ICCV 2007 tutorial

Part III

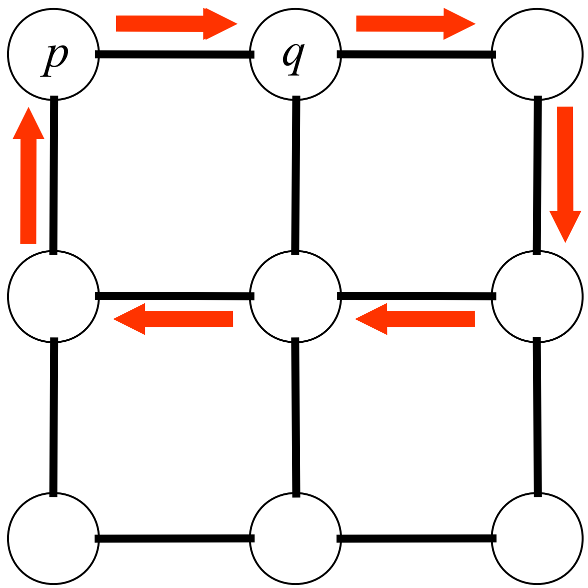
Message-passing algorithms for energy minimization

Vladimir Kolmogorov

University College
London

Message passing

$$E(\mathbf{x}) = \sum_{p} \theta_p(x_p) + \sum_{p,q} \theta_{pq}(x_p, x_q)$$



- *Iteratively pass messages between nodes...*
- Message update rule?
  - Belief propagation (BP)
  - Tree-reweighted belief propagation (TRW)

  - max-product (minimizing an energy function, or *MAP estimation*)
  - sum-product (computing marginal probabilities)
- Schedule?
  - Parallel, sequential, ...

Outline

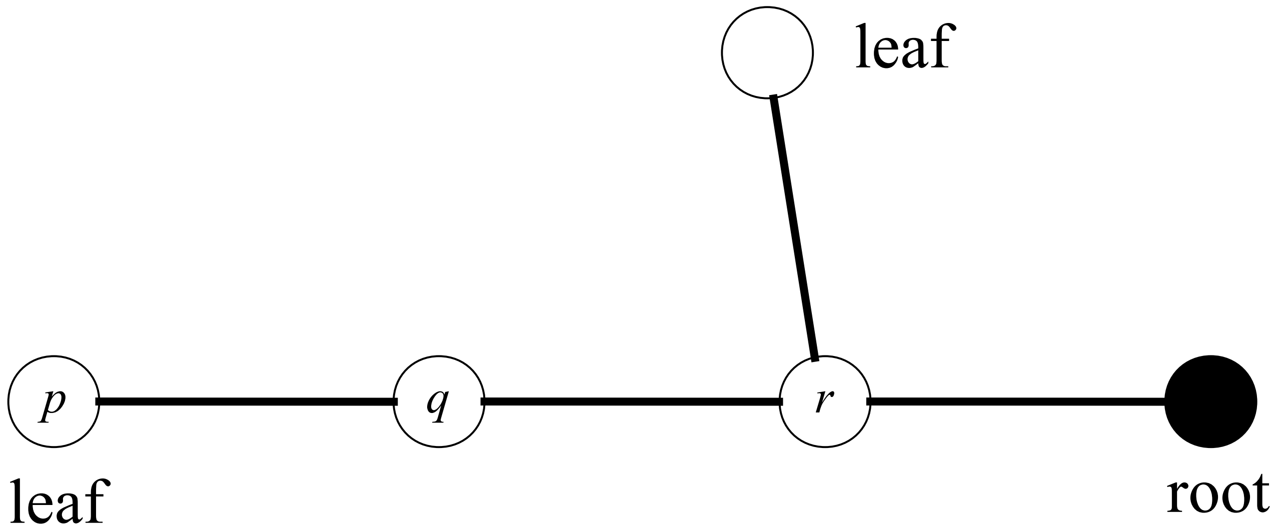
- Belief propagation
  - BP on a tree
    - Min-marginals
  - BP in a general graph
  - Distance transforms

- Reparameterization

- Tree-reweighted message passing
  - Lower bound via combination of trees
  - Message passing
  - Sequential TRW

Belief propagation (BP)

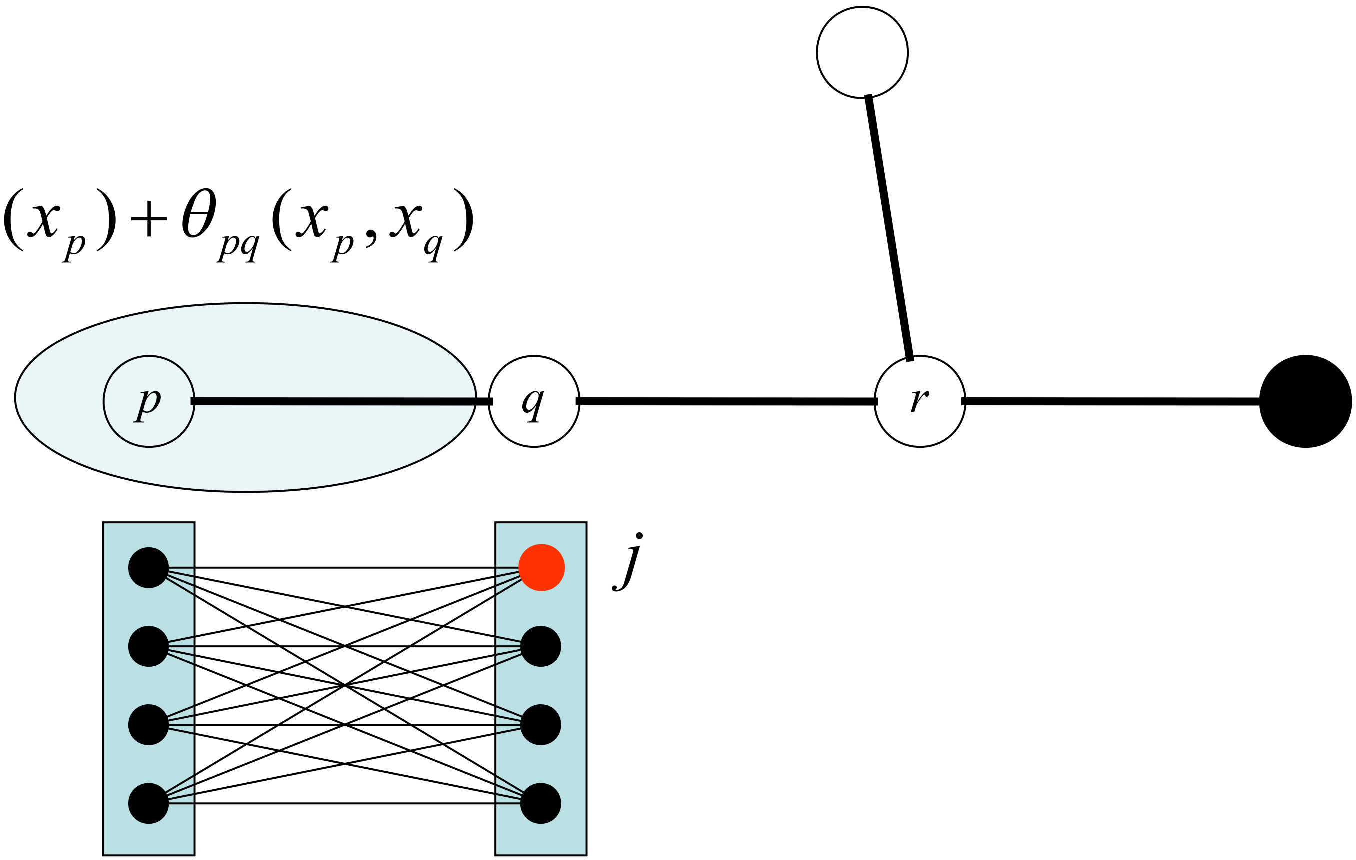
BP on a tree [Pearl'88]



- Dynamic programming: global minimum in linear time
- BP:
  - Inward pass (dynamic programming)
  - Outward pass
  - Gives min-marginals

Inward pass (dynamic programming)

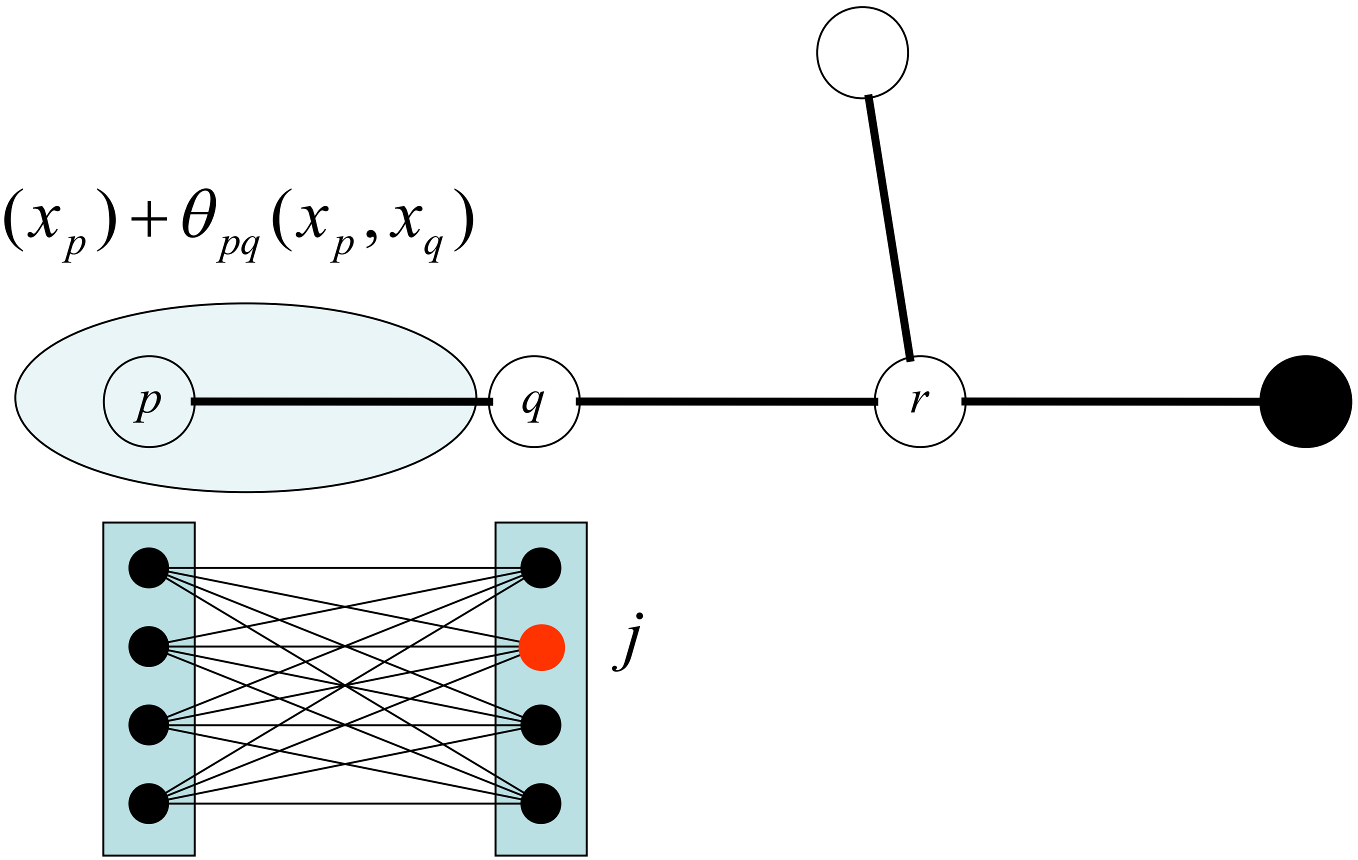
$$\theta_p(x_p) + \theta_{pq}(x_p, x_q)$$



