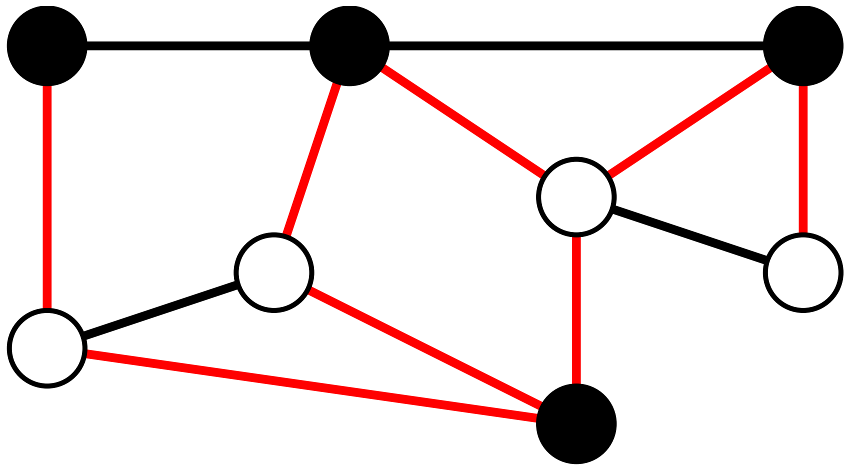
Image Segmentation Using Minimum st Cut

Tomas Werner

Center for Machine Perception
Czech Technical University
Prague

- Undirected graph (V, E) with nodes $v \in V$ and edges $vv' \in E \subseteq \binom{V}{2}$
- Every edge $vv' \in E$ has a non-negative weight $w_{vv'} \geq 0$
- Cut (S, T) is a partition of V into S and T such that $V = S \cup T$ and $S \cap T = \emptyset$
- Weight of cut (S, T) is $$W(S,T) = \sum_{v \in S,\ v' \in T} w_{vv'}$$

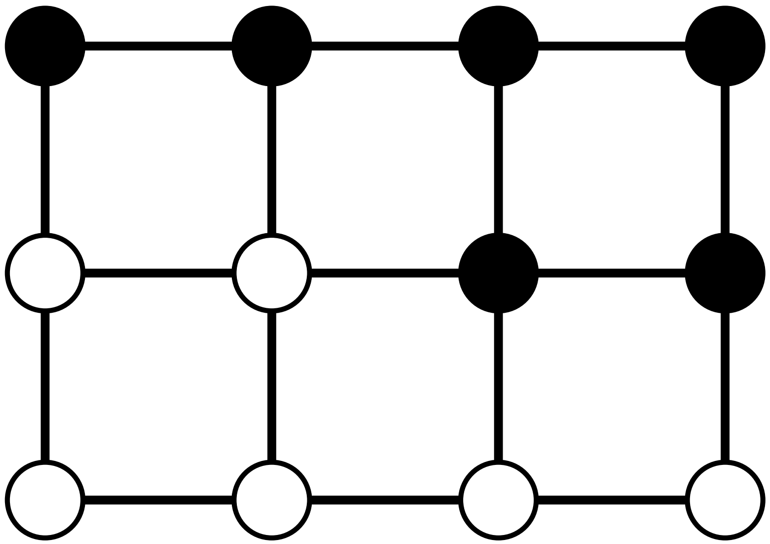


- Given two special nodes s and t, any cut (S, T) such that $s \in S, t \in T$ is an st cut
- Minimum st cut problem: Find st cut (S, T) that minimizes $W(S, T)$
  - There are fast algorithms for computing minimum st cut in large sparse graphs!
  - (They solve the related task, maximum flow.)

Image segmentation: Label each pixel either as background or as foreground

Formalize this task as follows:

- Model the image as grid graph (V, E)
  - Pixels are nodes $v \in V$
  - Pairs of neighboring pixels are edges $vv' \in E$
- x_v = label of pixel v ; $x_v \in \{\mathsf{F}, \mathsf{B}\}$ (F is foreground, B is background)



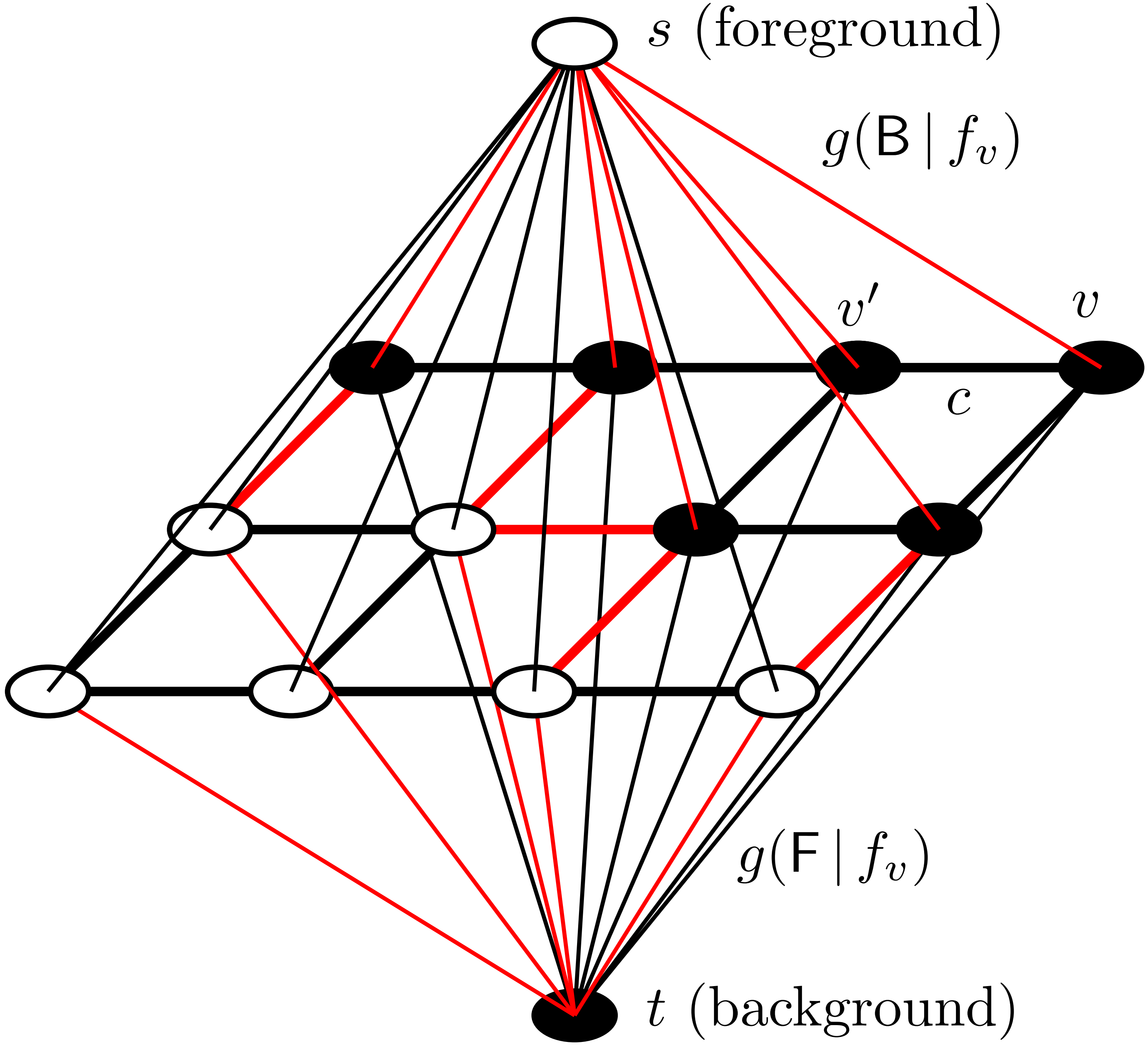
- f_v = intensity/color of pixel v ; all intensities form vector $\mathbf{f} = (f_v \mid v \in V)$
- Segmentation: Compute the 'best' labeling $\mathbf{x} = (x_v \mid v \in V)$ from intensities $\mathbf{f}$

What is the 'Best' Labeling?

To be a good segmentation, a labeling $\mathbf{x}$ must sastisfy two requirements:

- **Agreement with the input image intensities:**
  - Defined for each pixel independently
  - $p(\mathsf{B} \mid f_v) =$ probability that pixel with intensity f_v belongs to background
  - $p(\mathsf{F} \mid f_v) =$ probability that pixel with intensity f_v belongs to foreground
- **Contiguity of background and foreground:**
  - It is more likely that two neighboring pixels belong both to background or both to foreground that one to background and one to foreground
  - Probability defined for each pixel pair independently
  - $p(x_v, x_{v'}) = \begin{cases} a & \text{if } x_v = x_{v'} \\ b & \text{if } x_v \neq x_{v'} \end{cases}$ where $a > b$

- Best labeling $\mathbf{x}$ must maximize $$\prod_{v \in V} p(x_v \mid f_v) \prod_{vv' \in E} p(x_v, x_{v'})$$
- Taking negative logarithm: **Minimize** $$F(\mathbf{x} \mid \mathbf{f}) = \sum_{v \in V} g(x_v \mid f_v) + \sum_{vv' \in E} g(x_v, x_{v'})$$
- $F(\mathbf{x} \mid \mathbf{f}) =$ 'image energy'

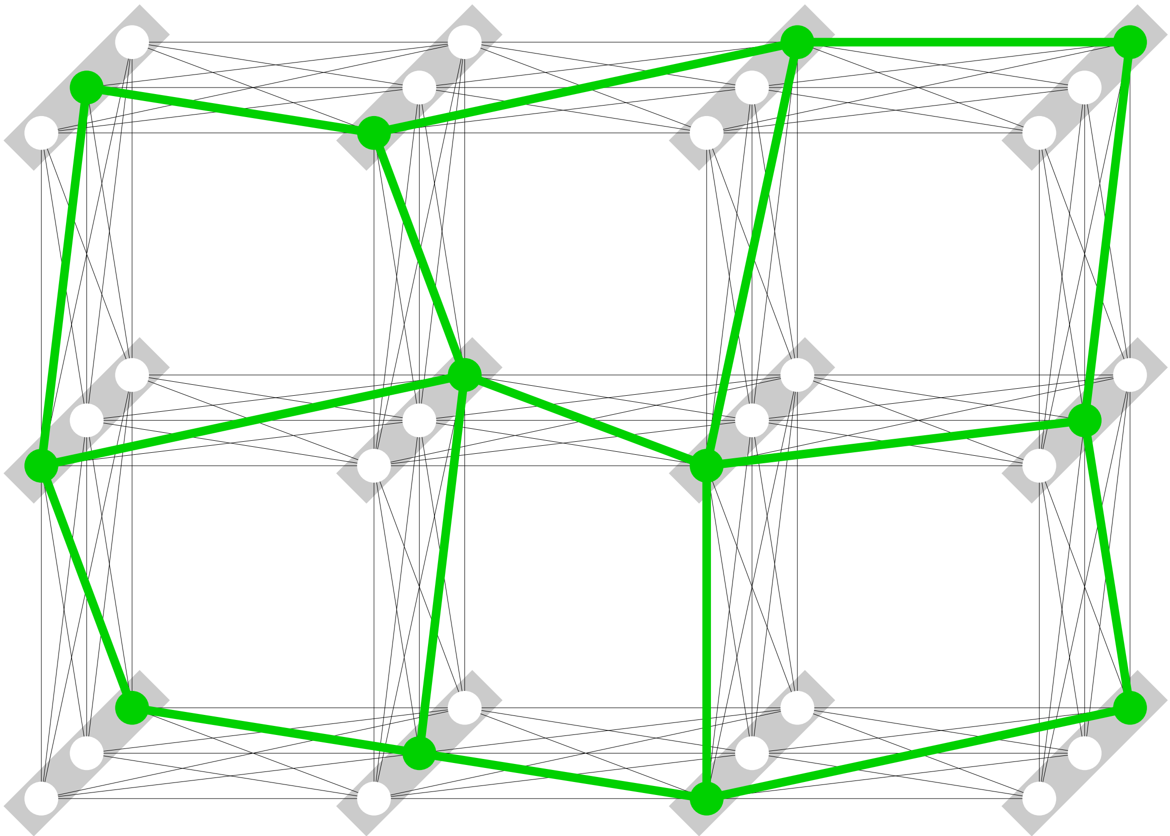

s (foreground)
$g(\mathsf{B} \mid f_v)$
v'
v
c
$g(\mathsf{F} \mid f_v)$
t (background)

