Eric Komperud – u0844210

A1 – CS4300 Lab Report

A Random Agent in Wumpus World

1. Introduction

Wumpus World is a dangerous world with promises of bountiful riches if one manages to find the stashed gold and make it out alive. This is a task which requires one's wits about them, but does a random agent truly have the necessary wits? In this lab, 1 of 3 actions for our agent: *forward*, *rotate_right*, or *rotate_left,* will be chosen at random and it will navigate this world. That is, the agent will traverse our static 4x4 board and "try" to get to the gold square while avoiding pits. Of course, the agent will not really try to do this, but it is the ideal outcome in my subjective opinion. The issues I will be addressing are:

- Can a random agent navigate a world and accomplish a goal while simultaneously avoiding pitfalls?
- Whether or not the random agent can do this, can we trust it to be consistent?
- How much freedom should we allow the agent to achieve the optimal outcome most consistently?

The board we will be using is as follows, with the agent starting at location [1,1]:

```
    +--------+--------+--------+-------+
    |        |        |        |       |
  4 |        |  Pit   |        |       |
    |        |        |        |       |
    +--------+--------+--------+-------+
    |        |        |        |       |
  3 |  Pit   |        |        |       |
    |        |        |        |       |
Y   +--------+--------+--------+-------+
    |        |        |        |       |
  2 |        | Gold   |  Pit   |  Pit  |
    |        |        |        |       |
    +--------+--------+--------+-------+
    |        |        |        |       |
  1 |        |        |        |       |
    |        |        |        |       |
    +--------+--------+--------+-------+
        1         2        3       4
                      X
```

Fig A.

Given the nature of random actions, I hypothesize that the agent will likely be able to pick up the gold some of the time but also fall down the various pits at a notably higher rate. Furthermore, I hypothesize that increasing the action limit will have a logarithmic payoff in both gold pickups as well as pitfalls.

2. Method

My agent will make use of Matlab's *rand()* function which generates numbers in a uniform distribution. A single trial consists of starting the agent at location [1,1] and allowing it to take up to N actions, then finally recording how many actions it was able to complete (before falling down a pit or using up all its N actions) as well as whether or not it landed on the gold square [2,2] at some point. I will run 2000 trials and calculate the mean, variance, 95% confidence interval, gold pickup rate, and gold pickup confidence interval over all those trials. I also want to determine how many actions the agent should be allowed to achieve the optimal success rate (success being finding the gold, and not falling down a pit). To do this, I will repeat the 2000 trials for N=10,20,30,…,100 and record the mean actions taken as well as gold-finding success rate for each in a bar graph.

3. Verification of Program

In order to verify that the process is correct, I created a ternary tree of all possible outcomes after 3 actions. The population mean of successful actions was 2.96 and the population mean of gold pickups was 0.04. In order to verify that my program is statistically sound, I ran 500 3-action trials to determine that the sample mean was within a 95% confidence interval of the actual population mean. The sample mean of successful steps taken was 2.958 with a confidence interval of ± 0.0176 and the sample mean of the gold pickup rate was 0.0460 with a confidence interval of ± 0.0184. Both of these samples contain the actual population means within their 95% confidence intervals, so I am 95% confident that my program works properly! See the last page for the ternary tree. (Fig E.)

4. Data and Analysis

After 10 2000-trial runs with a steadily increasing action limit, I obtained the following data:

