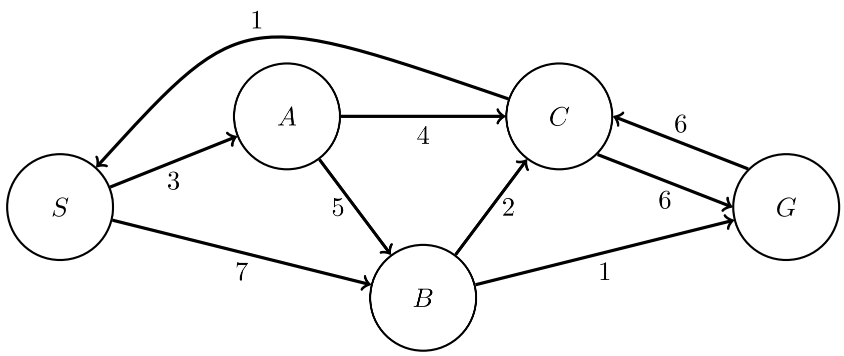
Learning Objectives

- To convert a search problem into an IP problem

Email or return the completed activity to Prof. Rosenthal for 1 point per full page completed (3 total) by the designated return time. Returning the assignment when not in class result in an academic integrity violation. Email must be titled “15281 Lecture 9 Activity”.

Q1. Search as IP

In this problem we are going to explore how to formulate search problems as Integer Programming problems. For the sake of simplicity we will be considering the weighted directed graph below (S is the start, and G is the goal):



We can derive a representation for any path in a graph by considering variables (called indicator variables) to represent edges in the graph each with domain {0,1}; value 0 if the edge is not in the path and 1 if the edge is in the path. Specifically, define a binary variable $x_{Y\to Z}$ if there is an edge $Y \to Z$ in the graph.

For example, the path $S \to A \to C \to G$ can also be seen as the set of edges $\{S \to A, A \to C, C \to G\}$. We can then define $x_{S\to A}$ to be the indicator variable for whether the edge $S \to A$ is on the path, $x_{S\to B}$ to be the indicator variable for whether the edge $S \to B$ is on the path, and so on. Using this binary representation, $S \to A \to C \to G$ can be represented as:

$$\left(x_{S\to A} = 1 \quad x_{S\to B} = 0 \quad x_{A\to B} = 0 \quad x_{A\to C} = 1 \quad x_{B\to C} = 0 \quad x_{B\to G} = 0 \quad x_{C\to S} = 0 \quad x_{C\to G} = 1 \quad x_{G\to C} = 0\right)$$

If we fix the order of the indicator variables in the order above, the path can be represented as a 9−tuple: (1, 0, 0, 1, 0, 0, 0, 1, 0)

(a) Answer the following questions below about the representation.

(i) Write the 9−tuple binary representation for the path $S \to A \to B \to C \to G$

Answer:
$(1, 0, 1, 0, 1, 0, 0, 1, 0)$

(ii) Write the 9−tuple binary representation for the path $A \to C \to S \to B$

Answer:
$(0, 1, 0, 1, 0, 0, 1, 0, 0)$

(iii) Write the path that corresponds to (0, 0, 1, 0, 1, 0, 0, 0, 0)

Answer:
$A \to B \to C$

Note with our binary representation, it is possible to make "paths" that do not start from S and end at G, and some of the 9−tuples are not even valid paths at all. Consider the constraints on valid paths in this representation.

From the binary representation itself, it is clear that we need to constrain the indicator variables to be an integer between 0 and 1 inclusive. To find valid paths from the start to goal, we need to start by imposing some additional constraints on the 9−tuples.

(b) Write the constraint(s) (in **inequality form**) that a valid path in the specific graph above must start at S. (Hint: What must be true about the values or sums of values of the indicator variables involving S defined above? Remember that constraints must be linear sums of variables.)

Answer:
Exactly one of the two outbound edges must be 1 and none of the inbound edges.

$$x_{S\to A} + x_{S\to B} \leq 1$$
$$-x_{S\to A} - x_{S\to B} \leq -1$$
$$x_{C\to S} \leq 0$$
$$-x_{C\to S} \leq 0$$

(c) Similarly, write the constraint(s) that a valid path must end at G (in inequality form).

Answer:
Exactly one of the two outbound edges must be 1 and none of the inbound edges.

$$x_{C\to G} + x_{B\to G} \leq 1$$
$$-x_{C\to G} - x_{B\to G} \leq -1$$
$$x_{G\to C} \leq 0$$
$$-x_{G\to C} \leq 0$$

(d) Write a 9−tuple (binary representation) that satisfies the constraints in **(i)** and **(ii)** but does **not** represent a valid path from S to G

Answer:
more than one answer, one possibility is $(1, 0, 0, 0, 0, 0, 0, 1, 0)$

Let us define a non-terminal node as a node that is neither the defined start node (S) nor the goal node (G). Note that the constraints in parts **(b/c)** do **not** guarantee that all other nodes in the path are non-terminal.

(e) We want to ensure that the path only passes through each non-terminal node (e.g., B) at most once. Write the constraint(s) that node B does not appear more than once on the path (inequality form).

Answer:
We need to write that the in-edges is at most 1 and the out-edges is at most 1. Both are not required when you consider the answer for f.

$$\begin{aligned} & x_{S\to B} + x_{A\to B} \leq 1 \quad \text{and correct f} \\ \text{OR} \quad & x_{B\to C} + x_{B\to G} \leq 1 \quad \text{and correct f} \\ \text{OR} \quad & \text{both} \end{aligned}$$

(f) Finally, to ensure B is non-terminal, we have to make sure that if there is an edge to B in a path, then there should also be an edge from B to some other node. If there is an edge to B but not from B, then the path has a dead-end at B. Similarly, if there is an edge from B to another node but no edge to B, then B is a starting node but not S. Write down the corresponding constraint(s) about B's non-terminal position (inequality form).

Answer:
From part e, we know that the in edges is at most one and the same for out edges. Now we need to specify that if there is an in edge, there is also an out edge. We do this by subtracting in minus out and requiring that it equal 0.

$$x_{S\to B} + x_{A\to B} - x_{B\to C} - x_{B\to G} \leq 0$$
$$-x_{S\to B} - x_{A\to B} + x_{B\to C} + x_{B\to G} \leq 0$$

(g) Are the constraints in part **(e)** and **(f** valid for S and G? Explain why.

Answer:
The first "pass at most once" (part **ci**) is valid because the start and the end have exactly one edge connected to them, but the second "non-terminal" (part **cii**) is not since the in-degree is different from the out-degree for terminal nodes.

(h) We have all the constraints to ensure a valid path. Write the objective function for the IP search problem representing the graph above. (Hint: the goal of solving this IP problem is to minimize the objective function that corresponds to the search objective.)

Answer:
$3x_{S\to A} + 7x_{S\to B} + 5x_{A\to B} + 4x_{A\to C} + 2x_{B\to C} + 1x_{B\to G} + 1x_{C\to S} + 6x_{C\to G} + 6x_{G\to C}$

(i) Define a new search algorithm to be: formulate the search as an IP problem, then run an IP solver and return the solution as a path. Is this new search algorithm:

(i) Complete? ● Yes ○ No

(ii) Optimal? ● Yes ○ No