Isabelle Chalhoub - u0678302
Karla Kraiss - u0830999
CS4300
November 9, 2017

A5 Lab Report

1. Introduction

Our agent in this lab performed statistics using Monte Carlo methods to predict where safe/unsafe cells may be on the board and try to maneuver around them. The agent performs similarly to the last lab but instead of deriving whether cells are safe from a knowledge base, we use probability and try to make the smartest decision we can, given the likelihood that a cell contains something dangerous. We would like to see:

- How successful is our Monte Carlo agent (in terms of score)?
- How impactful is the number of boards we generate for our probabilities on the average score of our agent?

2. Method

CS4300_WP_Estimates:

This method generates a random board and compares if that board fits the given percepts. It heavily relies on a helper function CS4300_Check_board which checks the neighbors of the agent to make sure the random board is viable. Once a viable board is generated, the method adds that to a growing sum of viable locations for pits and wumpuses. Once the method has generated the given number of viable boards or reached the threshold, it calculates the probability that any given cell will have a pit or wumpus based on the viable boards.

CS4300_MC_agent:

This method has absolute information (wumpus, pits, gold, safe) that it passes to the WP_Estimates. Once it has an unsafe estimate it runs through a checklist of what to do. First priority is to check if the agent is at the gold, and if it is then grab the gold and go home. Next, it will try to shoot the wumpus if wumpus location is known using the probability tables in conjunction with it's absolute knowledge of the board so far. Once the wumpus is dead we no longer consider this as a viable option for the agent to make. If it cannot kill the wumpus or attain the gold, it needs to figure out where to go to next. The agent picks the safest option by first checking neighbors and neighbors of visited for lowest risk and makes a plan to go there.

If the agent has visited every cell that it can, it either kills the wumpus if it's not dead or goes home because the gold is impossible to obtain.

3. Verification of Program

WP Estimates:

From the solution of Quiz 9 we know the expected probabilities for these different breeze and stench boards and our WP estimates

```
1a.     breezes            pit probabilities

-1    -1    -1    -1        0.2 0.2 0.2 0.2
-1    -1    -1    -1        0.2 0.2 0.2 0.2
-1    -1    -1    -1        0.0 0.2 0.2 0.2
 0    -1    -1    -1        0.0 0.0 0.2 0.2

      stenches            Wumpus probabilities

-1    -1    -1    -1      0.08 0.08 0.08 0.08
-1    -1    -1    -1      0.08 0.08 0.08 0.08
-1    -1    -1    -1      0.0  0.08 0.08 0.08
 0    -1    -1    -1      0.0  0.0  0.08 0.08
```

```
>> [p,w] = CS4300_WP_estimates(breezes,stenches,10000)

p =

    0.1916    0.2012    0.1882    0.1917
    0.2103    0.1954    0.1964    0.1953
         0    0.1951    0.2006    0.1904
         0         0    0.1995    0.1995


w =

    0.0669    0.0650    0.0713    0.0700
    0.0735    0.0693    0.0682    0.0678
         0    0.0713    0.0667    0.0697
         0         0    0.0715    0.0752
```

calculations:

```
1c.     breezes            pit probabilities

-1    -1    -1    -1        0.2 0.2 0.2 0.2
-1    -1    -1    -1        0.2 0.2 0.2 0.2
-1    -1    -1    -1        0.0 0.2 0.2 0.2
 0    -1    -1    -1        0.0 0.0 0.2 0.2

      stenches            Wumpus probabilities

-1    -1    -1    -1        0.0 0.0 0.0 0.0
-1    -1    -1    -1        0.0 0.0 0.0 0.0
-1    -1    -1    -1        0.5 0.0 0.0 0.0
 1    -1    -1    -1        0.0 0.5 0.0 0.0
```

