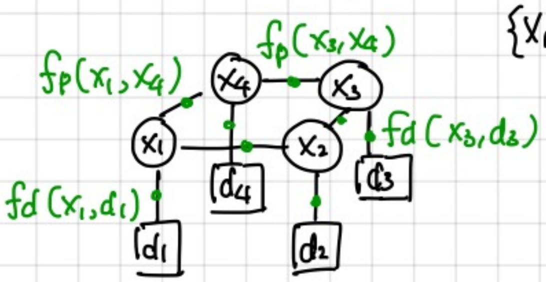
LECTURE 9: MRF OPTIMIZATION

[I] Motivation

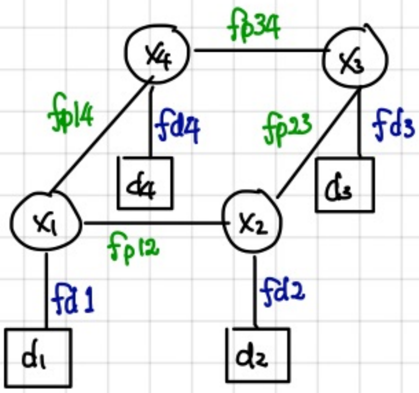
MRF :



$$\{x_1, \ldots, x_4\}^* = \underset{\{x\}}{\operatorname{argmin}} \underbrace{\sum_i fd(x_i, d_i) + \sum_{j \in N_i} fp(x_i, x_j)}$$

the value of each term indicates the values of the edges in the graph.

Having defined the data term (fd) & the prior term (fp) for every node (pixel), we can now have a complete graph:



Q: How can we optimize the graph so that we can predict the labels of x_1, x_2, x_3, x_4?

Q: What does it mean by 'optimizing' a graph?

A: Example: x_1 — x_2 & $x_i \in \{0, 1\}$

$$\{x_1, x_2\}^* = \underset{\{x_1, x_2\}}{\operatorname{argmin}} \; fp(x_1, x_2)$$

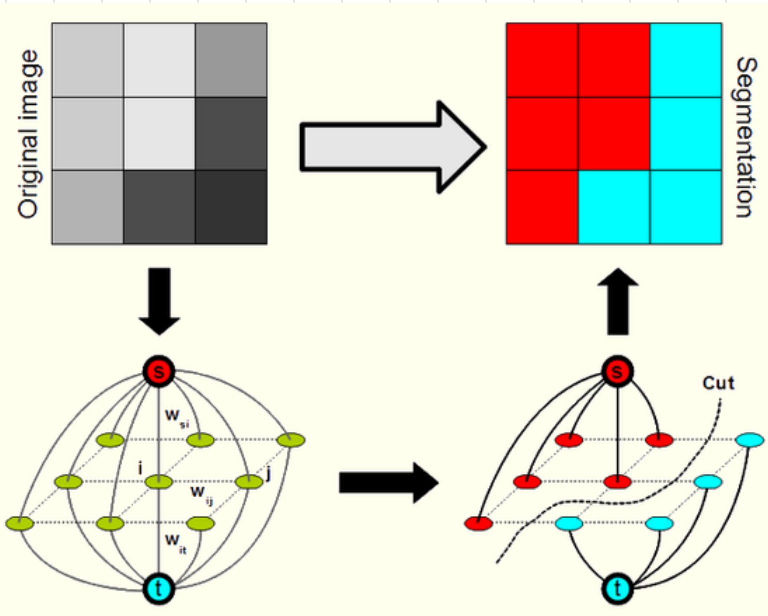
$$= \underset{\{x_1, x_2\}}{\operatorname{argmin}} \big[fp(x_1=0, x_2=0),\ fp(x_1=1, x_2=0),\ fp(x_1=1, x_2=0),\ fp(x_1=1, x_2=1) \big]$$

We find the values of x_1 & x_2 that minimize fp( ), or to choose one of four possible configurations.

[2] Graphcuts

There are two methods of graph optimization: Graphcuts & Belief Propagation. Here, we focus on graphcuts, particularly on binary graphcuts.

