Graphical Models
Discrete Inference and Learning
Lecture 3

MVA

2020 – 2021

http://thoth.inrialpes.fr/~alahari/disinflearn

Slides based on material from M. Pawan Kumar

Practical matters

- Course website
  - http://thoth.inrialpes.fr/~alahari/disinflearn
  - (linked from my webpage)

- Questions ?

Project suggestions (also sent by email)

- Implement BP on trees, then graph, extend to TRW, compare
- Implement graph cut + extension (Ishikawa, other multi-label) or variation of implementation + small application
- Complex application of graph cut, requiring modelling (e.g., sequence of images)
- Geometric scene labelling with graph cuts
- Joint modelling of two labelling problems (e.g., segmentation + detection)
- Implement fast primal-dual algorithm + evaluate
- Implement deformable parts model for object detection
- …

- Or your own (but check with us first)
- **Select projects before 25th January and email us (karteek.alahari@inria.fr, guillaume.charpiat@inria.fr)**

Practical matters

- Questions ?

Recap: Lectures 1&2

- Graphical Models
  - Making **global** predictions from **local** observations
  - Learning from large quantities of data

- Two types of models studied in the class
  - Bayesian nets
  - Markov nets

Recap: Lectures 1&2

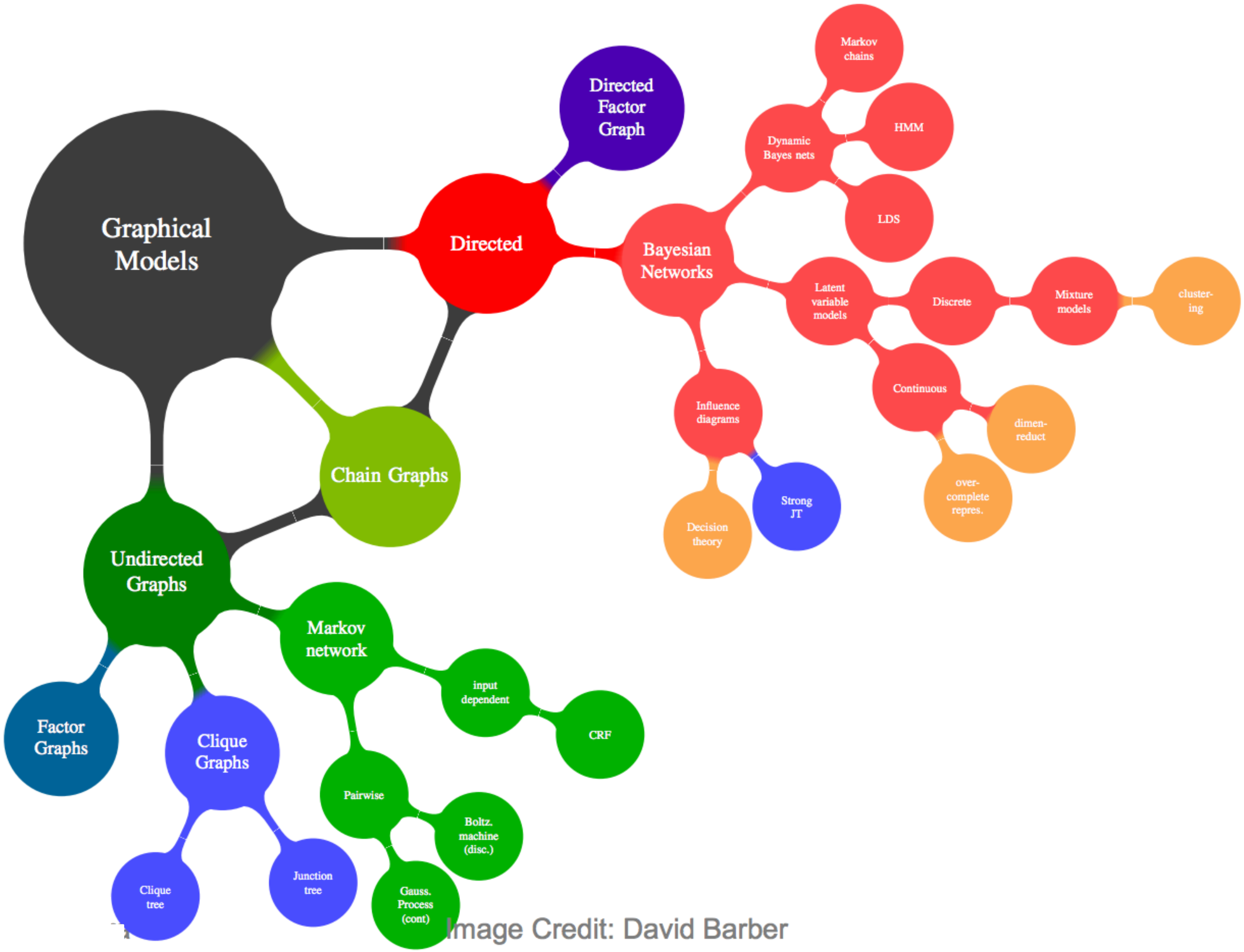


Image Credit: David Barber

Recap: Lectures 1&2

- Question: What is the core of these models?

- Question: Can you compute probabilities in Markov nets? If yes, how and if no, why?

- Question: What is the difference between Markov and Conditional random fields?

Recap: Lectures 1&2

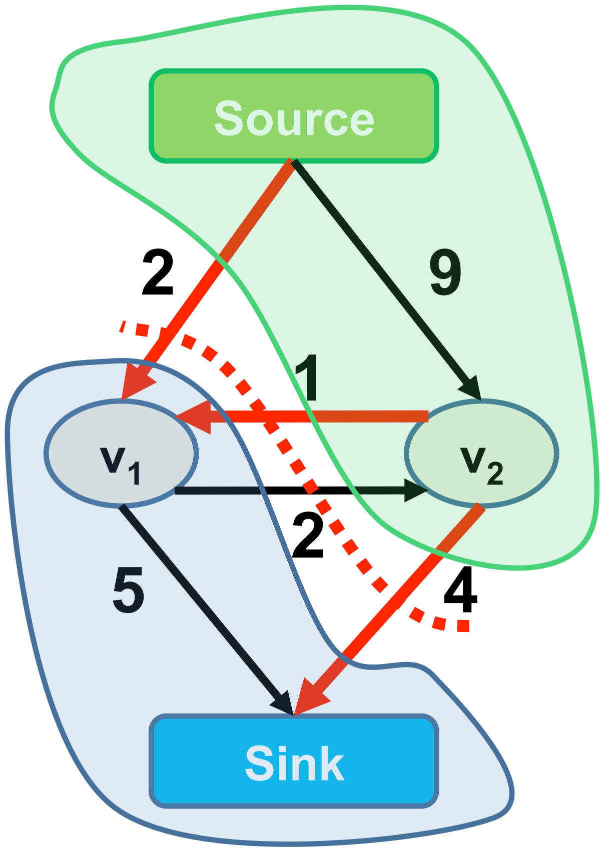
The st-mincut problem

Connection between st-mincut and energy minimization?

What problems can we solve using st-mincut?

st-mincut based Move algorithms

The st-Mincut Problem



2 + 1 + 4 = 7

What is an st-cut?

An st-cut (**S**,**T**) divides the nodes between source and sink

What is the cost of an st-cut?

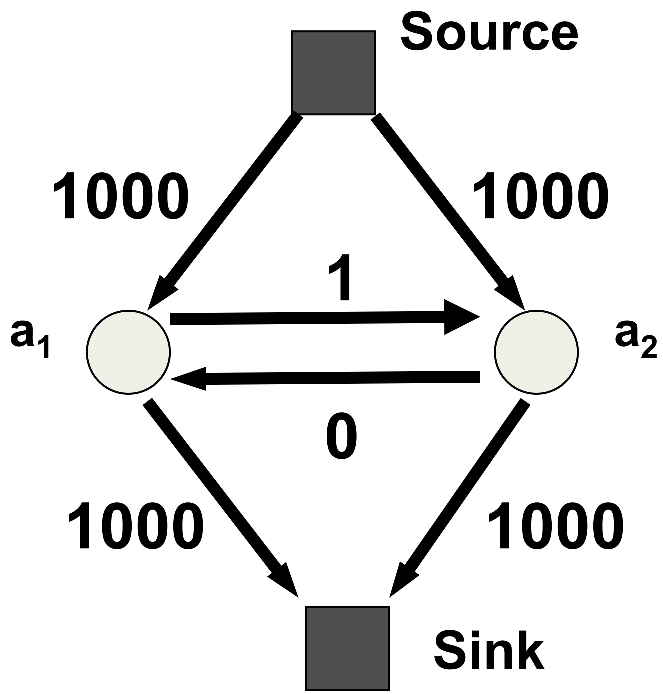
Sum of cost of all edges going from S to T

What is the st-mincut?

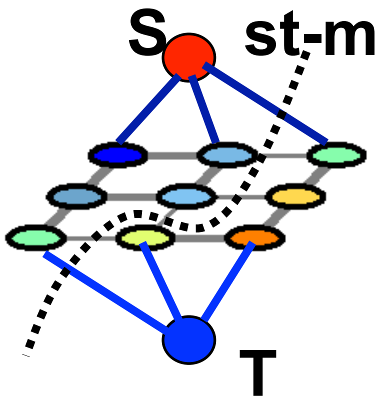
st-cut with the minimum cost

Recap: Lectures 1&2

Are all paths equally good?



st-mincut and Energy Minimization



Minimizing a Qudratic Pseudoboolean function E(x)

Functions of boolean variables

Pseudoboolean?

$$E: \{0,1\}^n \rightarrow R$$

$$E(y) = \sum_i c_i y_i + \sum_{i,j} c_{ij} y_i (1-y_j)$$

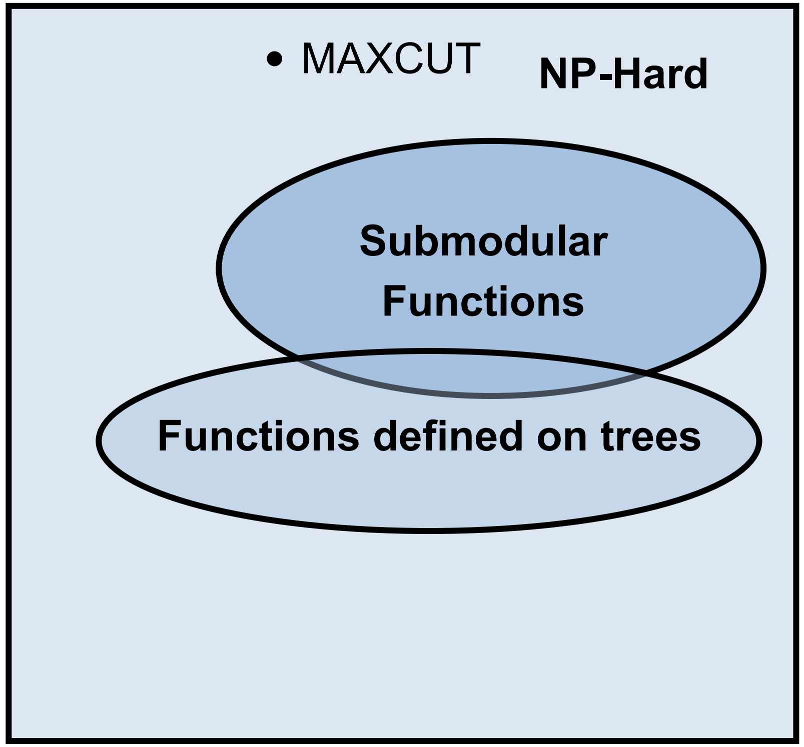
$c_{ij} \geq 0$

Polynomial time st-mincut algorithms require non-negative edge weights

Minimizing Energy Functions

- **General Energy Functions**
  - NP-hard to minimize
  - Only approximate minimization possible

- **Easy energy functions**
  - Solvable in polynomial time
  - Submodular ~ $O(n^6)$



Space of Function Minimization Problems

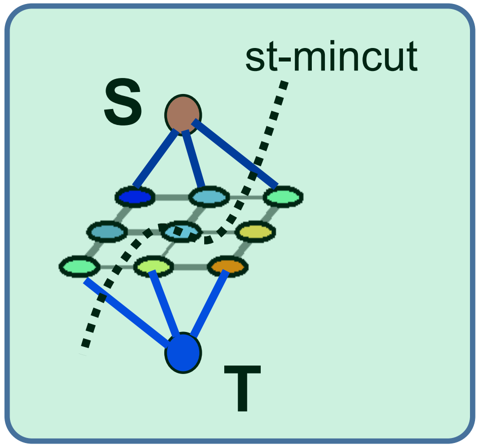
Quadratic Submodular Pseudoboolean Functions

y in $\{0,1\}^n$

$$E(y) = \sum_i \theta_i(y_i) + \sum_{i,j} \theta_{ij}(y_i,y_j)$$

For all ij $\quad \theta_{ij}(0,1) + \theta_{ij}(1,0) \geq \theta_{ij}(0,0) + \theta_{ij}(1,1)$

Equivalent (transformable)

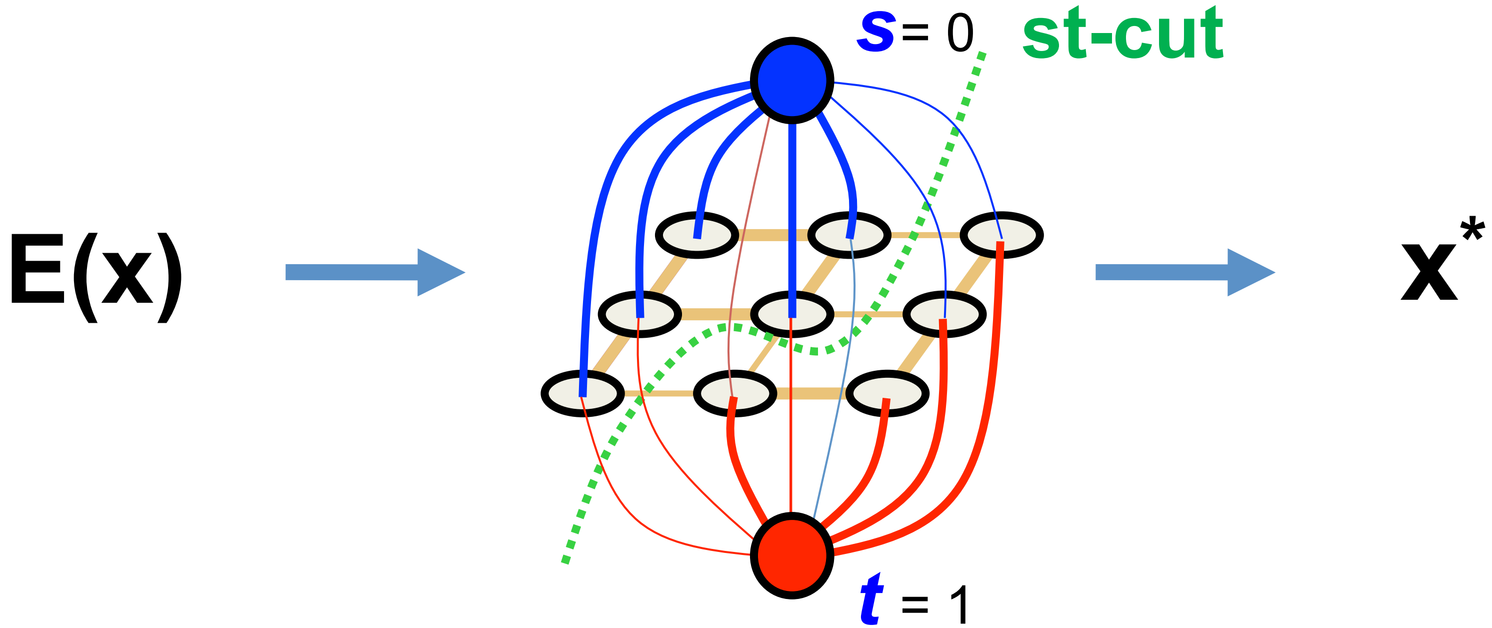


Today’s lecture

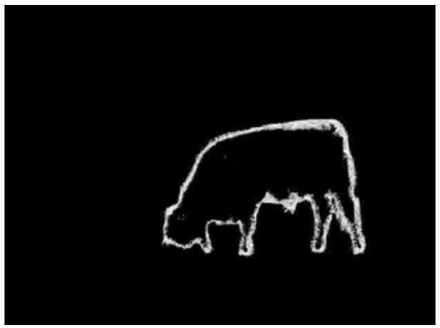
- A few more graph cut based approaches
- Belief propagation
- TRW

Dynamic Energy Minimization

Image Segmentation in Video



Image



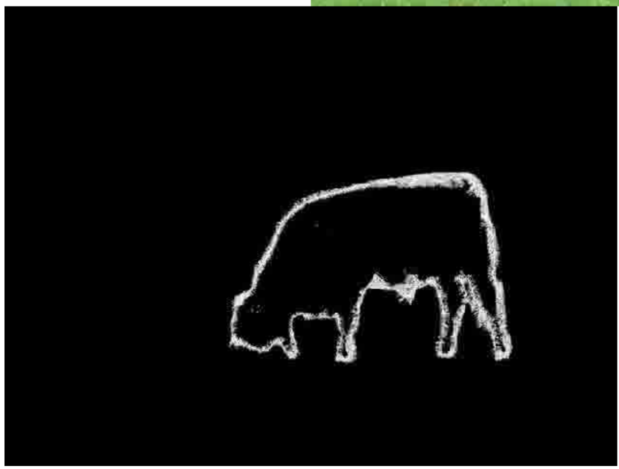
Flow

Global Optimum

Image Segmentation in Video

Image

Flow



Global Optimum

Dynamic Energy Minimization

E_A $\xrightarrow{\text{minimize}}$ S_A

Can we do better?

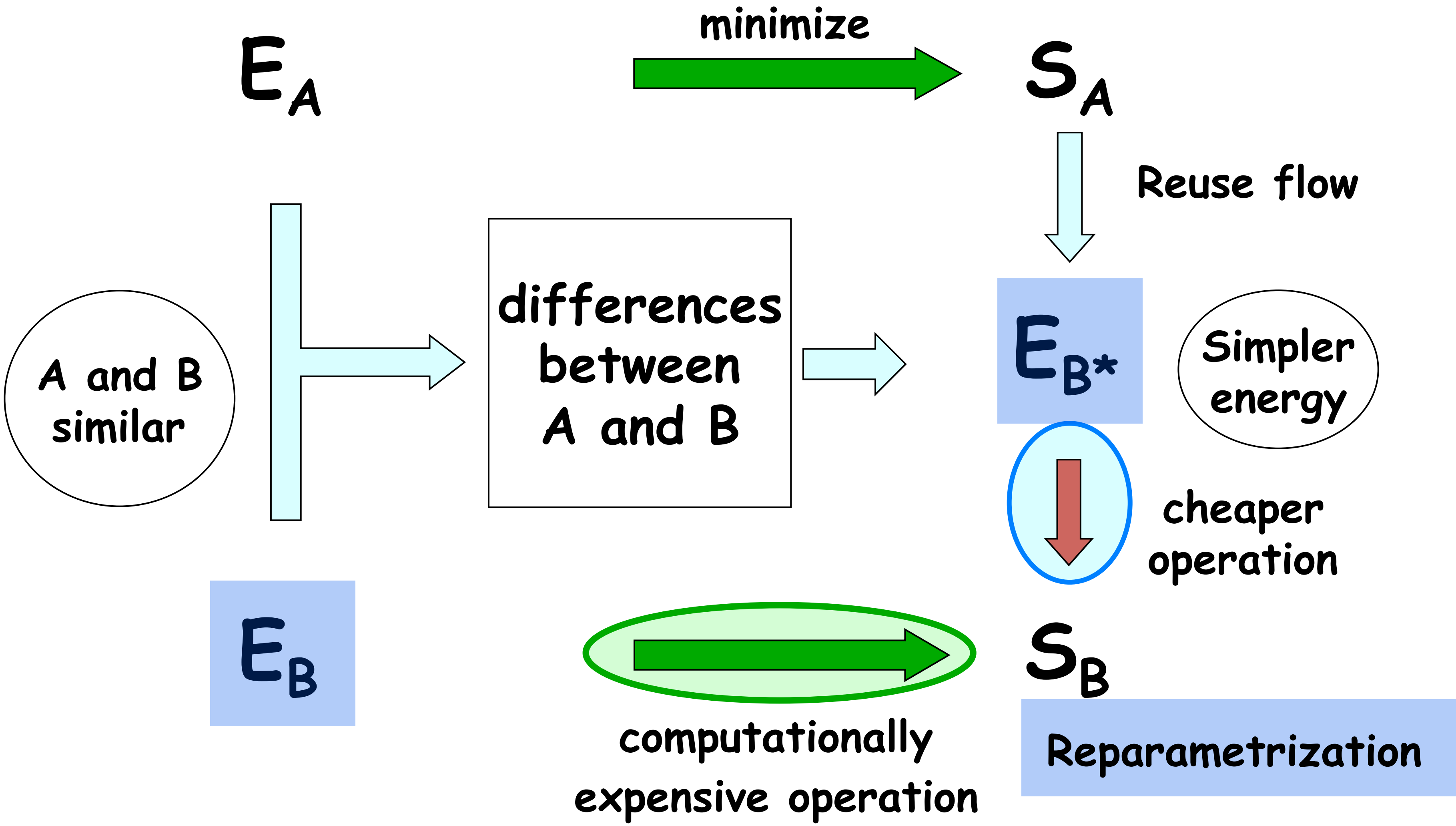
Recycling Solutions

E_B $\longrightarrow$ S_B

computationally expensive operation

Boykov & Jolly ICCV'01, Kohli & Torr (ICCV05, PAMI07)

Dynamic Energy Minimization



Boykov & Jolly ICCV'01, Kohli & Torr (ICCV05, PAMI07)

Dynamic Energy Minimization

Original Energy

$$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$

Reparameterized Energy

$$E(a_1,a_2) = 8 + \bar{a}_1 + 3a_2 + 3\bar{a}_1 a_2$$

New Energy

$$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 7a_1\bar{a}_2 + \bar{a}_1 a_2$$

New Reparameterized Energy

$$E(a_1,a_2) = 8 + \bar{a}_1 + 3a_2 + 3\bar{a}_1 a_2 + 5a_1\bar{a}_2$$

Boykov & Jolly ICCV'01, Kohli & Torr (ICCV05, PAMI07)

Transforming Multi-label Problems

Transforming problems in QBFs

Multi-label Functions $\longrightarrow$ Pseudoboolean Functions

Multi-label to Pseudo-boolean

So what is the problem?

$E_m(y_1, y_2, \ldots, y_n)$ → $E_b(x_1, x_2, \ldots, x_m)$

$y_i \in L = \{l_1, l_2, \ldots, l_k\}$

$x_i \in L = \{0,1\}$

Multi-label Problem

Binary label Problem

such that:

Let Y and X be the set of feasible solutions, then

1. For each binary solution $x \in X$ with finite energy there exists exactly one multi-label solution $y \in Y$

-> One-One encoding function $T: X \to Y$

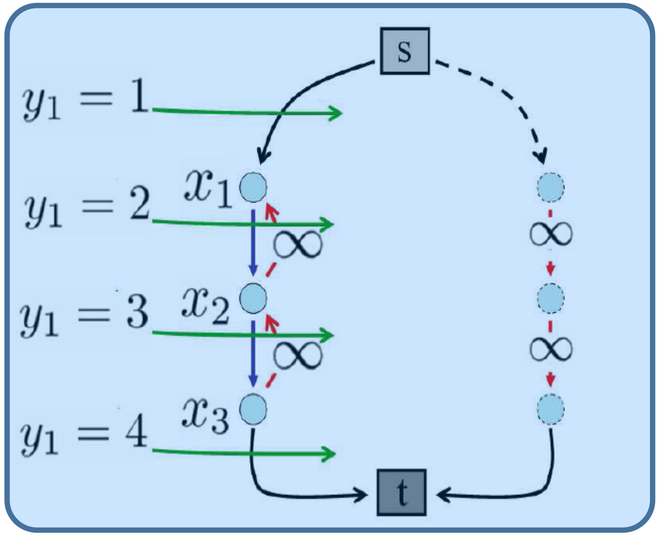
2. $\arg\min E_m(y) = T(\arg\min E_b(x))$

Multi-label to Pseudo-boolean

- **Popular encoding scheme**

[Roy and Cox '98, Ishikawa '03, Schlesinger & Flach '06]

$$\{y_1 = 1\} \leftrightarrow \{x_1 = 1, x_2 = 1, x_3 = 1\},$$
$$\{y_1 = 2\} \leftrightarrow \{x_1 = 0, x_2 = 1, x_3 = 1\},$$
$$\{y_1 = 3\} \leftrightarrow \{x_1 = 0, x_2 = 0, x_3 = 1\},$$
$$\{y_1 = 4\} \leftrightarrow \{x_1 = 0, x_2 = 0, x_3 = 0\}.$$



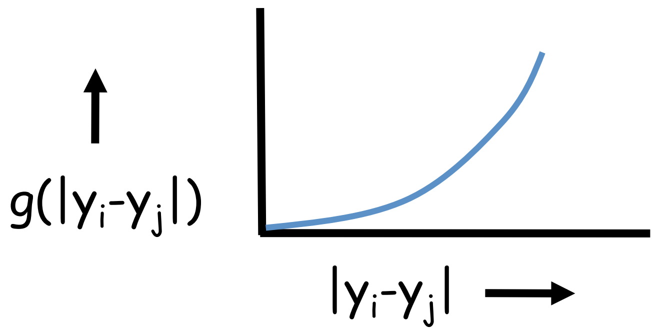
Ishikawa's result:

$$E(\mathbf{y}) = \sum_i \theta_i(y_i) + \sum_{i,j} \theta_{ij}(y_i, y_j)$$

$y \in$ Labels $L = \{l_1, l_2, \ldots, l_k\}$

$$\theta_{ij}(y_i, y_j) = \boxed{g(|y_i - y_j|)}$$

→ Convex Function



Belief Propagation

A Computer Vision Application

Binary Image Segmentation

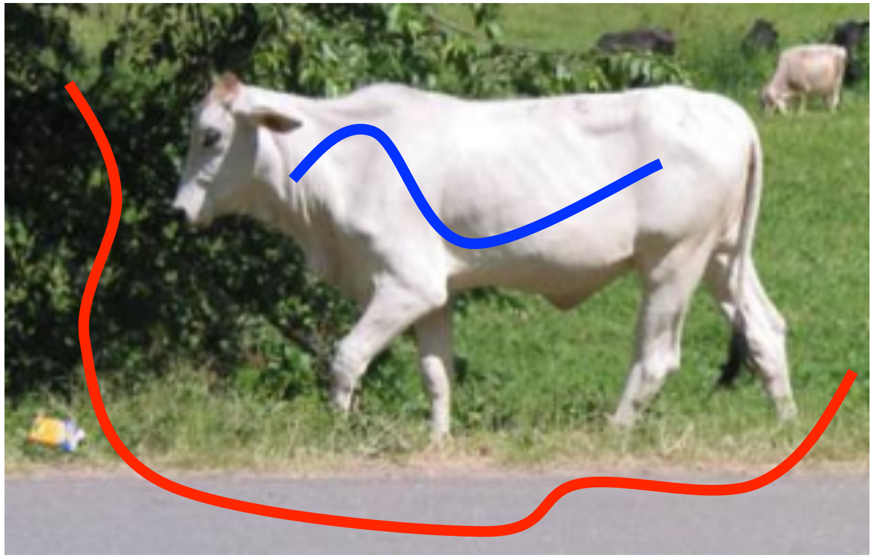
How ?

Cost function   Models *our* knowledge about natural images

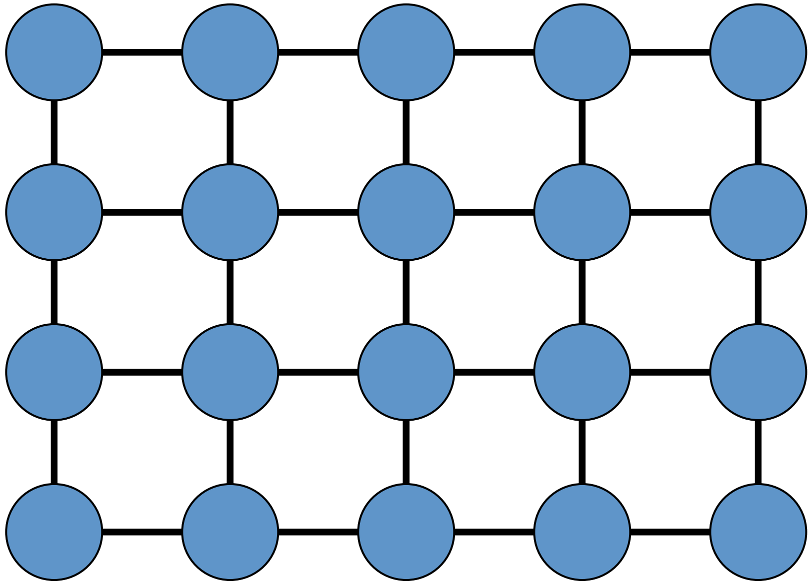
Optimize cost function to obtain the segmentation

A Computer Vision Application

Binary Image Segmentation



Object - white, Background - green/grey



Graph G = (V,E)

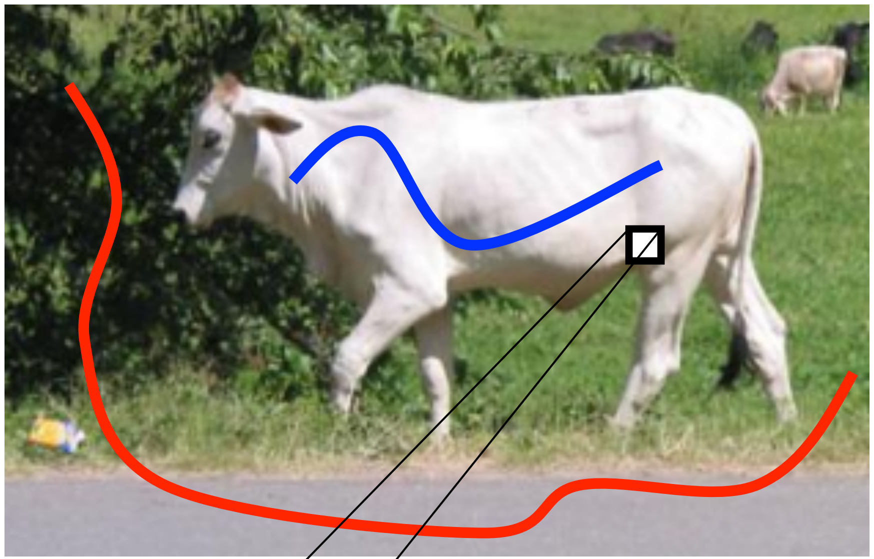
Each vertex corresponds to a pixel

Edges define a 4-neighbourhood *grid* graph

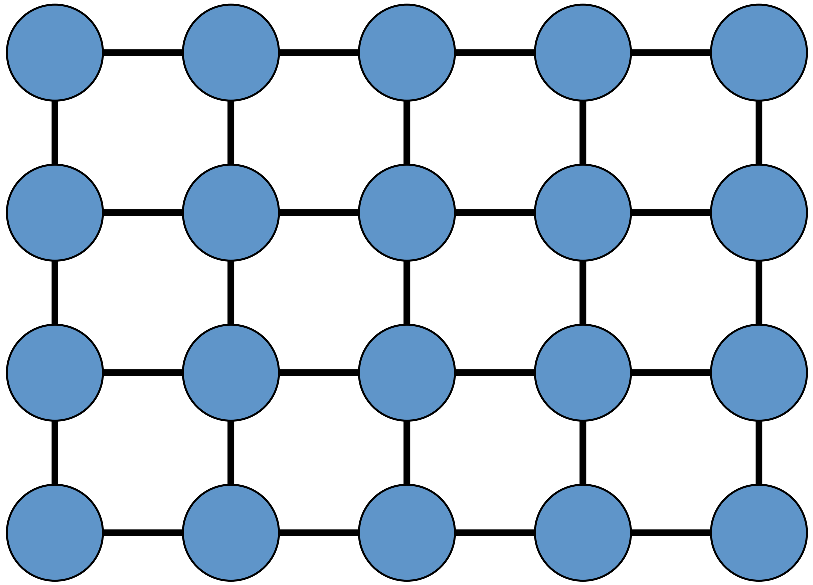
Assign a label to each vertex from L = {obj,bkg}

A Computer Vision Application

Binary Image Segmentation



Object - white, Background - green/grey



Graph G = (V,E)

