S21 - MDL Assignment 2 part 2

Team 7
Rutvij Menavlikar (2019111032)
Tejas Chaudhari (2019111013)

April 8, 2021

Value Iteration Algorithm

The Bellman equation is the basis of the value iteration algorithm for solving Markov Decision Process. Let $U_i(I)$ be the utility for state s at the i^{th} iteration. The iteration step, called a Bellman update , looks like this:

$$U_{i+1}(I) = \max_A [R(I, A) + \sum_j P(J|I, A) \cdot U_i(J)]$$

Where the Rewards R(I,A) for each state is the expected reward of taking action A in State I. That is:

$$R(I, A) = \sum_j P(J|I, A) \cdot R(J, A, I)$$

The value iteration algorithm is as follows:

function Value-Iteration(mdp, ϵ) **returns** a utility function
inputs: mdp, an MDP with states S, actions A(s), transition model P($s'|s, a$), rewards R(s), discount γ,
the maximum error allowed in the utility of any state ϵ
local variables: U, U', vectors of utilities for states in S, initially zero
δ, the maximum change in the utility of any state in an iteration

repeat
$U \leftarrow U'; \delta \leftarrow 0$
for each state s **in** S **do**
$U'[s] \leftarrow \max_{a \in A(s)} (R(s, a) + \gamma \sum_{s'} P(s'|s, a) U[s'])$
if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]||$
until $\delta < \epsilon$
return U

Task 1

γ: 0.999
δ: 0.001
iterations required to converge: 118

From the trace file part_2_trace.txt we can infer the following

- In ***West Square***:
  - When *MM* is in dormant state, if *IJ* has sufficient arrows then he chooses to *SHOOT* when *MM* has low health, otherwise he moves *RIGHT* to attack *MM* with *HIT* before *MM* comes back to ready state.
  - When *MM* is in ready state, *IJ* tries to stay out of *MM*'s attack range if he has arrows and *MM*'s health is low, otherwise he moves *RIGHT* to *HIT*, *CRAFT* or *GATHER*, depending on how much materials he has and *MM*'s health. Depending on his arrows and *MM*'s health, *IJ* chooses to *SHOOT* or *STAY*
- In ***North Square***:
  - When *MM* is in dormant state, *IJ* moves *DOWN* when he has no materials, he has sufficient arrows or *MM*'s health is 25 and *IJ* has 1 arrow. In all other cases *IJ* chooses to *CRAFT* arrows.

- When *MM* is in ready state, *IJ* tries to *STAY* out of *MM*'s attack range. *IJ* chooses to stay when he has no material or when he has 3 arrows. Otherwise *IJ* chooses to *CRAFT*. *IJ* chooses to go *DOWN* to *GATHER* materials when *MM*'s health is full, and he doesn't have sufficient arrows and no materials. Also, *IJ* chooses to *STAY* when he has sufficient arrows and *MM*'s health is 25

- In ***East Square***:
  - When *MM* is in dormant state, *IJ* always attacks. When he is out of arrows or when *MM* has full health and *IJ* does not have all his arrows, he attacks with *HIT*. Otherwise, he attacks by *SHOOT*
  - When *MM* is in ready state, *IJ* still attacks. When he is out of arrows, or when *MM* has full health, he attacks with *HIT*, otherwise he attacks by *SHOOT*

- In ***South Square***:
  - When *MM* is in dormant state, *IJ* chooses to *GATHER* only when he does not have arrows and materials, and when *MM*'s health is 25. In all other case, he goes *UP* to attack or *CRAFT*.
  - When *MM* is in ready state, *IJ* goes *UP* when *MM*'s health is 100 and he has insufficient arrows, or when he has 2 materials and insufficient arrows. He either chooses to *GATHER* materials or *STAY* depending on how many materials and arrows he has.

- In ***Center Square***:
  - When *MM* is in dormant state, *IJ* moves *RIGHT* to attack when he has no materials, and moves *UP* to craft or again *RIGHT* to attack depending on how many arrows he has and *MM*'s health
  - When *MM* is in ready state, *IJ* ties to stay out of *MM*'s attack range by going *UP* or *DOWN*. When *IJ* has sufficient arrows, he goes *LEFT* to attack *MM* by shooting. If *IJ* has arrows and *MM*'s health is 25, he attacks *MM* by *SHOOT*.

Task 2

Case 1

γ: 0.999
δ : 0.001
iterations required to converge: 119

From the trace file `part_2.1_trace.txt` we can infer the following

- In ***West Square***:
  - When *MM* is in dormant state, if *IJ* has sufficient arrows then he chooses to *SHOOT* when *MM* has low health, otherwise he moves *RIGHT* to attack *MM* with *HIT* before *MM* comes back to ready state.
  - When *MM* is in ready state, *IJ* tries to stay out of *MM*'s attack range if he has arrows and *MM*'s health is low, otherwise he moves *RIGHT* to *HIT*, *CRAFT* or *GATHER*, depending on how much materials he has and *MM*'s health. Depending on his arrows and *MM*'s health, *IJ* chooses to *SHOOT* or *STAY*

- In ***North Square***:
  - When *MM* is in dormant state, *IJ* moves *DOWN* when he has no materials, he has sufficient arrows or *MM*'s health is 25 and *IJ* has 1 arrow. In all other cases *IJ* chooses to *CRAFT* arrows.

