Lecture Notes CS:5350 Design and Analysis of Algorithms
Lecture 2: Aug 29, 2019
Scribe: Ali Asgher

1 Max Flow - Min Cut

1.1 Example

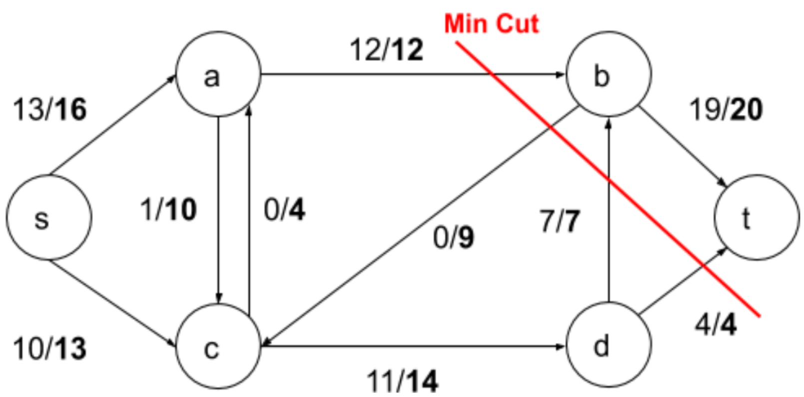

Figure 1: Graph showing the relation between min-cut and max-flow.

Fig 1 shows an example graph with maximum flow achieved from source s to target t and a minimum cut $S = \{s, a, c, d\}$ and $T = \{b, t\}$.

Value of flow $= f_{s\to a} + f_{s\to c} = 10 + 13 = 23$
Capacity of cut (S, T) = sum of total capacities of its edges going from S to T
$= 12_{a\to b} + 7_{d\to b} + 4_{d\to t} = 23$

Note: Edge b to c is not part of cut because it is not in the direction of $S \to T$

Because the value of flow is equal to the capacity of cut so it is a min-cut and flow through the graph from s to t has been maximized. i.e. all edges going from s to t will be saturated and all edges from t to s will be 0.

1.2 Applications

Fig 2 shows different categories of problems which can be reduced to Maxflow. Each of these problems can be reduced to maxflow by constructing a directed graph $G = (V, E)$ with appropriate

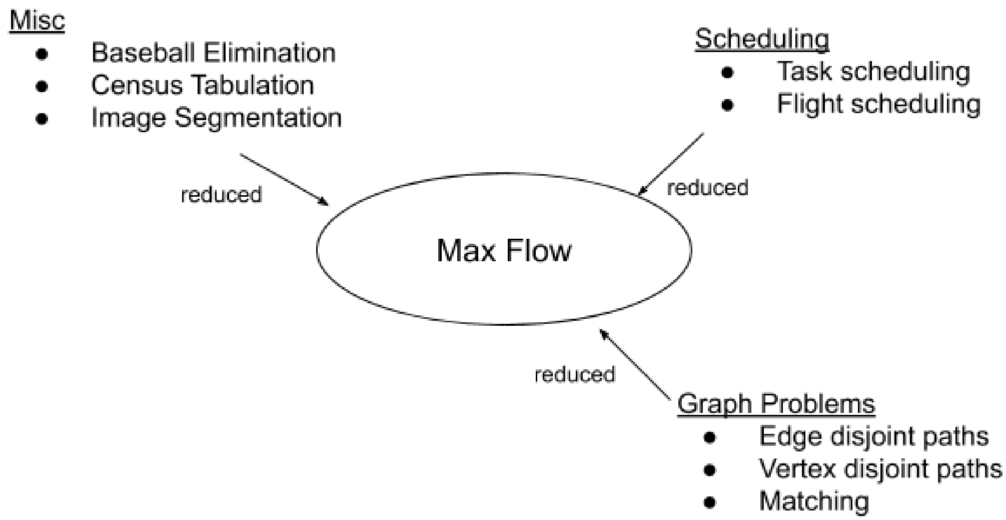

Figure 2: Applications which can be reduced to Max flow

capacities for that problem and call a Maxflow subroutine to solve it. Many of these problems are not just reduced to simple version of MaxFlow. Sometimes they are reduced to more complicated variants of MaxFlow (e.g. MaxFlow with both a lower bound and an upper bound on the amount of flow across each edge.) You can learn more about these applications of Maxflow in Chapter 11 from Jeff Erickson notes.

2 Baseball Elimination Problem

"Every year millions of American baseball fans eagerly watch their favorite team, hoping they will win a spot in the playoffs, and ultimately the World Series. Sadly, most teams are mathematically eliminated days or even weeks before the regular season ends. Often, it is easy to spot when a team is eliminated - they cannot win enough games to catch up to the current leader in their division. In the mid - 1960s, Benjamin Schwartz observed that this question can be modeled as a max-flow problem." [**?**]

To model the baseball elimiation problem to maxflow, lets say there are n teams numbered $1, 2, ..., n$ and the season is partially completed.