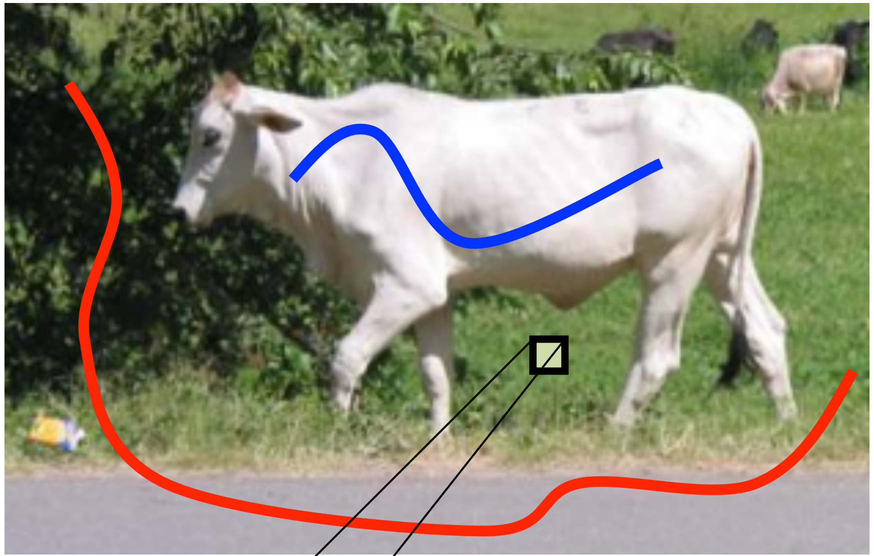
Cost of a labelling f : V ➔ L

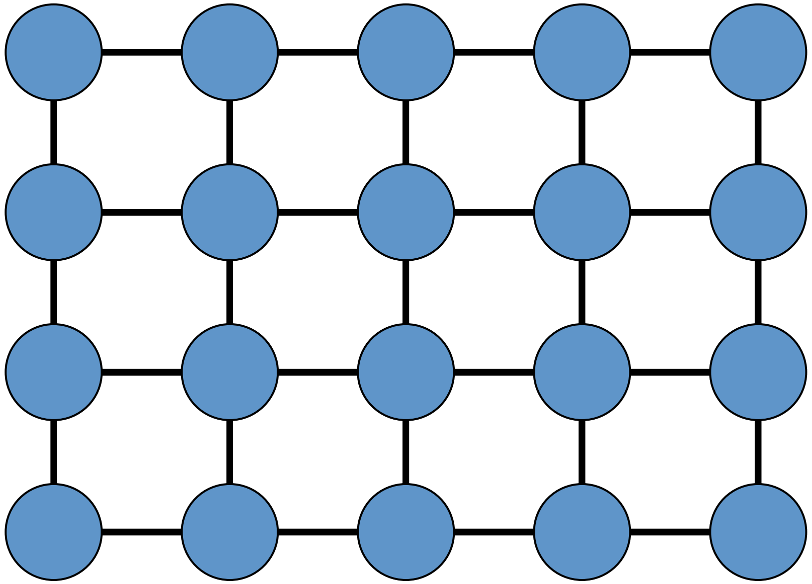
Per Vertex Cost

Cost of label 'obj' low Cost of label 'bkg' high

A Computer Vision Application

Binary Image Segmentation



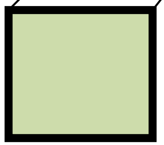


Object - white, Background - green/grey

Graph G = (V,E)

Cost of a labelling f : V ➔ L

Per Vertex Cost

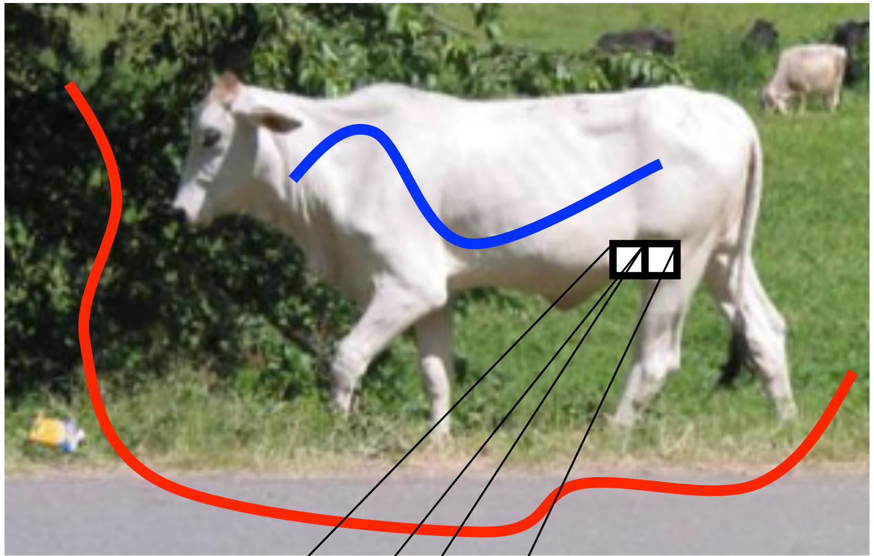


Cost of label 'obj' high Cost of label 'bkg' low

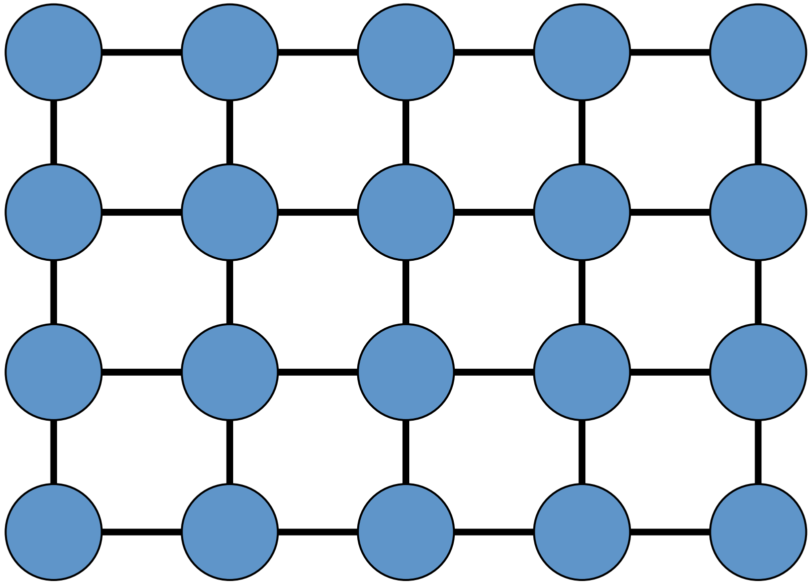
UNARY COST

A Computer Vision Application

Binary Image Segmentation



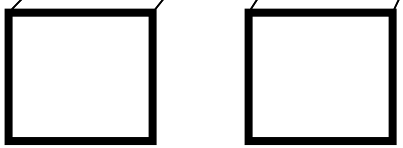
Object - white, Background - green/grey



Graph G = (V,E)

Cost of a labelling f : V ➔ L

Per Edge Cost

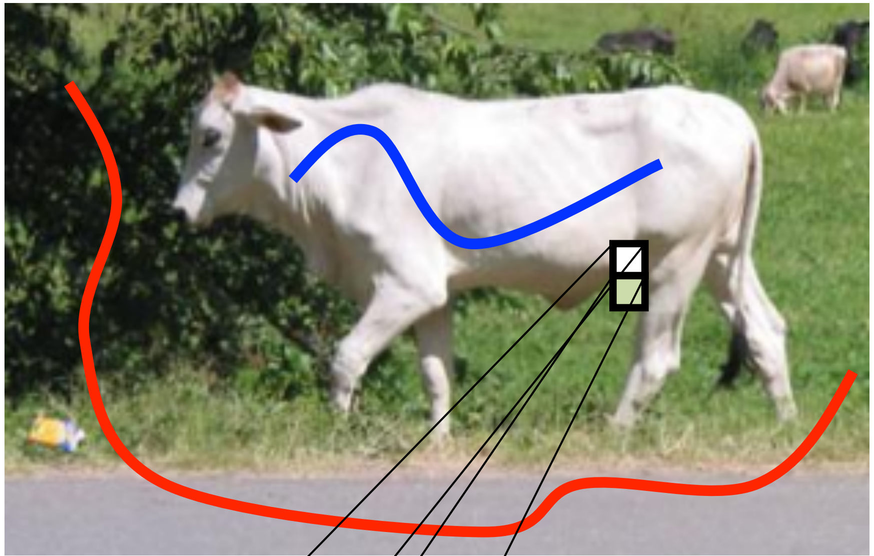


Cost of same label low

Cost of different labels high

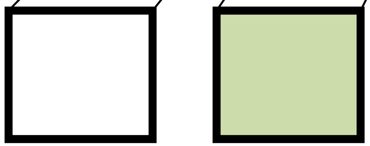
A Computer Vision Application

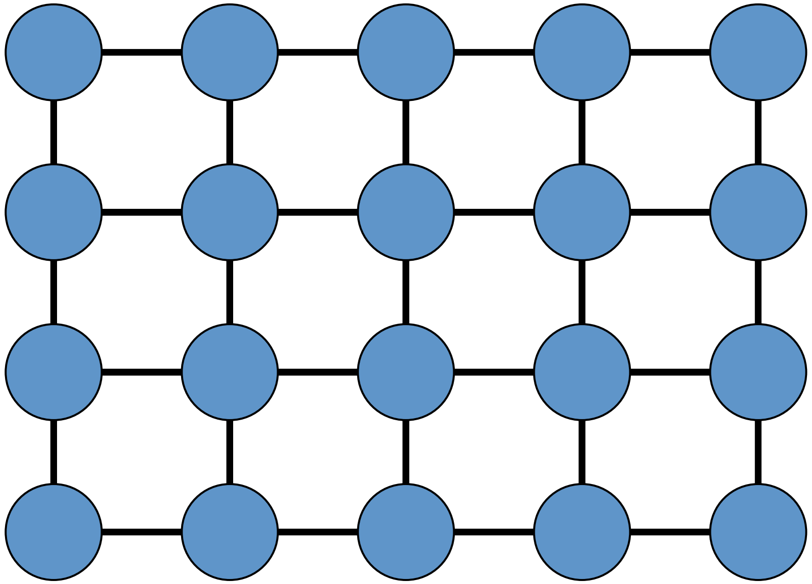
Binary Image Segmentation



Object - white, Background - green/grey

Cost of a labelling f : V ➔ L





Graph G = (V,E)

Per Edge Cost

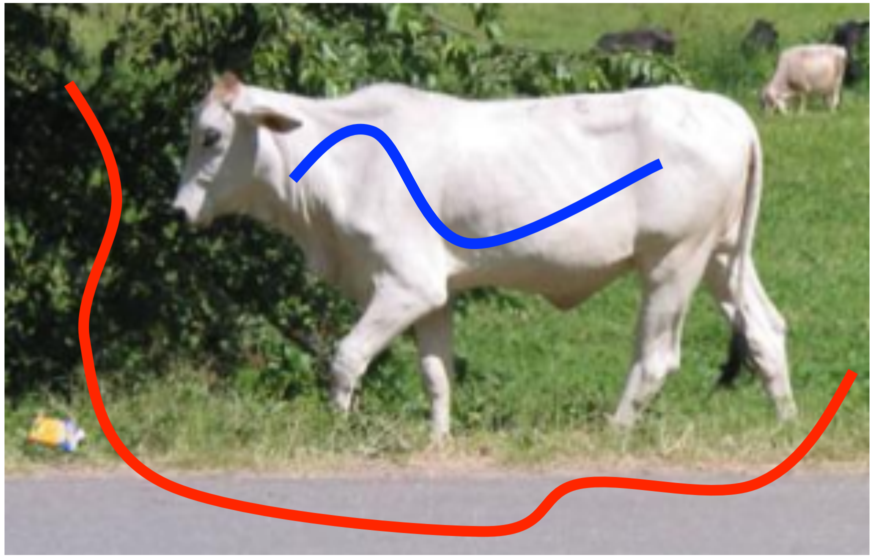
Cost of same label high

Cost of different labels low

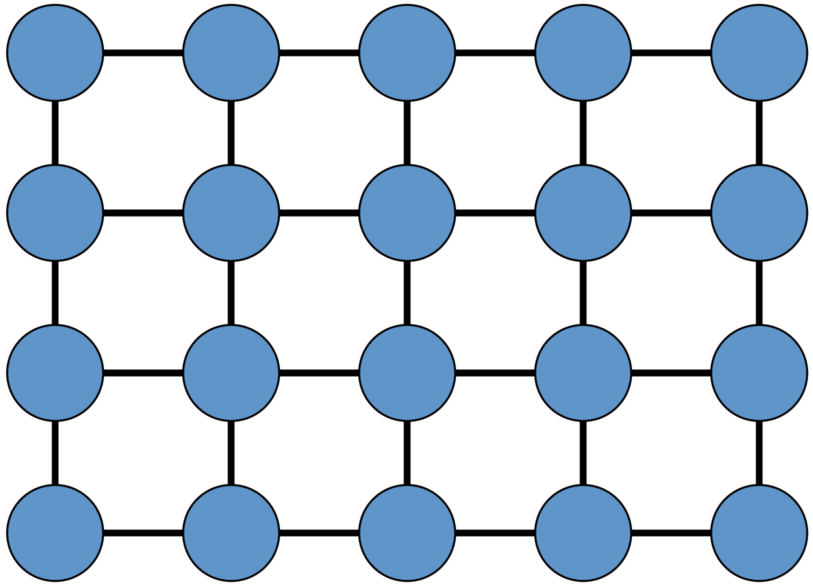
PAIRWISE COST

A Computer Vision Application

Binary Image Segmentation



Object - white, Background - green/grey

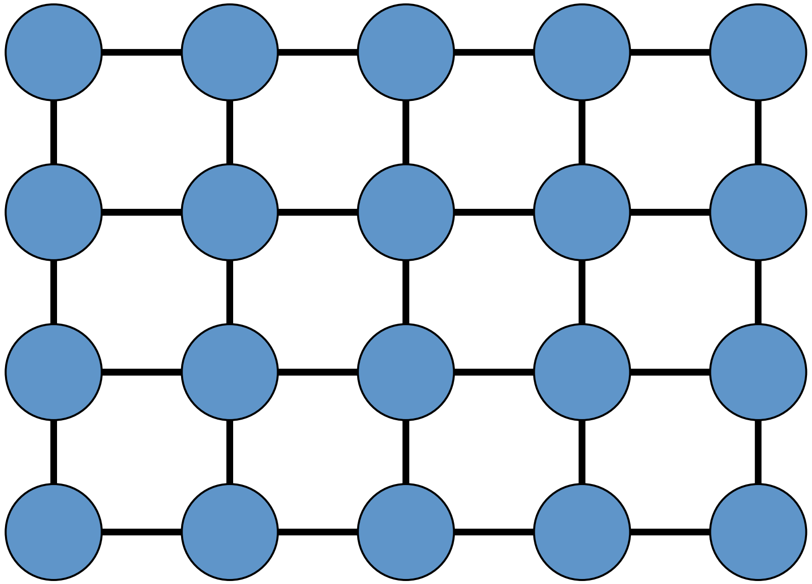


Graph G = (V,E)

Problem: Find the labelling with minimum cost f*

A Computer Vision Application

Binary Image Segmentation

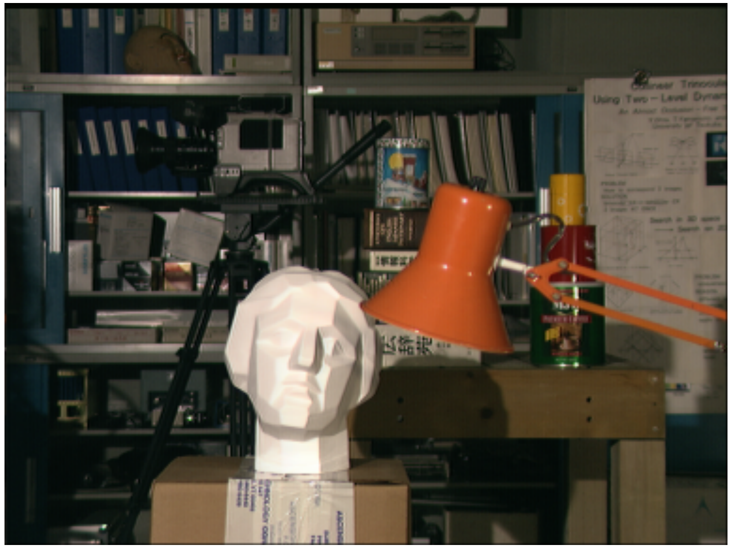


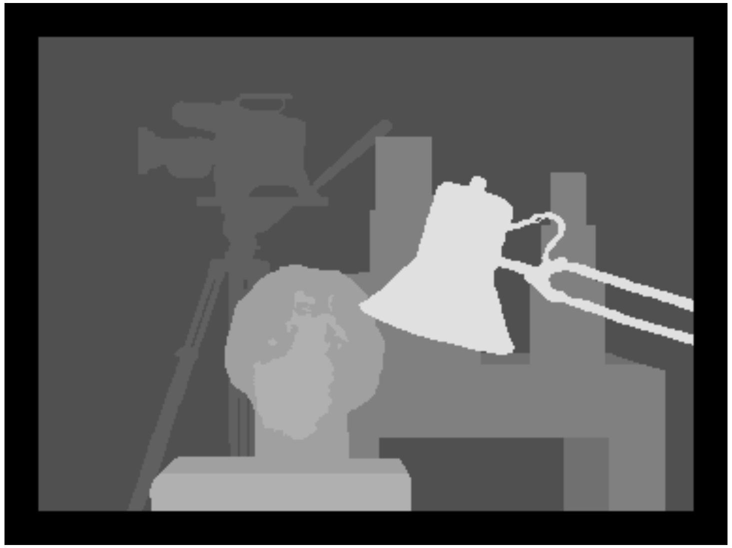
Graph G = (V,E)

Problem: Find the labelling with minimum cost f*

Another Computer Vision Application

Stereo Correspondence





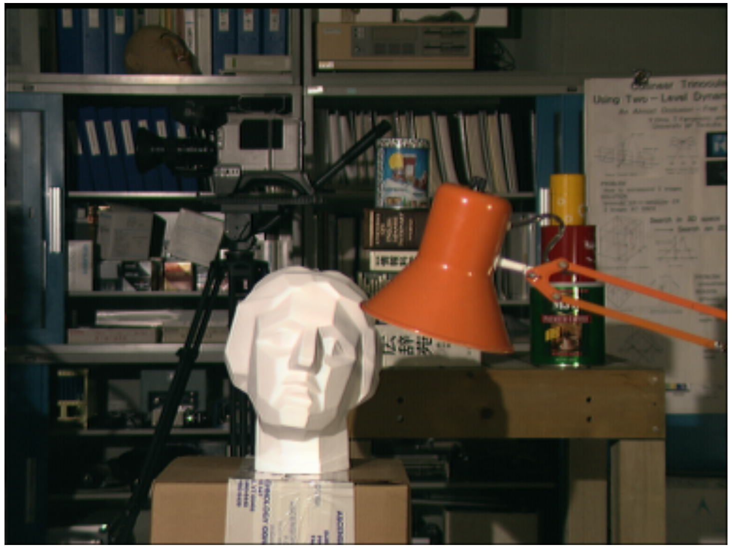
Disparity Map

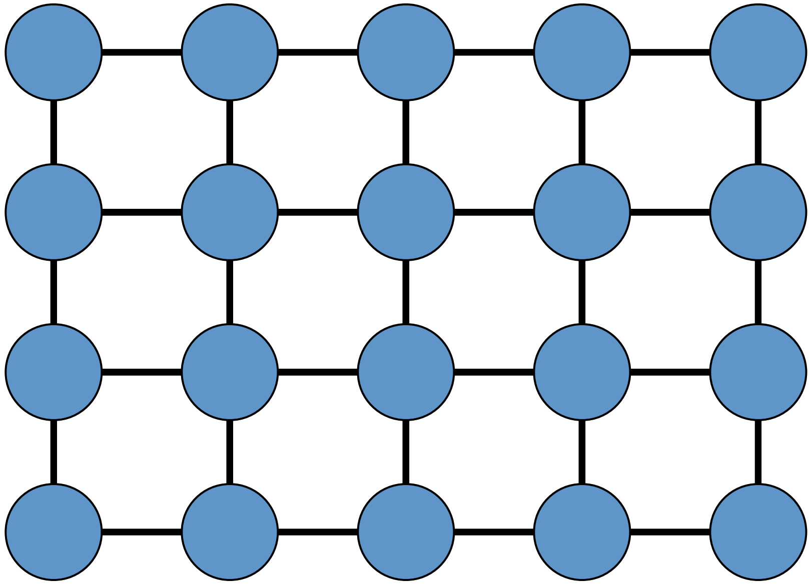
How ?

Minimizing a cost function

Another Computer Vision Application

Stereo Correspondence





Graph G = (V,E)

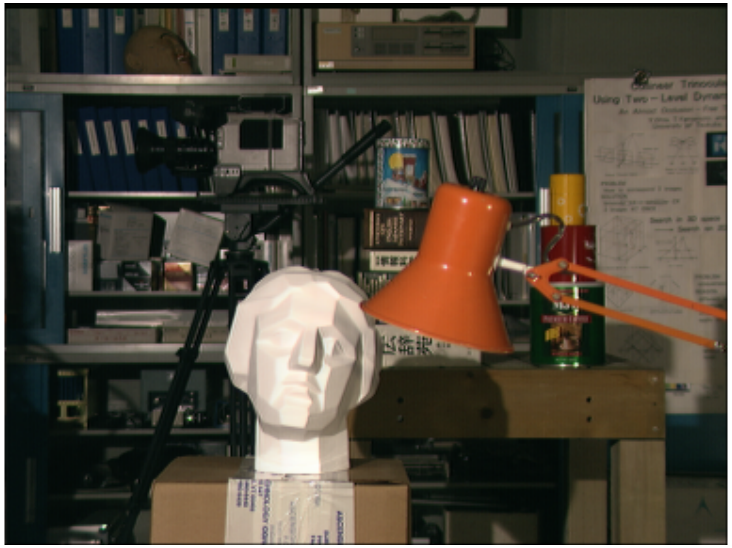
Vertex corresponds to a pixel

Edges define grid graph

L = {disparities}

Another Computer Vision Application

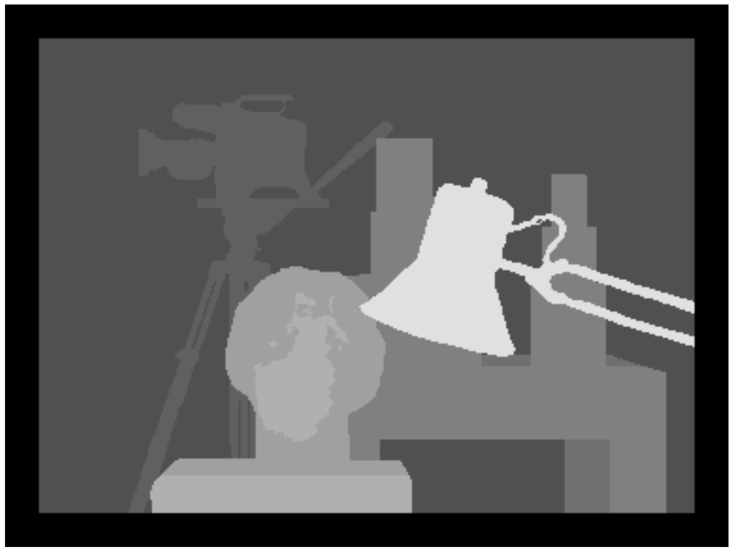
Stereo Correspondence



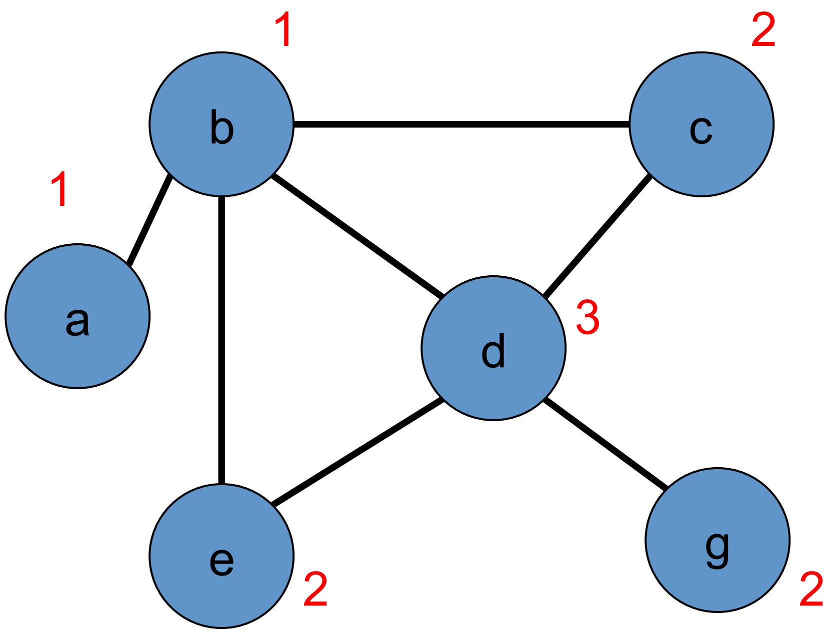
Cost of labelling f :

Unary cost + Pairwise Cost

Find minimum cost f*



The General Problem



Graph G = (V, E)

Discrete label set L = {1,2,…,h}

Assign a label to each vertex
f: V ➜ L

Cost of a labelling Q(f)

Unary Cost   Pairwise Cost

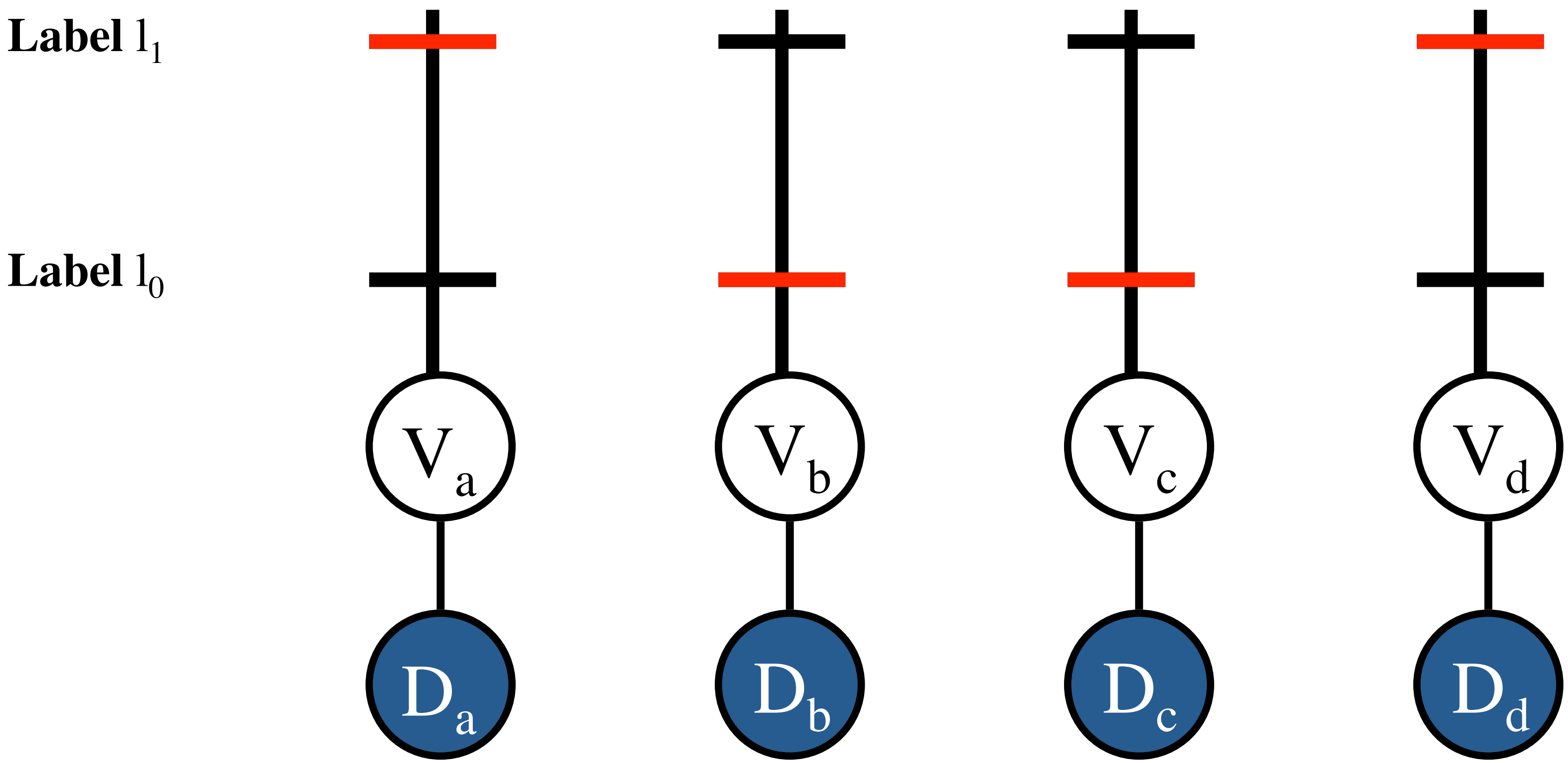
Find f* = arg min Q(f)

Overview

- Basics: problem formulation
  - Energy Function
  - MAP Estimation
  - Computing min-marginals
  - Reparameterization

- Solutions
  - Belief Propagation and related methods [Lecture 3]
  - Graph cuts [Lecture 2]

