# A5 – CS4300 Lab Report – 11/9/17

# An Agent Using Monte Carlo Probability Prediction to Navigate Wumpus World

## Introduction

Each iteration of the agents of AI past seemed to be stricken by ails of one kind or another. My random agent was easy to code but had no idea where he was going or even why he was going there. My RTP agent was cold and calculating, but often found himself lost in thought about where pits could be for far too long. This time, I aimed to see if using Monte Carlo probability prediction could both keep him up and on his feet while simultaneously keeping those feet out of pits and the mouths of Wumpuses. Monte Carlo methods use mass simulation to predict the possibilities of one single action. In my case, CS4300\_MC\_agent was to randomly generate a large number of boards in order to find boards which would match its own known board (i.e., the known percept values of its board). Once it had done that, it would sample the actual hazards of the generated boards and come up with actual probabilities of those hazards existing in various locations in its own board. The known weaknesses of this method for our problem are:

1. There are many cases where the probability of pits in two neighboring squares are actually theoretically equal. In cases like these, the only option the agent has is to take a 50/50 risk.
2. As the board becomes more explored, it becomes exponentially less likely that a randomly generated board will match the known board’s percept values. Thus, it was take exponentially more time to generate such a board as the agent progresses.

Thus, a few relevant questions I aim to answer in this report are:

1. Does increasing the number of matched boards in the Monte Carlo estimate function increase the potential to beat a board
2. Does putting a limiting factor in the Monte Carlo estimate function to reduce time spent decrease the potential to beat a board?
3. How much does exploration level affect the time performance of our Monte Carlo estimate function?

## 2. Method

Our agent used the hybrid agent model, that is, it produced plans of action based on priority in the following order (each successive plan is only formed if the previous one was not formed):

1. If gold is found, pick it up and return home.
2. Use A\* pathfinding to explore an unexplored known safe location.
3. Try to kill the Wumpus if a stench has been smelled and you still have an arrow.
4. Use A\* pathfinding to explore the least unsafe location.

Our agent used the Monte Carlo method to attempt to estimate where the Wumpus might be as well as which locations were more and less likely to have pits. Because Monte Carlo’s precision can potentially affect the accuracy of its predictions, I made the number of correctly generated boards a variable with which to test against. As mentioned earlier as well, a large number of known percepts will make randomly generated boards much less likely to match the known board. Thus, I also put a limit on the number of total boards to be generated and decided to test how this affected performance as well. Note that in some cases, a matching board could not be generated within the limit of boards to generate. In cases like these, I forced the estimate function to at least generate one matching board before it was allowed to quit. Lastly, I wanted to know how the number of known explored locations affected the performance of the Monte Carlo estimates function so I tested the time needed to run said function vs the number of explored squares as well.

## 3. Verification

For verification of my program, I will provide various example boards with hand-worked theoretical probabilities of pits and Wumpuses existing, as well as the estimates calculated by my Monte Carlo function:

B = Breeze • S = Wumpus • 0 = Clear • U = Unknown

Board 1

Percepts P(Pits)

|  |  |  |  |
| --- | --- | --- | --- |
| U | U | U | U |
| U | U | U | U |
| B | U | U | U |
| 0 | B | U | U |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| 0 |  |  |  |
| 0 | 0 |  |  |

## 4. Data and Analysis