Basic concept: Imagine we have a 3x3 image. Some of the pixels belong to the foreground & some belong to the background. However, due to noise/outlier and color variations, we are not sure & thus want to determine which pixels belongs to which label.

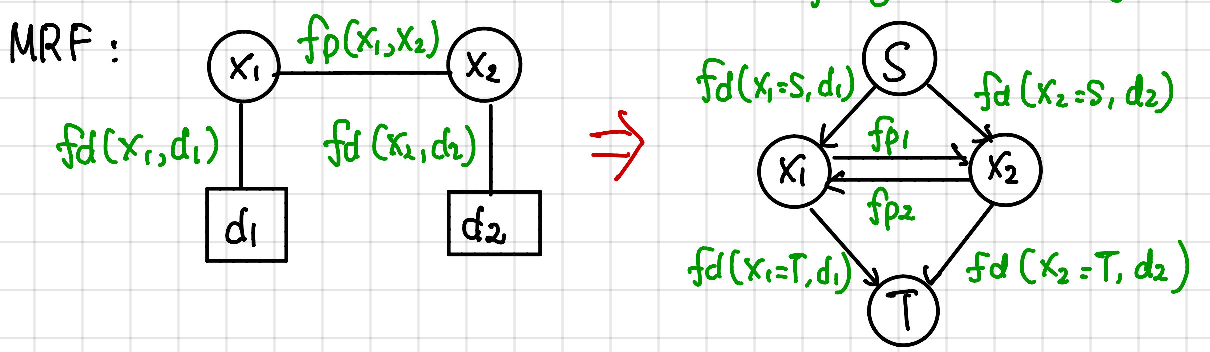


[3] Graphcuts: Data & Prior Terms

Q: How to assign the data & prior terms in a graph?

A: Suppose we have: x_1 & x_2 where $x_i = \{S, T\}$ (S: foreground, T: background)

and d_1, d_2

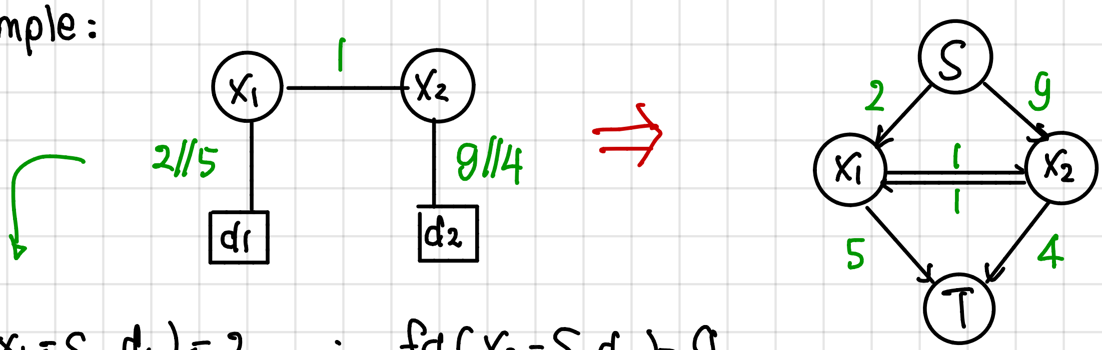


Assume:

$$f_p(x_1=T, x_2=S) = f_p(x_1=S, x_2=T) \rightarrow fp_1 = fp_2$$

$$f_p(x_1=T, x_2=T) = f_p(x_1=S, x_2=S) = 0$$

Example:



$fd(x_1 = S, d_1) = 2$; $fd(x_2 = S, d_2) = 9$

$fd(x_1 = T, d_1) = 5$; $fd(x_2 = T, d_2) = 4$

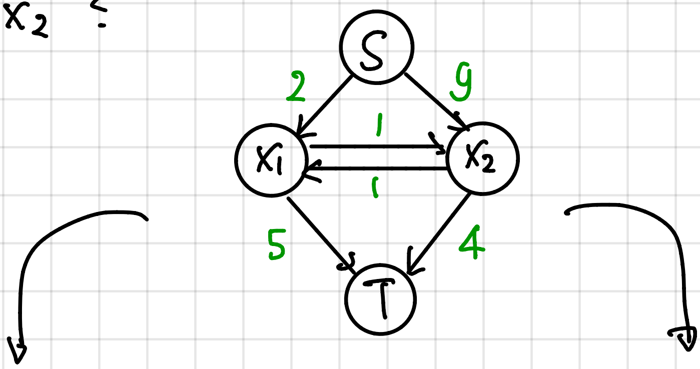
$fp(x_1 = S, x_2 = T) = fp(x_1 = T, x_2 = S) = 1$