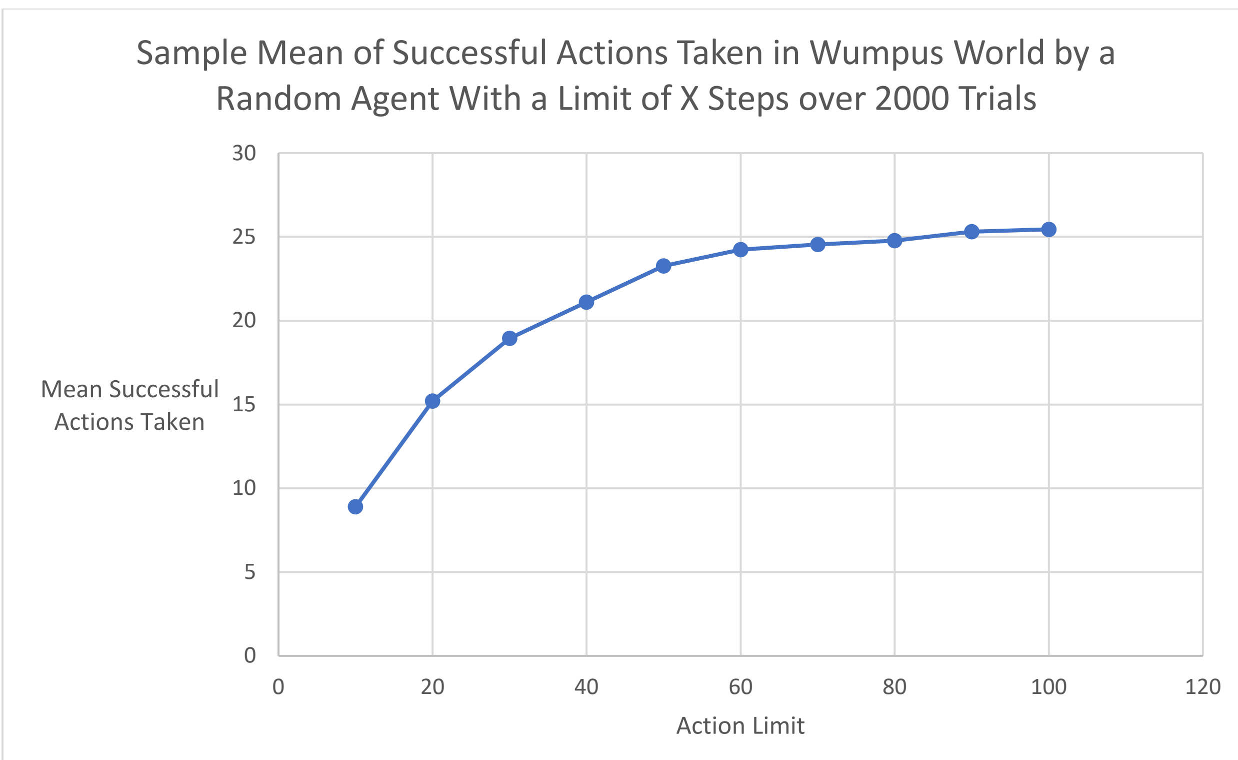


Fig B.