$$M_{pq}(j) = \min_i \left\{ \theta_p(i) + \theta_{pq}(i, j) \right\}$$

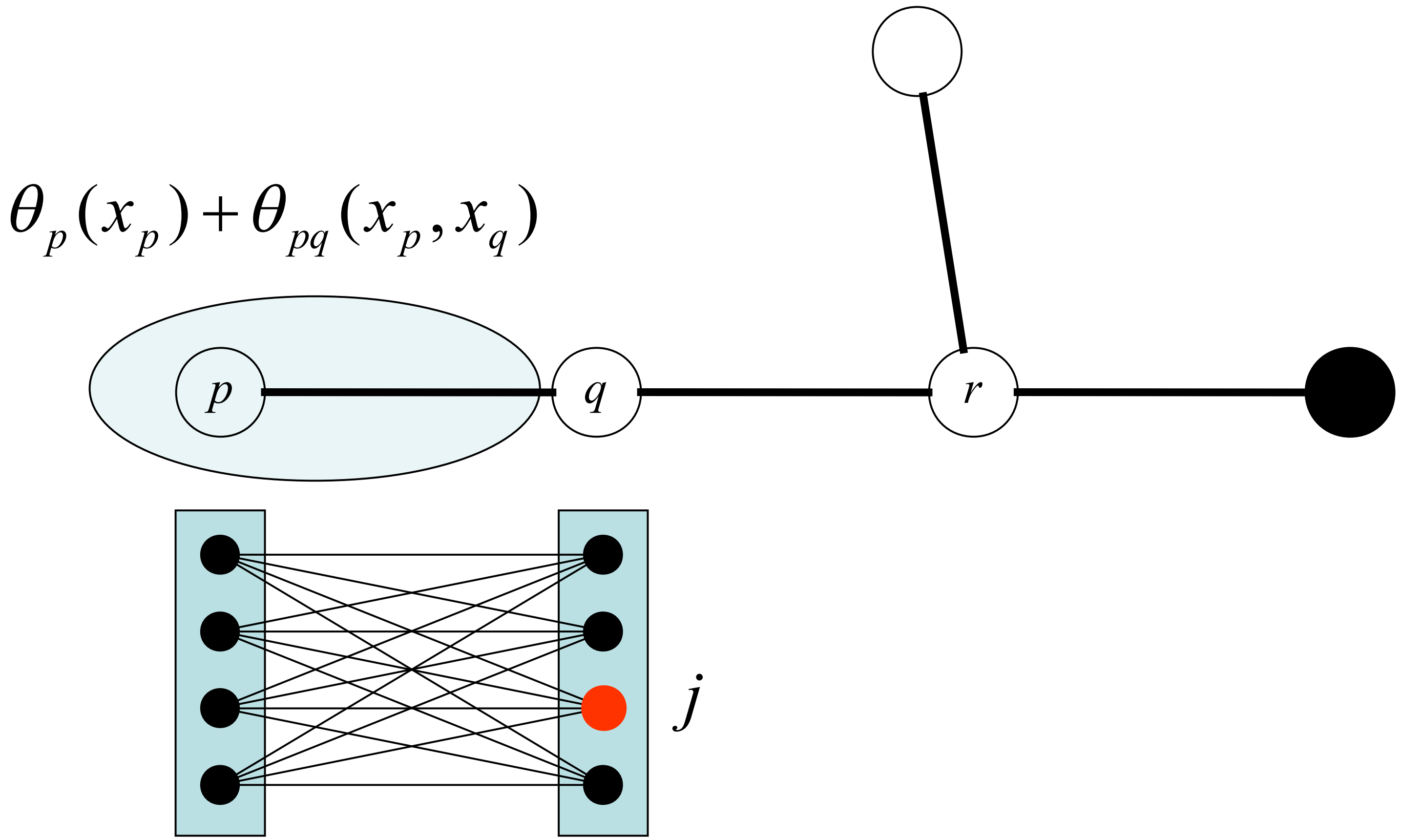
Inward pass (dynamic programming)

$$\theta_p(x_p) + \theta_{pq}(x_p, x_q)$$



$$M_{pq}(j) = \min_i \left\{ \theta_p(i) + \theta_{pq}(i, j) \right\}$$

Inward pass (dynamic programming)

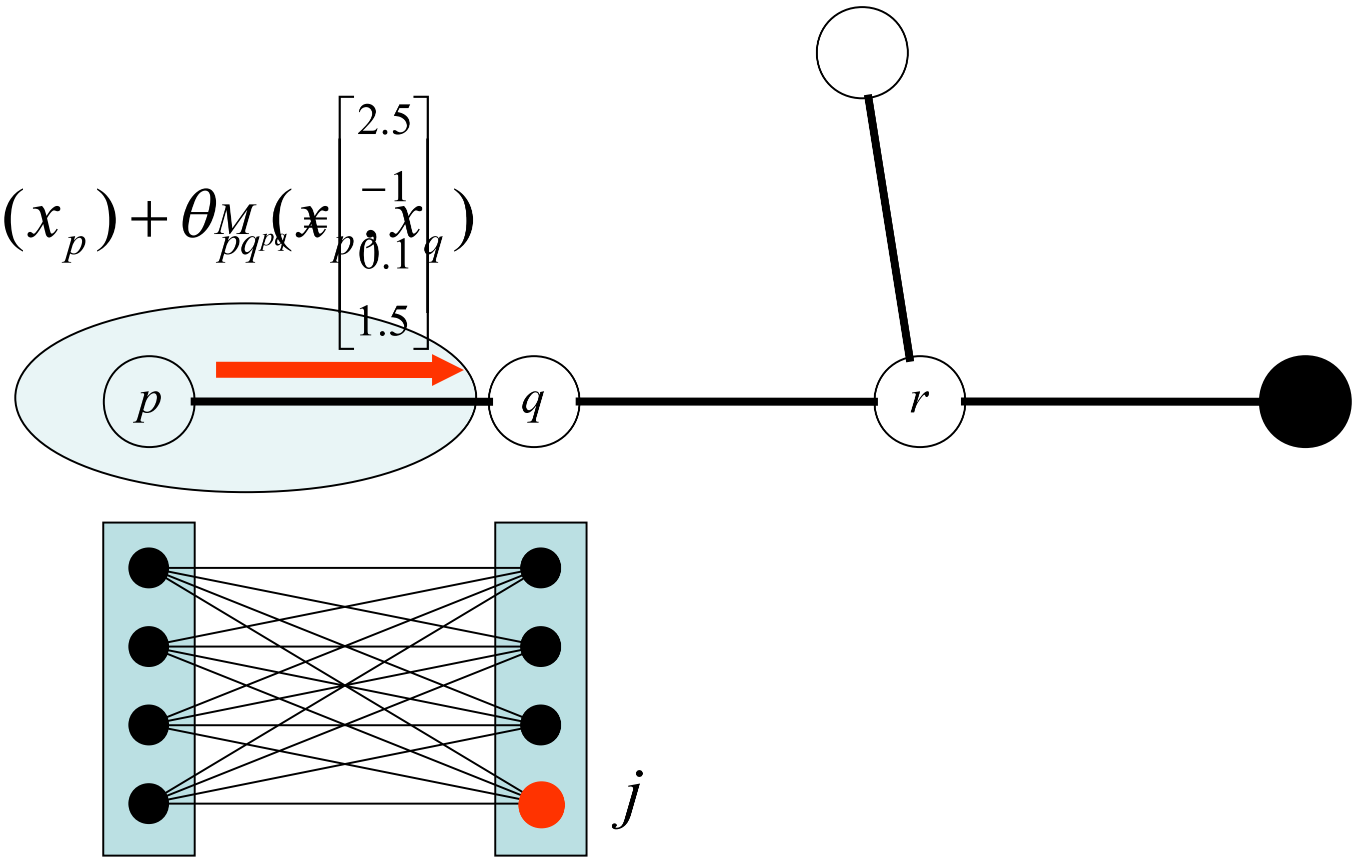


$$M_{pq}(j) = \min_i \left\{ \theta_p(i) + \theta_{pq}(i, j) \right\}$$

Inward pass (dynamic programming)

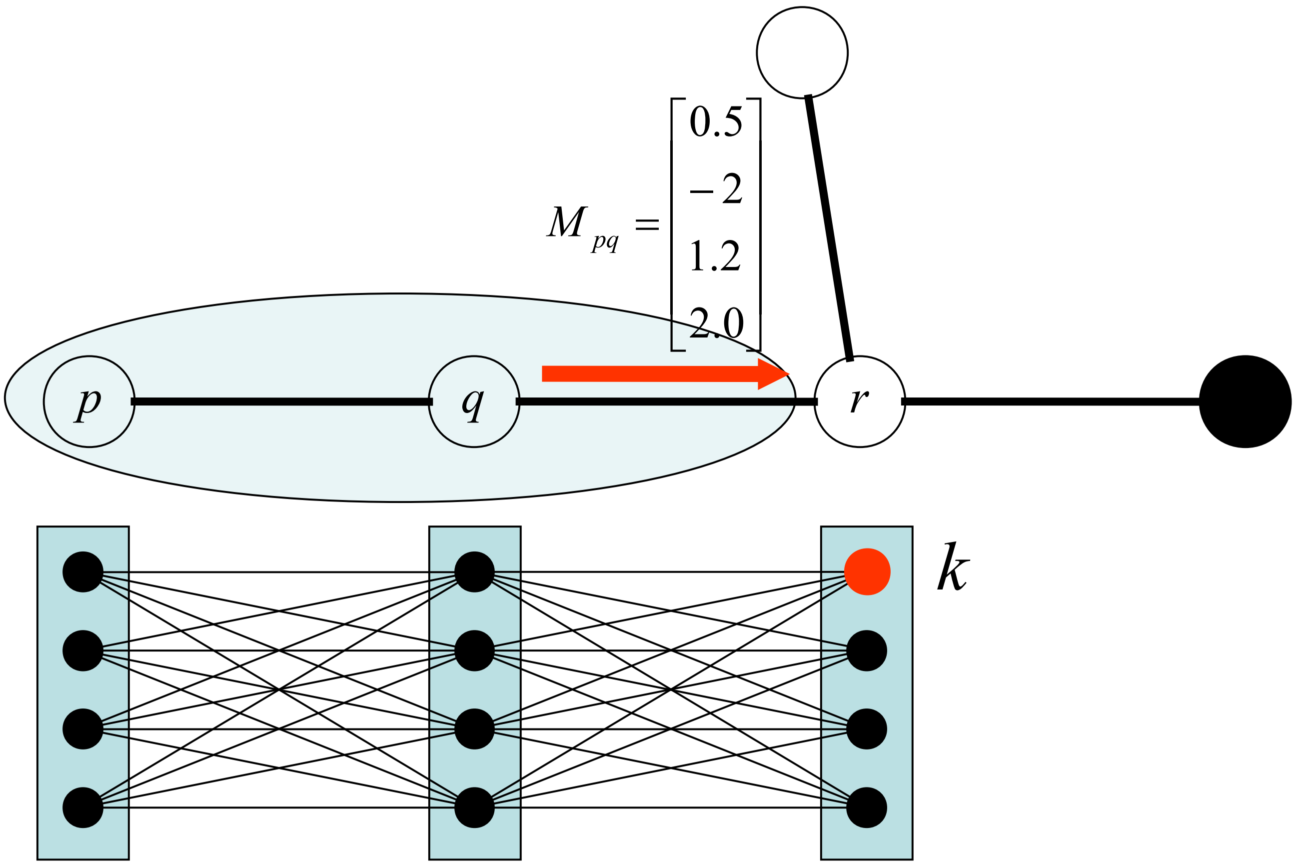
$$\theta_p(x_p) + \theta_{pq}(x_p, x_q)$$

