

# Assignment 2

---

## Machine, Data and Learning

---

Tejasvi Chebrolu, Ruthvik Kodati

2019114005, 2019101035

---

### Task 1:

Commenting on the trace files -

For the step cost (-20) to converge, it took 124 iterations.

#### **Rate of Convergence**

During the initial stages of the algorithm, the convergence rate increased rapidly which implies that the algorithm began to diverge slightly. After some time, the rate became constant. Finally, it exponentially decreased during the end of the algorithm which means that the algorithm adequately converged.

#### **Commenting on different squares IJ is present at :**

##### **1. South**

In the south state, if MM is in the Ready state, IJ chooses an action in which he will remain in the south state. Depending on the number of materials he has, he either GATHERS or STAYS. However, if MM is in a Dormant state and IJ has enough arrows, IJ goes up to the center state.

##### **2. East**

In the east state, IJ always chooses to attack and will finish the game. If MM's health is 100, IJ always HITS. He SHOOTS if he has arrows and if MM is in the Dormant state with less than 100 health.

##### **3. Center**

If MM is in a Ready state, IJ either goes UP or DOWN depending on MM's health. However, if MM is in a Dormant state while IJ is in the center state, IJ always goes RIGHT to the east state except when he has no material or arrows.

##### **4. North**

If MM is in a Dormant state, IJ prefers to go DOWN to the center state depending on the number of materials and arrows he has as well as MM's health. If MM is in a Ready state and IJ is in the north state, IJ prefers to STAY but he performs CRAFT if his material is not full.

##### **5. West**

If MM is in a Dormant state, IJ almost always goes RIGHT to the center state. However, if MM's health is 25 and IJ has arrows, he chooses to SHOOT. If MM is in a Ready state and IJ has material, he chooses to go RIGHT to CRAFT them later, else he STAYS.

The value iteration for the case 0 converges after 125 iterations of the algorithm, which is considerable since we have such a high(close to 1) discount factor( $\gamma$ ). As we decrease the  $\gamma$ , the number of iterations is reduced as seen in case 3 (where we keep the discount factor as 0.25) and the iterations of the value iteration algorithm are reduced to 9.

## Game Simulation -

For (W, 0, 0, D, 100)

We find after multiple simulations that the best strategy for IJ is to directly head from W to E and then repeatedly HIT MM. The fact that we have no arrows and a high negative STEP\_COST means we should try to end the game as soon as possible which is what IJ tries to do.

```
START STATE: ('W', 0, 0, 'D', 100)
('W', 0, 0, 'D', 100) RIGHT ('C', 0, 0, 'D', 100) -20.0
('C', 0, 0, 'D', 100) RIGHT ('E', 0, 0, 'D', 100) -39.98
('E', 0, 0, 'D', 100) HIT ('E', 0, 0, 'D', 50) -59.94
('E', 0, 0, 'D', 50) HIT ('E', 0, 0, 'D', 0) -79.88
Overall Reward: -29.88
```

```
START STATE: ('W', 0, 0, 'D', 100)
('W', 0, 0, 'D', 100) RIGHT ('C', 0, 0, 'D', 100) -20.0
('C', 0, 0, 'D', 100) RIGHT ('E', 0, 0, 'R', 100) -39.98
('E', 0, 0, 'R', 100) HIT ('E', 0, 0, 'R', 100) -59.94
('E', 0, 0, 'R', 100) HIT ('E', 0, 0, 'D', 100) -119.76
('E', 0, 0, 'D', 100) HIT ('E', 0, 0, 'D', 100) -139.68
('E', 0, 0, 'D', 100) HIT ('E', 0, 0, 'R', 100) -159.581
('E', 0, 0, 'R', 100) HIT ('E', 0, 0, 'R', 100) -179.461
('E', 0, 0, 'R', 100) HIT ('E', 0, 0, 'R', 50) -199.321
('E', 0, 0, 'R', 50) HIT ('E', 0, 0, 'R', 50) -219.162
('E', 0, 0, 'R', 50) HIT ('E', 0, 0, 'R', 0) -238.983
```

For (C, 2, 0, R, 100)

We find after multiple simulations that the best strategy for IJ is to go up from C to N and keep crafting arrows until MM becomes dormant. The fact that MM is in the ready state initially and IJ is in C state means that MM's attack on IJ can be successful. Thus, to prevent this, MM goes to N where he can make arrows which will be helpful later on to kill MM. IJ STAYS there until MM becomes DORMANT, after which IJ goes down and right to E to attack MM. If he has gathered arrows, he uses them as its expected value of doing damage with it is more and he keeps HIT ing MM until it dies.

```

START STATE: ('C', 2, 0, 'R', 100)
('C', 2, 0, 'R', 100) UP ('N', 2, 0, 'R', 100) -20.0
('N', 2, 0, 'R', 100) CRAFT ('N', 1, 1, 'R', 100) -39.98
('N', 1, 1, 'R', 100) CRAFT ('N', 0, 2, 'D', 100) -59.94
('N', 0, 2, 'D', 100) DOWN ('C', 0, 2, 'D', 100) -79.88
('C', 0, 2, 'D', 100) RIGHT ('E', 0, 2, 'D', 100) -99.8
('E', 0, 2, 'D', 100) HIT ('E', 0, 2, 'D', 50) -119.7
('E', 0, 2, 'D', 50) SHOOT ('E', 0, 1, 'D', 25) -139.581
('E', 0, 1, 'D', 25) SHOOT ('E', 0, 0, 'R', 0) -159.441
Overall Reward: -109.441

