CPSC 500:Fundamentals of Algorithm Design and Analysis Scribe

Nov.13, 2013.

Scribe: Fang Yuan Chi

1. Introduction

In this lecture, we continued discussion on the pennant race problem, which was introduced during our last lecture. Then we investigated in open pit mining problem and duality of linear programming.

2. Pennant Race Problem (Continued)

2.1 Formal Problem definition:

Input:

- A list of teams $T_1, T_2, T_3, \ldots, T_n$ with win/loss record for each team.
- A special team A.
- A list of games remaining to be played.

Output: Determine if it is possible for team A to end the season winning the most (at least as many, the results could have tie) games as any teams.

2.2 Example

Suppose we have five teams: A, T_1, T_2, T_3 and T_4. Each team and its associated number of wins can be represented by (T_i, W_i) and the special team A has a number of wins W. An example can be shown as follows:

Team	Number of Wins
A	$W = 3$
T_1	$W_1 = 4$
T_2	$W_2 = 6$
T_3	$W_3 = 5$
T_4	$W_4 = 4$

Also, there are 7 remaining games for this season:

$$(\text{A}:T_1)(\text{A}:T_3)\ (\text{A}:T_4)\ (T_1:T_3)\ (T_1:T_2)\ (T_2:T_4)\ (T_2:T_3)$$

How do we determine if it is possible for team A to win the most at the end of this season?

First we assume that team A will win all the games it is going to play. Then, the number of wins for team A increases (the change is marked in red):

Team	Number of Wins
A	$W = 6$
T_1	$W_1 = 4$
T_2	$W_2 = 6$
T_3	$W_3 = 5$
T_4	$W_4 = 4$

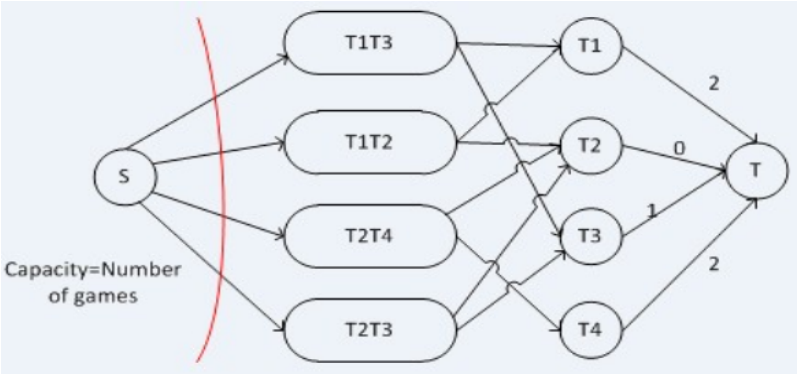
And there are only 4 games left:

$$(T_1{:}T_3)\ (T_1{:}T_2)\ (T_2{:}T_4)\ (T_2{:}T_3)$$

Then we check $W > w_i$ for all T_i. That is, in order for team A to have the maximum number of wins, each team should have at most 6 wins at the end of the season. By examining the table, we define the following constraints:

1. T_1 can win 2 more games;
2. T_2 can win 0 more games;
3. T_3 can win 1 more game;
4. T_4 can win 2 more games.

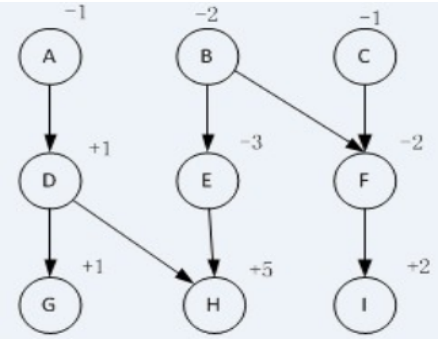
We can solve this problem by finding the maximum flow in the flow network shown in the figure below.



As seen in the diagram, if the maximum flow is less than the number of games, then that means one of the teams will have to win more than the respective constraint, which means A will have no hope of winning. Therefore, if the flow value is equal to the number of games remaining, then team A has a chance to win, otherwise team A has no chance of winning.

