Markov Random Fields for Computer Vision (Part 2)

Machine Learning Summer School (MLSS 2011)

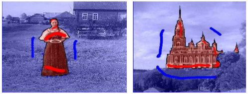
Stephen Gould
stephen.gould@anu.edu.au

Australian National University

13–17 June, 2011

Recap: Pixel Labeling

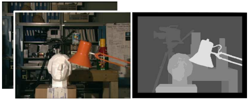
Many problems in computer vision can be formulated as inference in a Markov random field.



Interactive segmentation

Surface context

Semantic labeling



Stereo matching

Photo montage

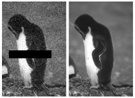
Denoising

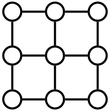
How do we minimize the resulting energy function?

Outline of Energy Minimization via Graph-cuts

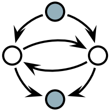
Big picture:

- Start with a pixel labeling problem
- Formulate as a (multilabel) graphical model inference problem
- Convert to a series of binary pairwise MRF inference problems
- Write MRF as a quadratic pseudo-Boolean function
- Convert pseudo-Boolean minimization to min-cut problem
- Equivalently, formulate as a max-flow problem
- Solve using augmented-path algorithm





$$\{0,1\}^n \to \mathbb{R}$$
$$\dots$$
$$\{0,1\}^n \to \mathbb{R}$$



A Note About Graphs

point of confusion:

graphs are used to represent many different things

In this talk we use graphs to...

- represent probabilistic models (or energy functions), e.g., Markov random fields and factor graphs.
- represent optimization problems, e.g., psuedo-Boolean function minimization.

Pseudo-boolean Functions [Boros and Hammer, 2001]

Pseudo-boolean Function

A mapping $f : \{0, 1\}^n \rightarrow \mathbb{R}$ is called a *pseudo-Boolean function*.

Pseudo-boolean Functions [Boros and Hammer, 2001]

Pseudo-boolean Function

A mapping $f : \{0,1\}^n \to \mathbb{R}$ is called a *pseudo-Boolean function*.

- Pseudo-boolean functions can be uniquely represented as *multi-linear polynomials*, e.g., $f(y_1, y_2) = 6 + y_1 + 5y_2 - 7y_1y_2$.

Pseudo-boolean Functions [Boros and Hammer, 2001]

Pseudo-boolean Function

A mapping $f : \{0, 1\}^n \rightarrow \mathbb{R}$ is called a *pseudo-Boolean function*.

- Pseudo-boolean functions can be uniquely represented as *multi-linear polynomials*, e.g., $f(y_1, y_2) = 6 + y_1 + 5y_2 - 7y_1 y_2$.
- Pseudo-boolean functions can also be represented in *posiform*, e.g., $f(y_1, y_2) = 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2 + 3\bar{y}_1 y_2 + 4y_1 \bar{y}_2$. This representation is not unique.

Pseudo-boolean Functions [Boros and Hammer, 2001]

Pseudo-boolean Function

A mapping $f : \{0,1\}^n \rightarrow \mathbb{R}$ is called a *pseudo-Boolean function*.

- Pseudo-boolean functions can be uniquely represented as *multi-linear polynomials*, e.g., $f(y_1, y_2) = 6 + y_1 + 5y_2 - 7y_1 y_2$.
- Pseudo-boolean functions can also be represented in *posiform*, e.g., $f(y_1, y_2) = 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2 + 3\bar{y}_1 y_2 + 4y_1 \bar{y}_2$. This representation is not unique.
- **A binary pairwise Markov random field (MRF) is just a quadratic pseudo-Boolean function.**

Representing a Binary Pairwise MRF

Consider a binary pairwise MRF over two variables:

	0	1
0	A	B
1	C	D

Representing a Binary Pairwise MRF

Consider a binary pairwise MRF over two variables:

	0	1
0	A	B
1	C	D

$$A + \begin{array}{|c|c|}\hline 0 & 0 \\ \hline C-A & C-A \\ \hline\end{array} + \begin{array}{|c|c|}\hline 0 & D-C \\ \hline 0 & D-C \\ \hline\end{array} + \begin{array}{|c|c|}\hline 0 & B+C-A-D \\ \hline 0 & 0 \\ \hline\end{array}$$

$$E(y_1, y_2) = A + (C - A)y_1 + (D - C)y_2 + (B + C - A - D)\bar{y}_1 y_2$$

[Kolmogorov and Zabih, 2004]

Pseudo-boolean Optimization [Boros and Hammer, 2001]

A large number of classical combinatorial optimization problems can be formulated in terms of pseudo-boolean optimization, e.g.,

- **Maximum independent set problem:** find the largest set of verticies in a graph such that no two are adjacent.

$$\alpha(G) = \max_{\mathbf{x}\in\{0,1\}^n}\left(\sum_{i\in\mathcal{V}} x_i - \sum_{(i,j)\in\mathcal{E}} x_i x_j\right)$$

- **Minimum vertex cover:** find the smallest set of verticies such that every edge in the graph is adjacent to at least one vertex in the set.

$$\tau(G) = \min_{\mathbf{x}\in\{0,1\}^n}\left(\sum_{i\in\mathcal{V}} x_i + \sum_{(i,j)\in\mathcal{E}} \bar{x}_i \bar{x}_j\right)$$

- **Maximum satisfiability problem:** find an assignment to a set of variables that satisfy as many clauses as possible.

$$\max_{\mathbf{x}\in\{0,1\}^n}\left(\sum_{C\in\mathcal{C}}\left(1 - \sum_{u\in C} \bar{u}\right)\right)$$

Pseudo-boolean Optimization [Boros and Hammer, 2001]

A large number of classical combinatorial optimization problems can be formulated in terms of pseudo-boolean optimization, e.g.,

- **Maximum independent set problem:** find the largest set of verticies in a graph such that no two are adjacent.

$$\alpha(G) = \max_{\mathbf{x} \in \{0,1\}^n} \left(\sum_{i \in \mathcal{V}} x_i - \sum_{(i,j) \in \mathcal{E}} x_i x_j \right)$$

- **Minimum vertex cover:** find the smallest set of verticies such that every edge in the graph is adjacent to at least one vertex in the set.

$$\tau(G) = \min_{\mathbf{x} \in \{0,1\}^n} \left(\sum_{i \in \mathcal{V}} x_i + \sum_{(i,j) \in \mathcal{E}} \bar{x}_i \bar{x}_j \right)$$

- **Maximum satisfiability problem:** find an assignment to a set of variables that satisfy as many clauses as possible.

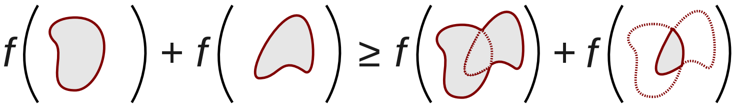
$$\max_{\mathbf{x} \in \{0,1\}^n} \left(\sum_{C \in \mathcal{C}} \left(1 - \sum_{u \in C} \bar{u} \right) \right)$$

These problems are all NP-hard.

Submodular Functions

Submodularity

Let $\mathcal{V}$ be a set. A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ is called *submodular* if $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$ for all subsets $X, Y \subseteq \mathcal{V}$.



Submodular Binary Pairwise MRFs

Submodularity

A pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$ is called *submodular* if $f(\mathbf{x}) + f(\mathbf{y}) \geq f(\mathbf{x} \vee \mathbf{y}) + f(\mathbf{x} \wedge \mathbf{y})$ for all vectors $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$.

Submodular Binary Pairwise MRFs

Submodularity

A pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$ is called *submodular* if $f(\mathbf{x}) + f(\mathbf{y}) \geq f(\mathbf{x} \vee \mathbf{y}) + f(\mathbf{x} \wedge \mathbf{y})$ for all vectors $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$.

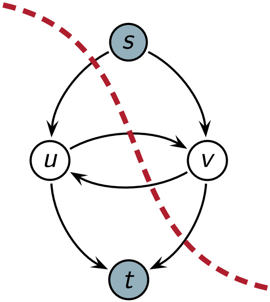
Submodularity checks for pairwise binary MRFs:

- polynomial form (of pseudo-boolean function) has negative coefficients on all bi-linear terms;
- posiform has pairwise terms of the form $u\bar{v}$;
- all pairwise potentials satisfy $\psi^P_{ij}(0,1) + \psi^P_{ij}(1,0) \geq \psi^P_{ij}(1,1) + \psi^P_{ij}(0,0)$.

Minimum-cut Problem

Graph Cut

Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be a capacitated digraph with two distinguished vertices s and t. An st-cut is a partitioning of $\mathcal{V}$ into two disjoint sets $\mathcal{S}$ and $\mathcal{T}$ such that $s \in \mathcal{S}$ and $t \in \mathcal{T}$. The cost of the cut is the sum of edge capacities for all edges going from $\mathcal{S}$ to $\mathcal{T}$.