Energy Function

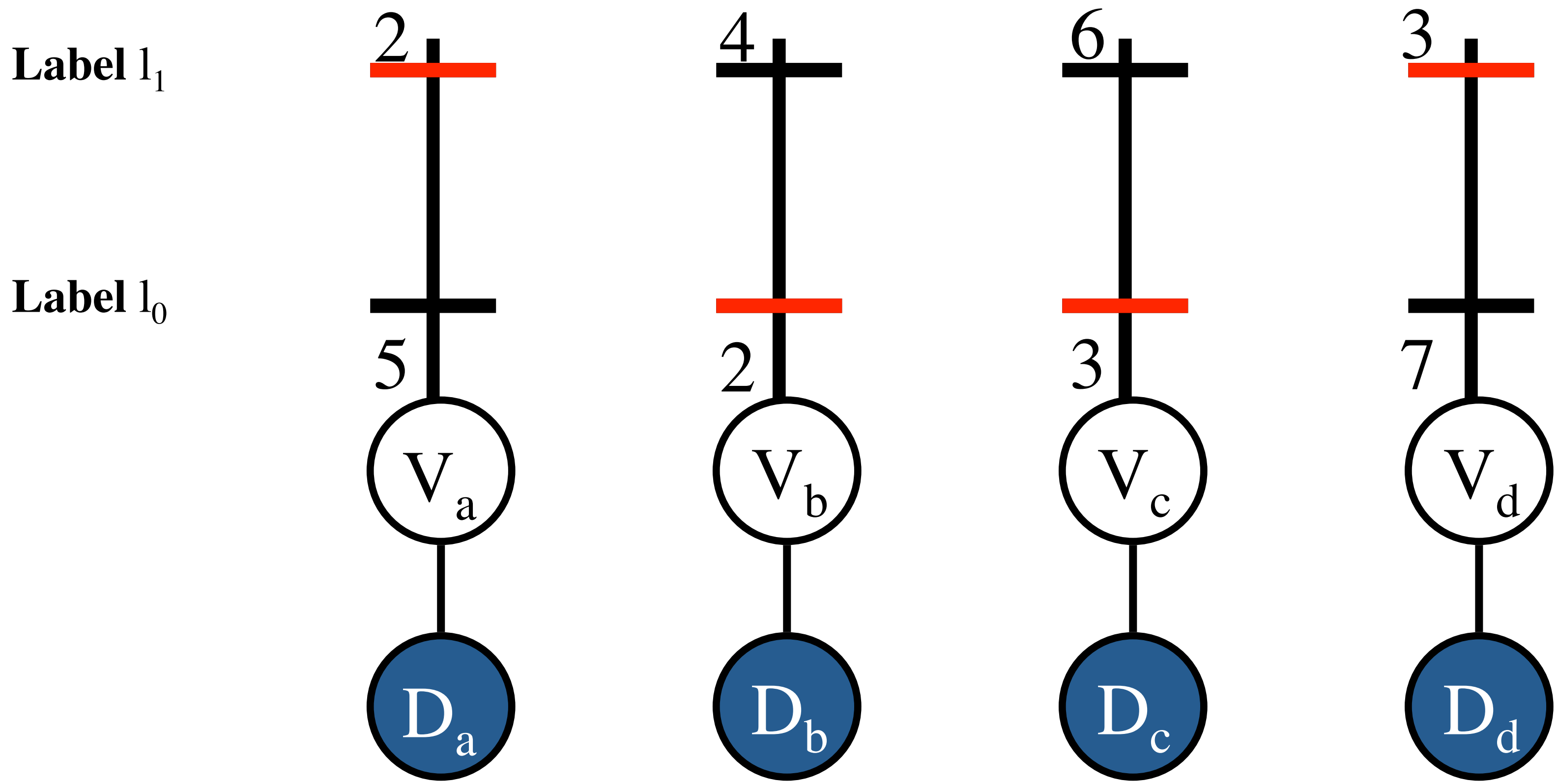


Random Variables $V = \{V_a, V_b, \ldots.\}$

Labels $L = \{l_0, l_1, \ldots.\}$ Data D

Labelling f: $\{a, b, \ldots. \} \rightarrow \{0,1, \ldots\}$

Energy Function



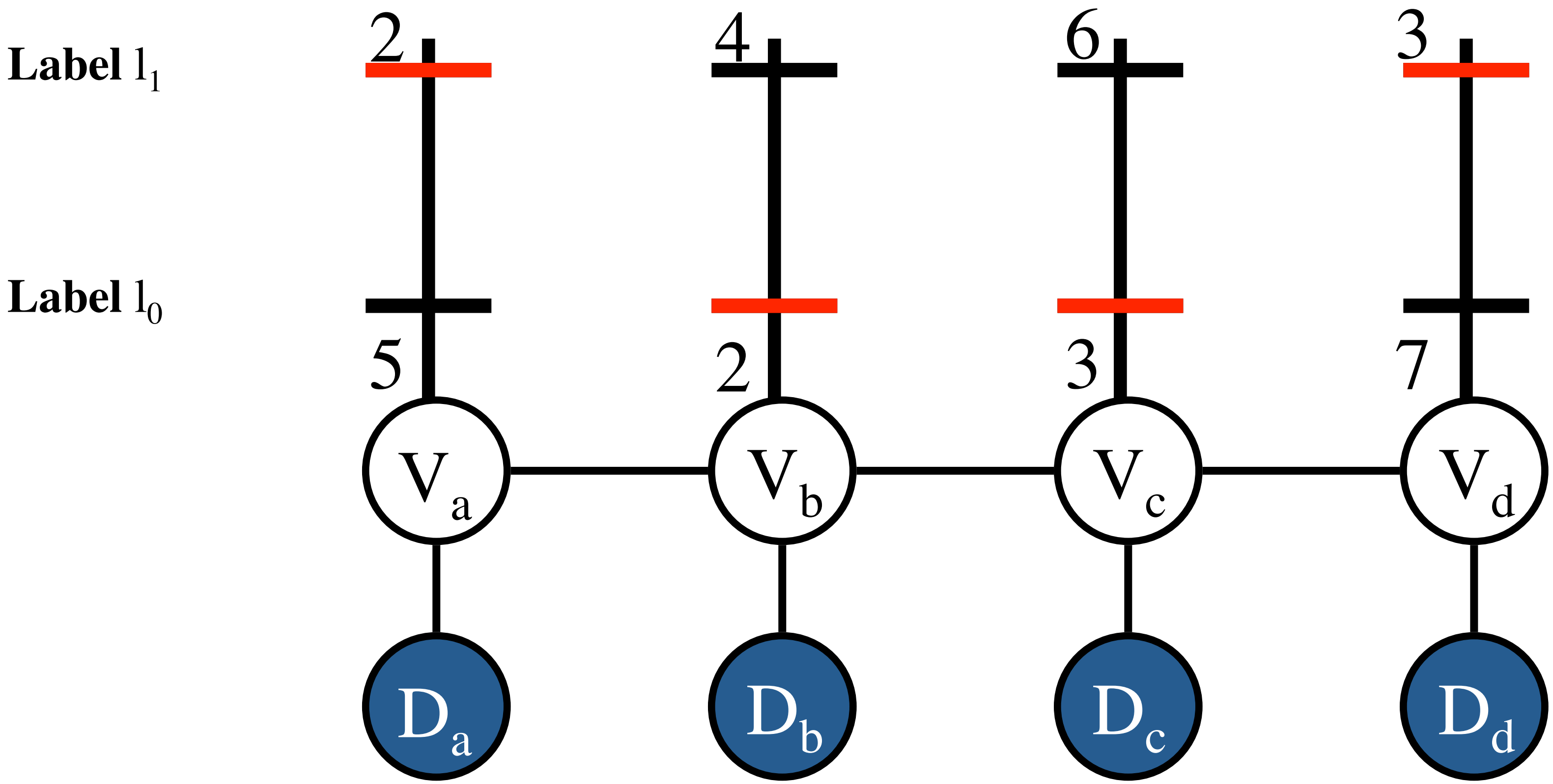
$$Q(f) \;=\; \sum_a \theta_{a;f(a)}$$

Unary Potential

Easy to minimize

Neighbourhood

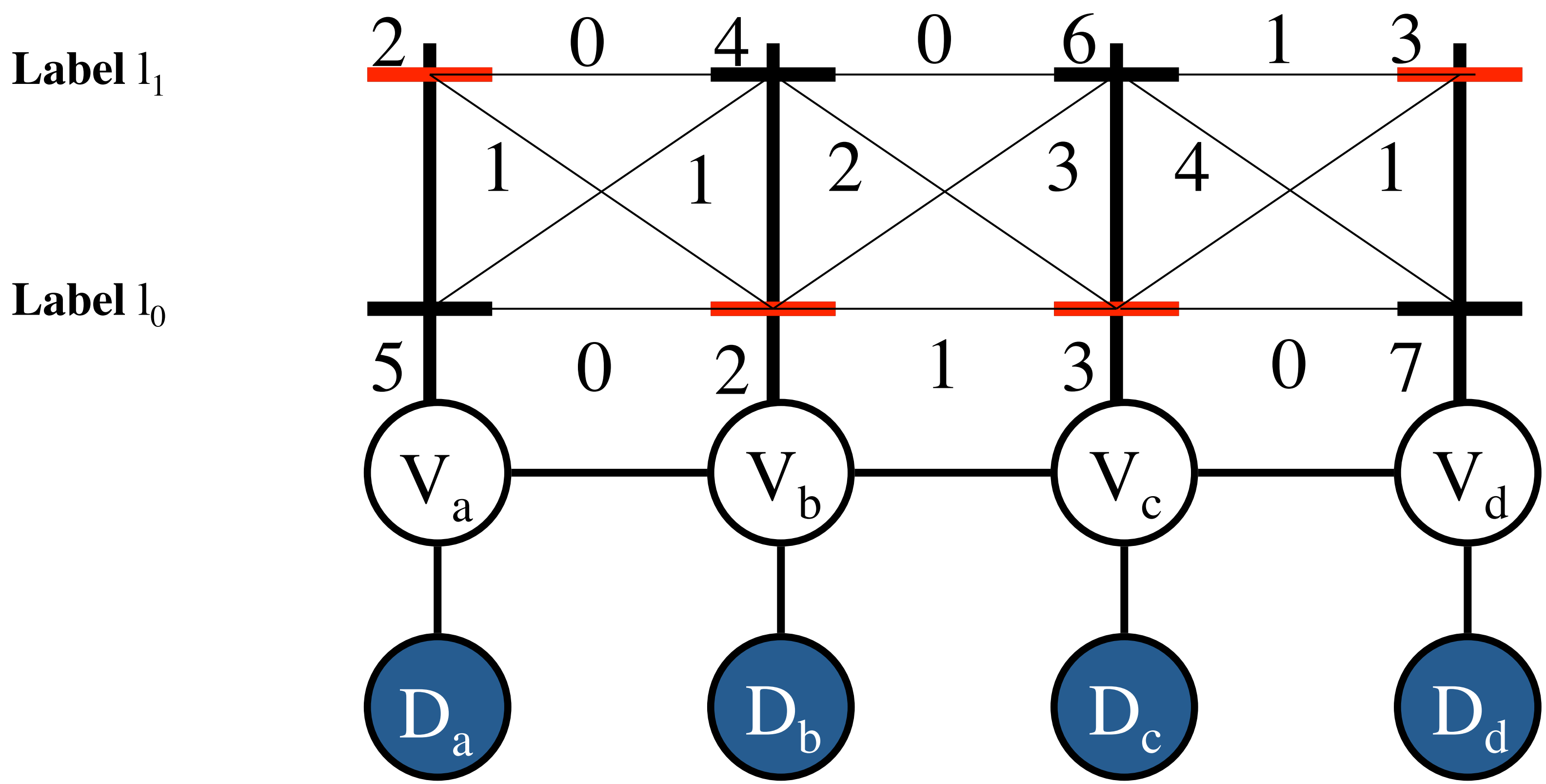
Energy Function



E : (a,b) $\in$ E iff V_a and V_b are neighbours

E = { (a,b) , (b,c) , (c,d) }

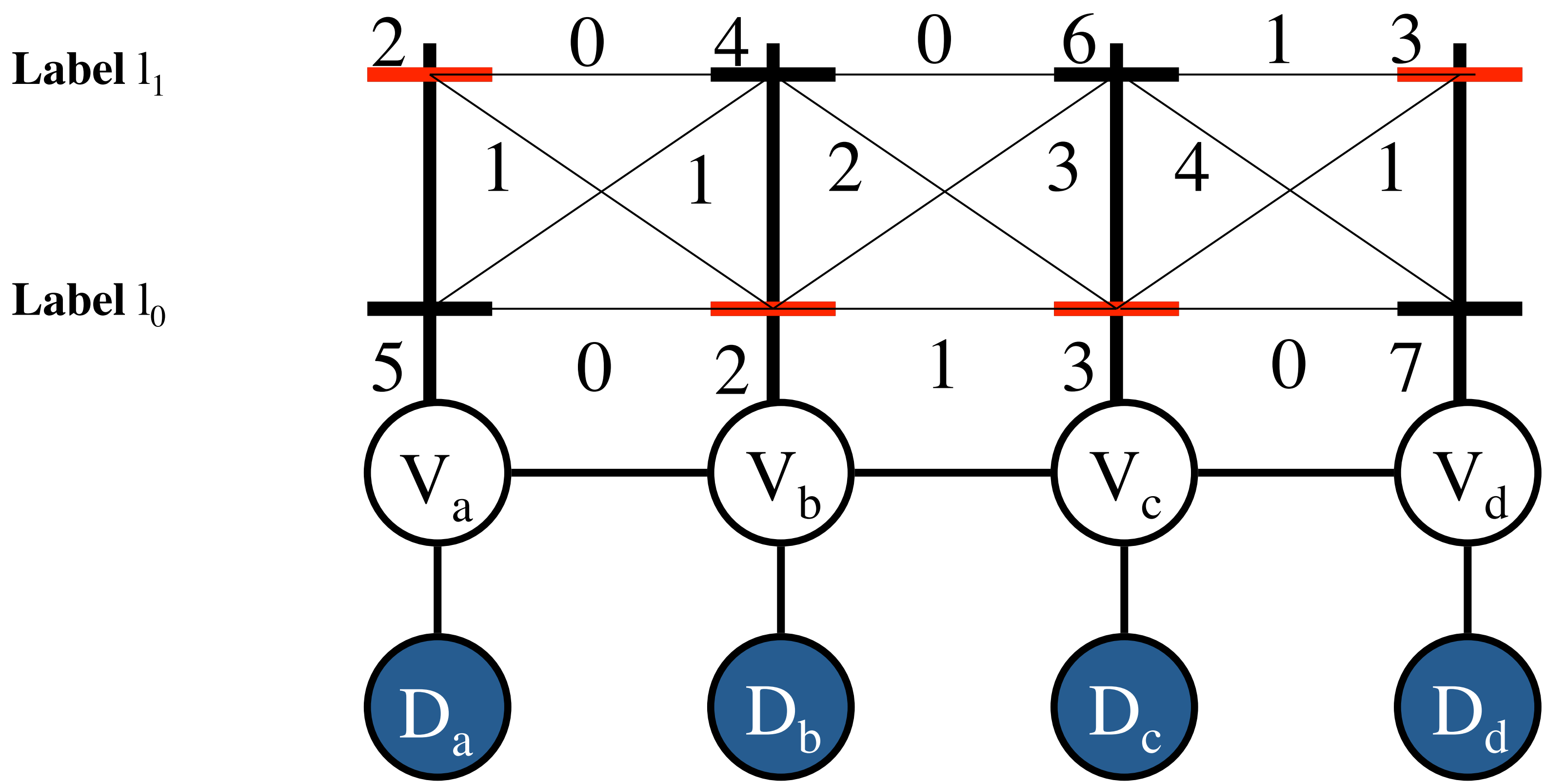
Energy Function



Pairwise Potential

$$Q(f) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

Energy Function



$$Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

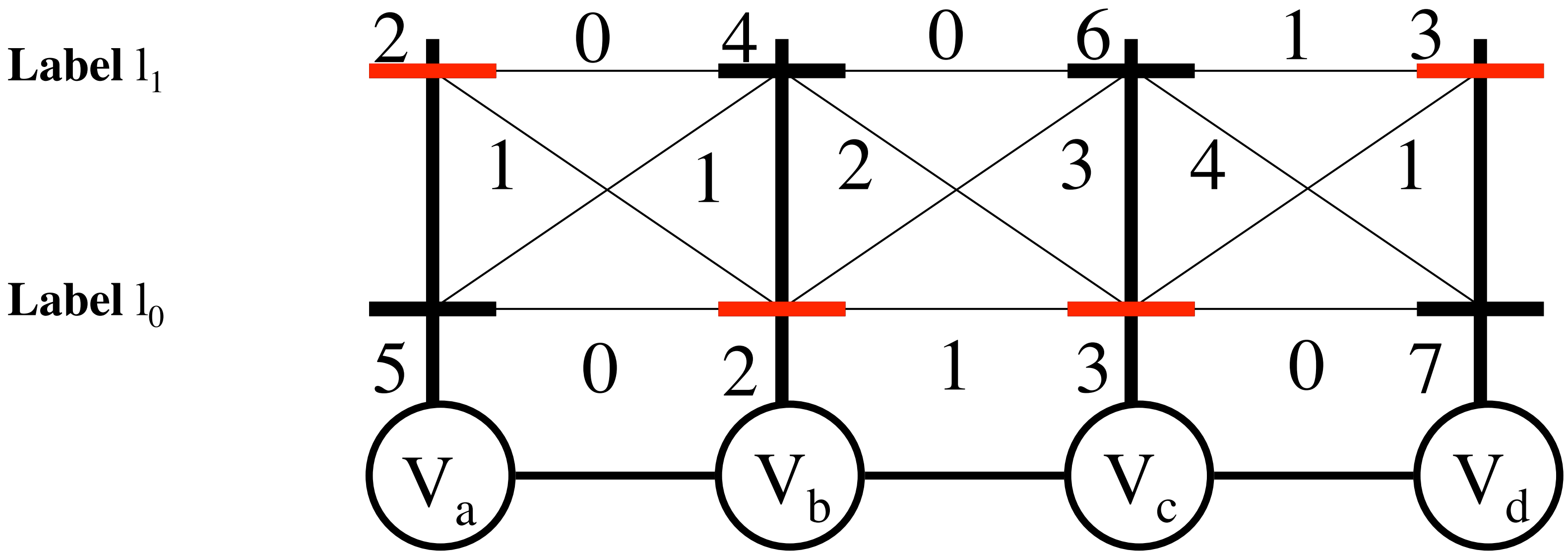
Parameter

Overview

- Basics: problem formulation
  - Energy Function
  - MAP Estimation
  - Computing min-marginals
  - Reparameterization

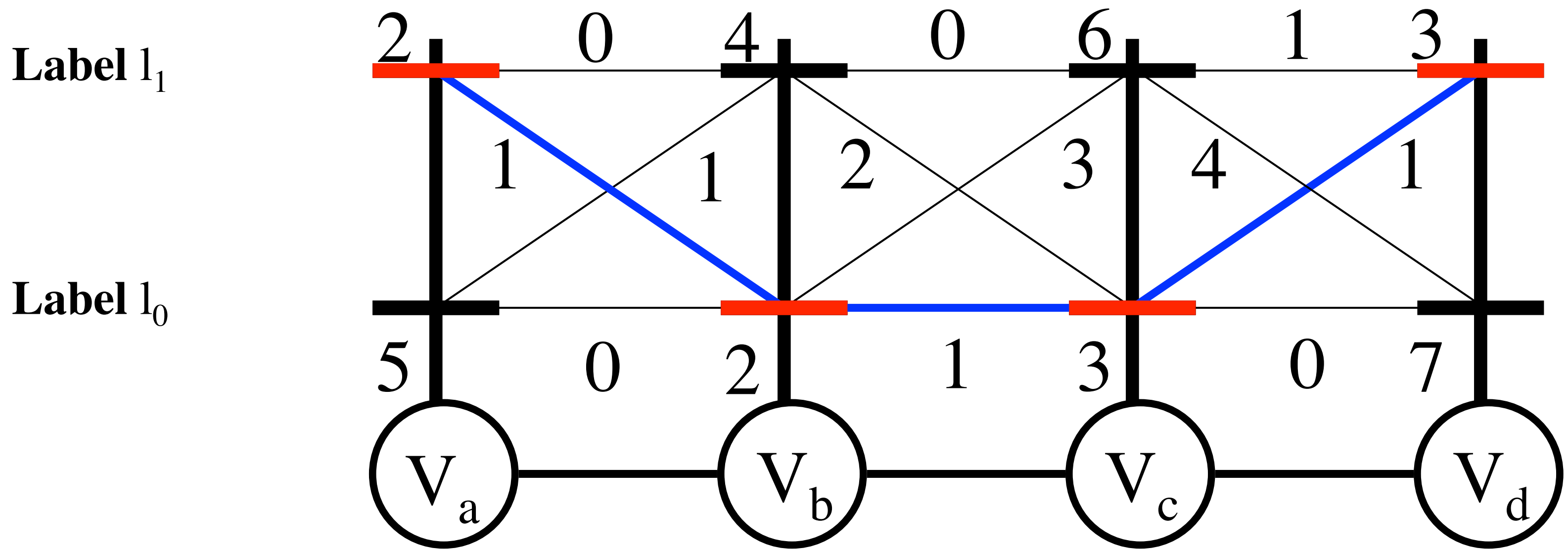
- Solutions
  - Belief Propagation and related methods [Lecture 3]
  - Graph cuts [Lecture 2]

MAP Estimation



$$Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

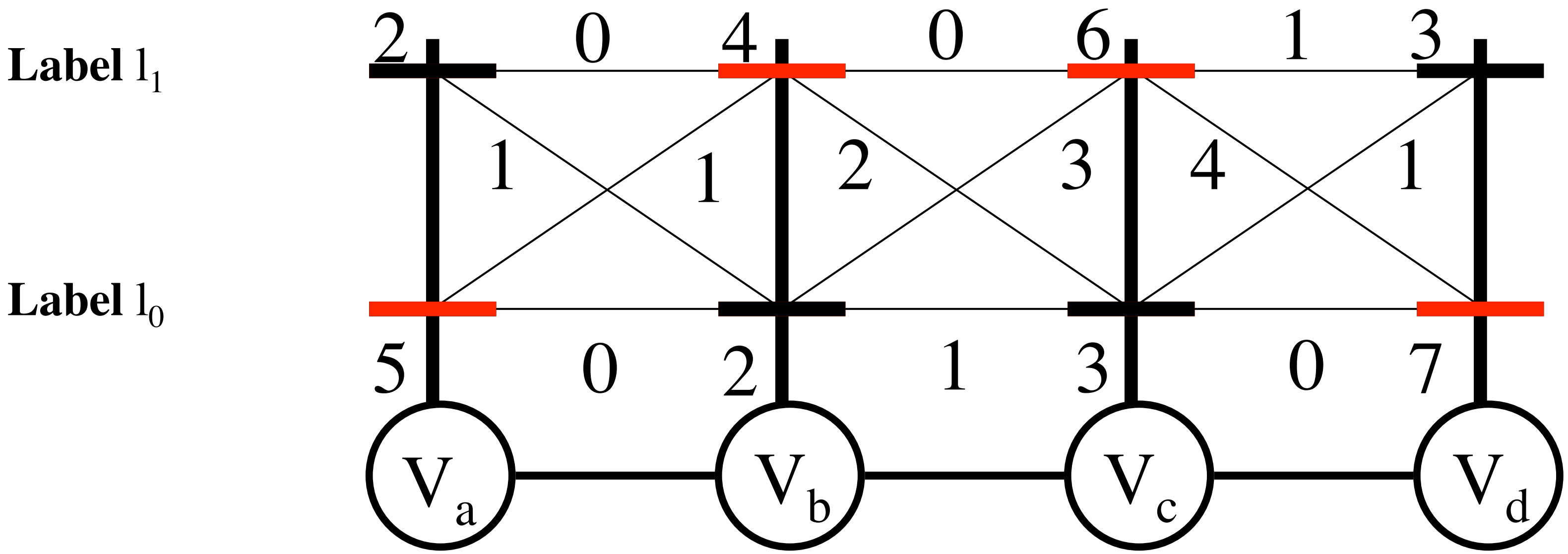
MAP Estimation



$$Q(f;\theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

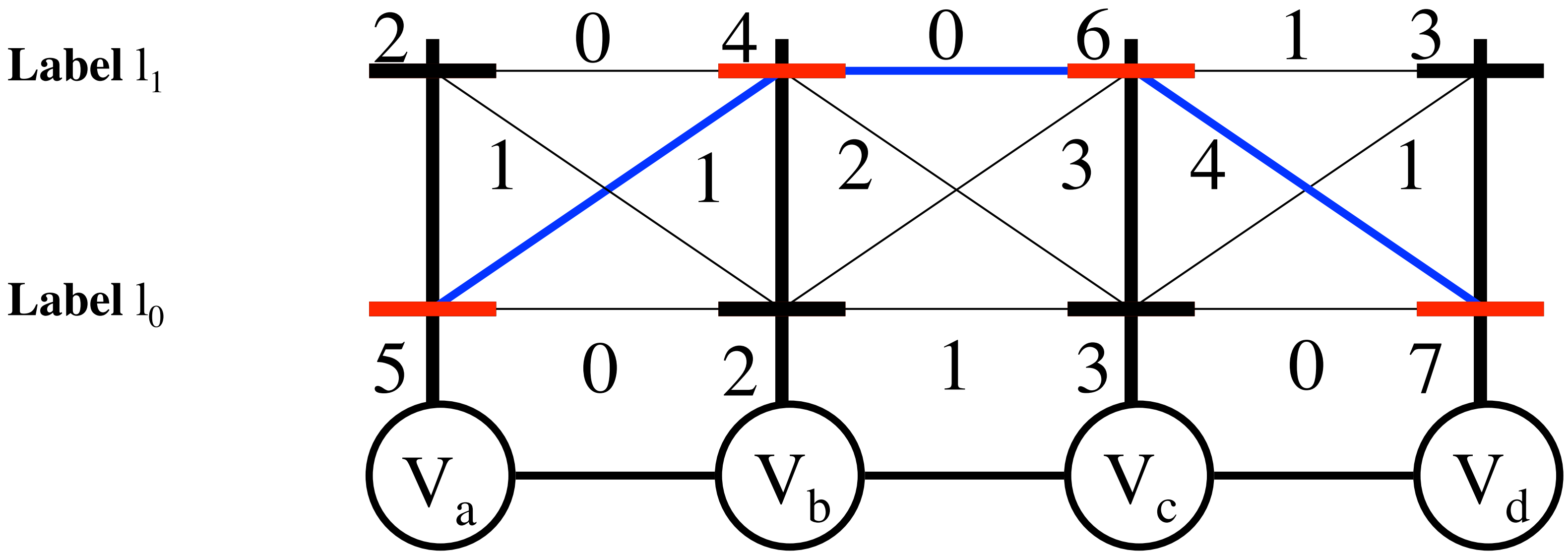
2 + 1 + 2 + 1 + 3 + 1 + 3 = 13

MAP Estimation



$$Q(f;\theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

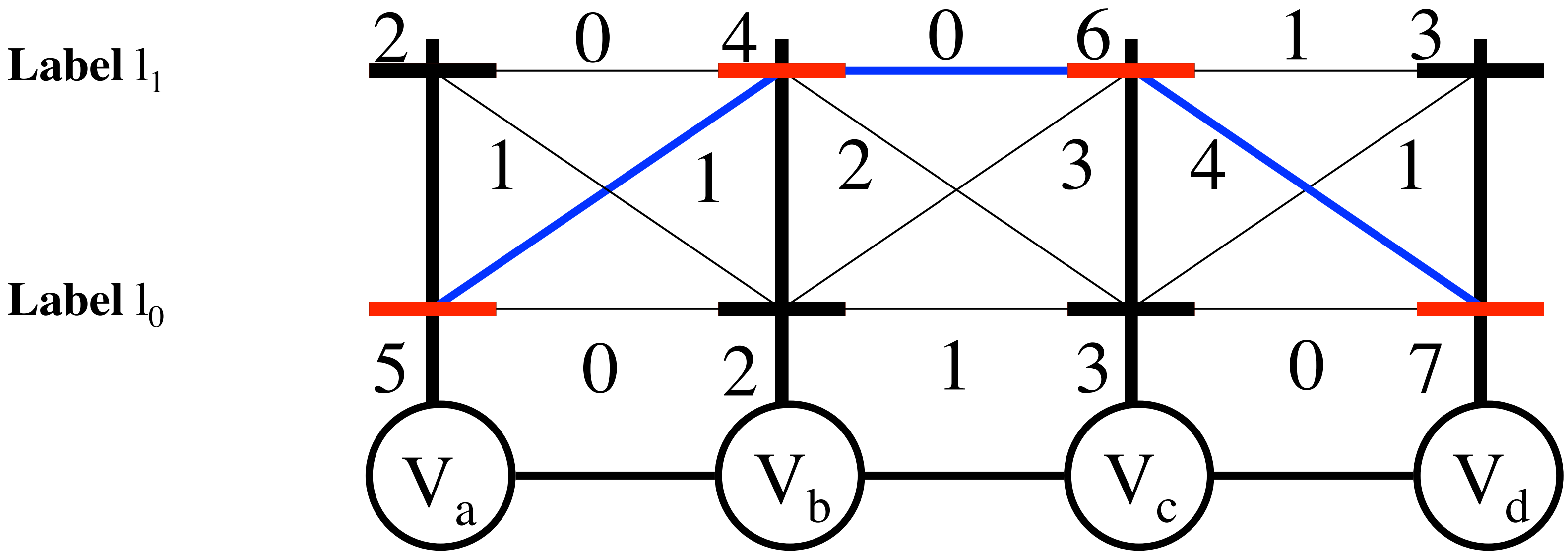
MAP Estimation



$$Q(f;\theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

5 + 1 + 4 + 0 + 6 + 4 + 7 = 27

MAP Estimation



$$q^* = \min Q(f; \theta) = Q(f^*; \theta)$$

$$Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

$$f^* = \arg\min Q(f; \theta)$$

Equivalent to maximizing the associated probability

MAP Estimation

16 possible labellings

$f^* = \{1, 0, 0, 1\}$

$q^* = 13$

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
0	0	0	0	18
0	0	0	1	15
0	0	1	0	27
0	0	1	1	20
0	1	0	0	22
0	1	0	1	19
0	1	1	0	27
0	1	1	1	20

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
1	0	0	0	16
1	0	0	1	13
1	0	1	0	25
1	0	1	1	18
1	1	0	0	18
1	1	0	1	15
1	1	1	0	23
1	1	1	1	16

Computational Complexity

Segmentation

$2^{|V|}$

$|V|$ = number of pixels ≈ 153600

Can we do better than brute-force?

MAP Estimation is NP-hard !!

MAP Inference / Energy Minimization

- Computing the assignment minimizing the energy in NP-hard in general

$$\operatorname*{argmin}_{\mathbf{y} \in \mathcal{Y}} E(\mathbf{y}; \mathbf{x}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y} \mid \mathbf{x})$$

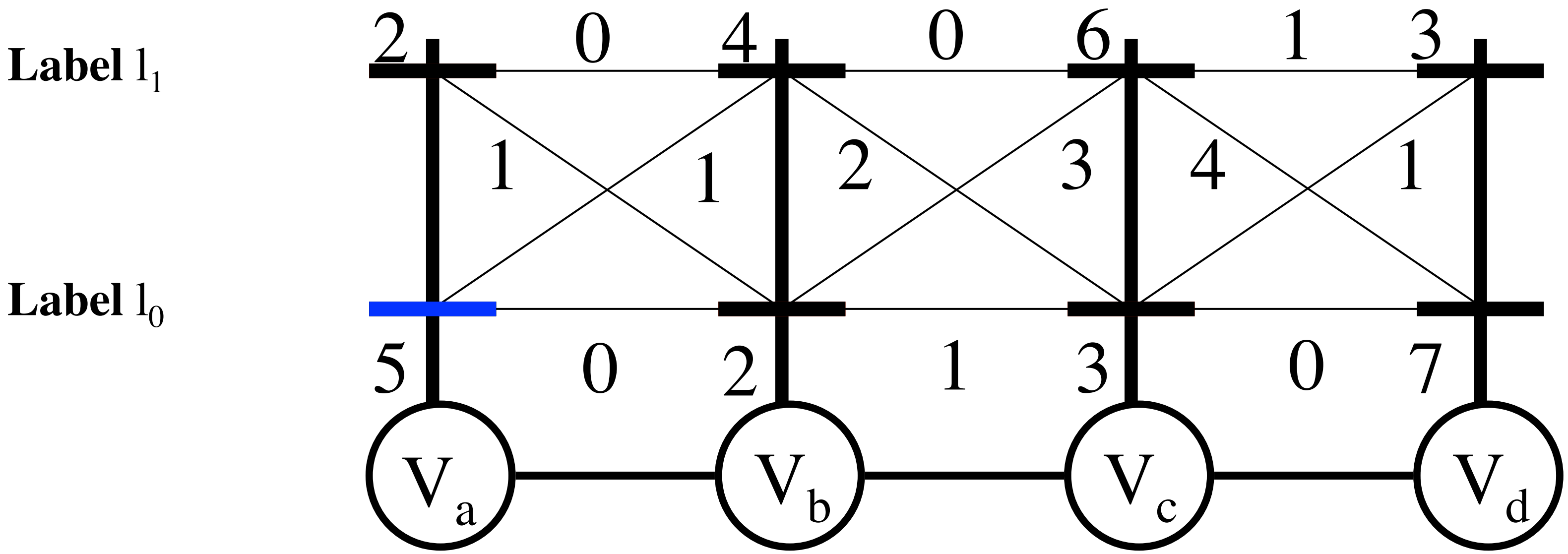
- Exact inference is possible in some cases, e.g.,
  – Low treewidth graphs → message-passing
  – Submodular potentials → graph cuts
- Efficient approximate inference algorithms exist
  – Message passing on general graphs
  – Move-making algorithms
  – Relaxation algorithms

Overview

- Basics: problem formulation
  – Energy Function
  – MAP Estimation
  – Computing min-marginals
  – Reparameterization

- Solutions
  – Belief Propagation and related methods [Lecture 3]
  – Graph cuts [Lecture 2]

Min-Marginals



Not a marginal (no summation)

f* = arg min Q(f; θ) **such that f(a) = i**

Min-marginal $q_{a;i}$

Min-Marginals

16 possible labellings

$q_{a;0} = 15$

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
0	0	0	0	18
0	0	0	1	15
0	0	1	0	27
0	0	1	1	20
0	1	0	0	22
0	1	0	1	19
0	1	1	0	27
0	1	1	1	20

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
1	0	0	0	16
1	0	0	1	13
1	0	1	0	25
1	0	1	1	18
1	1	0	0	18
1	1	0	1	15
1	1	1	0	23
1	1	1	1	16

Min-Marginals

16 possible labellings

$q_{a;1} = 13$

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
0	0	0	0	18
0	0	0	1	15
0	0	1	0	27
0	0	1	1	20
0	1	0	0	22
0	1	0	1	19
0	1	1	0	27
0	1	1	1	20

f(a)	f(b)	f(c)	f(d)	Q(f; θ)
1	0	0	0	16
1	0	0	1	13
1	0	1	0	25
1	0	1	1	18
1	1	0	0	18
1	1	0	1	15
1	1	1	0	23
1	1	1	1	16

Min-Marginals and MAP

- Minimum min-marginal of any variable = energy of MAP labelling

$$\min_i \; q_{a;i}$$

$$\min_i \left(\min_f Q(f; \theta) \quad \text{such that } f(a) = i \right)$$

V_a has to take one label

$$\min_f Q(f; \theta)$$

Summary

Energy Function

$$Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

MAP Estimation

$$f^* = \arg\min Q(f; \theta)$$

Min-marginals

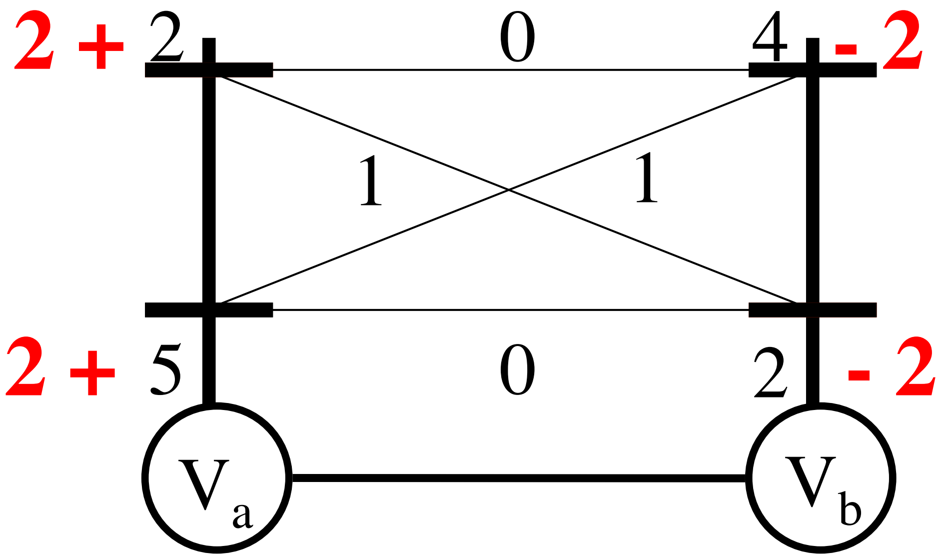
$$q_{a;i} = \min Q(f; \theta) \quad \text{s.t. } f(a) = i$$

Overview

- Basics: problem formulation
  - Energy Function
  - MAP Estimation
  - Computing min-marginals
  - Reparameterization

- Solutions
  - Belief Propagation and related methods [Lecture 3]
  - Graph cuts [Lecture 2]

Reparameterization



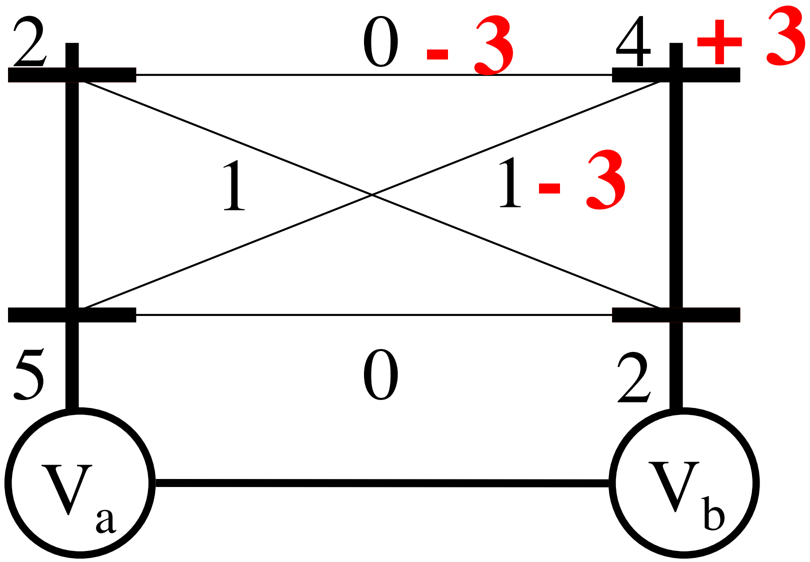
f(a)	f(b)	Q(f; θ)
0	0	7 + 2 - 2
0	1	10 + 2 - 2
1	0	5 + 2 - 2
1	1	6 + 2 - 2

Add a constant to all $\theta_{a;i}$

Subtract that constant from all $\theta_{b;k}$

$Q(f;\theta') = Q(f;\theta)$

Reparameterization



f(a)	f(b)	Q(f; θ)
0	0	7
0	1	10 - 3 + 3
1	0	5
1	1	6 - 3 + 3

Add a constant to one $\theta_{b;k}$

Subtract that constant from $\theta_{ab;ik}$ for all 'i'

$Q(f; \theta') = Q(f; \theta)$

Reparameterization

θ' is a reparameterization of θ, iff

$$Q(f; \theta') = Q(f; \theta), \text{ for all } f \qquad \theta' \equiv \theta$$

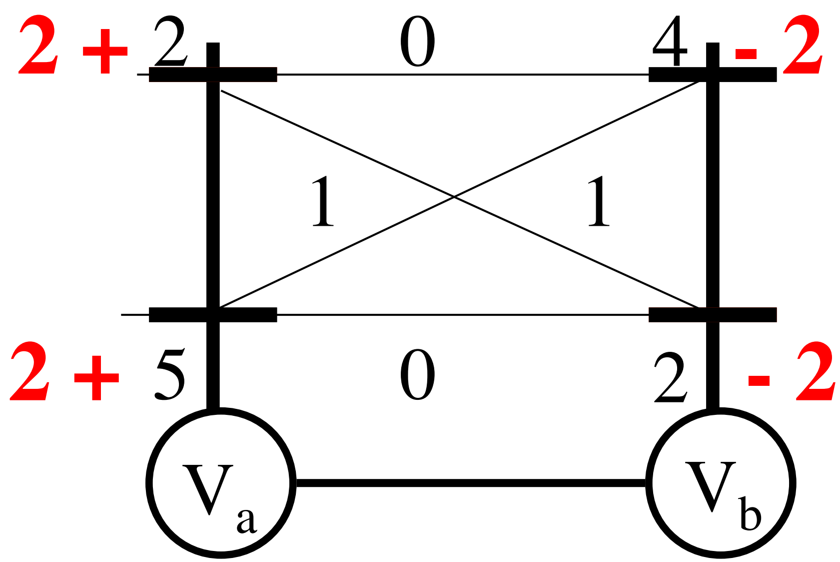
Equivalently

Kolmogorov, PAMI, 2006

$$\theta'_{a;i} = \theta_{a;i} + M_{ba;i}$$

$$\theta'_{b;k} = \theta_{b;k} + M_{ab;k}$$

$$\theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k} - M_{ba;i}$$



Recap

MAP Estimation

$$\mathbf{f^* = \arg\min Q(f; \theta)}$$

$$Q(f; \theta) = \sum_a \theta_{a;f(a)} + \sum_{(a,b)} \theta_{ab;f(a)f(b)}$$

Min-marginals

$$\mathbf{q_{a;i} = \min Q(f; \theta) \quad \text{s.t. } f(a) = i}$$

Reparameterization

$$\mathbf{Q(f; \theta') = Q(f; \theta), \text{ for all } f} \qquad \theta' \equiv \theta$$

Overview

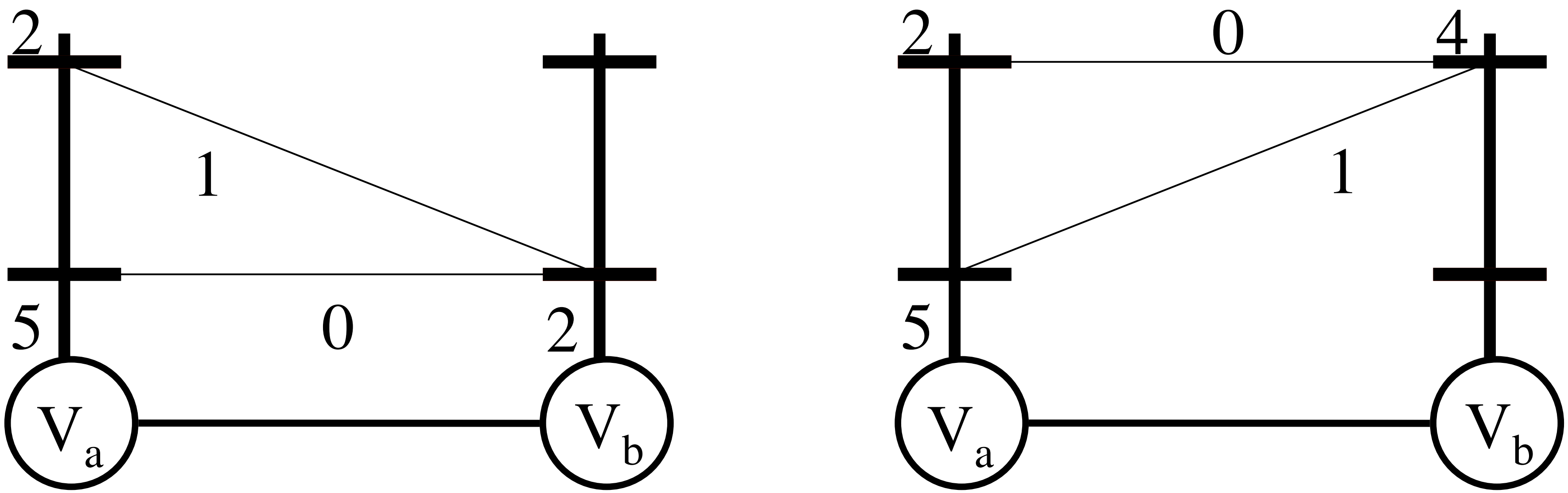
- Basics: problem formulation
  - Energy Function
  - MAP Estimation
  - Computing min-marginals
  - Reparameterization

- Solutions
  - Belief Propagation and related methods [Lecture 3]
  - Graph cuts [Lecture 2]

Belief Propagation

- Remember, some MAP problems are easy
- Belief Propagation gives exact MAP for chains
- Exact MAP for trees
- Clever Reparameterization

Two Variables

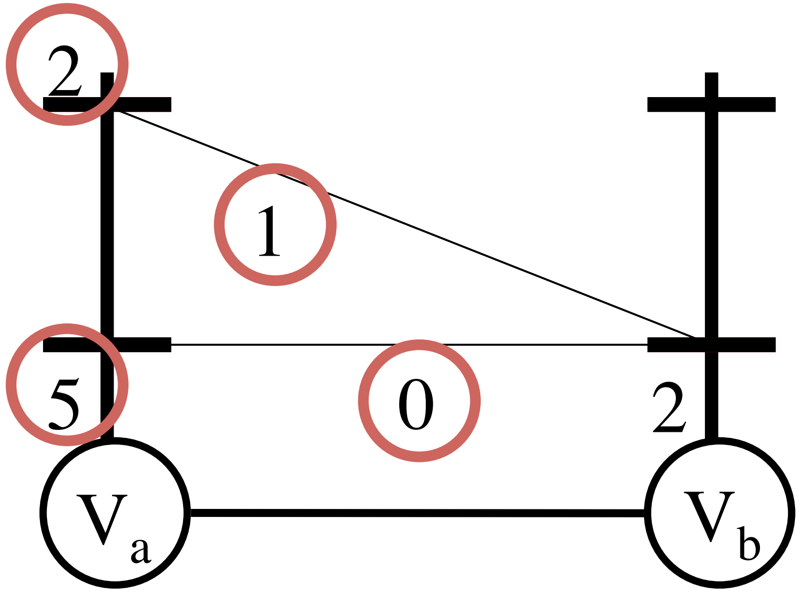


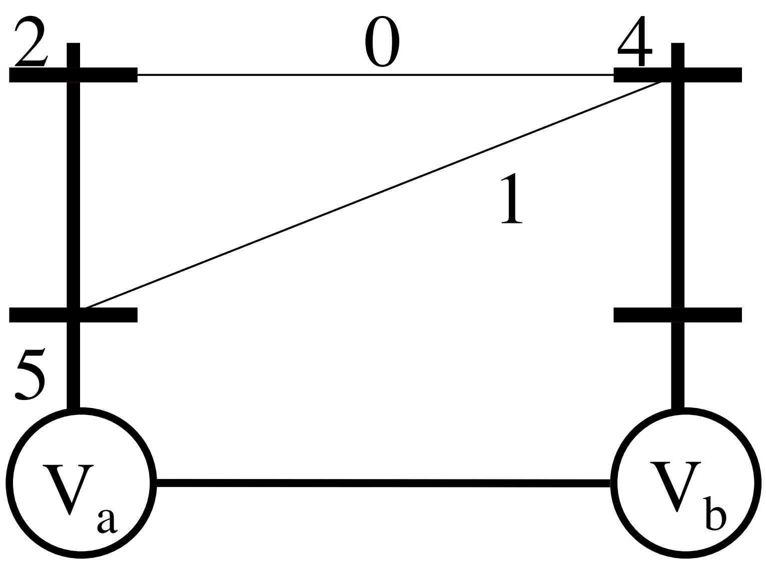
Add a constant to one $\theta_{b;k}$

Subtract that constant from $\theta_{ab;ik}$ for all 'i'

Choose the ***right*** constant $\theta'_{b;k} = q_{b;k}$

Two Variables

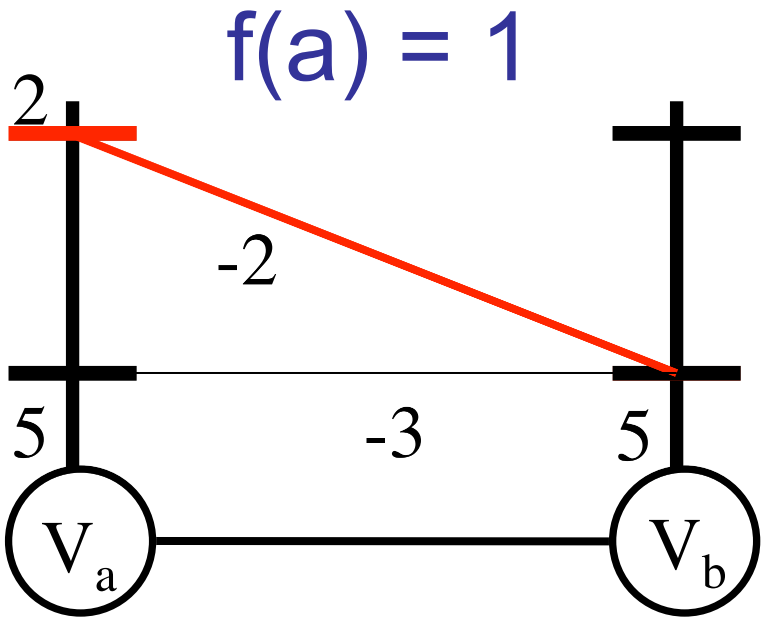


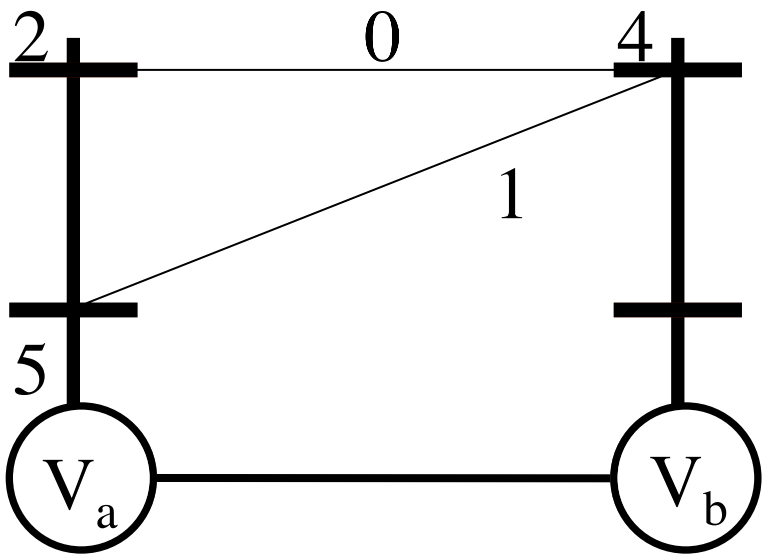


$$M_{ab;0} = \min \begin{array}{ll} \theta_{a;0} + \theta_{ab;00} & = 5 + 0 \\ \theta_{a;1} + \theta_{ab;10} & = 2 + 1 \end{array}$$

Choose the ***right*** constant $\theta'_{b;k} = q_{b;k}$

Two Variables



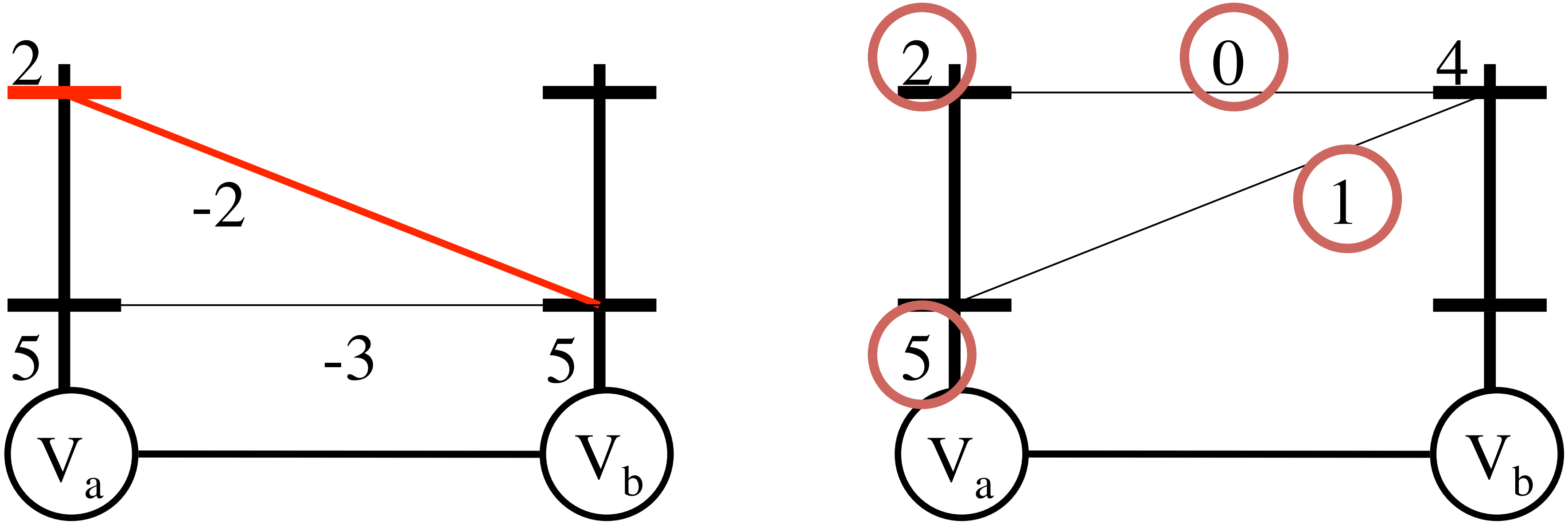


$$\theta'_{b;0} = q_{b;0}$$

Potentials along the red path add up to 0

Choose the ***right*** constant $\theta'_{b;k} = q_{b;k}$

Two Variables

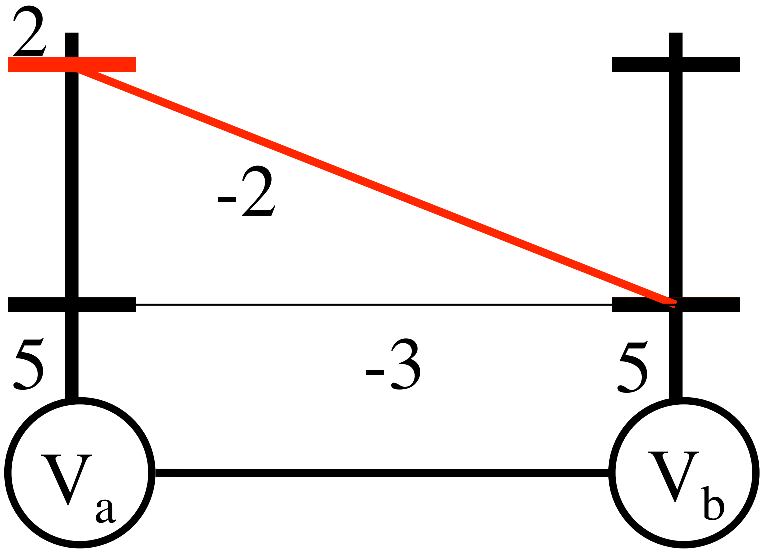


$$M_{ab;1} = \min \begin{array}{ll} \theta_{a;0} + \theta_{ab;01} & = 5 + 1 \\ \theta_{a;1} + \theta_{ab;11} & = 2 + 0 \end{array}$$

Choose the ***right*** constant $\theta'_{b;k} = q_{b;k}$

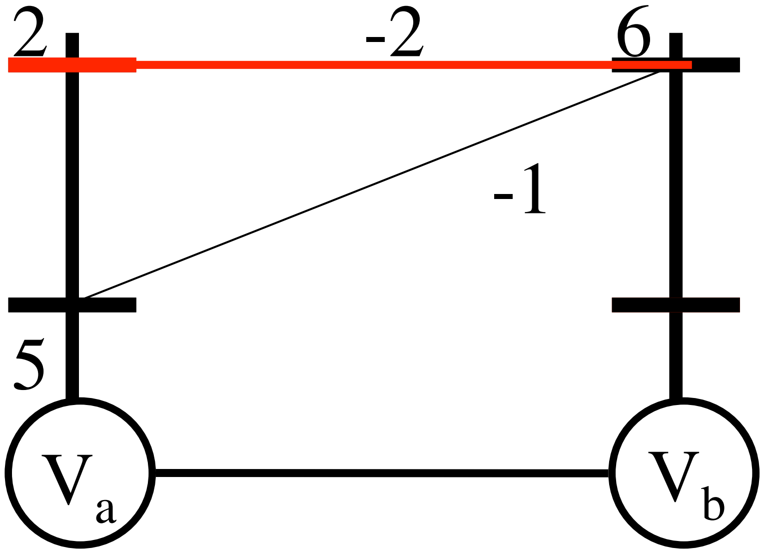
Two Variables

$f(a) = 1$



$\theta'_{b;0} = q_{b;0}$

$f(a) = 1$

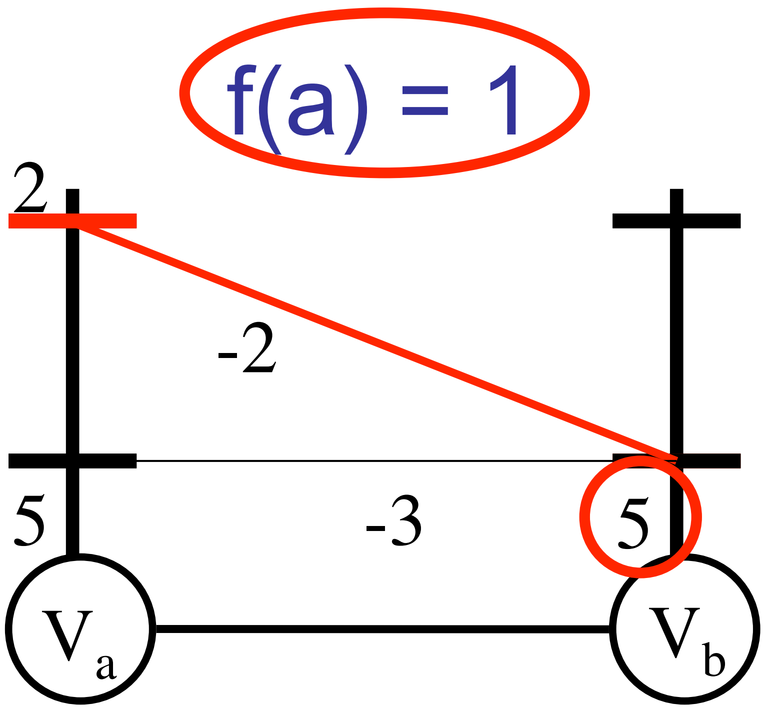


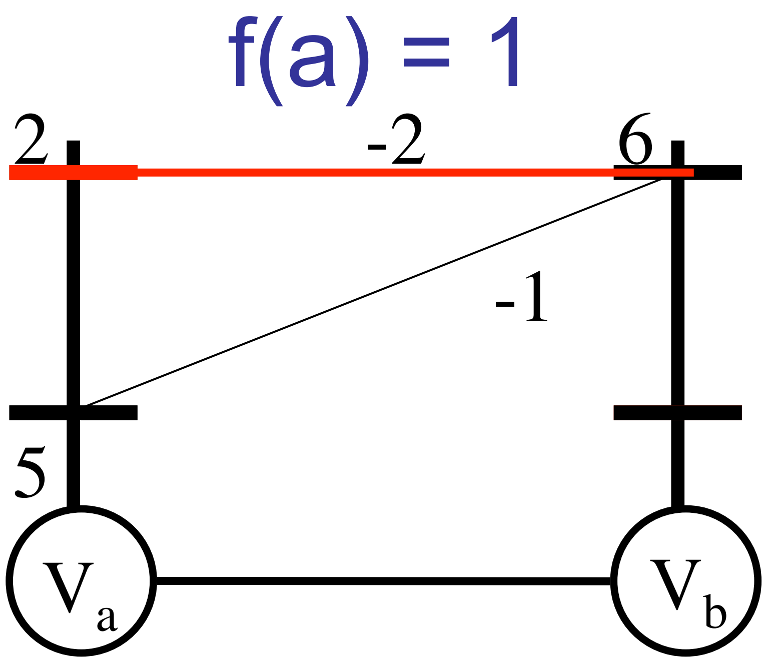
$\theta'_{b;1} = q_{b;1}$

Minimum of min-marginals = MAP estimate

Choose the ***right*** constant $\theta'_{b;k} = q_{b;k}$

Two Variables



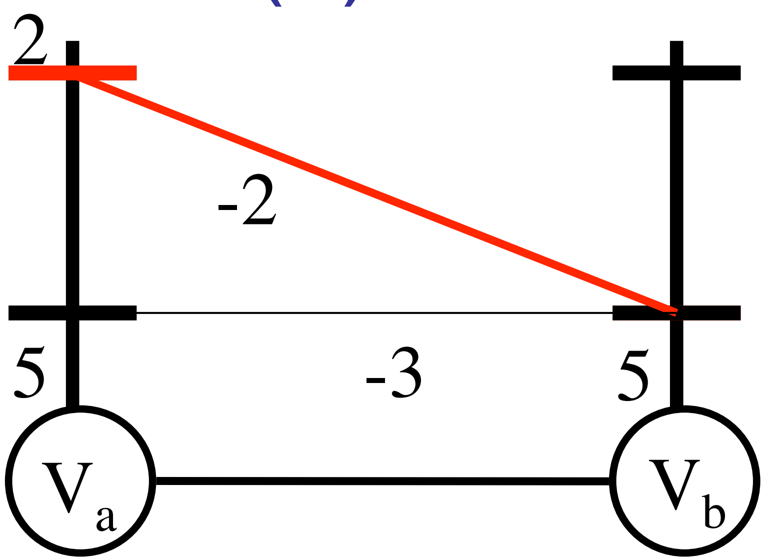


$\theta'_{b;0} = q_{b;0}$ $\theta'_{b;1} = q_{b;1}$

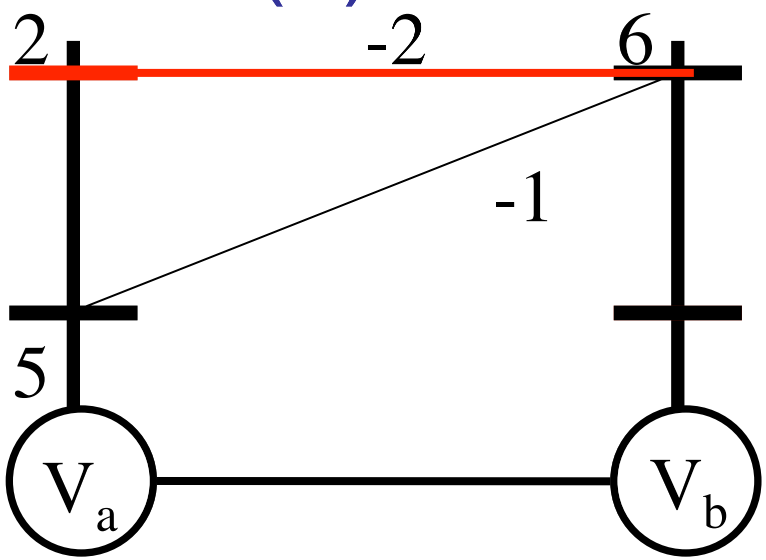
f*(b) = 0 f*(a) = 1

Choose the ***right*** constant $\theta'_{b;k} = q_{b;k}$

Two Variables



$\theta'_{b;0} = q_{b;0}$



$\theta'_{b;1} = q_{b;1}$

We get all the min-marginals of V_b

Choose the ***right*** constant $\theta'_{b;k} = q_{b;k}$

Recap

We only need to know two sets of equations

General form of Reparameterization

$$\theta'_{a;i} = \theta_{a;i} + M_{ba;i} \qquad \theta'_{b;k} = \theta_{b;k} + M_{ab;k}$$

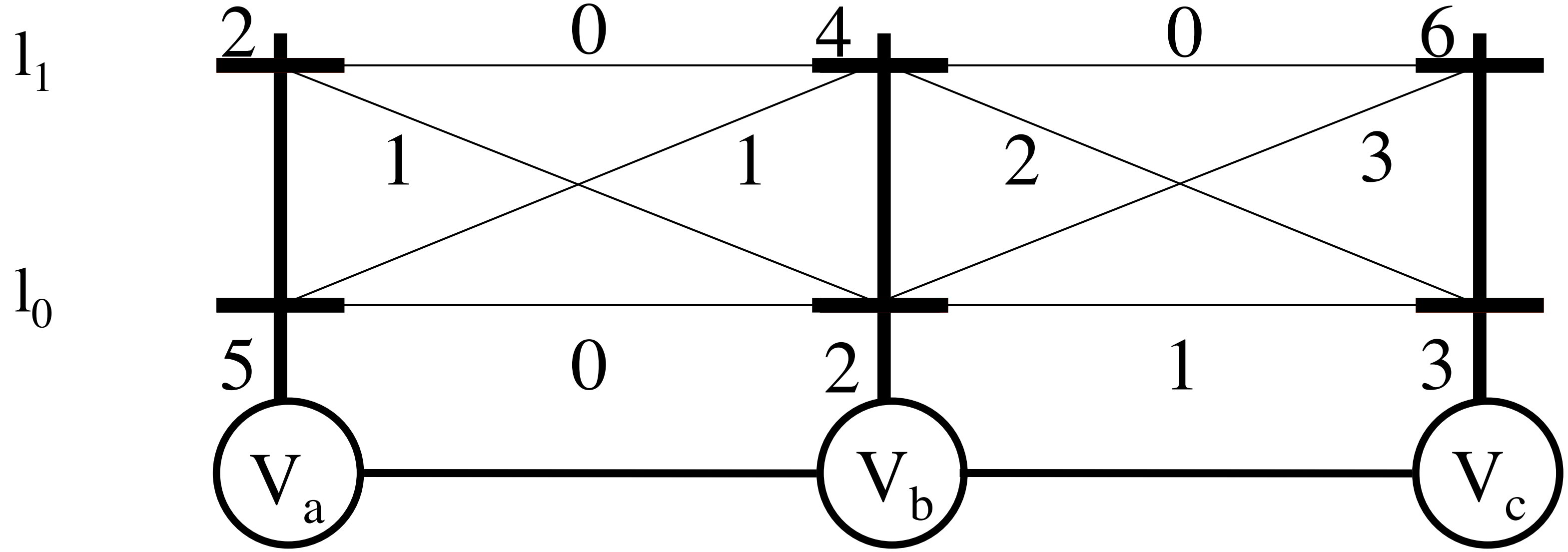
$$\theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k} - M_{ba;i}$$

Reparameterization of (a,b) in Belief Propagation

$$M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \}$$

$$M_{ba;i} = 0$$

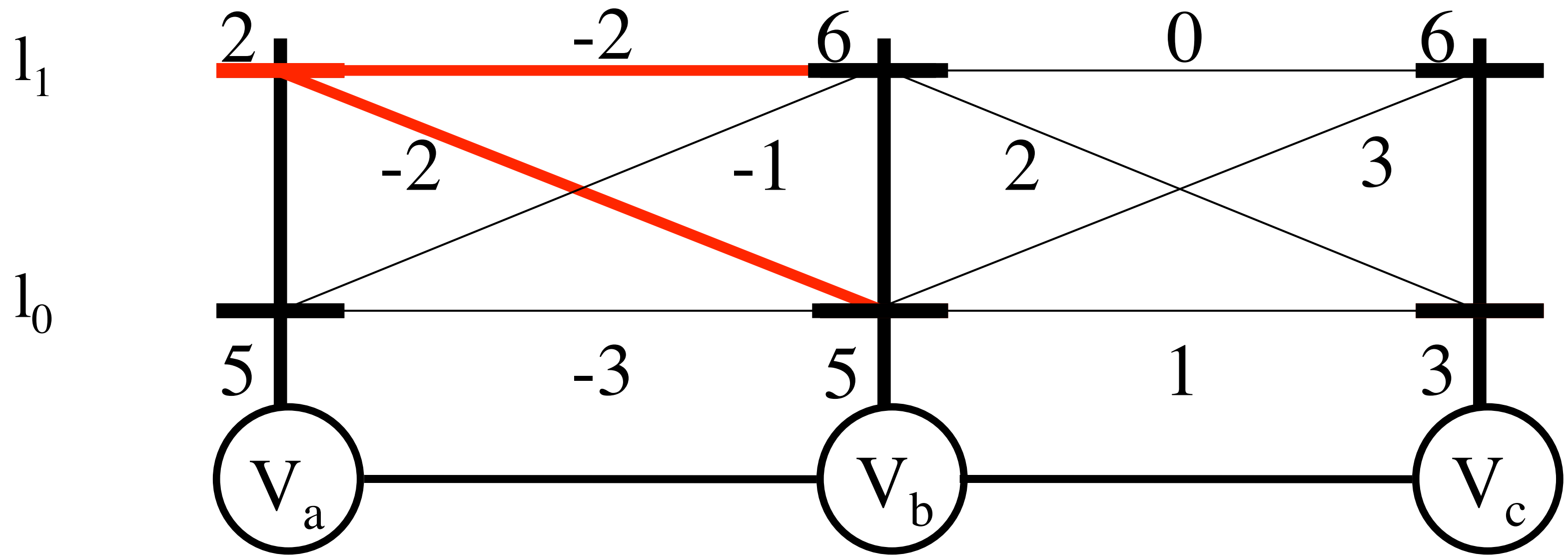
Three Variables



Reparameterize the edge (a,b) as before

Three Variables

f(a) = 1



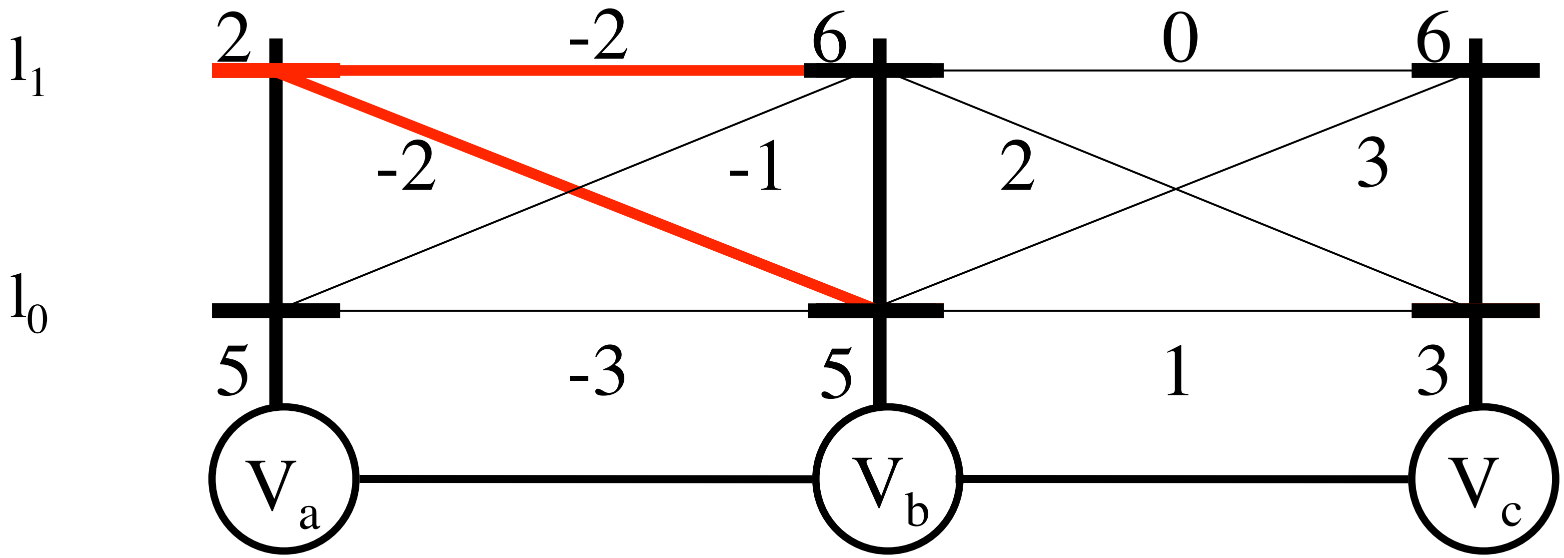
f(a) = 1

Reparameterize the edge (a,b) as before

Potentials along the red path add up to 0

Three Variables

f(a) = 1

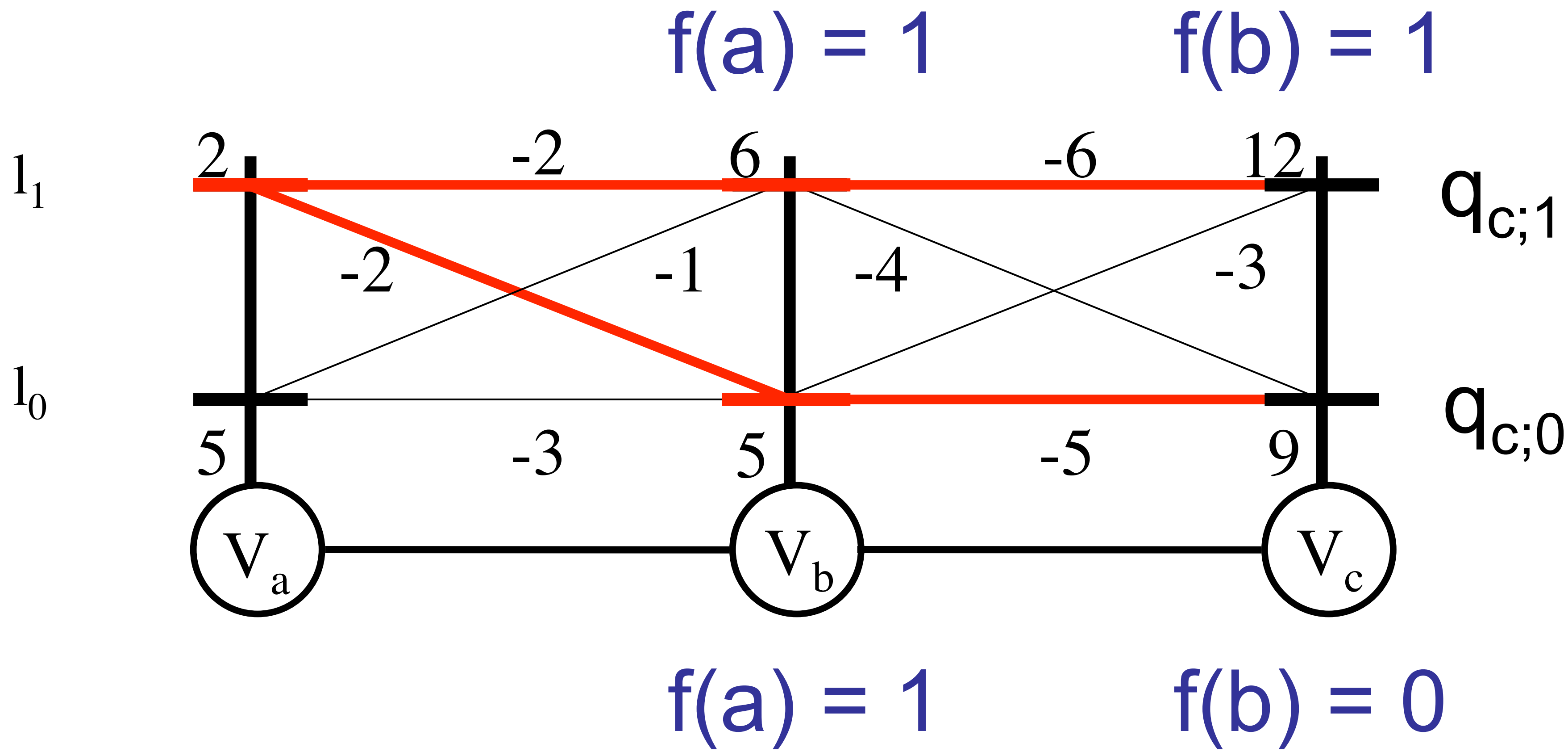


f(a) = 1

Reparameterize the edge (b,c) as before

Potentials along the red path add up to 0

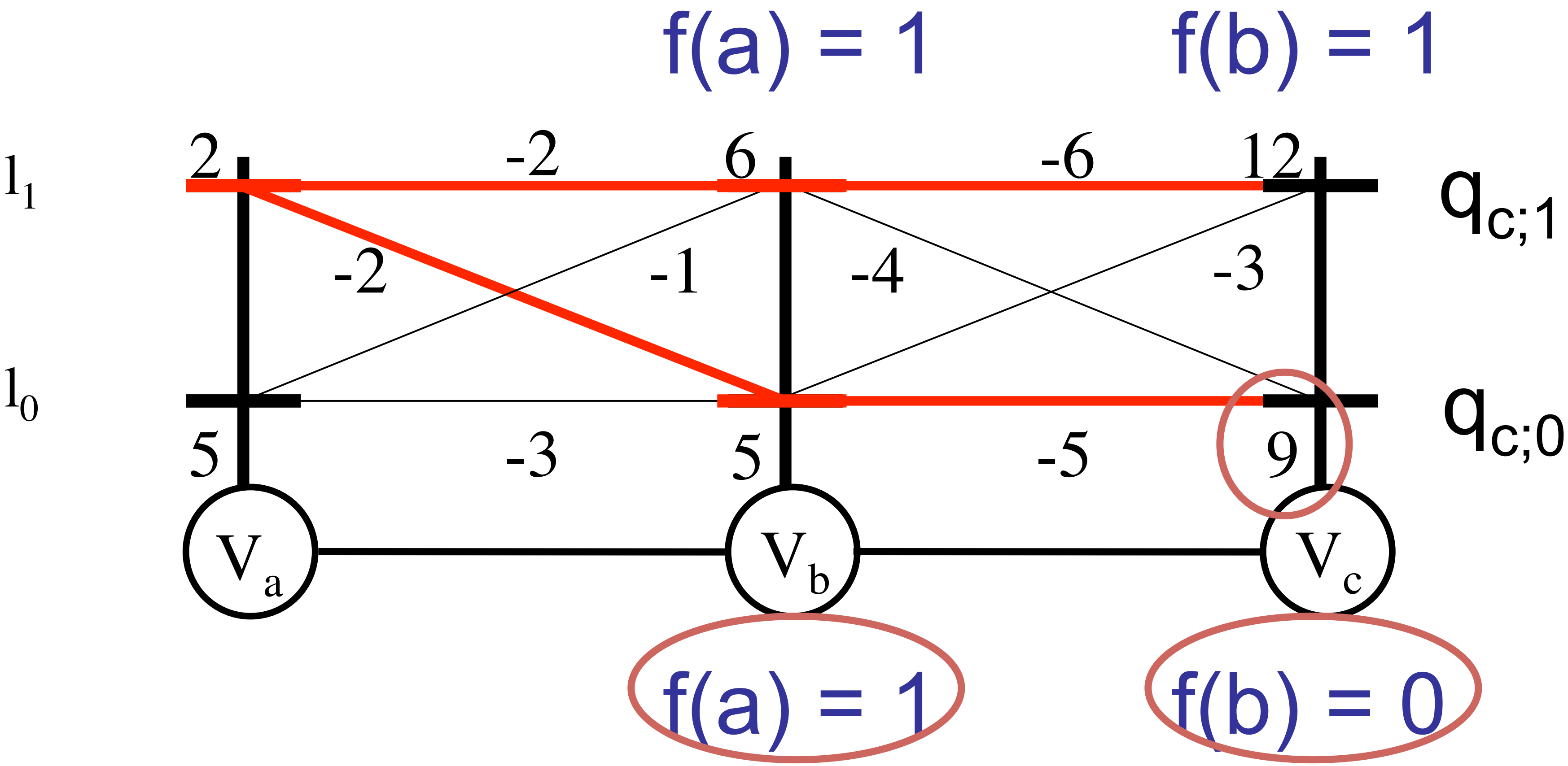
Three Variables



Reparameterize the edge (b,c) as before

Potentials along the red path add up to 0

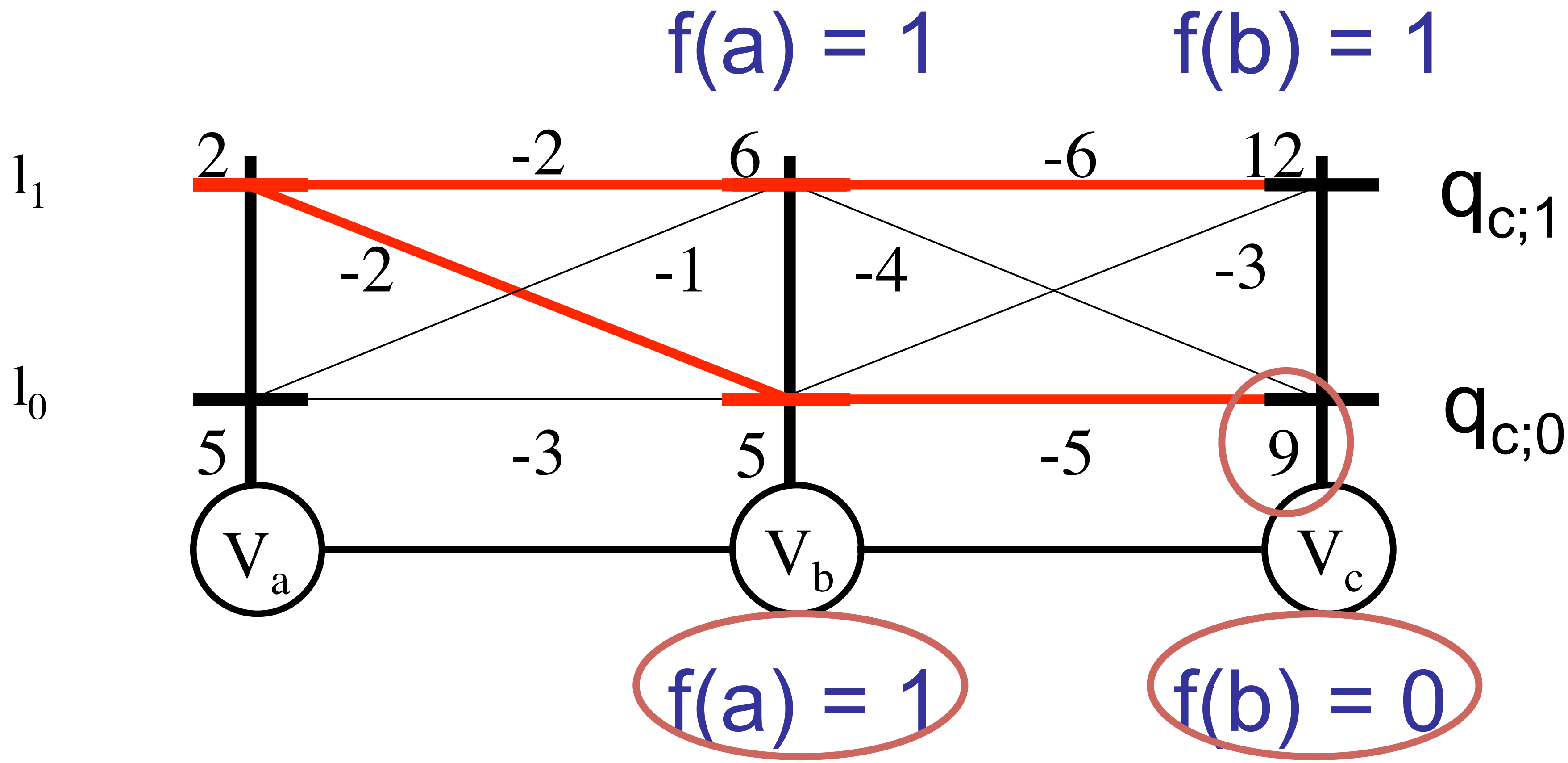
Three Variables



$f^*(c) = 0$ $f^*(b) = 0$ $f^*(a) = 1$

Generalizes to any length chain

Three Variables



f*(c) = 0 f*(b) = 0 f*(a) = 1

Only Dynamic Programming

Why Dynamic Programming?

3 variables ≡ 2 variables + book-keeping

n variables ≡ (n-1) variables + book-keeping

Start from left, go to right

Reparameterize current edge (a,b)

$$M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \}$$

$$\theta'_{b;k} = \theta_{b;k} + M_{ab;k} \quad \theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k}$$

Repeat

Why Dynamic Programming?

Messages   Message Passing

Why stop at dynamic programming?

Start from left, go to right

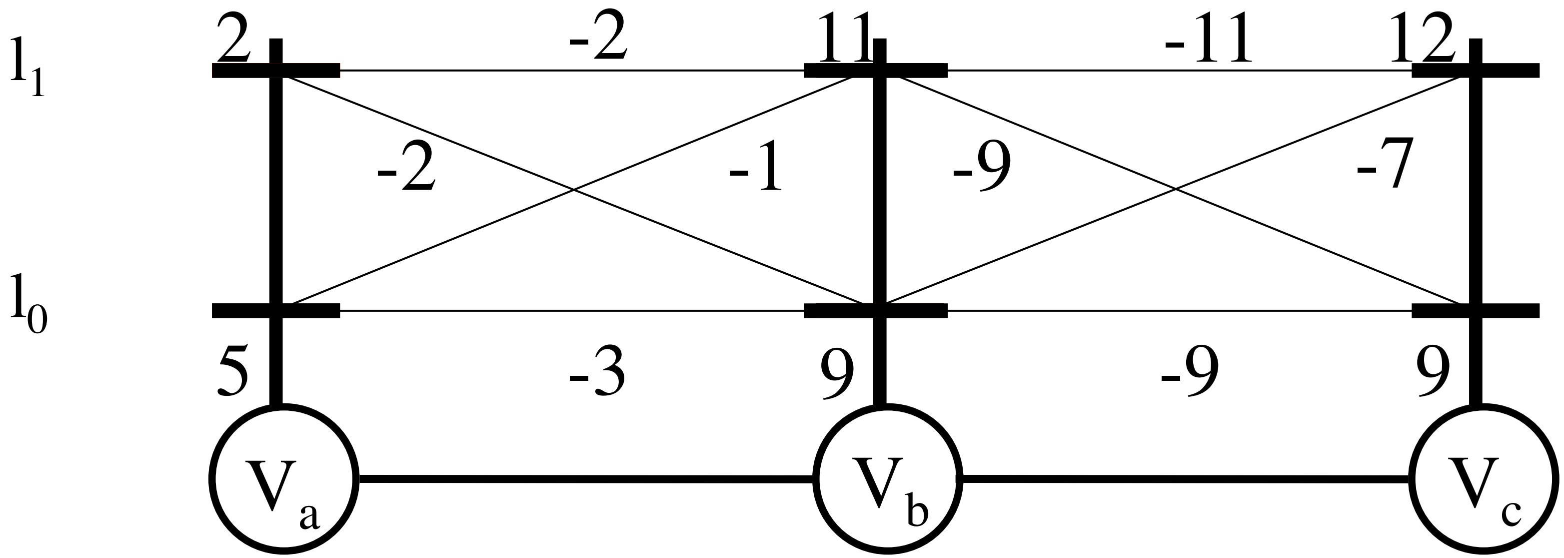
Reparameterize current edge (a,b)

$$M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \}$$

$$\theta'_{b;k} = \theta_{b;k} + M_{ab;k} \quad \theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k}$$

Repeat

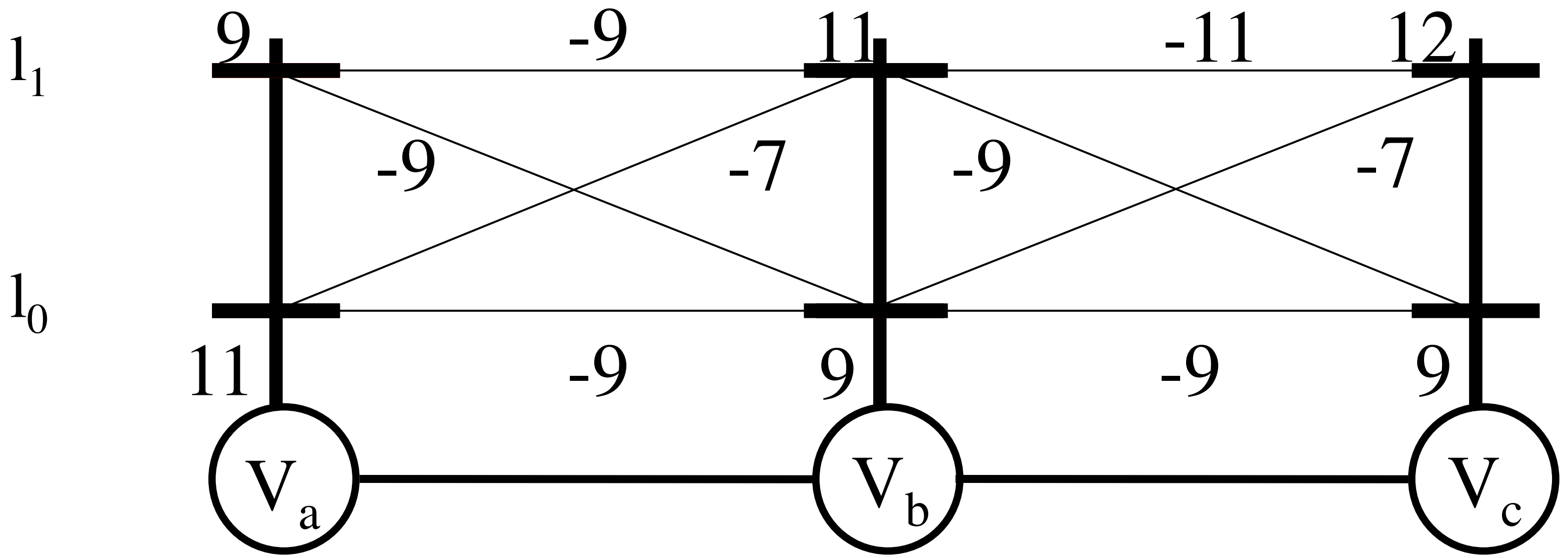
Three Variables



Reparameterize the edge (c,b) as before

$$\theta'_{b;i} = q_{b;i}$$

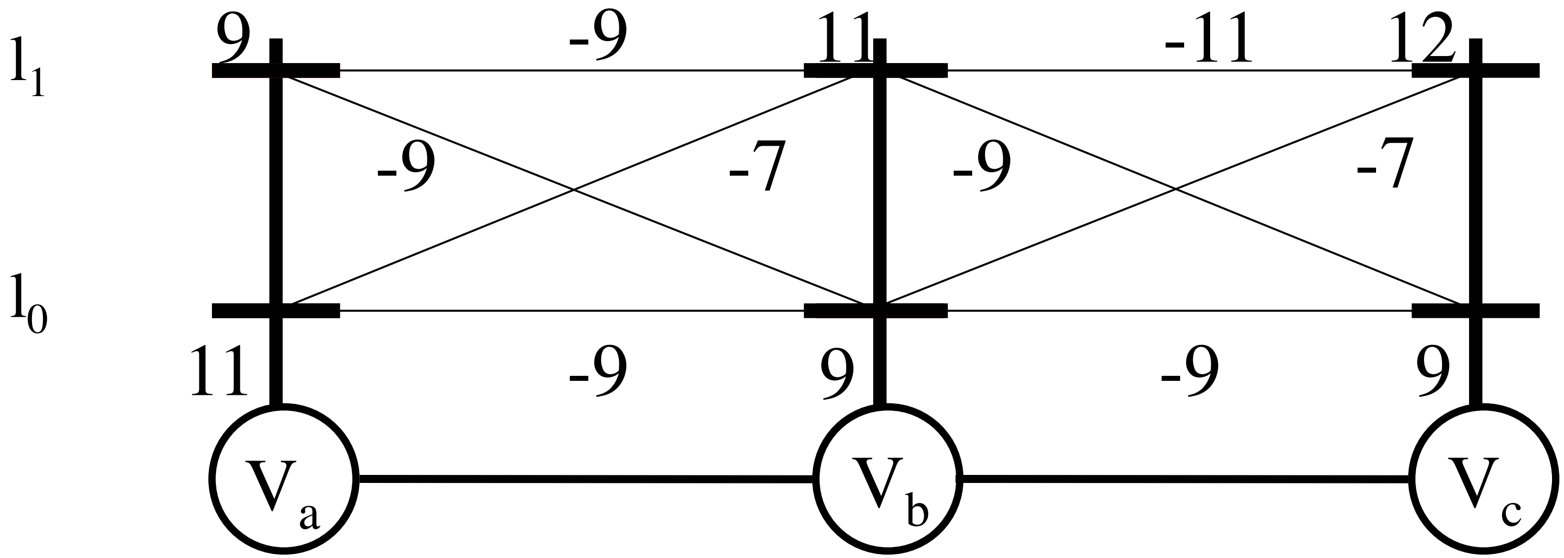
Three Variables



Reparameterize the edge (b,a) as before

$$\theta'_{a;i} = q_{a;i}$$

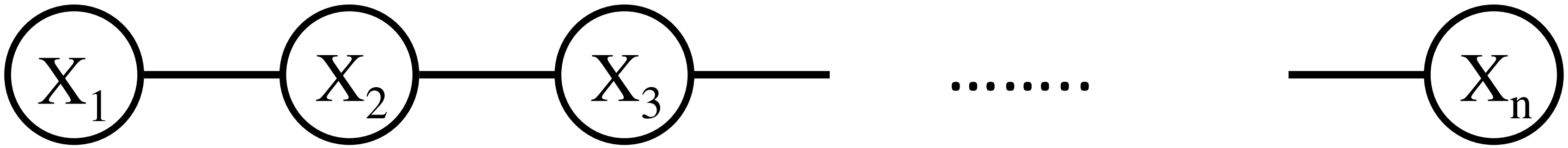
Three Variables



Forward Pass ➔ 🡐 Backward Pass

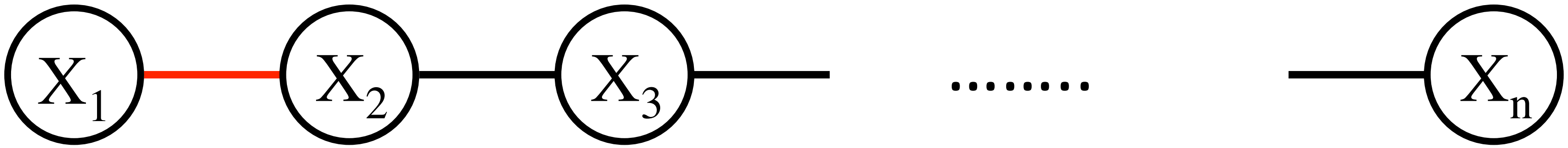
All min-marginals are computed

Chains



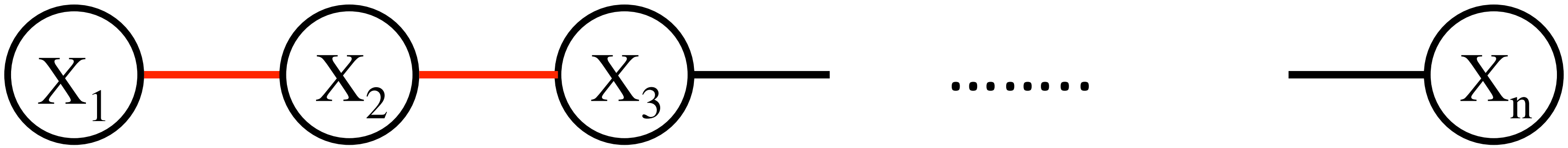
Reparameterize the edge (1,2)

Chains



Reparameterize the edge (1,2)

Chains



Reparameterize the edge (2,3)

Chains

Reparameterize the edge (n-1,n)

Min-marginals $e_n(i)$ for all labels

Belief Propagation on Chains

Start from left, go to right

Reparameterize current edge (a,b)

$$M_{ab;k} = \min_i \{ \theta_{a;i} + \theta_{ab;ik} \}$$

$$\theta'_{b;k} = \theta_{b;k} + M_{ab;k} \quad \theta'_{ab;ik} = \theta_{ab;ik} - M_{ab;k}$$

Repeat till the end of the chain

Start from right, go to left

Repeat till the end of the chain

Belief Propagation on Chains

- Generalizes to chains of any length
- A way of computing reparam constants
- Forward Pass - Start to End
  - MAP estimate
  - Min-marginals of final variable
- Backward Pass - End to start
  - All other min-marginals

Computational Complexity

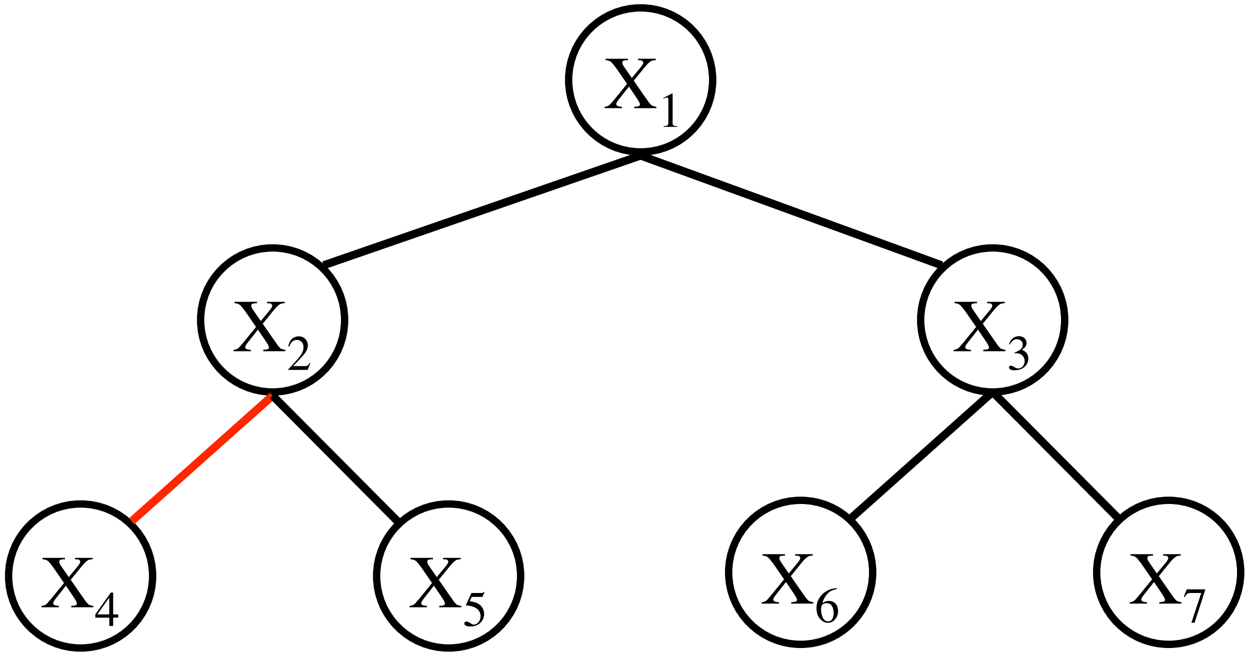
Number of reparameterization constants = (n-1)h

Complexity for each constant = O(h)

Total complexity = $O(nh^2)$

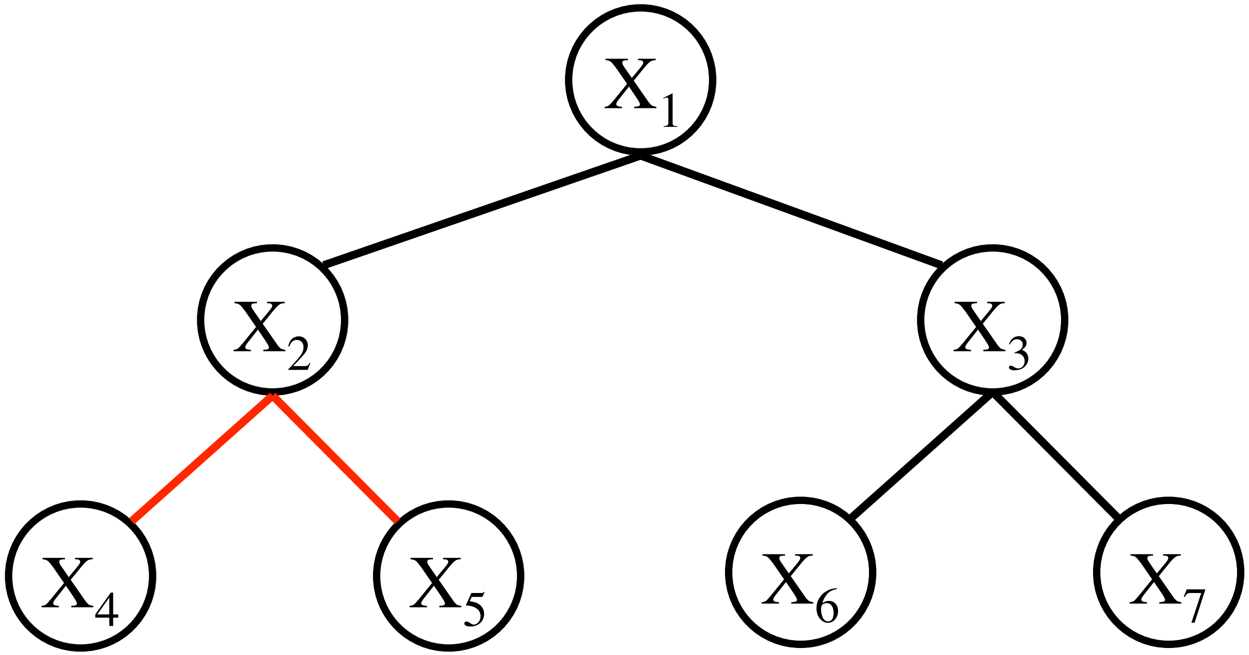
Better than brute-force $O(h^n)$

Trees



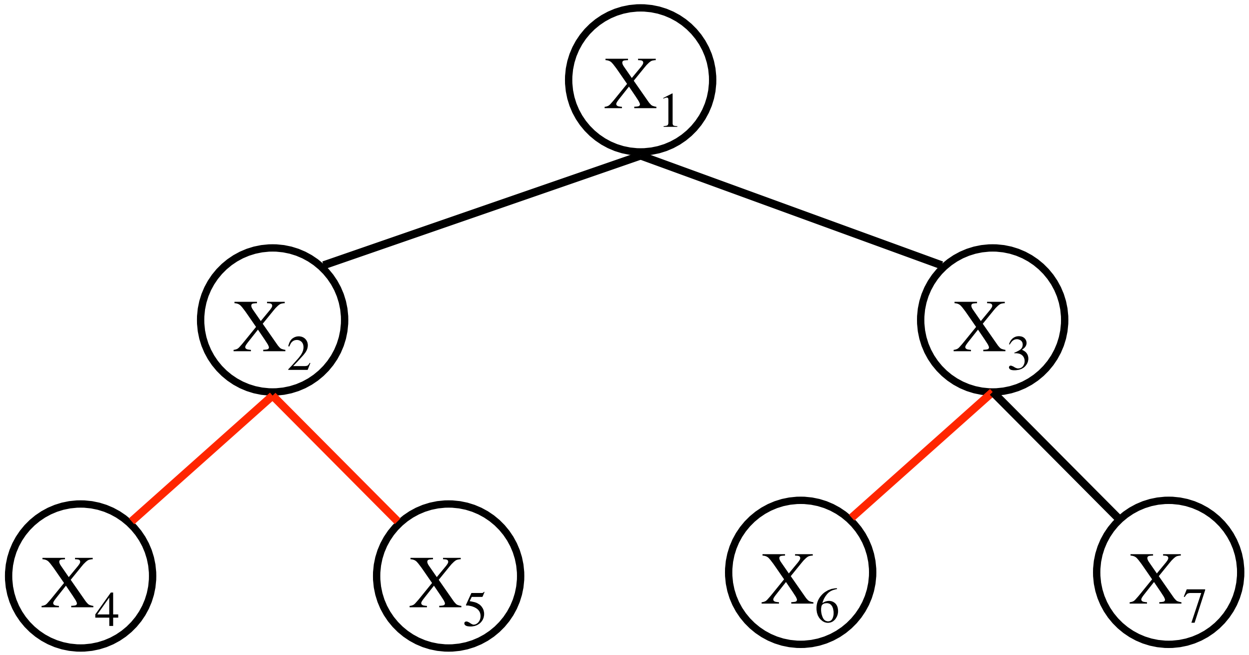
Reparameterize the edge (4,2)