3. Open Pit Mining Problem

Open pit mines are used when deposits of valuable minerals are buried near the surface but the earth covering the valuable mineral is structurally unsuitable for tunneling. We need to dig through the covering earth to get to the mineral without collapsing the mine. With set of nodes of the covering earth and minerals defined as nodes V, E= (u, v) where u is the covering earth, we can illustrate an open pit mine in a direct graph G= (V, E). An example open pit mine is shown in the graph below. The goal is to find which nodes should be removed to gain the most profit.



3.1. Formal problem definition:

Input: directed graph: G=(V, E) where

- V: set of nodes of the covering earth
- E: $\{(u,v)|u\ must\ be\ removed\ before\ v\}$
- A function $W(v)$ equals the value of node v

Output: The set of nodes that should be removed to gain the most profit.

Structural Property: We define an *initial set* to be a set of nodes that has no additional incoming edges where the tail of the edge is not in the initial set. For example: node A, D and G is an initial set since there is no incoming edge; node D and G is not an initial set since there is an edge from A to G. Also, there should be an edge with the head at all nodes being removed (with the other end of the edge at another node of the initial set), except for the nodes residing at the topmost level.

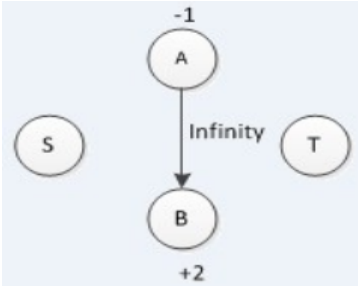
The output for this problem is to find the most profitable initial set.

3.2 Formulate as a Network Flow Problem

The Idea: Convert the problem to a network flow so that any finite capacity cut corresponds to an initial set (We also claim that any cut with infinite capacity corresponds to nodes not in the initial set), and a minimum capacity cut corresponds to a maximum profit initial set.

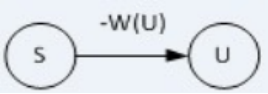
One side of the cut represents the nodes that should be removed (T side) and another side represents nodes that should not be removed (S side). If any nodes with negative values are on T side, we increase the capacity of the cut by that value. The steps shown below demonstrate how we can set up the flow network of an example graph.

1. Put infinite capacity to all existing edges:

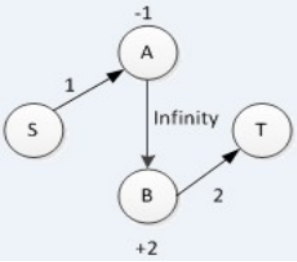


2. If $W(u)$ is positive, add to T side:

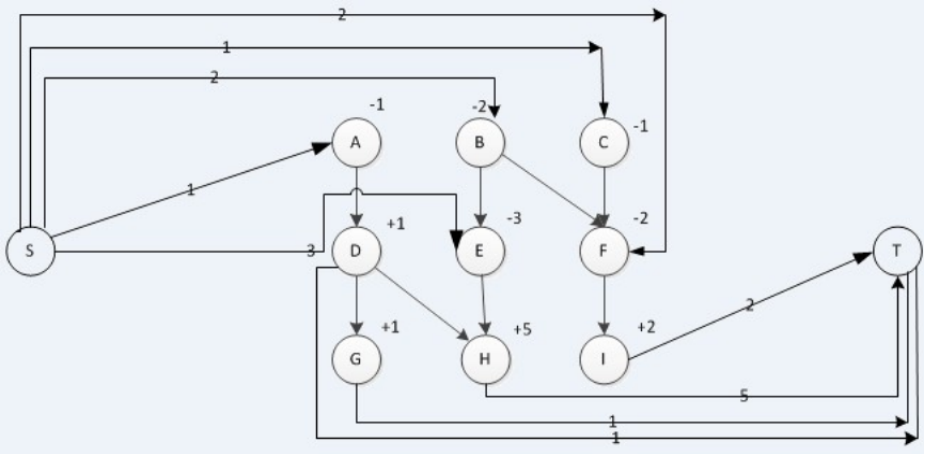
3. If $W(u)$ is negative, add to S side:



Graph of final model:



Follow this technique, our flow network of this problem is shown in the figure below.