- When *MM* is in ready state, *IJ* tries to *STAY* out of *MM*'s attack range. *IJ* chooses to stay when he has no material or when he has 3 arrows. Otherwise *IJ* chooses to *CRAFT*. *IJ* chooses to go *DOWN* to *GATHER* materials when *MM*'s health is full, and he doesn't have sufficient arrows and no materials. Also, *IJ* chooses to *STAY* when he has sufficient arrows and *MM*'s health is 25

- In ***East Square***:
  - When *MM* is in dormant state, *IJ* always attacks. When he is out of arrows or when *MM* has full health and *IJ* does not have all his arrows, he attacks with *HIT*. Otherwise, he attacks by *SHOOT*
  - When *MM* is in ready state, *IJ* still attacks. When he is out of arrows, or when *MM* has full health, he attacks with *HIT*, otherwise he attacks by *SHOOT*

- In ***South Square***:
  - When *MM* is in dormant state, *IJ* chooses to *GATHER* only when he does not have arrows and materials, and when *MM*'s health is 25. In all other case, he goes *UP* to attack or *CRAFT*.
  - When *MM* is in ready state, *IJ* goes *UP* when *MM*'s health is 100 and he has insufficient arrows, or when he has 2 materials and insufficient arrows. He either chooses to *GATHER* materials or *STAY* depending on how many materials and arrows he has.

- In ***Center Square***:
  - When *MM* is in dormant state, *IJ* moves *RIGHT* to attack when he has no materials, and moves *UP* to craft or again *RIGHT* to attack depending on how many arrows he has and *MM*'s health
  - When *MM* is in ready state, *IJ* ties to stay out of *MM*'s attack range by going *UP* or *DOWN*. When *IJ* has sufficient arrows, he goes *LEFT* to attack *MM* by shooting. If *IJ* has arrows and *MM*'s health is 25, he attacks *MM* by *SHOOT*.

Case 2

γ: 0.999
δ : 0.001
iterations required to converge: 57
From the trace file part_2.2_trace.txt we can infer the following

- In ***West Square***:
  - When *MM* is in dormant state, if *IJ* has sufficient arrows then he chooses to *SHOOT* when *MM* has low health, otherwise he chooses to *STAY*
  - When *MM* is in ready state, *IJ* tries to stay out of *MM*'s attack range. Depending on his arrows and *MM*'s health, *IJ* chooses to *SHOOT* or *STAY*

- In ***North Square***:
  - When *MM* is in dormant state, *IJ* moves *DOWN* when he has no materials. He chooses to *STAY* when he has sufficient arrows and *MM*'s health is low. In all other cases *IJ* chooses to craft arrows.
  - When *MM* is in ready state, *IJ* tries to stay out of *MM*'s attack range. *IJ* chooses to *STAY* when he has no materials or when *MM* has high health. Otherwise *IJ* chooses to craft.

- In ***East Square***:

- When *MM* is in dormant state, *IJ* attacks when *MM*'s health is low. When he is out of arrows, he attacks with *HIT*, otherwise he attacks by *SHOOT*. When *MM* has high health, *IJ* move *LEFT*.
- When *MM* is in ready state, *IJ* attacks with *SHOOT* when *MM*'s health is low and he has sufficient arrows, otherwise he goes *LEFT*.

- In ***South Square***:
  - When *MM* is in dormant state, *IJ* chooses to stay if he has sufficient arrows. In all other cases he goes *UP* to attack or *CRAFT*.
  - When *MM* is in ready state, *IJ* always chooses to *STAY*.

- In ***Center Square***:
  - When *MM* is in dormant state, if *MM*'s health is high, *IJ* moves *LEFT*, otherwise, if he has sufficient arrows, he attacks with *SHOOT*. If he has enough materials, he moves, *UP*, otherwise he moves *RIGHT*.
  - When *MM* is in ready state, if *MM*'s health is high *IJ* moves *LEFT*, otherwise if he has sufficient arrows then he moves *UP*.

Case 3

γ: 0.25
δ : 0.001
iterations required to converge: 8
From the trace file part_2.3_trace.txt we can infer the following

- In ***West Square***:
  - When *MM* is in dormant state, if *IJ* has arrows then he chooses to *SHOOT*, otherwise he chooses to *STAY* when *MM*'s health is high and move *RIGHT* when health is low.
  - When *MM* is in ready state, *IJ* tries to stay out *MM*'s attack range. If *IJ* has arrows, he chooses to *SHOOT*, otherwise he chooses to *STAY*.

- In ***North Square***:
  - When *MM* is in dormant state, *IJ* moves *DOWN* if *MM*'s health is low. Otherwise, he choose to *STAY* when he has no materials and *CRAFT* if he has materials
  - When *MM* is in ready state, *IJ* tries to stay out of *MM*'s attack range. *IJ* goes *DOWN* when he has arrows, otherwise he chooses to *CRAFT* when he has materials, and *STAY* when he doesn't.

- In ***East Square***:
  - When *MM* is in dormant state, *IJ* attacks with *SHOOT* when *MM*'s health is low and he has arrows, otherwise he attacks by *HIT*.
  - When *MM* is in ready state, *IJ* attacks with *SHOOT* if *MM*'s health is very low and he has arrows, attacks with *HIT* if *MM*'s health is medium, otherwise he goes *LEFT*

- In ***South Square***:
  - When *MM* is in dormant state, *IJ* chooses to *GATHER* if *MM*'s health is high and go *UP* when health is low
  - When *MM* is in ready state, *IJ* goes up when *MM*'s is the least and he has arrows, otherwise, he chooses to *GATHER*

- In ***Center Square***:
  - When *MM* is in dormant state, if *MM* health is high *IJ* moves *LEFT*, if he has arrows and *MM*'s health is low, he attacks with *SHOOT*, otherwise he attacks with *HIT*
  - When *MM* is in ready state, if *MM*'s health is high, *IJ* moves *LEFT*, if he has arrows and *MM*'s health is low, he attacks with *SHOOT*, otherwise he attacks with *HIT*.