Quadratic Pseudo-boolean Optimization

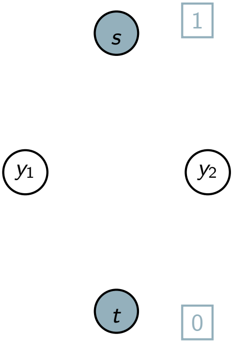
Main idea:

- construct a graph such that every *st*-cut corresponds to a joint assignment to the variables $\mathbf{y}$
- the cost of the cut should be equal to the energy of the assignment, $E(\mathbf{y}; \mathbf{x})$.*
- the minimum-cut then corresponds to the the minimum energy assignment, $\mathbf{y}^\star = \operatorname{argmin}_{\mathbf{y}} E(\mathbf{y}; \mathbf{x})$.

*Requires non-negative energies.

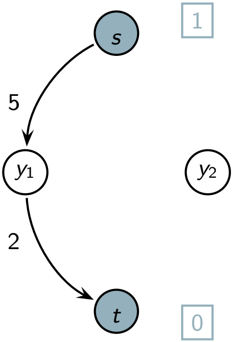
Example *st*-Graph Construction for Binary MRF

$$E(y_1, y_2) = \psi_1(y_1) + \psi_2(y_2) + \psi_{ij}(y_1, y_2)$$



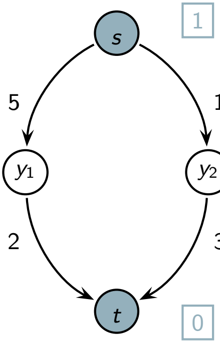
Example *st*-Graph Construction for Binary MRF

$$
\begin{aligned}
E\left(y_1, y_2\right) &= \psi_1(y_1) + \psi_2(y_2) + \psi_{ij}(y_1, y_2) \\
&= 2y_1 + 5\bar{y}_1
\end{aligned}
$$



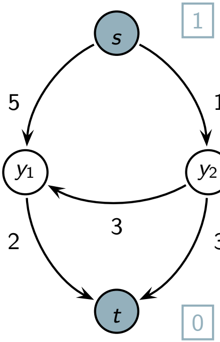
Example *st*-Graph Construction for Binary MRF

$$
\begin{aligned}
E(y_1, y_2) &= \psi_1(y_1) + \psi_2(y_2) + \psi_{ij}(y_1, y_2) \\
&= 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2
\end{aligned}
$$



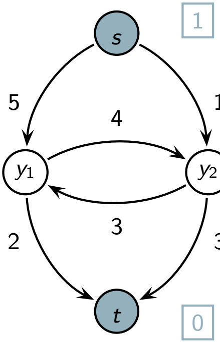
Example *st*-Graph Construction for Binary MRF

$$
\begin{aligned}
E(y_1, y_2) &= \psi_1(y_1) + \psi_2(y_2) + \psi_{ij}(y_1, y_2) \\
&= 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2 + 3\bar{y}_1 y_2
\end{aligned}
$$



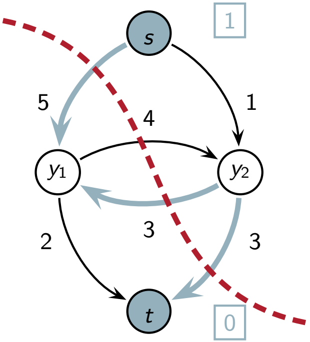
Example *st*-Graph Construction for Binary MRF

$$E(y_1, y_2) = \psi_1(y_1) + \psi_2(y_2) + \psi_{ij}(y_1, y_2)$$
$$= 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2 + 3\bar{y}_1 y_2 + 4y_1\bar{y}_2$$



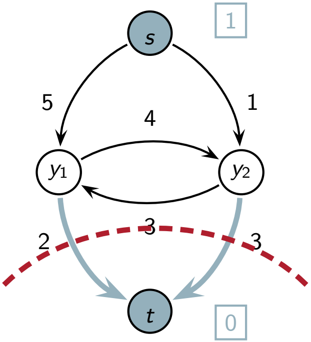
An Example *st*-Cut

$$E(0,1) = \psi_1(0) + \psi_2(1) + \psi_{ij}(0,1)$$
$$= 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2 + 3\bar{y}_1 y_2 + 4y_1\bar{y}_2$$



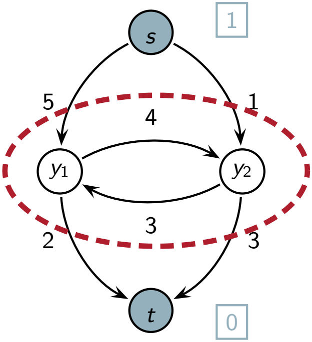
Another *st*-Cut

$$E\,(1,1) = \psi_1(1) + \psi_2(1) + \psi_{ij}(1,1)$$
$$= 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2 + 3\bar{y}_1 y_2 + 4y_1\bar{y}_2$$



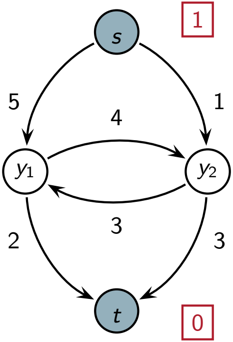
Invalid *st*-Cut

This is not a valid cut, since it does not correspond to a partitioning of the nodes into two sets—one containing s and one containing t.



Alternative *st*-Graph Construction

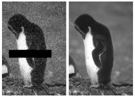
Sometimes you will see the roles of s and t switched.

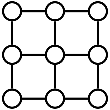


These graphs represent the same energy function.

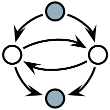
Big Picture: Where are we?

We can now formulate inference in a submodular binary pairwise MRF as a minimum-cut problem.





$\{0, 1\}^n \rightarrow \mathbb{R}$

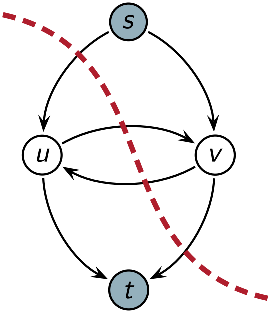


How do we solve the minimum-cut problem?

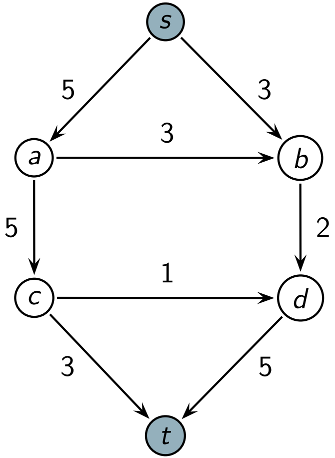
Max-flow/Min-cut Theorem

Max-flow/Min-cut Theorem [Fulkerson, 1956]

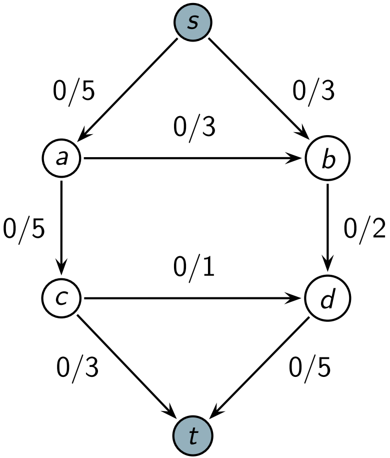
The maximum flow f from vertex s to vertex t is equal to the minimum cost st-cut.



Maximum Flow Example



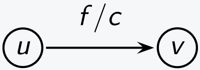
Maximum Flow Example (Augmenting Path)



flow

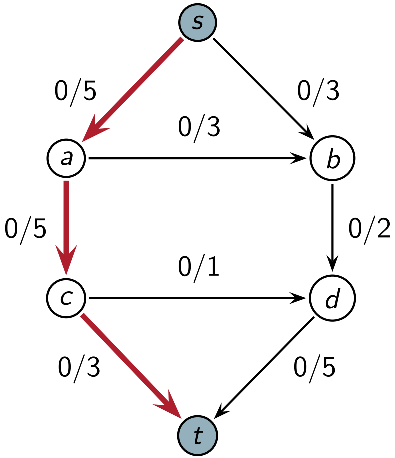
0

notation



edge with capacity c, and current flow f.

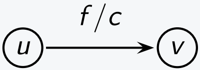
Maximum Flow Example (Augmenting Path)



flow

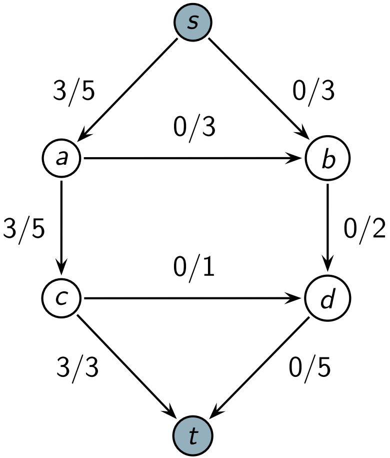
0

notation



edge with capacity c, and current flow f.