$$M_{pq} = \begin{bmatrix} 2.5 \\ -1 \\ 0.1 \\ 1.5 \end{bmatrix}$$



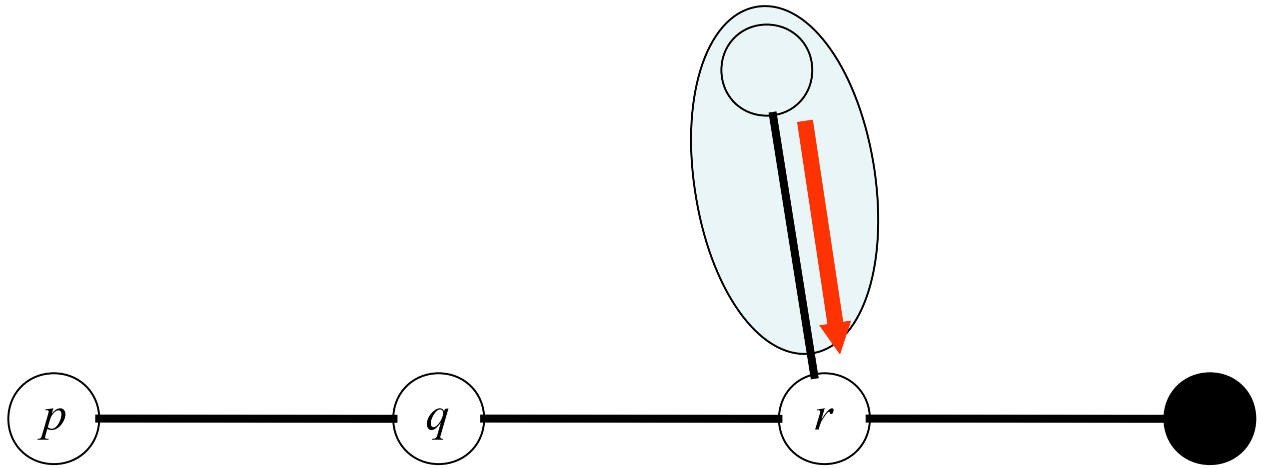
$$M_{pq}(j) = \min_i \left\{ \theta_p(i) + \theta_{pq}(i, j) \right\}$$

Inward pass (dynamic programming)

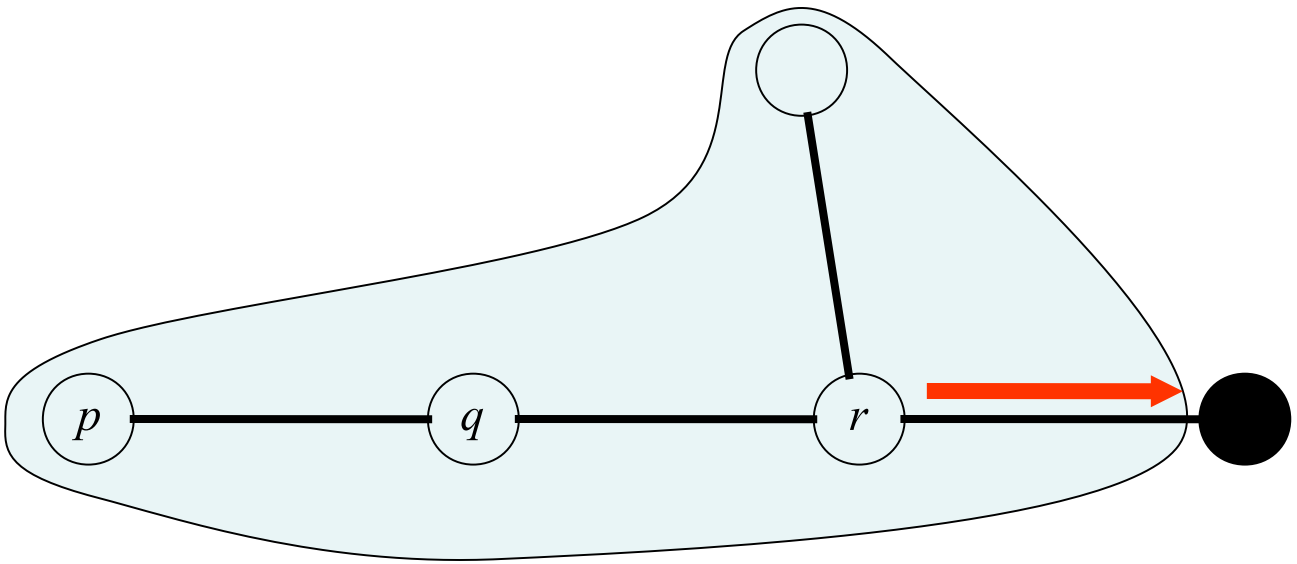


$$M_{qr}(k) = \min_j \left\{ \left(\theta_q(j) + M_{pq}(j) \right) + \theta_{qr}(j,k) \right\}$$

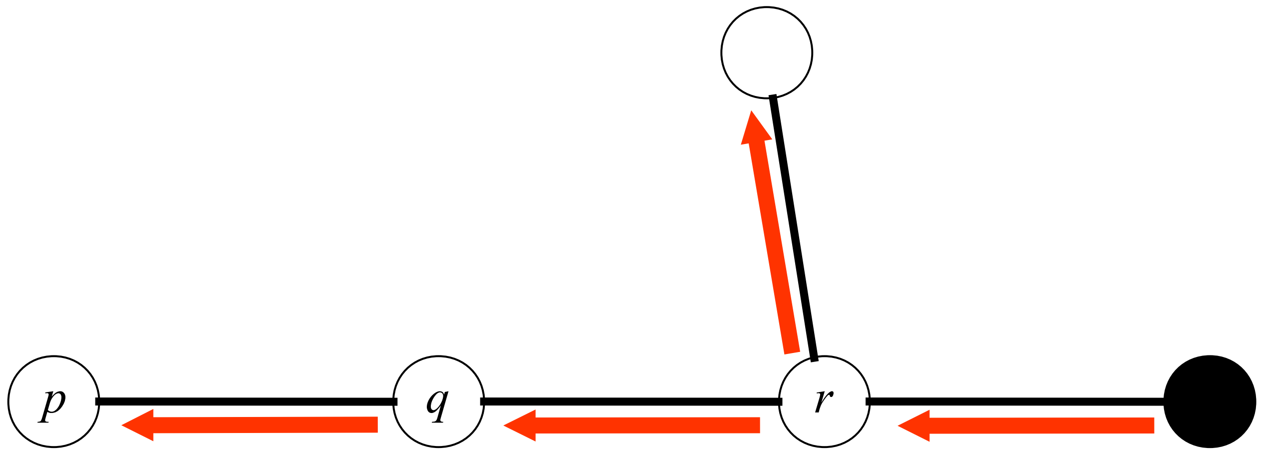
Inward pass (dynamic programming)



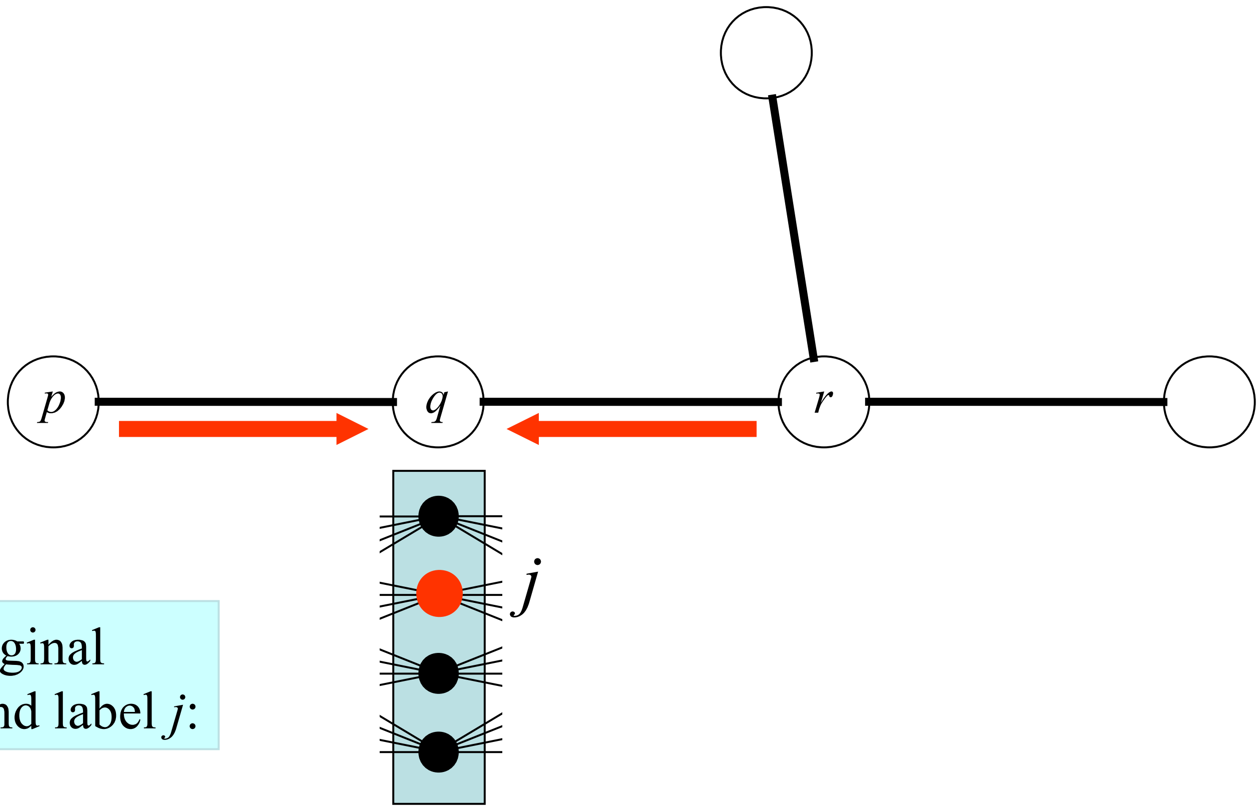
Inward pass (dynamic programming)



