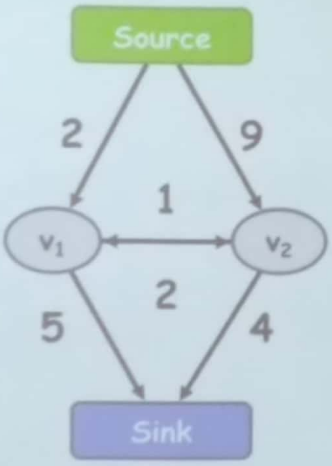
CSE578: Computer Vision
Spring 2019

Image Segmentation using Graph Cut

Anoop M. Namboodiri and Avinash Sharma
Center for Visual Information Technology
IIIT Hyderabad, INDIA

The st-Mincut Problem



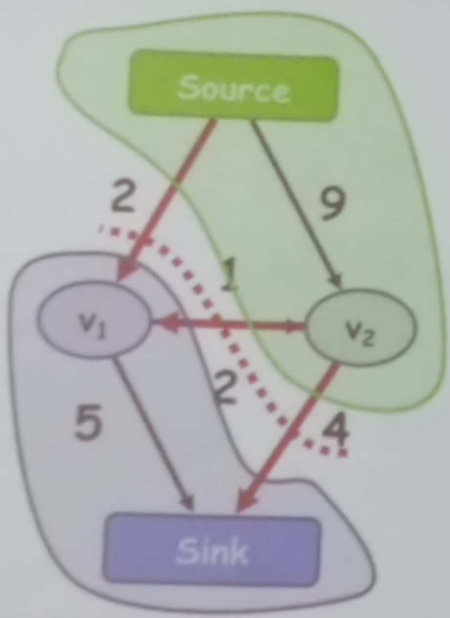
Graph (V, E, C)

Vertices $V = \{v_1, v_2 \dots v_n\}$

Edges $E = \{(v_1, v_2) \dots\}$

Costs $C = \{c_{(1, 2)} \dots\}$

The st-Mincut Problem



$2 + 1 + 4 = 7$

What is an st-cut?

An st-cut (S,T) divides the nodes between source and sink.

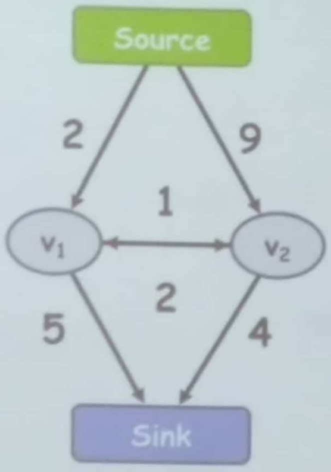
What is the cost of an st-cut?

Sum of cost of all edges going from S to T

What is the st-mincut?

st-cut with the minimum cost

How to compute the st-Mincut



Solve the dual **maximum flow** problem

Compute the maximum flow between Source and Sink

Constraints

Edges: Flow < Capacity

Nodes: Flow in = Flow out

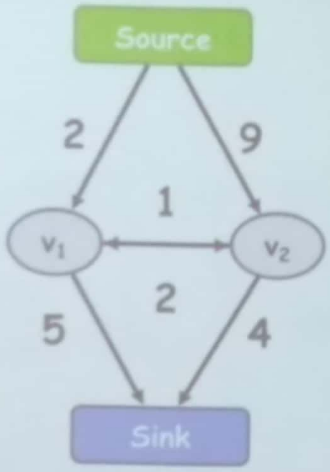
Min-cut/Maxflow Theorem

In every network, the maximum flow equals the cost of the st-mincut

Dantzig, Dinitz, Edmonds & Karp, Ford & Fulkerson, Goldberg & Tarjan, Karzanov, …

Maxflow Algorithm

Flow = 0



Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity

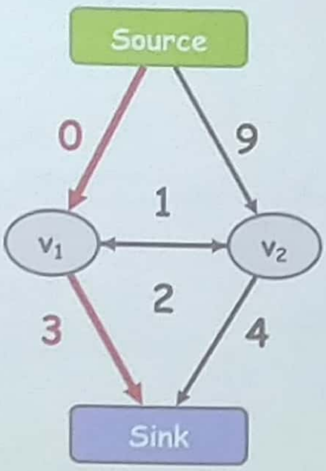
2. Push maximum possible flow through this path