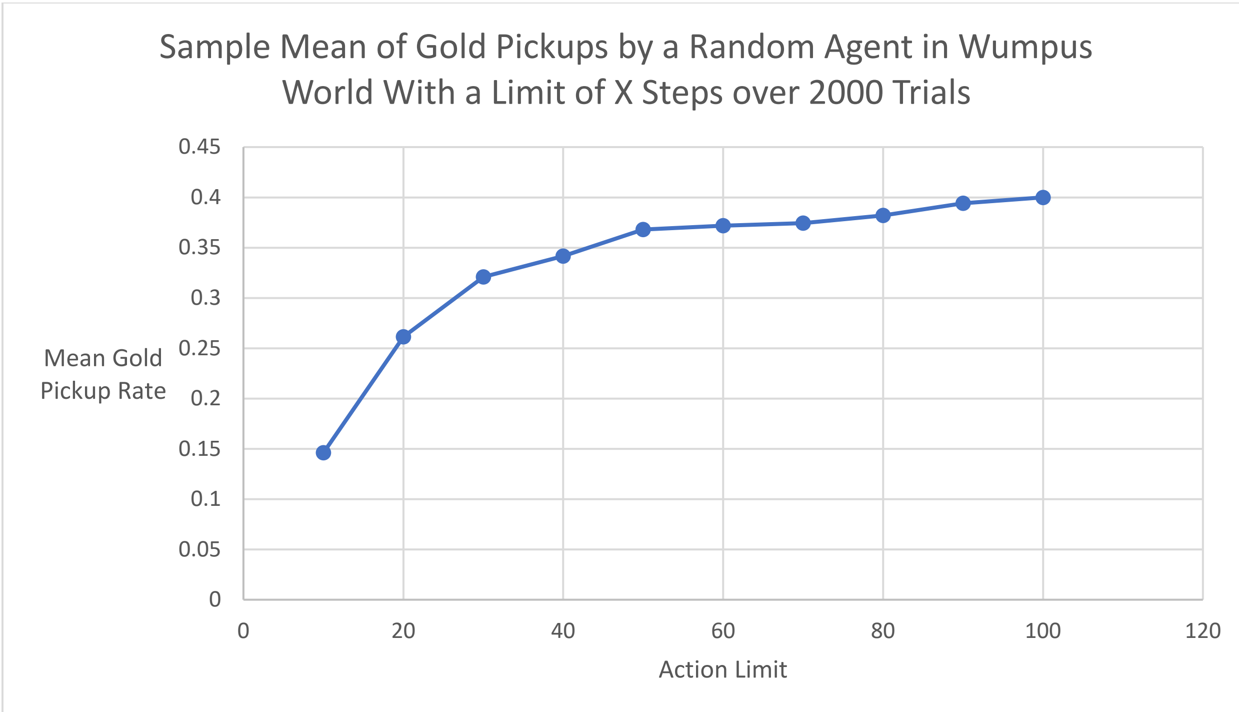


Fig C.

Action Limit	Mean	Variance	Confidence Interval	Gold Pickup Rate
10	8.8905	4.89851	0.097000297	0.146
20	15.1965	36.53789	0.264918808	0.2615
30	18.946	95.83808	0.429052202	0.321
40	21.1035	171.3488	0.573695696	0.3415
50	23.264	243.2273	0.68351372	0.368
60	24.233	313.5187	0.776019807	0.372
70	24.548	366.5107	0.839043351	0.3745
80	24.78	390.2186	0.865755096	0.382
90	25.311	415.6243	0.893493769	0.394
100	25.4515	435.3986	0.914501899	0.4

Fig D.

As I anticipated, the random agent was able to pick up the gold, albeit at a very low rate at low action limits. At low action limits, the agent also had a fairly high survival rate (Almost 90% at 10 actions). Furthermore, the mean of both the successful actions taken as well as gold pickup rate increased logarithmically as the action limit increased. It should be noted that at action limit 10, the pickup rate was only 14.6%, but the survival rate was 88.9%. At action limit 100, the pickup rate was at 40%, but the survival rate was only 25.45%. I never defined any actual optimal ratios, but it would seem to me that somewhere between 10-20 actions would result in a good pickup to survival ratio. Anything beyond this area certainly increases the pickup rate, but only at a marginal rate compared to the drop in survivability.

5. Interpretation

Ironically, a random agent is still pretty predictable over a large amount of trials if not a single trial. However, it is predictably bad. Increasing the number of available actions barely dented the pickup rate after a limit of 30 or so, whereas the survival rate dropped from > 50% to just above 25%. There are some other environments that a random agent might fare better in though. The random agent would likely embarrass itself much less in an environment without as many hazards. If this were true, it would be a much better use of time for AI programmers to simply create a random agent.

6. Critique

Calculating the survival rate for each action limit would have been a nice addition to some otherwise very nice-looking data. It was nice to know that the means dropped off eventually, but having a smaller delta action limit would have given better insight into the changes at lower action limits. In addition to all this, setting up a concrete null hypothesis should be a part of any good experiment and I will make sure to do that next time.