For any initial set U, the capacity of the corresponding cut can be represented as:

$$\big((V-U)\cup\{s\}, U\cup\{t\}\big)$$

Suppose that $S=(V-U)\cup\{s\}$ and $T=U\cup\{t\}$. The capacity of this cut is,

$$C(S,T)=\sum_{u\notin U, w\,(u)>0} w\,(u)+\sum_{v\in U, w\,(v)<0} -w\,(v)$$

Add sum of all negative average weight values to both sides of equation.

$$C(S,T)+\sum_{v\in V\ w\,(v)<0} w\,(v)=\sum_{u\notin U\ w\,(u)>0} w\,(u)+\sum_{v\notin U, w\,(v)<0} w\,(v)$$

where $\sum_{v\in V\ w\,(v)<0} w\,(v)$ represents the value for nodes independent of U, and $\sum_{u\notin U\ w\,(u)>0} w\,(u)+\sum_{v\notin U, w\,(v)<0} w\,(v)$ represents profit of nodes not removed. The maximum profit can be found by minimizing the profit of nodes not removed.

4. Duality for linear programming

Each linear programing problem can be converted into a dual problem, which provides the upper bound to the optimal value of the original problem. By following the similar argument to the maximum flow minimum cut theorem, we can find that every linear maximization problem has a dual minimization problem.

Consider the chocolate problem introduced before:

$$\text{Maximize: } x_1+6x_2$$

Subject to the following constrains:

$$x_1 \leq 200$$

$$x_2 \leq 300$$

$$x_1 + x_2 \leq 400$$

$$x_1, x_2 \geq 0$$

Solution to this problem could be $x_1 = 100$ and $x_2 = 300$ and the profit value is 1900. We can verify this is a correct solution, but how do we verify that it is an optimal solution?

We can verify by the duality of linear programming:

1. Introduce a multiplier (y_1, y_2, y_3) to each constraint:
$$y_1 * x_1 \leq 200 * y_1$$
$$y_2 * x_2 \leq 300 * y_2$$
$$y_3 * (x_1 + x_2) \leq 400 * y_3$$
Note that these multipliers must be non-negative, otherwise the inequality sign will be flipped from $\leq$ to $\geq$.
2. Add these constrains up, we have:
$$(y_1 + y_3)x_1 + (y_2 + y_3)x_2 \leq 200y_1 + 300y_2 + 400y_3$$

We want the left-side to look like the original objective function and the right side to be the upper bound on the optimal solution. We can see this can only be true if:

$$y_1, y_2, y_3 \geq 0$$

$$y_1 + y_3 \geq 1$$

$$y_2 + y_3 \geq 6$$

Thus, the dual problem can be formulated as:

$$\text{Minimize: } 200y_1 + 300y_2 + 400y_3$$

$$y_1, y_2, y_3 \geq 0$$

$$y_1 + y_3 \geq 1$$

$$y_2 + y_3 \geq 6$$

Theorem: If the original linear problem has a bounded optimum, then so do its dual, and the two optimum values coincide.

Any solution which satisfies the dual LP (the above constraints for y_1, y_2, y_3) is an upper bound for the original LP. However, if we can find a solution for the dual LP and the original LP with values that are equal, then that would be the most optimal solution.

5. Conclusion

In this lecture, two problems were discussed, the pennant race problem and the open pit mining problem. Both the pennant race problem and the open pit mining problem can be considered as network flow problems, and can be solved by using the maximum flow minimum cut theorem. At the end of the lecture, we introduced duality of linear programming, which claims that every LP problem can be converted to a dual problem and both the original and the dual problem have the same optimal solution.