Trees



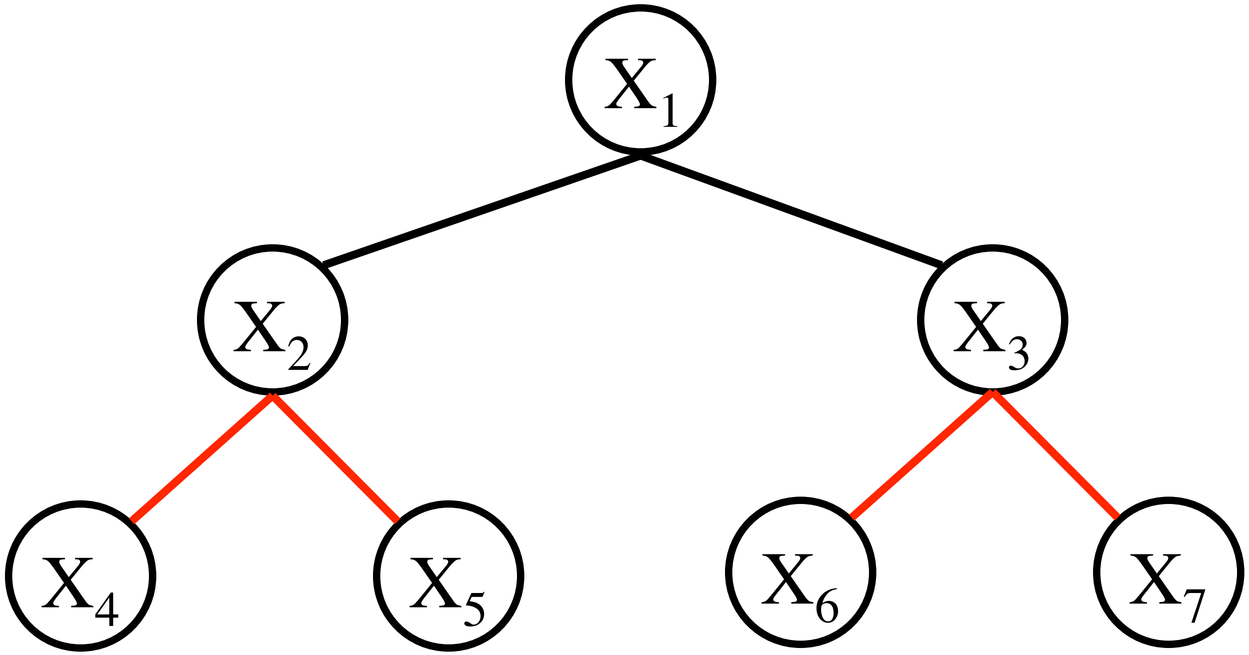
Reparameterize the edge (5,2)

Trees



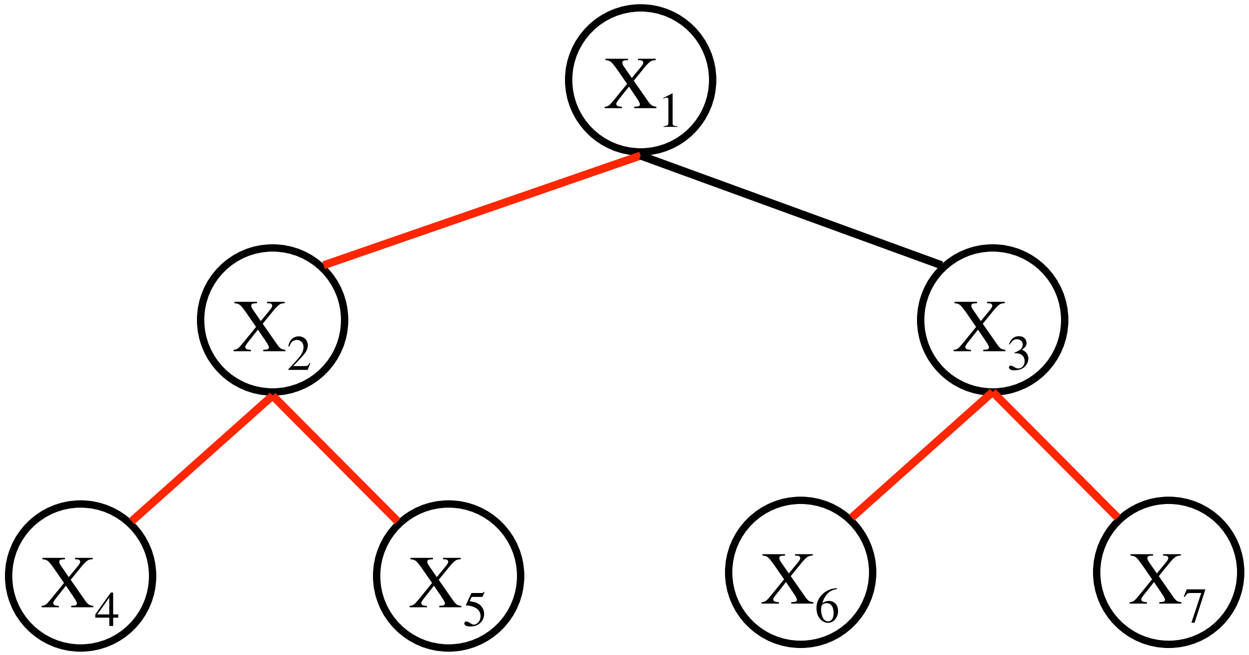
Reparameterize the edge (6,3)

Trees



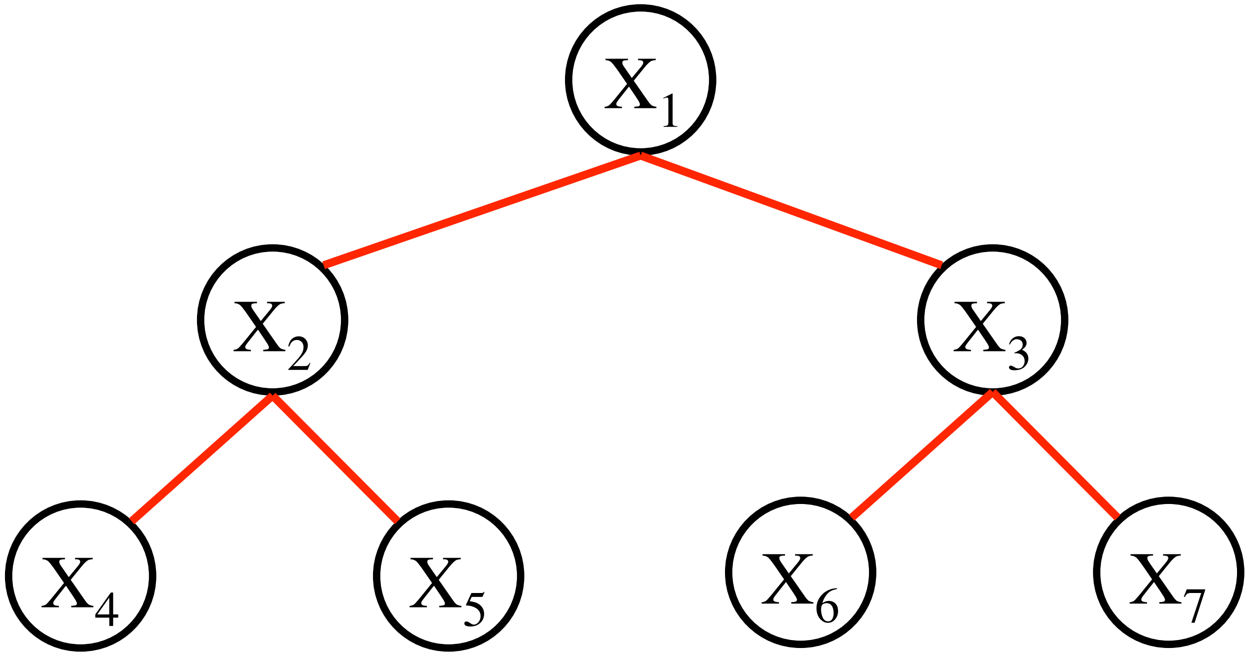
Reparameterize the edge (7,3)

Trees



Reparameterize the edge (2,1)

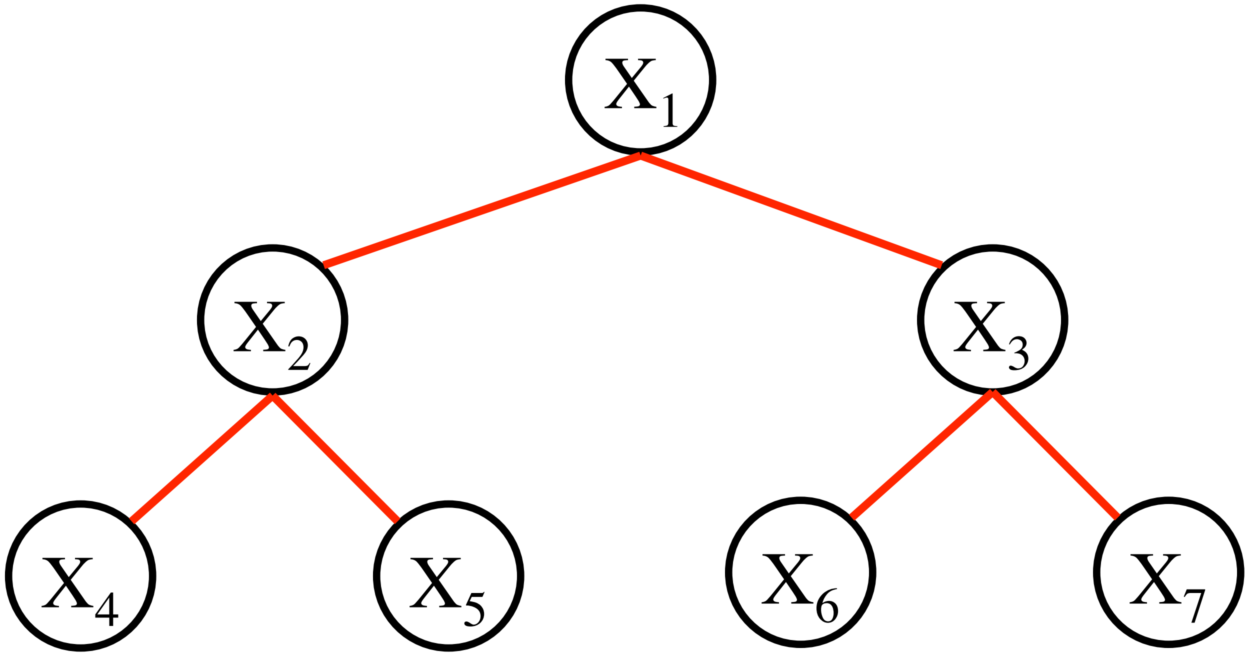
Trees



Reparameterize the edge (3,1)

Min-marginals $e_1(i)$ for all labels

Trees

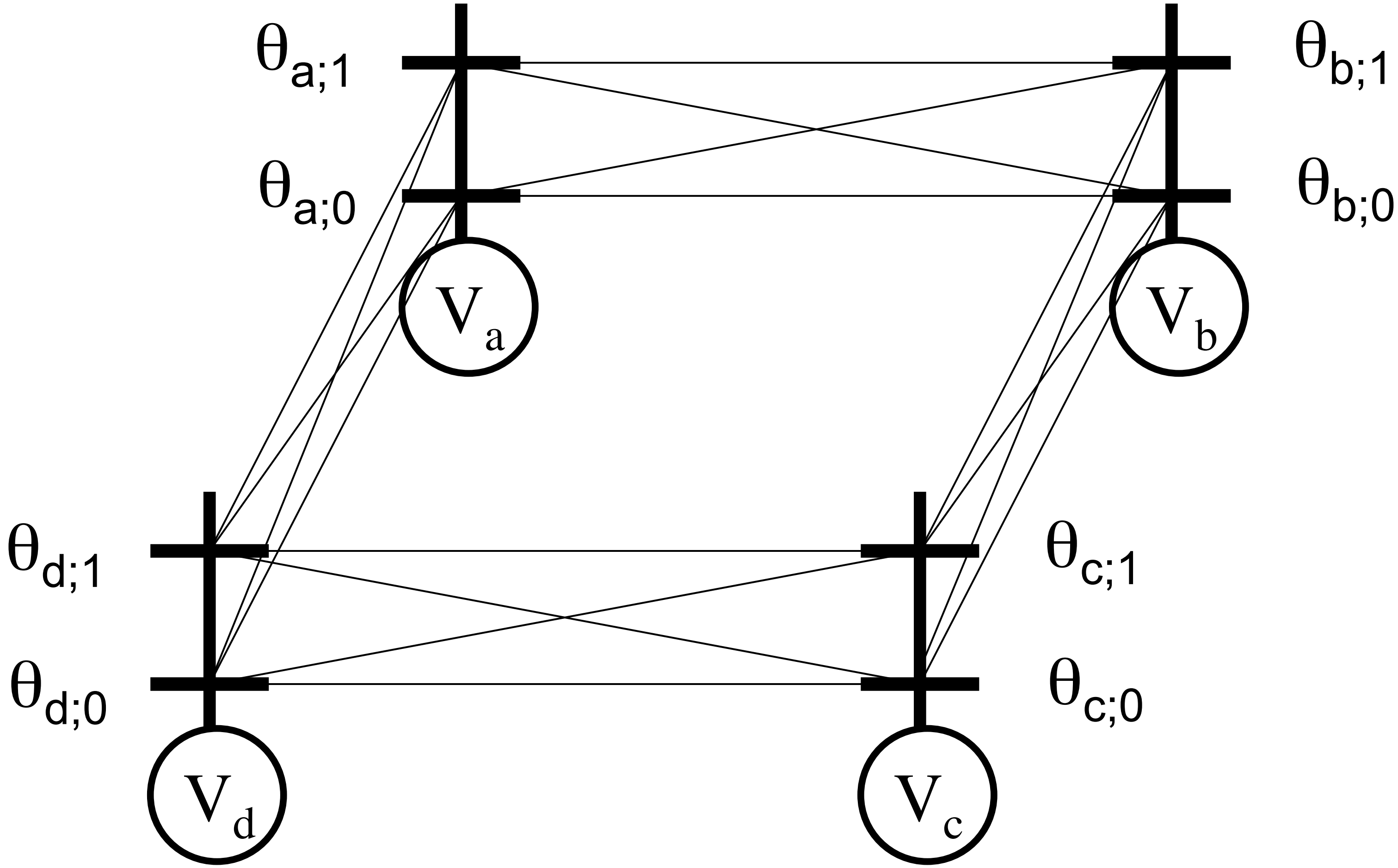


Start from leaves and move towards root

Pick the minimum of min-marginals

Backtrack to find the best labeling **x**

Belief Propagation on Cycles

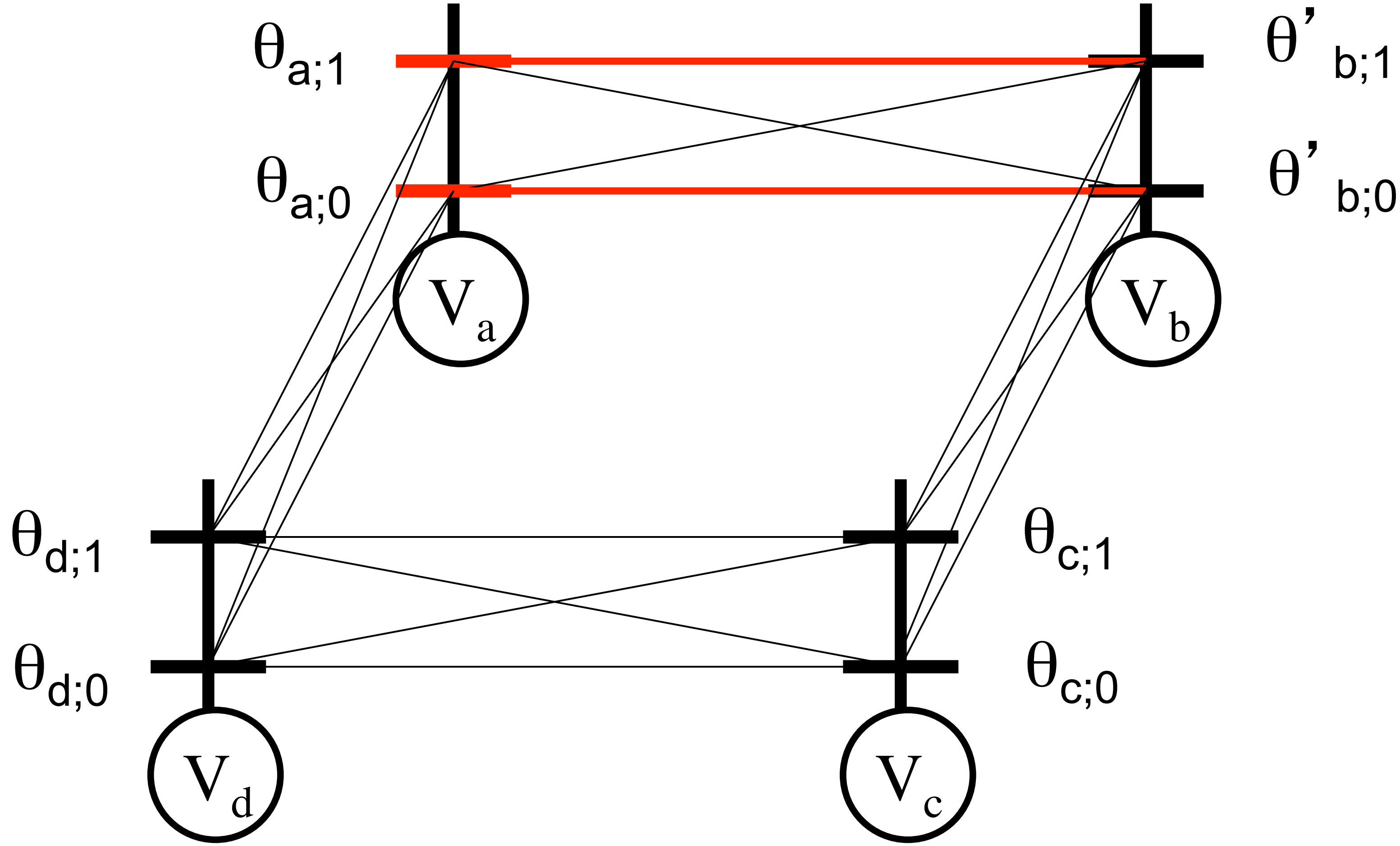


Where do we start?

Arbitrarily

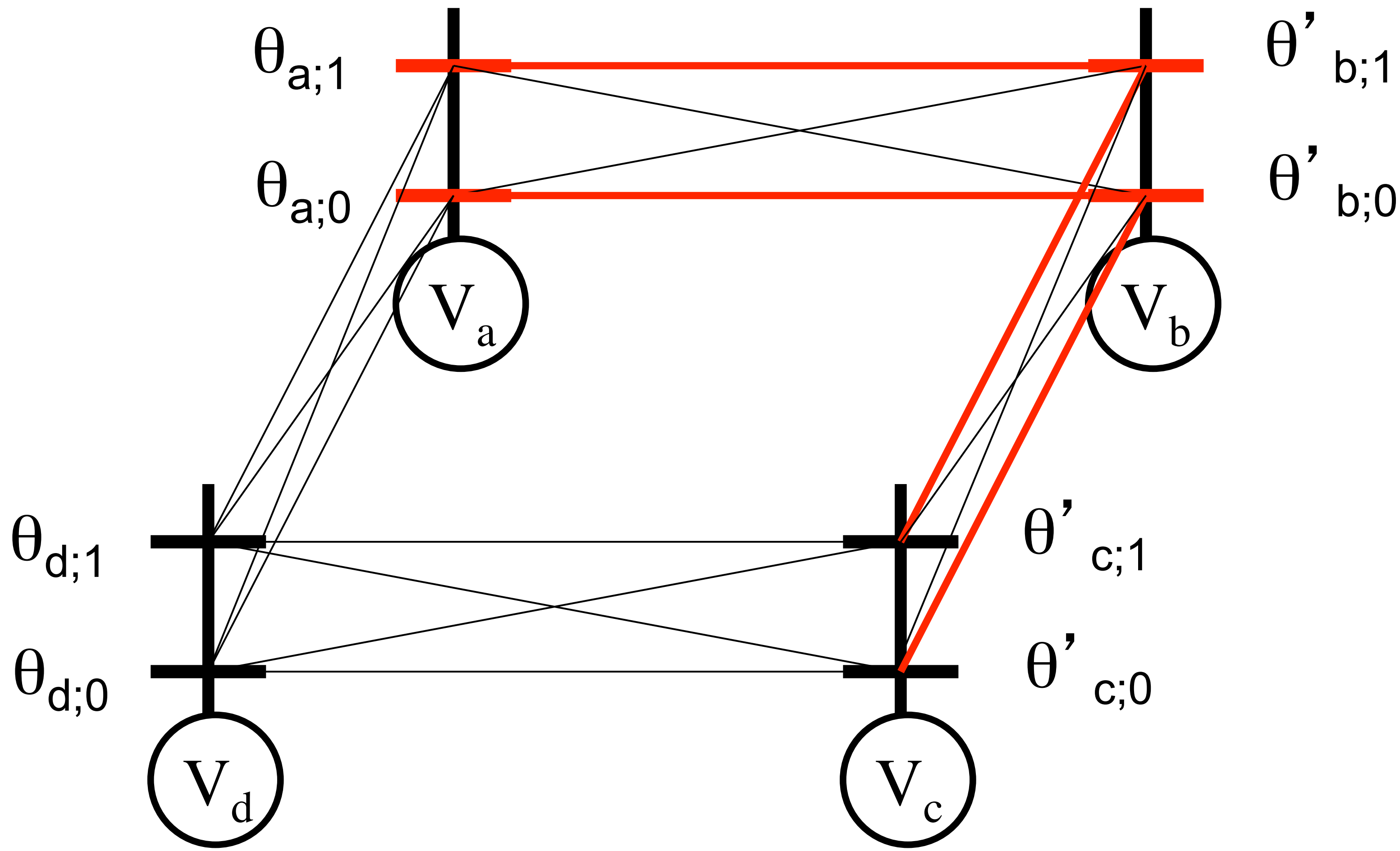
Reparameterize (a,b)

Belief Propagation on Cycles



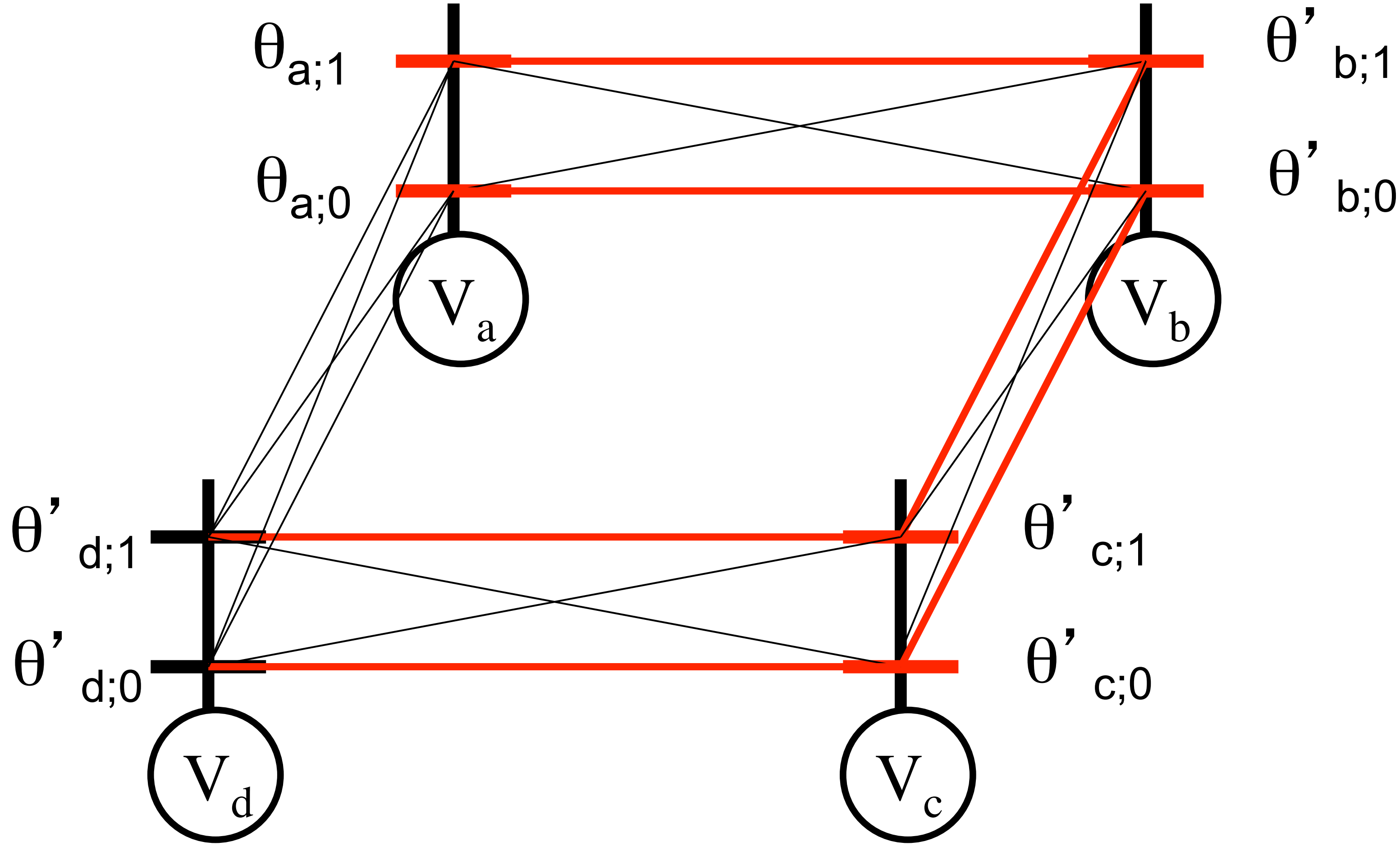
Potentials along the red path add up to 0

Belief Propagation on Cycles



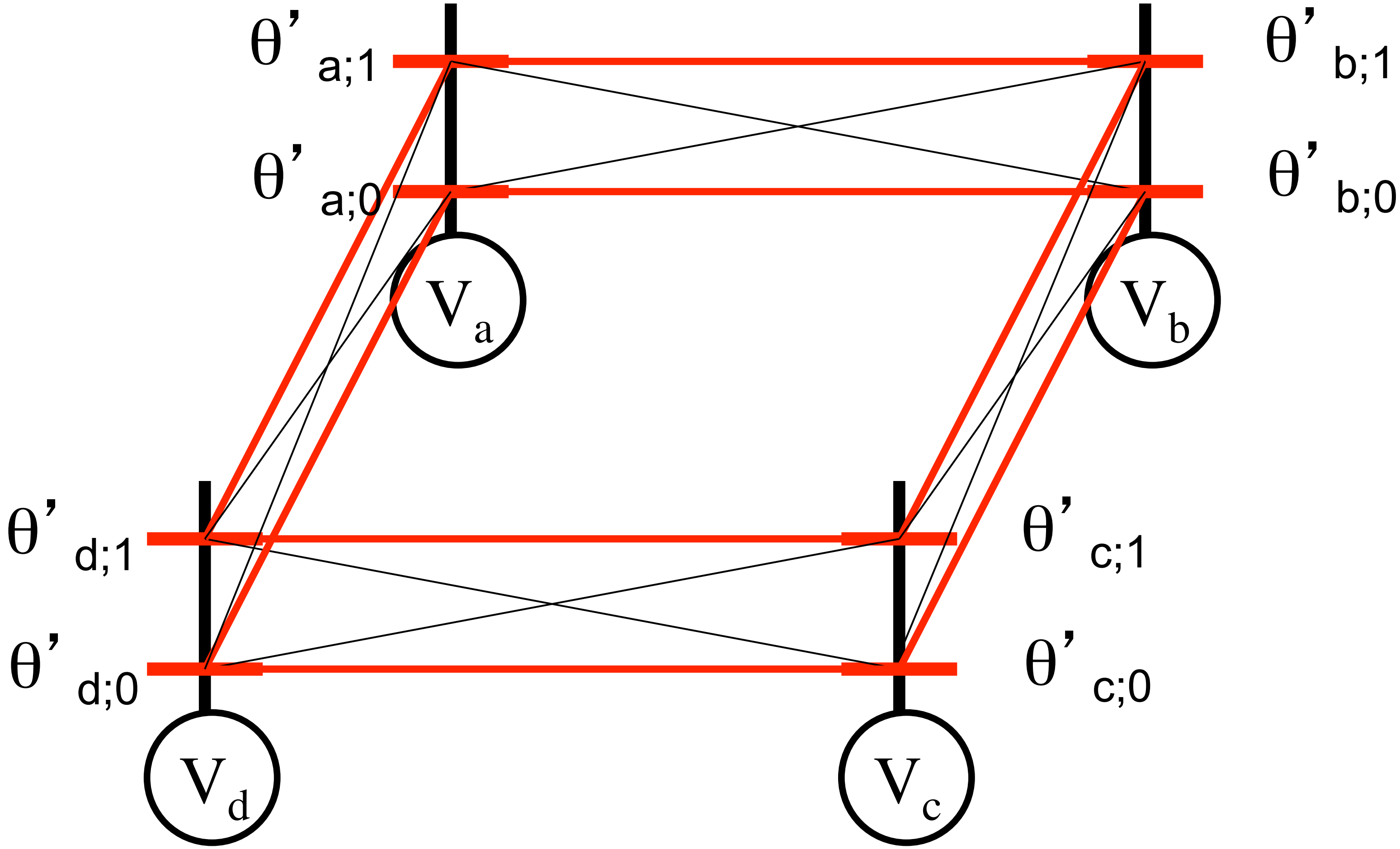
Potentials along the red path add up to 0

Belief Propagation on Cycles



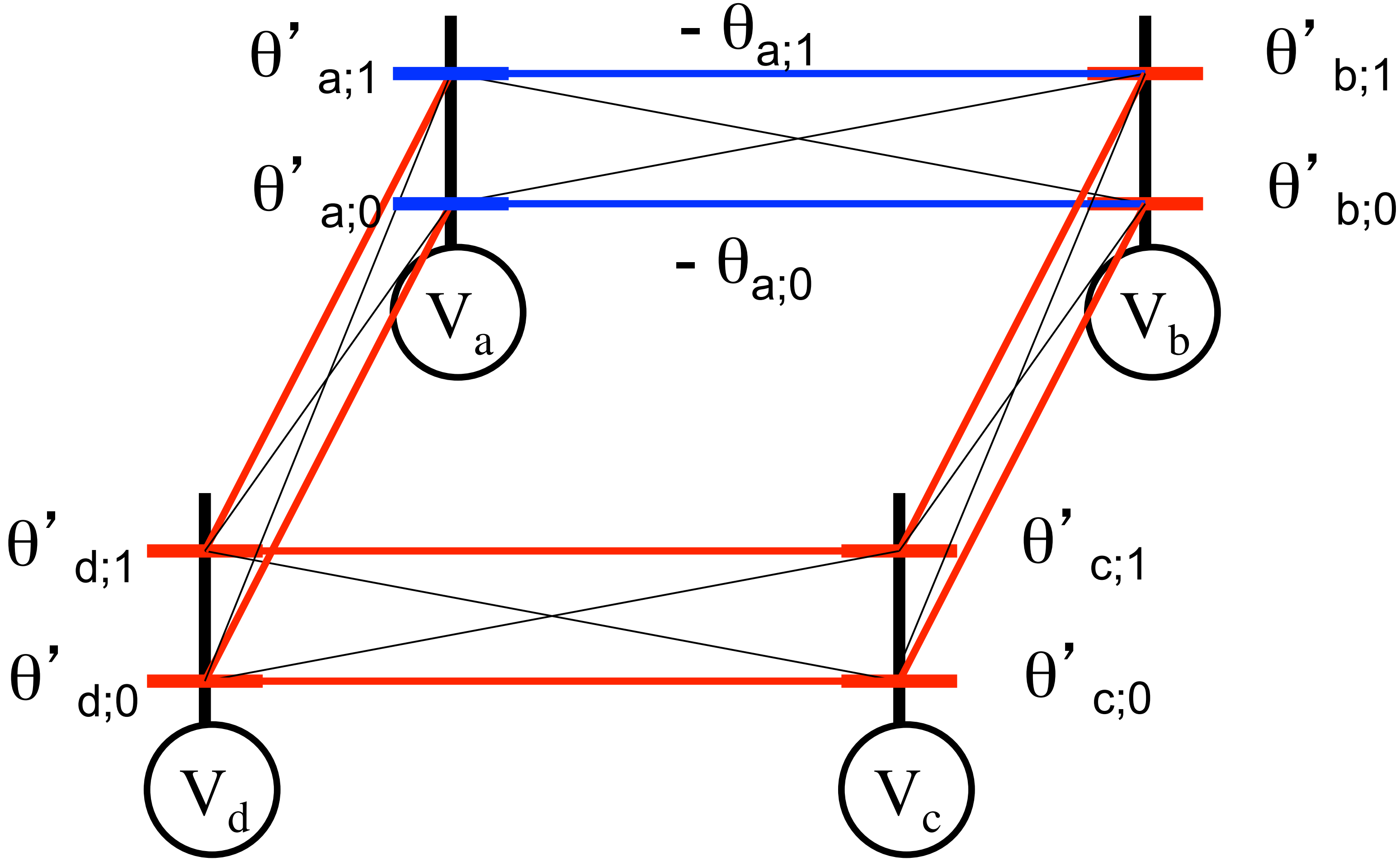
Potentials along the red path add up to 0

