20667036
Tomas Sousa Pereira

Assignment 2

CSIT6000F
November, 2019

Problem 1

To simplicity lets assume that our points are cities

- States: cities
- Possible Initial States: any city on the graph
- Goal Test: traveling salesman visit every city and start and end in the same city
- Operators: chose one connected unvisited city to go
- Operators cost: the cost of travelling from one city to another

Problem 2

- Variables: words to fit in the blank spaces
- Variables values domains: words present in dictionary
- Constraints: when row intersects with a column the letter has to match

Problem 3

- State: (year, amount)
- Initial State: (0,1)
- Final State: (2,x) the last choice on year 2, x is the amount at that year
- Actions: {CD, Stocks}
- Transitions:

$$T(s, CD, s') = \max \begin{cases} 1 & , s'_{amount} = 1.1 s_{amount} \\ 0 & , otherwise \end{cases} \tag{1}$$

$$T(s, Stocks, s') = \max \begin{cases} 0.7 & , s'_{amount} = 1.3 s_{amount} \\ 0.3 & , s'_{amount} = 0.9 s_{amount} \\ 0 & , otherwise \end{cases} \tag{2}$$

In the figure below we can observe the possible decisions and its possible states. The node in blue is the same the one in green therefore it wasn't expanded. The best policy is to buy always with stocks.

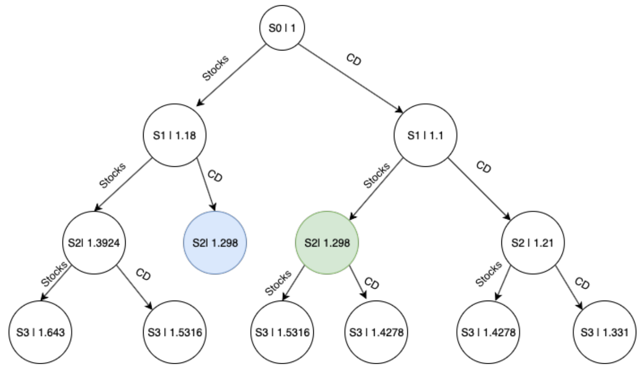

Figure 1: Markov decision

Problem 4

We will use a priority queue to order by heuristic value.

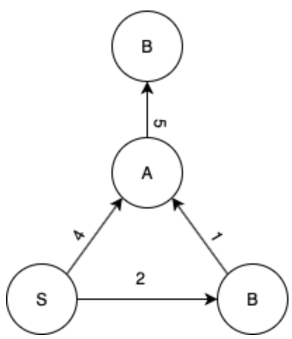

Figure 2: Tree

s	h(s)
S	7
A	1
B	6
G	0

Figure 3: Heuristic state-values

Iteration0
Present Node: S Successors: A, B f(A) = 4 + 1 = 4 f(B) = 2 + 6 = 8
PriorityQueue: [{[S,A],5},{[S,B],8}]

Iteration1
Present Node: A Successors: G
f(G) = 9 + 0 = 9
PriorityQueue: [{[S,B],8},{[S,A,G],9}]

Iteration2
Present Node: B Successors: A
f(A) = (2+1) + 1 = 4
PriorityQueue: [{[S,B,A],4},{[S,A,G],9}]

Iteration3
Present Node: A Successors: G

f(A) = (2+1+5) + 0 = 8
PriorityQueue: [{[S,B,A,G],8},{[S,A,G],9}]

The optimal path is the following: S $\rightarrow B \rightarrow A \rightarrow G$

Problem 5

Observing the Figure below we can assume that the node H is the optimal result.

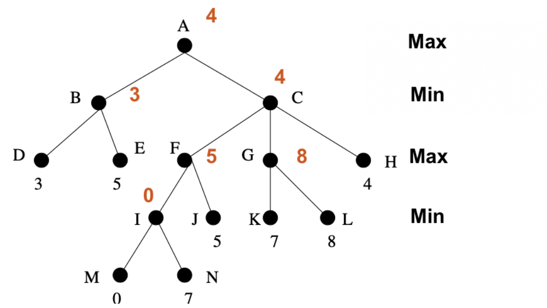

Figure 4: Min Max Search

Problem 6

Node H is the optimal result. For Left-Right we prune the nodes N and L. For Righ-Right we prune the nodes N and K.

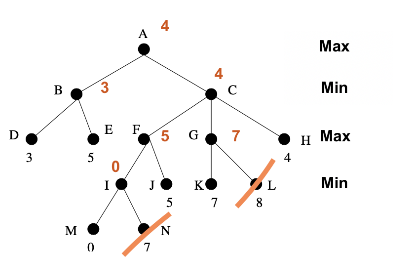

Figure 5: Left Right α β prunning

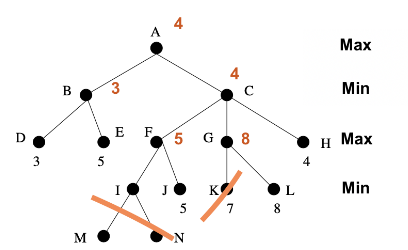

Figure 6: Right Left α β prunning