Chapter 7

Network Flow
7.5 Bipartite Matching
Matching

- Input: undirected graph $G = (V, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a max cardinality matching.
Bipartite Matching

Bipartite matching.

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a max cardinality matching.

Matching $1-2'$, $3-1'$, $4-5'$
Bipartite matching.

- Input: undirected, bipartite graph $G = (L \cup R, E)$.
- $M \subseteq E$ is a matching if each node appears in at most one edge in $M$.
- Max matching: find a max cardinality matching.

Max matching: $1-1'$, $2-2'$, $3-3'$, $4-4'$
Max flow formulation.

- Create digraph $G' = (L \cup R \cup \{s, t\}, E')$.
- Direct all edges from $L$ to $R$, and assign unit (or infinite) capacity.
- Add source $s$, and unit capacity edges from $s$ to each node in $L$.
- Add sink $t$, and unit capacity edges from each node in $R$ to $t$. 

```
Bipartite Matching
```

```
\[ G' \]
```

```
\[ s \rightarrow 1 \rightarrow 1' \rightarrow t \]
```

```
\[ s \rightarrow 2 \rightarrow 2' \rightarrow t \]
```

```
\[ s \rightarrow 3 \rightarrow 3' \rightarrow t \]
```

```
\[ s \rightarrow 4 \rightarrow 4' \rightarrow t \]
```

```
\[ s \rightarrow 5 \rightarrow 5' \rightarrow t \]
```

```
\[ L \rightarrow 1 \rightarrow 1' \rightarrow R \]
```

```
\[ L \rightarrow 2 \rightarrow 2' \rightarrow R \]
```

```
\[ L \rightarrow 3 \rightarrow 3' \rightarrow R \]
```

```
\[ L \rightarrow 4 \rightarrow 4' \rightarrow R \]
```

```
\[ L \rightarrow 5 \rightarrow 5' \rightarrow R \]
```

```
Theorem. Max cardinality matching in $G = \text{value of max flow in } G'$.

Pf. ≤
- Given max matching $M$ of cardinality $k$.
- Consider flow $f$ that sends 1 unit along each of $k$ paths.
- $f$ is a flow, and has value at least $k$. □
Bipartite Matching: Proof of Correctness

Theorem. Max cardinality matching in $G = \text{value of max flow in } G'$.

Pf. ≥

- Let $f$ be a max flow in $G'$ of value $k$.
- Integrality theorem $\Rightarrow$ $k$ is integral and can assume $f$ is 0-1.
- Consider $M = \text{set of edges from L to R with } f(e) = 1$.
  - each node in L and R participates in at most one edge in $M$
  - $|M| = k$: consider cut $(L \cup s, R \cup t)$
Bipartite Matching: Running Time

Which max flow algorithm to use for bipartite matching?

- Generic augmenting path: $O(m \text{val}(f^*)) = O(mn)$.
- Capacity scaling: $O(m^2 \log C) = O(m^2)$.
- Shortest augmenting path: $O(m n^{1/2})$.

Which max flow algorithm to use for Non-bipartite matching?

- Structure of non-bipartite graphs is more complicated, but well-understood. [Tutte-Berge, Edmonds-Galai]
- Blossom algorithm: $O(n^4)$. [Edmonds 1965]
- Best known: $O(m n^{1/2})$. [Micali-Vazirani 1980]
7.6 Disjoint Paths
Disjoint path problem. Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find the max number of edge-disjoint $s$-$t$ paths.

Def. Two paths are edge-disjoint if they have no edge in common.

Ex: communication networks.
Disjoint path problem. Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find the max number of edge-disjoint $s$-$t$ paths.

Def. Two paths are edge-disjoint if they have no edge in common.

Ex: communication networks.
**Max flow formulation:** assign unit capacity to every edge.

**Theorem.** Max number edge-disjoint s-t paths equals max flow value.

**Pf.** ≤
- Suppose there are \( k \) edge-disjoint paths \( P_1, \ldots, P_k \).
- Set \( f(e) = 1 \) if \( e \) participates in some path \( P_i \); else set \( f(e) = 0 \).