```

```

START STATE: ('C', 2, 0, 'R', 100)
('C', 2, 0, 'R', 100) UP ('N', 2, 0, 'R', 100) -20.0
('N', 2, 0, 'R', 100) CRAFT ('N', 1, 1, 'R', 100) -39.98
('N', 1, 1, 'R', 100) CRAFT ('N', 0, 3, 'D', 100) -59.94
('N', 0, 3, 'D', 100) DOWN ('C', 0, 3, 'D', 100) -79.88
('C', 0, 3, 'D', 100) RIGHT ('E', 0, 3, 'D', 100) -99.8
('E', 0, 3, 'D', 100) SHOOT ('E', 0, 2, 'D', 75) -119.7
('E', 0, 2, 'D', 75) SHOOT ('E', 0, 1, 'D', 50) -139.581
('E', 0, 1, 'D', 50) SHOOT ('E', 0, 0, 'R', 50) -159.441
('E', 0, 0, 'R', 50) HIT ('E', 0, 0, 'D', 75) -218.963
('E', 0, 0, 'D', 75) HIT ('E', 0, 0, 'D', 75) -238.784
('E', 0, 0, 'D', 75) HIT ('E', 0, 0, 'D', 75) -258.584
('E', 0, 0, 'D', 75) HIT ('E', 0, 0, 'D', 75) -278.366
('E', 0, 0, 'D', 75) HIT ('E', 0, 0, 'D', 75) -298.127
('E', 0, 0, 'D', 75) HIT ('E', 0, 0, 'R', 25) -317.868
('E', 0, 0, 'R', 25) HIT ('E', 0, 0, 'R', 25) -337.59
('E', 0, 0, 'R', 25) HIT ('E', 0, 0, 'R', 25) -357.292
('E', 0, 0, 'R', 25) HIT ('E', 0, 0, 'R', 0) -376.975
Overall Reward: -326.975

```

## Task 2:

### Case 1 -

Indiana now on the LEFT action at East Square will go to the West Square.

#### Rate of Convergence

The rate of convergence for case 1 is a similar rate of convergence to the original case where the only difference is that its initial convergence rate is a little less in value, which implies that it converges better than the actual case in the beginning. In this scenario, Value Iteration took 125 iterations to converge.

#### In terms of policies

These are no significant policy changes in this case.

## Case 2 -

The step cost of the STAY action is now zero.

### Rate of convergence

As the step cost for STAY action is now zero, the policy underwent many changes and it can be seen that the convergence rate remains constant at a lower level than the previous two. This implies that it converges faster than the previous algorithms and also took fewer iterations to converge. Finally, in this case, 62 iterations were required for convergence.

### In terms of policies

When in west state, IJ does STAY in all cases and remains in west indefinitely. This is because as the step cost is now 0, the reward is maximized by just remaining in the same place and not doing anything. The step cost of all the other actions is negative, except when IJ would attack and MM would die. However, the probability of that happening is so low that staying still remains beneficial. Further at C, if MM's health is greater than 25, anyway 2 actions would be needed to kill MM so he prefers to go left to W always. At E, MM chooses either to SHOOT, STAY or go left to C depending upon the number of arrows he has and MM's health. If MM is ready and IJ is at N or S, he prefers to STAY to prevent getting hurt from MM in future. At S is MM is dormant, IJ mostly goes up to C. At N when MM is dormant, IJ mostly goes down to C when MM's health is  $\geq 50$  while other times using the actions STAY or CRAFT.

---

## Case 3 -

Change the value of gamma to 0.25

### Rate of convergence

As the discount factor changed to 0.25, the convergence rate is lower compared to the previous cases. Therefore, it converges quickly compared to the other algorithms and as a result, it only took 8 iterations to converge.

### In terms of policies

In this case, the value iteration takes just 8 iterations to complete, instead of 124 when gamma was 0.999. This is because as the reward decreases substantially, the value iteration converges below the Bellman error significantly faster as well.

The policy obtained in this seems sub-optimal as compared to the policy obtained before. This is because IJ often times does non-risky things such as craft when he is at the state (N, 1, 0, D, 25). The previous policy recognized that given the step cost, it would be better to just go DOWN. However, this one still does CRAFT. It often stays at N, S and W while sometimes taking action CRAFT at N and GATHER at S. Also, it doesn't HIT unless it is at E. At C, it may SHOOT or HIT to try to kill MM while also trying to move to other positions depending upon parameters like arrows and health of MM.