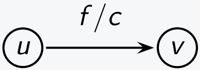
Maximum Flow Example (Augmenting Path)



flow

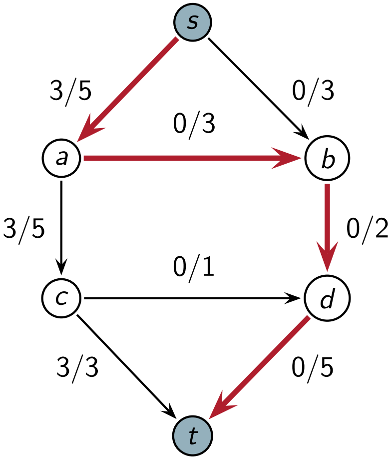
3

notation



edge with capacity c, and current flow f.

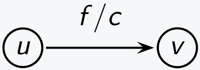
Maximum Flow Example (Augmenting Path)



flow

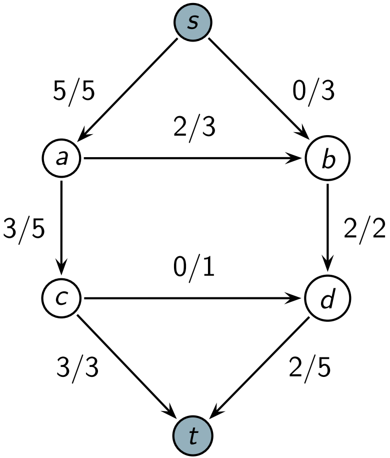
3

notation



edge with capacity c, and current flow f.

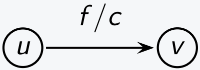
Maximum Flow Example (Augmenting Path)



flow

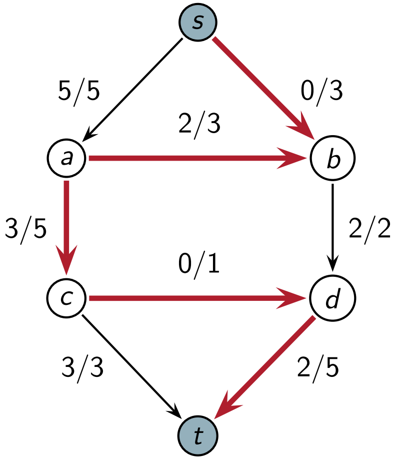
5

notation



edge with capacity c, and current flow f.

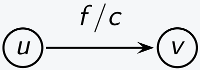
Maximum Flow Example (Augmenting Path)



flow

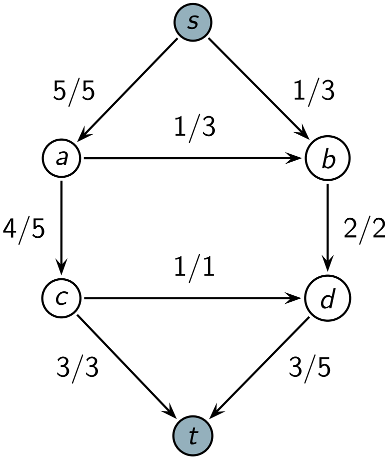
5

notation



edge with capacity c, and current flow f.

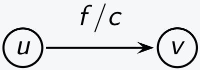
Maximum Flow Example (Augmenting Path)



flow

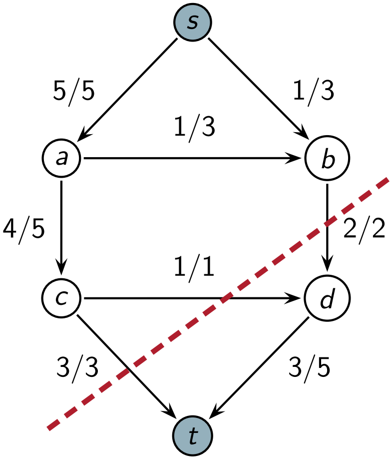
6

notation



edge with capacity c, and current flow f.

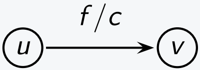
Maximum Flow Example (Augmenting Path)



flow

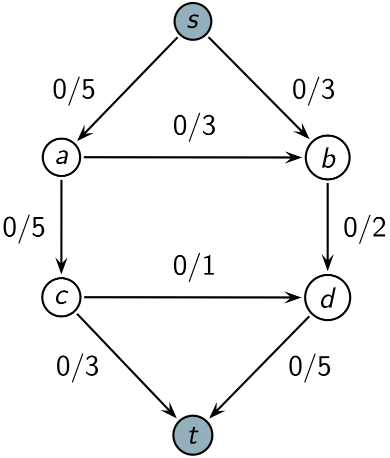
6

notation



edge with capacity c, and current flow f.

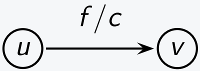
Maximum Flow Example (Push-Relabel)



state

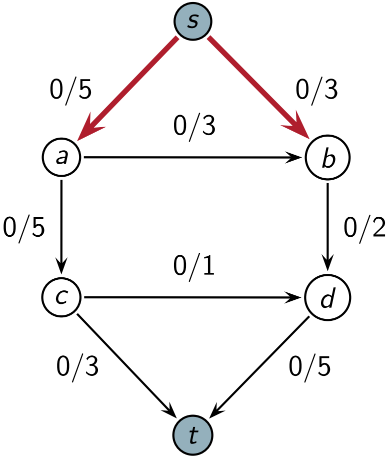
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	0	0
b	0	0
c	0	0
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

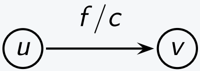
Maximum Flow Example (Push-Relabel)



state

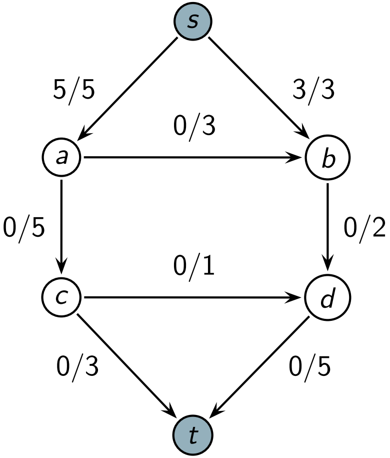
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	0	0
b	0	0
c	0	0
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

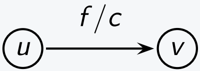
Maximum Flow Example (Push-Relabel)



state

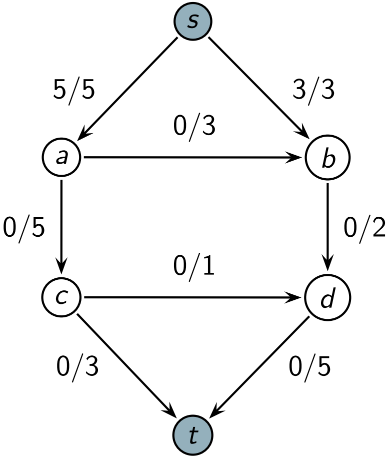
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	0	5
b	0	3
c	0	0
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

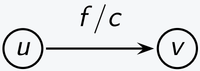
Maximum Flow Example (Push-Relabel)



state

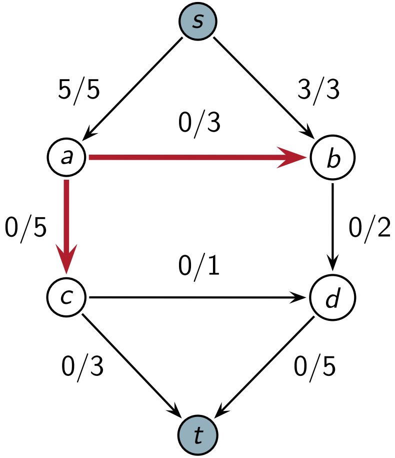
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	5
b	0	3
c	0	0
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

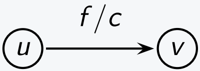
Maximum Flow Example (Push-Relabel)



state

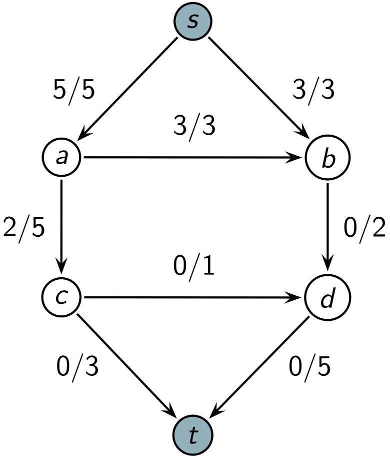
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	5
b	0	3
c	0	0
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

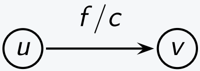
Maximum Flow Example (Push-Relabel)



state

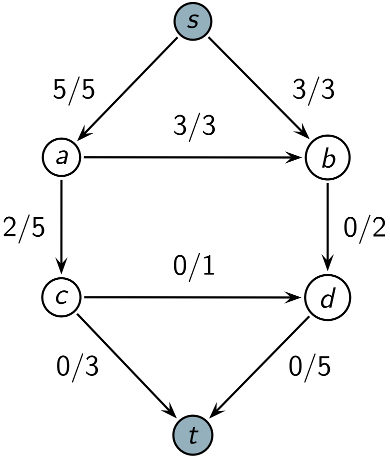
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	0	6
c	0	2
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