Outward pass



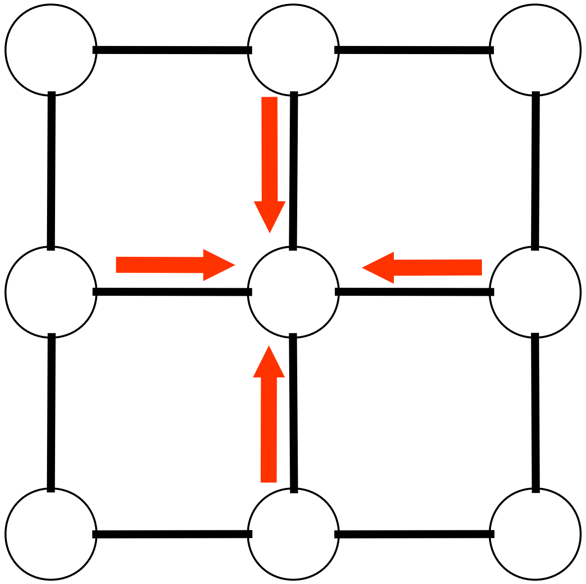
BP on a tree: min-marginals



Min-marginal
for node q and label j:

$$\min_{\mathbf{x}} \left\{ E(\mathbf{x}) \;\middle|\; x_q = j \right\} \qquad = \theta_q(j) + M_{pq}(j) + M_{rq}(j)$$

BP in a general graph

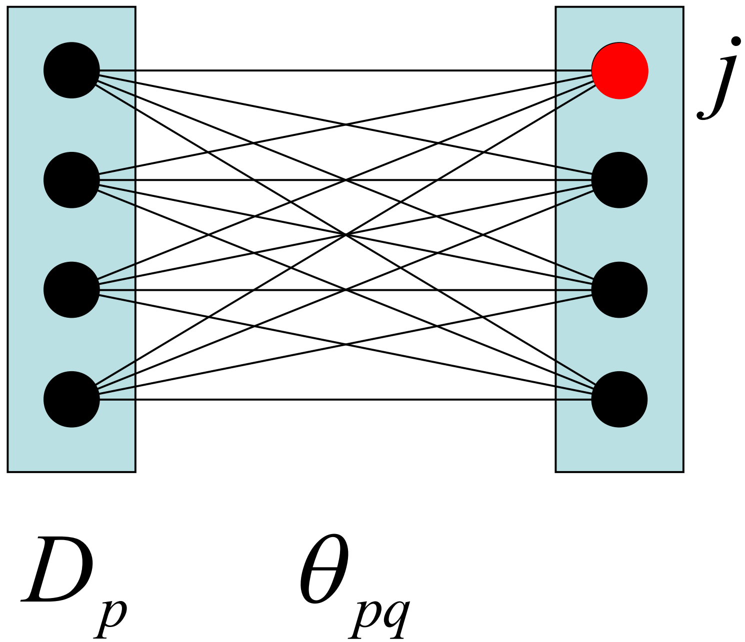


- Pass messages using same rules
  - Empirically often works quite well
- May not converge
- “Pseudo” min-marginals
- Gives local minimum in the “tree neighborhood” [Weiss&Freeman’01],[Wainwright et al.’04]
  - Assumptions:
    - BP has converged
    - no ties in pseudo min-marginals

Distance transforms

[Felzenszwalb & Huttenlocher'04]

- Naïve implementation: $O(K^2)$
- Often can be improved to $O(K)$
  - Potts interactions, truncated linear, truncated quadratic, ...

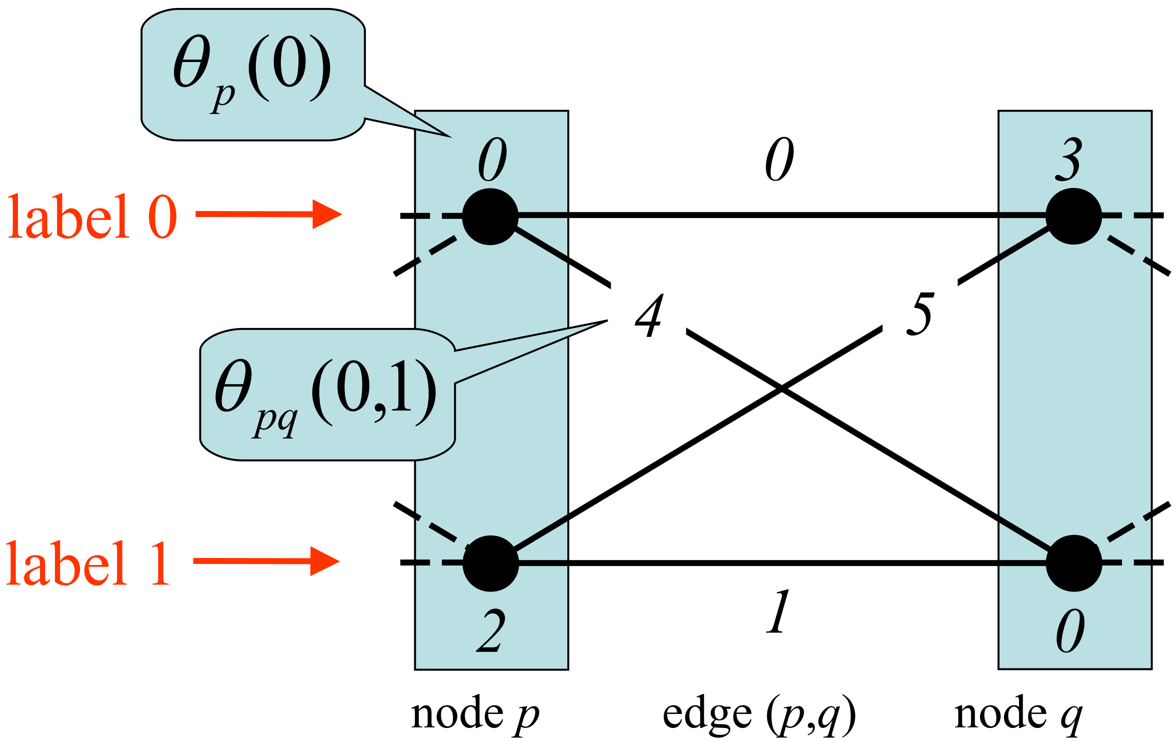


$$M_{pq}(j) = \min_i \left\{ D_p(i) + \theta_{pq}(i,j) \right\}$$

Reparameterization

Energy function - visualization

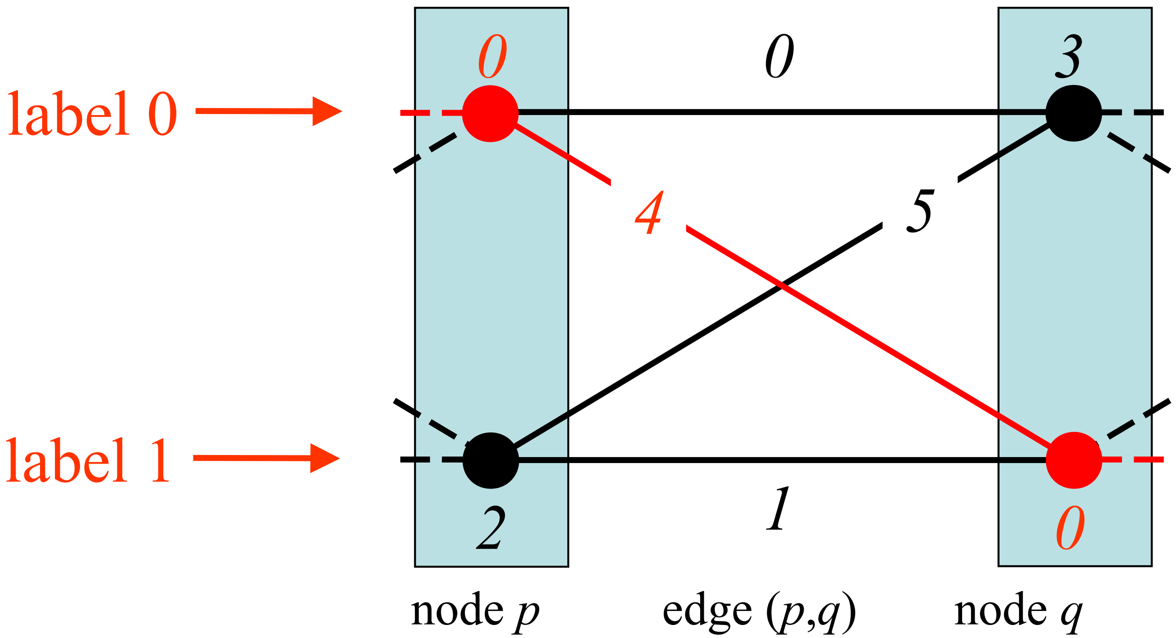
$$E(\mathbf{x} \mid \theta) = \sum_{p} \theta_p(x_p) + \sum_{p,q} \theta_{pq}(x_p, x_q)$$



$\theta =$ vector of all parameters

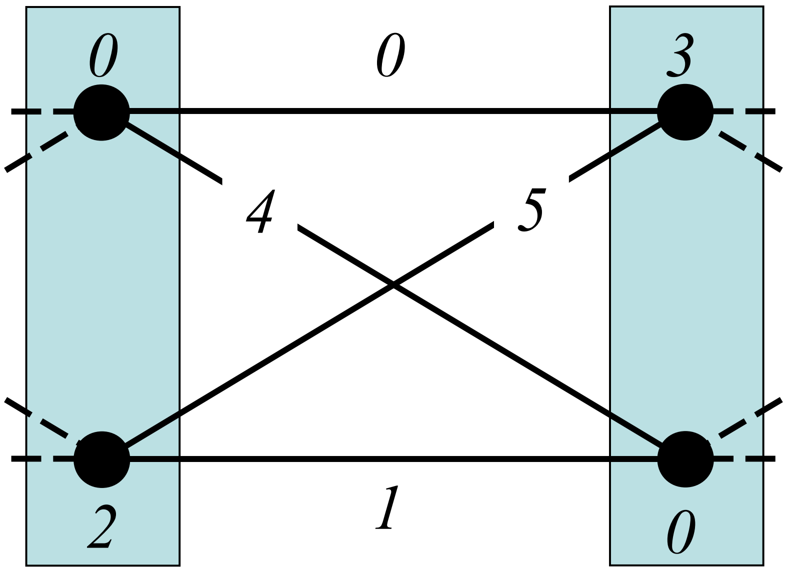
Energy function - visualization

$$E(\mathbf{x} \mid \theta) = \sum_p \theta_p(x_p) + \sum_{p,q} \theta_{pq}(x_p, x_q)$$