Max-sum Labeling Problem

Tomas Werner

Center for Machine Perception
Czech Technical University
Prague

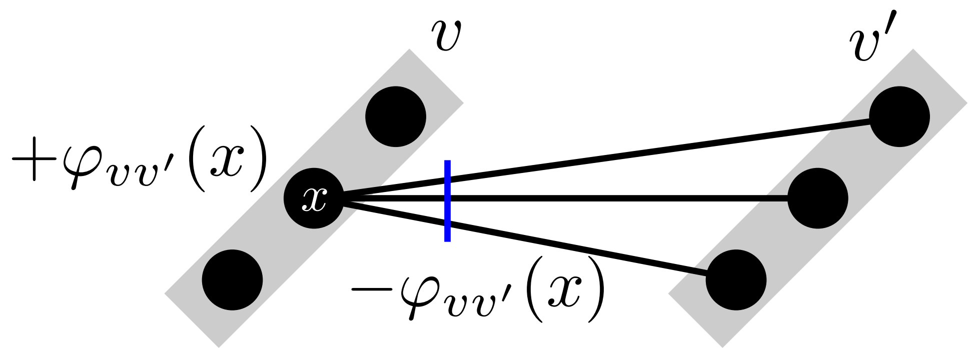
- Pixel v has label $x_v \in X$
- node qualities $g_v(x) \in [-\infty, \infty)$, edge qualities $g_{vv'}(x, x') \in [-\infty, \infty)$



Quality function: $$F(\mathbf{x} \,|\, \mathbf{g}) = \sum_{v \in V} g_v(x_v) + \sum_{\{v,v'\} \in E} g_{vv'}(x_v, x_{v'})$$

Find a labelling with maximal quality: $$\max_{\mathbf{x} \in X^V} F(\mathbf{x} \,|\, \mathbf{g})$$

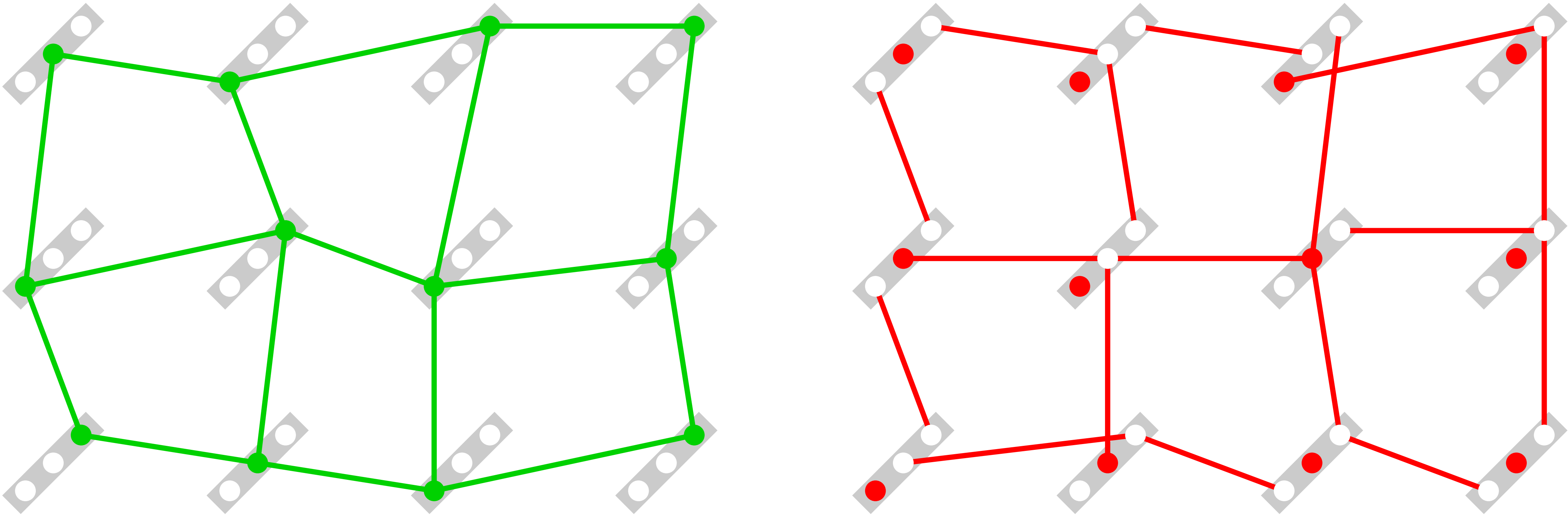
- Max-sum problems $\mathbf{g}$ and $\mathbf{g}'$ are equivalent iff they have the same quality for all labellings.
- Equivalent problems are denoted by $\mathbf{g} \sim \mathbf{g}'$
- Elementary equivalent transformation:



Upper bound

weight of configuration $\mathbf{x}$ — upper bound

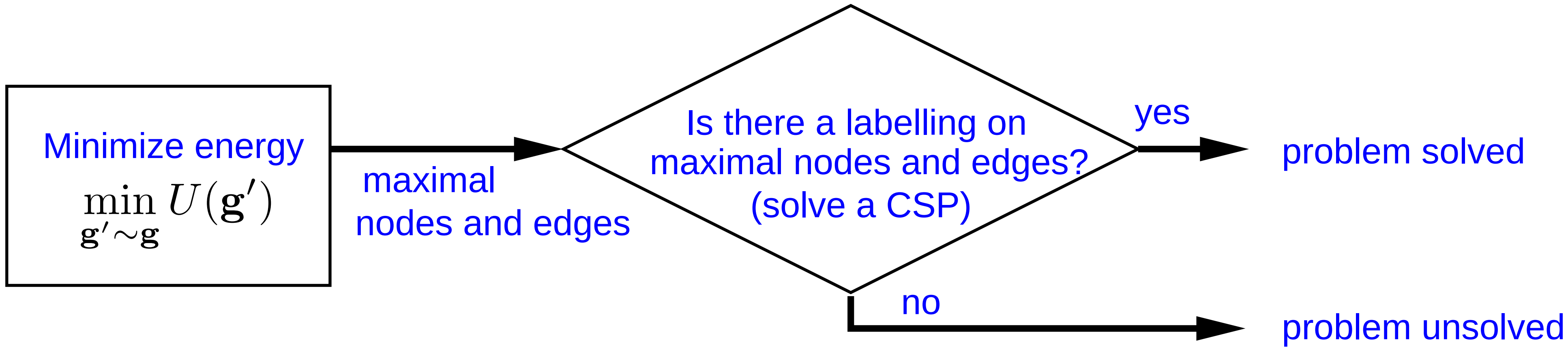
$$F(\mathbf{x}\,|\,\mathbf{g}) = \sum_{v\in V} g_c(x_v) + \sum_{\{v,v'\}\in E} g_{vv'}(x_v, x_{v'}) \quad \leq \quad U(\mathbf{g}) = \sum_{v\in V} \max_{x\in X} g_v(x) + \sum_{\{v,v'\}\in E} \max_{x,x'\in X} g_{vv'}(x, x')$$

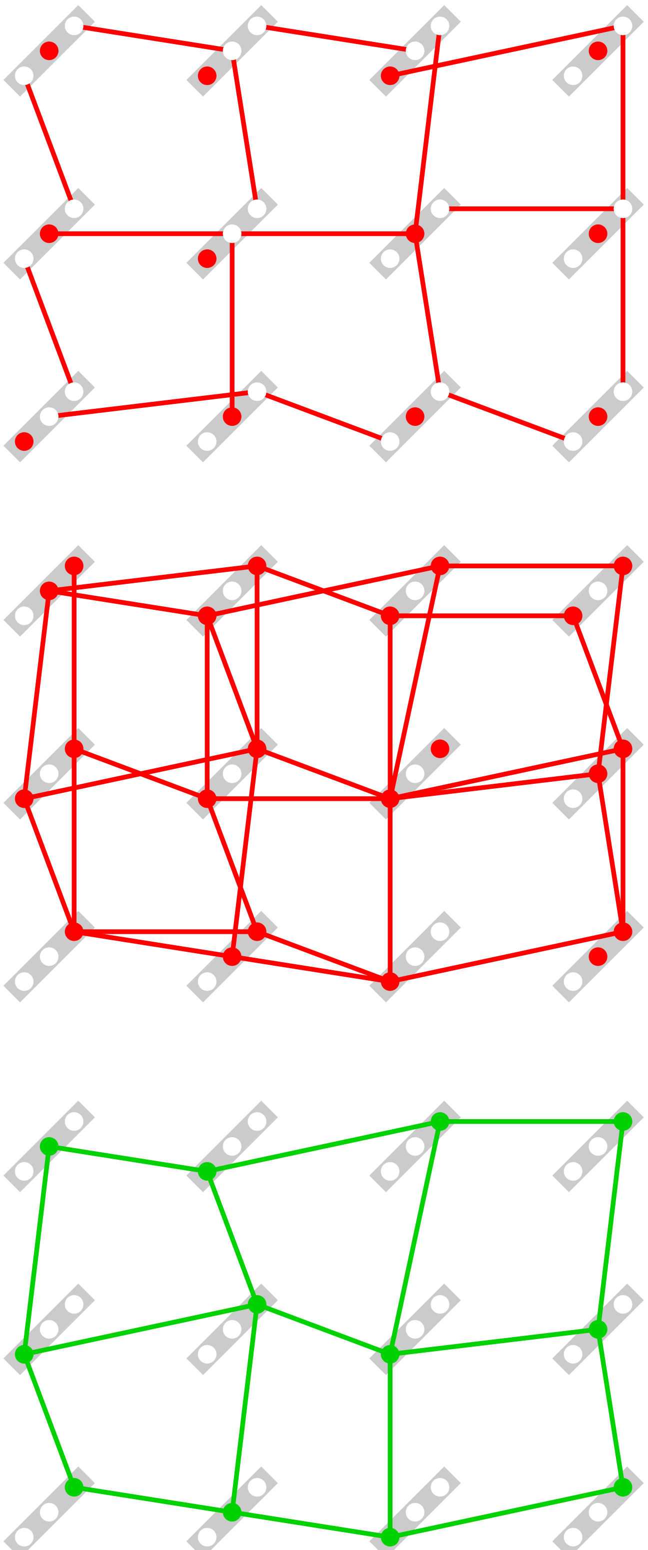


Upper bound on quality:

- $F(\mathbf{x}\,|\,\mathbf{g}) \leq U(\mathbf{g})$ for any $\mathbf{g}$ and $\mathbf{x}$
- $F(\mathbf{x}\,|\,\mathbf{g}) = U(\mathbf{g})$ if and only if $\mathbf{x}$ is composed of maximal nodes and edges