3. Repeat until no path can be found

Algorithms assume non-negative capacity

Maxflow Algorithm

Flow = 2



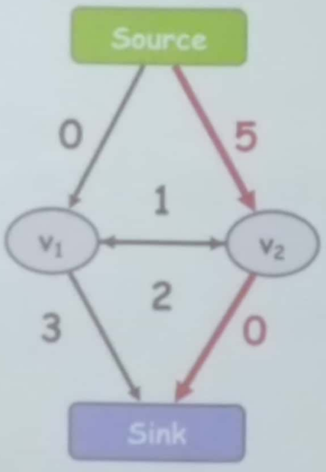
Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Maxflow Algorithm

Flow = 2+4



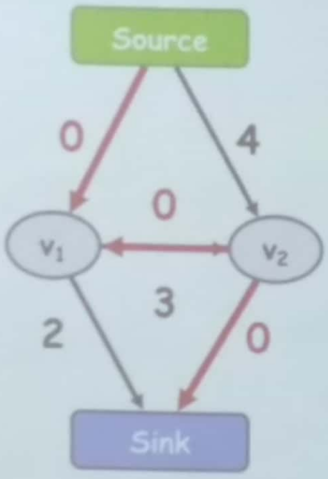
Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Maxflow Algorithm

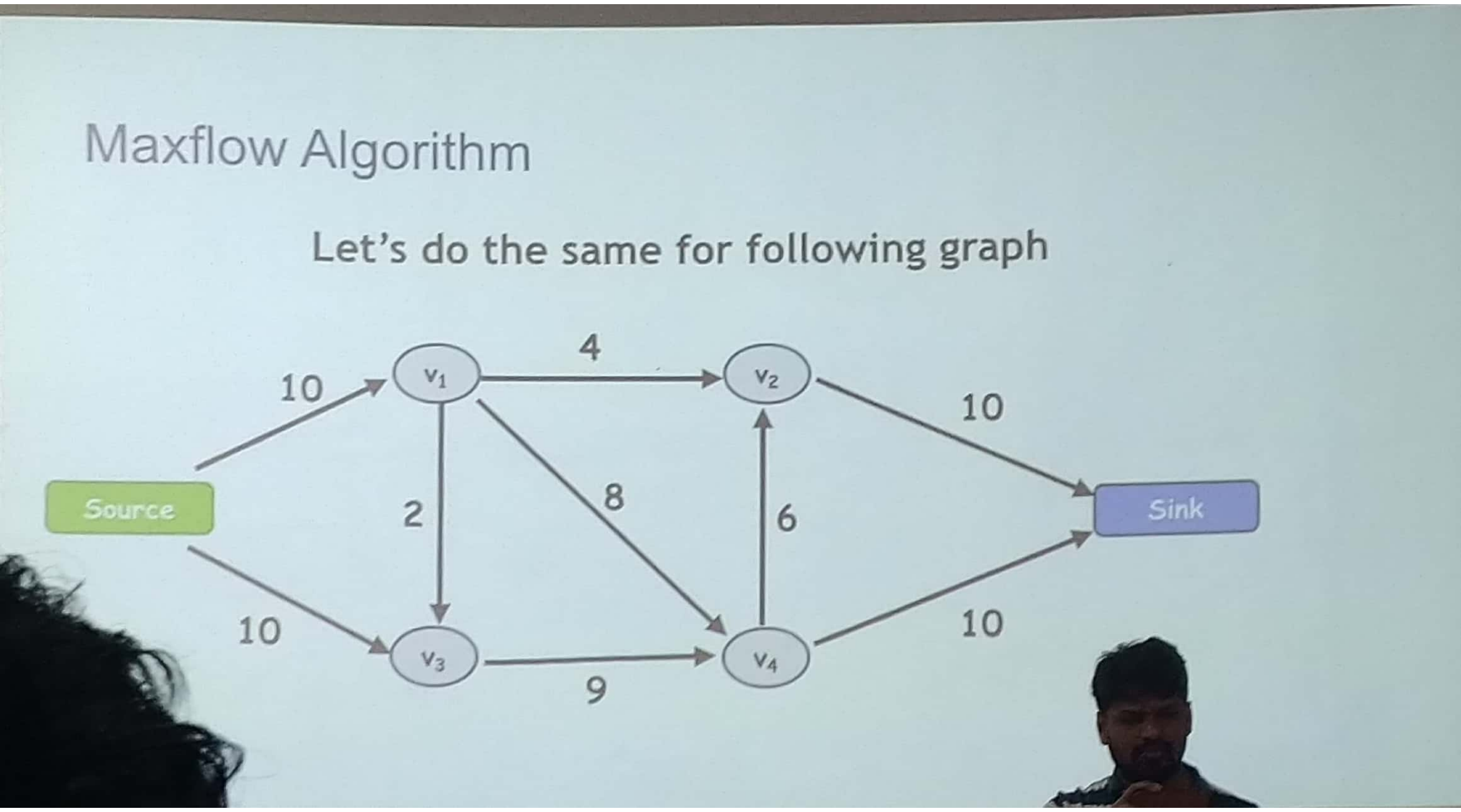
Flow = 7



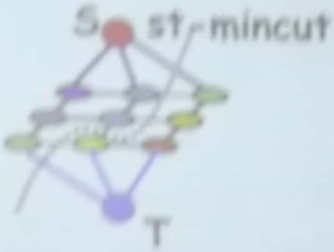
Finding MinCut Edges

1. Run Augementing path based algorithm and consider the final residual graph.
2. Find the set of vertices that are reachable from source in the residual graph.
3. All edges which are from a reachable vertex to non-reachable vertex are minimum cut edges.

