UNIVERSITÉ Concordia UNIVERSITY

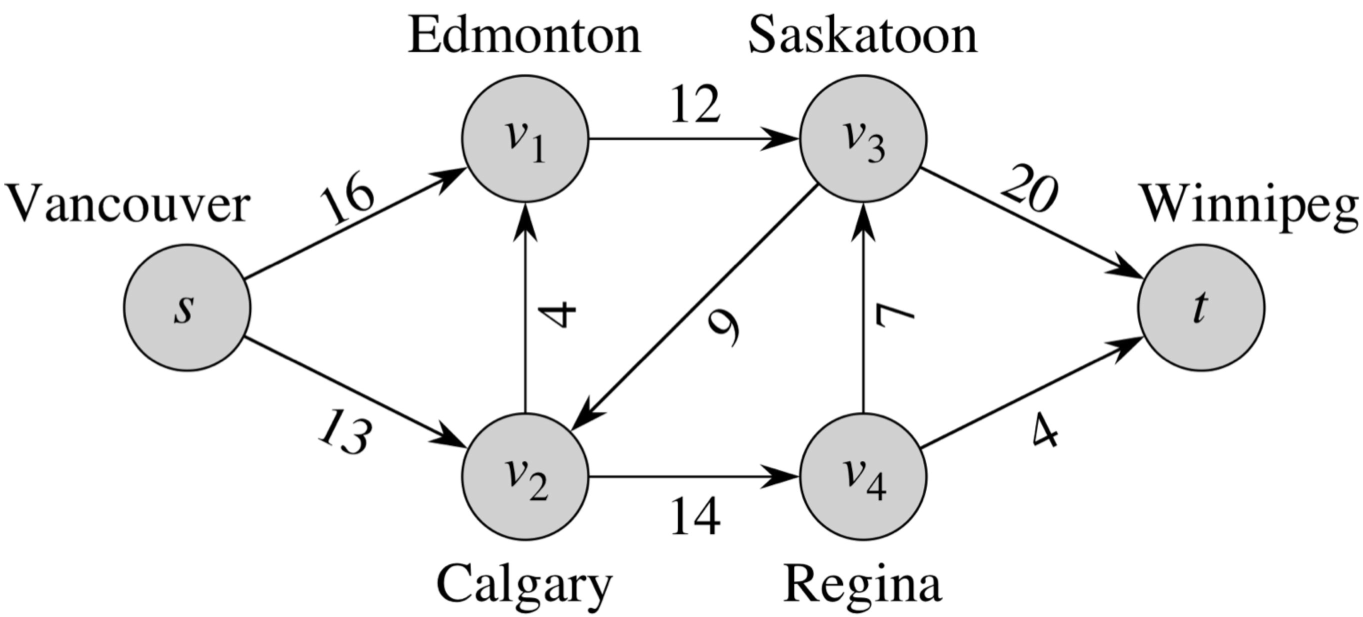
COMP 6651
Algorithm Design Techniques
Week 9

Max flow. Graph-cut.

Ack: Some slides from Fredo Durand MIT, Daniel Heilper from Haifa University and the web

Theory

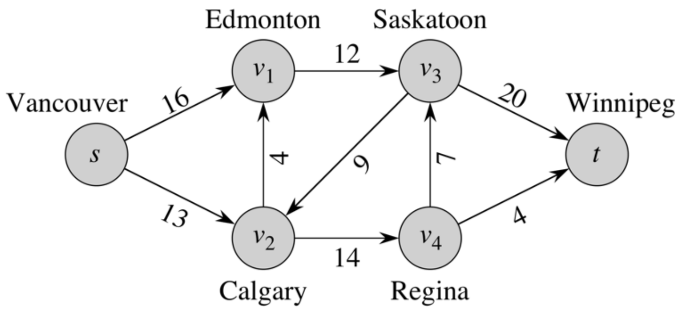
- Max flow problem



(a)

Theory

- Max flow problem



(a)

Capacity constraint: For all $u, v \in V$, we require $0 \le f(u,v) \le c(u,v)$.

Flow conservation: For all $u \in V - \{s,t\}$, we require

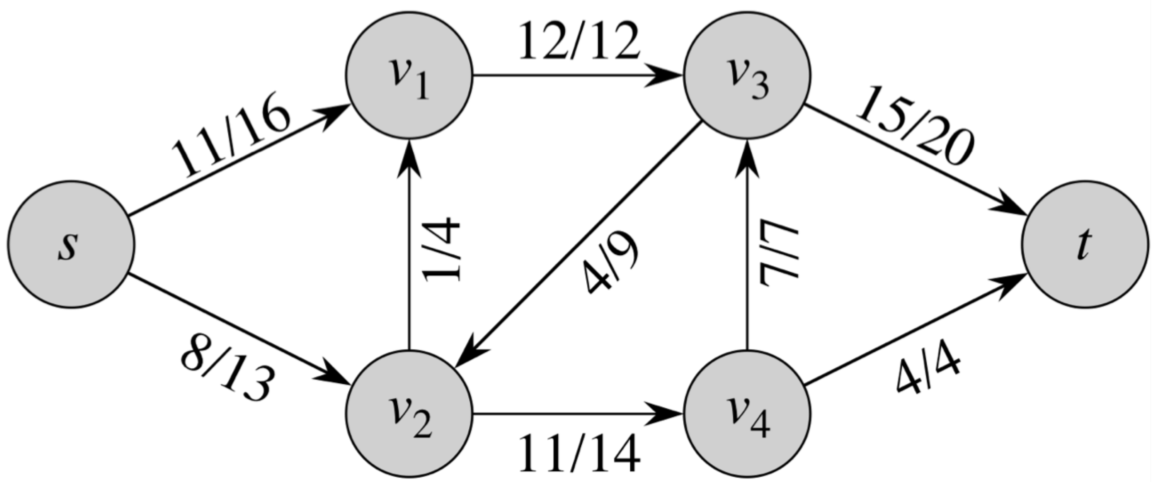
$$\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v) .$$

When $(u,v) \notin E$, there can be no flow from u to v, and $f(u,v) = 0$.

$$|f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) ,$$

Theory

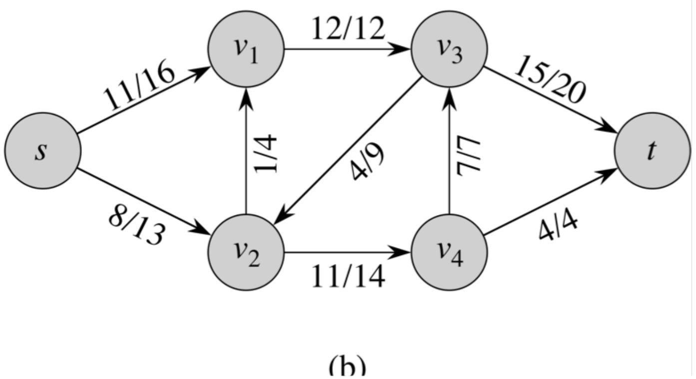
- Max flow problem



(b)

Theory

- Max flow problem
- Ford-Fulkerson algorithm



Capacity constraint: For all $u, v \in V$, we require $0 \leq f(u, v) \leq c(u, v)$.

