Minesweeper Final AI Report

Team name: MinAI
Member #1 (name/id): Sameer Ganesh Shinde/19644693
Member #2 (name/id): Vaibhav Yengul/ 91844112

I. Minimal AI

I.A. Briefly describe your Minimal AI algorithm. What did you do that was fun, clever, or creative?

0	0	1	1	1
0	0	1	M	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0

Fig-1

0	0	1	1	1
0	0	1	M	1
0	0	1	1	1
0	0	0	0	0
0	0	0	0	0

Fig-2

1. We are eliminating the selection/uncovering of visited nodes, making sure that every time getAction is called, a new Node is selected for an Action.
2. Created a structure for holding node information viz its percept number and Action status (flag, unflag, Uncovered) and initialized a 2-d array of this structure so that node info along with its location is present.
3. As shown in Fig-1, the nodes with unknown percept number (marked by ‘x’) are put in a queue of “to-be-visited” nodes. It’s because since there is a zero all the adjacent nodes are safe to be uncovered.
4. Similarly, for Fig-2, since we uncovered a node with Percept number 1, all adjacent nodes (marked by !!) have equal probability of being a mine. And since there exists only 1 mine in Minimal AI, the other nodes (marked by ‘x’) are safe to be uncovered and hence are added to the queue mentioned in the previous point.
5. For both points 3 and 4, before adding to the queue we have to make sure that, the location of the node is in bounds of the grid/arena.

If you open 0, that means all the neighbors of 0 are safe to uncover. You can simply add these neighbors to the actions to be taken queue of DFS. If you open 1, that means one of its 8 neighbors is a mine, so ignore these cells, uncover everything else	In this example, if we open 3,3 (yellow grid) we will not open all neighbors, because all of them can have one mine. But apart from these 8, everything else is safe. Therefore, we will uncover them. We will continue the same rule and we will uncover all except mine.

I.B Describe your Minimal AI algorithm's performance:

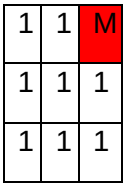
Board Size	Sample Size	Score	Worlds Complete
5x5	1000	1000	1000

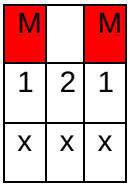
II. Final AI

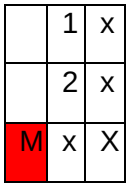
II.A. Briefly describe your Final AI algorithm, focusing mainly on the changes since Minimal AI:

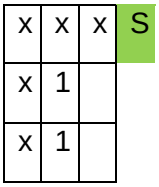
We added two important aspects to our logic:

1. Pattern Matching









2. Voting method:
   Requirements:
   - After uncovering all the zeros and attempting pattern matching, you can be stranded on a position where your algorithm doesn't have any idea of going forward, because there is nothing in the queue and none of the patterns matched. In such cases how to progress?

   Idea:
   - Maintain another grid of the same size called "Voting Grid"
   - After getting the value at a particular location on board, treat that value as the information for voting, i.e. suppose after uncovering (3,3) we found 1. Then it means, there can be one mine in (3,3) 's neighbors. So, we add 1 to all neighboring positions voting grid.
   - Similarly, when we uncover 0, it means definitely all its neighbors are safe and there is no mine. So, we overwrite the value of that cell in voting grid as 0. We can't update this value ever. Because we for sure know it's safe and can't contain a mine. Vote from 0 supersedes everything.
   - The intuition is to count how many people are saying this grid is a mine.

3. Random Restart:
   - In case if we don't have substantial information for any Cell in voting grid, then we apply random restart.
   - Through our observations we found, 4 corners are safe for 65% of times.
   - Therefore, whenever we are stuck after doing 3 methods, we apply random restart to any of the four corners of a grid.

II.B Describe your Final AI algorithm's performance:

Board Size	Sample Size	Score	Worlds Complete
5x5	1000	1000	1000
8x8	1000	746	746
16x16	1000	1354	677
16x30	1000	495	165
Total Summary	4000	3595	2588

III. In about 1/4 page of text or less, provide suggestions for improving this project (*this section does NOT count as past of your two-page total limit.*)

- We can better plan this project by having weekly guidance on coding strategies to be used while building Artificial Intelligent agent. A proper discussion on AI strategies (search, learning, etc.) that can be implemented is a necessity for this project
- It would be great if students actually learn how to implement strategies like mini max, alpha beta pruning and local search, else solve it using constraint satisfaction like Wumpus world in textbook.
- Guest lectures from industry experts will be helpful to understand real world scenarios of Artificial Intelligence.
- Project should support AI specific languages like Lisp, Prolog, etc.