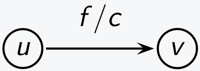
Maximum Flow Example (Push-Relabel)



state

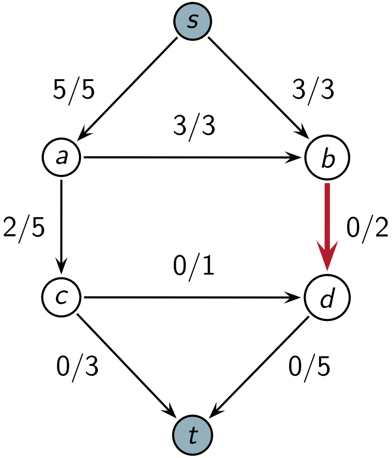
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	6
c	0	2
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

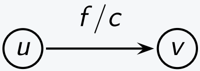
Maximum Flow Example (Push-Relabel)



state

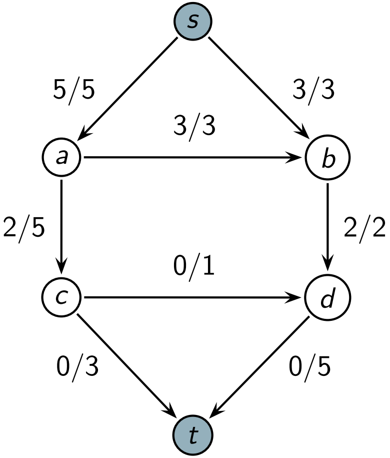
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	6
c	0	2
d	0	0
t	0	0

notation



edge with capacity c, current flow f.

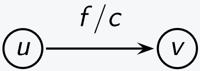
Maximum Flow Example (Push-Relabel)



state

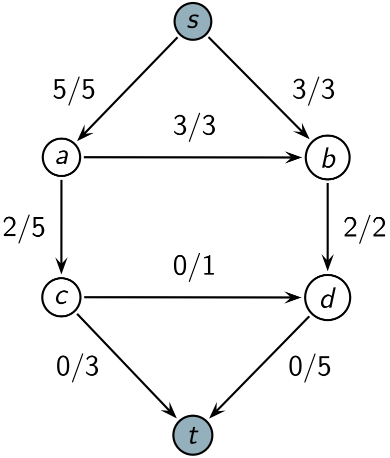
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	4
c	0	2
d	0	2
t	0	0

notation



edge with capacity c, current flow f.

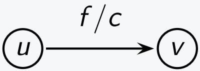
Maximum Flow Example (Push-Relabel)



state

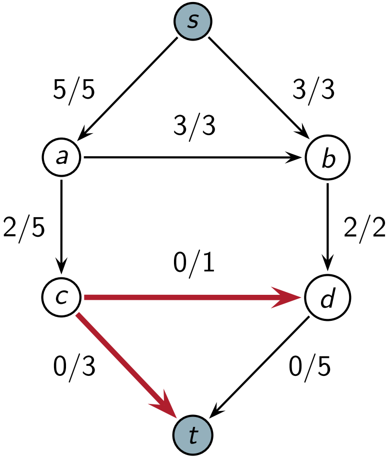
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	4
c	1	2
d	0	2
t	0	0

notation



edge with capacity c, current flow f.

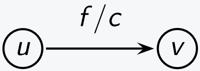
Maximum Flow Example (Push-Relabel)



state

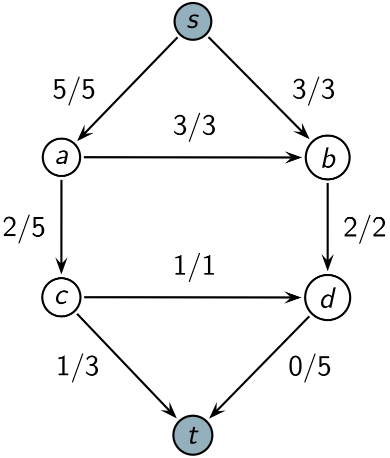
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	4
c	1	2
d	0	2
t	0	0

notation



edge with capacity c, current flow f.

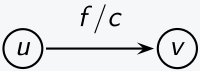
Maximum Flow Example (Push-Relabel)



state

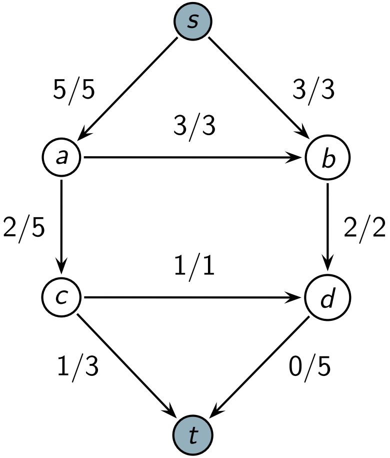
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	4
c	1	0
d	0	3
t	0	1

notation



edge with capacity c, current flow f.

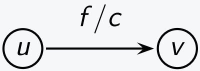
Maximum Flow Example (Push-Relabel)



state

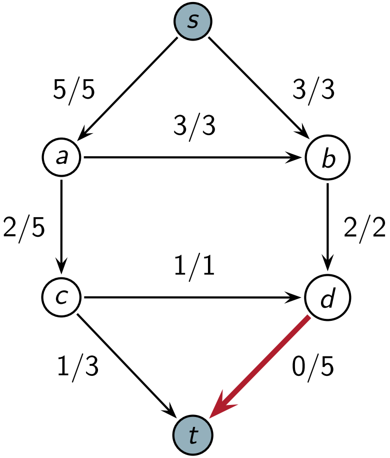
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	4
c	1	0
d	1	3
t	0	1

notation



edge with capacity c, current flow f.

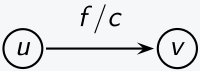
Maximum Flow Example (Push-Relabel)



state

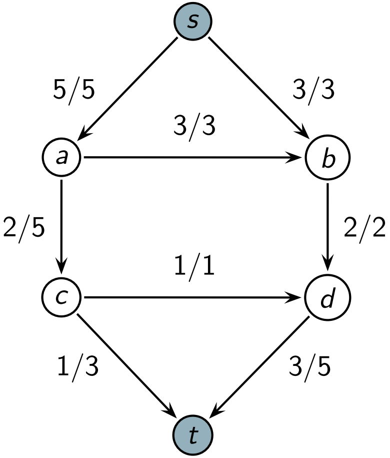
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	4
c	1	0
d	1	3
t	0	1

notation



edge with capacity c, current flow f.

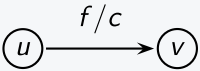
Maximum Flow Example (Push-Relabel)



state

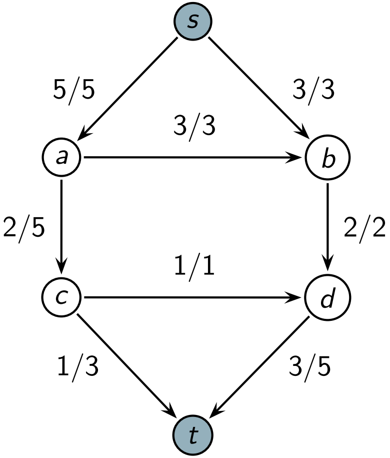
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	1	4
c	1	0
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

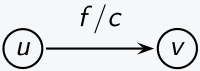
Maximum Flow Example (Push-Relabel)



state

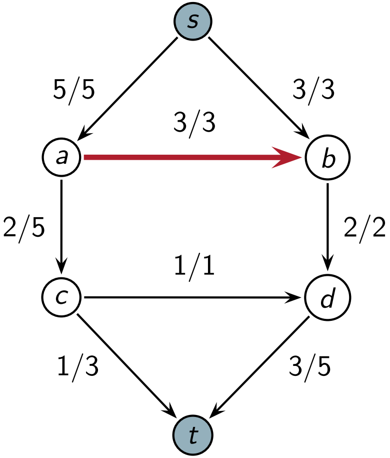
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	2	4
c	1	0
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

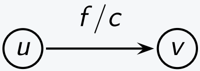
Maximum Flow Example (Push-Relabel)



state

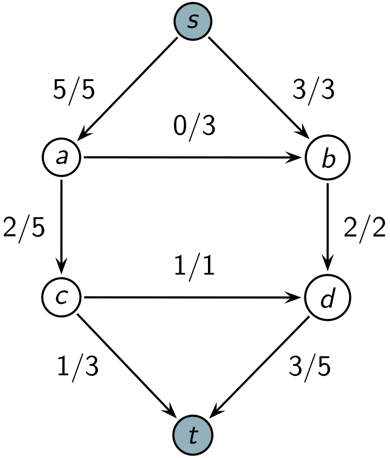
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	0
b	2	4
c	1	0
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

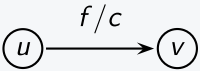
Maximum Flow Example (Push-Relabel)



state

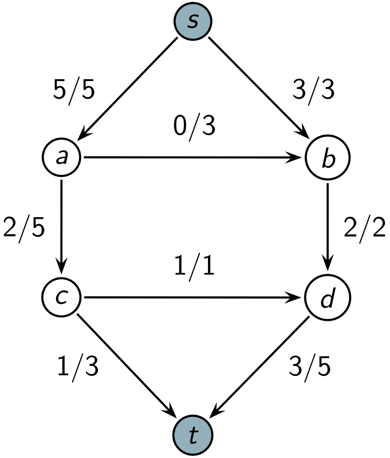
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	1	3
b	2	1
c	1	0
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