Flow conservation: For all $u \in V - \{s, t\}$, we require

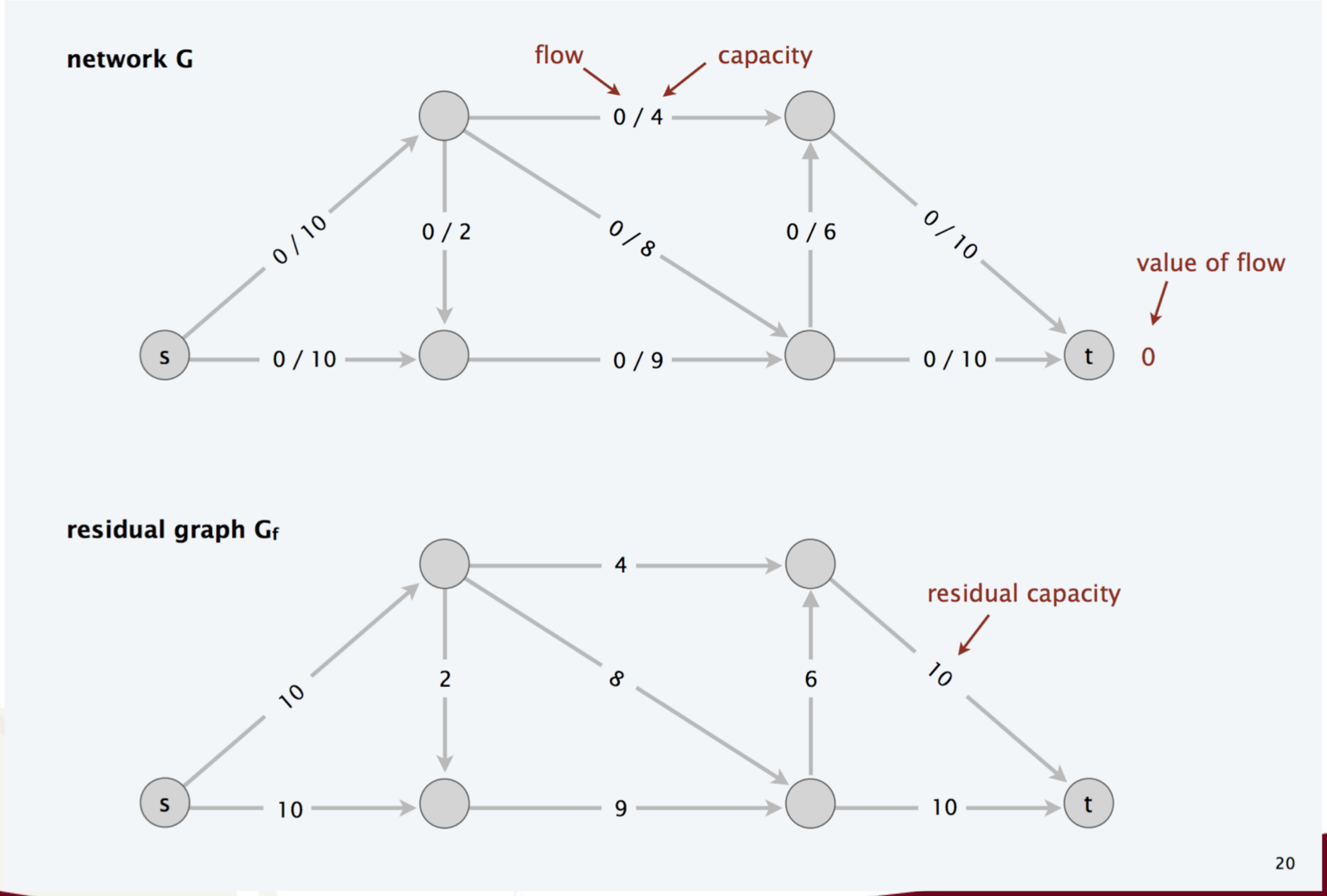
$$\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) .$$

When $(u, v) \notin E$, there can be no flow from u to v, and $f(u, v) = 0$.