$fp(x_1 = S, x_2 = S) = fp(x_1 = T, x_2 = T) = 0$

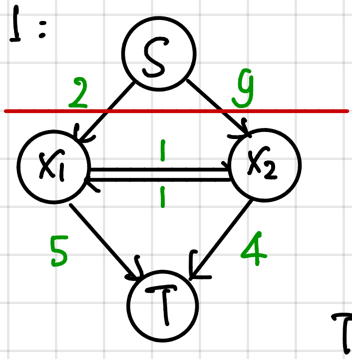
[4] Graphcuts: Max-Flow / Min-Cut

Q: How to optimize the graph, so that we know the labels of x_1 and x_2?

A:



cut 1:



cut cost:

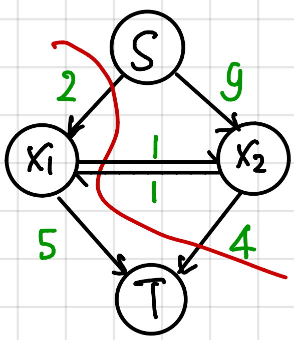
$9 + 2 = 11$

This cut implies:

$\begin{cases} x_1 = S \\ x_2 = S \end{cases}$

This is for minimization. For maximization: $x_1 = T$, $x_2 = T$

cut 2:



cut cost:

$2 + 1 + 4 = 7$

Implying:

$x_1 = S$

$x_2 = T$

Q: Which cut is better?

A: Cut 2 is cheaper than cut 1!

Q: How to find the minimum cut for all possible cuts?

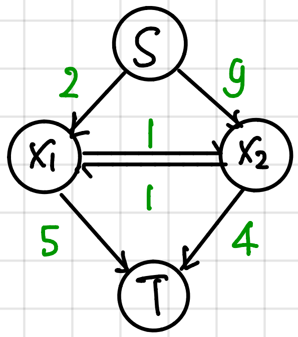
[5] Graphuals : Augmenting Path
(the Ford - Fulkerson method)

Basic components

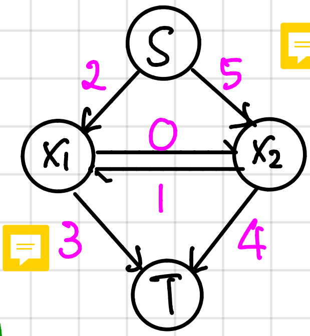
(1) Capacity graph

(2) Flow graph

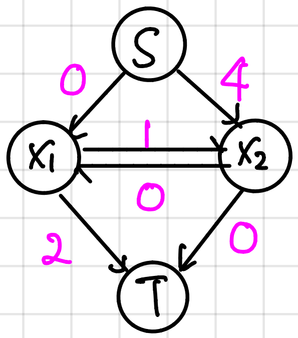
(3) Residual graph



−

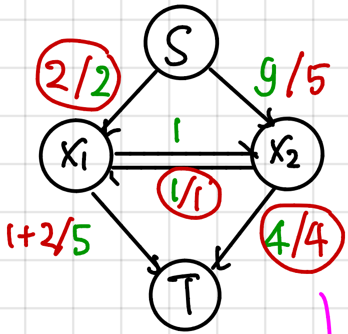


=

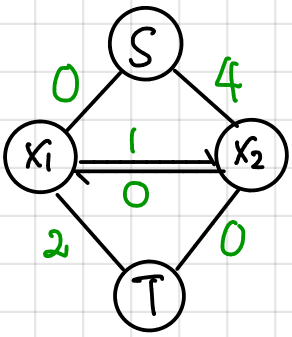


Calculation:

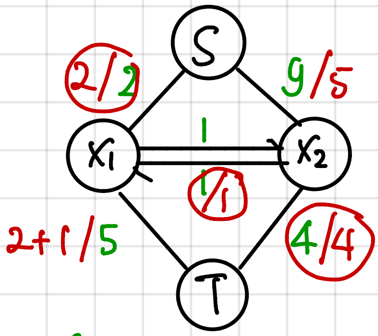
Start from 2 :



Residual :



Alternative 1 (Start from 9):

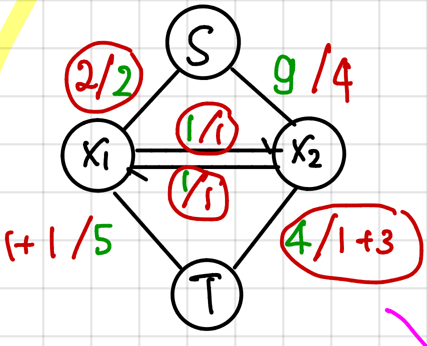


Must be the same!

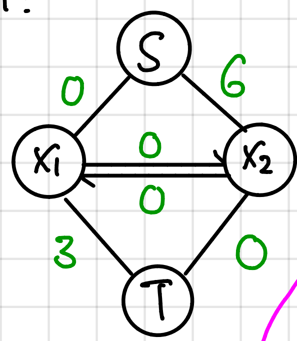
Cut cost = 2 + 1 + 4 = 7
(Min. cut)

Max-flow = 2 + 5 = 3 + 4 = 7

Alternative 2
(from 2 then 1):



Residual :

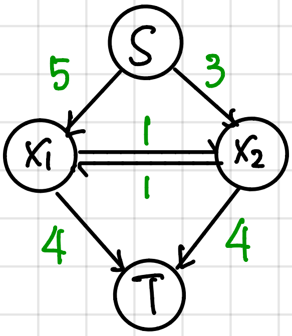


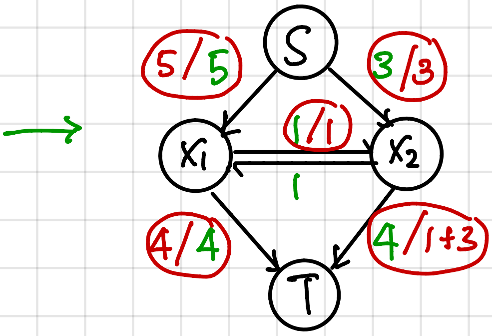
Cut cost = 2 + 1 + 1 + 4 = 8
(Min-cut)

Max-flow = 2 + 4 = 1 + 1 + 1 + 3 = 6

Other examples:

1.

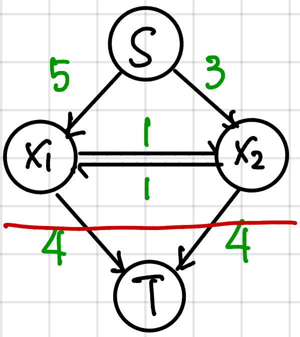




Min cut = 8

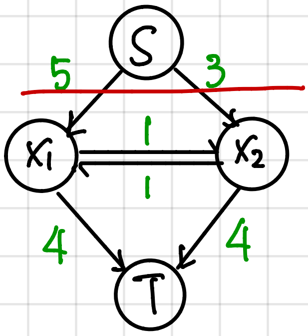
Max flow = 8

For this case, there are a few different ways of cutting:



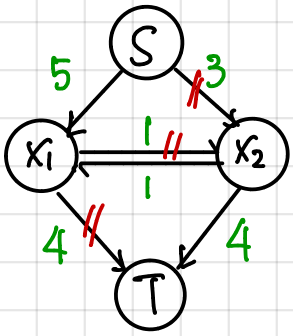
cut cost = 8

$x_1 = x_2 = T$



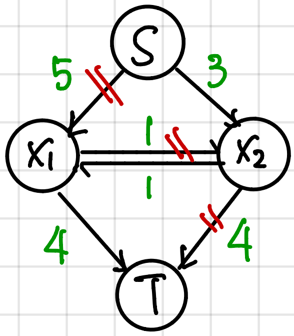
cut cost = 8

$x_1 = x_2 = S$



cut cost = 8

$x_1 = T$; $x_2 = S$



cut cost : 10

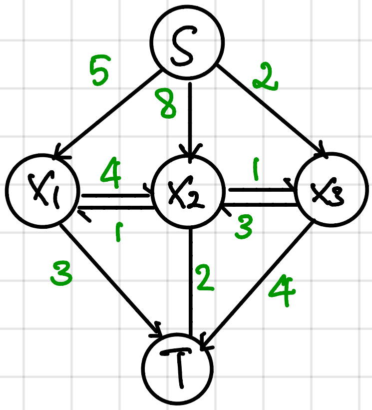
$x_1 = S$, $x_2 = T$

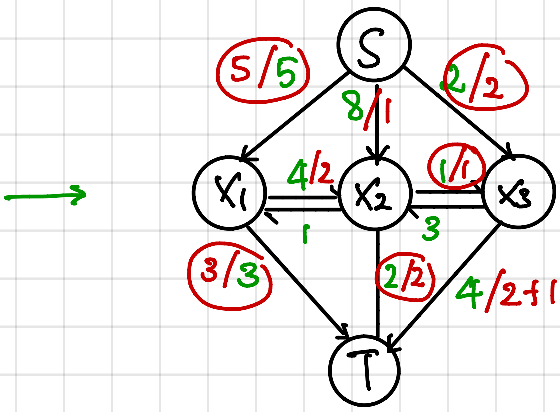
(not preffered because the cost is high)

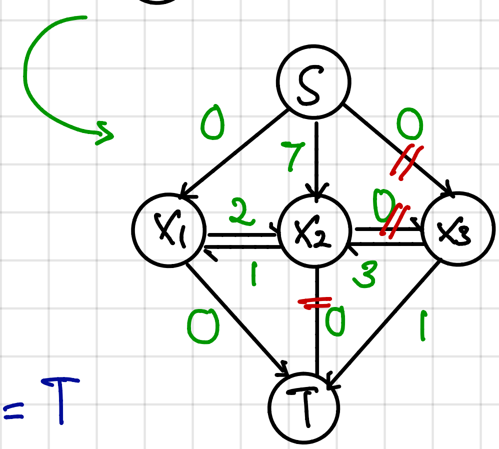
all these labelings are correct.

This implies there are ambiguities in the labeling.

2.







$x_2 = T$, $x_3 = S$ → total cost : $2+1+2=5$

$x_1 = \begin{cases} S \rightarrow \text{additional cost} = 5 \\ T \rightarrow \text{additional cost} = 3 \end{cases}$

We know the Max flow = 8, thus $x_1 = T$

[6] Belief Propagation (BP)

Aside from graphcuts, BP is another popular method for graph optimization. It's more general than graphcuts (can accept many labels) but much slower.

to prevent a very small / large value due to the product of m_{ji}

Formulation:

Scalar

Data term

Belief: $$b_i(x_i) = k \; \phi_i(x_i) \prod_{j \in N_i} m_{ji}(x_i)$$

Message passing: $$m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i) \, \psi_{ij}(x_i, x_j) \prod_{k \in N_i/j} m_{ki}(x_i)$$

Prior term

Neighbors of i except j.