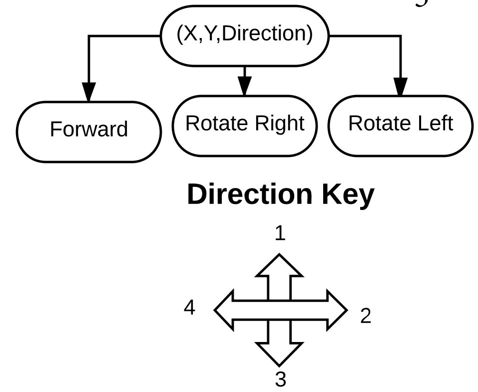
7. Log

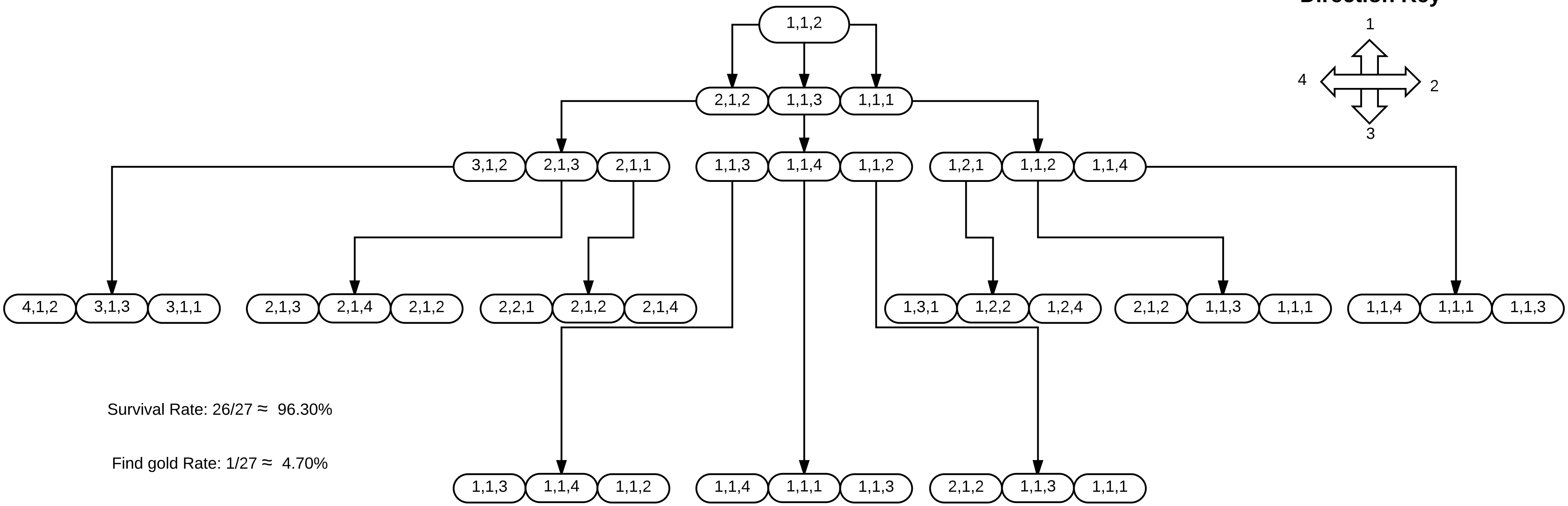
I spent approximately 1.5 hours analyzing the code and learning new concepts, functions, etc. for Matlab so that I could understand what was already being done for me and write my own code proficiently. I spent approximately 2 hours writing my own code and collecting data. I spent approximately 1.5 hours writing this lab report and performing verification.

Die on (1,3,Z), Survive elsewhere

Find Gold on (2,2,Z). Poor elsewhere

Locations and Directions of a Uniformly Distributed Random Agent After 4 Actions





Survival Rate: 26/27 ≈ 96.30%

Find gold Rate: 1/27 ≈ 4.70%

Fig E.