Algorithms assume non-negative capacity


Maxflow Algorithm
Let's do the same for following graph
Source
Sink
v_1
v_2
v_3
v_4
10
10
4
2
8
6
9
10
10

St-mincut and Energy Minimization



Minimizing a Qudratic Pseudoboolean function E(x)

Functions of boolean variables

$$E: \{0,1\}^n \rightarrow \mathbf{R}$$

$$E(x) = \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i (1 - x_j)$$

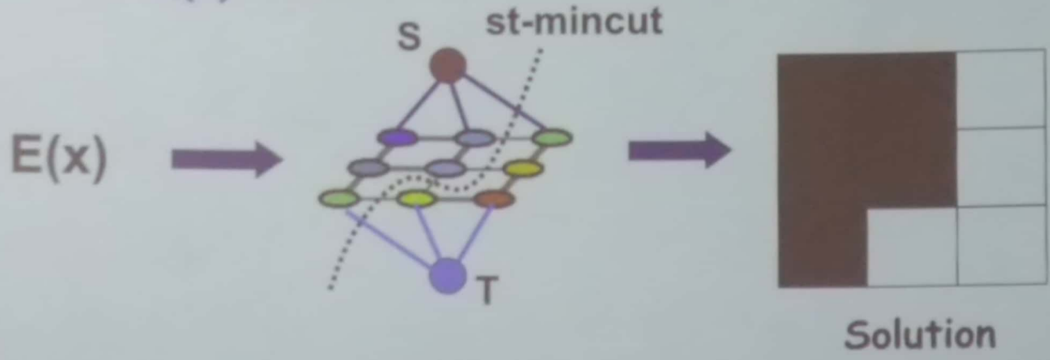
$$c_{ij} \geq 0$$

Polynomial time st-mincut algorithms require non-negative edge weights

How does it Work ?

Construct a graph such that:

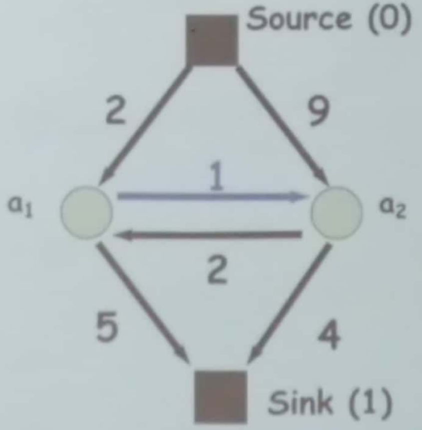
1. Any st-cut corresponds to an assignment of x
2. The cost of the cut is equal to the energy of x : E(x)