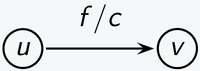
Maximum Flow Example (Push-Relabel)



state

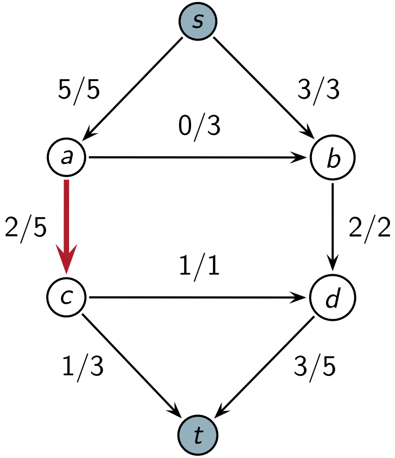
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	3
b	2	1
c	1	0
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

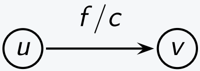
Maximum Flow Example (Push-Relabel)



state

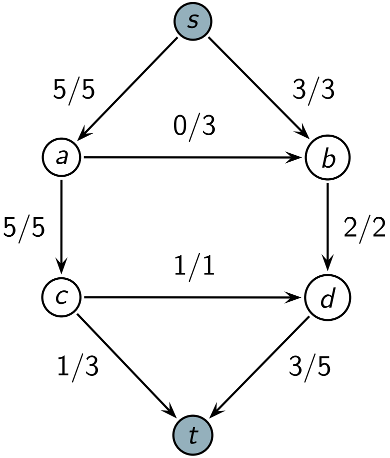
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	3
b	2	1
c	1	0
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

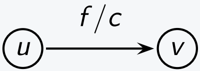
Maximum Flow Example (Push-Relabel)



state

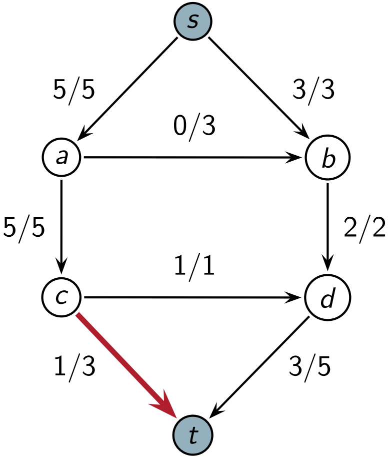
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	2	1
c	1	3
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

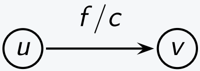
Maximum Flow Example (Push-Relabel)



state

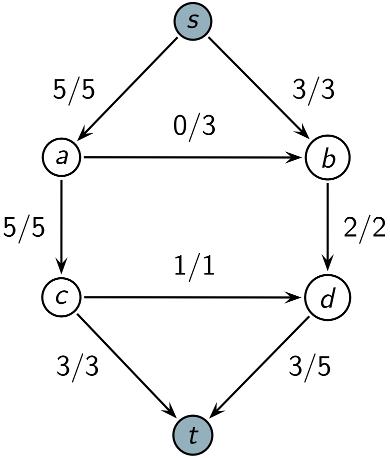
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	2	1
c	1	3
d	1	0
t	0	4

notation



edge with capacity c, current flow f.

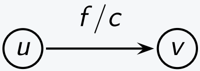
Maximum Flow Example (Push-Relabel)



state

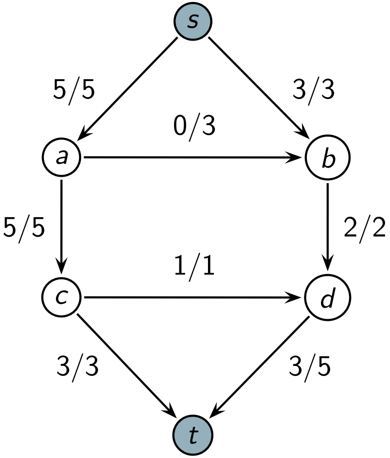
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	2	1
c	1	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

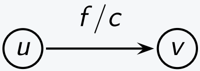
Maximum Flow Example (Push-Relabel)



state

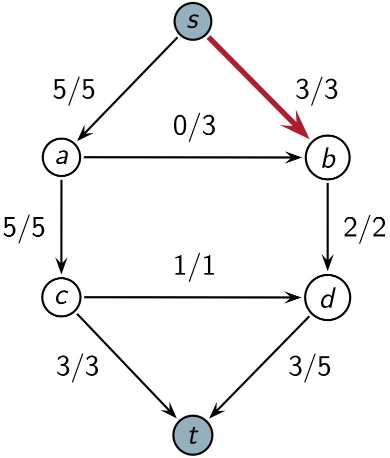
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	7	1
c	1	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

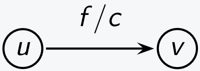
Maximum Flow Example (Push-Relabel)



state

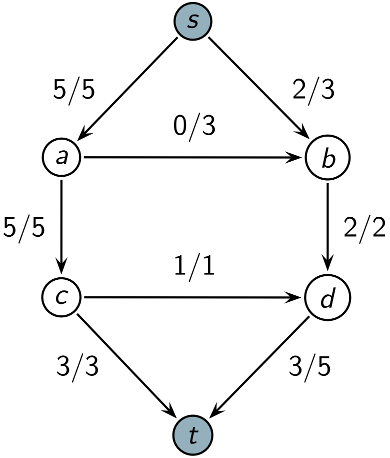
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	7	1
c	1	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

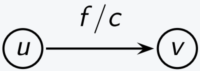
Maximum Flow Example (Push-Relabel)



state

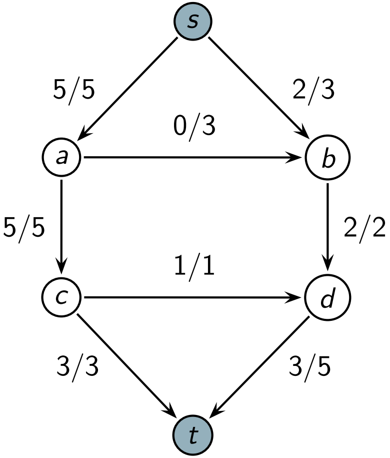
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	7	0
c	1	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

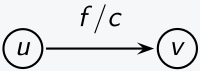
Maximum Flow Example (Push-Relabel)



state

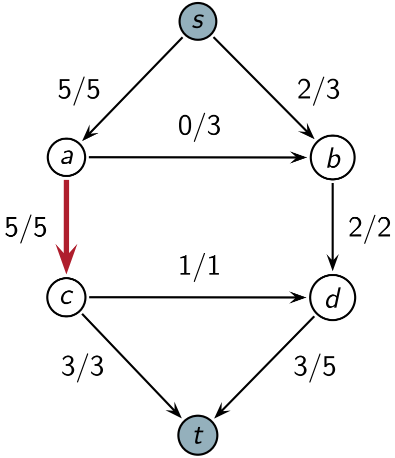
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	7	0
c	3	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

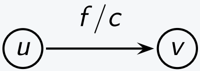
Maximum Flow Example (Push-Relabel)



state

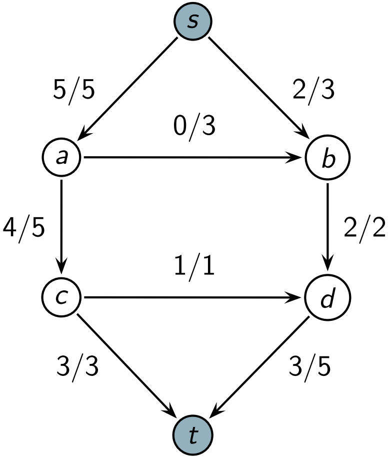
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	0
b	7	0
c	3	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

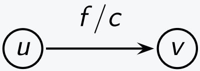
Maximum Flow Example (Push-Relabel)



state

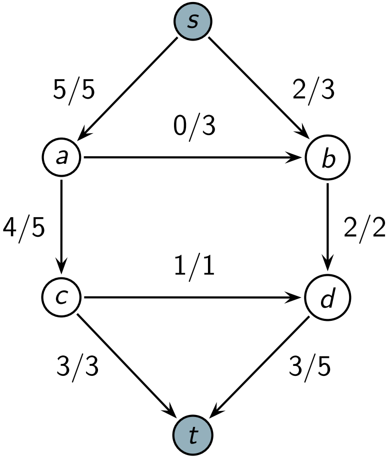
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	2	1
b	7	0
c	3	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

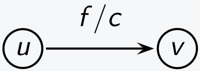
Maximum Flow Example (Push-Relabel)



state

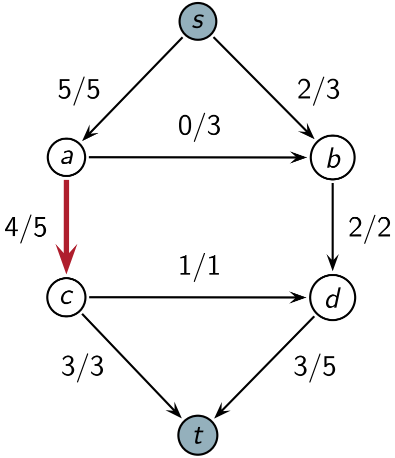
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	4	1
b	7	0
c	3	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