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s) ,$$

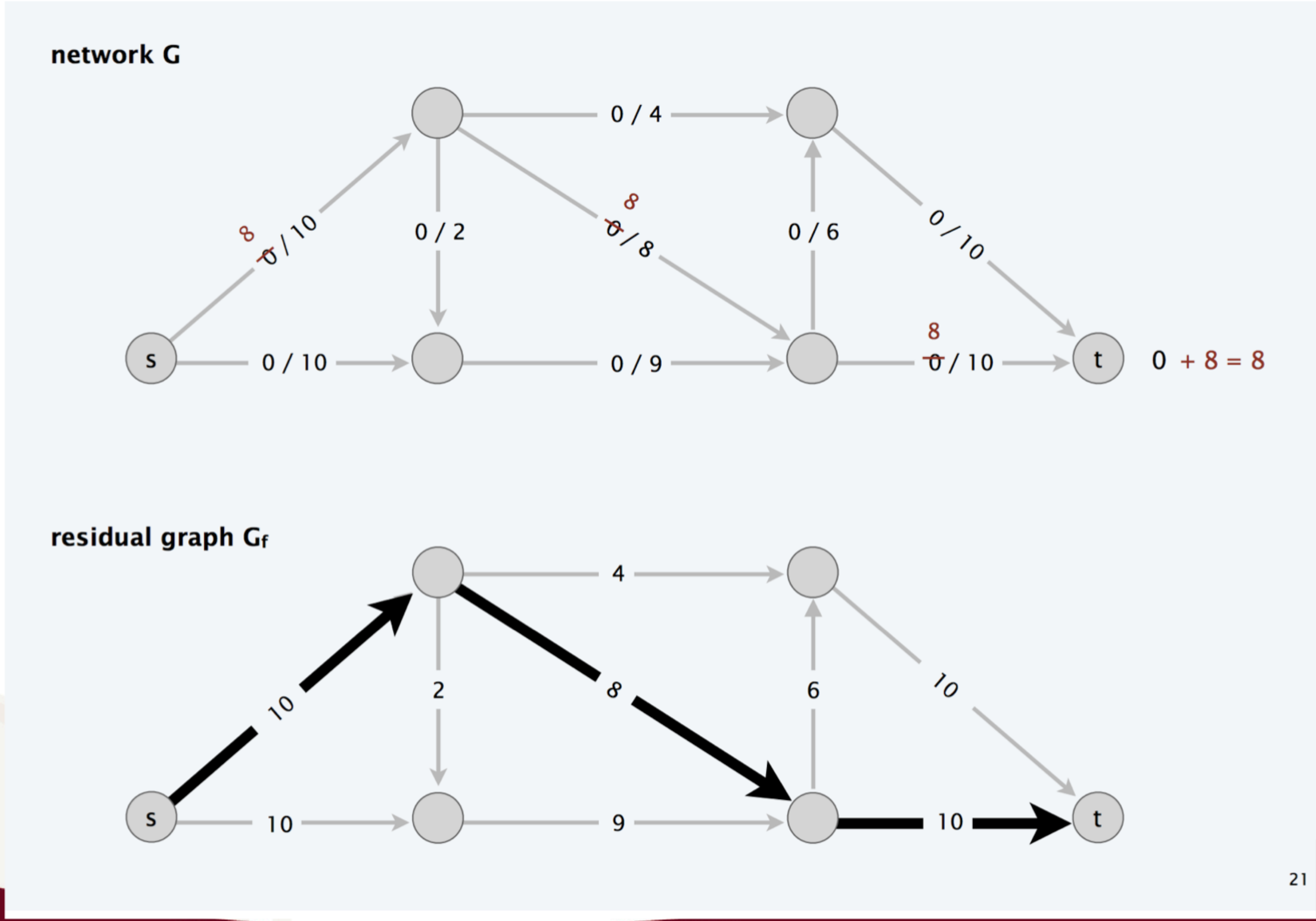
Theory

- Max flow problem - example



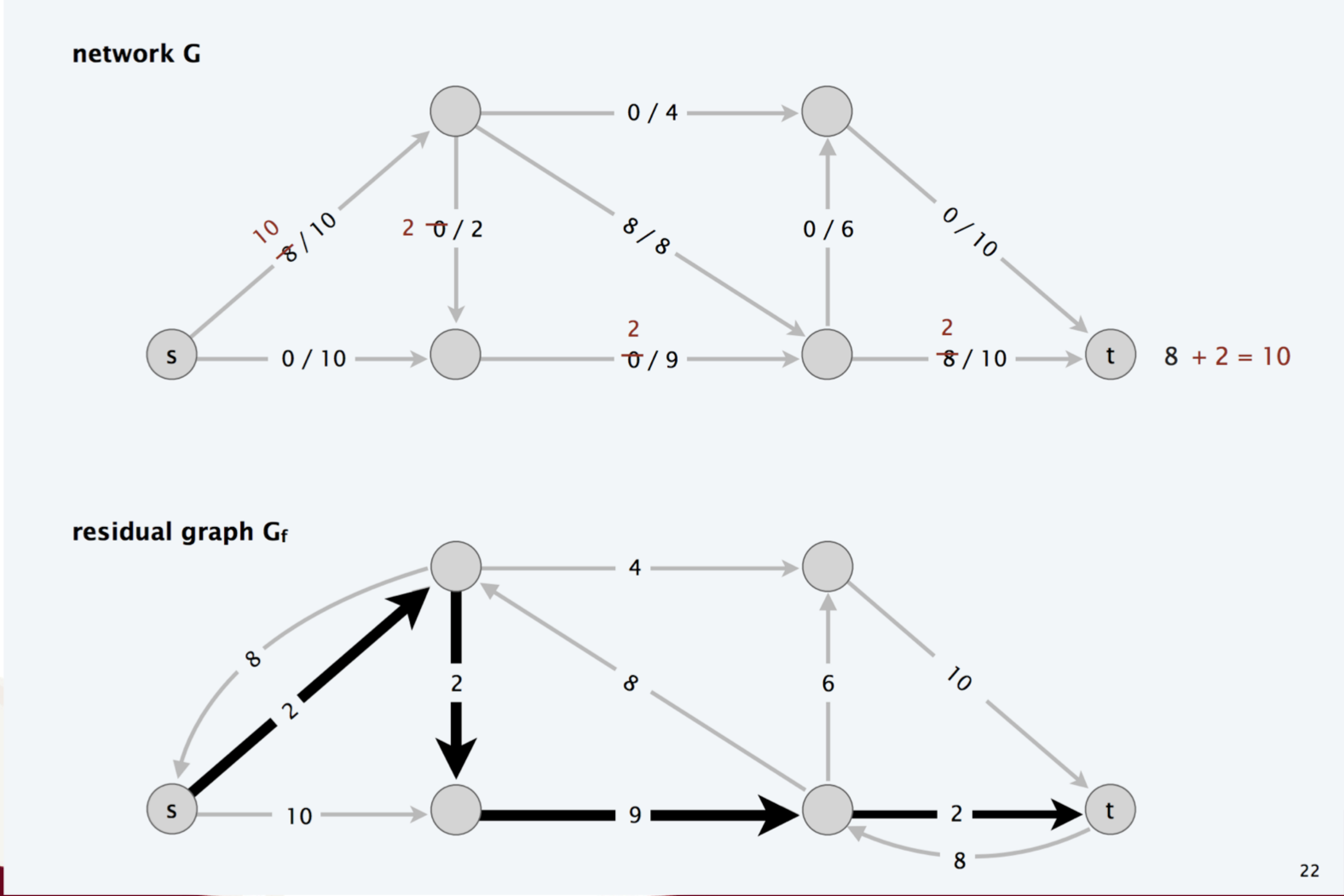
Theory

- Max flow problem - example



Theory

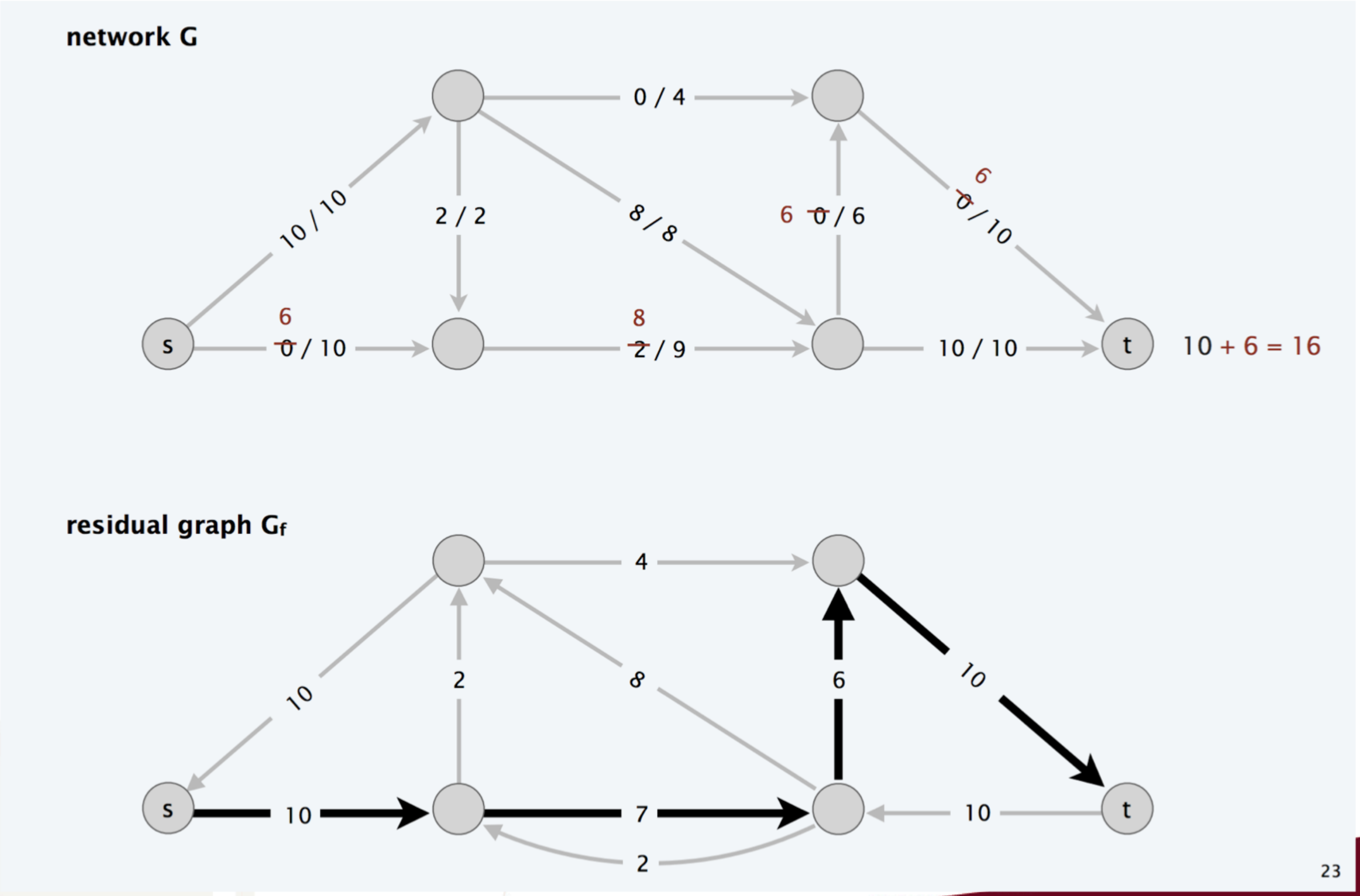
- Max flow problem - example



UNIVERSITÉ Concordia UNIVERSITY