Slide Courtesy: P

Graph Construction and Energy Function

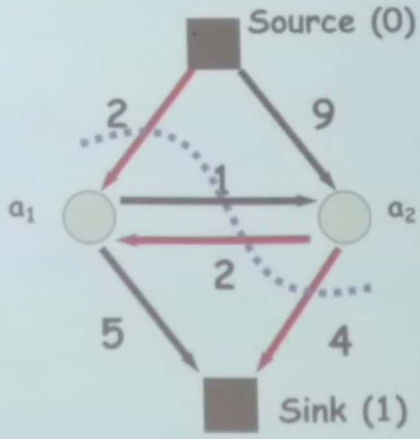
$$E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



Slide Courtesy: Pawan, Pushmeet

Graph Construction and Energy Function

$$E(a_1,a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1a_2$$



st-mincut cost = 8

$a_1 = 1 \quad a_2 = 0$

$E(1,0) = 8$

Slide Courtesy: Pawa

Energy Function Reparameterization

Two functions E_1 and E_2 are reparameterizations if

$$E_1(x) = E_2(x) \quad \text{for all } x$$

For instance:

$E_1(a_1) = 1 + 2a_1 + 3\bar{a}_1$

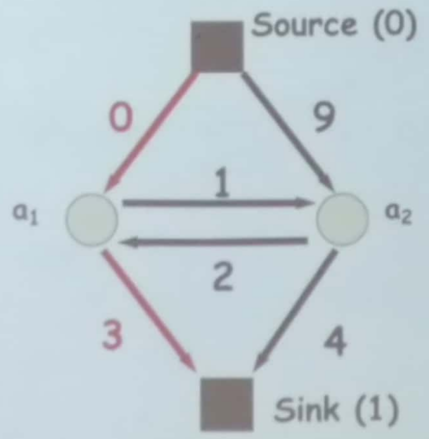
$E_2(a_1) = 3 + \bar{a}_1$

a_1	$\bar{a}_1$	$1 + 2a_1 + 3\bar{a}_1$	$3 + \bar{a}_1$
0	1	4	4
1	0	3	3

Slide Courtesy: Pawan, Pushmeet

Flow and Reparameterization

$$E(a_1,a_2) = 2 + 3\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$

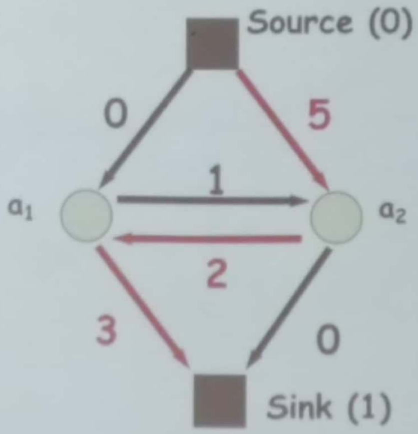


$$\begin{aligned} &2a_1 + 5\bar{a}_1 \\ &= 2(a_1+\bar{a}_1) + 3\bar{a}_1 \\ &= 2 + 3\bar{a}_1 \end{aligned}$$

Slide Courtesy: Pawan, Pushmeet

Flow and Reparameterization

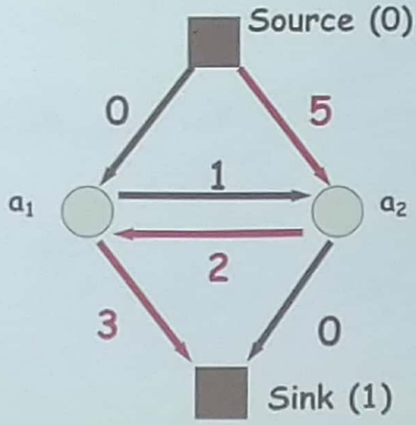
$$E(a_1, a_2) = 6 + 3\bar{a}_1 + 5a_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