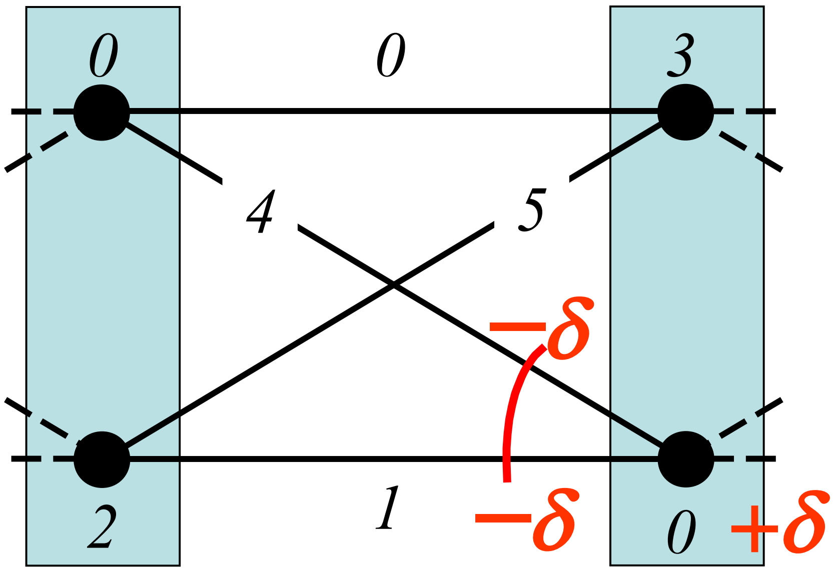


$\theta =$ vector of all parameters

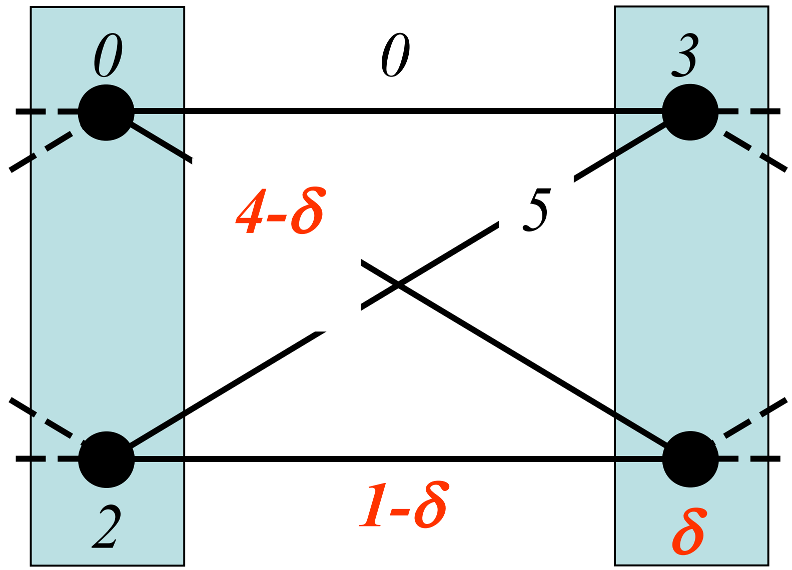
Reparameterization



Reparameterization



Reparameterization



- **<u>Definition.</u>** θ' is a reparameterization of θ if they define the same energy:

$$E(\mathbf{x} \mid \theta') = E(\mathbf{x} \mid \theta) \qquad \forall \mathbf{x}$$

- Maxflow, BP and TRW perform reparameterisations

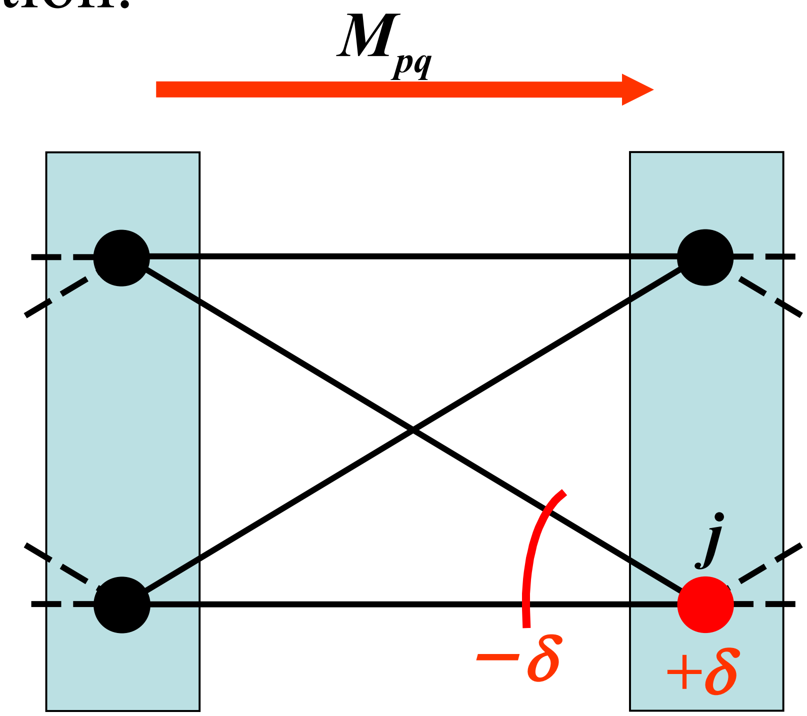
BP as reparameterization

[Wainwright et al. 04]

- Messages define reparameterization:

$$\theta'_{pq}(i,j) = \theta_{pq}(i,j) - M_{pq}(j) - M_{qp}(i)$$

$$\theta'_q(j) = \underbrace{\theta_q(j) + \sum_{p,q} M_{pq}(j)}_{\text{min-marginals (for trees)}}$$



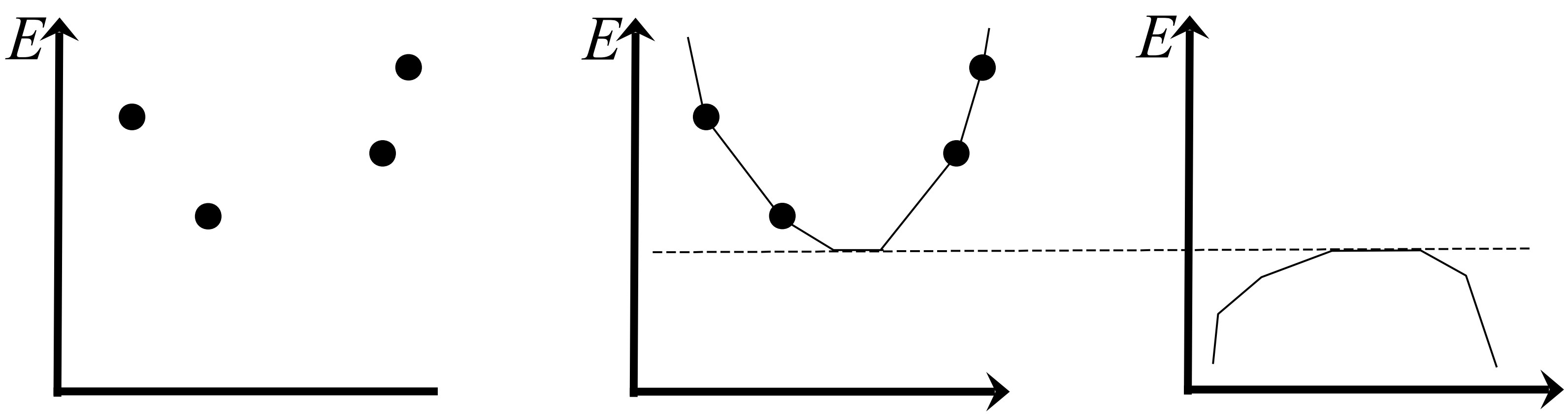
$\boldsymbol{\delta = M_{pq}(j)}$

- BP on a tree: reparameterize energy so that unary potentials become min-marginals

Tree-reweighted message passing (TRW)

Linear Programming relaxation

- Energy minimization: NP-hard problem
- Relax integrality constraint: $x_p \in \{0,1\} \;\Rightarrow\; x_p \in [0,1]$
  - LP relaxation [Schlesinger'76,Koster et al.'98,Chekuri et al.'00,Wainwright et al.'03]
- Try to solve dual problem:
  - Formulate lower bound on the function
  - Maximize the bound



Energy function with discrete variables

LP relaxation

Lower bound on the energy function

Convex combination of trees

[Wainwright, Jaakkola, Willsky '02]

- Goal: compute minimum of the energy for θ:

$$\Phi(\theta) = \min_{\mathbf{x}} E(\mathbf{x} \mid \theta)$$

- Obtaining lower bound:
  - Split θ into several components: $\theta = \theta^1 + \theta^2 + \ldots$
  - Compute minimum for each component:

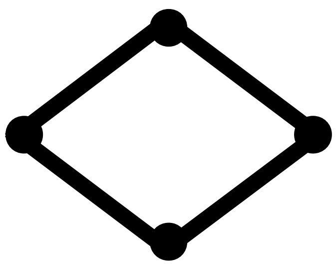
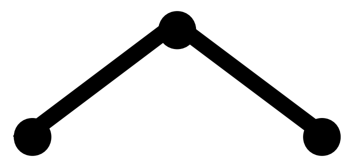
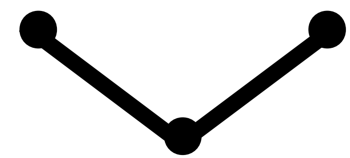
$$\Phi(\theta^i) = \min_{\mathbf{x}} E(\mathbf{x} \mid \theta^i)$$

  - Combine $\Phi(\theta^1)$, $\Phi(\theta^2)$, ... to get a bound on $\Phi(\theta)$

- Use trees!

Convex combination of trees (cont'd)

graph | tree T | tree T'

$$\theta \equiv \frac{1}{2}\theta^{T} + \frac{1}{2}\theta^{T'}$$

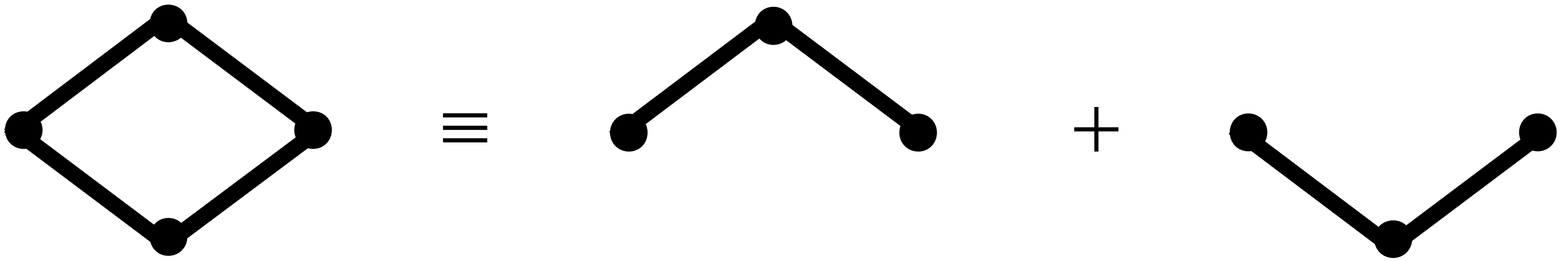
$$\Phi(\theta) \geq \underbrace{\frac{1}{2}\Phi(\theta^{T}) + \frac{1}{2}\Phi(\theta^{T'})}_{\text{lower bound on the energy}}$$

maximize

Maximizing lower bound

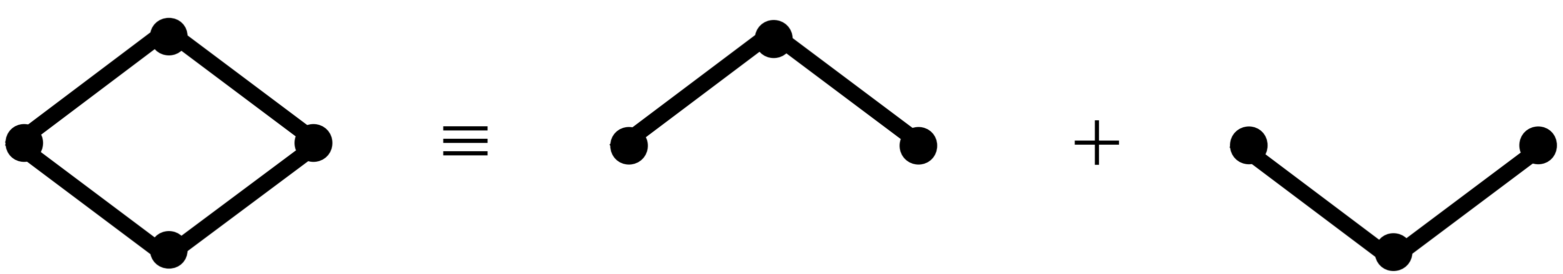
- Subgradient methods
  - [Schlesinger&Giginyak'07], [Komodakis et al.'07]

