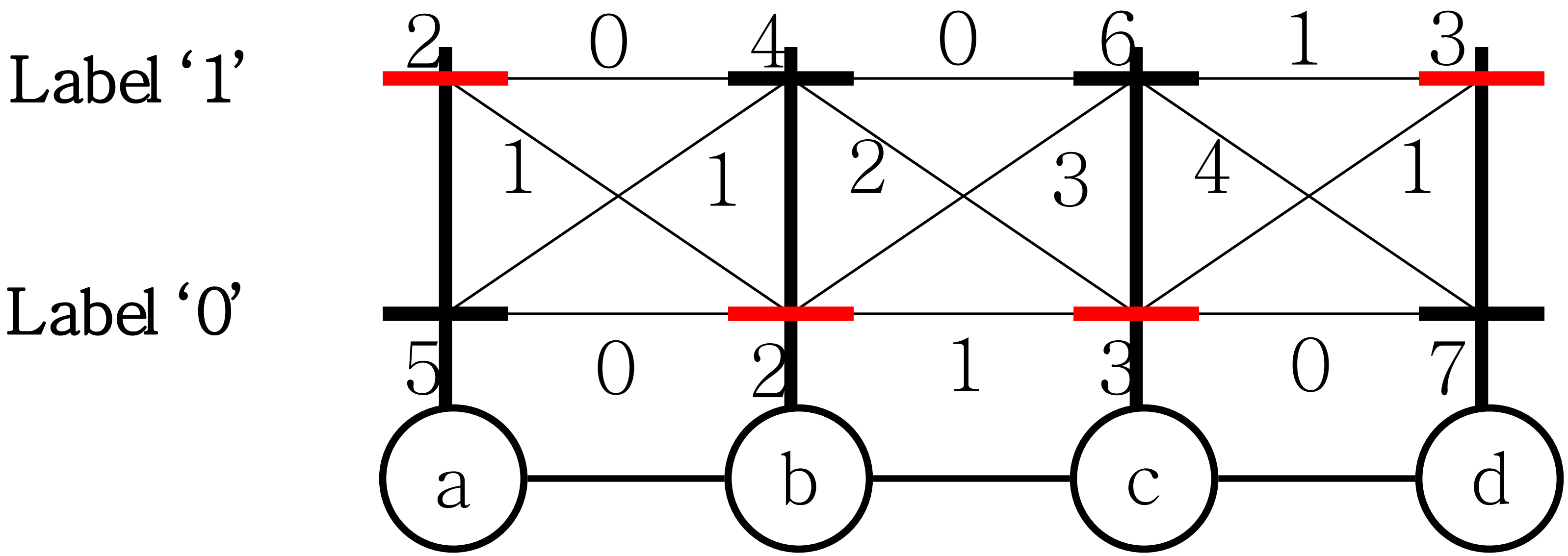
Efficiently Solving Convex Relaxations for MAP Estimation

M. Pawan Kumar
University of Oxford

Philip Torr
Oxford Brookes University

Aim

- To solve convex relaxations of MAP estimation



Random Variables V = {a, b, c, d}

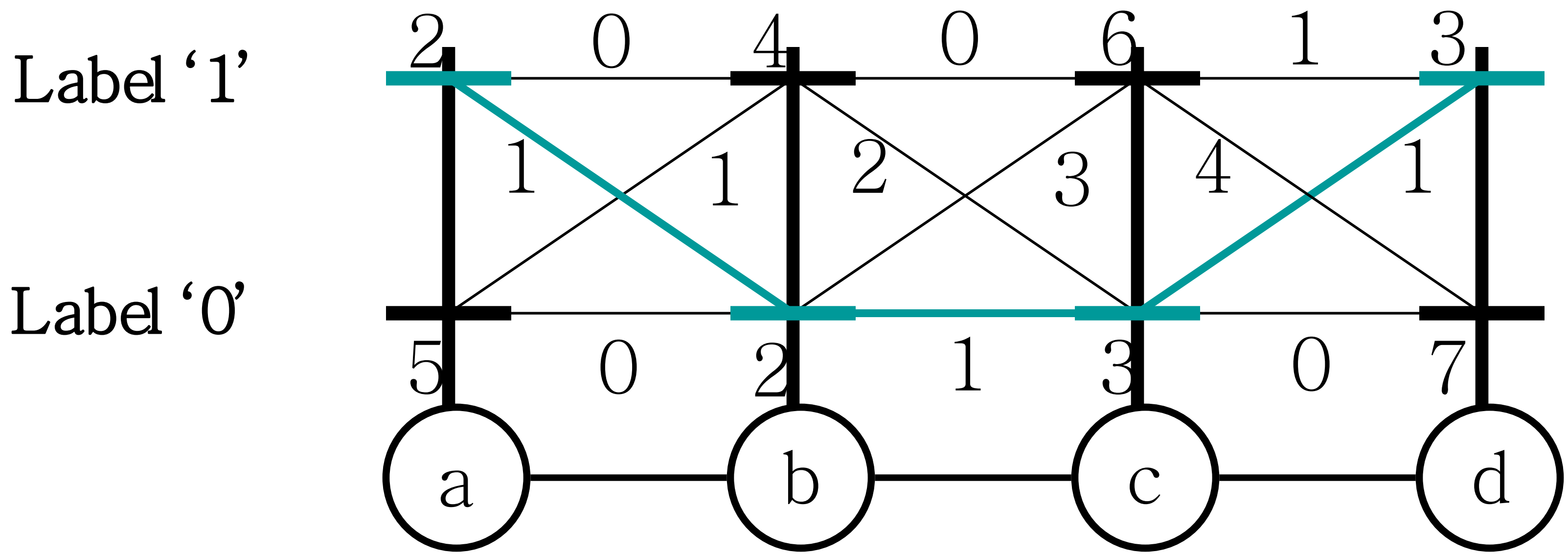
Edges E = {(a, b), (b, c), (c, d)}

Label Set L = {0, 1}

Labelling m = {1, 0, 0, 1}

Aim

- To solve convex relaxations of MAP estimation



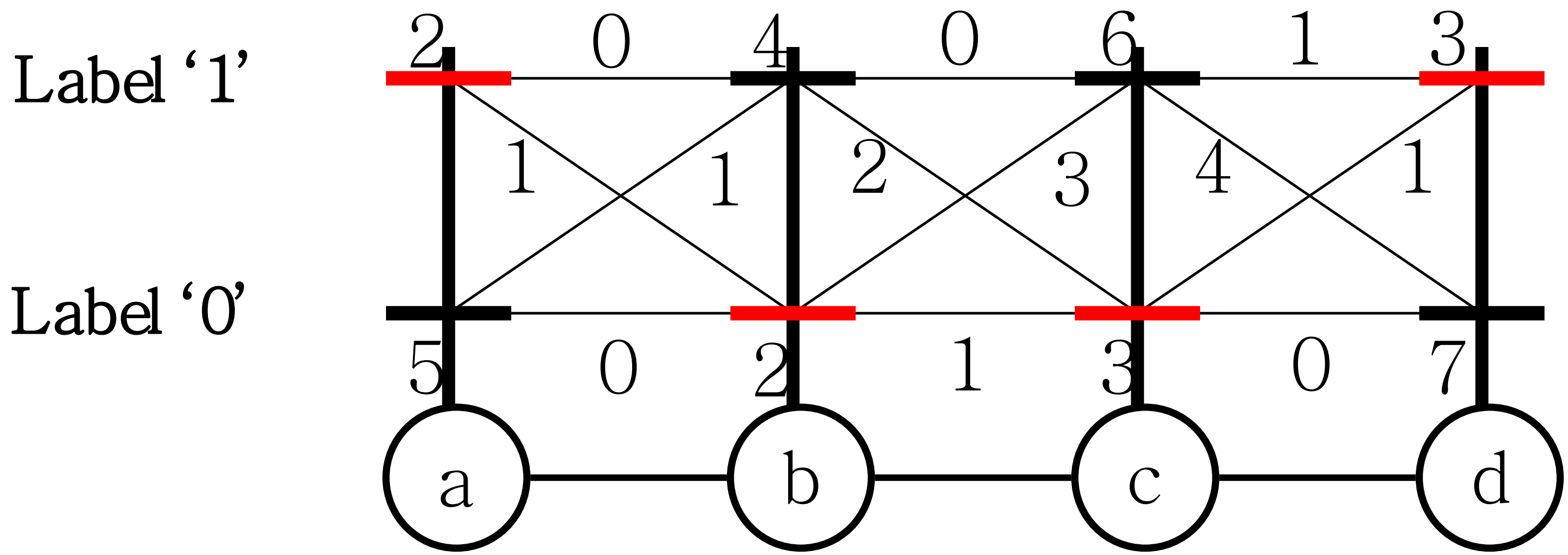
$\text{Cost(m)} = 2 + 1 + 2 + 1 + 3 + 1 + 3 = 13$

Minimum Cost Labelling? NP-hard problem

Approximate using Convex Relaxations

Aim

- To solve convex relaxations of MAP estimation



Objectives

- Solve *tighter* convex relaxations – LP and SOCP

- Handle large number of random variables, e.g. image pixels

Outline

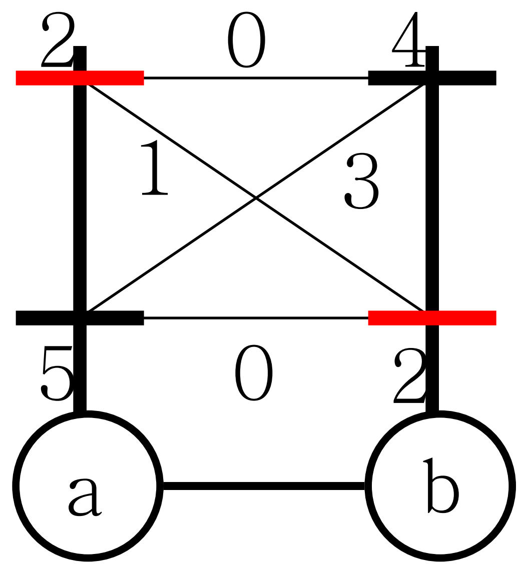
- Integer Programming Formulation
- Linear Programming Relaxation
- Additional Constraints
- Solving the Convex Relaxations
- Results and Conclusions