Slide Courtesy: Pawan, Pushmeet

Flow and Reparameterization

$$E(a_1,a_2) = 6 + 3\bar{a}_1 + 5a_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$$



$$3\bar{a}_1 + 5a_2 + 2a_1\bar{a}_2$$
$$= 2(\bar{a}_1 + a_2 + a_1\bar{a}_2) + \bar{a}_1 + 3a_2$$
$$= 2(1 + \bar{a}_1 a_2) + \bar{a}_1 + 3a_2$$

$F1 = \bar{a}_1 + a_2 + a_1\bar{a}_2$

$F2 = 1 + \bar{a}_1 a_2$

a_1	a_2	F1	F2
0	0	1	1
0	1	2	2
1	0	1	1
1	1	1	1

Slide Courtesy: Pawan, Pushmeet

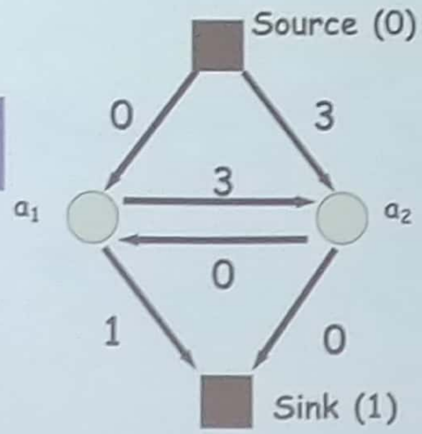
Flow and Reparameterization

$$E(a_1, a_2) = \boxed{8} + \boxed{\bar{a}_1 + 3a_2 + 3\bar{a}_1 a_2} \longrightarrow$$

Residual Graph
(positive coefficients)

Total Flow

bound on the optimal solution



Slide Courtesy: Pawan, Pushmeet

Image Segmentation Scenario

$$E(x) = \sum_{i} c_i\, x_i + \sum_{i,j} c_{ij}\, x_i(1 - x_j)$$

$E: \{0,1\}^n \rightarrow R$

$0 \rightarrow fg$

$1 \rightarrow bg$

n = number of pixels

Image (D)

