A1 – CS4300

Random Actions in a Wumpus World

# Introduction

# Method

Most algorithms used for calculations are loops that create enough data and store it into a vector. The data contained in these vectors is then used to calculate the mean, variance, confidence intervals, and success percentages needed for the different problems.

The most complex code is in CS4300\_A1\_Part\_II\_2.m where it simulates the actions given by CS4300\_agent1.m. It is a loop where it always asks the agent for the next step and as long the step doesn’t take the agent out of bounds it is taken and then evaluated. If the step resulted in the agent falling into a pit or reaching the gold then it is the end of the execution. When its execution ends it returns the number of steps and result so those can be used for computations.

# Verification of Program

# Data and Analysis

The data for Part I comes from calculations produced by CS4300\_A1\_Part\_I\_2.m when computed with 5000 boards per test.

Solvable Boards with 0-14 pits:

* + Mean — 1.8055e+03
  + Variance — 1.8055e+03
  + 95% Confidence Interval — (Mean) ± 635.9227

The data for Part II comes from calculations produced by CS4300\_A1\_Part\_II\_2\_runner.m when computed with 2000 trials. This test uses CS4300\_A1\_Part\_II\_2.m for each simulation. The simulation gets its instructions from CS4300\_agent1.m.

* + Mean agent steps — 5.1310
  + Variance of agent steps — 10.5768
  + Successful agent percentage — 1%
  + 95% Confidence Interval of mean agent steps — (Mean agent steps) ± 0.1196

The data from Part I shows that a large portion of the boards are solvable with a fairly large margin of error because of the many combinations of boards possible. When you compare this information with the data from Part II which shows that CS4300\_agent1.m is not very effective at solving the boards there is you can learn.

Having just random actions results in about 1% of the games having the agent reach the gold. This means that if the agent was to just continue to make random choices for each action step and that it was not only attempting to get the gold but to return to location (1,1) it would only do so 0.01% of the time. This shows the importance of creating agents that are rational and not just random.

# Interpretation

# Critique

This lab was effective for introducing MATLAB, the concepts of agents, testing, and beginning an evaluation of the Wumpus World. With the simplified constrains as compared to the normal full set of rules it was able to teach how complex it can become for an agent to make decisions based on the percepts it is given despite the fact that percepts could be completely ignored in this lab.

There were no specific issues or problems in this lab. Any difficulty experienced was from a lack of knowledge about MATLAB and the need to review some basic probability and statistics equations.

# Log

Braden Scothern

* + - 60 min — Part I code
    - 60 min — Part II code
    - 45 min — Lab Report — Sections 2, 4, and 6

# Appendix

MATLAB Code Files with brief description:

* + - CS4300\_agent1.m — A simple agent for Wumpus World. It selects randomly from the actions: MOVE\_FORWARD, TURN\_RIGHT, and TURN\_LEFT
    - CS4300\_A1\_Part\_I\_1.m — A function that takes a number of boards to generate and the number of pits for each board. It then returns how many of the randomly generated boards are solvable.
    - CS4300\_A1\_Part\_I\_2.m — Take a number of boards to generate for each test and tests that many boards across the entire interval of 0-14 pits. It then returns the mean number of solvable boards, the variance of solvable boards, and the 95% confidence interval.
    - CS4300\_A1\_Part\_II\_2.m — This will use CS4300\_agent1.m to get instructions and use it to simulate a game of Wumpus. It is very similar in train of thought to the given CS4300\_run\_A1\_trials.m but differs in the fact that it returns the number of steps taken and the result of the simulation.
    - CS4300\_A1\_Part\_II\_2\_runner.m — Takes as input a number of times to use CS4300\_A1\_Part\_II\_2.m to simulate games. It then calculates the mean steps, variance of steps, success rate, and 95% confidence interval for the mean.