Integer Programming Formulation

Unary Cost

Label '1'

Label '0'



Labelling m = {1 , 0}

Unary Cost Vector **u** = [5 2 ; 2 4]

Cost of a = 0

Cost of a = 1

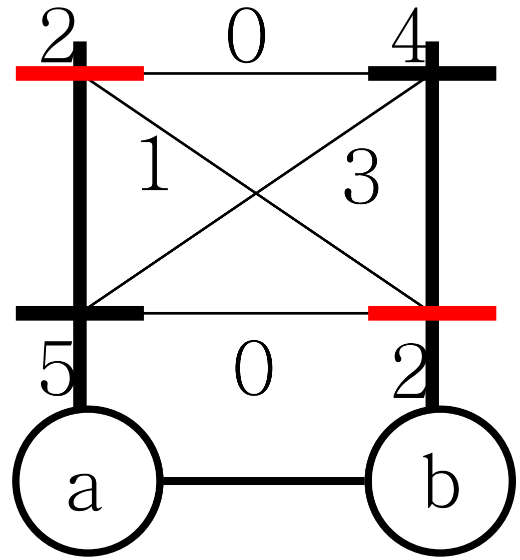
Integer Programming Formulation

Unary Cost

Label '1'

Label '0'

Labelling m = {1 , 0}



Unary Cost Vector $\mathbf{u} = [\ 5 \quad 2 \ ; \ 2 \quad 4\ ]^T$

Label vector $\mathbf{x} = [\ -1 \quad 1 \ ; \ 1 \quad -1\ ]^T$

a;0 = 1

Recall that the aim is to find the optimal **x**

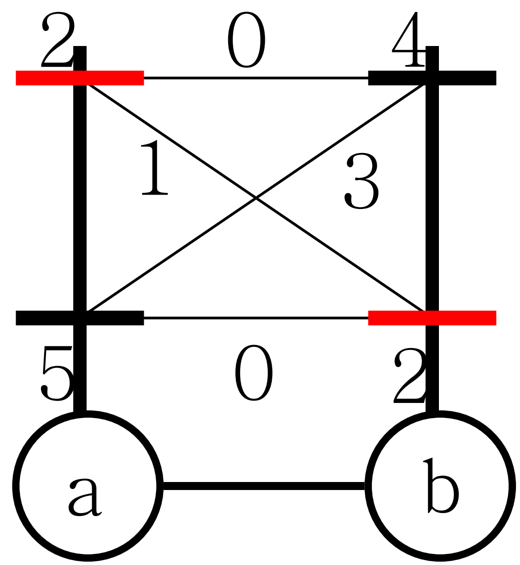
Integer Programming Formulation

Unary Cost

Label '1'

Label '0'

Labelling m = {1 , 0}



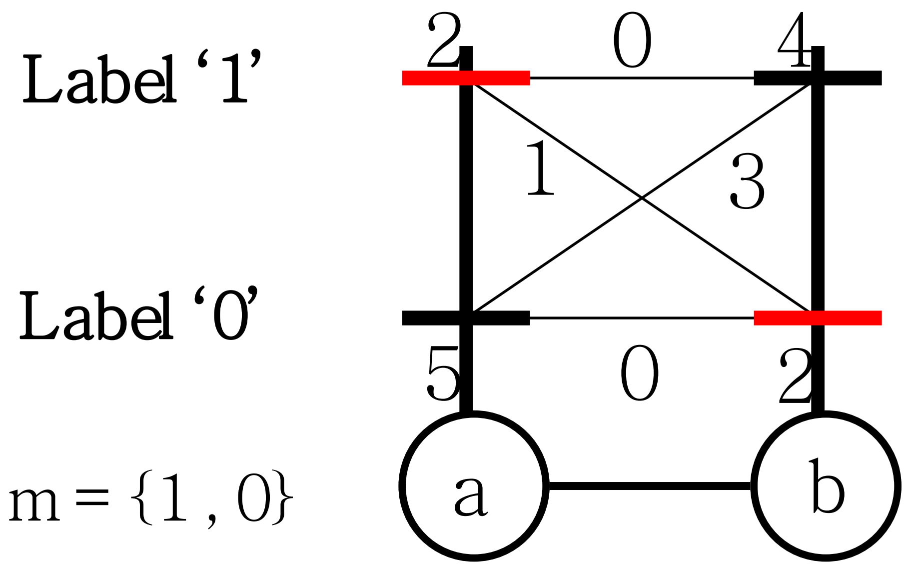
$$\text{Unary Cost Vector } \mathbf{u} = [\ 5 \quad 2 \ ; \ 2 \quad 4\ ]^T$$

$$\text{Label vector } \mathbf{x} = [\ -1 \quad 1 \ ; \ 1 \quad -1\ ]^T$$

$$\text{Sum of Unary Costs} = \frac{1}{2} \sum_i \mathbf{u}_i (1 + \mathbf{x}_i)$$

Integer Programming Formulation

Pairwise Cost



Labelling m = {1 , 0}

Pairwise Cost Matrix **P**

0	0	0	3
0	0	1	0
0	1	0	0
3	0	0	0

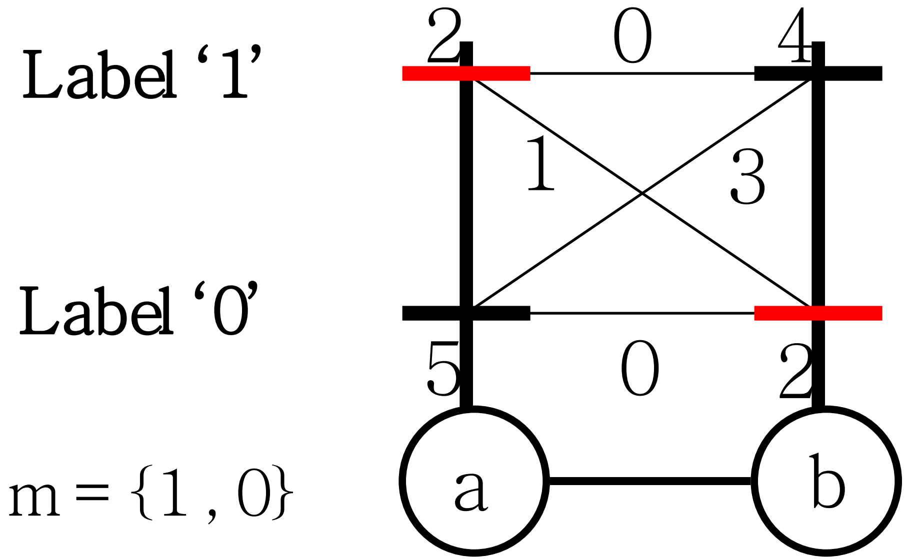
Pairwise Cost of a and a

Cost of a = 0 and b = 0

Cost of a = 0 and b = 1

Integer Programming Formulation

Pairwise Cost



Labelling m = {1 , 0}

Pairwise Cost Matrix **P**

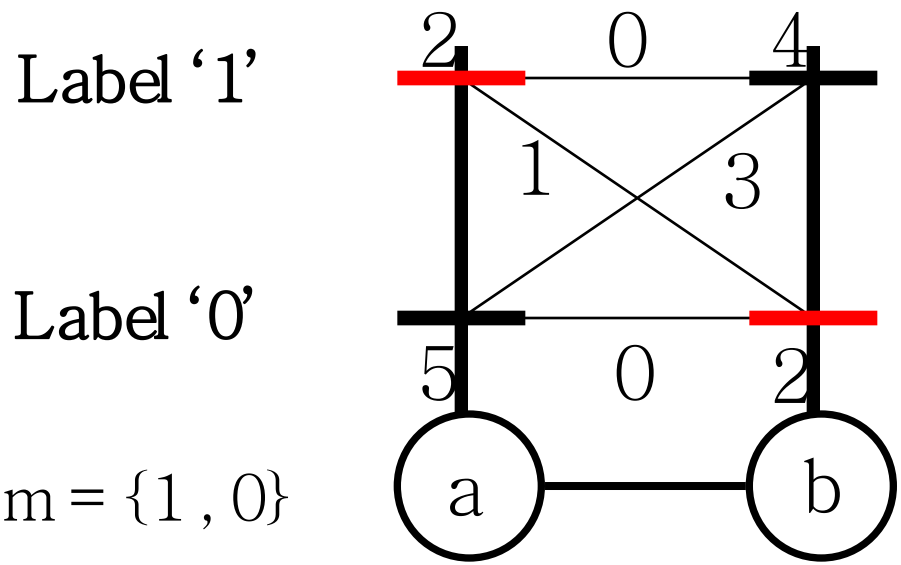
0	0	0	3
0	0	1	0
0	1	0	0
3	0	0	0

Sum of Pairwise Costs

$$\frac{1}{4} \sum_{ij} \mathbf{P}_{ij} (1 + \mathbf{x}_i)(1+\mathbf{x}_j)$$

Integer Programming Formulation

Pairwise Cost



Labelling m = {1 , 0}

Pairwise Cost Matrix **P**

0	0	0	3
0	0	1	0
0	1	0	0
3	0	0	0

Sum of Pairwise Costs

$$\frac{1}{4} \sum_{ij} P_{ij} (1 + \mathbf{x}_i + \mathbf{x}_j + \mathbf{x}_i\mathbf{x}_j)$$

$$= \frac{1}{4} \sum_{ij} P_{ij} (1 + \mathbf{x}_i + \mathbf{x}_j + X_{ij})$$

$X = \mathbf{x}\,\mathbf{x}^T \qquad X_{ij} = \mathbf{x}_i\,\mathbf{x}_j$

Integer Programming Formulation

Constraints

- Integer Constraints

$$\mathbf{x}_i \in \{-1, 1\}$$

$$X = \mathbf{x}\,\mathbf{x}^T$$

- Uniqueness Constraint

$$\sum_{i \in a} \mathbf{x}_i = 2 - |L|$$

Integer Programming Formulation

$$\mathbf{x}^* = \operatorname{argmin} \quad \frac{1}{2} \sum \mathbf{u}_i (1 + \mathbf{x}_i) + \frac{1}{4} \sum P_{ij} (1 + \mathbf{x}_i + \mathbf{x}_j + X_{ij})$$

$$\sum_{i \in a} \mathbf{x}_i = 2 - |L|$$

Convex

$$\mathbf{x}_i \in \{-1, 1\}$$

$$X = \mathbf{x}\,\mathbf{x}^T$$

Non-Convex

Outline

- Integer Programming Formulation
- Linear Programming Relaxation
- Additional Constraints
- Solving the Convex Relaxations
- Results and Conclusions