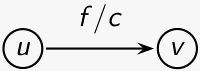
Maximum Flow Example (Push-Relabel)



state

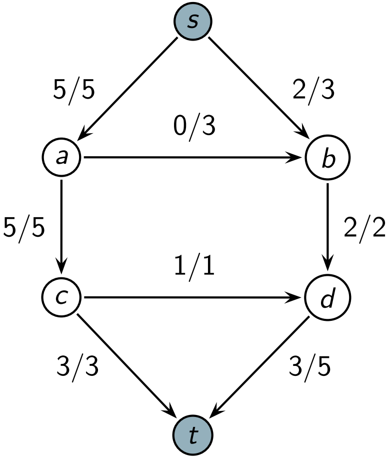
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	4	1
b	7	0
c	3	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

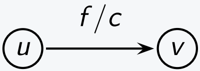
Maximum Flow Example (Push-Relabel)



state

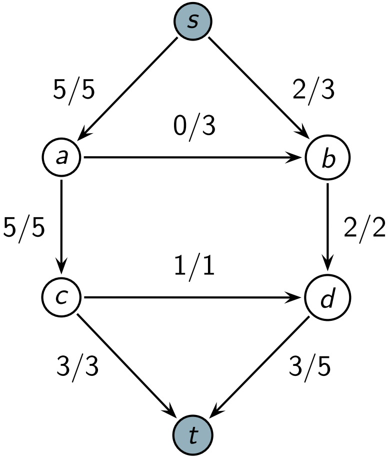
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	4	0
b	7	0
c	3	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

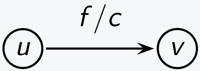
Maximum Flow Example (Push-Relabel)



state

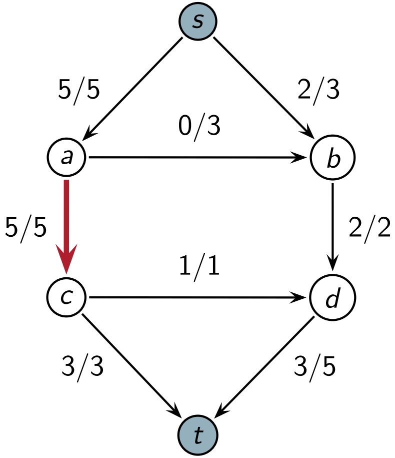
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	4	0
b	7	0
c	5	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

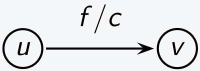
Maximum Flow Example (Push-Relabel)



state

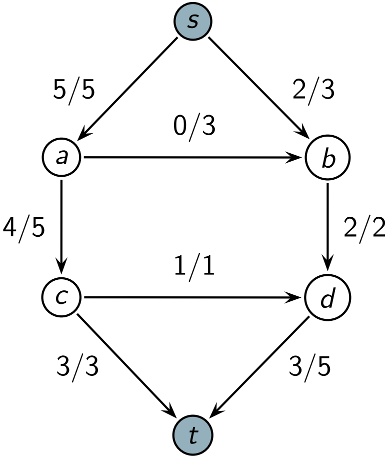
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	4	0
b	7	0
c	5	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

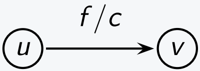
Maximum Flow Example (Push-Relabel)



state

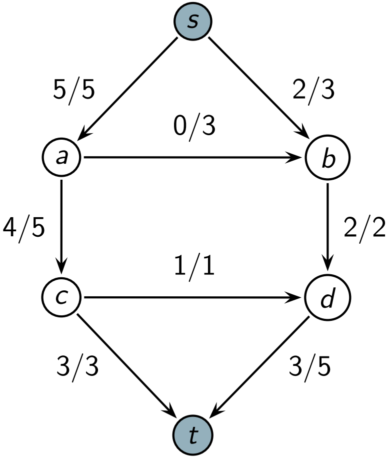
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	4	1
b	7	0
c	5	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

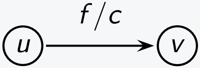
Maximum Flow Example (Push-Relabel)



state

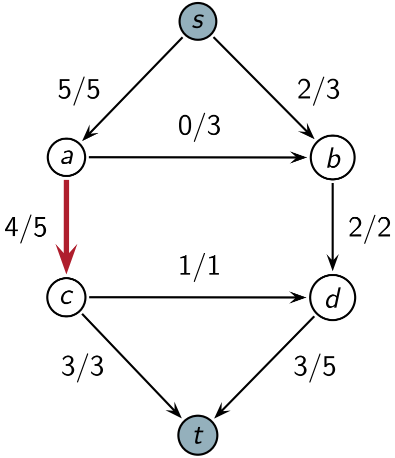
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	6	1
b	7	0
c	5	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

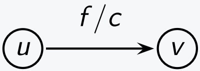
Maximum Flow Example (Push-Relabel)



state

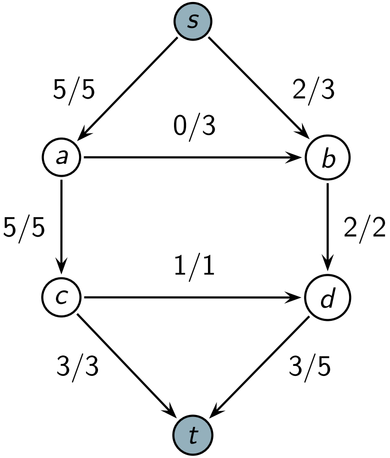
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	6	1
b	7	0
c	5	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

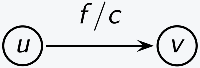
Maximum Flow Example (Push-Relabel)



state

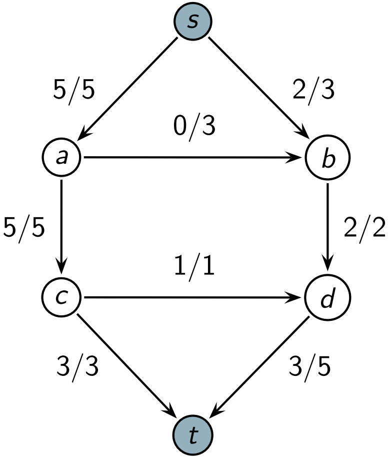
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	6	0
b	7	0
c	5	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

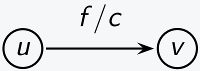
Maximum Flow Example (Push-Relabel)



state

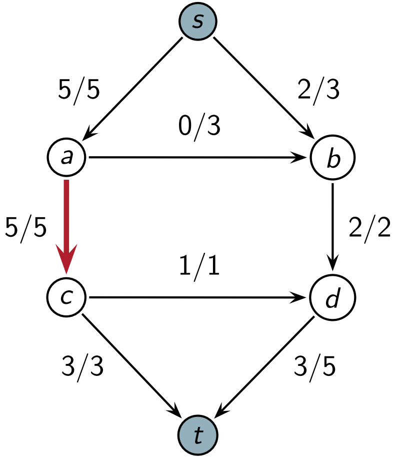
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	6	0
b	7	0
c	7	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

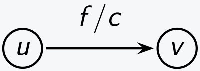
Maximum Flow Example (Push-Relabel)



state

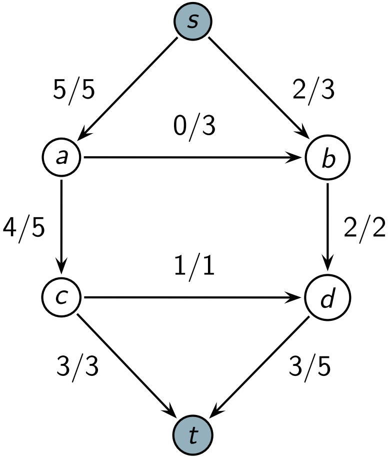
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	6	0
b	7	0
c	7	1
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

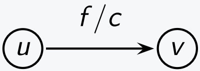
Maximum Flow Example (Push-Relabel)



state

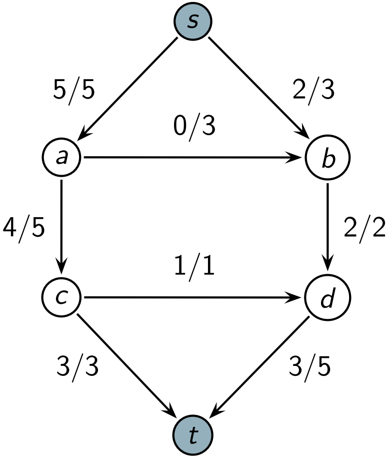
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	6	1
b	7	0
c	7	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

