MDL Project

Markov Decision Process

Part II

Made by - Team 92

Kshitijaa Jaglan (2019115005)

Rishabh Khanna (2019113025)

TASK 1

Trends observed:

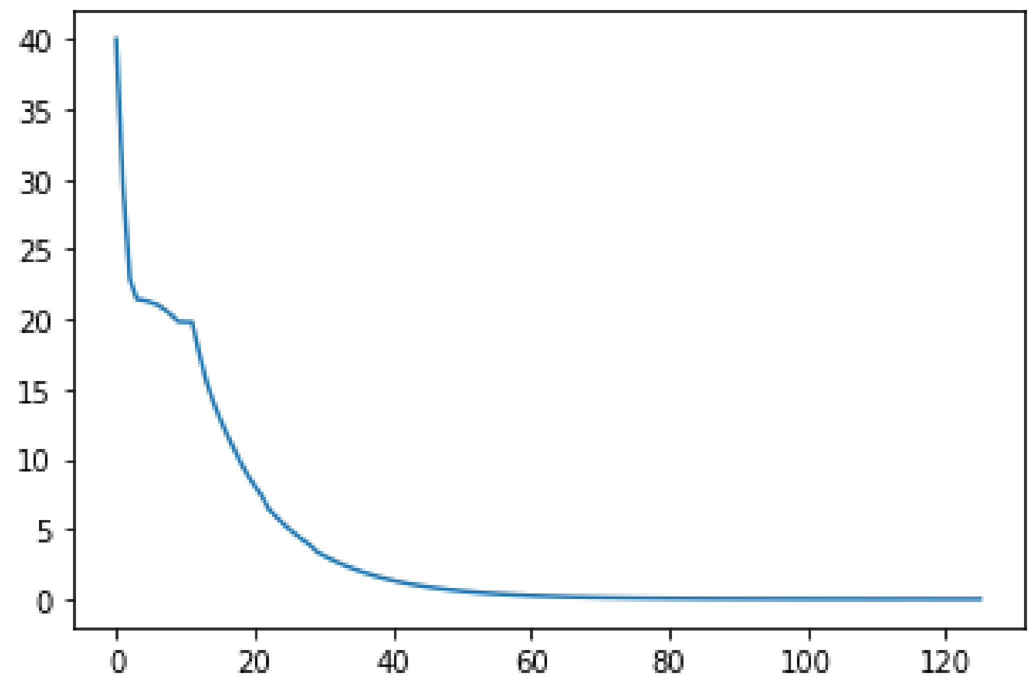
- **Minimising step costs:** From the trace we notice that IJ is inclined to spend turns hiding from MM attacks to avoid the negative rewards. The reason for the above is our low step cost (-5). For higher step costs this is not the case. *For example* IJ will often choose to STAY when he is at WEST and has 0 ARROWS.
- **Attack vs gather:** IJ has a tendency to spam HIT since continuously hitting instead of gathering materials is optimal as this reduces the number of steps involved in killing MM.
- **Preferred locations:** His preferred attacking position is EAST as attacking from here results in the highest accuracy of 0.9 for the action SHOOT (provided IJ has arrows) as compared to locations like WEST where the same is 0.25. Moreover, this location is also preferred by the algorithm itself, as it teleports IJ to EAST with a 0.15 probability when he is at NORTH, CENTER or SOUTH.
  For example, we can see the the first simulation below where, once IJ reaches the location EAST, he continuously uses the action HIT and has barely used it at any other locations
- **By location**:
  - **Center:** In the initial iterations, we see that the preferred actions are HIT, SHOOT and sometimes UP, but towards the end, we see that almost always the preferred action is RIGHT to go to EAST and attack as it had the highest chance of dealing damage to MM.
  - **West:** Initially, almost all the actions were either RIGHT or SHOOT, which was kind of maintained till the end. Here, IJ basically has two options - SHOOT with a 0.25 probability, or go RIGHT to CENTER followed by EAST and attack from there.
  - **East:** Initially, the actions were a mix of LEFT, SHOOT and HIT but by the end, almost all of them changed to HIT and SHOOT depending on the arrows available, in order to deal the required damage to MM
  - **South:** Initially almost all the actions were UP but by the end they gradually changed to a mixture of GATHER and UP depending on the

amount of material already available, which can be used later in the stages of crafting to create arrows

- **North:** Like SOUTH, even NORTH had mainly only the DOWN actions initially; but by the end, these were converted to a combination of DOWN and CRAFT to create arrows to be used for attack on MM

- **By MM's State**
    - **Dormant:** IJ tried to HIT and SHOOT MM as much as possible to yield the damage while not being at the risk of getting attacked himself
    - **Ready:** IJ tries to STAY in most of the cases to decrease chances if getting hit

Graph of Convergence:



X: Number of iterations
Y: Maximum absolute between successive iterations

The above mentioned graph shows how the maximum absolute difference between successive iterations decreased and converged to be less than the given value of gamma.