Linear Programming Relaxation

Schlesinger, 1976

Retain Convex Part

$$\mathbf{x}_i \in \{-1,1\}$$

$$X = \mathbf{x}\,\mathbf{x}^T$$

Linear Programming Relaxation

Schlesinger, 1976

Retain Convex Part

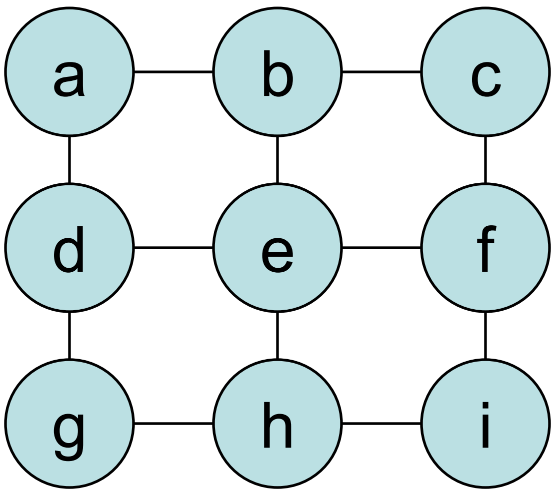
$$x_i \in [-1, 1]$$

$$X_{ij} \in [-1, 1] \qquad 1 + x_i + x_j + X_{ij} \geq 0$$

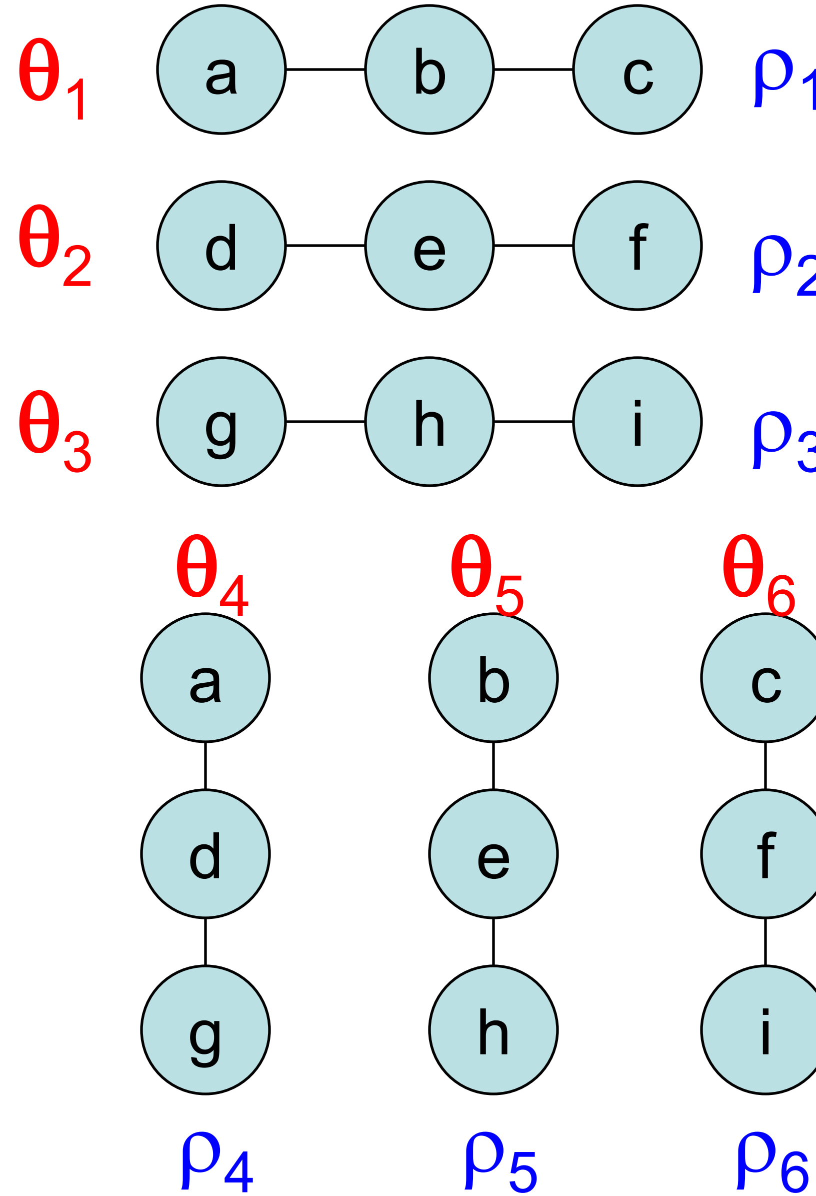
$$\sum_{j \in b} X_{ij} = (2 - |L|)\, x_i$$

