COMP5211 2019 Fall Semester Assignment #2 - The Written Part

Date assigned: Monday, Oct 14.

Due time: 23:59 on Saturday, Oct 26.

How to submit it: Submit your written answers as a pdf file on canvas.ust.hk.

Penalties on late papers: 20% off each day (anytime after the due time is considered late by one day)

Problem 1 (10%). Given a graph and a node in it as the starting point, the traveling salesman problem (TSP) is about finding a least cost path that starts and ends at the starting node, and goes through each other node in the graph once and exactly once. The edges of the graph are labeled by a number representing the cost of traveling between the connecting two nodes. Formulate this problem as a search problem by specifying the states, possible initial states, goal test, operators, and operator costs.

Problem 2. (10%) (Adapted from Russell and Norvig) Consider the problem of *constructing* crossword puzzles: fitting words into a grid of intersecting rows and columns of squares. Assume that a list of words (i.e. dictionary) is provided, and that the task is to fill in the rows and columns with words from this list so that if a row intersects with a column, their intersecting square has the same letter. Formulate this problem as an assignment (constraint satisfaction) problem by specifying the variables, their domains of possible values, and the constraints.

Problem 3. (25%) Consider the following two investment decisions: buy a certified deposit (CD) which will pay you a 10% annual interest rate, and buy some stocks which will either give you a 30% annual return (with 0.7 probability) or incur a 10% loss for you (with 0.3 probability). In terms of expected utility, you should of course buy stocks. However, suppose you are conservative and cannot tolerate any loss of your principal. In this case, you have no choice but to deposit your money into a CD. Now consider this problem for a 3 years time span. You start with 1 unit of money. Each year you can choose only one way to invest your entire fund (meaning you are not allowed to diversify, like 50% CD, 50% stocks). You definitely don't want to lose any money at the end of the 3 years, but you are okay with a "temporary" loss during the time (for example, it is okay for you to lose some money in the first year as long as you can make it up later). Formalize this problem as a Markov decision process and compute its optimal policy.

Hints A state needs to contain information about how many money you have in this state. In other words, a state needs to encode the history: which actions you have done so far and what their outcomes are if the actions are non-deterministic.

Problem 4. (15%) Consider the following search problem:

- state space: $\{S, A, B, G\}$;
- operators: O_1 maps S to A with cost 4, O_2 maps S to B with cost 2, O_3 maps B to

A with cost 1, and O_4 maps from A to G with cost 5;

- initial state S;
- goal state G;
- heuristic function h: $h(S) = 7$, $h(A) = 1$, $h(B) = 6$, and $h(G) = 0$.

Find a solution using A^* search by tree. Number the nodes according to their order of expansion, and for each node give its $f(n) = g(n) + h(n)$ value.

Problem 5. (10%) Perform minimax with perfect decision on the tree given in the following figure (the leaves are terminal nodes, and the numbers next to them are their utility values).

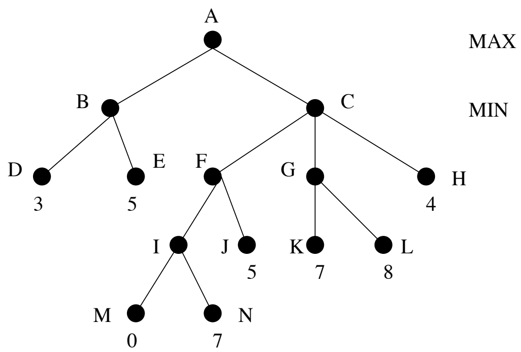


Figure 1: A minimax search tree

Problem 6. (10%) Perform a left-to-right alpha-beta prune on the tree in the above exercise. Perform a right-to-left prune on the same tree. Left-to-right or right-to-left means the order by which the leaf nodes are generated. So for left-to-right, the first leaf node considered is D, followed by E, followed by M and so on.

COMP5211 2019 Fall Semester Assignment #2
WANG Sukai 20528424

P1. states: All (node list) nodes have been going through where no two nodes are same.

possible initial states: Any start point.

goal test: The number of nodes been through equals the number of all nodes.

operators: add a new node which connected to the last node in node list.

operator cost: the edge value between new added node with last node in node list.

P2. Variables: $w_1, w_2 \cdots, w_n$, the word chosen in blank place $1, 2, \cdots n$.

domains: The same for all variables, $\{1, 2 \cdots, n\}$

constraints: $W_1 \neq W_2 \neq \cdots \neq W_n$

intersection square position in two intersected words have same letter.

P3. Formulate this problem as a MDP:

States: { year (y), decisions ($\{\cdots d_i\}$), ~~total~~ money (m))

Start state: $\{0, \phi, 1\}$

End (s) iff year ≥ 3 or $m \cdot 1.1^{3-y} < 1$ (which means you might lose money at the end)

action: { next year stock, next year CD }

Transitions: $T(\{y, \{\cdots d_i\}, m\}, CD, \{y+1, \{\cdots d_i, CD\}, 1.1m\}) = 1$

$T(\{y, \{\cdots d_i, m\}, stock, \{y+1, \{\cdots d_i, stock\}, 1.3m\}) = 0.7$

$T(\{y, \{\cdots d_i, m\}, stock, \{y+1, \{\cdots d_i, stock\}, 0.9m\}) = 0.3$

Rewards: $\text{Reward}(\{y, \{\dots d_i\}, m\}, X, \{y+1, \{\dots d_i, X\}, m'\}) = m' - m$

Now we try to find the optimal result.