Game Simulations:

1. Start state (W,0 ,0, D, 100):

```
[W, 0, 0, D, 4] :  RIGHT
[C, 0, 0, R, 4] :  DOWN
[S, 0, 0, R, 4] :  GATHER
[S, 0, 0, D, 4] :  UP
[C, 0, 0, D, 4] :  RIGHT
[E, 0, 0, D, 4] :  HIT
[E, 0, 0, D, 4] :  HIT
[E, 0, 0, R, 2] :  HIT
[E, 0, 0, R, 2] :  HIT
[E, 0, 0, D, 3] :  HIT
[E, 0, 0, R, 1] :  HIT
[E, 0, 0, D, 2] :  HIT
[E, 0, 0, D, 2] :  HIT
[E, 0, 0, R, 2] :  HIT
[E, 0, 0, D, 3] :  HIT
[E, 0, 0, D, 1] :  HIT
[E, 0, 0, R, 0] :  NONE
```

- In this simulation we notice that **EAST** is the **preferred location** for IJ. This is an example of an optimal method of killing MM.
- The reason for IJ to move down and gather is to **dodge** the incoming attacks from MM. Gathering serves as a "STAY" action with a 100% success rate. Thus we see IJ to prefer gathering material though he doesn't choose to use the material, the same holds for crafting.
- Here, **MM** never enters the **ready** state, except when he's killed. Thus there is no need for IJ to dodge attacks.
- Once IJ **reaches east** we notice that he chooses to **continuously hit** instead of gathering materials as this reduces the steps counts involved in gathering the required material. Staying still and constantly choosing to hit is more optimal

2. Start state (C, 2 ,0, R, 100):

```
[C, 2, 0, R, 4] :  UP
[N, 2, 0, R, 4] :  STAY
[N, 2, 0, D, 4] :  CRAFT
[N, 1, 1, D, 4] :  CRAFT
[N, 0, 3, R, 4] :  STAY
[N, 0, 3, R, 4] :  STAY
[E, 0, 3, R, 4] :  HIT
[E, 0, 0, D, 4] :  HIT
[E, 0, 0, D, 4] :  HIT
[E, 0, 0, D, 2] :  HIT
[E, 0, 0, D, 2] :  HIT
[E, 0, 0, D, 0] :  NONE
```

- In this simulation we notice that **NORTH** is the **preferred location** for IJ. This is an example of IJ attempting to dodge MM.
- IJ then chooses to **CRAFT** arrows to **SHOOT** MM from.
- Here, MM enters the ready state again. So IJ chooses to **STAY** dodging MM's attacks
- He is then teleported to the **EAST** where **HIT** is the most optimal method of killing the now DORMANT MM.

Change in Policy for different cases:

> **Case 1 :** Indiana now on the LEFT action at East Square will go to the West Square

- Trends:

We see that there is not much change in this case as compared to Task 1 as there was not much difference in the constraints. We also notice that it took 8 more iterations to converge to a value less than gamma.

Apart from this, some of the HIT and SHOOT actions were now changed to LEFT to go to EAST and then attack.

> **Case 2 :** The step cost of the STAY action is now zero.

- Trends:

We see that there is a considerable difference in the number of iterations used.
This was because now IJ can wait till MM goes back to DORMANT state to attack him as there was no cost associated with waiting by using the action STAY.
We also notice that IJ suffers minimal attacks in this case as it tries not to give many chances of attack to MM by using STAY and protecting himself

> **Case 3 :** Change the value of gamma to 0.25

- Trends:

We notice that the number of iterations in this case were merely 8 as compared to 115 used in Task 1. This is because a gamma value of 0.25 is significantly bigger than the initial value of gamma, thus setting a limit easily achievable in less iterations as seen in the trace files.