Dual of the LP Relaxation

Wainwright et al., 2001



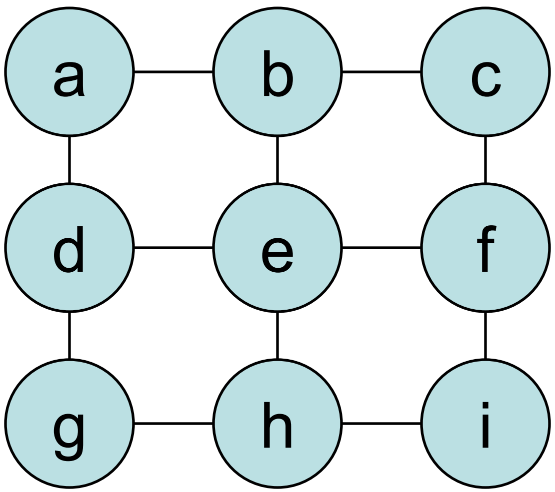
$\theta = (\mathbf{u}, \mathbf{P})$



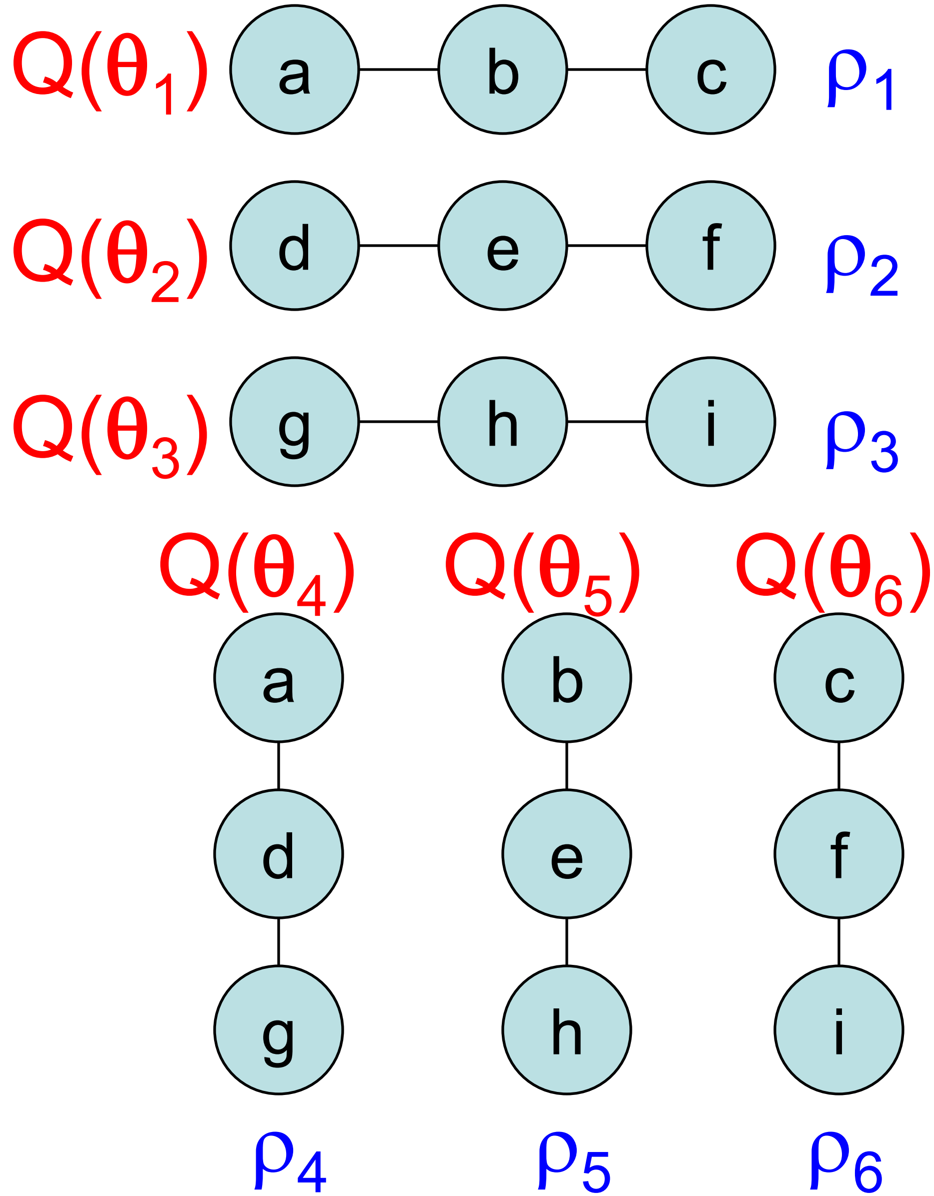
$$\sum \rho_i \theta_i \equiv \theta$$

Dual of the LP Relaxation

Wainwright et al., 2001



$\boldsymbol{\theta} = (\mathbf{u}, \mathbf{P})$



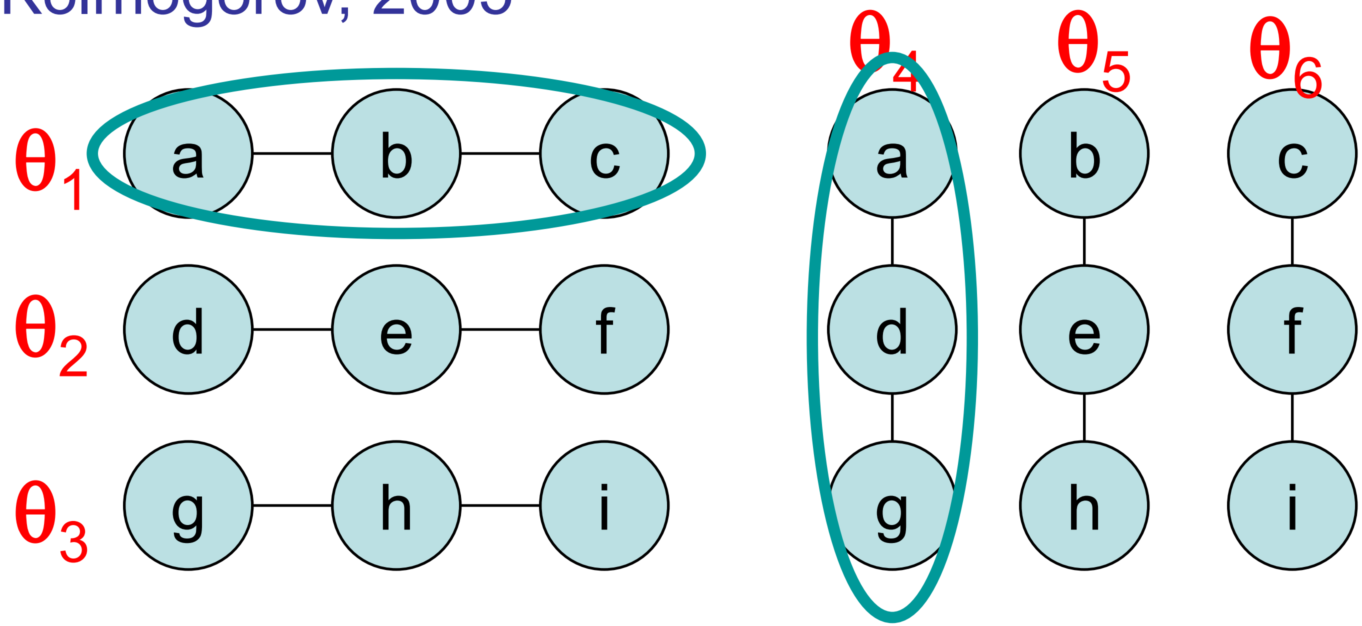
Dual of LP

$$\max \sum \rho_i Q(\theta_i)$$

$$\sum \rho_i \theta_i \equiv \theta$$

Tree-Reweighted Message Passing

Kolmogorov, 2005



Pick a variable

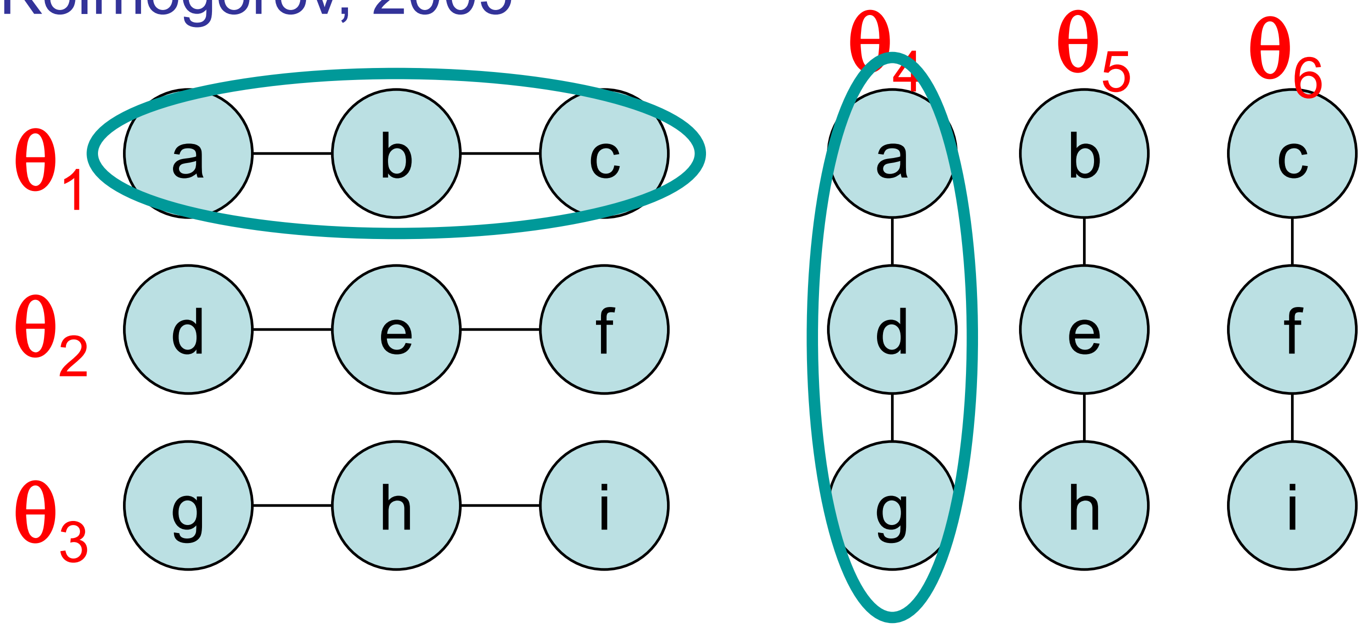
a

Reparameterize such that u_i are min-marginals

Only one pass of belief propagation

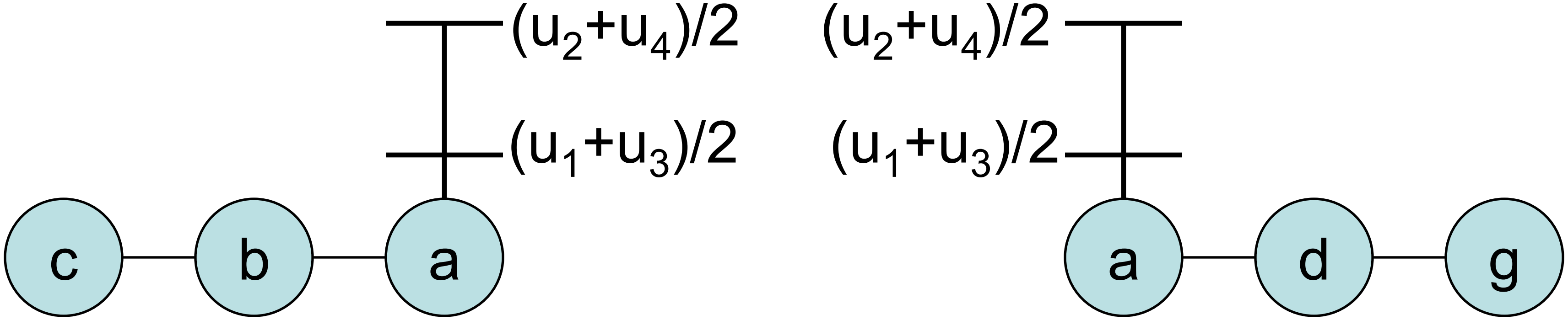
Tree-Reweighted Message Passing

Kolmogorov, 2005



Pick a variable

a



Average the unary costs

Repeat for all variables

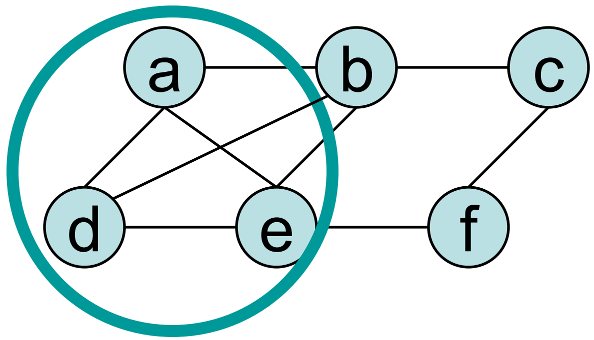
TRW-S

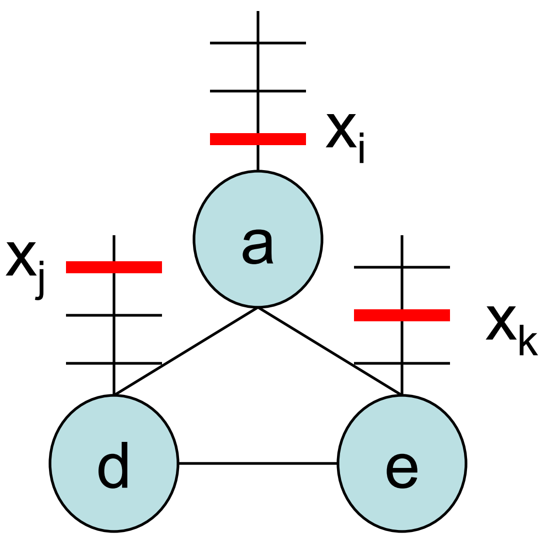
Outline

- Integer Programming Formulation
- Linear Programming Relaxation
- Additional Constraints
- Solving the Convex Relaxations
- Results and Conclusions