- Since paths are edge-disjoint, \( f \) is a flow of value \( k \).
Max flow formulation: assign unit capacity to every edge.

Theorem. Max number edge-disjoint s-t paths equals max flow value.

Pf. ≥

- Suppose max flow value is k.
- Integrality theorem ⇒ there exists 0-1 flow $f$ of value $k$.
- Consider edge $(s, u)$ with $f(s, u) = 1$.
  - by conservation, there exists an edge $(u, v)$ with $f(u, v) = 1$
  - continue until reach t, always choosing a new edge
- Produces $k$ (not necessarily simple) edge-disjoint paths.

can eliminate cycles to get simple paths if desired
Network Connectivity

Network connectivity. Given a digraph $G = (V, E)$ and two nodes $s$ and $t$, find min number of edges whose removal disconnects $t$ from $s$.

Def. A set of edges $F \subseteq E$ disconnects $t$ from $s$ if all $s$-$t$ paths uses at least one edge in $F$. 
Theorem. [Menger 1927] The max number of edge-disjoint s-t paths is equal to the min number of edges whose removal disconnects t from s.

Pf. $\leq$

- Suppose the removal of $F \subseteq E$ disconnects t from s, and $|F| = k$.
- All s-t paths use at least one edge of F. Hence, the number of edge-disjoint paths is at most k. $\blacksquare$
Disjoint Paths and Network Connectivity

Theorem. [Menger 1927] The max number of edge-disjoint $s$-$t$ paths is equal to the min number of edges whose removal disconnects $t$ from $s$.

Pf. ≥
- Suppose max number of edge-disjoint paths is $k$.
- Then max flow value is $k$.
- Max-flow min-cut $\Rightarrow$ cut $(A, B)$ of capacity $k$.
- Let $F$ be set of edges going from $A$ to $B$.
- $|F| = k$ and disconnects $t$ from $s$. ▪
7.12 Baseball Elimination

Acton Vale

Thetford Mines
Baseball Elimination

<table>
<thead>
<tr>
<th>Team i</th>
<th>Wins $w_i$</th>
<th>Losses $l_i$</th>
<th>To play $r_i$</th>
<th>Against $= r_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV</td>
<td>83</td>
<td>71</td>
<td>8</td>
<td>- 1 6 1</td>
</tr>
<tr>
<td>SHA</td>
<td>80</td>
<td>79</td>
<td>3</td>
<td>1 - 0 2</td>
</tr>
<tr>
<td>TM</td>
<td>78</td>
<td>78</td>
<td>6</td>
<td>6 0 - 0</td>
</tr>
<tr>
<td>MTL</td>
<td>77</td>
<td>82</td>
<td>3</td>
<td>1 2 0 -</td>
</tr>
</tbody>
</table>

Which teams have a chance of finishing the season with most wins?

- Montréal eliminated since it can finish with at most 80 wins, but Acton Vale already has 83.
- $w_i + r_i < w_j \Rightarrow$ team $i$ eliminated.
- Only reason sports writers appear to be aware of.
- Sufficient, but not necessary!
### Baseball Elimination

<table>
<thead>
<tr>
<th>Team</th>
<th>Wins $w_i$</th>
<th>Losses $l_i$</th>
<th>To play $r_i$</th>
<th>Against = $r_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV</td>
<td>83</td>
<td>71</td>
<td>8</td>
<td>- 1 6 1</td>
</tr>
<tr>
<td>SHA</td>
<td>80</td>
<td>79</td>
<td>3</td>
<td>1 - 0 2</td>
</tr>
<tr>
<td>TM</td>
<td>78</td>
<td>78</td>
<td>6</td>
<td>6 0 - 0</td>
</tr>
<tr>
<td>MTL</td>
<td>77</td>
<td>82</td>
<td>3</td>
<td>1 2 0 -</td>
</tr>
</tbody>
</table>

**Which teams have a chance of finishing the season with most wins?**

- Shawinigan can win 83, but is still eliminated...
- If Acton Vale loses a game, then some other team wins one.

**Remark.** Answer depends not just on how many games already won and left to play, but also on whom they're against.
Baseball elimination problem.

- Set of teams $S$.
