CPSC 500 Fundamentals of Algorithm Design and Analysis

Lecturer: Prof. Will Evans **Lecture Date: Nov. 13, 2013** **Scribe: Ben Zhu**

1. Overview

In this lecture, we continued on our discussion about *max flow*/min cut problem. We began our discussion from *Pennant Race Problem* which was introduced at the end of last lecture. Then we discussed another problem, *Open Pit Mining Problem*. Afterwards we started to discuss *Linear Programming Duality* with how to verify our solutions to linear programming problems.

2. Pennant Race Problem

Input

A list of teams T_1, T_2, …, T_n with win/less record for each team
A special team A
A list of games remaining to be played

Output

Is it possible for team A to end the season winning the most (at least as many) as any team?

Example

Team	Wins
A	3
T_1	4
T_2	6
T_3	5
T_4	4

Games to play:

(A, T_1) (A, T_3) (A, T_4) (T_1, T_2) (T_1, T_3) (T_2, T_3) (T_2, T_4)

Solution

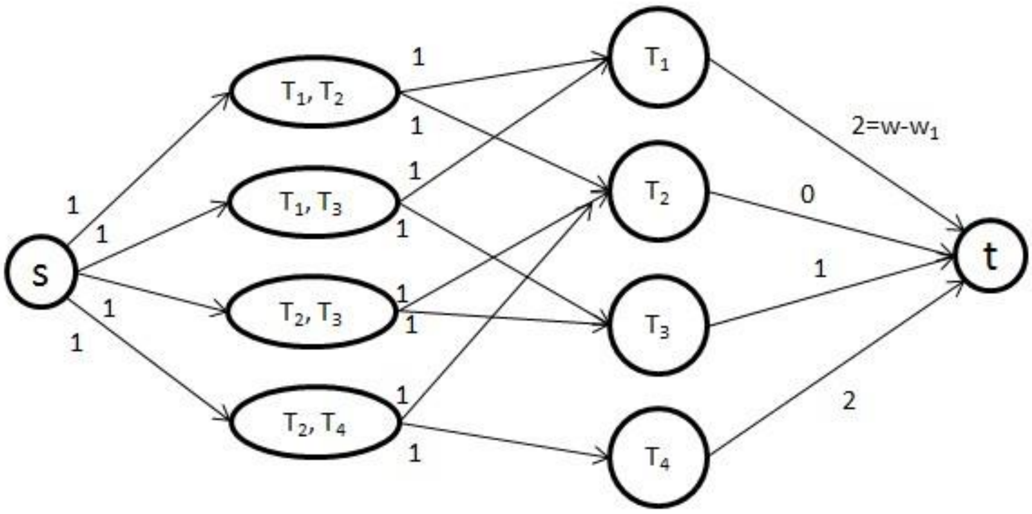
Step 1: Assume A wins all games, let w=6 and we will have

Team	Wins
A	6=w
T_1	4=w_1
T_2	6=w_2
T_3	5=w_3
T_4	4=w_4

Step 1.5: Check that $w \geq w_i, for\ all\ T_i$

If not, A has no hope.

Step 2: Calculate max flow using the following flow network



If flow value = # games remaining then A has chance, otherwise A has no hope.

3. Open Pit Mining Problem

Input

Directed graph G = (V, E) where

V = set of cubes of earth

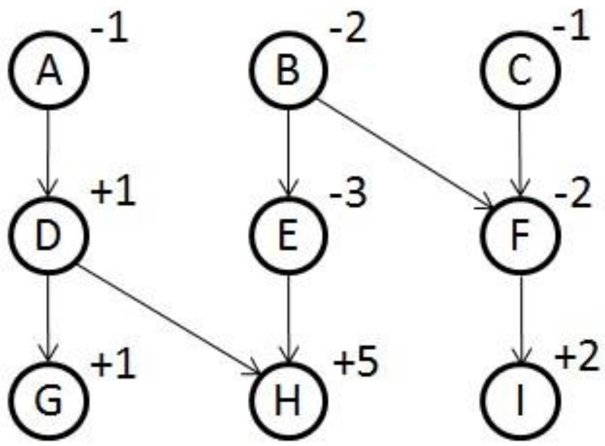
E = {(u, v) | u must be removed before v}

And a function w(v) = value of cube v

Output

Which cubes should be removed?