Cycle Inequalities

Chopra and Rao, 1991





At least two of them have the same sign

$x_i x_j \quad x_j x_k \quad x_k x_i$

$X = xx^T$

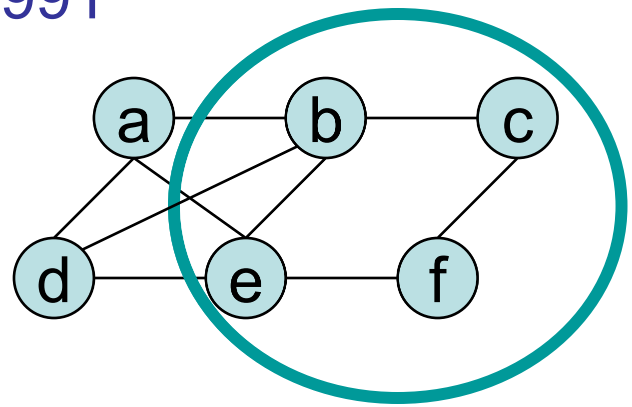
$X_{ij} \quad X_{jk} \quad X_{ki}$

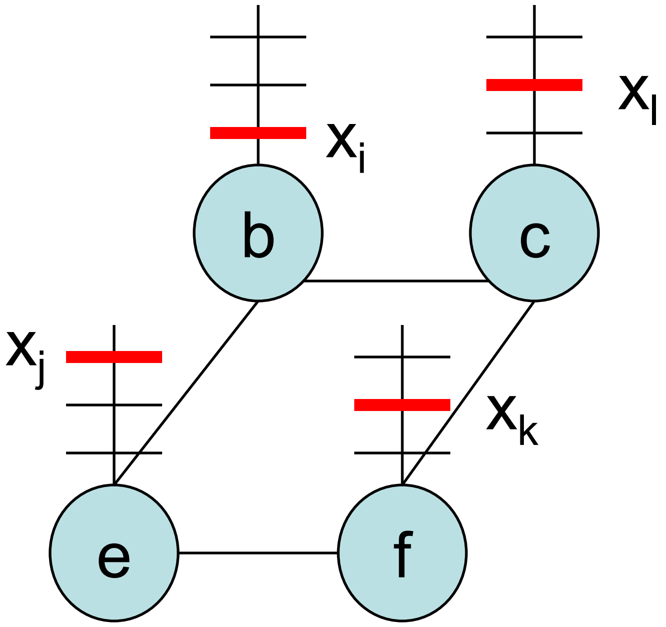
At least one of them is 1

$$X_{ij} + X_{jk} + X_{ki} \geq -1$$

Cycle Inequalities

Chopra and Rao, 1991





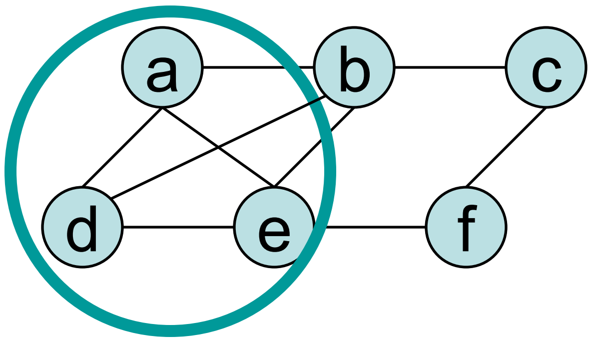
$$X_{ij} + X_{jk} + X_{kl} - X_{li} \geq -2$$

Generalizes to all cycles

LP-C

Second-Order Cone Constraints

Kumar et al., 2007



$$\mathbf{x}_c = \begin{pmatrix} x_i \\ x_j \\ x_k \end{pmatrix} \qquad \mathbf{X}_c = \begin{pmatrix} 1 & X_{ij} & X_{ik} \\ X_{ij} & 1 & X_{jk} \\ X_{ik} & X_{jk} & 1 \end{pmatrix}$$

$$X_c = x_c x_c^T$$

$$X_c \succeq x_c x_c^T$$

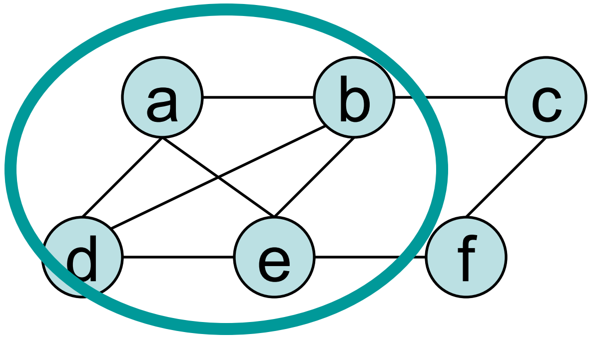
$$\mathbf{1} \bullet (\mathbf{X}_c - \mathbf{x}_c \mathbf{x}_c^T) \geq 0$$

SOCP-C

$$(x_i + x_j + x_k)^2 \leq 3 + X_{ij} + X_{jk} + X_{ki}$$

Second-Order Cone Constraints

Kumar et al., 2007



$$\mathbf{x}_c = \begin{bmatrix} x_i \\ x_j \\ x_k \\ x_l \end{bmatrix} \qquad \mathbf{X}_c = \begin{pmatrix} 1 & X_{ij} & X_{ik} & X_{il} \\ X_{ij} & 1 & X_{jk} & X_{jl} \\ X_{ik} & X_{jk} & 1 & X_{kl} \\ X_{il} & X_{jl} & X_{kl} & 1 \end{pmatrix}$$

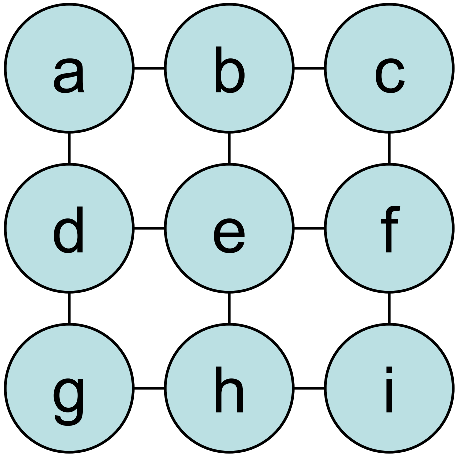
$$\mathbf{1} \bullet (\mathbf{X}_c - \mathbf{x}_c \mathbf{x}_c^T) \geq 0$$

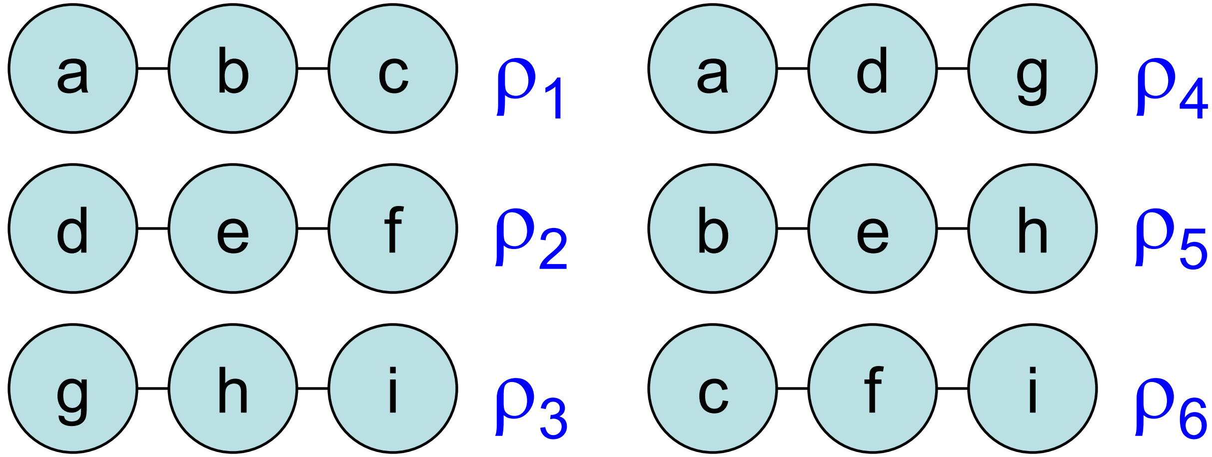
SOCP-Q

Outline

- Integer Programming Formulation
- Linear Programming Relaxation
- Additional Constraints
- Solving the Convex Relaxations
- Results and Conclusions