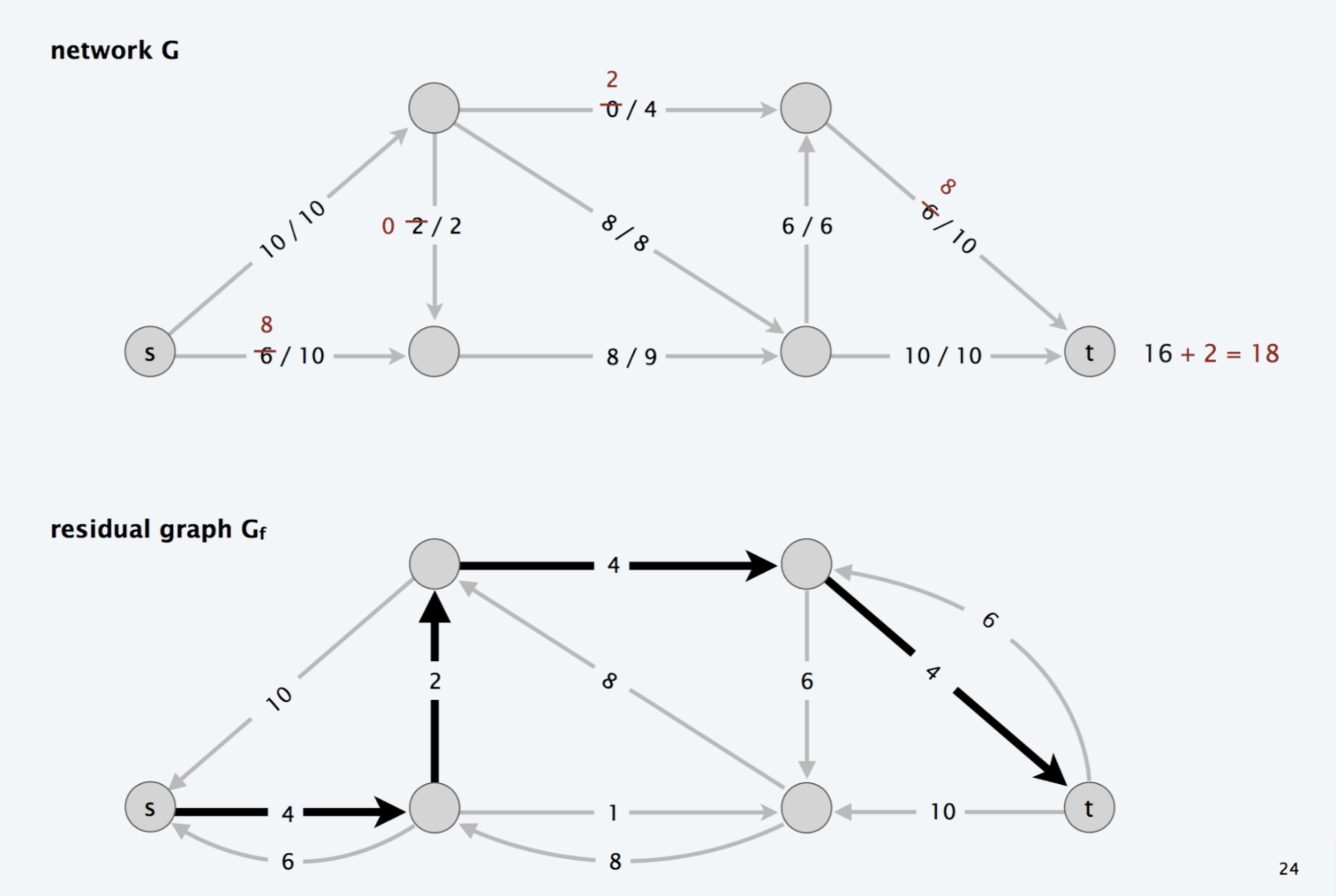
Theory

- Max flow problem - example



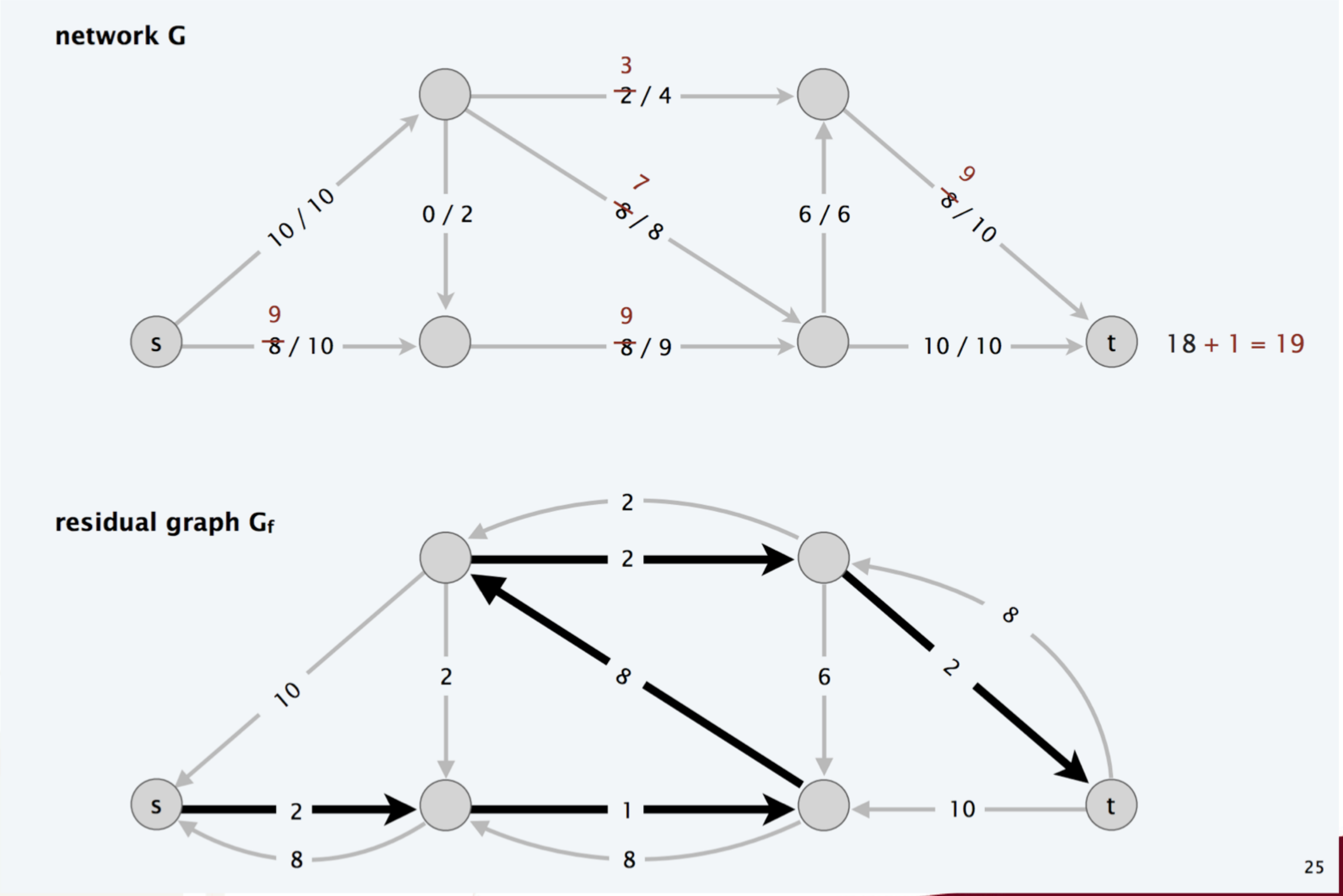
Theory

- Max flow problem - example



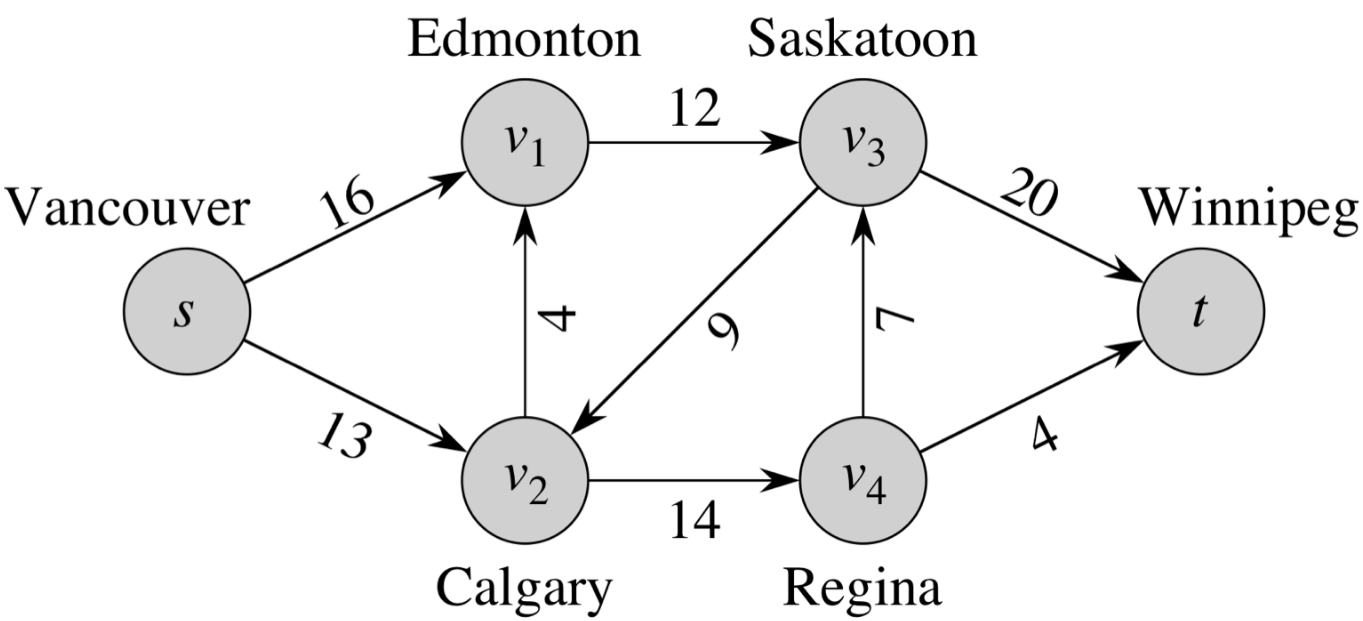
Theory

- Max flow problem - example



Theory

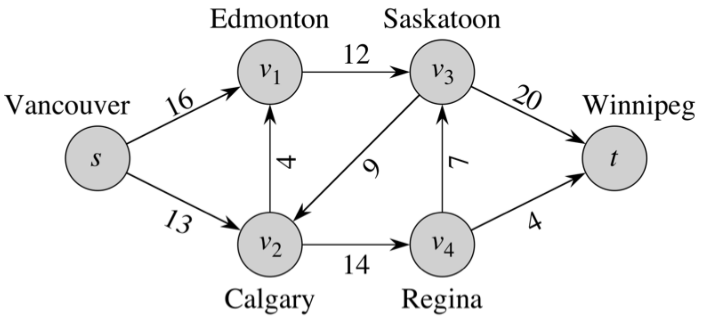
- Min Graph-cut ←→ Max flow problem



(a)

