Today’s topics
• Image Segmentation
• Strip Mining
• Reading: 7.5, 7.6, 7.10-7.12

Minimum Cut Applications
• Image Segmentation
• Open Pit Mining / Task Selection Problem
• Reduction to Min Cut problem

S, T is a cut if S, T is a partition of the vertices with s in S and t in T.
The capacity of an S, T cut is the sum of the capacities of all edges going from S to T.

Image Segmentation

Separate Lion from Savana
Image analysis

- $a_i$: value of assigning pixel $i$ to the foreground
- $b_i$: value of assigning pixel $i$ to the background
- $p_{ij}$: penalty for assigning $i$ to the foreground, $j$ to the background or vice versa
- $A$: foreground, $B$: background
- $Q(A, B) = \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, i \in A, j \in B} p_{ij}$

Pixel graph to flow graph

MinCut Construction

S, T is a cut if $S, T$ is a partition of the vertices with $s$ in $S$ and $t$ in $T$

The capacity of an $S, T$ cut is the sum of the capacities of all edges going from $S$ to $T$

Application of Min-cut

- Open Pit Mining Problem
- Task Selection Problem
- Reduction to Min Cut problem

Open Pit Mining

- Each unit of earth has a profit (possibly negative)
- Getting to the ore below the surface requires removing the dirt above
- Test drilling gives reasonable estimates of costs
- Plan an optimal mining operation
Generalization

- Precedence graph $G=(V,E)$
- Each $v$ in $V$ has a profit $p(v)$
- A set $F$ is feasible if when $w$ in $F$, and $(v,w)$ in $E$, then $v$ in $F$.
- Find a feasible set to maximize the profit

Min cut algorithm for profit maximization

- Construct a flow graph where the minimum cut identifies a feasible set that maximizes profit

Precedence graph construction

- Precedence graph $G=(V,E)$
- Each edge in $E$ has infinite capacity
- Add vertices $s$, $t$
- Each vertex in $V$ is attached to $s$ and $t$ with finite capacity edges

Find a finite value cut with at least two vertices on each side of the cut
The sink side of a finite cut is a feasible set

- No edges permitted from S to T
- If a vertex is in T, all of its ancestors are in T

Setting the costs

- If \( p(v) > 0 \),
  - \( \text{cap}(v,t) = p(v) \)
  - \( \text{cap}(s,v) = 0 \)
- If \( p(v) < 0 \)
  - \( \text{cap}(s,v) = -p(v) \)
  - \( \text{cap}(v,t) = 0 \)
- If \( p(v) = 0 \)
  - \( \text{cap}(s,v) = 0 \)
  - \( \text{cap}(v,t) = 0 \)

Minimum cut gives optimal solution

Why?

Computing the Profit

- \( \text{Cost}(W) = \sum_{w \in W; p(w) < 0} p(w) \)
- \( \text{Benefit}(W) = \sum_{w \in W; p(w) > 0} p(w) \)
- \( \text{Profit}(W) = \text{Benefit}(W) - \text{Cost}(W) \)

- Maximum cost and benefit
  - \( C = \text{Cost}(V) \)
  - \( B = \text{Benefit}(V) \)

Express \( \text{Cap}(S,T) \) in terms of \( B, C, \text{Cost}(T), \text{Benefit}(T), \) and \( \text{Profit}(T) \)

\[
\text{Cap}(S,T) = \text{Cost}(T) + \text{Ben}(S) + \text{Cost}(T) + \text{Ben}(S) + \text{Ben}(T) - \text{Ben}(T) \\
= B + \text{Cost}(T) - \text{Ben}(T) = B - \text{Profit}(T)
\]