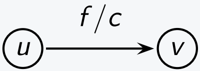
Maximum Flow Example (Push-Relabel)



state

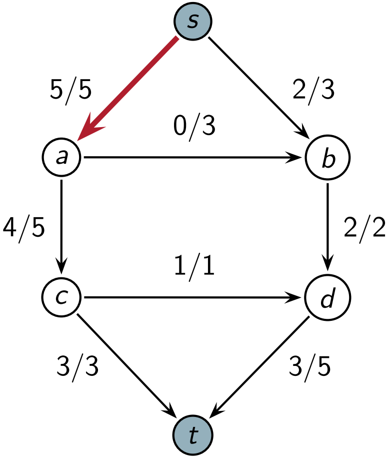
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	7	1
b	7	0
c	7	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

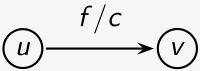
Maximum Flow Example (Push-Relabel)



state

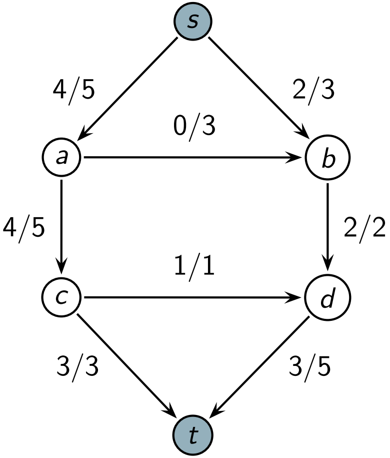
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	7	1
b	7	0
c	7	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

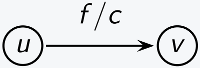
Maximum Flow Example (Push-Relabel)



state

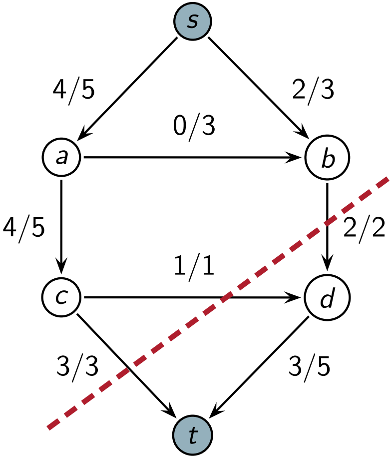
	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	7	0
b	7	0
c	7	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

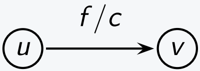
Maximum Flow Example (Push-Relabel)



state

	$h(\cdot)$	$e(\cdot)$
s	6	∞
a	7	0
b	7	0
c	7	0
d	1	0
t	0	6

notation



edge with capacity c, current flow f.

Comparison of Maximum Flow Algorithms

Current state-of-the-art algorithm for exact minimization of general submodular pseudo-Boolean functions is $O(n^5 T + n^6)$, where T is the time taken to evaluate the function [Orlin, 2007].

† assumes integer capacities

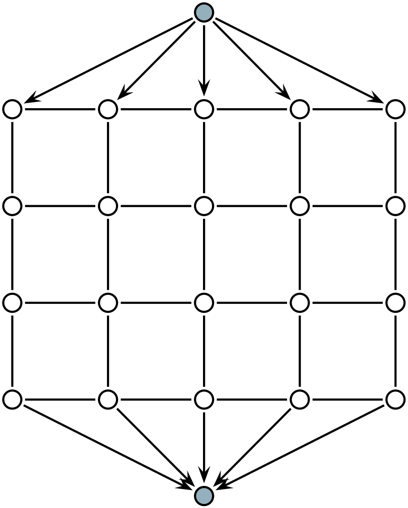
Comparison of Maximum Flow Algorithms

Current state-of-the-art algorithm for exact minimization of general submodular pseudo-Boolean functions is $O(n^5 T + n^6)$, where T is the time taken to evaluate the function [Orlin, 2007].

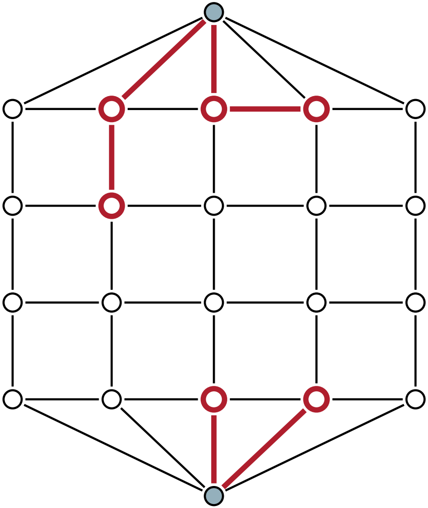
Algorithm	**Complexity**
Ford-Fulkerson	$O(E \max f)^\dagger$
Edmonds-Karp (BFS)	$O(VE^2)$
Push-relabel	$O(V^3)$
Boykov-Kolmogorov	$O(V^2 E \max f)$ ($\sim O(V)$ in practice)

† assumes integer capacities

Maximum Flow (Boykov-Kolmogorov, PAMI 2004)



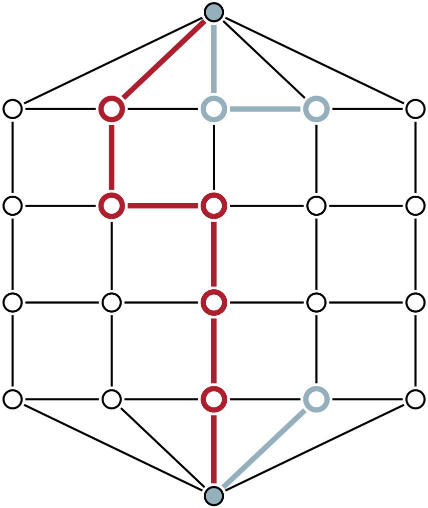
Maximum Flow (Boykov-Kolmogorov, PAMI 2004)



growth stage

search trees from s and t grow until they touch

Maximum Flow (Boykov-Kolmogorov, PAMI 2004)



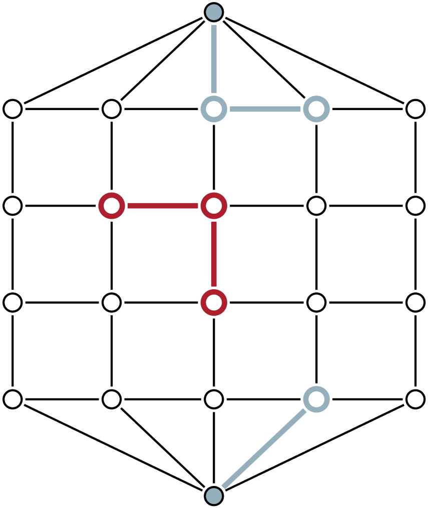
growth stage

search trees from s and t grow until they touch

augmentation stage

the path found is augmented

Maximum Flow (Boykov-Kolmogorov, PAMI 2004)



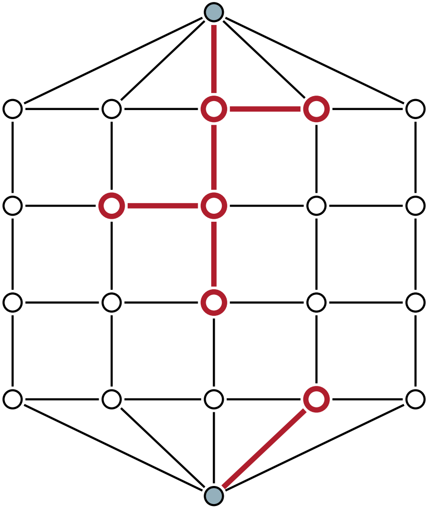
growth stage

search trees from s and t grow until they touch

augmentation stage

the path found is augmented; trees break into forests

Maximum Flow (Boykov-Kolmogorov, PAMI 2004)



growth stage

search trees from s and t grow until they touch

augmentation stage

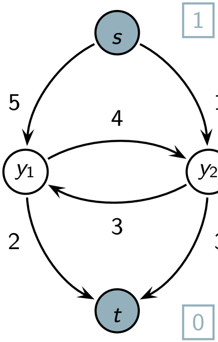
the path found is augmented; trees break into forests

adoption stage

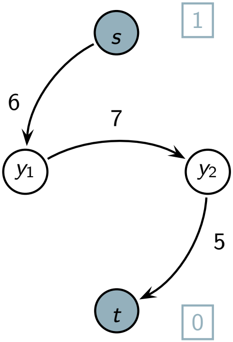
trees are restored

Reparameterization of Energy Functions

$$E(y_1, y_2) = 2y_1 + 5\bar{y}_1 + 3y_2 + \bar{y}_2 + 3\bar{y}_1 y_2 + 4y_1\bar{y}_2$$

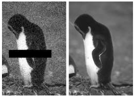


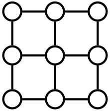
$$E(y_1, y_2) = 6\bar{y}_1 + 5y_2 + 7y_1\bar{y}_2$$



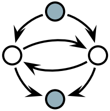
Big Picture: Where are we now?

We can perform inference in submodular binary pairwise Markov random fields exactly.



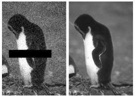


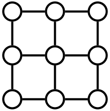
$\{0,1\}^n \to \mathbb{R}$



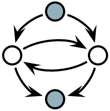
Big Picture: Where are we now?

We can perform inference in submodular binary pairwise Markov random fields exactly.





$$\{0,1\}^n \to \mathbb{R}$$



What about...