Theory

- Min Graph-cut ←→ Max flow problem
- Remove edges that partition the graph into two disjoint components

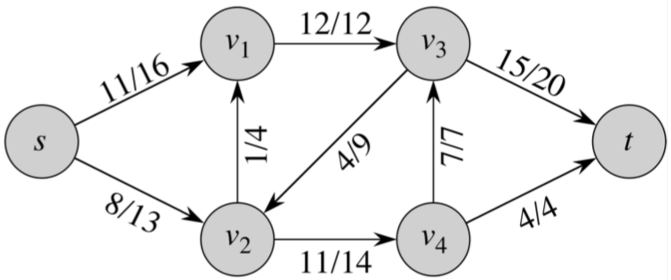


(a)

$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v)$$

Theory

- Min Graph-cut ←→ Max flow problem
- What is the relation?



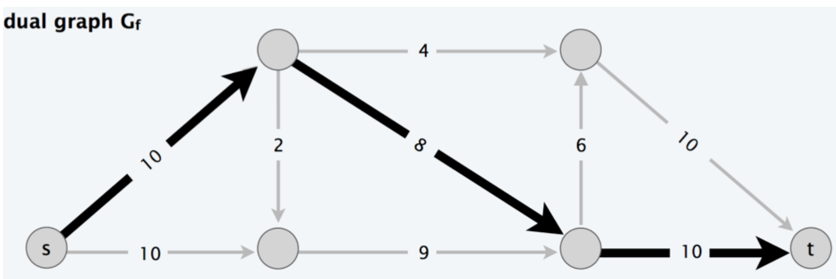
