CS420+500: Advanced Algorithm Design and Analysis

Lectures: Feburary 6 + Feburary 8, 2017

Prof. Will Evans *Scribe: Xiaomeng Ju*

In this lecture we covered:

- Maximum matching in bipartite graphs
- The pennant race problem

1 Maximum Matching in Bipartite Graphs

Recall in the last lecture, we discussed the maximum matching in a bipartite graph. A matching in a bipartite graph is a set of the edges chosen in such a way that no two edges share an endpoint. A maximum matching is a matching of maximum size (maximum number of edges). To solve this problem, given a bipartite graph $G = (V_1, V_2, E)$ we construct a flow network as follows:

- Add a source node s to the left side of the graph, and a sink node t to the right side of the graph.
- Direct all the edges from source to sink. The edge set in the flow network includes the original edge set E, edges from s to V_1, and edges from V_2 to t.
- Give all edges unit weight

Thus, we saw the following transformation

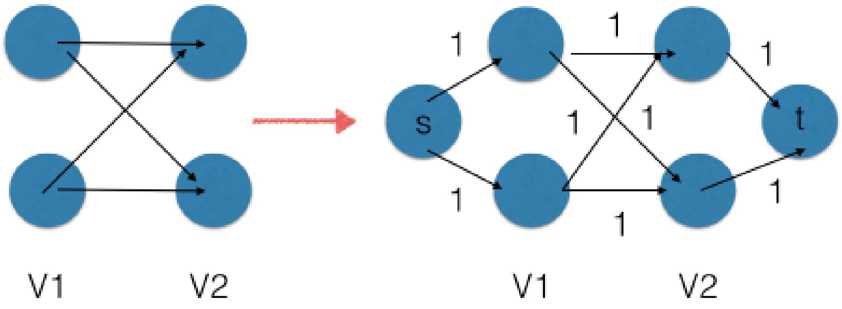

Figure 1: Transformation from a bipartite graph to a flow network

To obtain a maximum matching on the bipartite graph, we run the Ford-Fulkerson algorithm on the flow network. The resulted edges connecting V_1 and V_2 that have unit flow passing through them are the matching edges in the original bipartitie graph. Note that for this specific flow network, the Ford-Fulkerson algorithm runs in $O(|V| \times |E|)$ time, where V and E are the vertices and the edges of the constructed flow network respectively.

2 Pennant Race Problem

2.1 Problem Description

In baseball, the team with the highest number of wins gets a pennant at the end of the season. We are interested to determine whether a team still get a chance at wining the pennant at a given point in the season. At that given point of the season, the number wins for each team and the schedule of games left to be played are known.

- Let $T_1, ..., T_n, A$ be a list of teams competing for the pennant, where A is the team under consideration: we are interested to determine if there's still hope for A to get the pennant.
- Let W be a list of wins, with $w_i \in W$ denotes the wins that the ith team has at this time
- Let S be the schedule of remaining games. S is a list of tuples of $(T_i, T_j), i \neq j$. Each tuple indicating the two teams playing that game.

Consider the an example of the input:

$$\begin{aligned} T &= T_1, T_2, T_3, T_4, A \\ W &= 4, 6, 5, 4, 3 \\ S &= (T_1, A), (T_1, T_3), (T_2, T_3), (A, T_3), (T_2, T_4), (A, T_4), (T_1, T_2), \end{aligned}$$

2.2 Problem Discussion

Given the above description, we wish to calculate if A can still win the pennant. First, we calculate the highest number of wins that A can achieve from the remaining games and compare it with the number of wins of the other teams. To be more specific, we employ the following notation

- Let w be the number wins of A assuming that A wins all remaining games.
- Let w_i be the number of wins of T_i assuming that A wins all remaining games.

Note that w and w_i's are defined as the number of wins amass up to the end of the season. That is the number of current wins plus the hypothetical victories in all remaining games. If $w < w_i$ for some team T_i, then A has no hope to get the pennant. However, if $w \geq w_i$ for all $i \in \{1, .., n\}$, there is hope. We want to calculate w first, then assign wins to $T_1, ..., T_n$. The idea is to determine if it is possible that the assignment of wins will not yield any of $T_1, ..., T_n$ having more wins than w.

2.3 Problem Solution

We will consider this as a flow problem in a bipartite graph. The bipartite graph is specified by the set of schedule nodes S and the team nodes $T = \{T_1, ..., T_n\}$. The network is constructed by adding a source node s and a sink node t. There are three types of edges in the flow network with corresponding capacities:

- $(s, (T_i, T_j))$, where $(T_i, T_j) \in S$ with capacity 1.
- $((T_i, T_j), T_i)$ and $((T_i, T_j), T_j)$ where $(T_i, T_j) \in S$, with capacity 1.
- (T_i, t), $i = 1, ..., n$ with capacity $w - w_i$.

If the size of max-flow equals to the number of games to play (number of tuples in S), the A still has hope. Otherwise, A has no hope. Thus, the pennant race problem is solved by the associated max flow problem, which can be solve by using Ford-Folkerson algorithm.

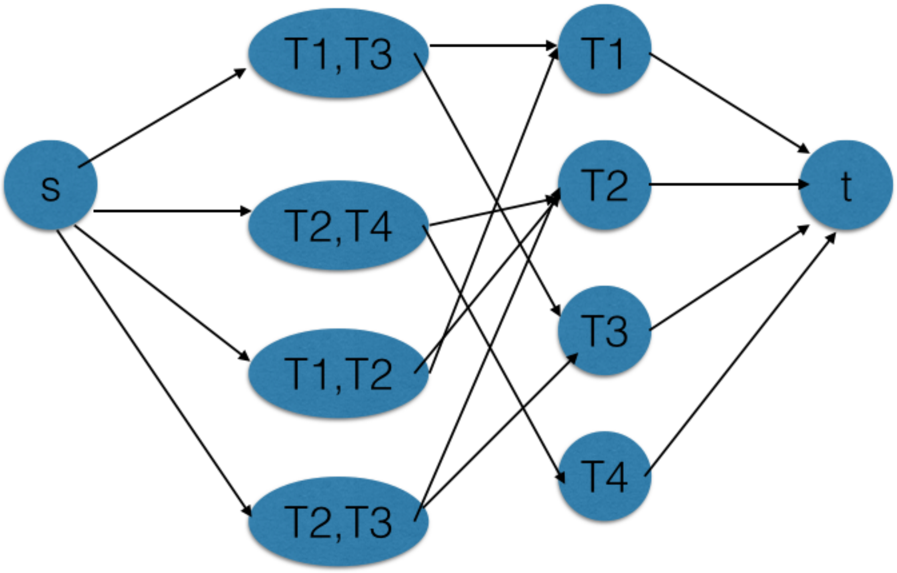

Figure 2: Constructed flow network