QUESTION: Can team n end the season with the most wins (possibly tied with other teams)?

This is a question of mathematical possibility. It doesn't depend on how good the team is or what form its players are in.

- $w[i], 1 \leq i \leq n$: represents number of wins of team i
- $r[i], 1 \leq i \leq n$: represents number of remaining games of team i
- $r[i,j], 1 \leq i,j \leq n$: represents number of remaining games between team i and team j
  Note: $r[i,j] = 0$
  $\implies r[i] = \sum_{j=1}^{n} r[i,j]$

INPUT: Given what are the outcomes of games already happened, can team n end up at top.

Problem: Can we find an arrangement of remaining games' outcomes so that team n end up at top. This problem can be reduced to maxflow.

Without Loss of Generality: Assume team n wins all its remaining games. (Another assumption could be that all teams don't win their remaining matches but it is not possible in baseball.)

2.1 Reduction

Since assumption is that team n has already won all games so in our graph, we won't have any nodes representing games that involve team n so the graph will be of teams from 1 to $n-1$.

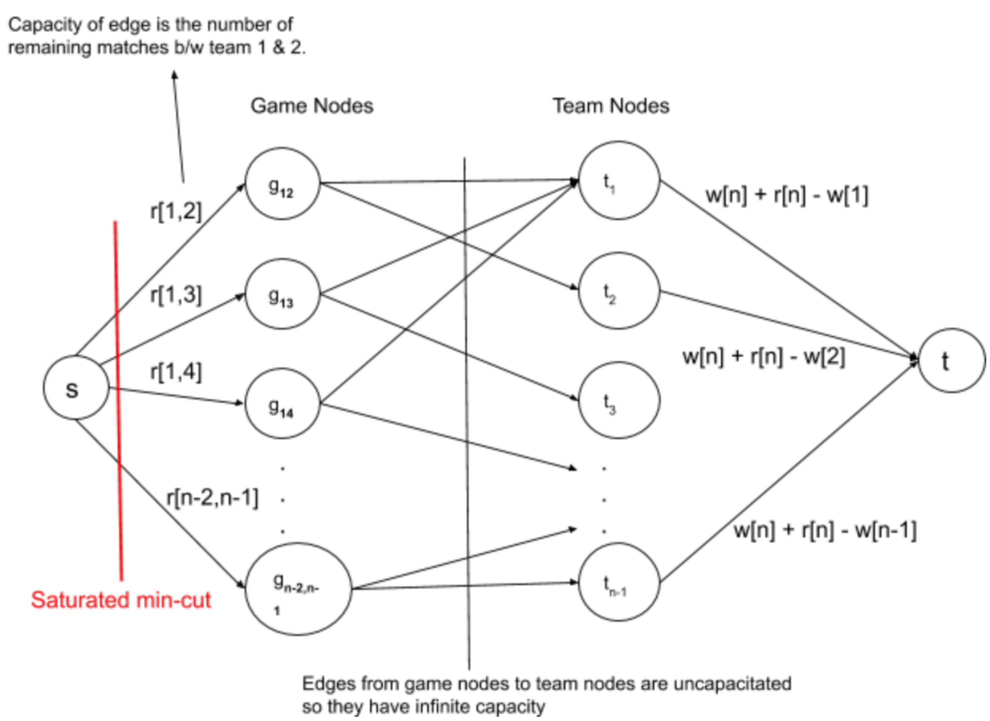

Figure 3: Graphical representation of baseball elimination problem

Figure 3 is the graphical representation of the baseball elimination problem. The capacity of an edge going from s to $g_{i,j}$ represents the number of remaining matches between team i and team j. A flow from $s \rightarrow g_i i, j$ represents remaining games which are to be played between team i and team j. The outcome of each game is mapped on an edge going from game node to team node therefore the edge from $g_{i,j}$ to t_i represents the games won by team i against team j and the edge from $g_{i,j}$ to t_j represents the games won by team j against team i. Finally, the flow that arrives into t_1 represents all games that t_1 has won. By flow conservation, this is also the amount of flow travelling from t_1 to t. Suppose that this quantity is x.

Since we want Team 1's wins $\leq$ Team n's wins

$$w[1] + x \leq w[n] + r[n]$$

Therefore,

$$x \leq w[n] + r[n] - w[1]$$

The above equation implies that placing a capacity $w[n]+r[n]-w[i]$ on each edge $t_i \rightarrow t$ ensures that the number of wins of Team i doesn't exceed the number of wins of Team n.