(b)

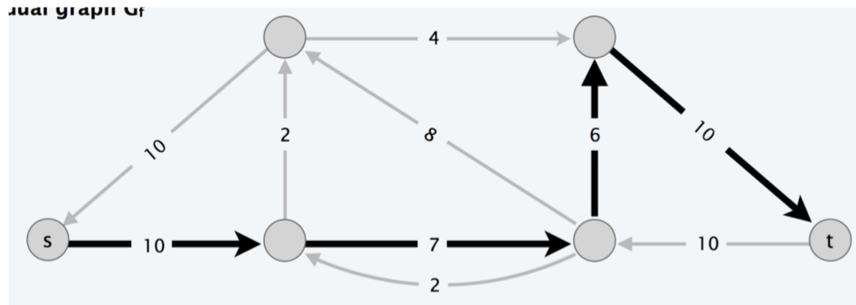
$$c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v)$$

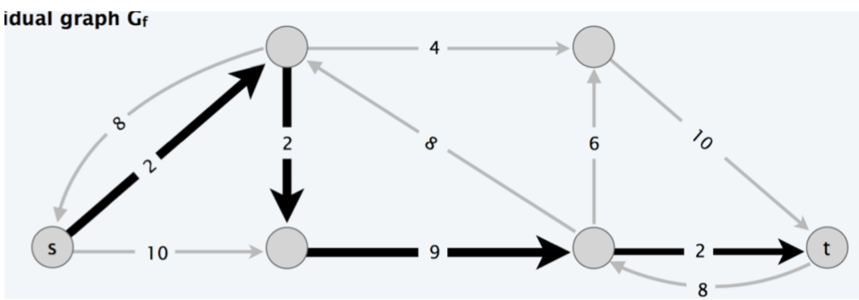
- Cost of the min cut equals min flow

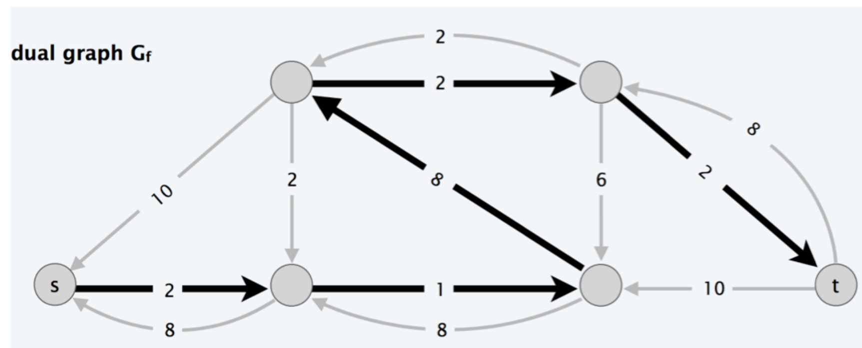
Theory

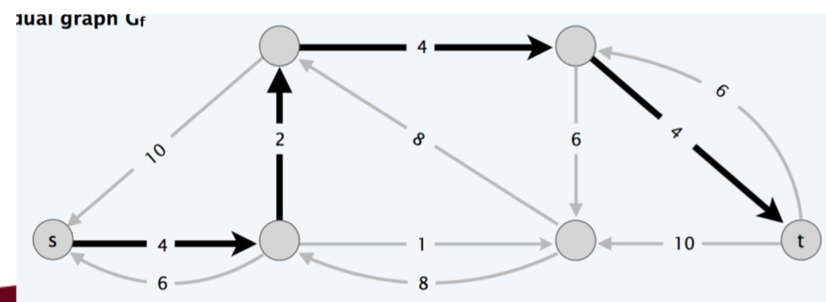
- How to get the min-cut
- Keep all residual graphs





