Value Iteration(Part-3)

This report explains the use Linear Programming to solve the Value Iteration problem given in the assignment.

Q1) Explain procedure of making A matrix:
Ans) *The dimension of A matrix is s*x, where s are all possible states and x is s*a i.e. sum of all possible actions for each states(a is all possible valid actions for a particular state). For each state s, the A matrix contains the in-going and out-going probabilities of a valid action for a corresponding state.*

*Total states can be found by combination of all state parameters of the form (Position,Materials,Arrows, Monster State, Health), i.e. 5*3*4*2*5 = 600 total states.*

Total state action pairs can be found by iterating through all the states, and finding the number of valid actions that can be taken from the set of 10 actions. (Up,Down,Left,Right,Stay,Shoot,Hit,Craft,Gather,None).In our case we got this number to be 1936.

*To find the 600*1936 dimensional matrix A, we iterate through all the states and for each state we iterate through all the actions and chooses the valid action. For this valid action we update the cuurret_state, action pair by 1.0 and subtracts the probability of getting state S from current State using action A from the S,A pair.*

Q2)Explain procedure of finding the policy and analyze the results:
Ans) *Once we have the Matrix A, vectors alpha and R, we can use cvxpy library to find the solution for x.*

We first make 1936 by 1 variable array , add constraints of
Ax=alpha
And
x>=0
And then maximize Rx by creating the cp.Problem and then solving it using problem.solve().

To find the policy, now that we have the x array, we find the maximum element of x corresponding to an action for a State s. Thus, for each state S, we get an action A and hence we got the policy.

Q3)Can there be multiple policies? Why? What changes can you make in your code to generate another policy?

Ans) *Yes, multiple policies are possible. This is because in the x array we got from solver may have the expected value of an action same for 2 or more then 2 actions for a particular state, giving us choice of choosing best action for a state as there will be more then 1 option now.*

To generate multiple policies, the code where we find the maximum expected value of an action for a state is found, there can be more than one maximum values. Thus, the code can be tweaked to consider all the policies, maximum can be selected randomly to get a random policy.