Announcements

- Homework 8 and 9
- Exam practice problems on course homepage
- Final Exam: Monday, December 14
  - 24 hour take home exam
  - Target: 2 to 4 hours of work time

| Wed, Dec 2 | Net Flow Applications |
| Fri, Dec 4 | Net Flow Applications + NP-Completeness |
| Mon, Dec 7 | NP-Completeness |
| Wed, Dec 9 | NP-Completeness |
| Fri, Dec 11 | Beyond NP-Completeness |
| Mon, Dec 14 | Final Exam |

Outline

- Network flow definitions
- Flow examples
- Augmenting Paths
- Residual Graph
- Ford Fulkerson Algorithm
- Cuts
- Maxflow-MinCut Theorem
- Maxflow Algorithms
  - Simple applications of Max Flow
  - Non-simple applications of Max Flow

Cut in a graph

- Cut: Partition of V into disjoint sets S, T with s in S and t in T.
- \( \text{Cap}(S,T) \): sum of the capacities of edges from S to T
- Problem: Find the s-t Cut with minimum capacity

Max Flow / Min Cut

Review

Max Flow - Min Cut Theorem

- There exists a cut S, T such that \( \text{Flow}(S,T) = \text{Cap}(S,T) \)
- Proof also shows that Ford Fulkerson algorithm finds a maximum flow
Network flow performance

- Ford-Fulkerson algorithm
  \(O(mn)\)
- Find the maximum capacity augmenting path
  \(O(m^2\log(C))\) time algorithm for network flow
- Find the shortest augmenting path
  \(O(m^2n)\) time algorithm for network flow
- Find a blocking flow in the residual graph
  \(O(mn\log n)\) time algorithm for network flow
- Preflow Push Algorithm
  \(O(mn\log n)\)

Problem Reduction

- Reduce Problem A to Problem B
  - Convert an instance of Problem A to an instance of Problem B
  - Use a solution of Problem B to get a solution to Problem A
- Practical
  - Use a program for Problem B to solve Problem A
- Theoretical
  - Show that Problem B is at least as hard as Problem A

Problem Reduction Example

- Reduce the problem of finding the Maximum of a set of integers to finding the Minimum of a set of integers

  Find the maximum of: 8, -3, 2, 12, 1, -6

  Construct an equivalent minimization problem

Reduce MST to MST+

- P1: MST
  - Find the Minimum spanning tree for a graph with integer costs
- P2: MST+
  - Find the Minimum Spanning Tree for a graph with non-negative integer costs

Undirected Network Flow

- Undirected graph with edge capacities
- Flow may go either direction along the edges (subject to the capacity constraints)

Bipartite Matching

- A graph \(G=(V,E)\) is bipartite if the vertices can be partitioned into disjoints sets \(X,Y\)
- A matching \(M\) is a subset of the edges that does not share any vertices
- Find a matching as large as possible

Construct an equivalent flow problem
Application

• A collection of teachers
• A collection of courses
• And a graph showing which teachers can teach which courses

RA  143
PB  373
ME  414
DG  415
AK  417

Multi-source network flow

• Multi-source network flow
  – Sources s₁, s₂, ..., sₖ
  – Sinks t₁, t₂, ..., tⱼ
• Solve with Single source network flow

Resource Allocation: Assignment of reviewers

• A set of papers P₁, ..., Pₙ
• A set of reviewers R₁, ..., Rₘ
• Paper Pᵢ requires Aᵢ reviewers
• Reviewer Rⱼ can review Bⱼ papers
• For each reviewer Rⱼ, there is a list of paper Lⱼ₁, ..., Lⱼₖ that Rⱼ is qualified to review

Baseball elimination

• Can the Dinosaurs win the league?
• Remaining games:

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ants</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Bees</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Cockroaches</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Dinosaurs</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Baseball elimination

• Can the Fruit Flies win or tie the league?
• Remaining games:

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ants</td>
<td>17</td>
<td>12</td>
</tr>
<tr>
<td>Bees</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>Cockroaches</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>Dinosaurs</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>Earthworms</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>Fruit Flies</td>
<td>12</td>
<td>15</td>
</tr>
</tbody>
</table>
Assume Fruit Flies win remaining games

- Fruit Flies are tied for first place if no team wins more than 19 games
- Allowable wins:
  - Ants (2)
  - Bees (3)
  - Cockroaches (3)
  - Dinosaurs (5)
  - Earthworms (5)
- 18 games to play:

<table>
<thead>
<tr>
<th></th>
<th>W</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ants</td>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>Bees</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Cockroaches</td>
<td>16</td>
<td>9</td>
</tr>
<tr>
<td>Dinosaurs</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Earthworms</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>Fruit Flies</td>
<td>19</td>
<td>15</td>
</tr>
</tbody>
</table>

Remaining games


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 foreground from background

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
- $Q(A,B) = \Sigma_{i\in A}a_i + \Sigma_{j\in B}b_j - \Sigma_{(i,j)\in E, i\in A, j\in B}p_{ij}$
Open Pit Mining (Task selection)

Application of Min-cut

- Open Pit Mining Problem
- 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

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
Determine an optimal mine

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 \( \text{Profit}(T) \)

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

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