- Distinguished team $x \in S$.
- Team $x$ has won $w_x$ games already.
- Teams $x$ and $y$ play each other $r_{xy}$ additional times.
- Is there any outcome of the remaining games in which team $x$ finishes with the most (or tied for the most) wins?
Can team 3 finish with most wins?

- Assume team 3 wins all remaining games ⇒ $w_3 + r_3$ wins.
- Divide remaining games so that all teams have $\leq w_3 + r_3$ wins.

Baseball Elimination: Max Flow Formulation

games left

$\infty$

$w_3 + r_3 - w_4$

team 4 can still win this many more games without eliminating team 3
Theorem. Team 3 is not eliminated iff max flow saturates all edges leaving source $s$.

- Integrality theorem $\implies$ each remaining game between $x$ and $y$ added to number of wins for team $x$ or team $y$.
- Capacity on $(x, t)$ edges ensure no team wins too many games.
### Baseball Elimination: Explanation for Sports Writers

<table>
<thead>
<tr>
<th>Team</th>
<th>Wins $w_i$</th>
<th>Losses $l_i$</th>
<th>To play $r_i$</th>
<th>Against $= r_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV</td>
<td>75</td>
<td>59</td>
<td>28</td>
<td>- 3 8 7 3</td>
</tr>
<tr>
<td>Sha</td>
<td>71</td>
<td>63</td>
<td>28</td>
<td>3 - 2 7 4</td>
</tr>
<tr>
<td>TM</td>
<td>69</td>
<td>66</td>
<td>27</td>
<td>8 2 - 0 0</td>
</tr>
<tr>
<td>She</td>
<td>63</td>
<td>72</td>
<td>27</td>
<td>7 7 0 - -</td>
</tr>
<tr>
<td>Mtl</td>
<td>49</td>
<td>86</td>
<td>27</td>
<td>3 4 0 0 -</td>
</tr>
</tbody>
</table>

Which teams have a chance of finishing the season with most wins?
- Montréal could finish season with $49 + 27 = 76$ wins.
Baseball Elimination: Explanation for Sports Writers

Which teams have a chance of finishing the season with most wins?
- Montréal could finish season with $49 + 27 = 76$ wins.

Certificate of elimination. $R = \{AV, Sha, TM, She\}$
- Have already won $w(R) = 278$ games.
- Must win at least $r(R) = 27$ more (loosing all non-$R$ games).
- Average team in $R$ wins at least $305/4 > 76$ games.
Certificate of elimination.

\[ T \subseteq S, \quad w(T) := \sum_{i \in T} w_i, \quad r(T) := \sum_{\{x,y\} \in T} r_{x,y}/2, \]

If \( \frac{w(T) + r(T)}{|T|} > w_z + r_z \) then z is eliminated (by subset T).

Theorem. [Hoffman-Rivlin 1967] Team z is eliminated iff there exists a subset \( T^* \) that eliminates z.

Proof idea. Let \( T^* = \) team nodes in A (on source side) of min cut (A,B).
Baseball Elimination: Explanation for Sports Writers

Pf of theorem.
- Use max flow formulation, and consider min cut \((A, B)\).
- Define \(T^*\) = team nodes in \(A\) (on source side of min cut).
- Observe \(x-y \in A\) iff both \(x \in T^*\) (\(\in A\)) and \(y \in T^*\) (\(\in A\)).
  - infinite capacity edges ensure if \(x-y \in A\) then \(x \in A\) and \(y \in A\)
  - if \(x \in A\) and \(y \in A\) but \(x-y \in B\), then adding \(x-y\) to \(A\) decreases capacity of cut

Team x can still win this many more games without eliminating team z.
Pf of theorem.

- Use max flow formulation, and consider min cut \((A, B)\).
- Define \(T^*\) = team nodes in \(A\) (on source side of min cut).
- Observe \(x-y \in A\) iff both \(x \in T^*\) and \(y \in T^*\).