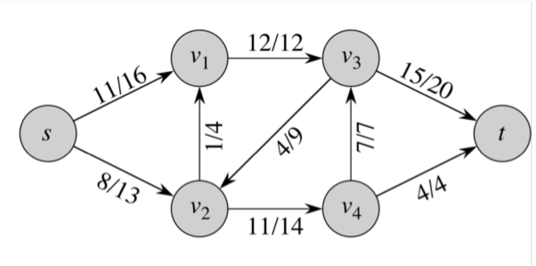






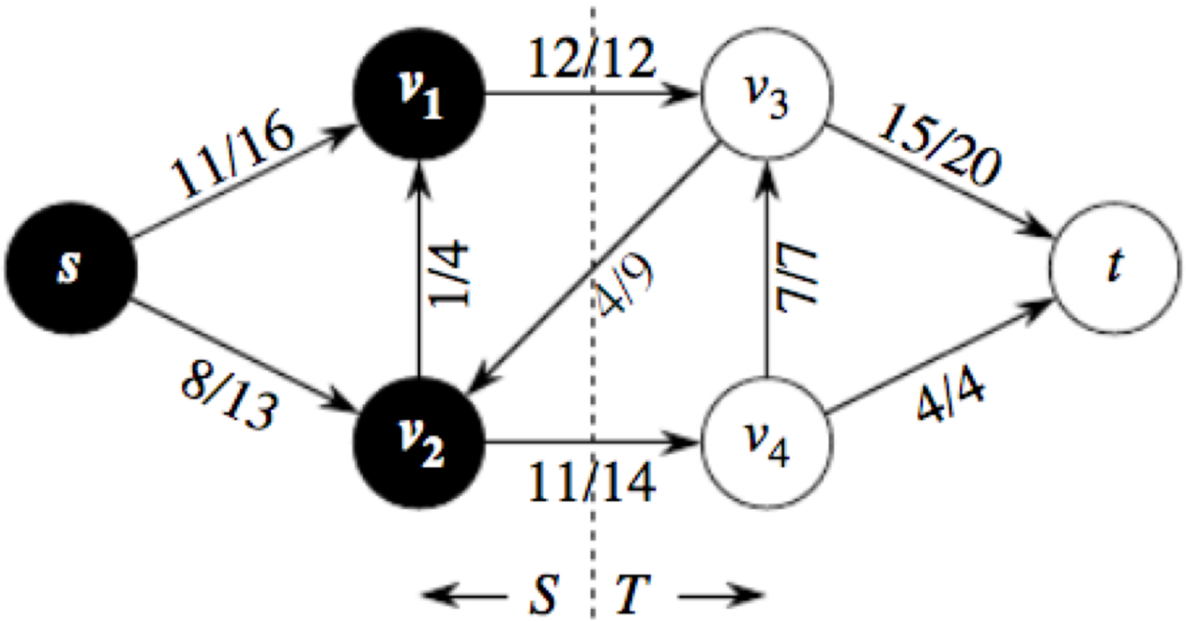
- Remove 1 edge from each
- Minimum sum
- Dynamic programming

Theory



s
v1
v2
v3
v4
t
11/16
8/13
12/12
1/4
4/9
7/7
11/14
15/20
4/4

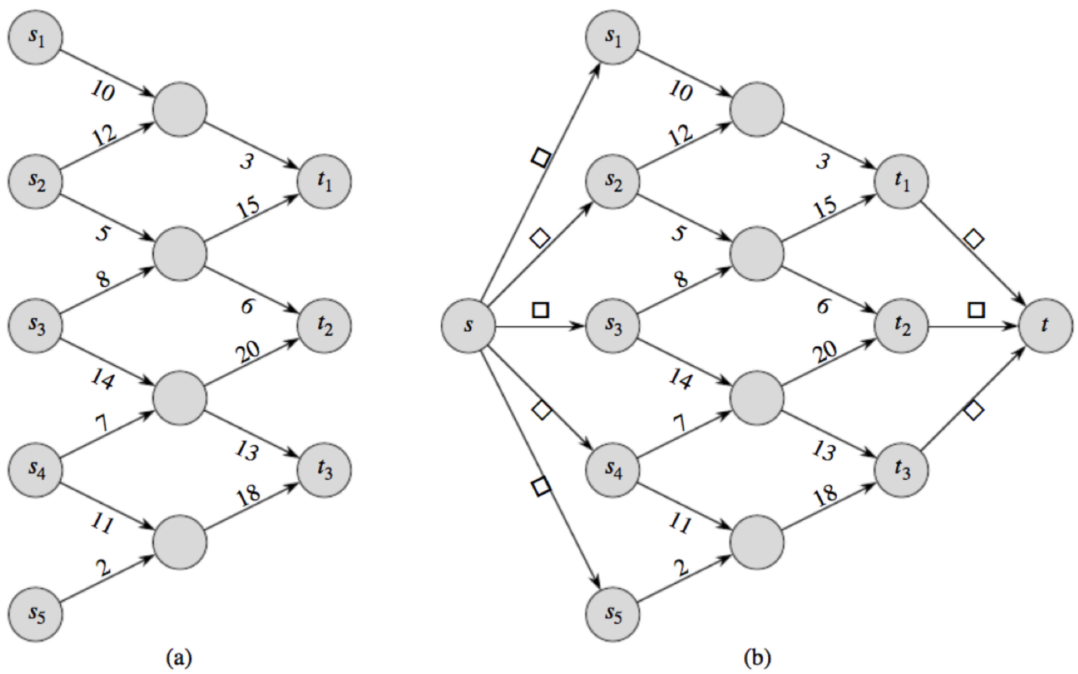
(b)



s
v1
v2
v3
v4
t
11/16
8/13
12/12
1/4
4/9
7/7
11/14
15/20
4/4
S
T

Theory

- Multiple sources and sinks



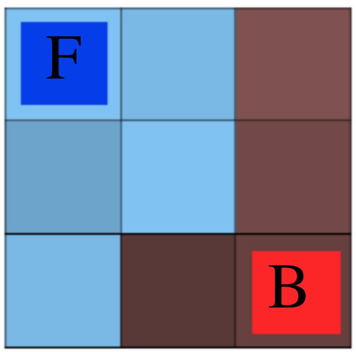
Theory

- Complexity
  - Planar graphs
  - Planar graphs on grid
- Undirected graph
  - Very similar
  - A bit simpler