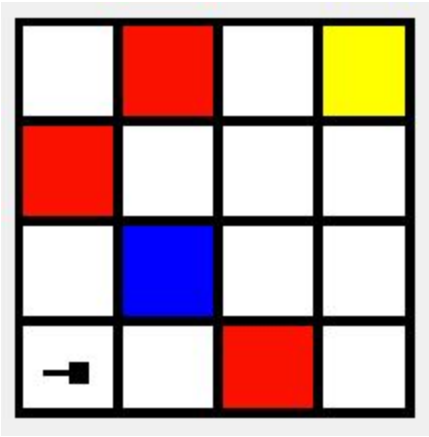
```
>> [p,w] = CS4300_WP_estimates(breezes,stenches,10000)

p =

    0.2278    0.2097    0.2198    0.2016
    0.2198    0.1986    0.1875    0.2087
         0    0.2228    0.2359    0.1754
         0         0    0.2157    0.2077


w =

         0         0         0         0
         0         0         0         0
    0.4718         0         0         0
         0    0.5282         0         0
```

```
1e.     breezes            pit probabilities

-1    -1    -1    -1        0.2 0.2 0.2 0.2
-1    -1    -1    -1        0.3 0.2 0.2 0.2
 1    -1    -1    -1        0.0 0.8 0.2 0.2
 0     1    -1    -1        0.0 0.0 0.3 0.2

      stenches            Wumpus probabilities

-1    -1    -1    -1        0.1 0.1 0.1 0.1
-1    -1    -1    -1        0.0 0.1 0.1 0.1
 0    -1    -1    -1        0.0 0.0 0.1 0.1
 0     0    -1    -1        0.0 0.0 0.0 0.1
```

```
>> [p,w] = CS4300_WP_estimates(breezes,stenches,10000)

p =

    0.2082    0.1914    0.2045    0.1821
    0.3119    0.1905    0.1867    0.1923
         0    0.8646    0.1979    0.1905
         0         0    0.3212    0.1989


w =

    0.0868    0.0915    0.0840    0.0980
         0    0.0738    0.0943    0.0915
         0         0    0.0887    0.0906
         0         0         0    0.0906
```

MC Agent:

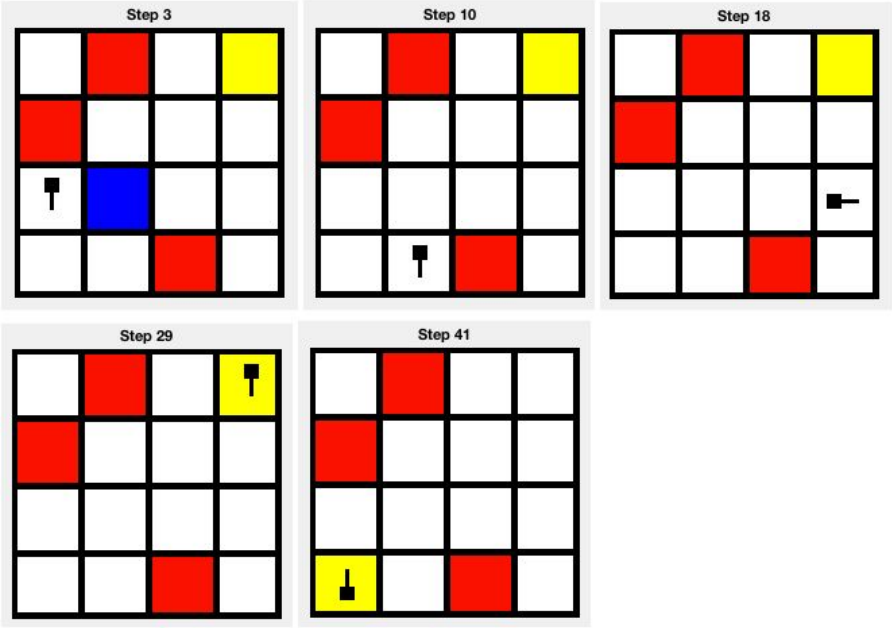
Given this board:



Our agent should be able to figure out where the Wumpus is, shoot it, explore further to find the gold, and return home.
We will show a few steps that the agent goes through(without showing all of them) and included with our file submission is a trace called:

mc_agent_verif_trace.mat
That can be viewed using: M = CS4300_show_trace(t,1);
After loading our .mat file.

As shown in these steps and in the full trace, the agent successfully predicts the location of the Wumpus and kills it. After that it is free to explore further on the board, discover the gold, and safely make it back home.