Example



We can consider this problem as to find the most profitable initial set, where an initial set is a set of vertices that has no incoming edges.

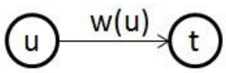
Idea

Convert the problem to a network flow so that any finite capacity cut corresponds to an initial set, and a minimum capacity cut corresponds to a max profit initial set.

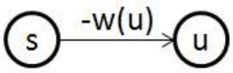
Solution

1. Make all existing edges have ∞ capacity.
   So any cut (S, T) where T is not an initial set has ∞ capacity.

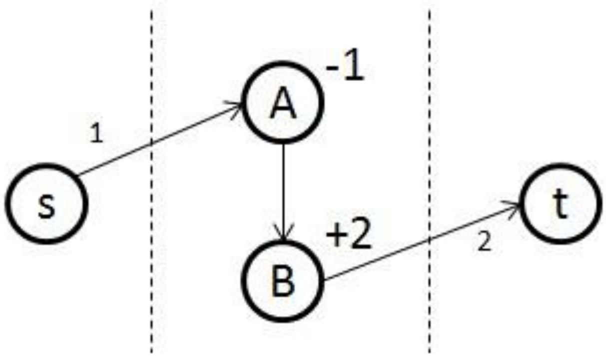
2. If w(u) is positive, add edge

   If w(u) is negative, add edge

For example,



3. For any initial set U, the capacity of the corresponding cut $(S,T), i.e. \left((V-U) \cup \{S\},\ U \cup \{t\}\right)$

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

4. Add sum of all negative weight values to both sides of equality, we will have

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

where $\sum_{v\in U, w(v)<0} w(v)$ is independent of U and $\sum_{u!\in U, w(u)>0} w(u) + \sum_{v!\in U, w(v)<0} w(v)$ is profit of cubes not removed.

So minimizing $c(S,T)$ is the same as maximizing profit of cubes that are removed.

4. Linear Programming Duality

Example

We want to compute: $\max\ x_1 + 6x_2$

Add we have these following constraints:

$x_1 \le 200$
$x_2 \le 300$
$x_1 + x_2 \le 400$
$x_1, x_2 \ge 0$

We may come up with a solution that $x_1 = 100, x_2 = 300, value = 1900$.

Then how should we verify our solution?

Intuitively we would think of using the constraints, like
$(5)\ x_2 \le 1500$
$(1)\ x_1 + x_2 \le 400$

Then we will have $x_1 + 6x_2 \le 1900$

Multipliers

Here we introduce the multipliers as follows,

$$
\begin{array}{llll}
y_1 & x_1 & & \le\ 200 \\
y_2 & & x_2 & \le\ 300 \\
y_3 & x_1 & x_2 & \le\ 400
\end{array}
$$

Then we have $(y_1 + y_3)x_1 + (y_2 + y_3)x_2 \le 200y_1 + 300y_2 + 400y_3\ , (*)$
and constraints for the multipliers as follows,

$y_1, y_2, y_3 \ge 0$
$y_1 + y_3 \ge 1$
$y_2 + y_3 \ge 6$

If the constraints are true, then the right side of $(*)$, $200y_1 + 300y_2 + 400y_3$, is the upper bound of $x_1 + 6x_2$.

So we want to choose y_1, y_2, y_3 to minimize $200y_1 + 300y_2 + 400y_3$ and we get another linear programming problem.

5. Conclusion

In this lecture, further discussed the *max flow/min cut problem* which was introduced during last lecture. We began our discussion from *Pennant Race Problem* and moved forward to another example, *Open Pit Mining Problem*. Afterwards, we started to discuss *Linear Programming Duality* with how to verify our solutions. We will have one more lecture to cover linear programming.

References

[1] Wikipedia, “Maximum flow problem,” http://en.wikipedia.org/wiki/Max_flow
[2] Wikipedia, “Max-flow min-cut theorem,”
http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem