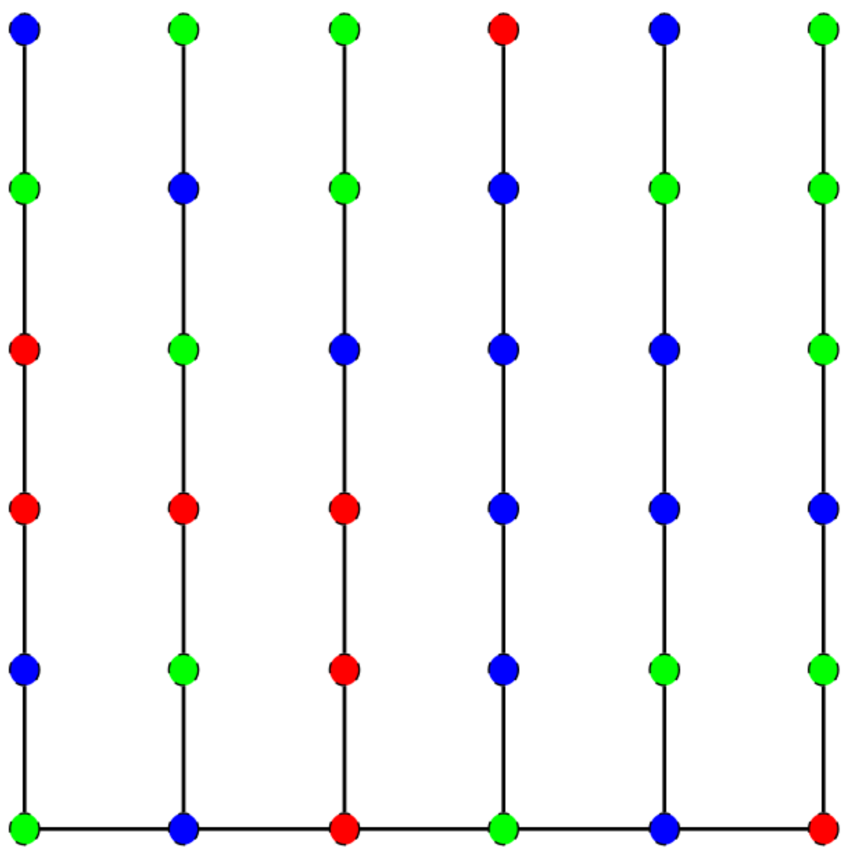
- Residual BP
  - e.g. [Elidan et al., 2006], [Sutton & McCallum, 2007]
  - Pass messages where cliques disagree the most about separators

- Tree-Based Message Passing
  - e.g. [Tarlow, Batra, Kohli, Kolmogorov, ICML11]
  - Pick a tree
  - Pass messages on it's edges
  - Pick another tree

How to prioritize messages?