4. Data and Analysis

For all boards combined:

	Mean Score	Successes	Failures
No MC (random agent)	-825.4776	0	2500
50 Samples	20.6740	1307	1193
100 Samples	27.9272	1313	1187
200 Samples	29.4604	1317	1183

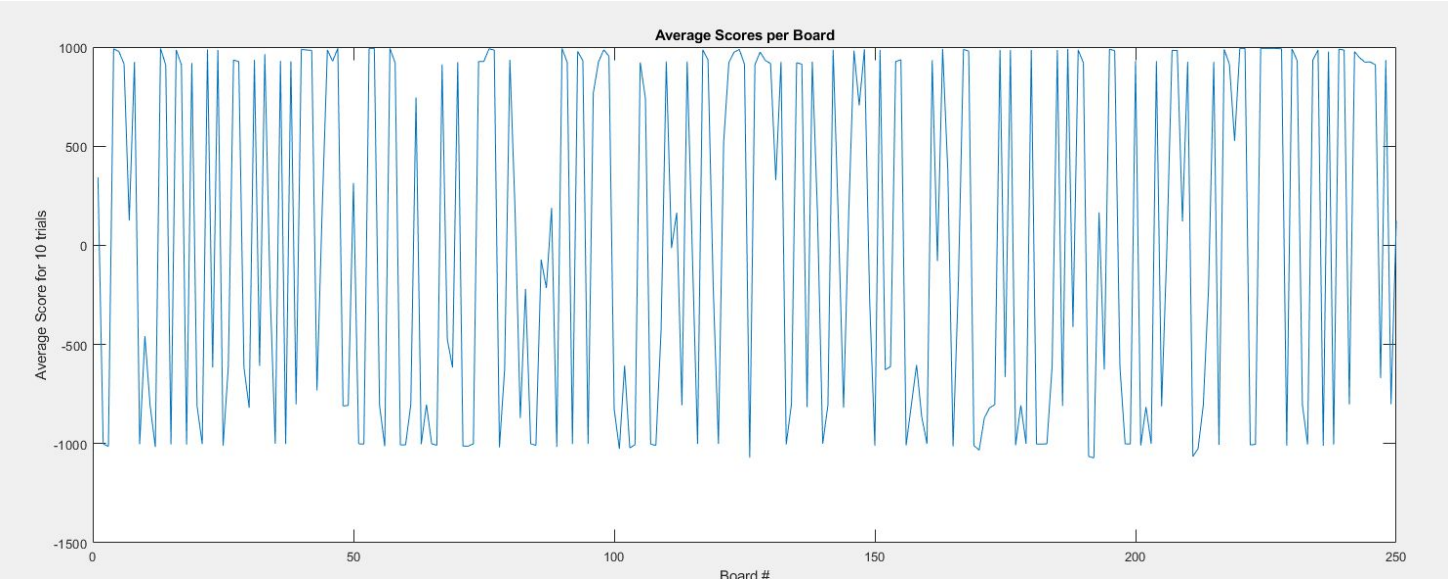
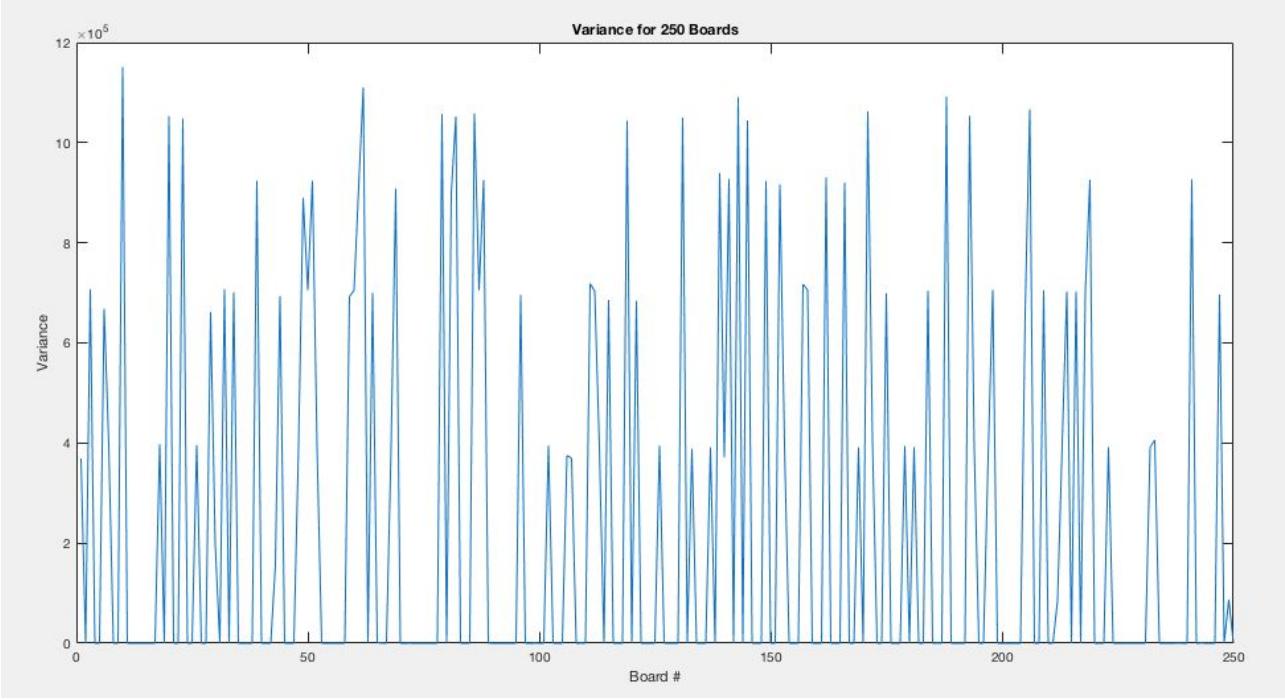