- \(r(S - \{z\}) > \text{cap}(A, B)\)

\[
\begin{align*}
\text{capacity of game edges leaving } A & \quad + \quad \sum_{x \in T^*} (w_z + r_z - w_x) \\
= & \quad r(S - \{z\}) - r(T^*) \\
= & \quad r(S - \{z\}) - r(T^*) - w(T^*) + |T^*| (w_z + r_z)
\end{align*}
\]

- Rearranging terms:

\[
w_z + r_z < \frac{w(T^*) + r(T^*)}{|T^*|}
\]

Team \(x\) can still win this many more games without eliminating team \(z\).
7.10 Image Segmentation
Image Segmentation

Image segmentation.
- Central problem in image processing.
- Divide image into coherent regions.

Ex: Three people standing in front of complex background scene.
Identify each person as a coherent object.
Image Segmentation

Foreground / background segmentation.
- Label each pixel in picture as belonging to foreground or background.
- $V =$ set of pixels, $E =$ pairs of neighbouring pixels.
- $a_i \geq 0$ is likelihood pixel $i$ in foreground.
- $b_i \geq 0$ is likelihood pixel $i$ in background.
- $p_{ij} \geq 0$ is separation penalty for labelling one of $i$ and $j$ as foreground, and the other as background.

Goals.
- Accuracy: if $a_i > b_i$ in isolation, prefer to label $i$ in foreground.
- Smoothness: if many neighbours of $i$ are labeled foreground, we should be inclined to label $i$ as foreground.
- Find partition $(A, B)$ that maximizes:
  $$\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij}$$
  where $|A \cap \{i,j\}| = 1$.
Image Segmentation

Formulate as min cut problem.

- Maximization.
- No source or sink.
- Undirected graph.

Turn into minimization problem.

- Maximizing $\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{(i,j) \in E} p_{ij}$

  is equivalent to minimizing $\left(\sum_{i \in V} a_i + \sum_{j \in V} b_j\right) - \sum_{i \in A} a_i - \sum_{j \in B} b_j + \sum_{(i,j) \in E} p_{ij}$

  or alternatively $\sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{(i,j) \in E} p_{ij}$
Image Segmentation

Formulate as min cut problem.

- $G' = (V', E')$.
- Add source $s$ to correspond to foreground; add sink $t$ to correspond to background.
- Use two anti-parallel edges instead of undirected edge.
Consider min cut \((A, B)\) in \(G'\).

- \(A = \text{foreground.}\)

\[
\text{cap}(A, B) = \sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{(i, j) \in E} p_{ij}
\]

if \(i\) and \(j\) on different sides,
\(p_{ij}\) counted exactly once

- Precisely the quantity we want to minimize.
7.11 Project Selection
Project Selection

Projects with prerequisites.

- Set $P$ of possible projects. Project $v$ has associated revenue $p_v$.
  - some projects generate money: create interactive e-commerce interface, redesign web page
  - others cost money: upgrade computers, get site license
- Set of prerequisites $E$. If $(v, w) \in E$, can’t do project $v$ and unless also do project $w$.
- A subset of projects $A \subseteq P$ is feasible if the prerequisite of every project in $A$ also belongs to $A$.

Project selection. Choose a feasible subset of projects to maximize revenue:

$$\sum_{v \in A} p_v$$
Prerequisite graph.

- Include an edge from $v$ to $w$ if can't do $v$ without also doing $w$.
- $\{v, w, x\}$ is feasible subset of projects.
- $\{v, x\}$ is infeasible subset of projects.
Min cut formulation.

- Assign capacity $\infty$ to all prerequisite edge.
- Add edge $(s, v)$ with capacity $p_v$ if $p_v > 0$.
- Add edge $(v, t)$ with capacity $-p_v$ if $p_v < 0$.
- For notational convenience, define $p_s = p_t = 0$. 
Project Selection: Min Cut Formulation

**Claim.** \((A, B)\) is min cut iff \(A - \{s\}\) is optimal set of projects.

- Infinite capacity edges ensure \(A - \{s\}\) is feasible.
- Max revenue because:

\[
\text{cap}(A, B) = \sum_{v \in B: p_v > 0} p_v + \sum_{v \in A: p_v < 0} (-p_v)
\]

\[
= \left( \sum_{v: p_v > 0} p_v \right) - \sum_{v \in A} p_v
\]

constant
Open-pit mining. (studied since early 1960s)

- Blocks of earth are extracted from surface to retrieve ore.
