Application of Min Cut 1: Image Segmentation

- An image is represented as matrix of pixels. We'd like to segment it into foreground and background.

Application of Min Cut 1: Image Segmentation

- An image is represented as matrix of pixels. We'd like to segment it into foreground and background.
- Each pixel i has a *likelihood* a_i for being in the foreground and a likelihood b_i for being in the background.

Application of Min Cut 1: Image Segmentation

- An image is represented as matrix of pixels. We'd like to segment it into foreground and background.
- Each pixel i has a *likelihood* a_i for being in the foreground and a likelihood b_i for being in the background.
- Each pair of adjacent pixels (i, j) has a *separation penalty* p_{ij} for being separated.

Application of Min Cut 1: Image Segmentation

- An image is represented as matrix of pixels. We'd like to segment it into foreground and background.
- Each pixel i has a *likelihood* a_i for being in the foreground and a likelihood b_i for being in the background.
- Each pair of adjacent pixels (i, j) has a *separation penalty* p_{ij} for being separated.
- We can naturally model this as a graph, with nodes V representing the pixels, and an edge (i, j) exists if pixels i and j are adjacent.

Application of Min Cut 1: Image Segmentation

- An image is represented as matrix of pixels. We'd like to segment it into foreground and background.
- Each pixel i has a *likelihood* a_i for being in the foreground and a likelihood b_i for being in the background.
- Each pair of adjacent pixels (i, j) has a *separation penalty* p_{ij} for being separated.
- We can naturally model this as a graph, with nodes V representing the pixels, and an edge (i, j) exists if pixels i and j are adjacent.
- Question: partition V into two subsets A and B, to maximize

$$\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Application of Min Cut 1: Image Segmentation

- An image is represented as matrix of pixels. We'd like to segment it into foreground and background.
- Each pixel i has a *likelihood* a_i for being in the foreground and a likelihood b_i for being in the background.
- Each pair of adjacent pixels (i, j) has a *separation penalty* p_{ij} for being separated.
- We can naturally model this as a graph, with nodes V representing the pixels, and an edge (i, j) exists if pixels i and j are adjacent.
- Question: partition V into two subsets A and B, to maximize

$$\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}| = 1} p_{ij}.$$

$$\max \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Idea: Turn the problem into a minimization problem.

$$\min - \sum_{i \in A} a_i - \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

$$\max \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Idea: Turn the problem into a minimization problem.

$$\min - \sum_{i \in A} a_i - \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Or

$$\min \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

$$\max \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Idea: Turn the problem into a minimization problem.

$$\min - \sum_{i \in A} a_i - \sum_{j \in B} b_j - \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Or

$$\min \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Construct a flow network (with added source and sink) so that the capacity of any s-t cut $(\{s\} \cup A, \{t\} \cup B)$ is exactly

$$\sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{(i,j) \in E, |A \cap \{i,j\}|=1} p_{ij}.$$

Application of Min Cut 2: Project Selection

- We are given an acyclic directed graph $G = (V, E)$, each node representing a project.

Application of Min Cut 2: Project Selection

- We are given an acyclic directed graph $G = (V, E)$, each node representing a project.
- Each project i, when selected, yields a profit of $p_i \in \mathbb{R}$ (note that p_i can be negative), but can be selected only if all its *prerequisites* are selected as well. A project j is i's prerequisite if $(i, j) \in E$.

Application of Min Cut 2: Project Selection

- We are given an acyclic directed graph $G = (V, E)$, each node representing a project.
- Each project i, when selected, yields a profit of $p_i \in \mathbb{R}$ (note that p_i can be negative), but can be selected only if all its *prerequisites* are selected as well. A project j is i's prerequisite if $(i, j) \in E$.
- Question: select a subset A of projects to maximize

$$\sum_{i \in A} p_i,$$

subject to the constraint that any project $i \in A$ must have all its prerequisites in A as well.

Application of Min Cut 2: Project Selection

- We are given an acyclic directed graph $G = (V, E)$, each node representing a project.
- Each project i, when selected, yields a profit of $p_i \in \mathbb{R}$ (note that p_i can be negative), but can be selected only if all its *prerequisites* are selected as well. A project j is i's prerequisite if $(i, j) \in E$.
- Question: select a subset A of projects to maximize

  $$\sum_{i \in A} p_i,$$

  subject to the constraint that any project $i \in A$ must have all its prerequisites in A as well.

Note: if every p_i is nonnegative, the problem becomes trivial. Negative profits are essential.

Solving the project selection problem

- First attempt: partition nodes into "selected" and "unselected", and design the flow network so that the cut's capacity is the total profit.
  - How do we deal with negative profits?

Solving the project selection problem

- First attempt: partition nodes into "selected" and "unselected", and design the flow network so that the cut's capacity is the total profit.
  - How do we deal with negative profits?
  - How about the prerequisite constraints?

Solving the project selection problem

- First attempt: partition nodes into "selected" and "unselected", and design the flow network so that the cut's capacity is the total profit.
  - How do we deal with negative profits?
  - How about the prerequisite constraints?
- Second idea: make it hugely costly for any prerequisite constraint to be broken, so that cuts that violate any cannot have minimum capacity.

Solving the project selection problem

- First attempt: partition nodes into "selected" and "unselected", and design the flow network so that the cut's capacity is the total profit.
  - How do we deal with negative profits?
  - How about the prerequisite constraints?
- Second idea: make it hugely costly for any prerequisite constraint to be broken, so that cuts that violate any cannot have minimum capacity.
- **Remark:** This is a commonly used idea when doing reductions among problems: convert "hard" constraints to "soft" ones, and when punishments are high enough, soft constraints become hard.