Belief Propagation on Cycles



Potentials along the red path add up to 0

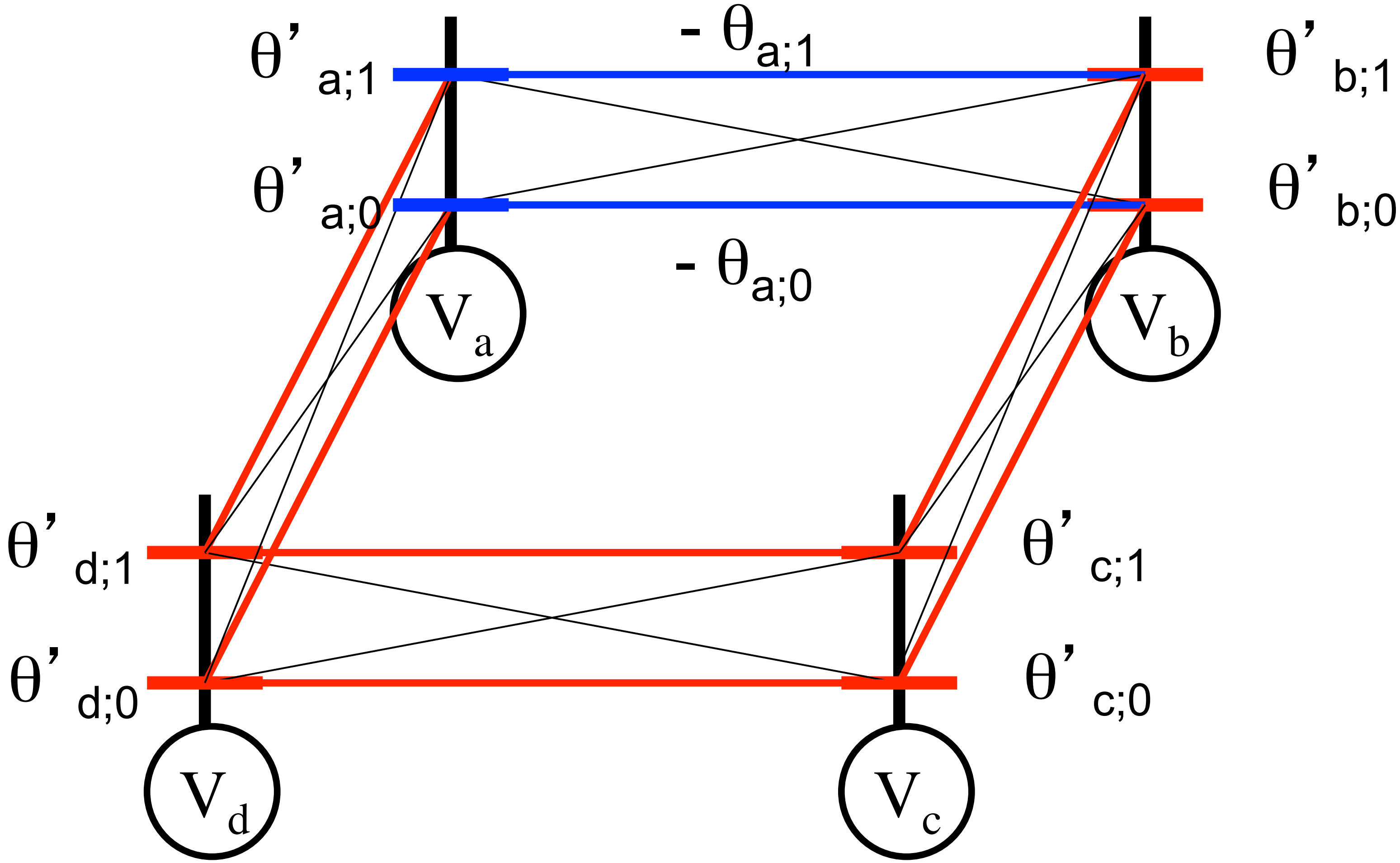
Belief Propagation on Cycles



$\theta'_{a;0} - \theta_{a;0} = q_{a;0}$ $\qquad$ $\theta'_{a;1} - \theta_{a;1} = q_{a;1}$

Potentials along the red path add up to 0

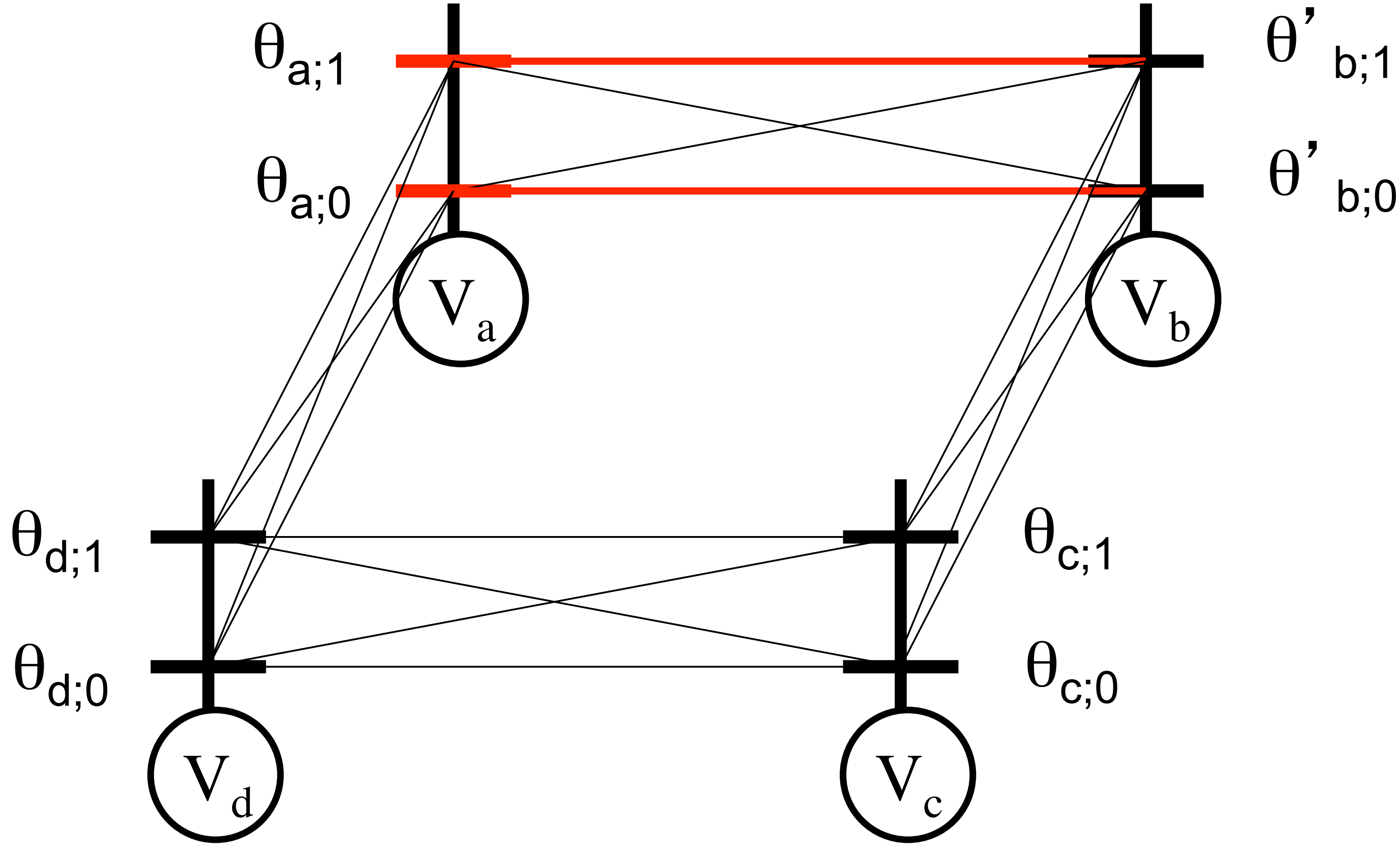
Belief Propagation on Cycles



$\theta'_{a;0} - \theta_{a;0} = q_{a;0}$ $\theta'_{a;1} - \theta_{a;1} = q_{a;1}$

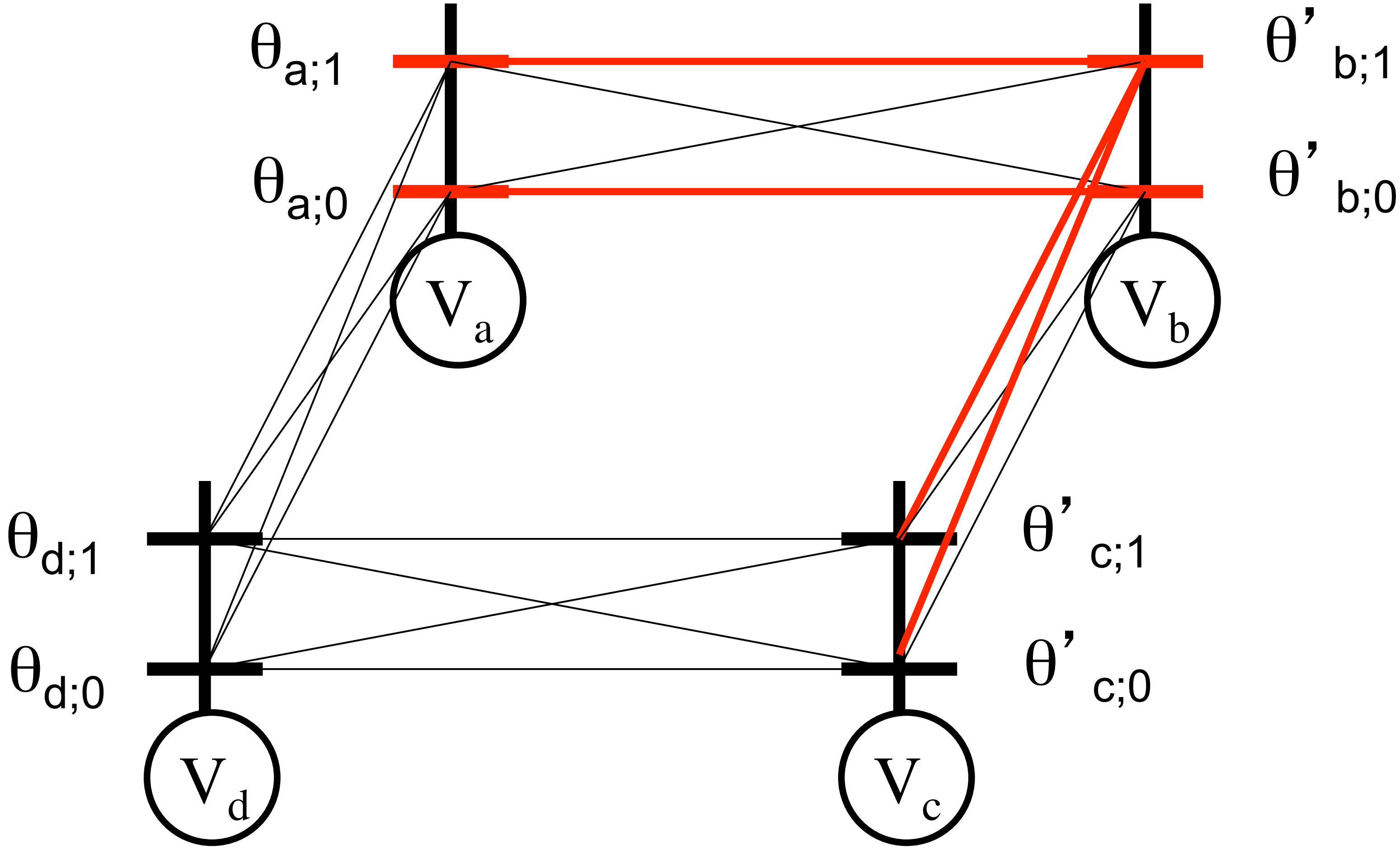
Pick minimum min-marginal. Follow red path.

Belief Propagation on Cycles



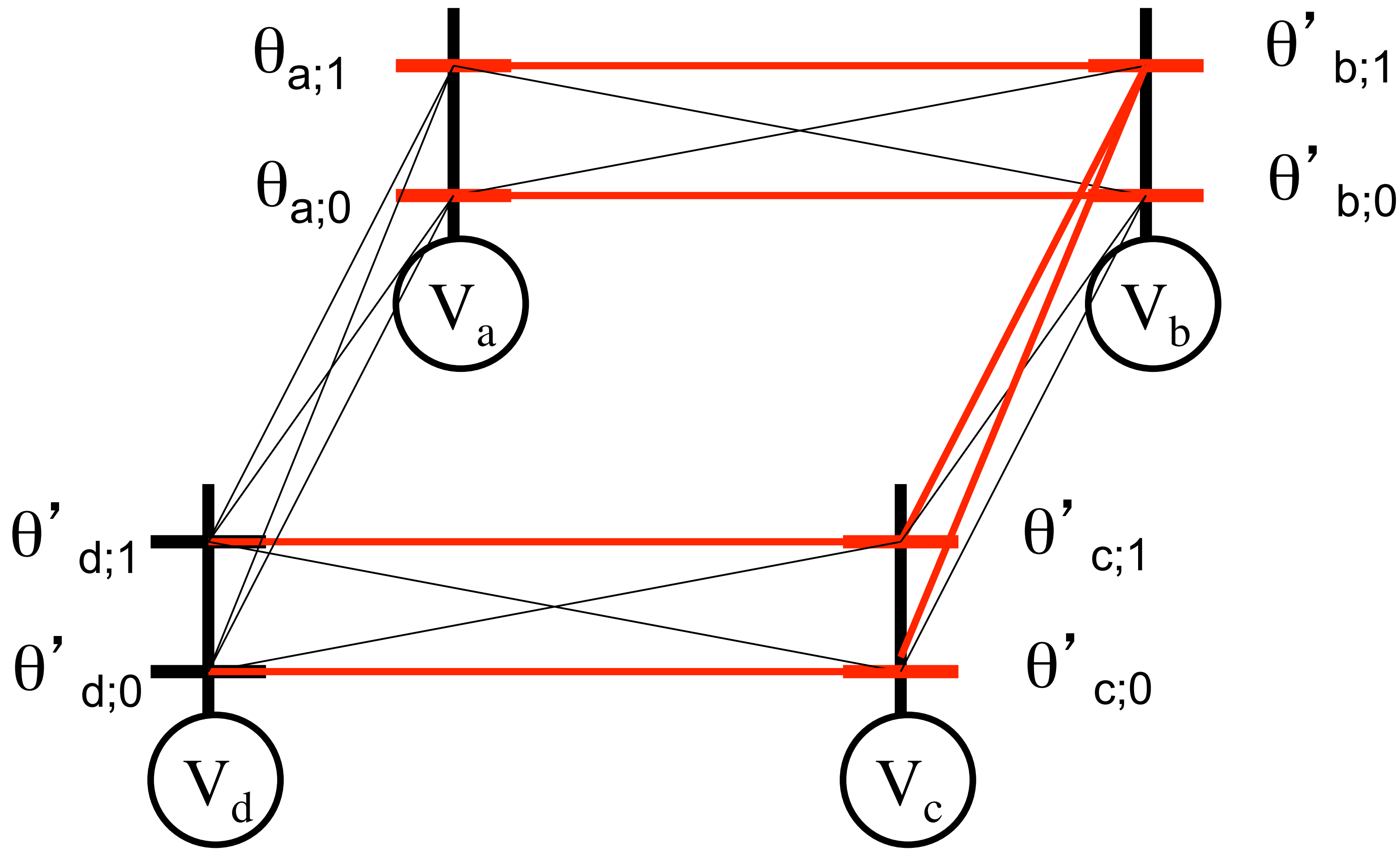
Potentials along the red path add up to 0

Belief Propagation on Cycles



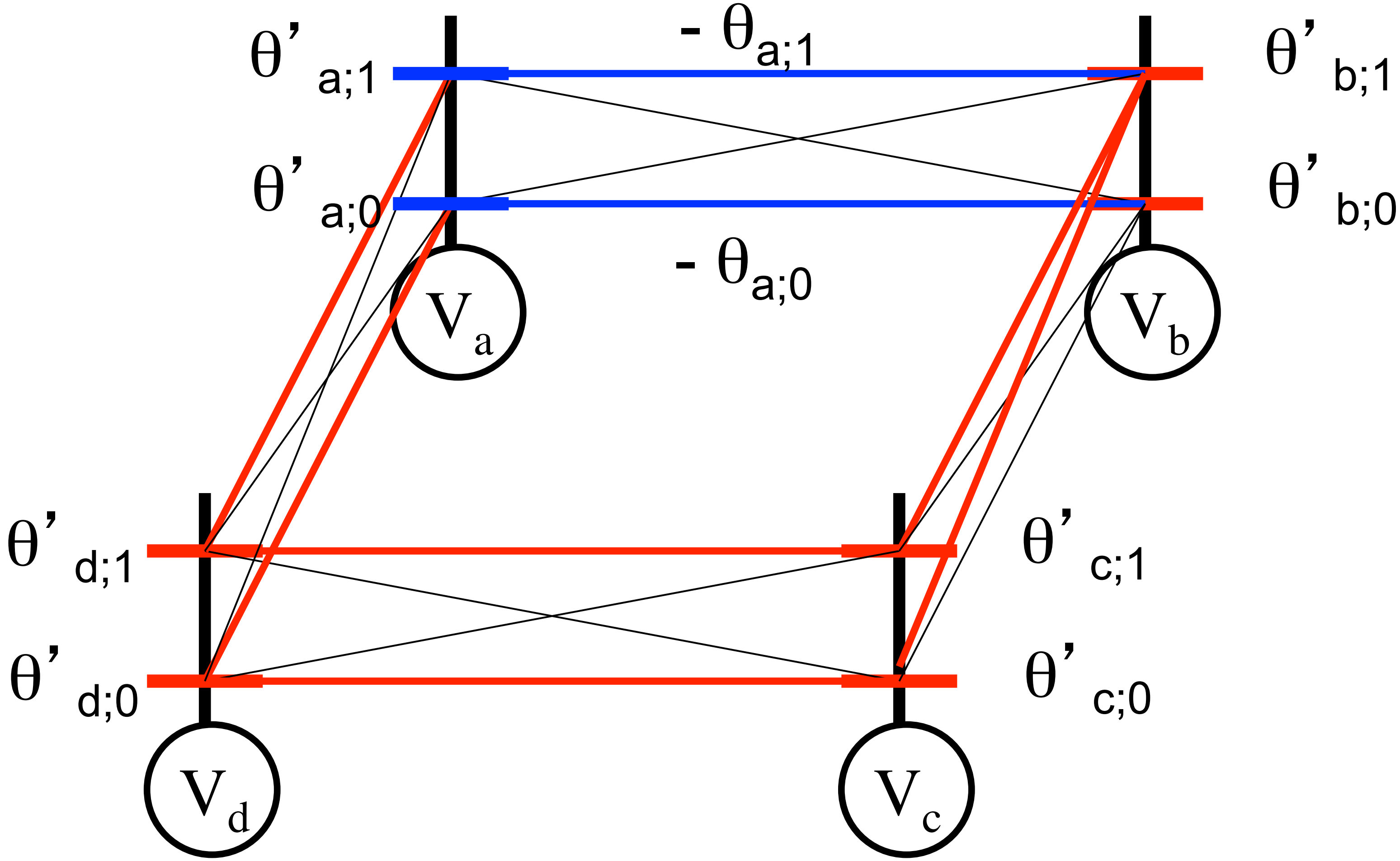
Potentials along the red path add up to 0

Belief Propagation on Cycles



Potentials along the red path add up to 0

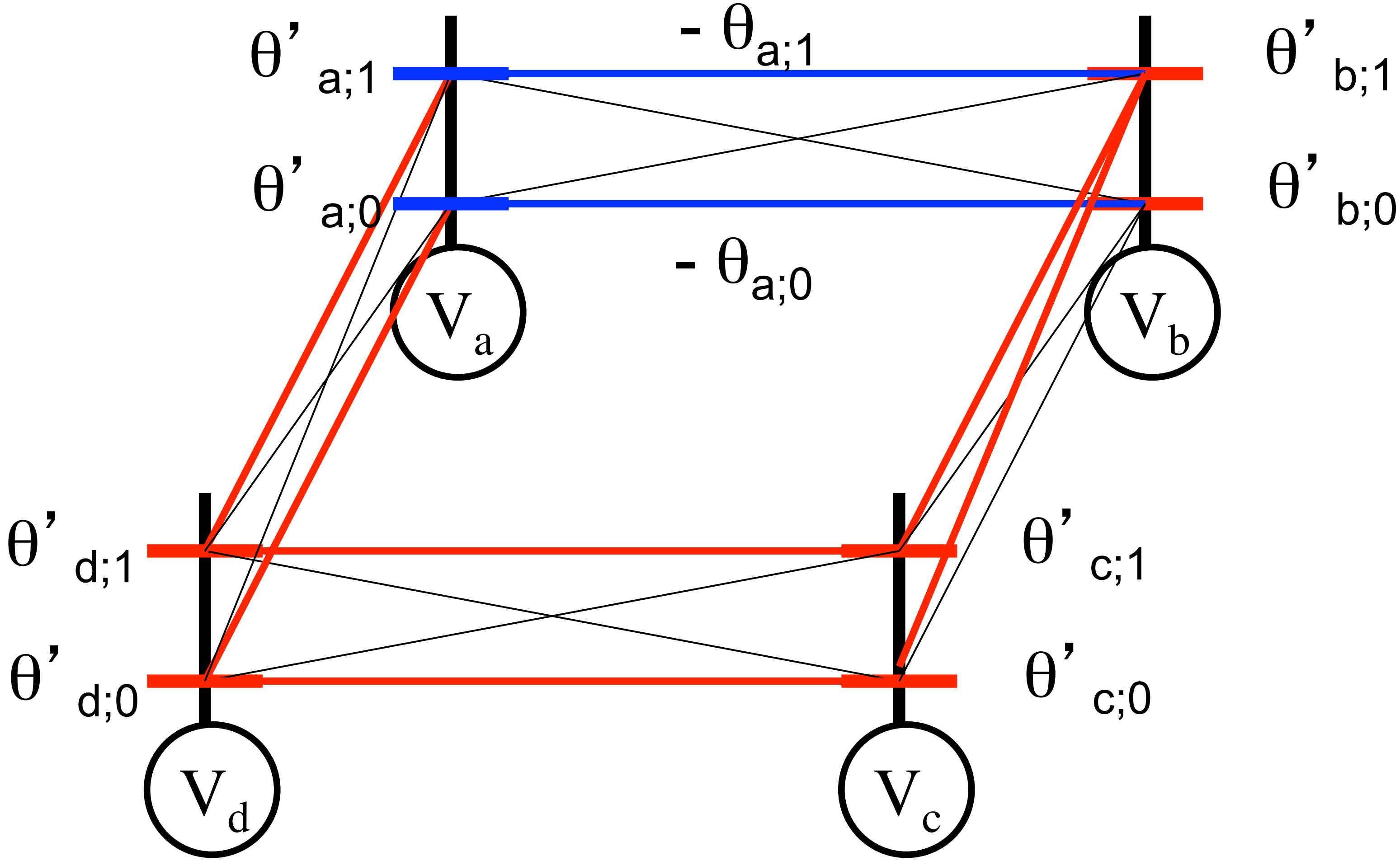
Belief Propagation on Cycles



$$\theta'_{a;1} - \theta_{a;1} = q_{a;1} \quad \leq \quad \theta'_{a;0} - \theta_{a;0} = q_{a;0}$$

Potentials along the red path add up to 0

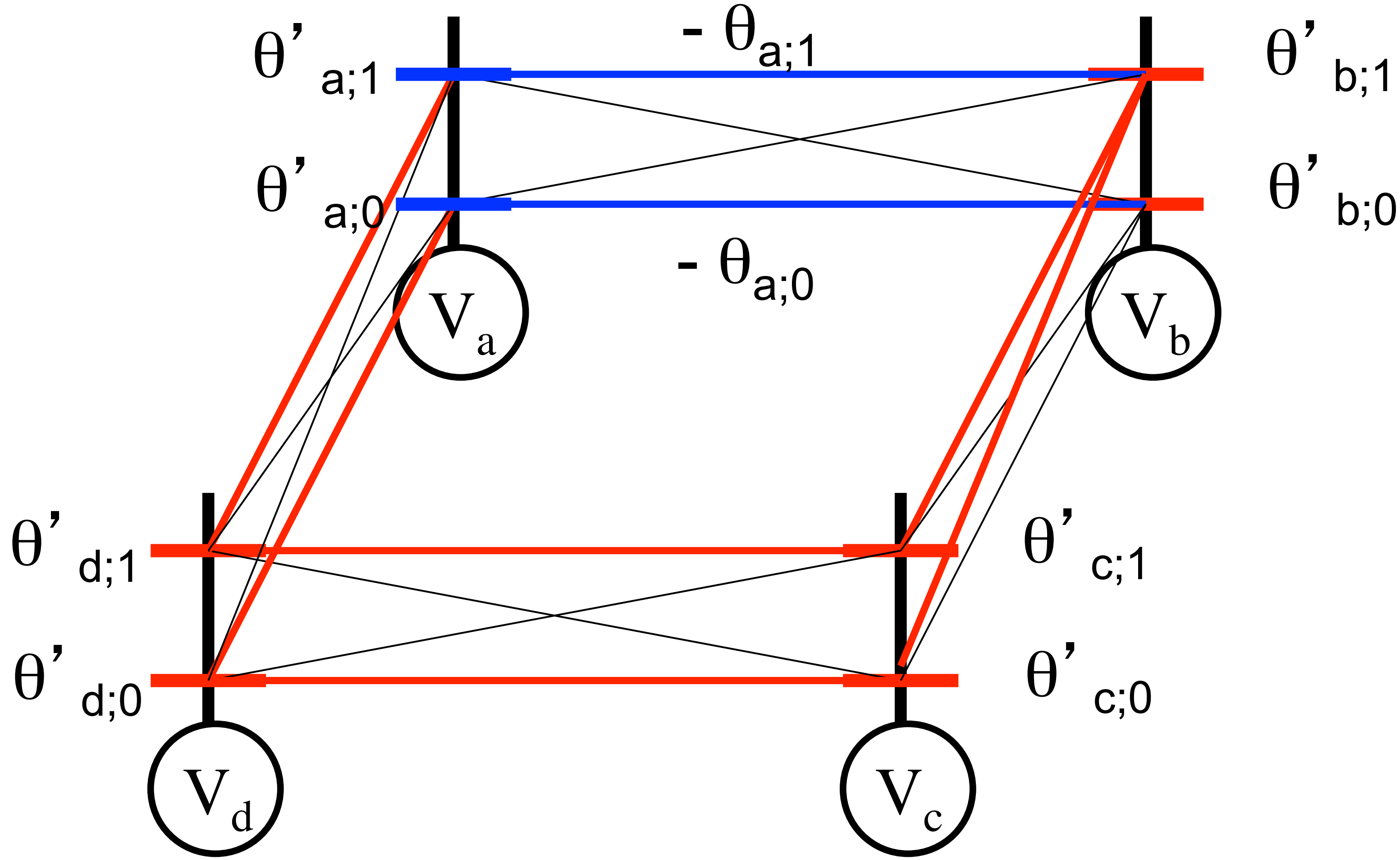
Belief Propagation on Cycles



$$\theta'_{a;1} - \theta_{a;1} = q_{a;1} \leq \theta'_{a;0} - \theta_{a;0} = q_{a;0}$$

Problem Solved

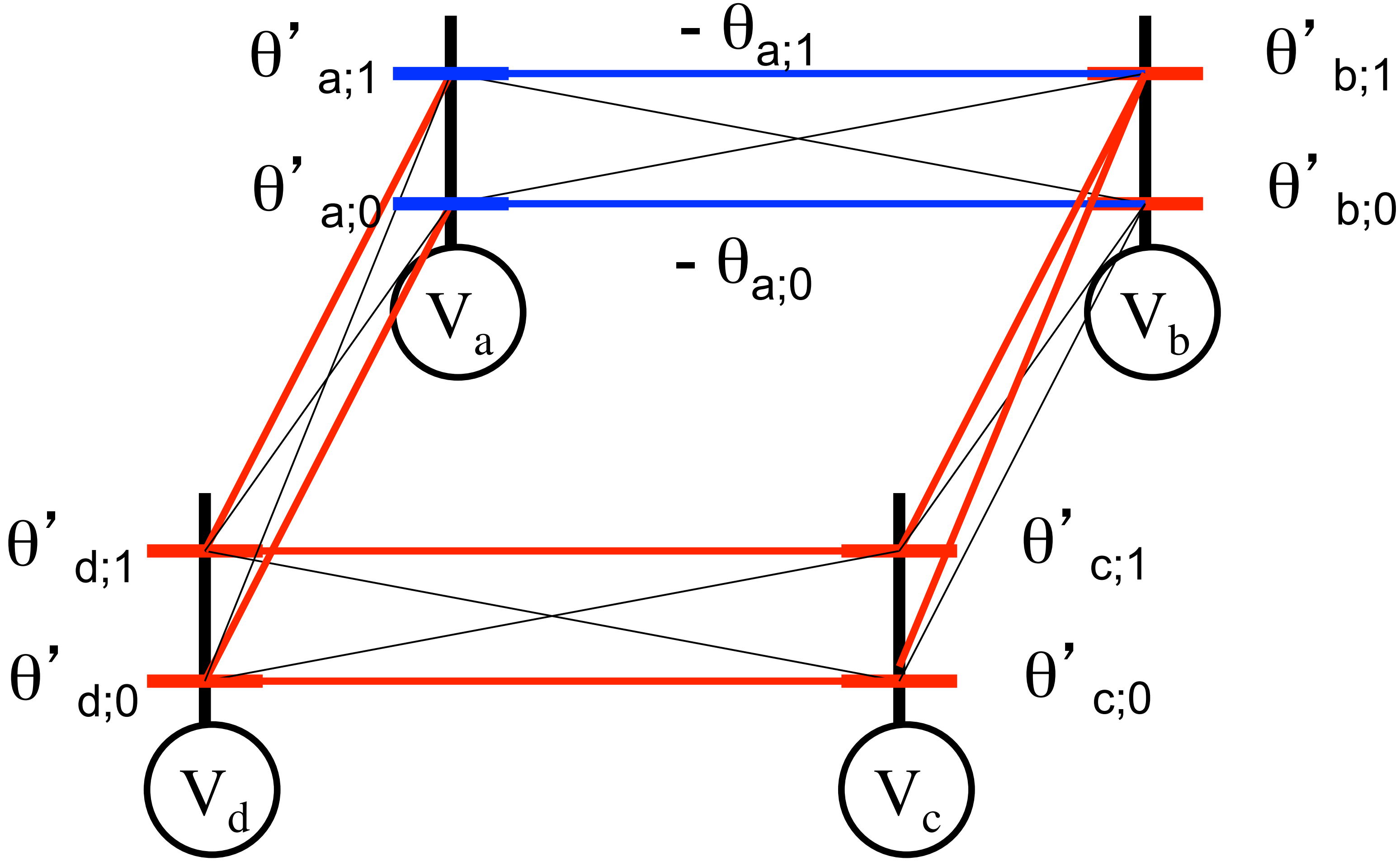
Belief Propagation on Cycles



$$\theta'_{a;1} - \theta_{a;1} = q_{a;1} \geq \theta'_{a;0} - \theta_{a;0} = q_{a;0}$$

