Applications of Network Flows

**Last time:** Max matching in bipartite graph is defined by the maximum flow.

**Proof:** Max flow will give a maximum matching.

Consider the main property of a bipartite graph: vertices in one partition have no connections between them. That allows us to set up "disjoint" unique paths from s to t. Where "disjoint" means that no two paths go through same vertex in either partition. For example, by having a unit flow on the s -> v -> u -> t path, we can't have flow on the s -> v -> w -> t path. On this property we’ll base our proof.

**Claim:**

1) If $M$ is a matching in a bipartite graph $G$, then there exists a flow $f$ in $F$ (flow network for $G$) such that $size(f) = M$. In other words, we can write:

   $size(f^*) \geq |M^*|$, where $f^*$ is a max flow and $M^*$ is a size of maximum matching set.

2) If $f$ is an integer value flow in $F$, then there exists a matching $M$ in $G$ of size $size(f)$.

   $size(f^*) \leq |M^*|$

For the claim # 2, let $M = \{(u, v) \in G \mid f(u, v) = 1\}$ then $M$ forms a matching.

Why? Because all capacities of the edges going out of $s$ (source) are equal to 1 and all capacities of the edges coming to $t$ (sink) are equal to 1. Therefore, we have both inequalities together, which defines: $size(f^*) = |M^*|$ maximum flow equals to maximum matching.

Pennant Race Problem (1965)

The Pennant Race problem is:

**Input:** Given a baseball team $A$, a list of any other teams $T_1, T_2, \ldots, T_n$ to win the pennant against team $A$. There is a win/loss rate for all teams which already played the games.

**Question:** Is it possible that team $A$ can end the season having won at least as many games as any other team?

**Conditions:**

- Since it’s baseball, each game has winner and loser, no draw.
- All schedule games are played.
**Initial set up:**

Consider a table where each team has # of wins over the played games, potential winning points from the future games, and the set of the remaining games. For $A$ we can calculate the potential winning points easily, just assume $A$ won all the remaining games.

<table>
<thead>
<tr>
<th>Team</th>
<th># of wins</th>
<th>Potential win points (w)</th>
<th>Remaining games</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>3</td>
<td>6</td>
<td>$(A, T_1), (A, T_3), (A, T_4)$, $(T_1, T_3), (T_2, T_3), (T_2, T_4)$, $(T_1, T_4)$</td>
</tr>
<tr>
<td>$T_1$</td>
<td>4 = $w_1$</td>
<td>$w_1$</td>
<td></td>
</tr>
<tr>
<td>$T_2$</td>
<td>6 = $w_1$</td>
<td>$w_1$</td>
<td></td>
</tr>
<tr>
<td>$T_3$</td>
<td>5 = $w_1$</td>
<td>$w_1$</td>
<td></td>
</tr>
<tr>
<td>$T_4$</td>
<td>4 = $w_1$</td>
<td>$w_1$</td>
<td></td>
</tr>
</tbody>
</table>

Under assumption that $A$ will win all the remaining games:

- Let $w$ = # wins after $A$ wins all the remaining games;
- Let $w_i$ = # wins for $T_i$ (current);
- If $w < w_i$ for some $i$, then $A$ has no hope. Otherwise, consider the set of games remaining to be played $\{(T_i, T_j)\}$ where $i \neq j$.

We can transform the problem into a Bipartite graph. The idea is to think about wins as of 1 unit flows. Thus each game will have one “flow” = “win” to a team. Assume, $w > w_i$ for all $i$, such that we still have a hope before we run an algorithm.

![Bipartite Graph](image)

**Edges:**

- $(s, T_i, T_j)$ with capacity 1;
- $(T_i, T_j, T_i)$ any capacity $\geq$ 1;
- $(T_i, t)$ with capacity $w - w_i$, the maximum number of games $T_i$ can win without getting more wins than team $A$.

After constructing the graph, we run any algorithm to find the max flow of the bipartite graph. If size(max flow) is less than the number of games remaining to be played (after assuming team $A$ wins all its remaining games), then team $A$ has no hope..