Dividing max-sum problem into two steps

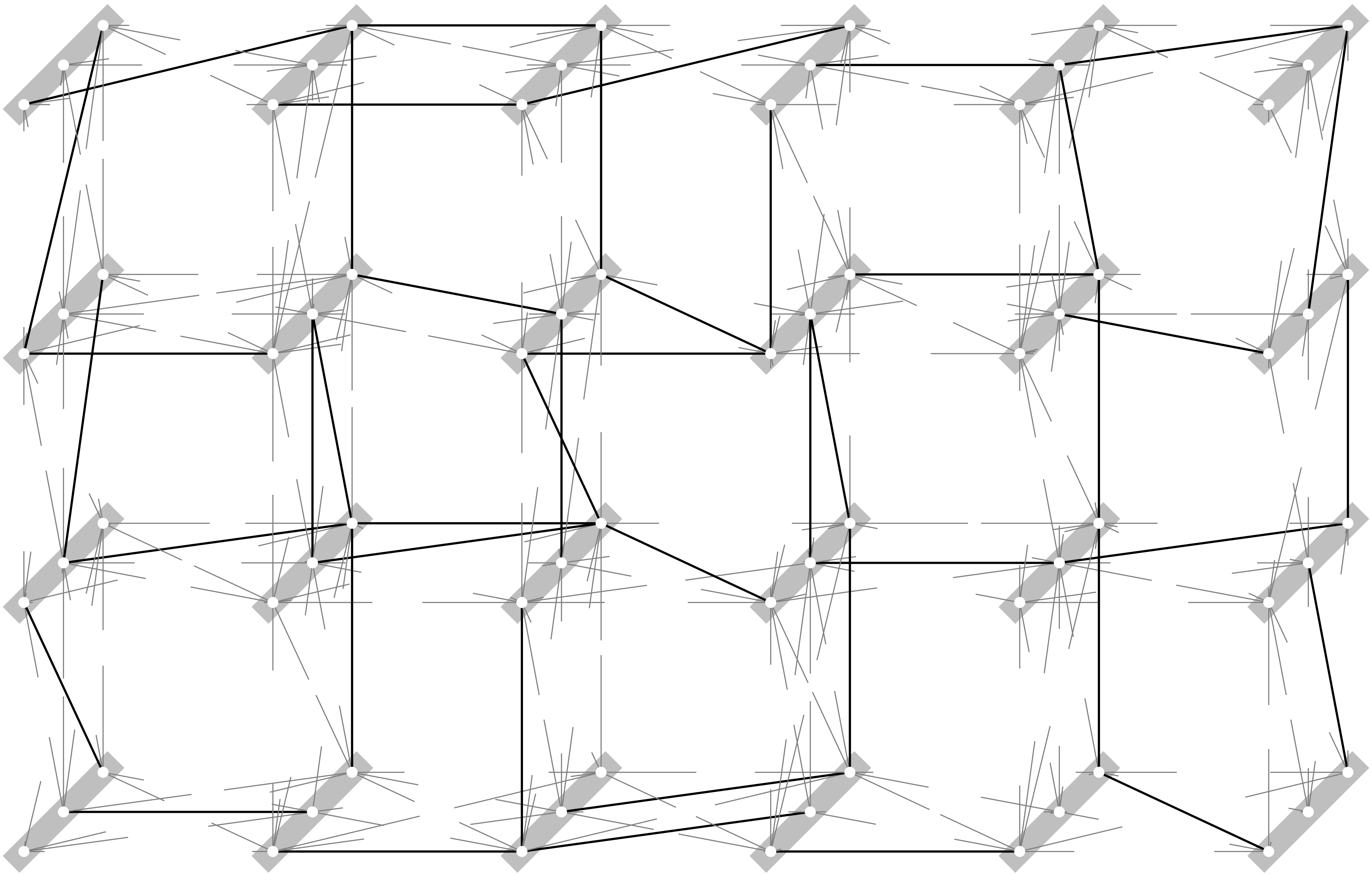




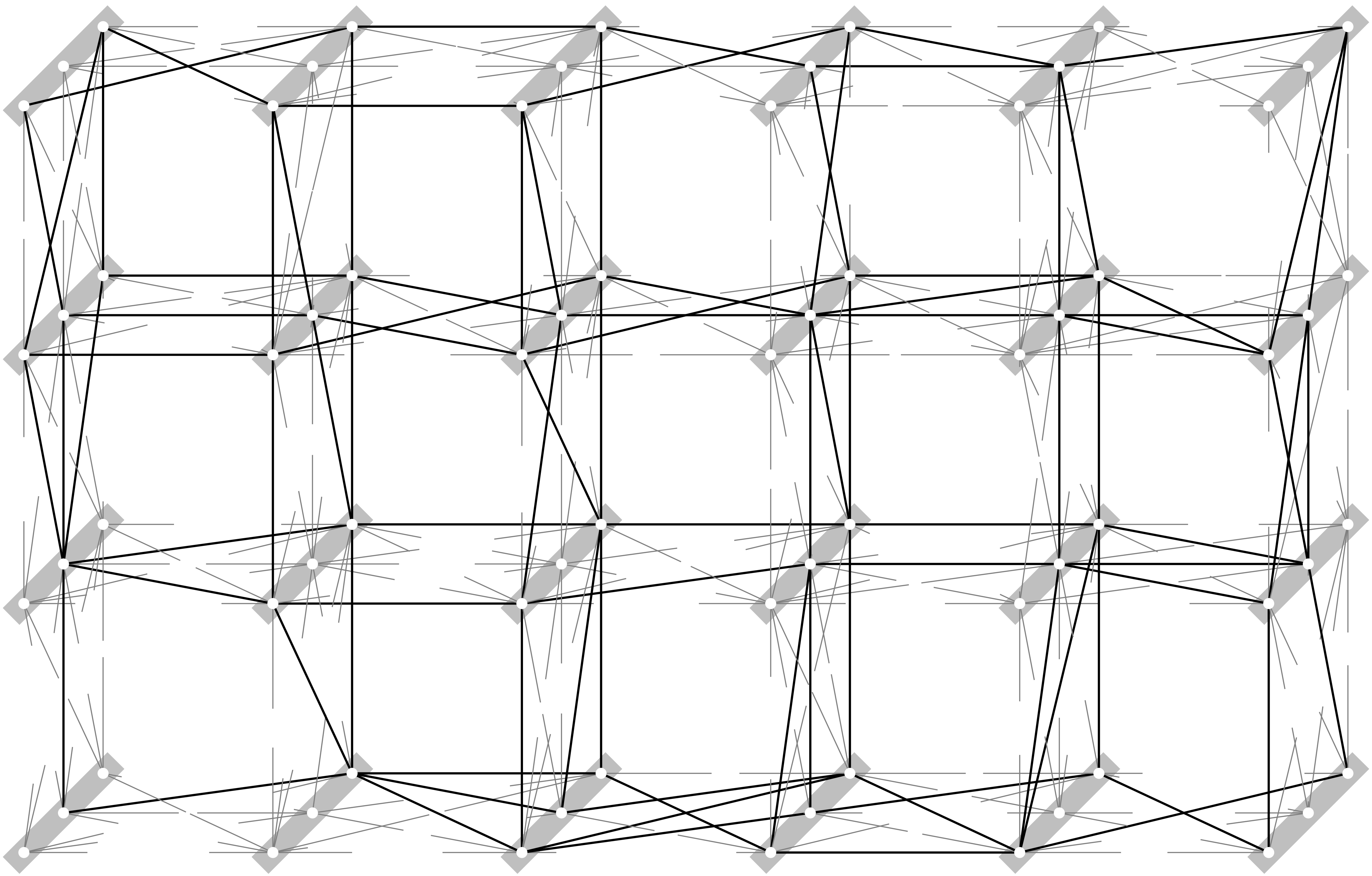
1. Minimize $U(\mathbf{g})$ by equivalent transformations (LP)