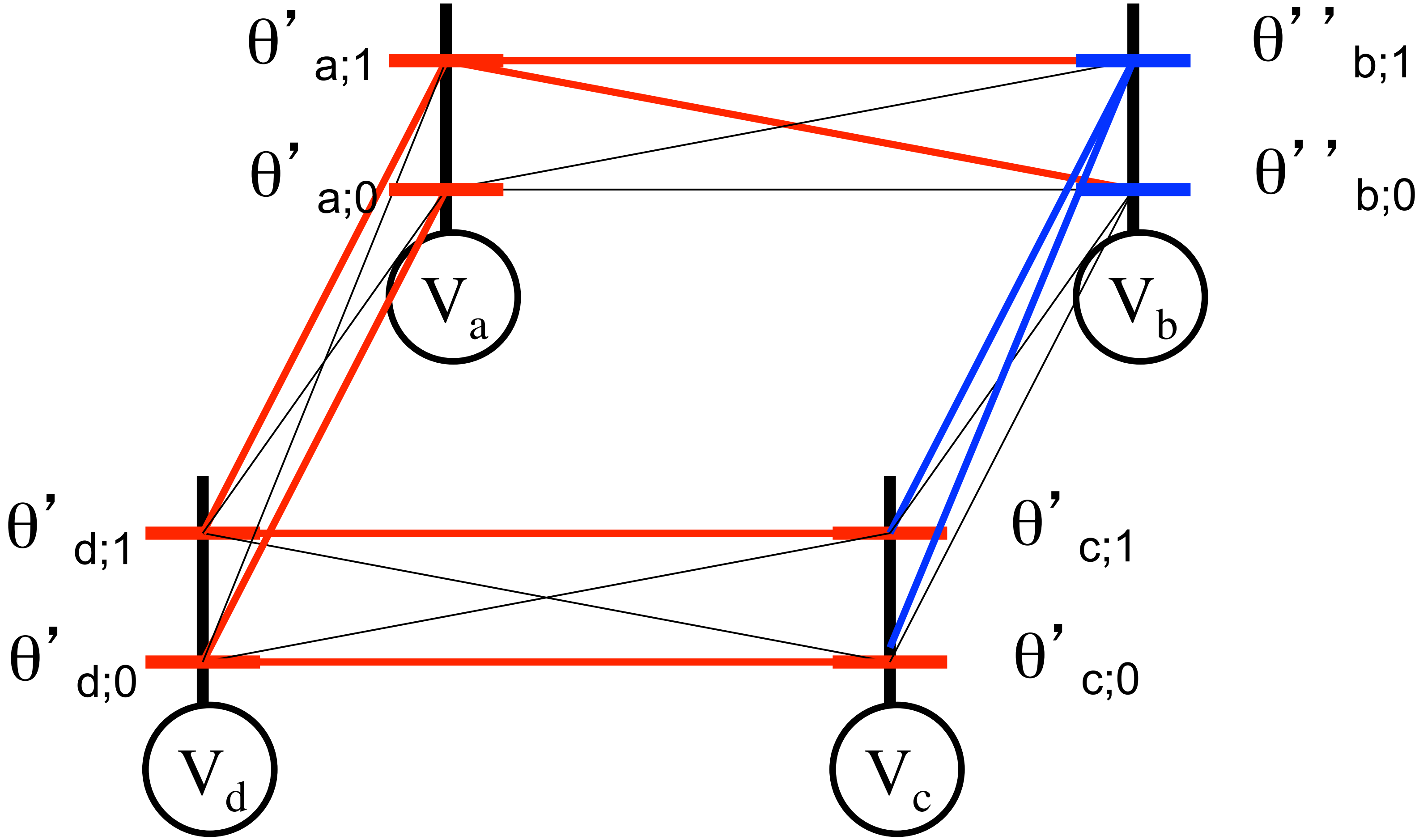
Problem Not Solved

Belief Propagation on Cycles



Reparameterize (a,b) again

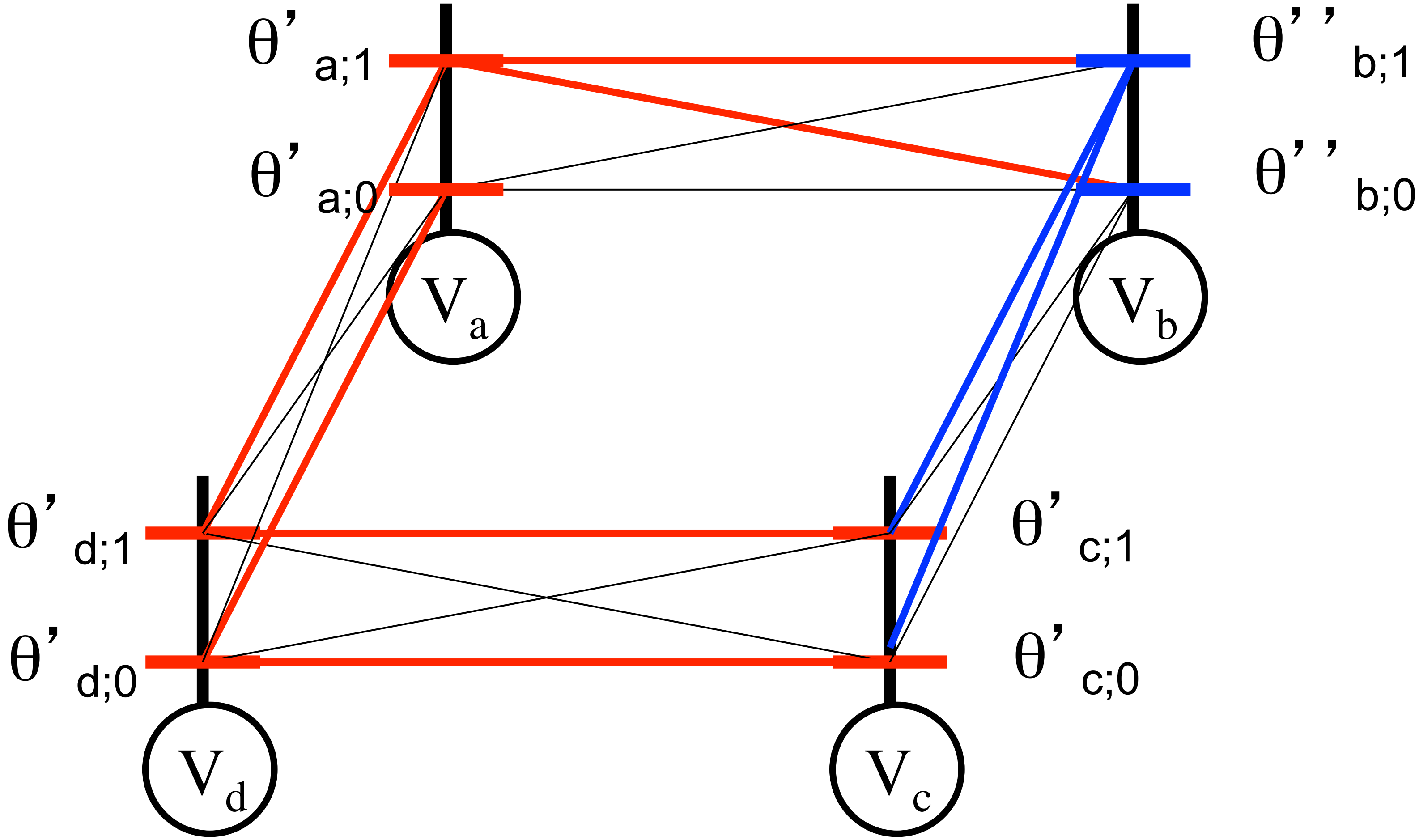
Belief Propagation on Cycles



Reparameterize (a,b) again

But doesn't this overcount some potentials?

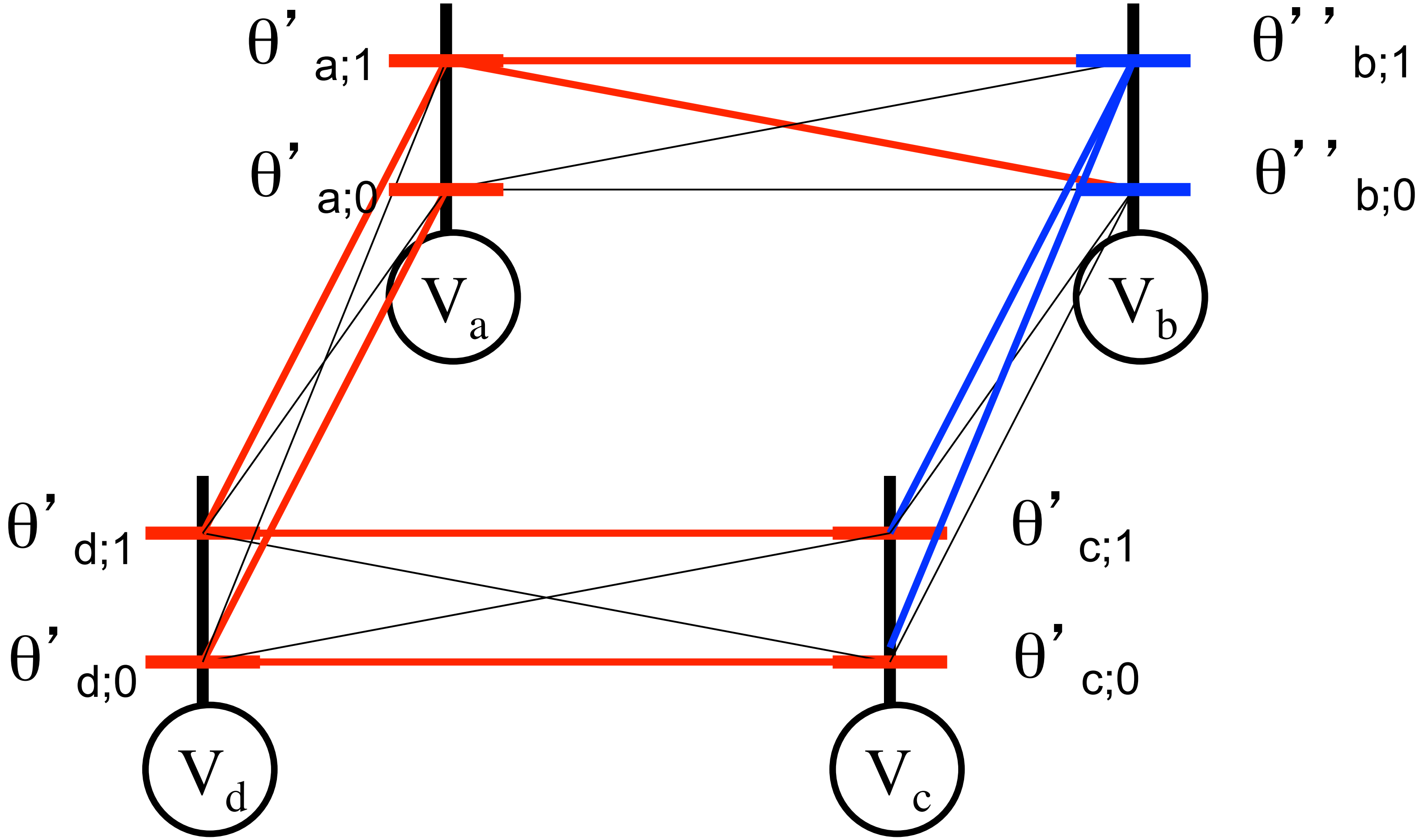
Belief Propagation on Cycles



Reparameterize (a,b) again

Yes. But we will do it anyway

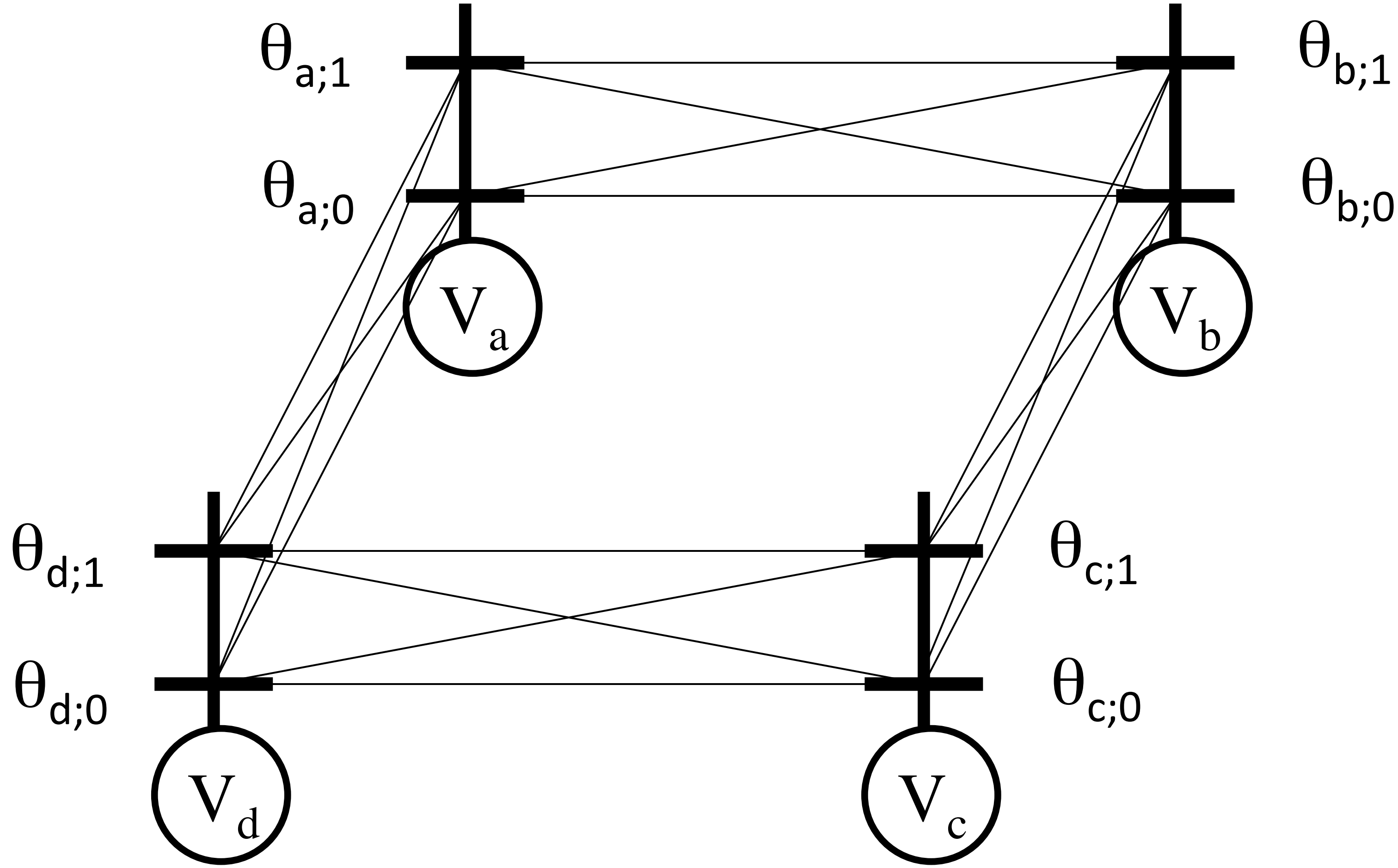
Belief Propagation on Cycles



Keep reparameterizing edges in some order

Hope for convergence and a good solution

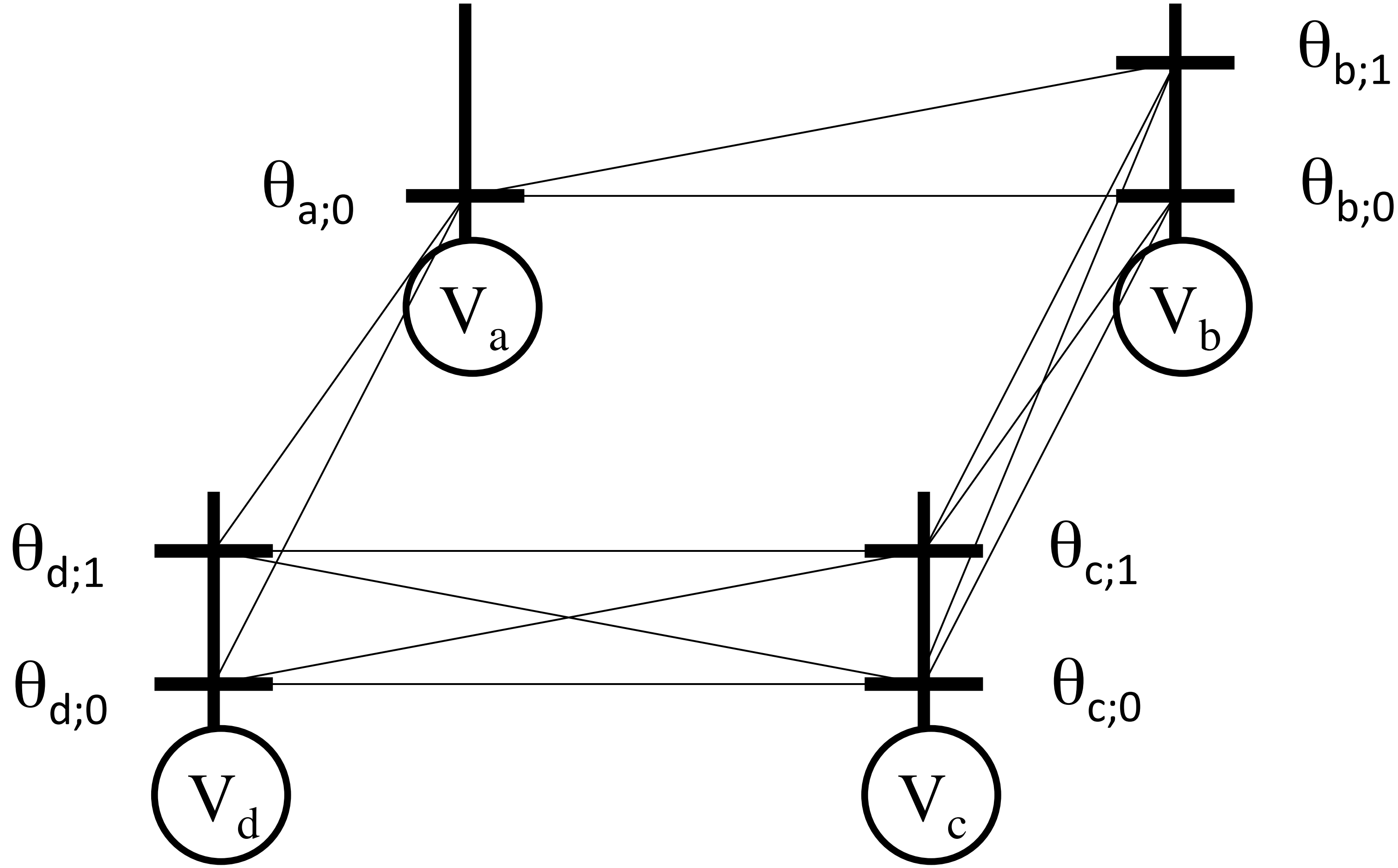
Belief Propagation on Cycles



Any suggestions?

Fix V_a to label l_0

Belief Propagation on Cycles

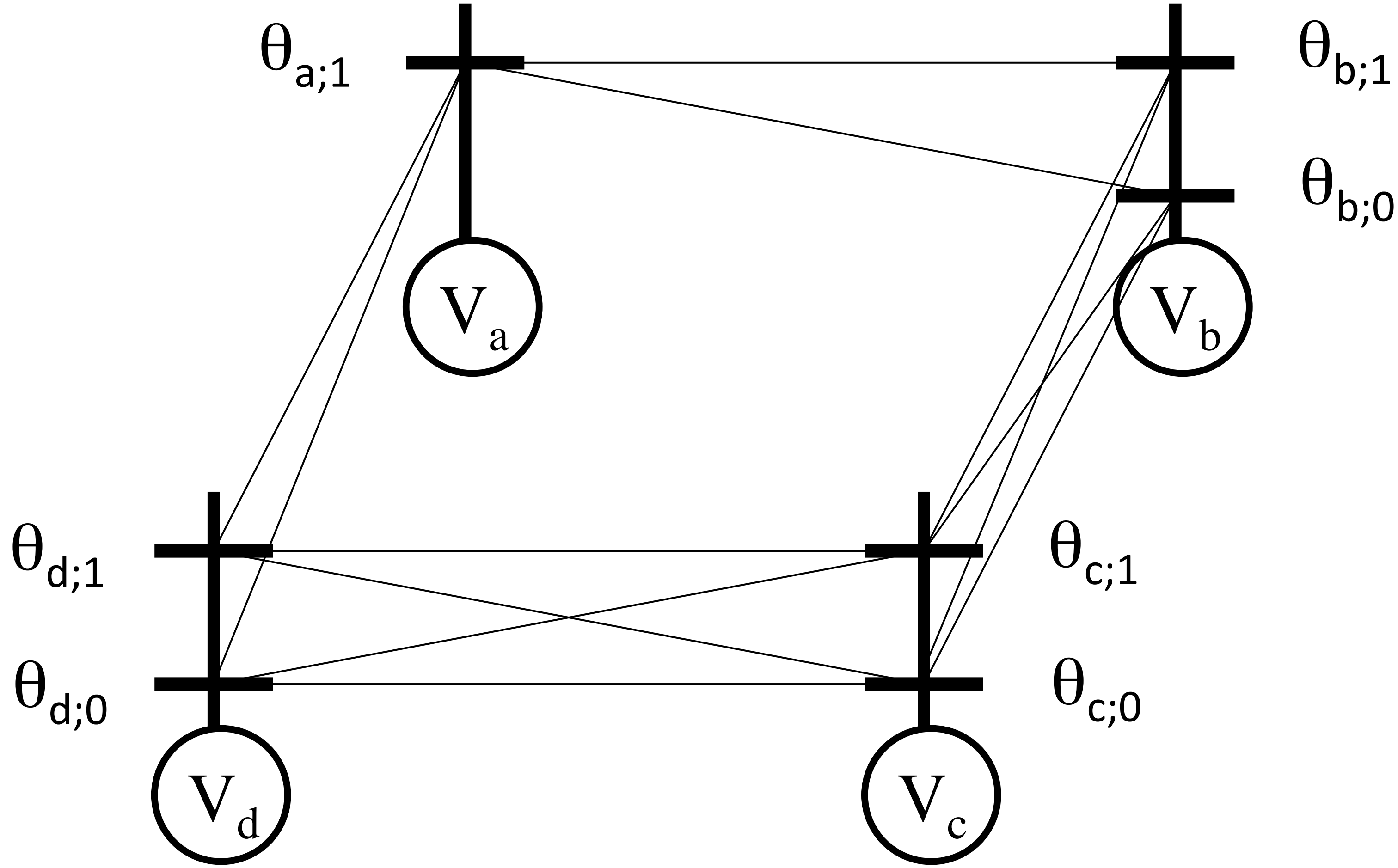


Any suggestions?

Fix V_a to label l_0

Equivalent to a tree-structured problem

Belief Propagation on Cycles

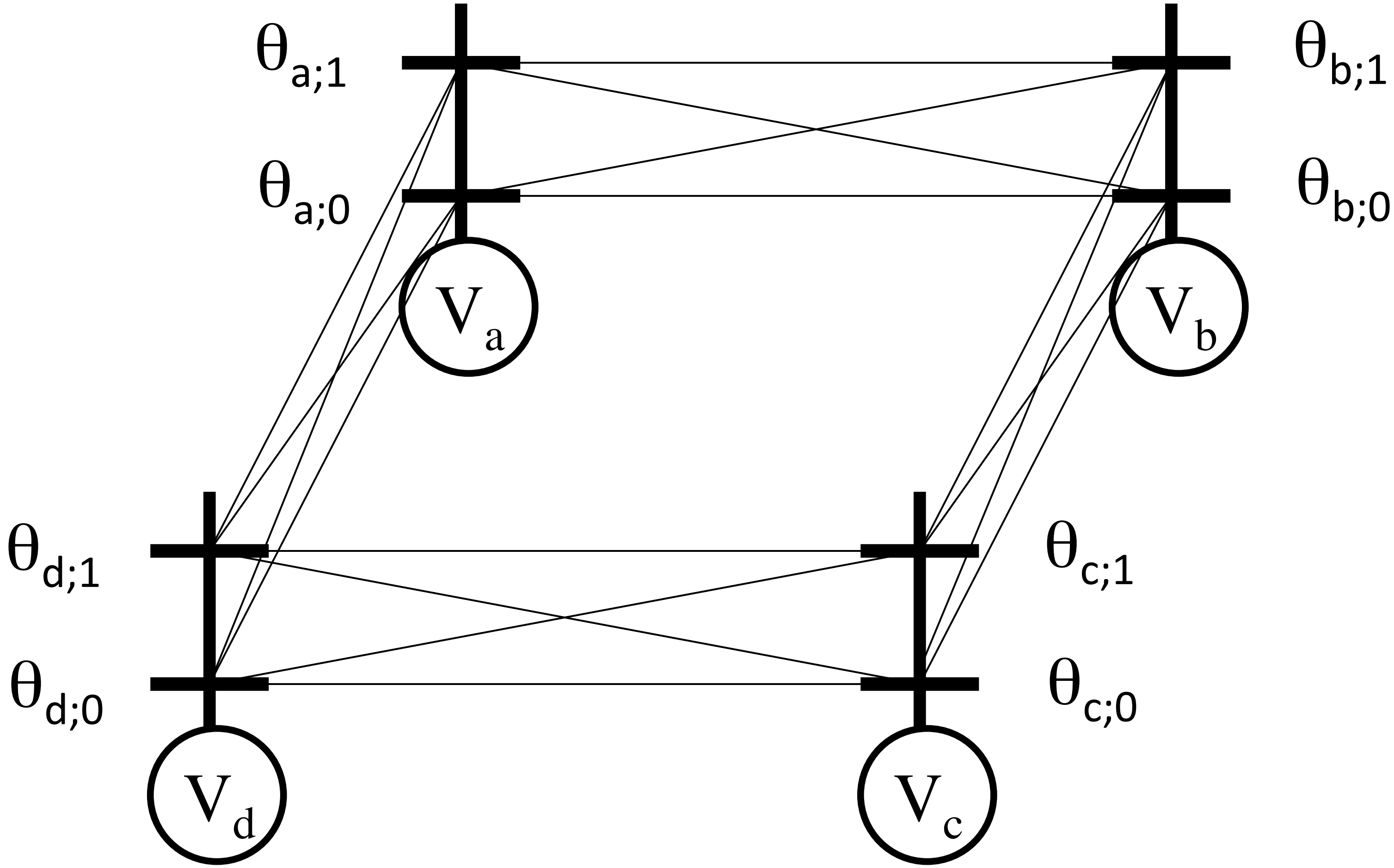


Any suggestions?

Fix V_a to label l_1

Equivalent to a tree-structured problem

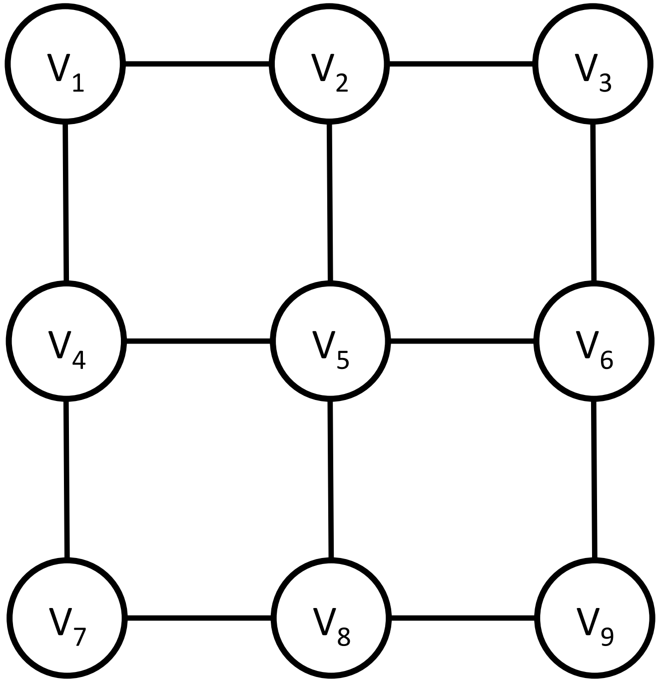
Belief Propagation on Cycles



This approach quickly becomes infeasible

Choose the minimum energy solution

Loopy Belief Propagation



Keep reparameterizing edges in some order

Hope for convergence and a good solution

Belief Propagation

- Generalizes to any arbitrary random field

- Complexity per iteration ?

$$O(|E||L|^2)$$

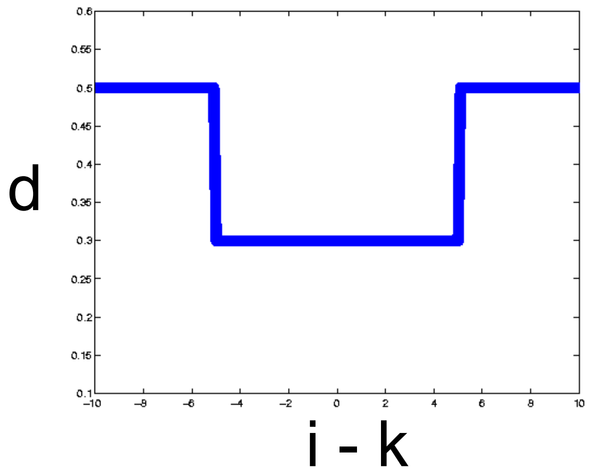
- Memory required ?

$$O(|E||L|)$$

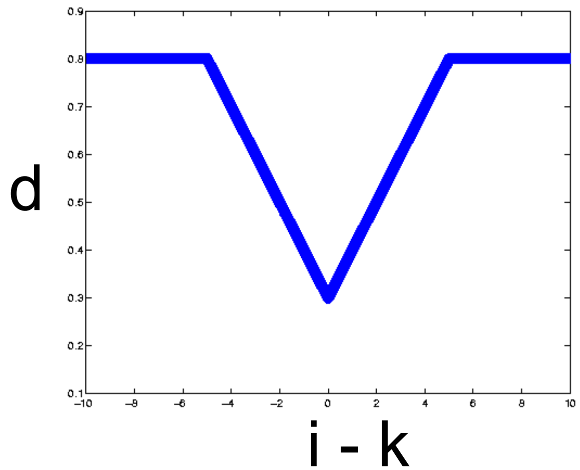
Computational Issues of BP

Complexity per iteration $O(|E||L|^2)$

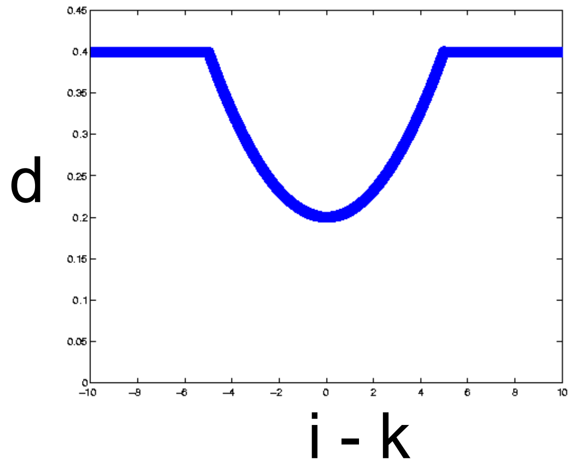
Special Pairwise Potentials $\theta_{ab;ik} = w_{ab} d(|i-k|)$



Potts



Truncated Linear



Truncated Quadratic

$O(|E||L|)$ Felzenszwalb & Huttenlocher, 2004

Computational Issues of BP

Memory requirements **$O(|E||L|)$**

Half of original BP Kolmogorov, 2006

Some approximations exist

Yu, Lin, Super and Tan, 2007

Lasserre, Kannan and Winn, 2007

But memory still remains an issue

Computational Issues of BP

Order of reparameterization

Randomly

In some fixed order

The one that results in maximum change

Residual Belief Propagation

Elidan et al., 2006

Summary of BP

Exact for chains

Exact for trees

Approximate MAP for general cases

Not even convergence guaranteed

So can we do something better?

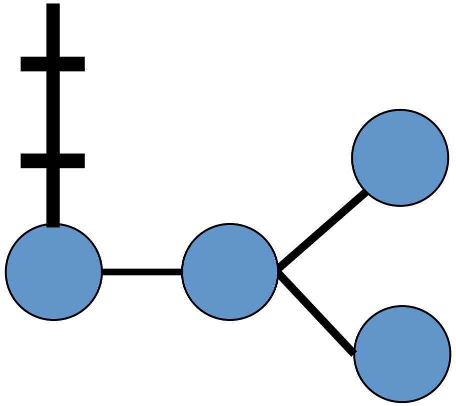
Other alternatives

- Integer linear programming and relaxation

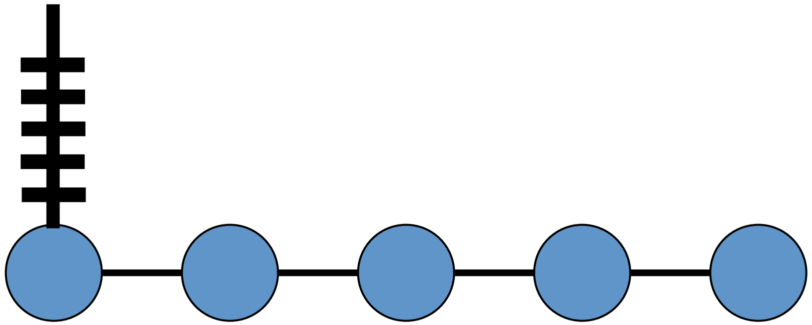
- TRW, Dual decomposition methods

- Extensively studied
    - Schlesinger, 1976
    - Koster et al., 1998, Chekuri et al., ’01, Archer et al., ’04
    - Wainwright et al., 2001, Kolmogorov, 2006
    - Globerson and Jaakkola, 2007, Komodakis et al., 2007
    - Kumar et al., 2007, Sontag et al., 2008, Werner, 2008
    - Batra et al., 2011, Werner, 2011, Zivny et al., 2014

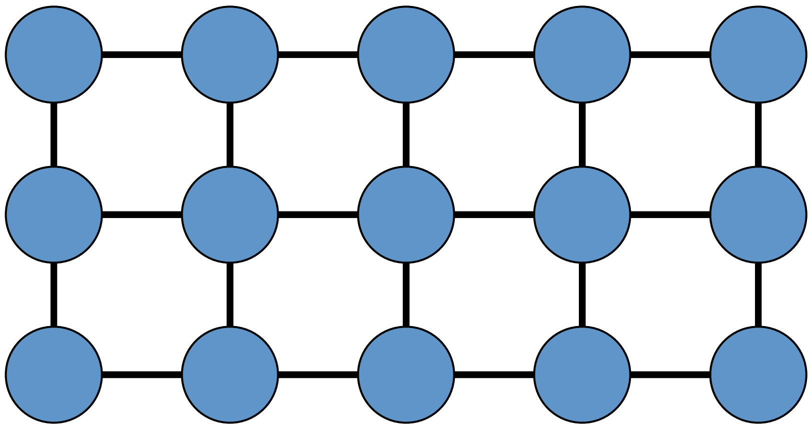
Where do we stand ?



Chain/Tree, 2-label: Use BP



Chain/Tree, multi-label: Use BP



Grid graph: Use TRW,
dual decomposition,
relaxation