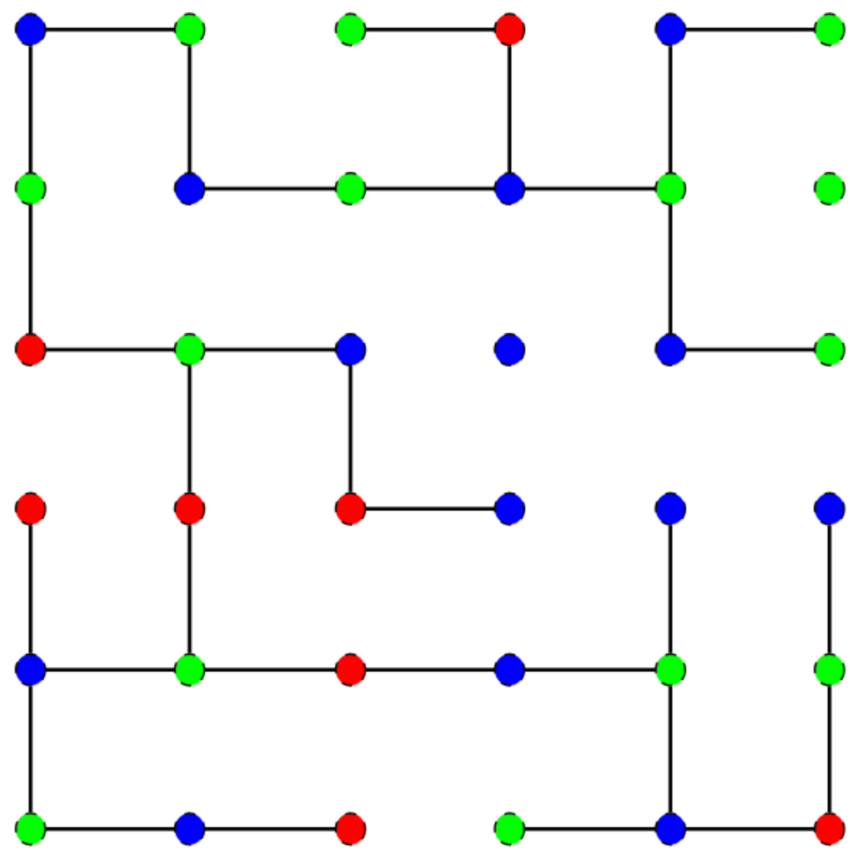
Static Schedule:

630 messages needed

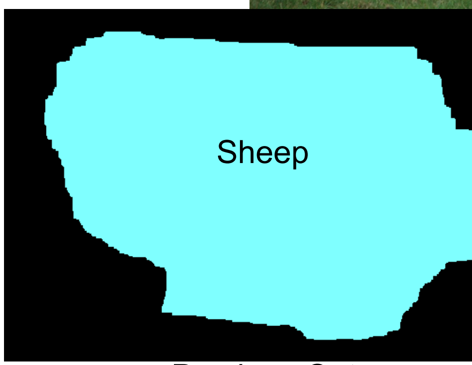


Dynamic Schedule:

276 messages needed

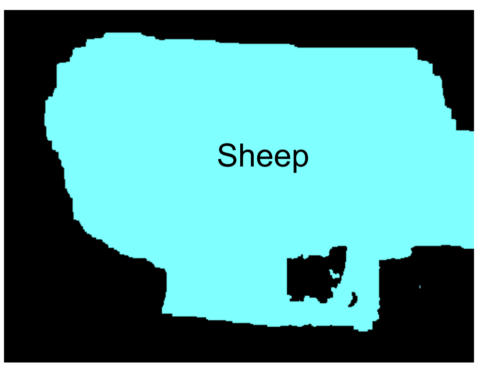


Dynamic Image Segmentation

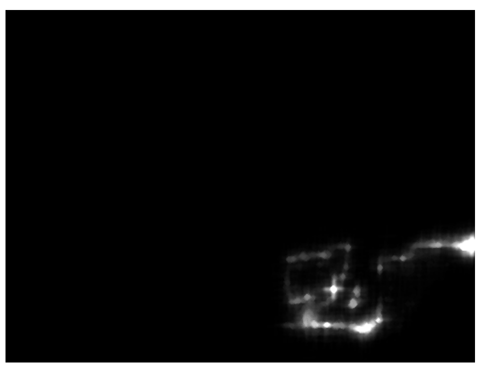


Previous Opt

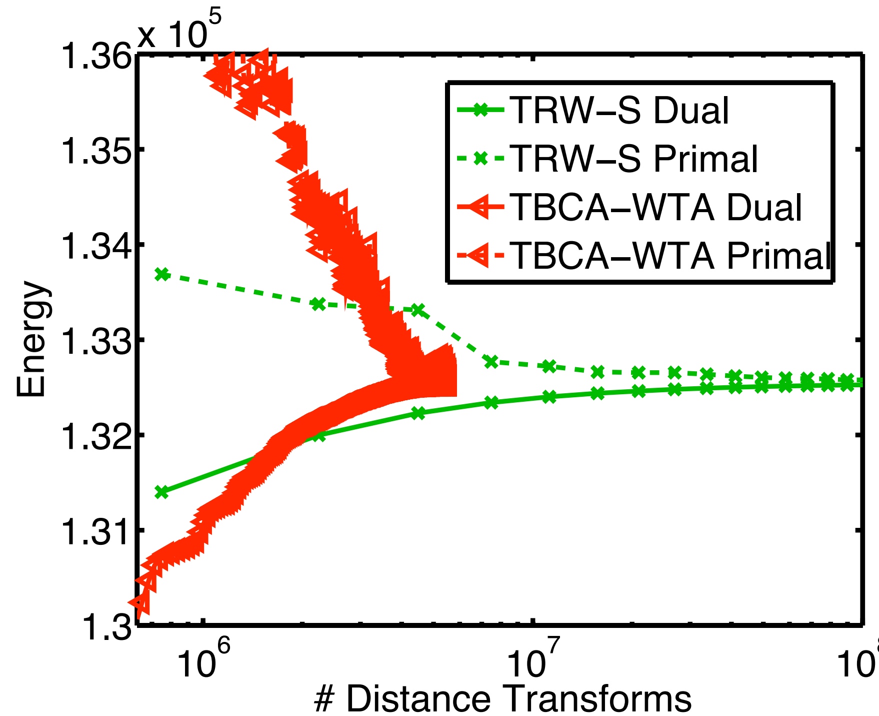
Modify White Unaries



New Opt



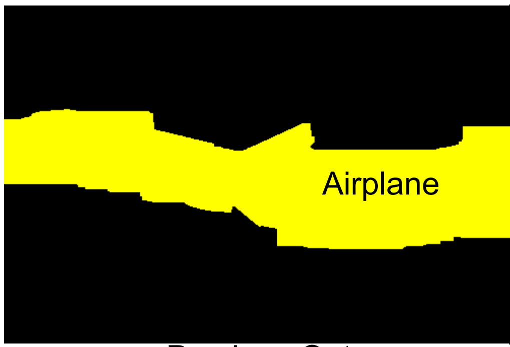
Heatmap of Messages



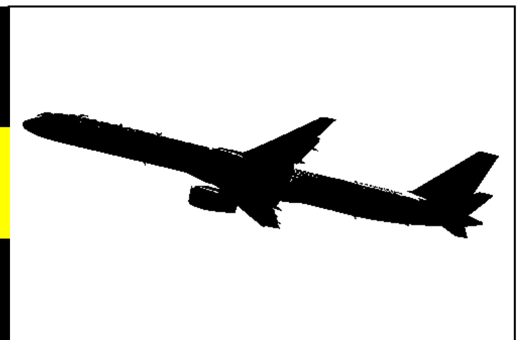
Warm-started DTBCA vs Warm-started TRW-S