Use policy iteration:

initial π to stock, stock, stock.

when we only consider the first end condition,

$$Q_\pi(s,a) = \sum_{s'} T(s,a,s')\left[\text{Reward}(s,a,s') + V_\pi(s')\right]$$

$$V_\pi(CD) = 1 \times (0.1m + V_\pi(\dots, 1.1m))$$

$$V_\pi(\text{stock}) = 0.7 \times (0.3m + V_\pi(\dots, 1.3m)) + 0.3 \times (-0.1m + V_\pi(\dots, 0.9m))$$

$$= 0.11m + 0.7\,V_\pi(\dots, 1.3m) + 0.3\,V_\pi(\dots, 0.9m)$$

So $V_\pi(\text{stock}) > V_\pi(CD)$

We can see Stock is always better then CD.

Then if we consider the second end condition, we must make sure the final total money is larger than 1.

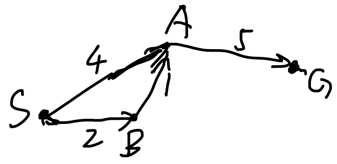
So if we choose stock in first year, and failed, then we need two next years to choose CD to compensate. if we choose second year stock and fail then third year need choose CD.

optimal policy: First time we choose stock, if success, then next two are also stock. if fail, then next two are CD.

Initialization

P4. Step 1: $f(A) = 4 + 1 = 5$

$f(B) = 2 + 6 = 8$



Since $f(A) < f(B)$, add A to CLOSED.

Next connected node is only G.

So by A^* algorithm, $S \to A \to G$ is the final path.

$f(S) = 7$ $f(A) = 5$ $f(G) = 9$

P5. forth is min: $e(I) = \min\{0, 7\} = 0$

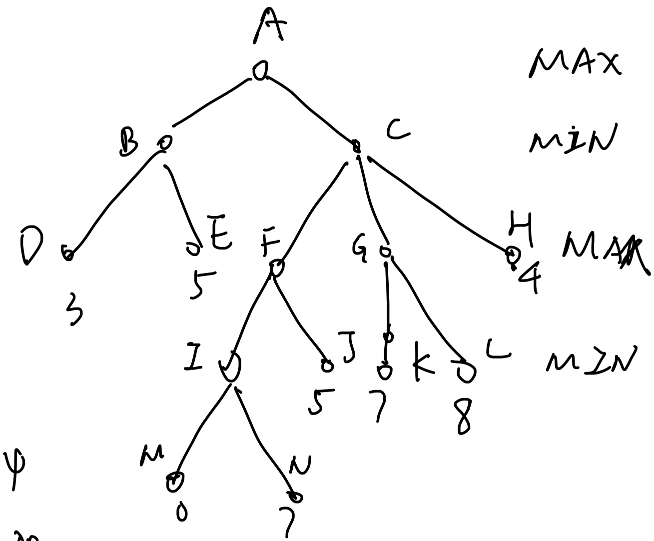
third is max: $e(F) = \max\{0, 5\} = 5$

$e(G) = \max\{7, 8\} = 8$

Second is min: $e(B) = \min\{3, 5\} = 3$

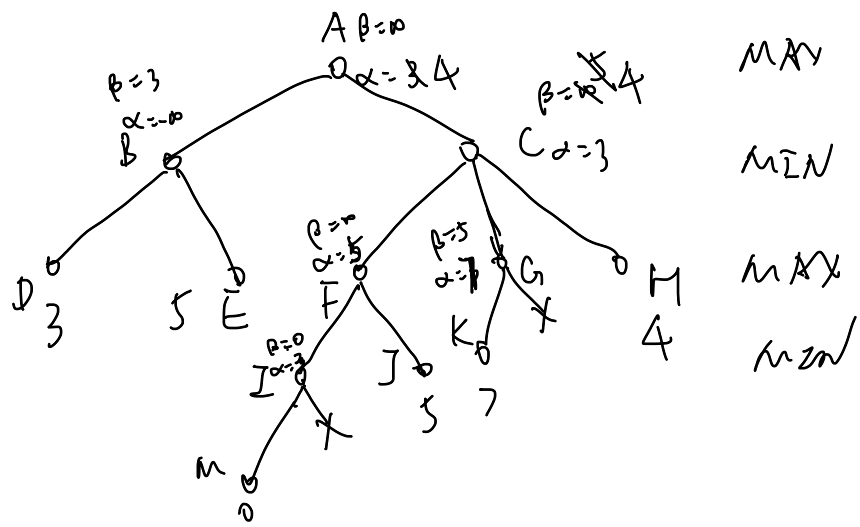
$e(C) = \min\{5, 8, 4\} = 4$

first is max: $e(A) = \max\{3, 4\} = 4$



So the final step: $A \to C$ (max), $C \to H$ (min)

P6. left-to-right



right-to-left