- non-submodular binary pairwise Markov random fields?
- multi-label Markov random fields?
- higher-order Markov random fields? (part 3)

Non-submodular Binary Pairwise MRFs

Non-submodular binary pairwise MRFs have potentials that do not satisfy $\psi^P_{ij}(0,1) + \psi^P_{ij}(1,0) \geq \psi^P_{ij}(1,1) + \psi^P_{ij}(0,0)$.

They are often handled in one of the following ways:

- approximate the energy function by one that is submodular (i.e., project onto the space of submodular functions);
- solve a relaxation of the problem using QPBO (Rother et al., 2007) or dual-decomposition (Komodakis et al., 2007).

Approximating Non-submodular Binary Pairwise MRFs

Consider the non-submodular potential

A	B
C	D

with $A + D > B + C$.

We can project onto a submodular potential by modifying the coefficients as follows:

$$\Delta = A + D - C - B$$
$$A \leftarrow A - \frac{\Delta}{3}$$
$$C \leftarrow C + \frac{\Delta}{3}$$
$$B \leftarrow B + \frac{\Delta}{3}$$

QPBO (Roof Duality) [Rother et al., 2007]

Consider the energy function

$$E(\mathbf{y}) = \underbrace{\sum_{i \in \mathcal{V}} \psi_i^U(y_i) + \sum_{ij \in \mathcal{E}} \psi_{ij}^P(y_i, y_j)}_{} + \sum_{ij \in \mathcal{E}} \tilde{\psi}_{ij}^P(y_i, y_j)$$

where the pairwise terms $\sum_{ij \in \mathcal{E}} \psi_{ij}^P(y_i, y_j)$ are submodular and the terms $\sum_{ij \in \mathcal{E}} \tilde{\psi}_{ij}^P(y_i, y_j)$ are non-submodular.

We can introduce duplicate variables $\bar{y}_i$ into the energy function, and write

$$E'(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i \in \mathcal{V}} \frac{\psi_i^U(y_i) + \psi_i^U(1 - \bar{y}_i)}{2} + \sum_{ij \in \mathcal{E}} \frac{\psi_{ij}^P(y_i, y_j) + \psi_{ij}^P(1 - \bar{y}_i, 1 - \bar{y}_j)}{2} + \sum_{ij \in \mathcal{E}} \frac{\tilde{\psi}_{ij}^P(y_i, 1 - \bar{y}_j) + \tilde{\psi}_{ij}^P(1 - \bar{y}_i, y_j)}{2}$$

QPBO (Roof Duality)

$$E'(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i \in \mathcal{V}} \tfrac{1}{2}\psi_i^U(y_i) + \tfrac{1}{2}\psi_i^U(1 - \bar{y}_i)$$
$$+ \sum_{ij \in \mathcal{E}} \tfrac{1}{2}\psi_{ij}^P(y_i, y_j) + \tfrac{1}{2}\psi_{ij}^P(1 - \bar{y}_i, 1 - \bar{y}_j)$$
$$+ \sum_{ij \in \mathcal{E}} \tfrac{1}{2}\tilde{\psi}_{ij}^P(y_i, 1 - \bar{y}_j) + \tfrac{1}{2}\tilde{\psi}_{ij}^P(1 - \bar{y}_i, y_j)$$

Observations

- if $y_i = 1 - \bar{y}_i$ for all i, then $E(\mathbf{y}) = E'(\mathbf{y}, \bar{\mathbf{y}})$.
- $E'(\mathbf{y}, \bar{\mathbf{y}})$ is submodular.

QPBO (Roof Duality)

$$
\begin{aligned}
E'(\mathbf{y}, \bar{\mathbf{y}}) = &\sum_{i \in \mathcal{V}} \tfrac{1}{2}\psi_i^U(y_i) + \tfrac{1}{2}\psi_i^U(1 - \bar{y}_i) \\
&+ \sum_{ij \in \mathcal{E}} \tfrac{1}{2}\psi_{ij}^P(y_i, y_j) + \tfrac{1}{2}\psi_{ij}^P(1 - \bar{y}_i, 1 - \bar{y}_j) \\
&+ \sum_{ij \in \mathcal{E}} \tfrac{1}{2}\tilde{\psi}_{ij}^P(y_i, 1 - \bar{y}_j) + \tfrac{1}{2}\tilde{\psi}_{ij}^P(1 - \bar{y}_i, y_j)
\end{aligned}
$$

Observations

- if $y_i = 1 - \bar{y}_i$ for all i, then $E(\mathbf{y}) = E'(\mathbf{y}, \bar{\mathbf{y}})$.
- $E'(\mathbf{y}, \bar{\mathbf{y}})$ is submodular.

Ignore the constraint on $\bar{y}_i$ and solve anyway. Result satisfies *partial optimality*: if $\bar{y}_i = 1 - y_i$ then y_i is the optimal label.

Multi-label Markov Random Fields

The quadratic pseudo-Boolean optimization techniques described above cannot be applied directly to multi-label MRFs.

However...

- ...for certain MRFs we can transform the multi-label problem into a binary one exactly.
- ...we can project the multi-label problem onto a series of binary problems in a so-called *move-making* algorithm.

The "Battleship" Transform [Ishikawa, 2003]

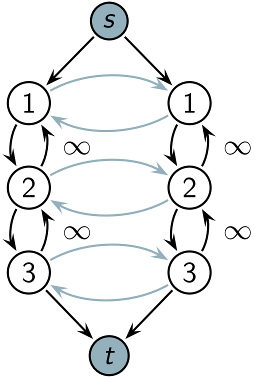
If the multi-label MRFs has pairwise potentials that are convex functions over the label differences, i.e., $\psi^P_{ij}(y_i, y_j) = g(|y_i - y_j|)$ where $g(\cdot)$ is convex, then we can transform the energy function into an equivalent binary one.

$$y = 1 \Leftrightarrow \mathbf{z} = (0, 0, 0)$$
$$y = 2 \Leftrightarrow \mathbf{z} = (1, 0, 0)$$
$$y = 3 \Leftrightarrow \mathbf{z} = (1, 1, 0)$$
$$y = 4 \Leftrightarrow \mathbf{z} = (1, 1, 1)$$

The "Battleship" Transform [Ishikawa, 2003]

If the multi-label MRFs has pairwise potentials that are convex functions over the label differences, i.e., $\psi^P_{ij}(y_i, y_j) = g(|y_i - y_j|)$ where $g(\cdot)$ is convex, then we can transform the energy function into an equivalent binary one.

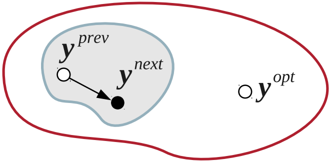
$$y = 1 \Leftrightarrow \mathbf{z} = (0, 0, 0)$$
$$y = 2 \Leftrightarrow \mathbf{z} = (1, 0, 0)$$
$$y = 3 \Leftrightarrow \mathbf{z} = (1, 1, 0)$$
$$y = 4 \Leftrightarrow \mathbf{z} = (1, 1, 1)$$



Move-making Inference

Idea:

- initialize $\mathbf{y}^{\text{prev}}$ to any valid assignment
- restrict the label-space of each variable y_i from $\mathcal{L}$ to $\mathcal{Y}_i \subseteq \mathcal{L}$ (with $y_i^{\text{prev}} \in \mathcal{Y}_i$)
- transform $E : \mathcal{L}^n \to \mathbb{R}$ to $\hat{E} : \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_n \to \mathbb{R}$
- find the optimal assignment $\hat{\mathbf{y}}$ for $\hat{E}$ and repeat



each move results in an assignment with lower energy

Iterated Conditional Modes [Besag, 1986]

Reduce multi-variate inference to solving a series of univariate inference problems.

ICM move

For one of the variables y_i, set $\mathcal{Y}_i = \mathcal{L}$. Set $\mathcal{Y}_j = \{y_j^{\text{prev}}\}$ for all $j \neq i$ (i.e., hold all other variables fixed).

Iterated Conditional Modes [Besag, 1986]

Reduce multi-variate inference to solving a series of univariate inference problems.

ICM move

For one of the variables y_i, set $\mathcal{Y}_i = \mathcal{L}$. Set $\mathcal{Y}_j = \{y_j^{\text{prev}}\}$ for all $j \neq i$ (i.e., hold all other variables fixed).

Can be used for arbitrary energy functions.

Alpha Expansion and Alpha-Beta Swap [Boykov et al., 2001]

Reduce multi-label inference to solving a series of binary (submodular) inference problems.

α-expansion move

Choose some $\alpha \in \mathcal{L}$. Then for all variables, set $\mathcal{Y}_i = \{\alpha, y_i^{\text{prev}}\}$.

$\alpha\beta$-swap move

Choose two labels $\alpha, \beta \in \mathcal{L}$. Then for each variable y_i such that $y_i^{\text{prev}} \in \{\alpha, \beta\}$, set $\mathcal{Y}_i = \{\alpha, \beta\}$. Otherwise set $\mathcal{Y}_i = \{y_i^{\text{prev}}\}$.

end of part 2