CPSC 500
November 13 2013

Scribe for: Dr. Will Evans
Scribe by: Imran Ahmed
aimran@cs.ubc.ca

November 18, 2013

Introduction

Continuing from the previous class where we saw how the bipartite matching problem can be reduced to a network flow problem, we investigate two more increasingly complex examples, the pennant race problem and the open pit mining problem.

Not seemingly network flows, yet, these problems can be solved by reducing them to network flows, and solved accordingly.

We follow up with an introduction to duality in linear programming.

Examples: Applications of Network Flow (Continued from last class)

Pennant Race Problem

Inputs:

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

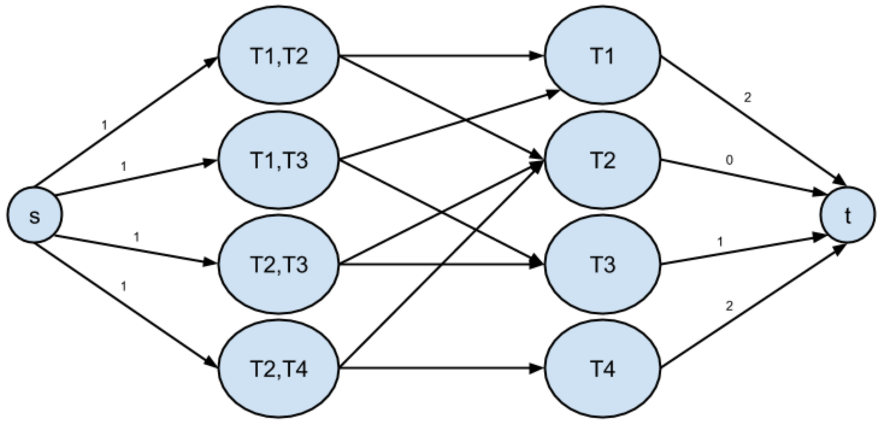
Output:

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

Team Name	# of Wins	
A	3	
T_1	4	w_1
T_2	6	w_2
T_3	5	w_3
T_4	4	w_4

Games to play:
$(A, T_1)(A, T_3)(A, T_4)$
$(T_1, T_2)(T_1, T_3)$
$(T_2, T_3)(T_2, T_4)$

Assume A wins all games
Check that $w \geq w_i$ for all T_i (otherwise team A has no hope of getting a pennant)

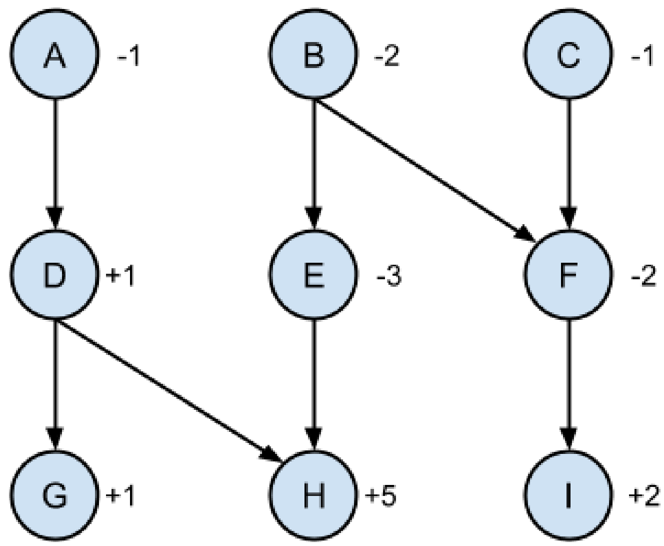


Calculate max flow:
if flow value = number of games remaining, A has chance, otherwise no hope.

Open Pit Mining Problem

Now, we look at the open pit mining problem. This problem deals with digging up cubes of earth that contain the most valuable ores from, as the name suggests, an open pit mine.
The problem here is: to dig up cubes of earth that are deeper in the ground, the top layers have to be removed so as not to make the mine collapse.