375x500 pixels, 21 labels. Potts potentials

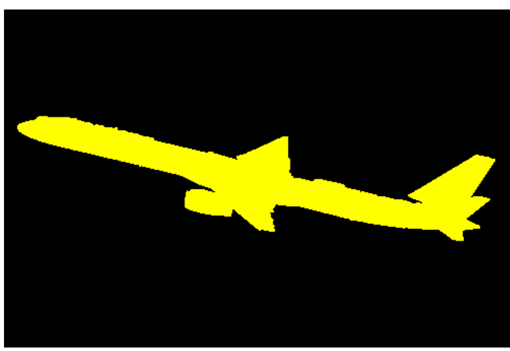
Dynamic Image Segmentation



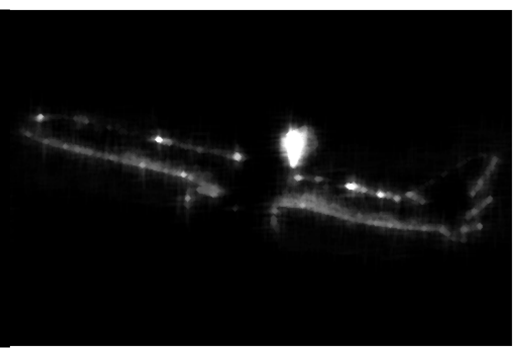
Previous Opt



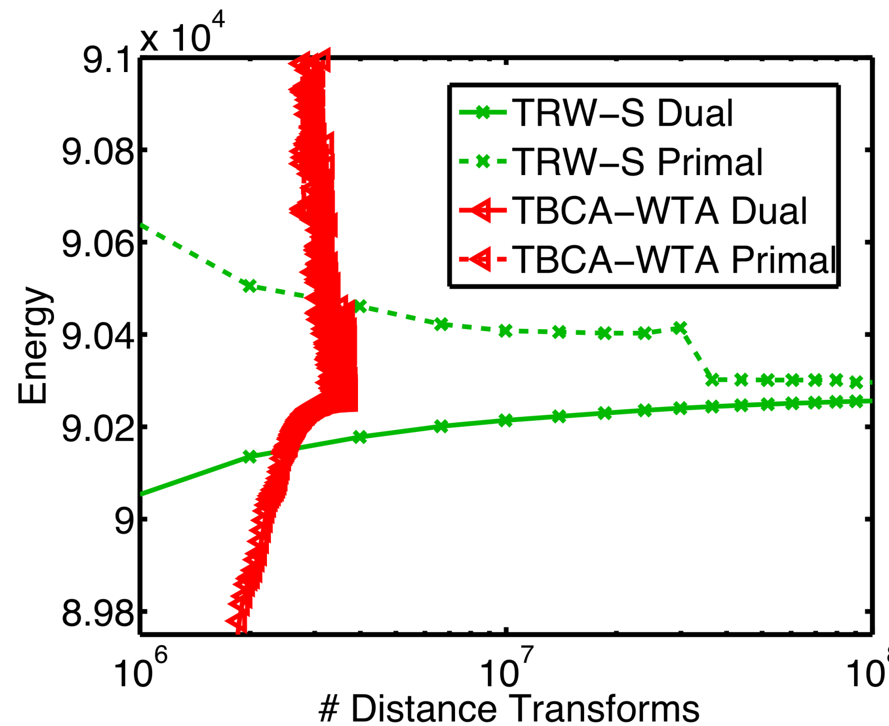
Modify White Unaries



New Opt



Heatmap of Messages



Warm-started DTBCA vs Warm-started TRW-S

375x500 pixels, 21 labels. Potts potentials

New Topic

- Making BP Exact
  - Connecting BP to VE on Junction Trees

Overview of BP

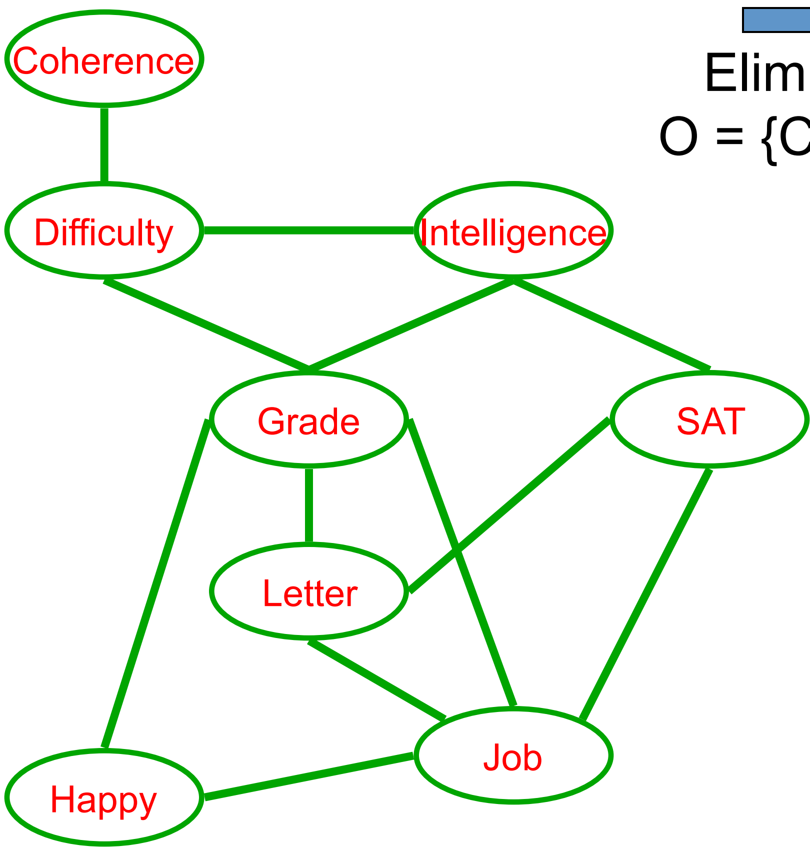
- Pick a graph to pass messages on
  - Cluster Graph

- Pick an ordering of edges
  - Round-robin
  - Leaves-Root-Leaves on a tree
  - Asynchonous

- Till convergence or exhaustion:
  - Pass messages on edges

- At vertices on graph compute *psuedo-marginals*

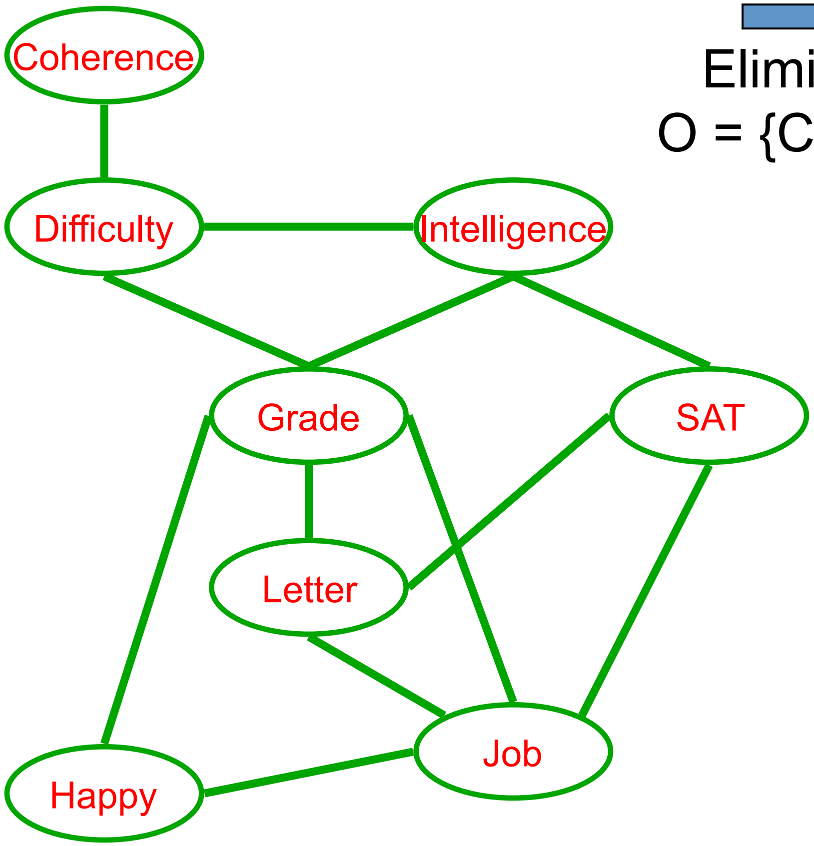
Induced graph



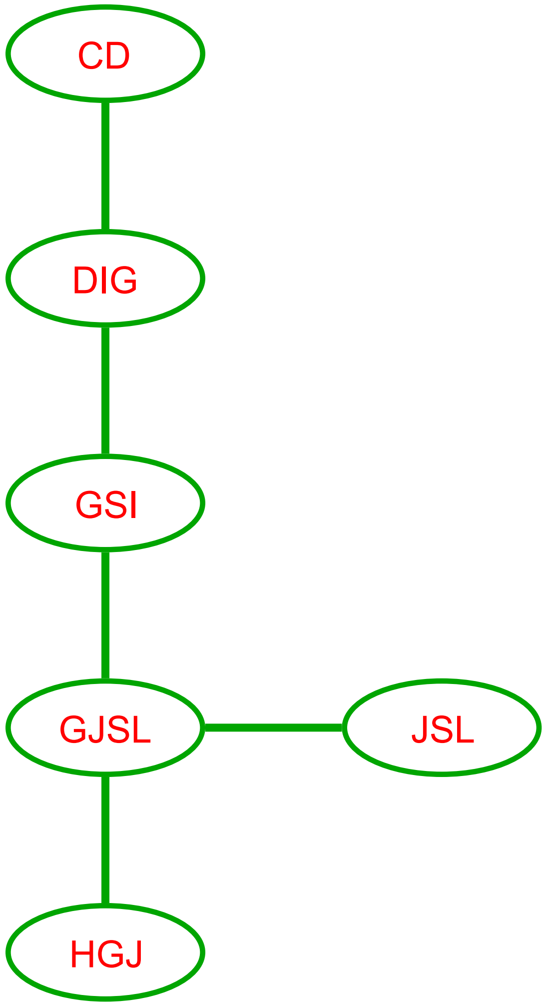
Elimination order:
O = {C,D,I,S,L,H,J,G}

The induced graph I_{FO} for elimination order O has an edge $X_i – X_j$ if X_i and X_j appear together in a factor generated by VE for elimination order O on factors F

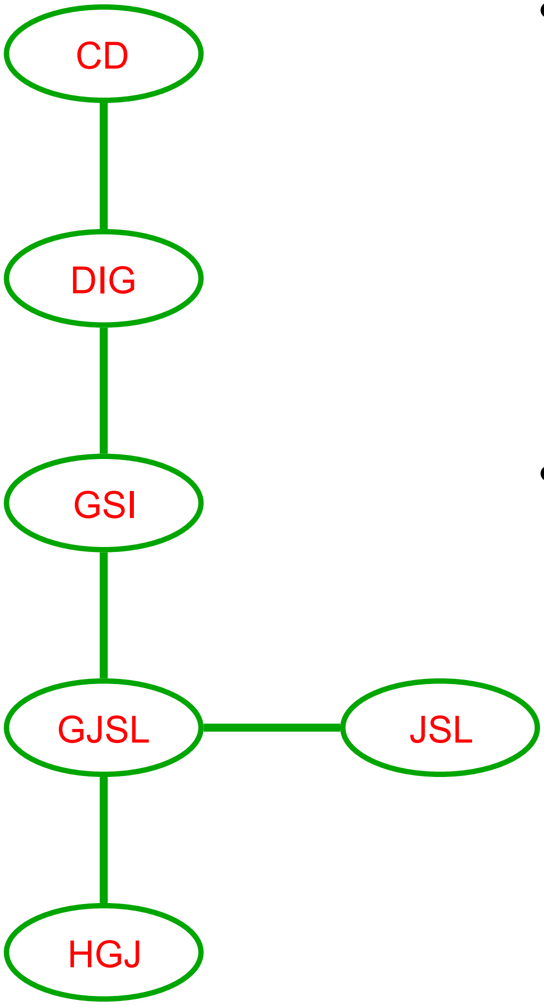
Factors Generated



Elimination order:
O = {C,D,I,S,L,H,J,G}



Cluster graph for VE



- **VE generates cluster tree!**
  - One cluster for each factor used/generated
  - Edge i – j, if f_i used to generate f_j
  - "Message" from i to j generated when marginalizing a variable from f_i
  - Tree because factors only used once
- **Proposition**:
  - "Message" δ_{ij} from i to j
  - Scope[δ_{ij}] $\subseteq \mathbf{S}_{ij}$