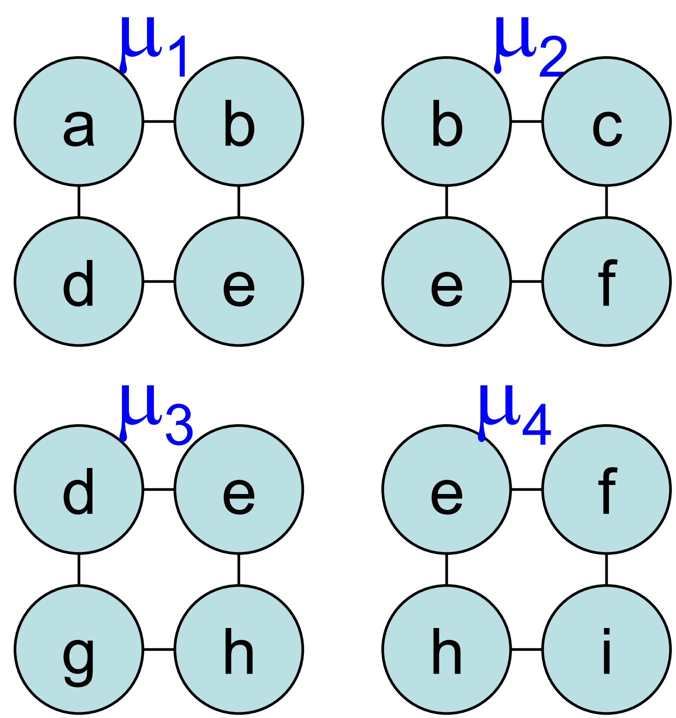
Modifying the Dual



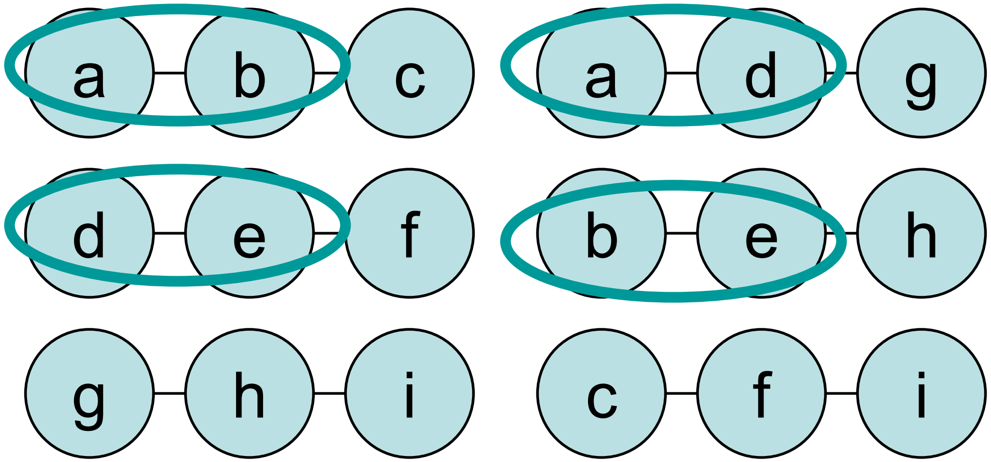


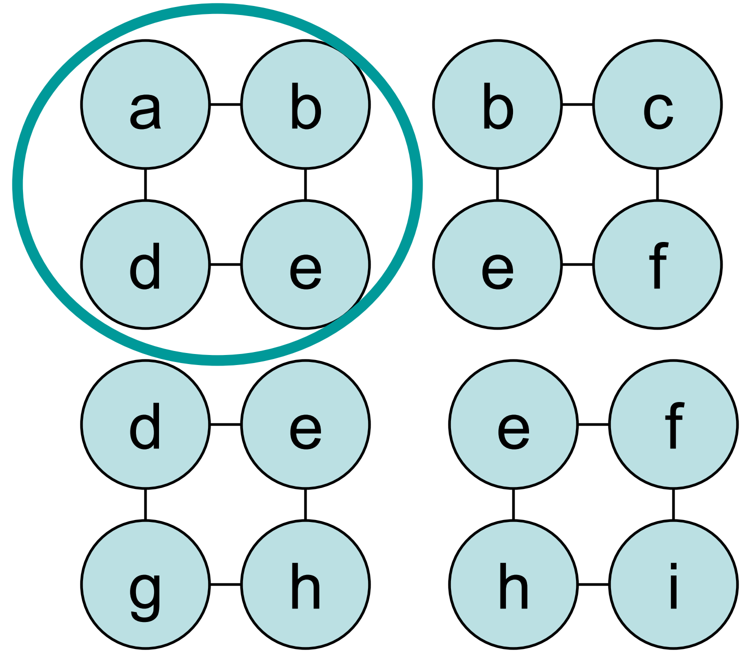
$$\max \sum \rho_i Q(\theta_i) + \sum \mu_j s_j$$

$$\sum \rho_i \theta_i \equiv \theta + \sum \mu_j s_j$$



Modifying TRW-S





Pick a variable --- **a**

Pick a cycle/clique with a

$\max \sum \rho_i Q(\theta_i) + \mu_j s_j$

$\sum \rho_i \theta_i \equiv \theta + \mu_j s_j$

Can be solved efficiently

REPEAT

Run TRW-S for trees with a

Properties of the Algorithm

Algorithm satisfies the reparametrization constraint

Value of dual never decreases CONVERGENCE

Solution satisfies Weak Tree Agreement (WTA)

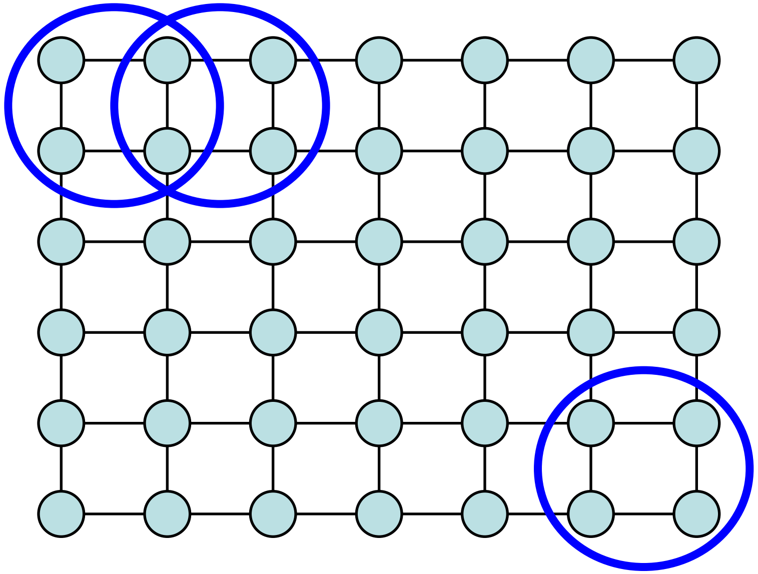
WTA not sufficient for convergence

More accurate results than TRW-S

Outline

- Integer Programming Formulation
- Linear Programming Relaxation
- Additional Constraints
- Solving the Convex Relaxations
- Results and Conclusions

4-Neighbourhood MRF



Test SOCP-C

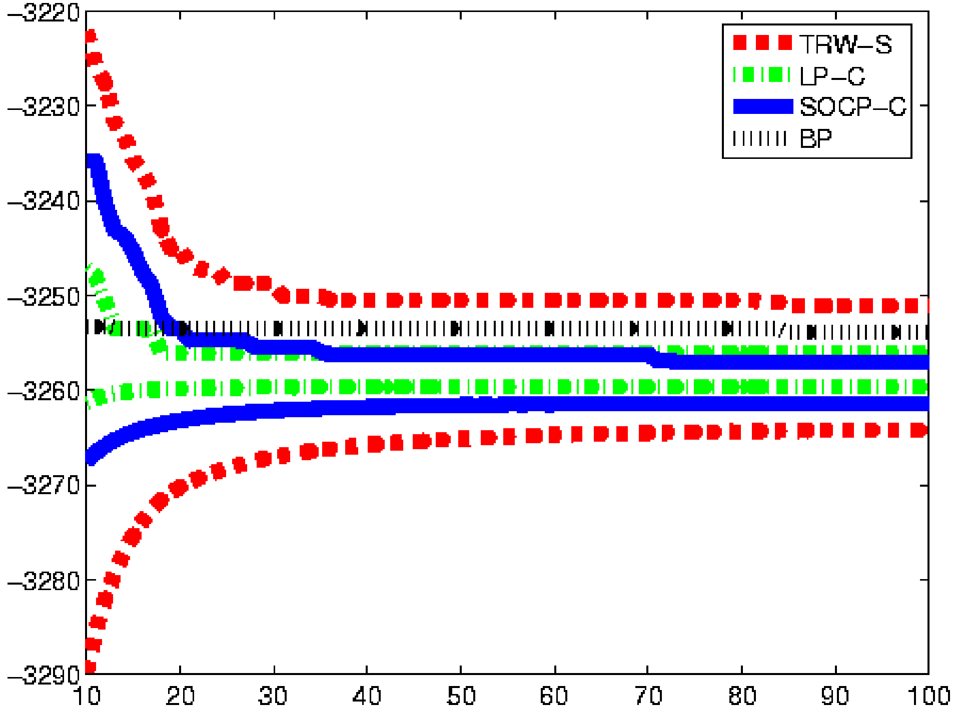
Test LP-C

50 binary MRFs of size 30x30

$$\mathbf{u} \approx \mathcal{N}(0,1)$$

$$\mathbf{P} \approx \mathcal{N}(0,\sigma^2)$$

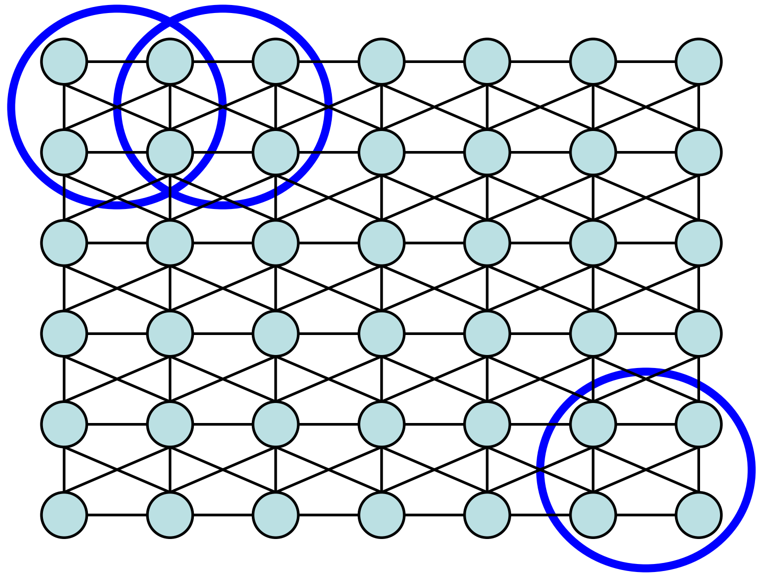
4-Neighbourhood MRF



σ = 5

LP-C dominates SOCP-C

8-Neighbourhood MRF



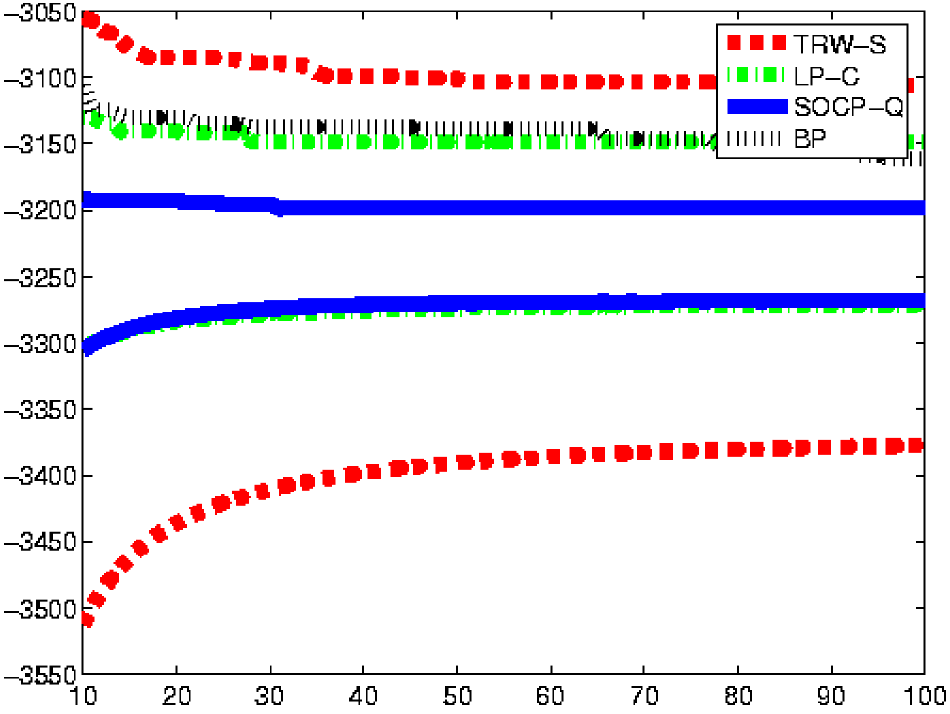
Test SOCP-Q

50 binary MRFs of size 30x30

$\mathbf{u} \approx \mathcal{N}(0,1)$

$\mathbf{P} \approx \mathcal{N}(0,\sigma^2)$

8-Neighbourhood MRF



$\sigma = 5 / \sqrt{2}$

SOCP-Q dominates LP-C

Conclusions

- Modified LP dual to include more constraints
- Extended TRW-S to solve tighter dual
- Experiments show improvement
- More results in the poster