This problem can be abstracted to a directed acyclic graph with node values. Nodes in the graph can be used to represent the cubes of earth, with the value of the node indicating the net profit for extracting that node. The graph also depicts the dependency that extracting cubes of earth deeper in the ground require extraction of cubes of earth or above them.



Again, this problem can be solved by reducing it to a network flow problem.

Input:
Directed graph $G = (V, E)$ where, V = set of cubes of earth, and
$E = \{(u, v) \mid u \text{ must be removed before } v\}$

Output:
An *initial set* is a set of vertices that has no incoming edge.
Find the most profitable *initial set*

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.

Make all existing edges have ∞ capacity
(so any $cut(S, T)$ where T is not an *initial set* has ∞ capacity)

if $w(u)$ is positive, then add edge u to t
if $w(u)$ is negative, then add edge s to u

For any *initial set* u, the capacity of the corresponding

$$cut((v - u) \cup \{s\}, u \cup \{t\})$$

where S is $((v - u) \cup \{s\})$, and T is $(u \cup \{t\})$

$$c(S, T) = \sum_{u \notin U} w(u) + \sum_{v \in U} -w(v)$$

Add sum of all negative weight values to both sides

$$c(S, T) + \sum_{v \in V} w(v) = \sum_{u \notin U} w(u) + \sum_{v \in U} w(v)$$

where, $\sum_{v \in V} w(v)$ is independent of U

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

This, relationship of maximization and minimization just shows a peek into what duality is in linear programming, which is explored more in the following section.

Duality in Linear Programming (To be continued)

Coming back to the original problem of the chocolate manufacturer, we need to

$$\text{maximize } x_1 + 6x_2$$

given the constraints,

$$\begin{aligned} x_1 &\leq 200 \\ x_2 &\leq 300 \\ x_1 + x_2 &\leq 400 \\ x_1, x_2 &\geq 0 \end{aligned}$$

We know that the solution to the problem is at $x_1 = 100$ and $x_2 = 300$,
giving a maximum profit of 1900.

Verifying that the correctness of the solution is another matter.

To proceed to verify the correctness, we can choose multiplier for the constraints such that:

$$\begin{array}{ccc} \text{Multiplier} & \text{Constraint} & \\ 5\times & x_2 \leq 300 & 5x_2 \leq 1500 \\ 1\times & x_1 + x_2 \leq 400 & x_1 + x_2 \leq 400 \\ \hline & & x_1 + 6x_2 \leq 1900 \end{array}$$

The goal is to find the proper multipliers to evaluate this.

Now, if the variables y_i are chosen as multipliers for each of the constraints, we get the following:

$$\begin{array}{cc} \text{Multipliers} & \\ y_1 & x_1 \leq 200 \\ y_2 & x_2 \leq 300 \\ y_3 & x_1 + x_2 \leq 400 \\ \hline & (y_1 + y_3)x_1 + (y_2 + y_3)x_2 \leq 200y_1 + 300y_2 + 400y_3 \end{array}$$

If the constraints,

$$\begin{aligned} y_1, y_2, y_3 &\geq 0 \\ y_1 + y_3 &\geq 1 \\ y_2 + y_3 &\geq 6 \end{aligned}$$

are true, then

$$200y_1 + 300y_2 + 400y_3$$

upper bounds $x_1 + 6x_2$

We want to choose y_1, y_2, y_3 such that $200y_1 + 300y_2 + 400y_3$ is minimized.

(to be continued next class ...)

Conclusion

We looked at two more problems that could be solved using network flow, the pennant race problem and the open pit mining problem. We noticed that these problems can be reduced to a network flow problem, and solved accordingly.

Both of these solutions used the max flow/min cut property of network flows, which hints that there is an inherent relationship between a maximization problem and a minimization problem in linear programming.

Afterwards, we begin to formally explore this duality in linear programming with the aid of a simple example from a previous class. By building upon that example, we try to point out the dual aspect of how a maximization problem can be converted to a minimization problem.