- Tree-reweighted message passing (TRW)
  - [Wainwright et al.'02], [Kolmogorov'05]



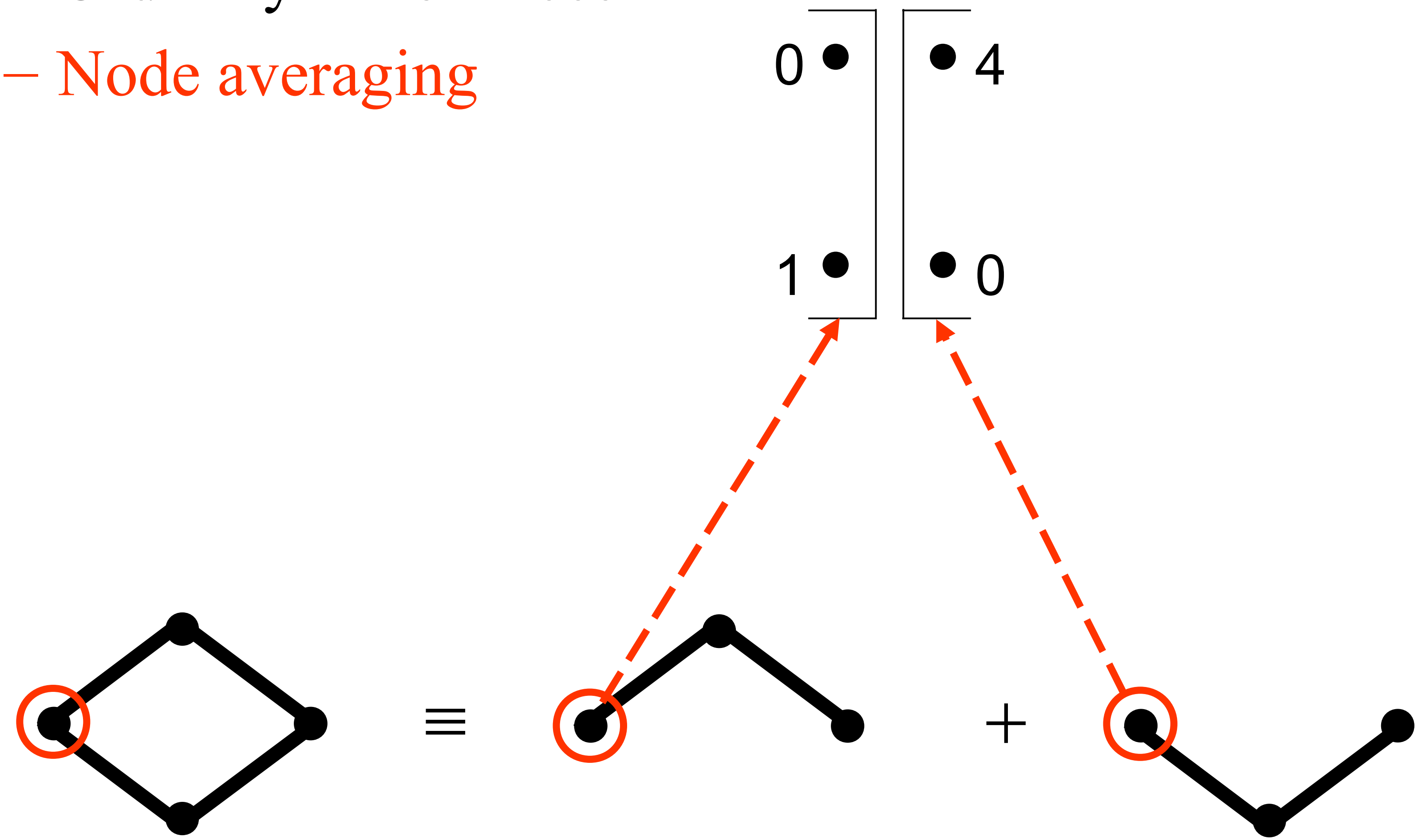
TRW algorithms

- Two reparameterization operations:
  - Ordinary BP on trees
  - Node averaging



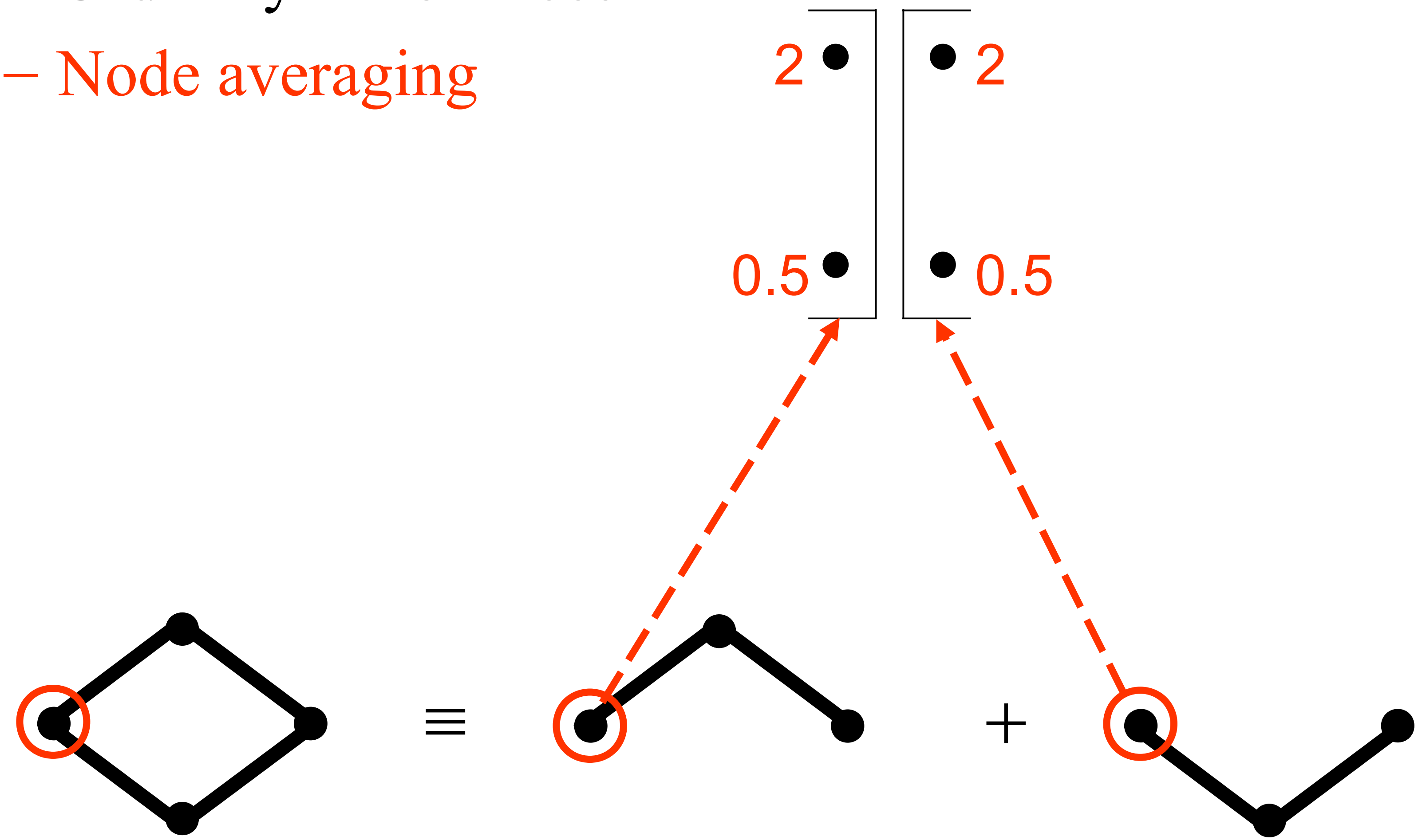
TRW algorithms

- Two reparameterization operations:
  - Ordinary BP on trees
  - Node averaging



TRW algorithms

- Two reparameterization operations:
  - Ordinary BP on trees
  - Node averaging



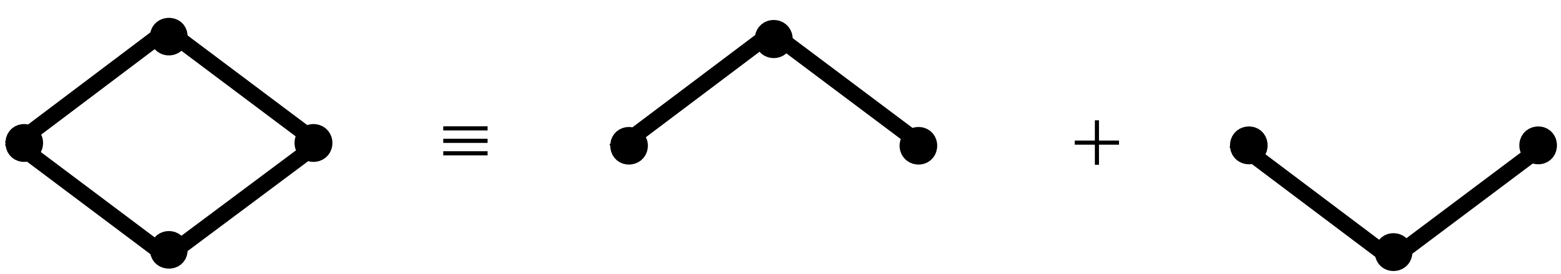
TRW algorithms

- Order of operations?
  - Affects performance dramatically
- Algorithms:
  - [Wainwright *et al*. '02]: parallel schedule (TRW-E, TRW-T)
    - May not converge
  - [Kolmogorov'05]: specific sequential schedule (TRW-S)
    - Lower bound does not decrease, convergence guarantees
    - Needs half the memory

TRW algorithm of Wainwright et al. with tree-based updates (TRW-T)

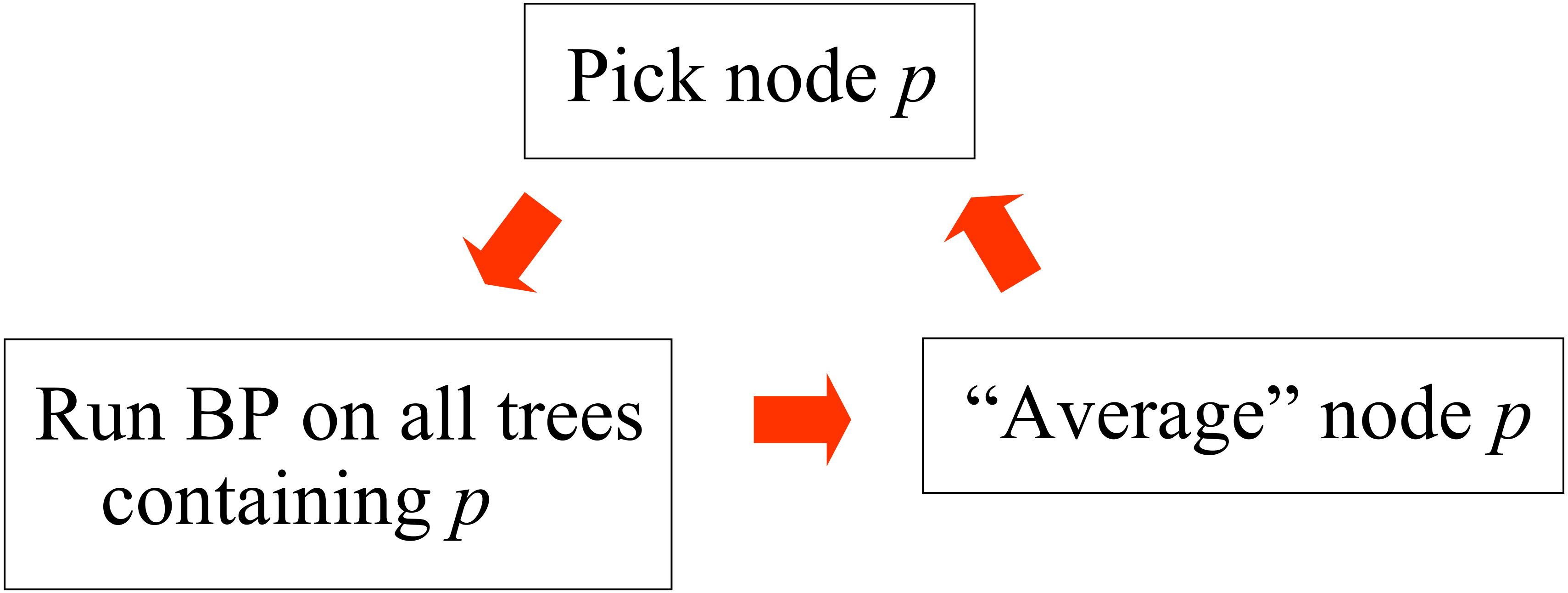
Run BP on *all* trees ⇄ "Average" *all* nodes

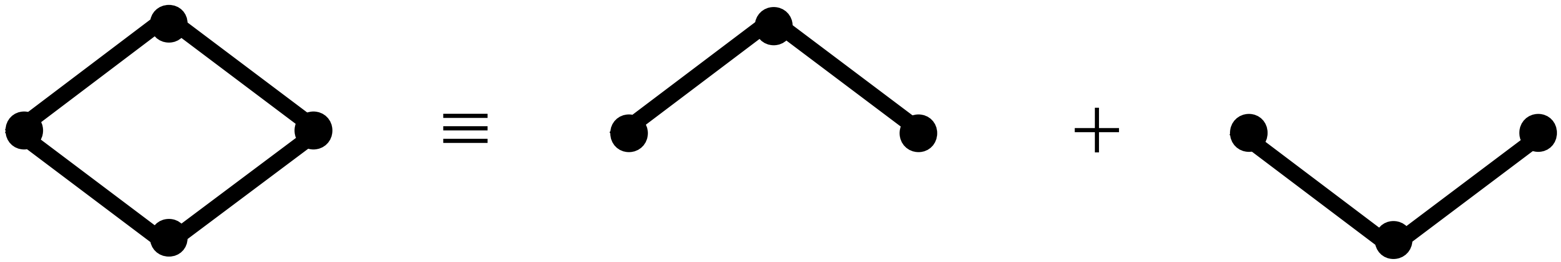
- **<u>If</u>** converges, gives (local) maximum of lower bound
- Not guaranteed to converge.
- Lower bound may go down.



Sequential TRW algorithm (TRW-S)

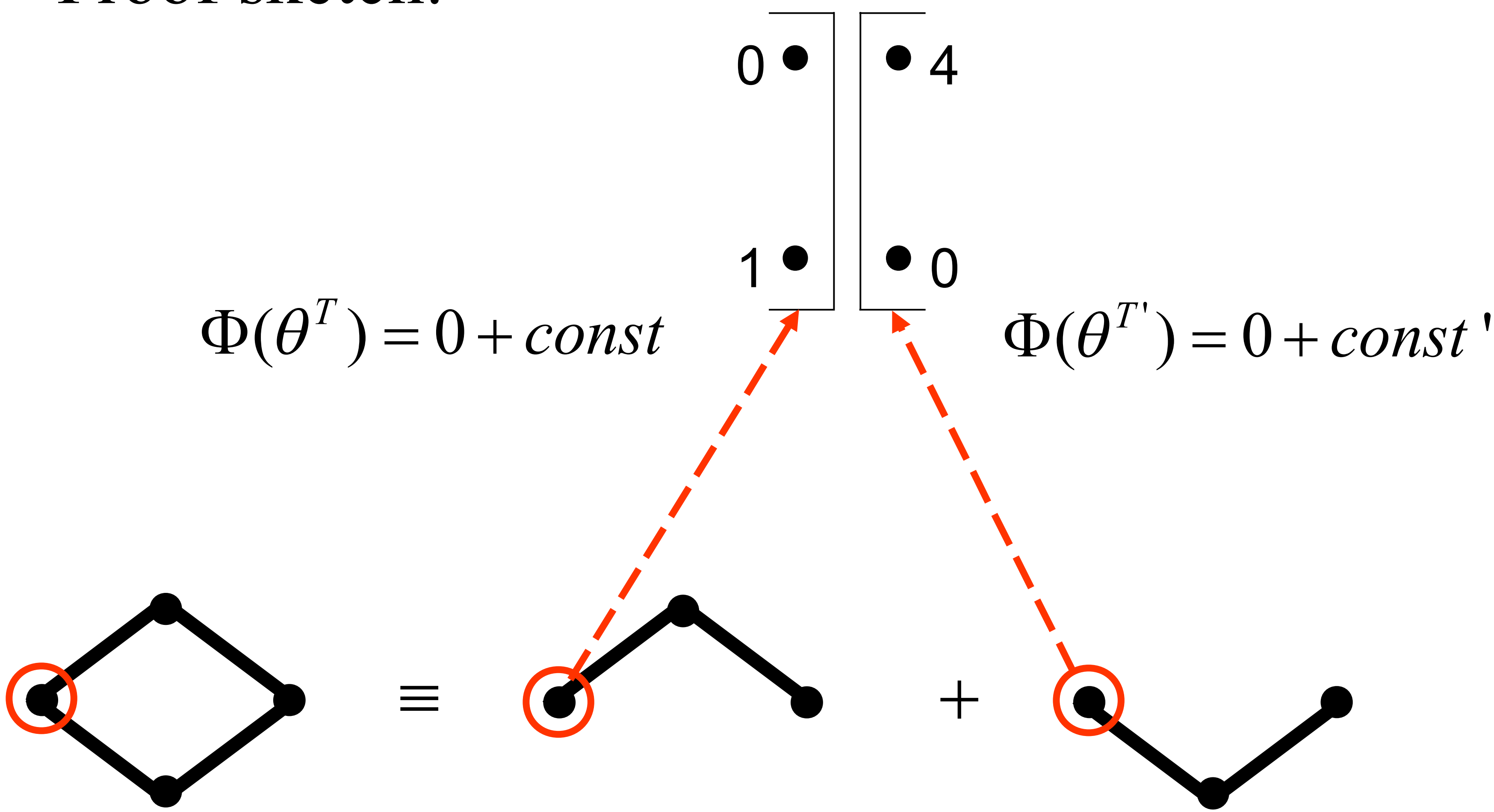
[Kolmogorov'05]





Main property of TRW-S

- **Theorem:** lower bound never decreases.
- Proof sketch:

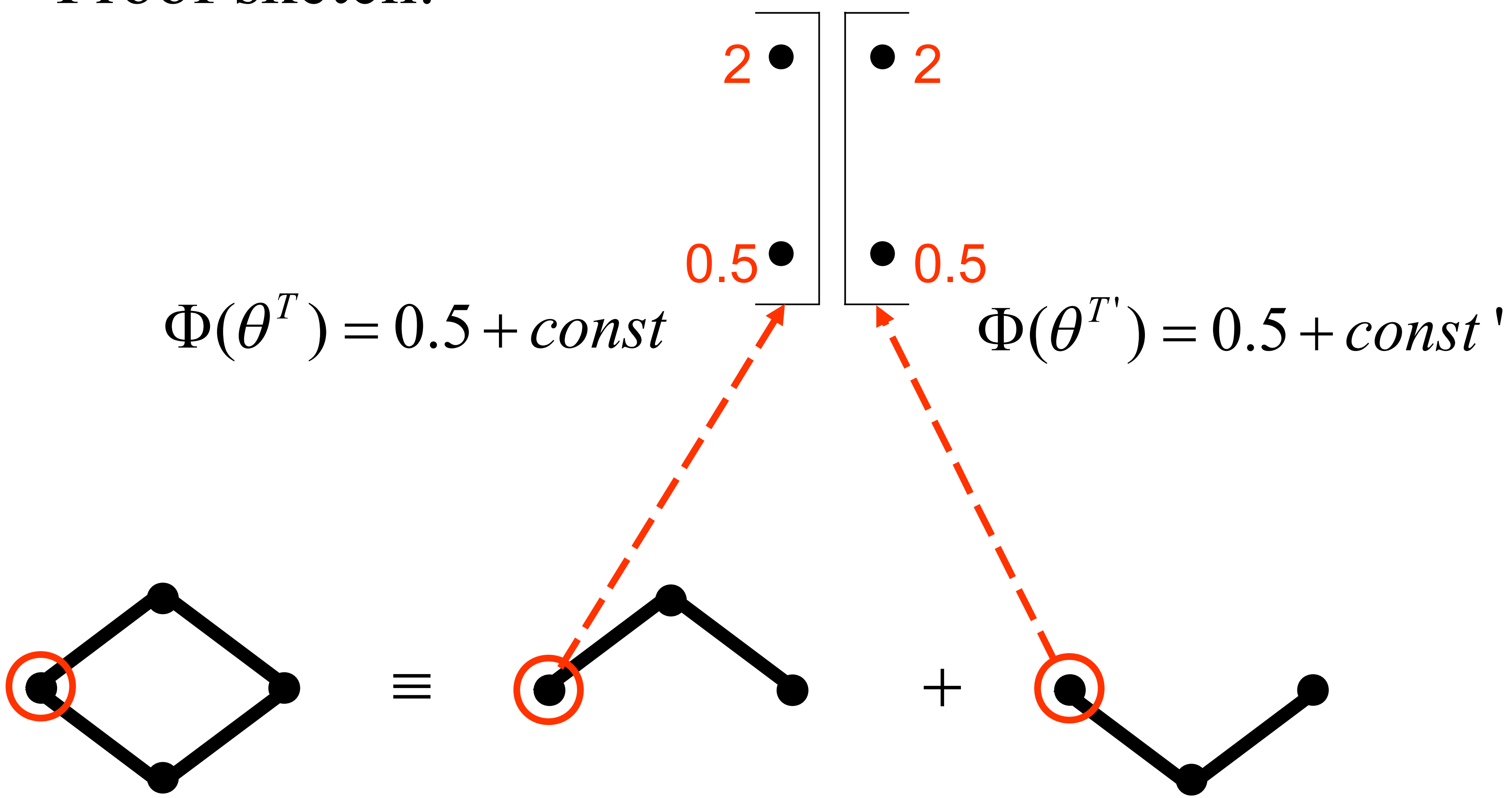


$$\Phi(\theta^{T}) = 0 + const$$

$$\Phi(\theta^{T'}) = 0 + const'$$

Main property of TRW-S

- **<u>Theorem:</u>** lower bound never decreases.
- Proof sketch:



TRW-S algorithm

- Particular order of averaging and BP operations

- Lower bound guaranteed not to decrease

- There exists limit point that satisfies
  weak tree agreement condition

- Efficiency?

