16.4 Baseball Elimination

Network flow has a lot of applications, with one of the more surprising being to a problem that arises in professional sports. Let T be a set of teams in a sports league, which, for historical reasons, let us assume is baseball. At any point during the season, each team, i, in T, will have some number, w_i, of wins, and will have some number, g_i, of games left to play. The ***baseball elimination*** problem is to determine whether it is possible for team i to finish the season in first place, given the games it has already won and the games it has left to play. Note that this depends on more than just the number of games left for team i, however; it also depends on the respective schedules of team i and the other teams. So let $g_{i,j}$ denote the number of games remaining between team i and team j, so that

$$g_i = \sum_{j \in T} g_{i,j}.$$

For example, see Table 16.12.

Team	Wins	Games Left	Schedule ($g_{i,j}$)			
i	w_i	g_i	LA	Oak	Sea	Tex
Los Angeles	81	8	-	1	6	1
Oakland	77	4	1	-	0	3
Seattle	76	7	6	0	-	1
Texas	74	5	1	3	1	-

Table 16.12: A set of teams, their standings, and their remaining schedule. Clearly, Texas is eliminated from finishing in first place, since it can win at most 79 games. In addition, even though it is currently in second place, Oakland is also eliminated, because it can win at most 81 games, but in the remaining games between LA and Seattle, either LA wins at least 1 game and finishes with at least 82 wins or Seattle wins 6 games and finishes with at least 82 wins.

With all the different ways for a team, k, to be eliminated, it might at first seem like it is computationally infeasible to determine whether team k is eliminated. Still, we can solve this problem by a reduction to a network flow problem. Let T' denote the set of teams other than k, that is, $T' = T - \{k\}$. Also, let L denote the set of games that are left to play among teams in T', that is,

$$L = \{\{i,j\}\colon\ i,j \in T' \text{ and } g_{i,j} > 0\}.$$

Finally, let W denote the largest number of wins that are possible for team k given the current standings, that is, $W = w_k + g_k$.

If $W < w_i$, for some team i, then k is eliminated directly by team i. So, let us assume that no single team eliminates team k. To consider how a combination of teams and game outcomes might eliminate team k, we create a graph, G, that has

as its vertices a source, s, a sink, t, and the sets T' and L. Then, let us include the following edges in G (see Figure 16.13):

- For each game pair, $\{i, j\}$, in L, add an edge $(s, \{i, j\})$, and give it capacity $g_{i,j}$.
- For each game pair, $\{i, j\}$, in L, add edges $(\{i, j\}, i)$ and $(\{i, j\}, j)$, and give these edges capacity $+\infty$.
- For each team, i, add an edge (i, t) and give it capacity $W - w_i$, which cannot be negative in this case, since we ruled out the case when $W < w_i$.

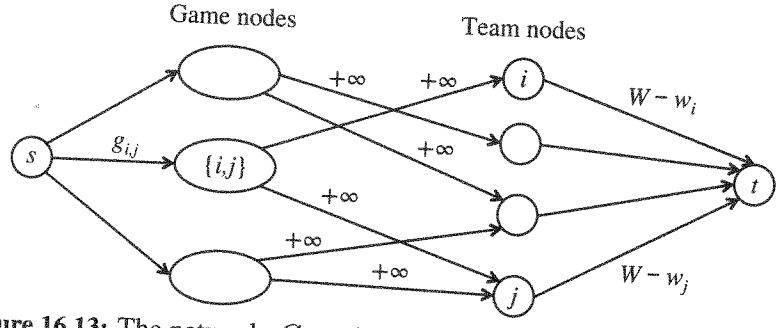

Figure 16.13: The network, G, to determine whether team k is eliminated.

The intuition behind the construction for G is that wins flow out from the source, s, are split at each game node, $\{i, j\}$, to allocate wins between each pair of teams, i and j, and then are absorbed by the sink, t. The flow on each edge, $(\{i, j\}, i)$, represents the number of games in which team i beats j, and the flow on each edge, (i, t), represents the number of remaining games that could be won by team i. Thus, maximizing the flow in G is equivalent to testing if it is possible to allocate wins among all the remaining games not involving team k so that no team goes above W wins. So we compute a maximum flow for G.

Suppose that the value of this maximum flow is

$$g(T') = \sum_{\{i,j\} \subseteq T'} g_{i,j},$$

which is the total number of games to be played by teams in T'. This implies that it is possible to allocate wins to all the remaining games so that no team has its win count go above W, that is, team k is not eliminated. If, on the other hand, the value of the maximum flow is strictly less than $g(T')$, then team k is eliminated, since it is not possible to allocate wins to all the remaining games with every team having a win count of at most W. Thus, we have the following:

Theorem 16.13: *We can solve the baseball elimination problem for any team in a set of n teams by solving a single maximum flow problem on a network with at most $O(n^2)$ vertices and edges.*