Example: x_1 — x_2 ; $x_i = \{F, B\}$

$$b(x_1 = F) = k \; fd(x_1 = F, d) \; m_{21}(x_1 = F)$$

$$m_{21}(x_1 = F) = \sum_{x_2 \in \{F, B\}} fd(x_2, d_2) \; fp(x_2, x_1 = F)$$

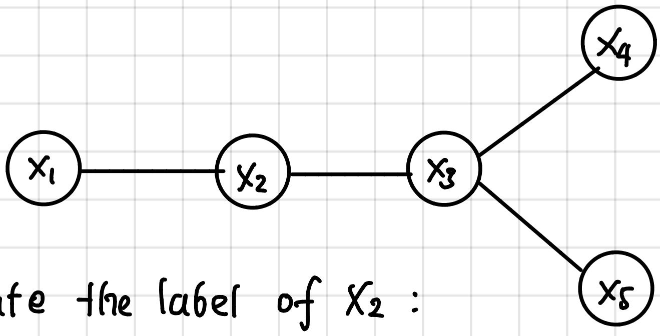
$$= fd(x_2 = F, d_2) \; fp(x_2 = F, x_1 = F) + fd(x_2 = B, d_2) \; fp(x_2 = B, x_1 = F)$$

Aside from $b(x_1 = F)$, we also compute $b(x_1 = B)$, and compare $b(x_1 = F)$ and $b(x_1 = B)$.

If $b(x_1 = F) > b(x_1 = B)$, then we label $x_1 = B$

For minimization

Another example:



Suppose we want to calculate the label of x_2:

$$b_2(x_2) = k\,\phi_2(x_2)\; m_{12}(x_2)\; m_{32}(x_2)$$

$$m_{12}(x_2) = \sum_{x_1} \phi_1(x_1)\; \psi_{12}(x_1, x_2)$$

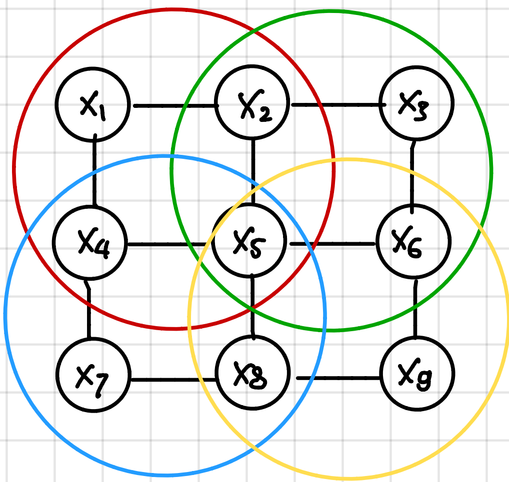
$$m_{32}(x_2) = \sum_{x_3} \phi_3(x_3)\; \psi_{32}(x_3, x_2)\; m_{43}(x_3)\; m_{53}(x_3)$$

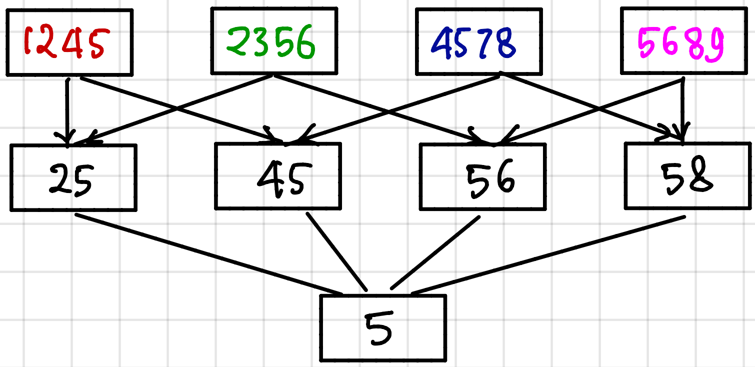
$$m_{43}(x_3) = \sum_{x_4} \phi_4(x_4)\; \psi_{43}(x_4, x_3)$$

$$m_{53}(x_3) = \sum_{x_5} \phi_5(x_5)\; \psi_{53}(x_5, x_3)$$

Note:

» The standard BP works only for a non-loopy graph.
For loopy graphs, one of the solutions is the Generalized BP.





$$b_5(x_5) = k\,\phi_5(x_5)\; m_{2\to5}\; m_{4\to5}\; m_{6\to5}\; m_{8\to5}$$

$$m_{4\to5}(x_5) = k \sum_{x_4} \phi_4(x_4)\; \psi_{45}(x_4, x_5)\; m_{12\to45}\; m_{78\to45}$$