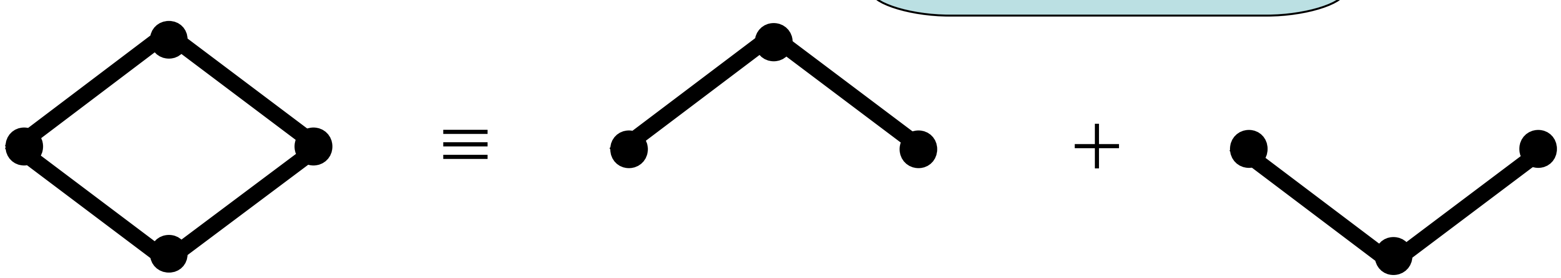
Efficient implementation

Pick node *p*

Run BP on all trees containing *p*

"Average" node *p*

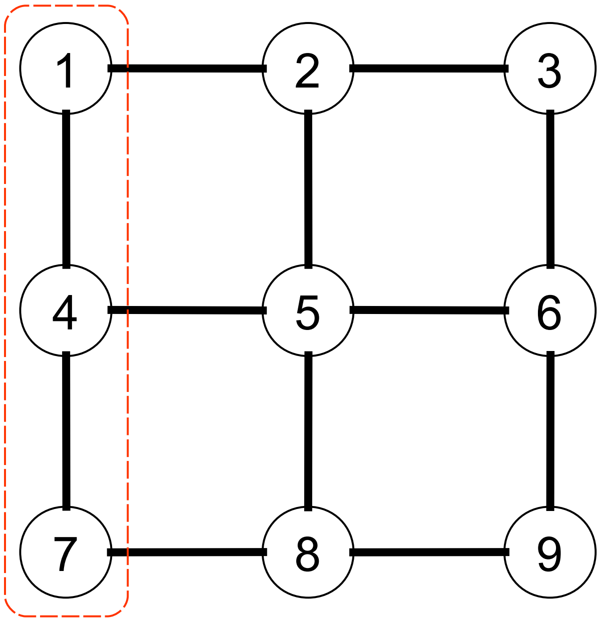
inefficient?



Efficient implementation

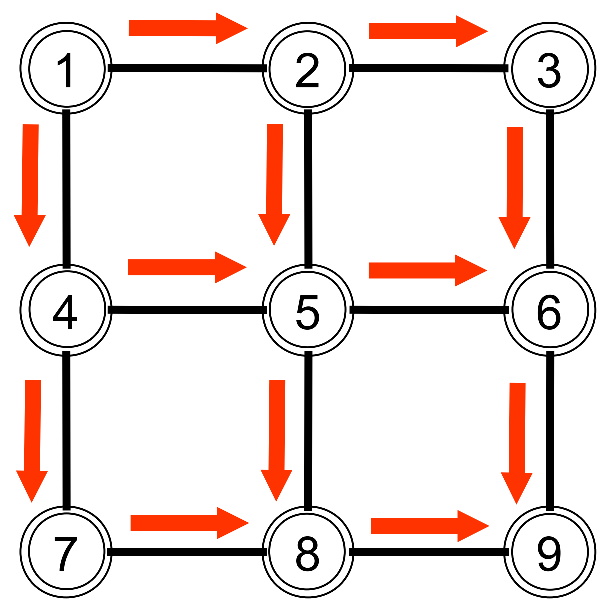
- **<u>Key observation</u>**:
  Node averaging operation preserves messages oriented towards this node
- Reuse previously passed messages!

- Need a special choice of trees:
  - Pick an ordering of nodes
  - Trees: *monotonic* chains



Efficient implementation

- Algorithm:
  - Forward pass:
    - process nodes in the increasing order
    - pass messages from lower neighbours
  - Backward pass:
    - do the same in reverse order

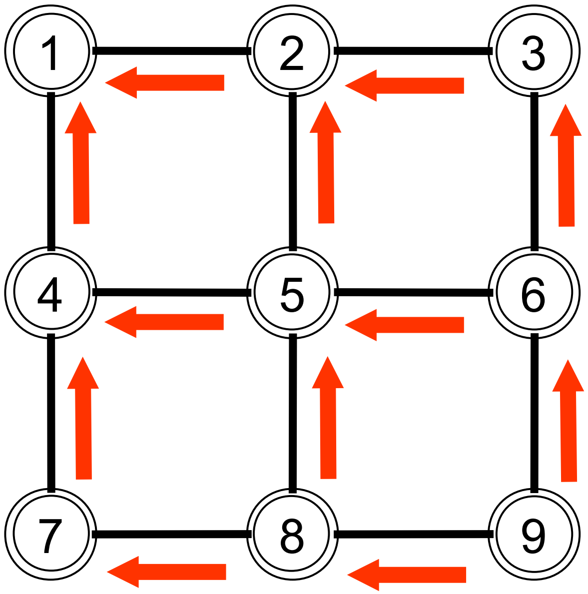


- Linear running time of one iteration

Efficient implementation

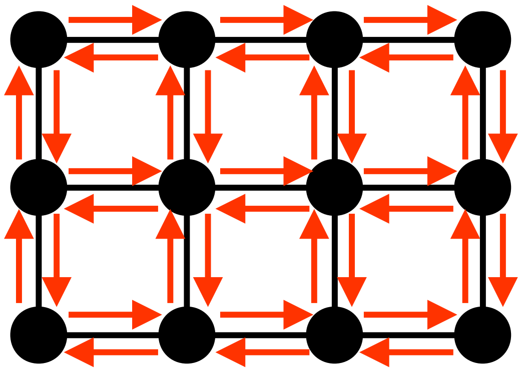
- Algorithm:
  - Forward pass:
    - process nodes in the increasing order
    - pass messages from lower neighbours
  - Backward pass:
    - do the same in reverse order

- Linear running time of one iteration

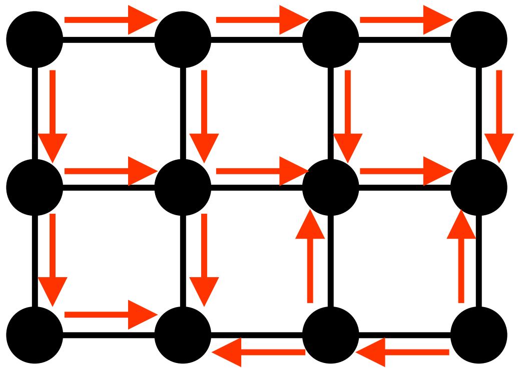


Memory requirements

- Standard message passing: 2 messages per edge
- TRW-S: 1 message per edge
  - Similar observation for bipartite graphs and parallel schedule in [Felzenszwalb&Huttenlocher'04]

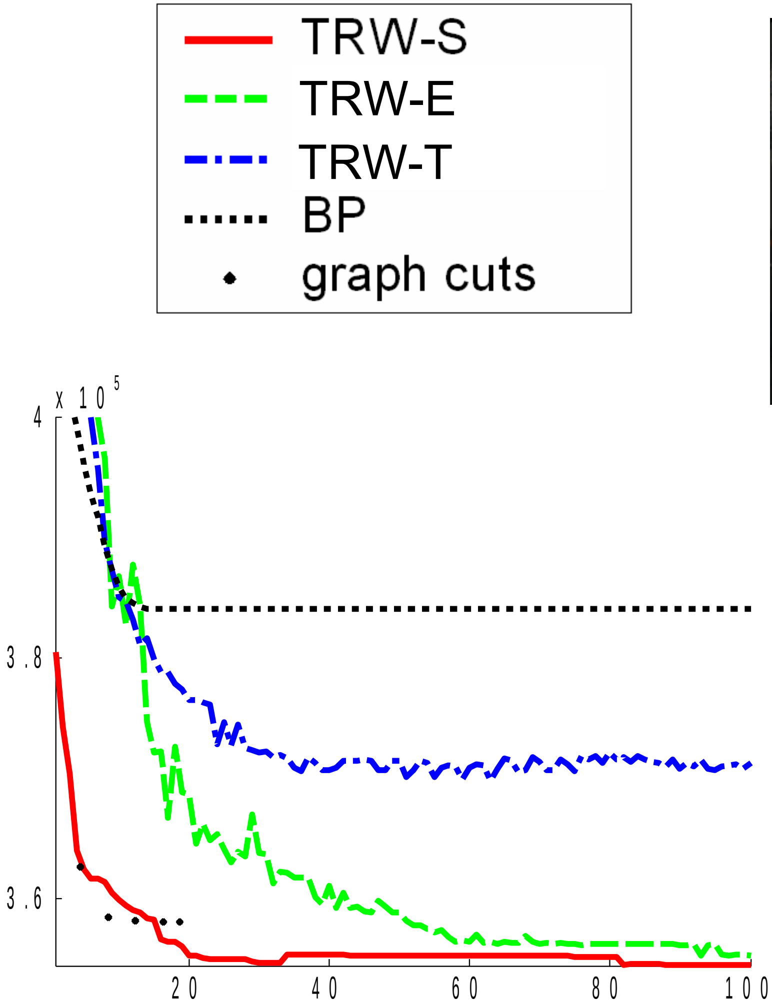


standard message passing



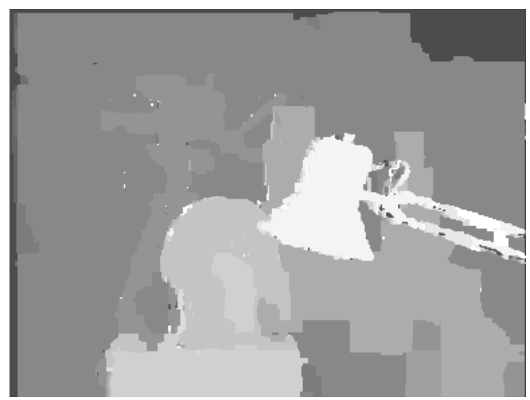
TRW-S

Experimental results: stereo



left image

ground truth



BP

TRW-S

- Global minima for some instances with TRW [Meltzer,Yanover,Weiss'05]
- See evaluation of MRF algorithms [Szeliski et al.'07]

Conclusions

- BP
  - Exact on trees
    - Gives min-marginals (unlike dynamic programming)
  - If there are cycles, heuristic
  - Can be viewed as reparameterization

- TRW
  - Tries to maximize a lower bound
  - TRW-S:
    - lower bound never decreases
    - limit point - weak tree agreement
    - efficient with monotonic chains

  - Not guaranteed to find an optimal bound!
    - See subgradient techniques [Schlesinger&Giginyak'07], [Komodakis et al.'07]