Future Work

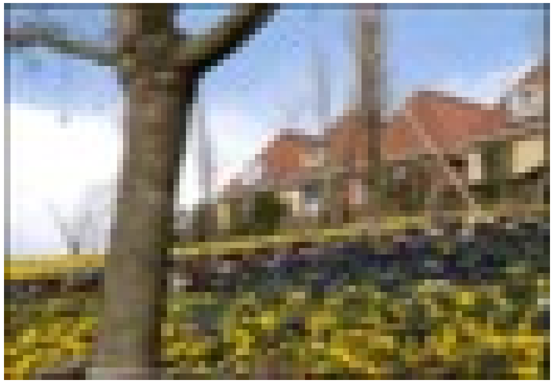
- More efficient subroutines for solving cycles/cliques
- Using more accurate LP solvers - proximal projections
- Analysis of SOCP-C vs. LP-C

Questions?

Timings

Linear in the number of variables!!

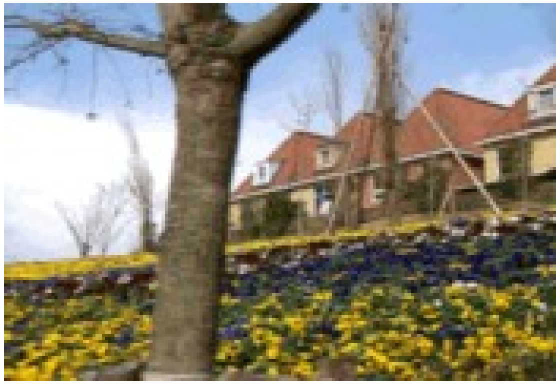
Video Segmentation



Keyframe

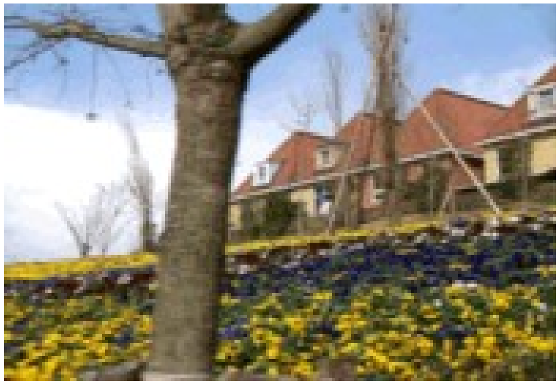
User Segmentation

Segment remaining video ….