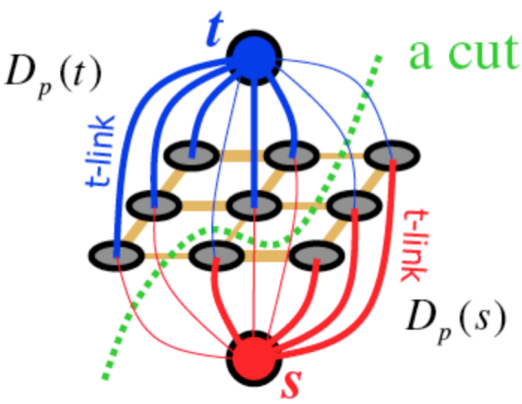
Image Segmentation using graph Cuts

How we apply to image segmentation?

Image forms a canonical graph



F	F	B
F	F	B
F	B	B



Reducing to finding min cut

BUT

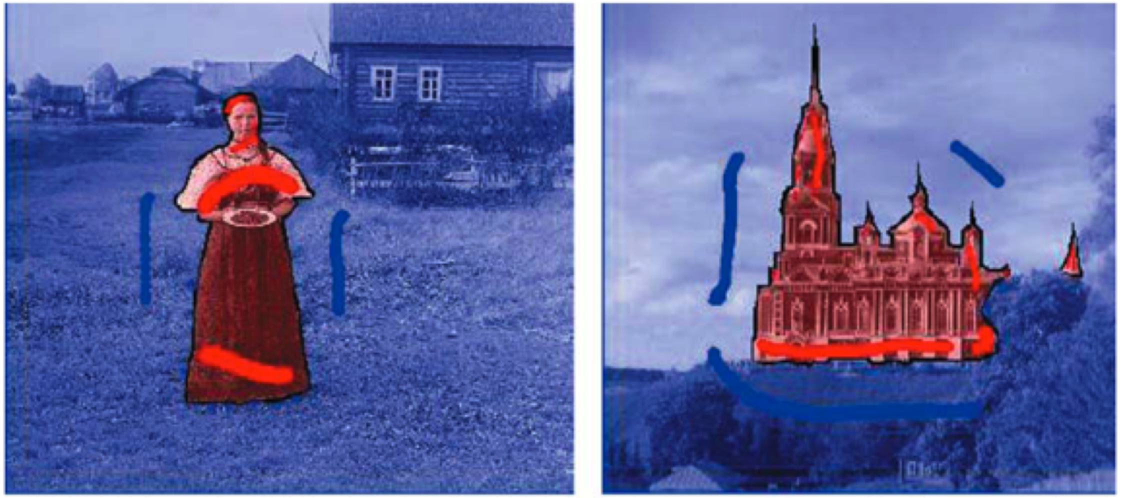
Problem is defining the weights/capacities

Who are s and t?

Image Segmentation using graph Cuts

Need an initial rough segmentation

1. Manual
2. Other heuristics
   1. Blurriness
   2. Depth

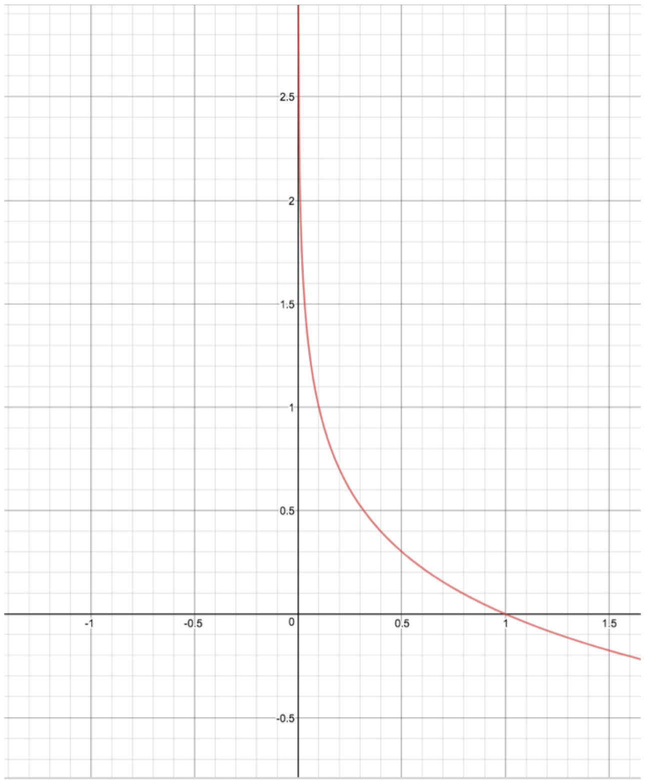


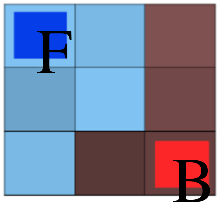
(a) A woman from a village (b) A church in Mozhaisk (near Moscow)

Image Segmentation using graph Cuts

How do we assign weights

1. Inf to edges from/to s and t
2. Use common split in computer vision: data + smoothness (regularization)
3. Compute probabilities that
   1. pixels are in different groups (regularization)
   2. Pixel belongs to F or G (data term) How?
      1. Initial clustering
      2. Blurriness
      3. Average color
4. Convert to energy (-log(p))





F F B
F F B
F B B

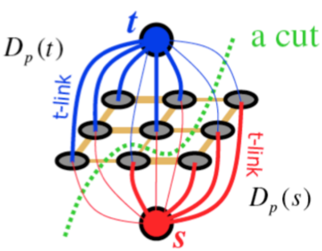
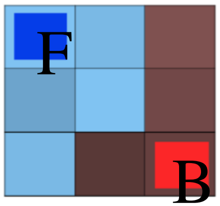


Image Segmentation using graph Cuts

- What is better graph-cut clustering?
  - Graph-cut:
    - Both unary and neighboring probability (Random Markov Fields)
  - Clustering:
    - Allows more clusters
  - Hidden Markov Model
    - Generalizes to multiple clusters
    - Not using graph-cut
- It is not the only approach
- Graph cut used a lot in classification
- Graphical models used a lot in visual computing



F	F	B
F	F	B
F	B	B