Figure 1 ^

Figure 2



5. Interpretation

With the various trials per board, our agent seems to be successful only half of the time. It was consistently succeeding or failing on specific boards for every trial. This is obvious in Figure 1 where the averages are either a peak at nearly 1000 or a dip at nearly -1000. This means that the probabilities it generates are fairly consistent and our agent tends to make the same decisions with little randomness. Depending on the scenario (likely the frequency/placement of pits), our agent is either completely successful or will die every time.

Furthermore, the number of MC samples used seems to have a very minimal effect on the success of the agent. It only slightly increased the success rate/score. We predict that if we used even more samples, the success rate would plateau around an average score of 30. We think this because when we increased the sample size by 50 (50-100) we noticed a larger jump in our average, but when increasing it by 100 (100-200) we only witnessed a 2 point jump.

We think the way we are predicting safe/unsafe cells is working perfectly but our agent might not be using that knowledge to the best of its ability. Our agent could use that knowledge even more intelligently to make better decisions based on the probabilities. Granted, a lot of the failures come from risking a choice between cells that have very similar unsafe probabilities. Our agent makes very consistent decisions. When posed with a 50/50 chance of death or survival, if it chooses incorrectly the first time, it is so consistent it will make the same incorrect decision on every trial of that board. A change we could have tried is when posed with a 50/50 decision of life/death, we could randomly decide instead of relying on probabilities that are virtually indistinguishable. The differences are so small that the agent can’t possibly rely on the one that is slightly smaller being the right decision.

6. Critique

In this assignment we had to figure out how to implement Monte Carlo in our Wumpus World scenario. Our solution naively generates random boards and checks if those boards are possible. Some important concepts we learned from this were capping these random board generations if we didn’t reach the desired number of correct boards and minimum meaningful decision threshold. The random board generation

was the bottleneck of our tests, since our maximum cap was set to 10,000. Our agent was so consistent that it would also consistently make the wrong choice leading to its death. In some cases where the probabilities are close it's a better to pick randomly. If we had implemented a minimum meaningful difference between probability choices, such as a .1 difference, our agent would likely have had better success. This is because our agent would often come across a .49 vs a .51 decision, and would always choose the .49 because it was a smaller risk. In reality these number are basically equivalent, and should have been treated as such.

Something else we could improve upon if doing this assignment again would be to not randomly generate the boards. If we instead tried to generate boards that always fit the percept, that would improve our agent's success rate and our testing speed. Our agent would have a higher success rate because it wouldn't have to give up if we exceed 10,000 board generations, and our testing would speed up because we would never generate more boards than the number given in WP_Estimate.

7. Log

Isabelle's Log: (Odd Sections)
I worked for roughly 22 hours on this assignment.

Karla's Log: (Even Sections)
I worked for roughly 19 hours during this assignment. 17 hours on code, 2 hours on the lab report.