

CS-171 Wumpus World Final AI Report

Team name NAXavierDog

Member #1 (name/id) Jiliang Ni (81224742) Member #2 (name/id or N/A) Yufei Wang (13911282)

I. In about 1/2 page of text, describe what you did to make your Final AI agent “smart.”

In fact, when we consider how to face the unobservable map, we let agent move along a fixed and pre-set route, and when agent meets breeze, stench, bump, it will make the corresponding action. Whenever agent finds gold, it can back to origin along the shortest route. If agent does not find gold, then it will walk along the fixed route until that there is no any safe node left.

However, I think the smartest idea of our AI is not about how to deal with meeting breeze, stench and bump, but the design of the pre-set and fixed route. For this route, at the beginning, we let the agent move forward along the initial direction (right) until it meets breeze or bump, if there is stench, just shoot then move. Until meeting first breeze or bump, we have a visited straight line, this line is called “main”. All nodes above this line are safe and not visited, because before meeting first breeze, all nodes in main are empty, which means there is no pitfall above these nodes.

Then, each visited node in “main” can be a new origin, from this origin, we let agent walk vertically and upward until the agent meeting the first breeze or bump. This line whose origin is each node in “main” and is perpendicular to “main” is called “part”. So the agent will walk along all “part” of the “main”, until finding gold or back to the origin of “main”. We use a “comb” metaphor for this “search algorithm”, because the “main” is like the handle of the “comb” and the “part” is like the “tooth” on the “comb”. Of course, the “teeth” on this “comb” are not necessary to have same length. Until now, we just do this search once, this is not enough.

Therefore, after the first “comb” search, we let the agent which is at the origin faces upward, because we will let agent do the second “comb” search, and this time, the new “main” is a line which has same origin with the old “main” but is perpendicular to the old “main”. Although the agent will visit many visited nodes, it will also visit many new nodes which the agent can not visit during the first “comb” search, like some nodes behind breeze. Just imagine, when two “combs” vertically overlap each other, most of nodes will be visited, the benefit brought by this “search algorithm” is much greater than the cost this algorithm takes.

What’s more, using the “comb” search, when the agent meets breeze, stench, or empty node, we do not require the computer to remember the states, so it does not need much memory. When agent meets breeze, it just turn left twice, and return by the way it came, rather than try to bypass the breeze, because for the nodes behind the breeze, there are too many unknow conditions we should deal with. But for those nodes behind breeze, we also do not worry that the agent will not visit them, because in the second “comb” search, the agent will visit those unvisited nodes.

In a word, this “comb” search algorithm does not require much memory from computer, it just needs some Boolean value to deal with breeze, stench, bump and gold.

II. In about 1/4 page of text, describe problems you encountered and how you solved them.

When agent meets some condition which requires it return by the path it came, like breeze, the agent has to turn left twice, then it can move forward. So at this time, we have to set a “flag” to note how many times the agent has turned. Otherwise, if the agent doesn’t know how many times it has turned, then the agent will not know what direction it faces. When the agent meets first stench, we will let it shoots. But if there is no scream, then it shows agent did not kill the Wumpus. But the agent still stand on the stench, and how the agent know if itself should move? So, we set a “justshooted” flag which tells the agent that itself just shot as last action, then the agent knows it can move forward now.

III. In about 1/4 page of text, provide suggestions for improving this project.

To improve our AI, I think that we can make more “subparts” on each “part” and each origin of “subpart” is every visited node on the “part”.

To improve the design of the project, it can change the limit of the number of arrows, and at same time, it can add more Wumpus into this map. And for grading everyone’s AI, do not only grade according to the average score, it can also grade by the complexity of the code, the time the computer needs to get the result, and the memory the computer needs to get the result