- Each block $v$ has net value $p_v = \text{value of ore} - \text{processing cost}$.
- Can't remove block $v$ before $w$ or $x$. 
7.7 Extensions to Max Flow
Circulation with Demands

Circulation with demands.
- Directed graph $G = (V, E)$.
- Edge capacities $c(e), e \in E$.
- Node supply and demands $d(v), v \in V$.

$\uparrow$

demand if $d(v) > 0$; supply if $d(v) < 0$; transshipment if $d(v) = 0$

Def. A circulation is a function $f$ that satisfies:
- For each $e \in E$: $0 \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem: given $(G, c, d)$, does there exist a circulation?
Circulation with Demands

Necessary condition: sum of supplies = sum of demands.

\[ \sum_{v : d(v) > 0} d(v) = \sum_{v : d(v) < 0} d(v) =: D \]

Pf. Sum conservation constraints for every demand node \( v \).
Circulation with Demands

Max flow formulation.

\[ G: \]

\[ \begin{align*}
G: & \quad \begin{array}{ccc}
\text{supply} & -6 & \text{demand} \\
10 & 7 & 7 \\
3 & 6 & 4 \\
10 & & 9 \\
& 4 & 11 \\
& & \\
\end{array}
\end{align*} \]
Circulation with Demands

Max flow formulation.

- Add new source $s$ and sink $t$.
- For each $v$ with $d(v) < 0$, add edge $(s, v)$ with capacity $-d(v)$.
- For each $v$ with $d(v) > 0$, add edge $(v, t)$ with capacity $d(v)$.
- Claim: $G$ has circulation iff $G'$ has max flow of value $D$. 

$G'$:

![Diagram of network flow with sources, sinks, and edges with capacities.]

saturates all edges leaving $s$ and entering $t$
Circulation with Demands

**Integrality theorem.** If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.

**Pf.** Follows from max flow formulation and integrality theorem for max flow.

**Characterization.** Given \((G, c, d)\), there does not exist a circulation iff there exists a node partition \((A, B)\) such that \(\sum_{v \in B} d(v) > \text{cap}(A, B)\).

**Pf idea.** Look at min cut in \(G'\).

- demand by nodes in \(B\) exceeds supply of nodes in \(B\) plus max capacity of edges going from \(A\) to \(B\)
Circulation with Demands and Lower Bounds

Feasible circulation.

- Directed graph $G = (V, E)$.
- Edge capacities $c(e)$ and lower bounds $\ell(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$.

**Def.** A circulation is a function that satisfies:

- For each $e \in E$: $\ell(e) \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

Circulation problem with lower bounds. Given $(G, \ell, c, d)$, does there exists a circulation?
Idea. Model lower bounds with demands.

- Send $\ell(e)$ units of flow along edge $e$.
- Update demands of both endpoints.

Theorem. There exists a circulation in $G$ iff there exists a circulation in $G'$. If all demands, capacities, and lower bounds in $G$ are integers, then there is a circulation in $G$ that is integer-valued.

Pf sketch. $f(e)$ is a circulation in $G$ iff $f'(e) = f(e) - \ell(e)$ is a circulation in $G'$.
7.8 Survey Design
Survey Design

Survey design.

- Design survey asking $n_c$ consumers about $n_p$ products.
- Can only survey consumer $i$ about a product $j$ if he owns it.
- Ask consumer $i$ between $c_i$ and $c_i'$ questions.
- Ask between $p_j$ and $p_j'$ consumers about product $j$.

Goal. Design a survey that meets these specs, if possible.
**Algorithm.** Formulate as a circulation problem with lower bounds.

- Include an edge \((i, j)\) if customer \(i\) owns product \(j\).
- Integer circulation \(\iff\) feasible survey design.
- \((t, s)\) makes sure problem has a circulation.
Chapter 7

Network Flow

Algorithm Design

JON KLEINBERG • ÉVA TARDOS

Slides by Kevin Wayne.
Copyright © 2005 Pearson-Addison Wesley.
All rights reserved.