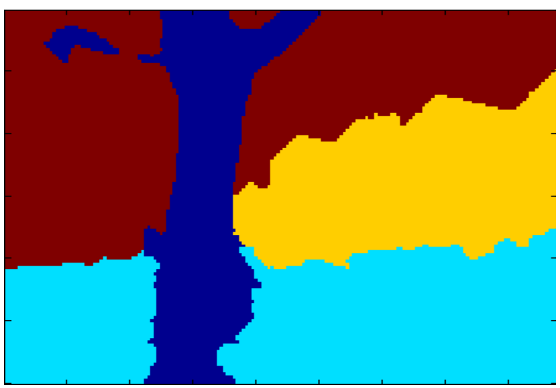


Video Segmentation

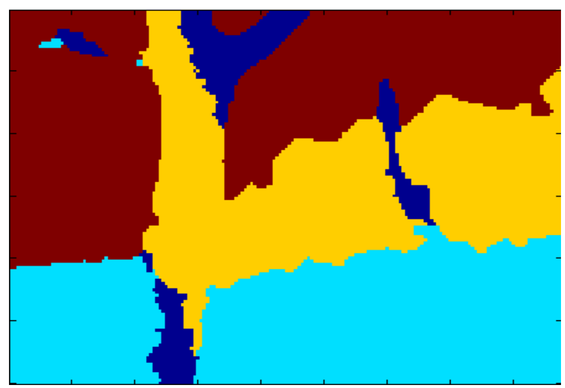
Input



Belief Propagation



8175



25620

18314

Video Segmentation

Input

αβ-swap

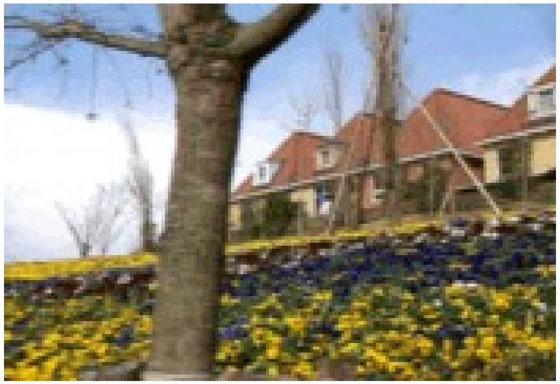
1187

1368

1289

Video Segmentation

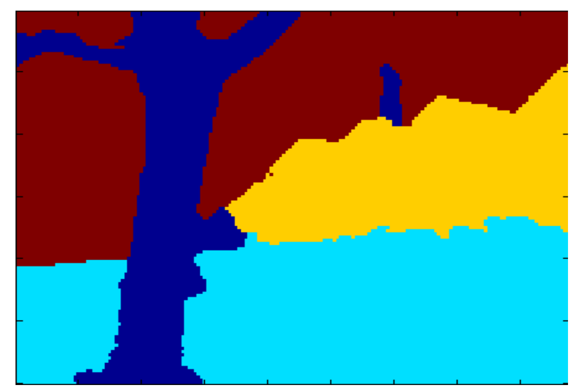
Input



α-expansion

2453

1266

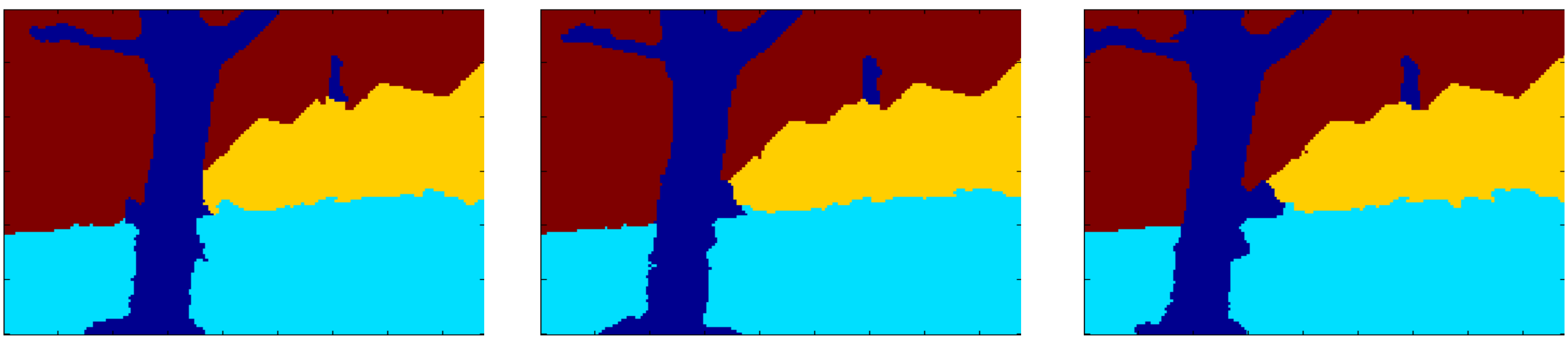


1225

Video Segmentation

Input

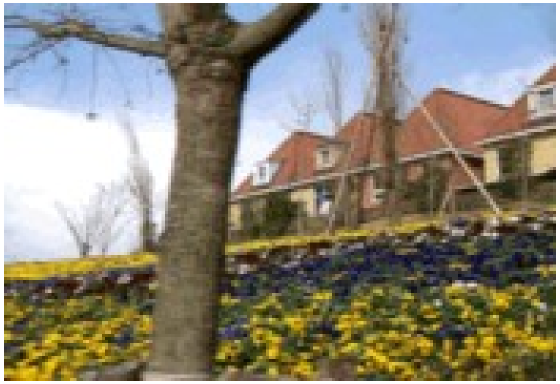
TRW-S



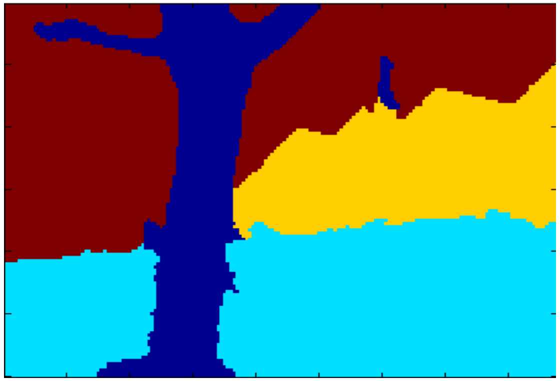
6425 1309 297

Video Segmentation

Input

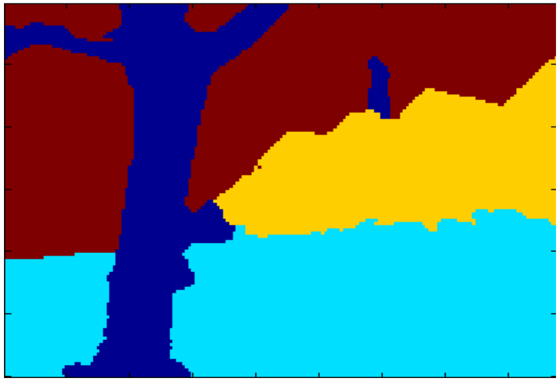


LP-C



719

264



294

Video Segmentation

Input

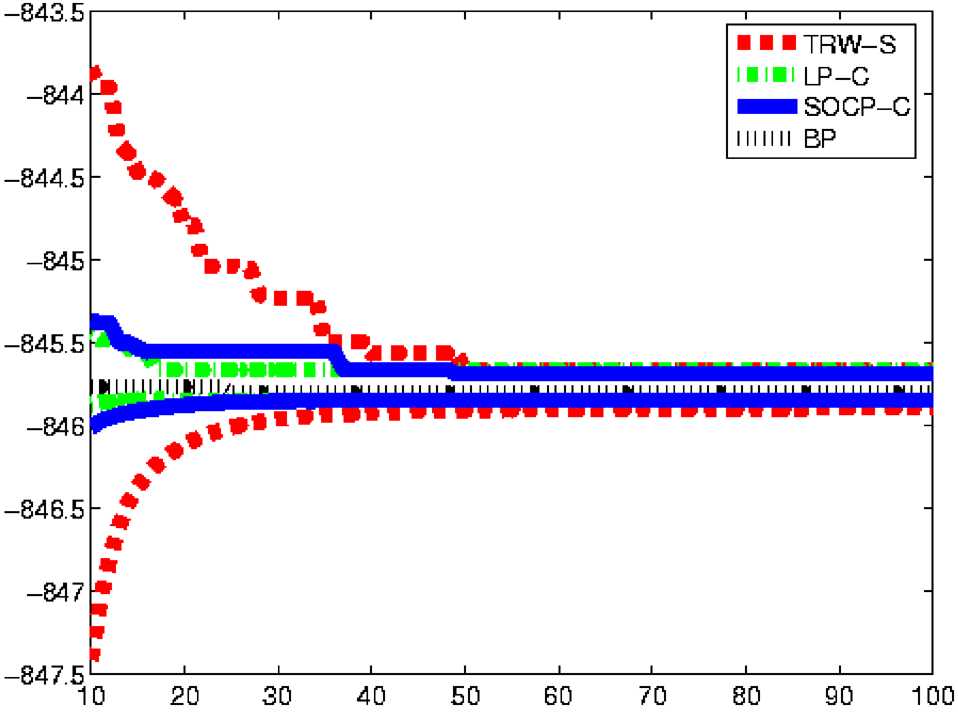
SOCP-Q

0

0

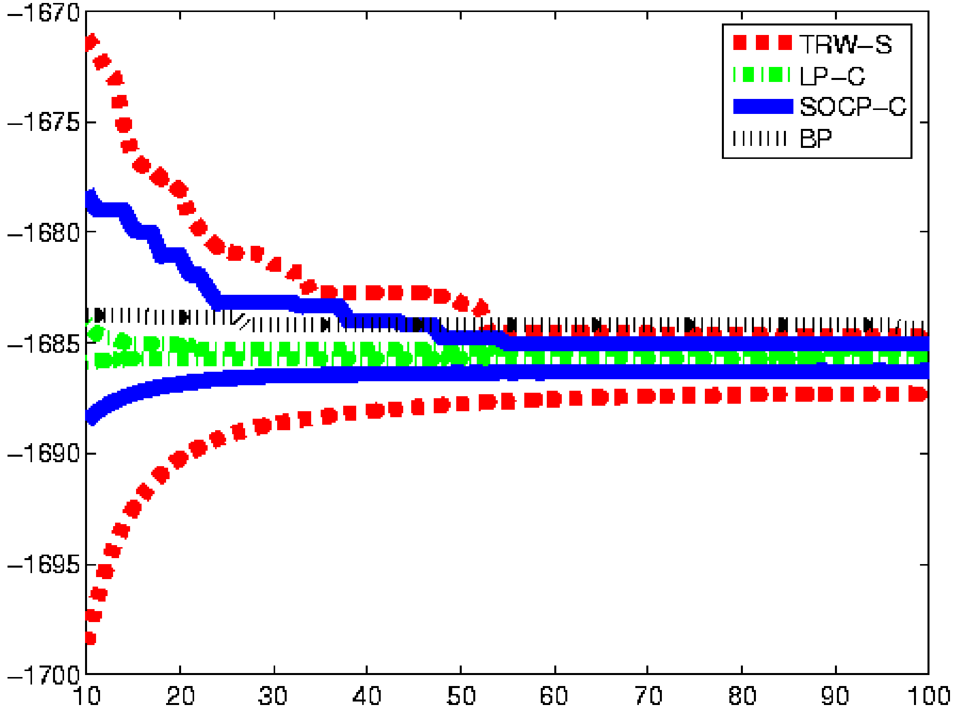
0

4-Neighbourhood MRF



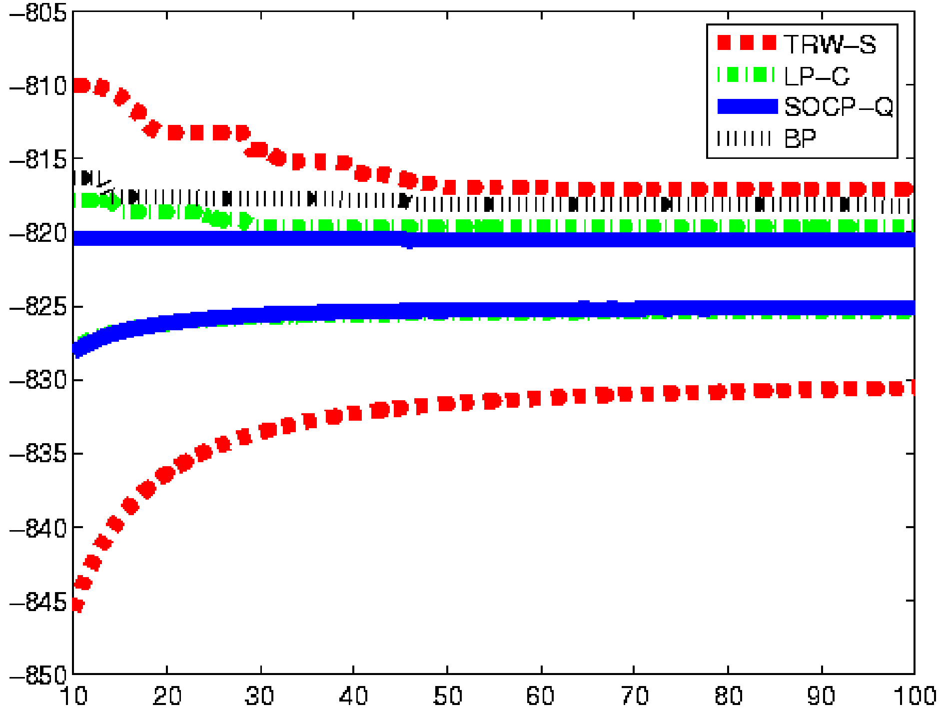
σ = 1

4-Neighbourhood MRF



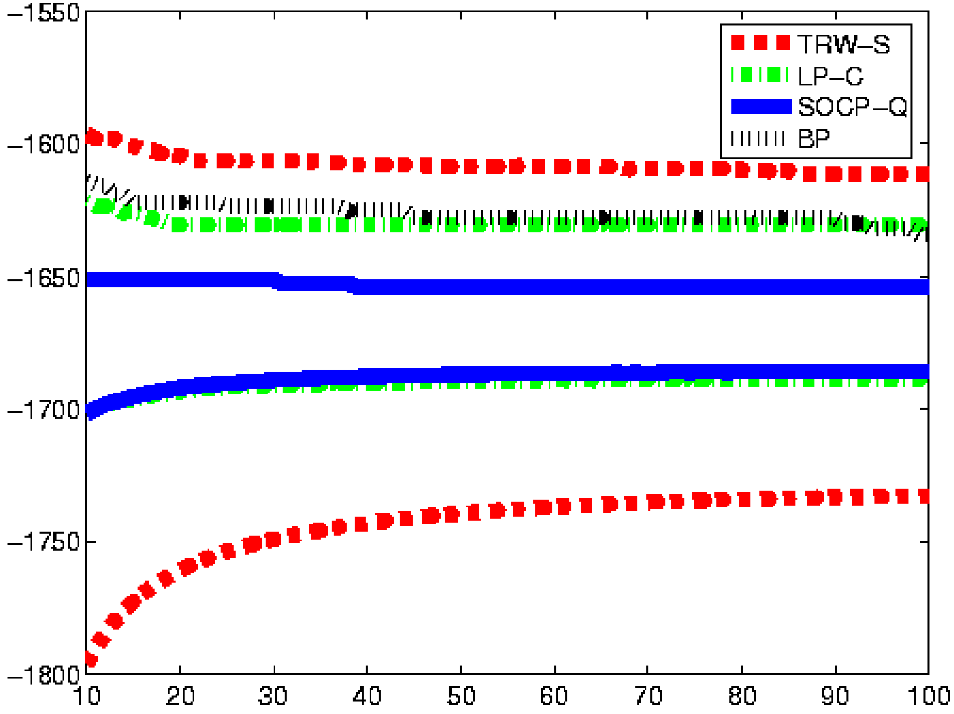
σ = 2.5

8-Neighbourhood MRF



σ = 1/√2

8-Neighbourhood MRF



$\sigma = 2.5 / \sqrt{2}$