If a flow saturates all edges coming out of s then this would mean that all teams have played all their remaining matches. So we will know we have reached end of season with team n finishing on top. On the other hand if there is no flow that can saturate all edges leaving s, it means there is no way to complete the season while ensuring that team n ends up on top.

CLAIM: The graph has a maxflow with value $\sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} r[i,j]$ iff team n has the mathematical possibility of ending at 1st place.
The proof of this claim can be found in Jeff Erickson notes. In this way we can reduce baseball elimination problem to maxflow.

3 Example of a Linear Program

In order to solve a maxflow problem efficiently, formulate it as a linear program. To better understand a linear program, lets look at an example.

Consider a farmer who wants to plant wheat and barley on a L km^2 of land with a certain budget for fertilizer and pesticides.

- Plant - Wheat(w) and Barley(b)
- Land - L km^2
- Budget - Fertilizer F Kg, Pesticide P Kg
- Amount of fertilizer and pesticide required to plant Wheat
  - Fertilizer F_w kg/km^2
  - Pesticide P_w kg/km^2
- Amount of fertilizer and pesticide required to plant Barley
  - Fertilizer F_b kg/km^2
  - Pesticide P_b kg/km^2
- Price of planting Wheat and Barley
  - Wheat S_w \$/km^2

– Barley S_b $\$/km^2$

Choice: How much of L km^2 of land should farmer use for wheat and how much for barley? So the farmer's choice variables are
$x_w = km^2$ of wheat planted
$x_b = km^2$ of barley planted

Optimization Problem: Maximize the objective function

$$\max_{x_b, x_w} S_w x_w + S_b x_b$$

with the following constraints

$$\begin{gathered} F_w x_w + F_b x_b \leq F \\ P_w x_w + P_b x_b \leq P \\ x_w + x_b \leq L \\ x_w, x_b \geq 0 \end{gathered}$$

This is an example of a linear program in 2 variables with a linear objective function and linear constraints.

3.1 Geometric View of a Linear Program

Figure 4 shows an example geometric representation of linear problem that we formulated. To maximize the objective function, we look at all vertices of the feasible region and find the one that maximizes our objective function. In the above example, the feasible region has 5 vertices.

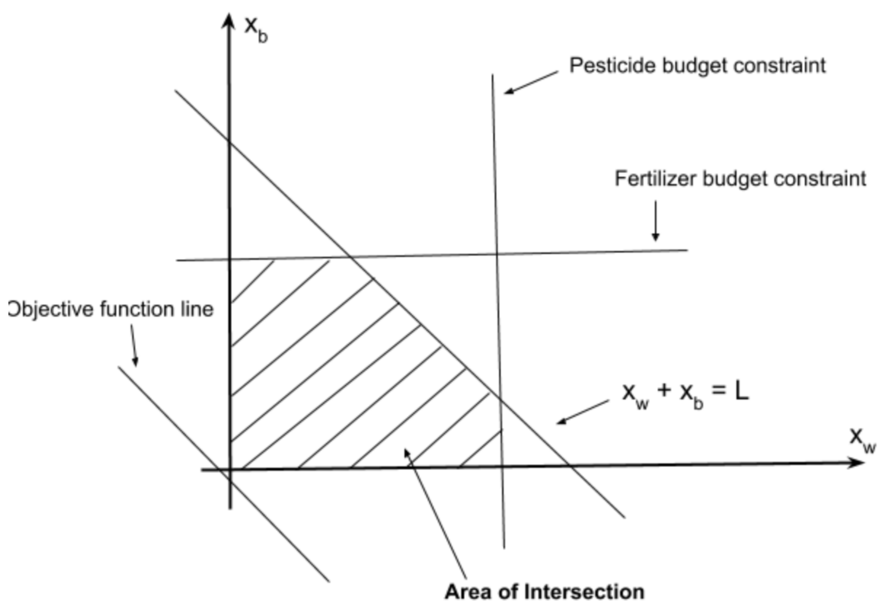

Figure 4: Example geometric representation of Farmer's linear program

3.2 History

Evaluating each vertex of the feasible region could result in an exponential time algorithm because the number of vertices can be exponential in number of variables and constraints. So there was to need to solve these LP problems efficiently

- Dantzig (1940) came up with *Simplex* method but Simplex method is not polynomial-time in the worst case.
- Khachiyan (1979) came up with first polynomial-time algorithm for linear programs, but his contribution was mostly theoretical.
- Karmakar (1984) designed *interior point method* which is both practical and runs in worst-case polynomial time.

References

[1] J. Erickson, "Algorithms by Jeff Erickson", Chapter 11.6 Baseball Elimination, Link: http://jeffe.cs.illinois.edu/teaching/algorithms/book/11-maxflowapps.pdf