x_i x_j

Slide Courtesy: Pawan, Pushmeet

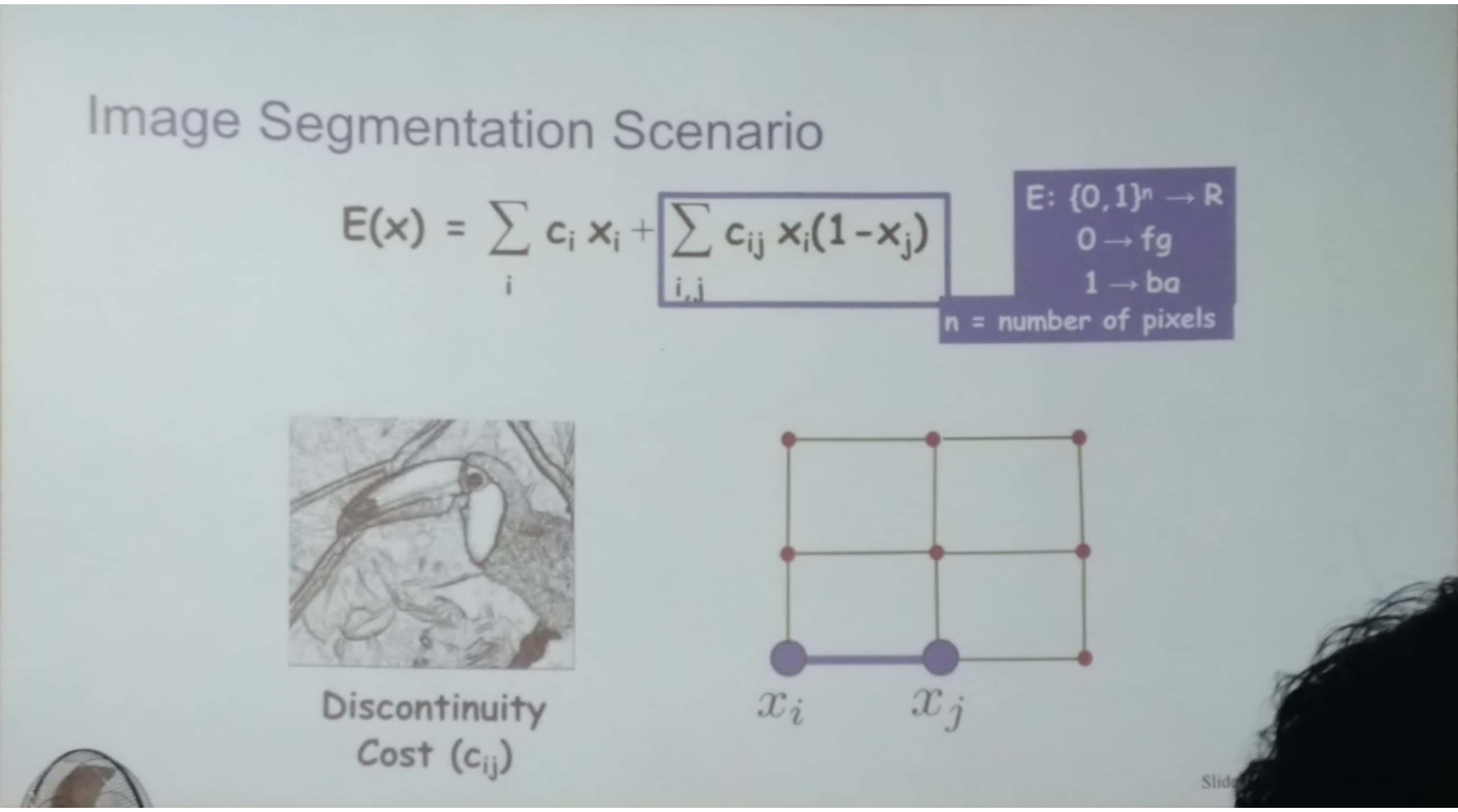


Image Segmentation Scenario
$E(x) = \sum_i c_i x_i + \boxed{\sum_{i,j} c_{ij} x_i(1-x_j)}$
$E: \{0,1\}^n \rightarrow R$
$0 \rightarrow fg$
$1 \rightarrow bg$
n = number of pixels
Discontinuity Cost (c_{ij})
x_i
x_j

Image Segmentation Scenario

$$E(x) = \sum_i c_i x_i + \sum_{i,j} c_{ij} x_i(1-x_j)$$

$E: \{0,1\}^n \rightarrow R$
$0 \rightarrow fg$
$1 \rightarrow bg$

Global Minimum (x^*)

$$x^* = \arg\min_x E(x)$$

How to minimize $E(x)$?

Image Segmentation Scenario:Code Level Walkthrough

```
Graph *g;

For all pixels p

        /* Add a node to the graph */
        nodeID(p) = g->add_node();

        /* Set cost of terminal edges */
        set_weights(nodeID(p), fgCost(p), bgCost(p));

end

for all adjacent pixels p,q
        add_weights(nodeID(p), nodeID(q),  cost);
end

g->compute_maxflow();

label_p = g->is_connected_to_source(nodeID(p));

// is the label of pixel p (0 or 1)
```

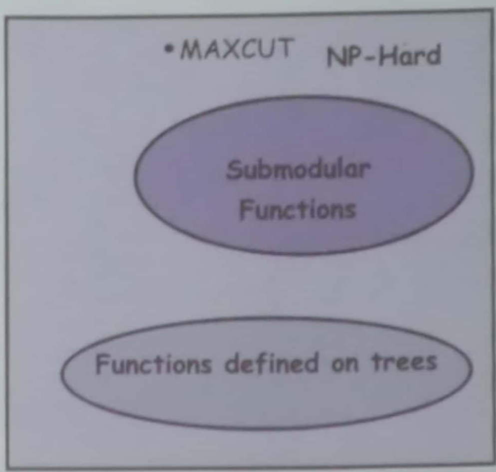
Source (0)

Sink (1)

Slide Courtesy: Pawan, Pushme

Minimizing Energy Functions

- General Energy Functions
  - NP-hard to minimize
  - Only approximate minimization possible
- Easy energy functions
  - Solvable in polynomial time
  - Submodular ~ $O(n^6)$



Space of Function Minimization Problems

Slide Courtesy: Pawan, Pushmeet