2. Try to find a configuration $\mathbf{x}$ composed of maximal nodes and edges (CSP, CLP):
   - if such a configuration exists, we have an exact solution
   - if not, we have only a strict upper bound

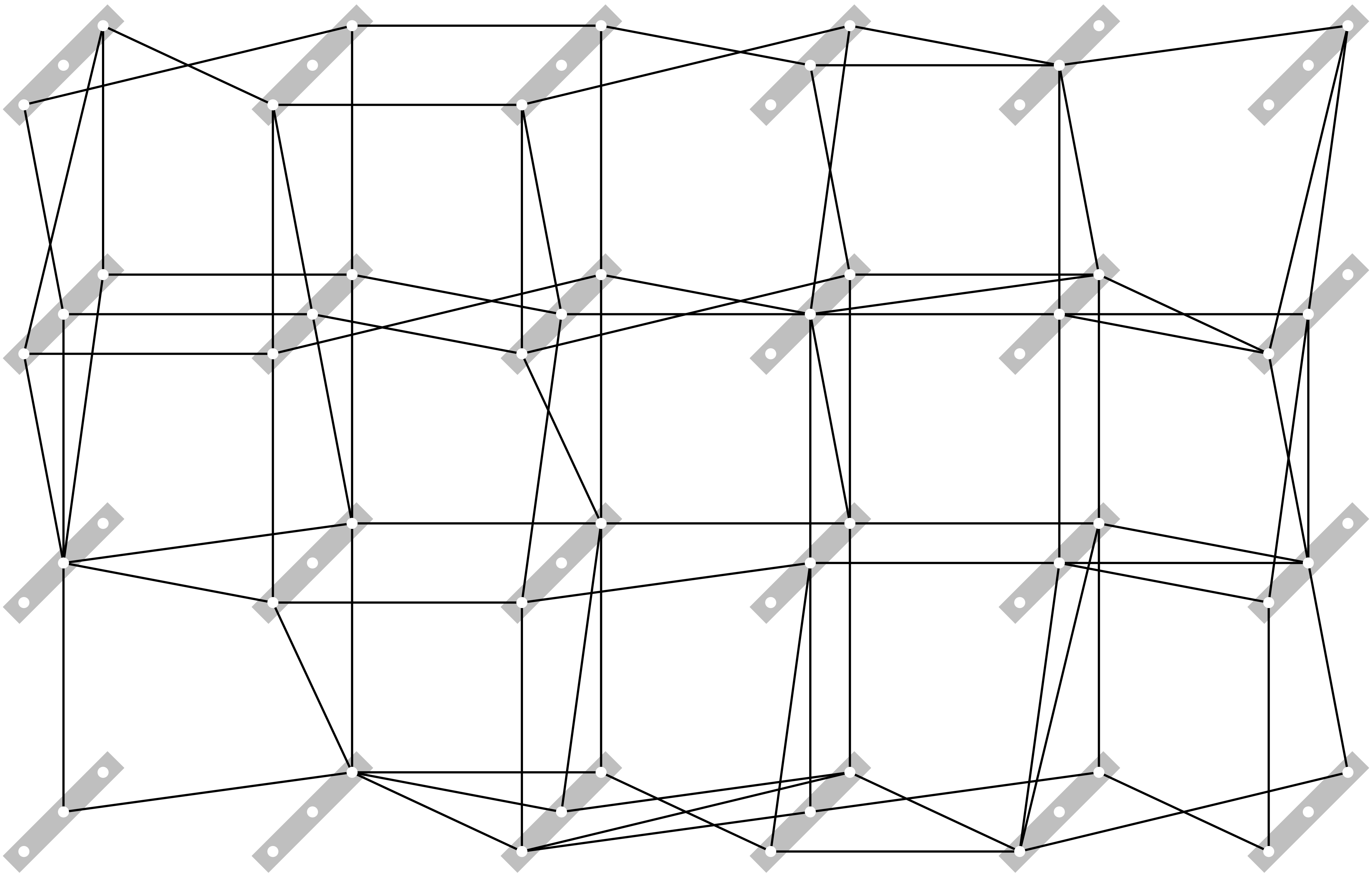
Original values of $g_v(x), g_{vv'}(x, x')$:



Optimal values of $g_v(x), g_{vv'}(x, x')$:



Only maximal nodes and edges shown. Any optimal labelling has to pass through them.

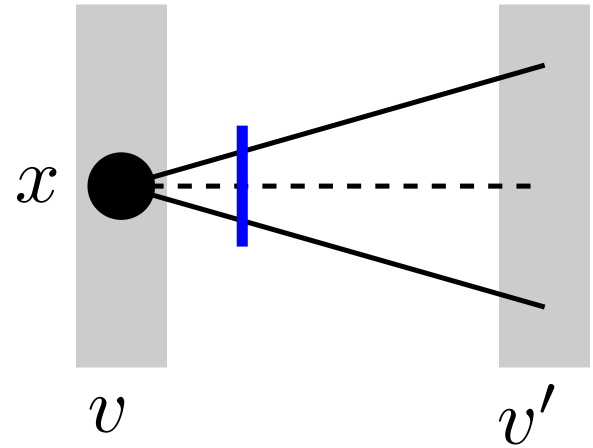


Max-sum diffusion

Max-sum diffusion [Koval-Kovalevsky-70's] finds an arc consistent equivalent problem with a lower height.

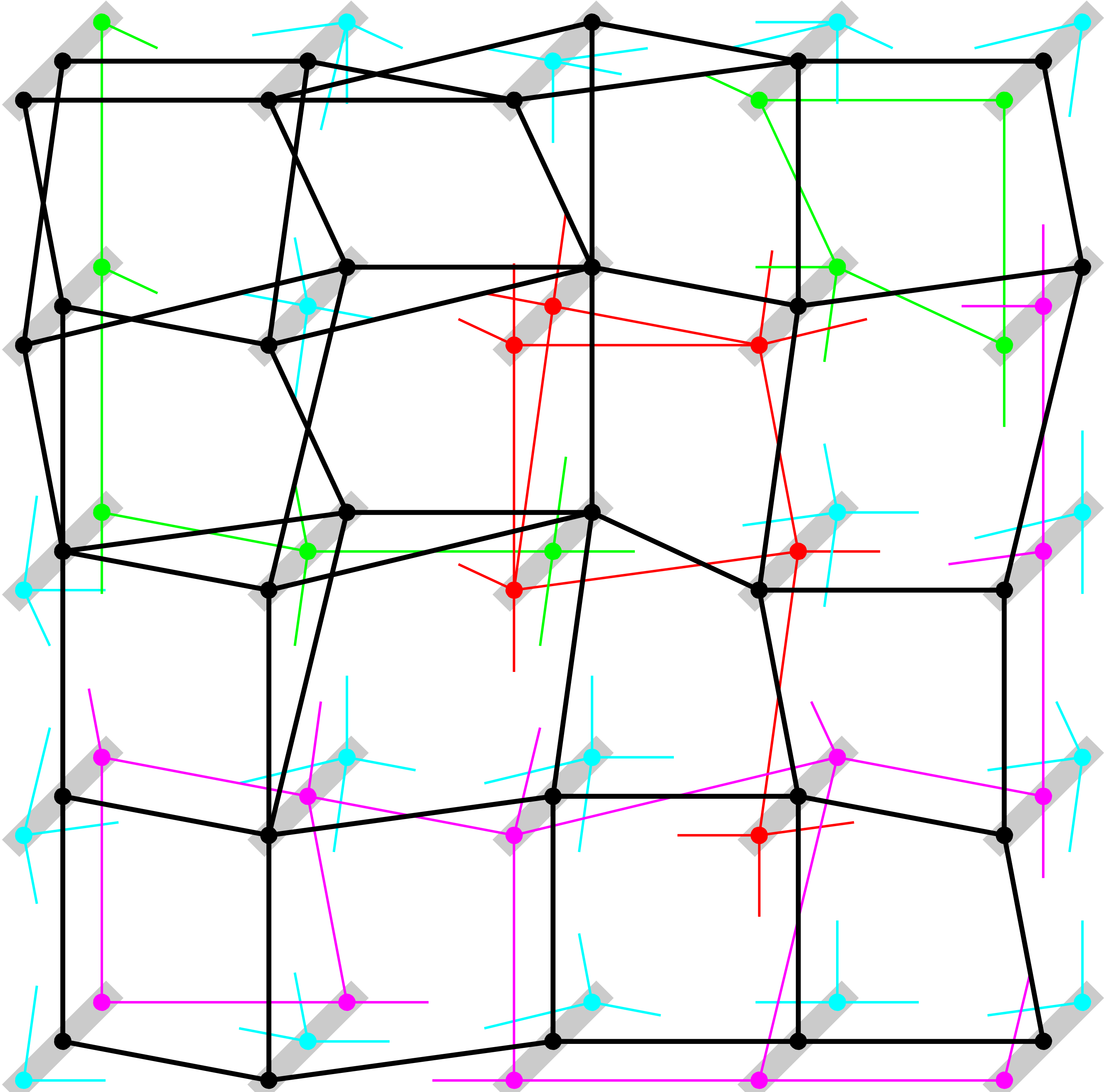
- **Pencil equalization** on pencil (v, v', x) is the equivalent transformation that sets

$$g_v(x) = \max_{x'} g_{vv'}(x, x')$$



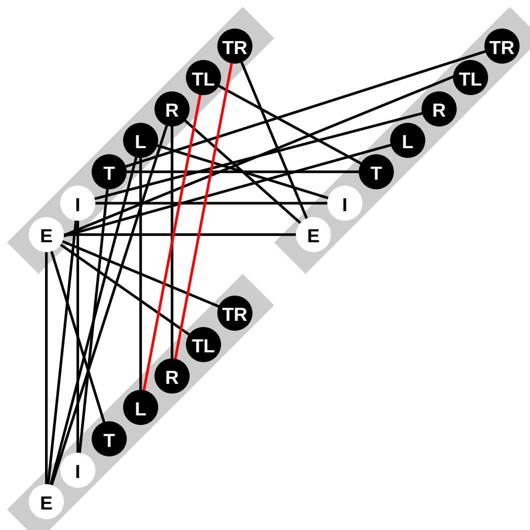
- Pencil equalization (on all labels of one pixel) decreases $U(\mathbf{g})$
- Algorithm: Repeat pencil equalization on all pencils until convergence.
- Conjecture:
  - Diffusion always converges (in both value $U(\mathbf{g})$ and argument $\mathbf{g}$!).
  - On convergence, all pencils (v, v', x) satisfy $g_v(x) = \max_{x'} g_{vv'}(x, x')$.

$g_v(x) = g_{vv'}(x, x') < g_{v'}(x') \iff$ line from node (v, x) aiming but not reaching (v', x')
$g_v(x) = g_{vv'}(x, x') = g_{v'}(x') \iff$ line joining nodes (v, x) and (v', x')

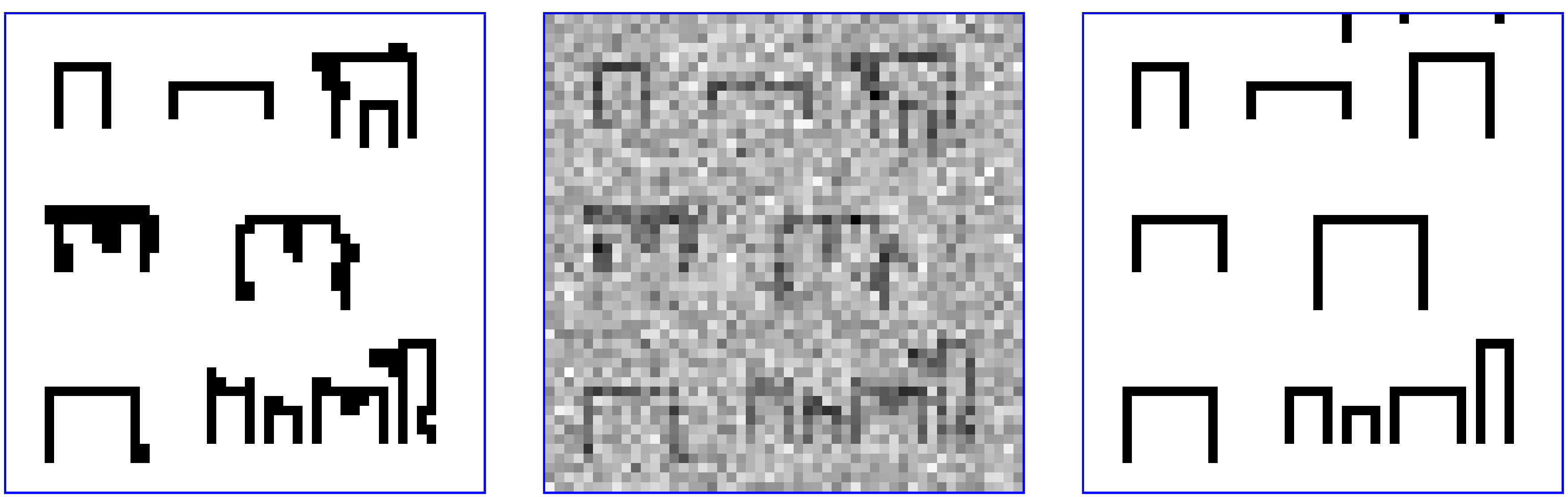


$$F(\mathbf{x}\,|\,\mathbf{g}) = \underbrace{\sum_{v\in V} g_v(x_v)}_{\text{data term}} + \underbrace{\sum_{\{v,v'\}\in E} g_{vv'}(x_v, x'_v)}_{\text{prior term}}$$

- Data term: accordance with the input signal
- Prior term: assigns quality (log-likelihood) to each labelling, assuming no signal

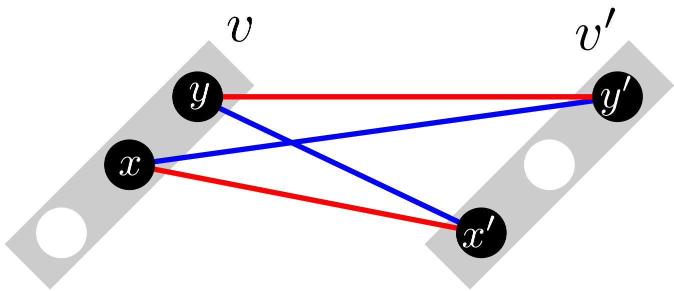


E	E	E	E	E	E
E	TL	T	T	TR	E
E	L	I	I	R	E
E	L	I	I	R	E
E	L	I	I	R	E
E	E	E	E	E	E



Given a total order $\leq$ on X.
Function $g_{vv'}(\bullet, \bullet)$ is supermodular iff

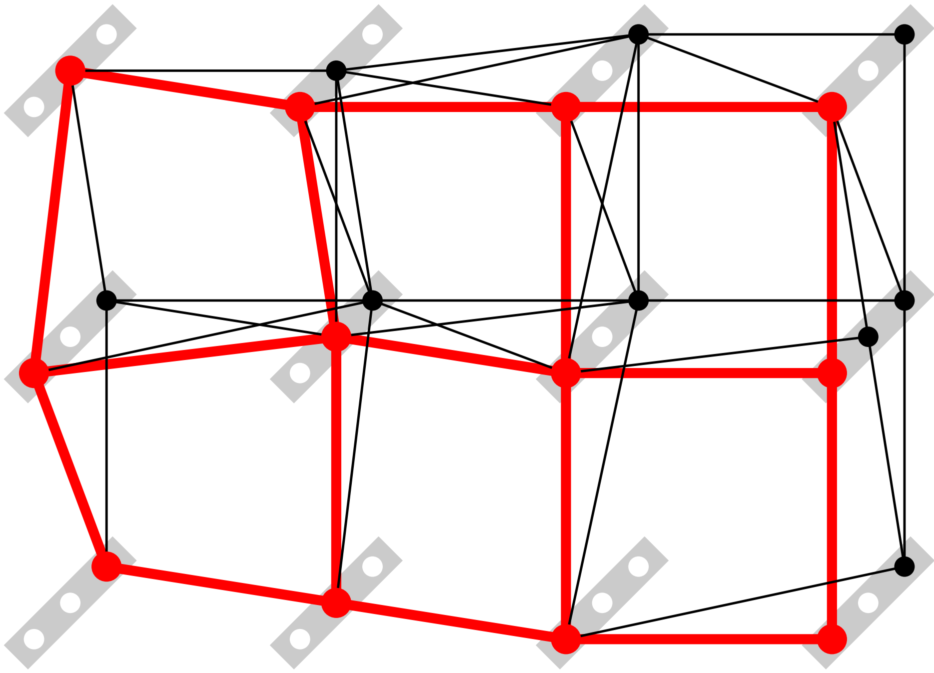
$$x \leq x', y \leq y' \implies$$
$$g_{vv'}(x, x') + g_{vv'}(y, y') \geq g_{vv'}(x, y') + g_{vv'}(y, x')$$



Theorem: Maximal nodes and edges are arc consistent $\iff$ they form a labelling.

Proof:

- Supermodularity is preserved under equivalent transformations
- If edges (x, y') and (y, x') are maximal, so are edges (x, x') and (y, y')
- On arc consistency, the lowest nodes form a labelling.



- Supermodular max-sum problems are reducible to max-flow/min-cut .