CS 520: Introduction to AI
Wes Cowan

Assignment 2 - MineSweeper, Inference-Informed Action
Bonus: Dealing with Uncertainty

197001190: Aditya Lakra (al1247)
196003272: Hari Priya (hp467)
197001859: Twisha Naik (tn268)

Bonus-1

• *When a cell is selected to be uncovered, if the cell is 'clear' you only reveal a clue about the surrounding cells with some probability. In this case, the information you receive is accurate, but it is uncertain when you will receive the information.*

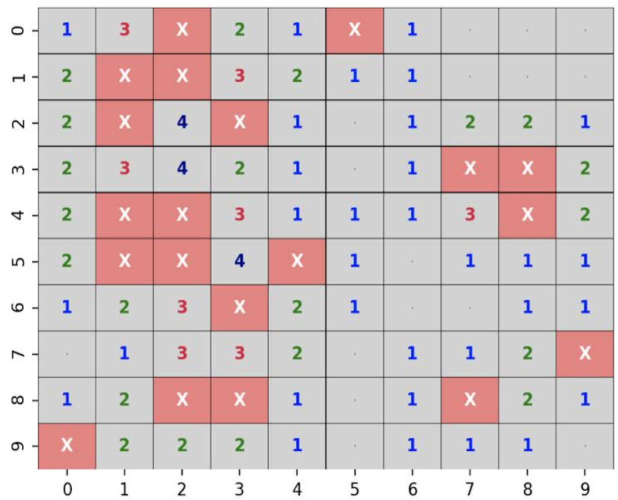
Code Specifications:

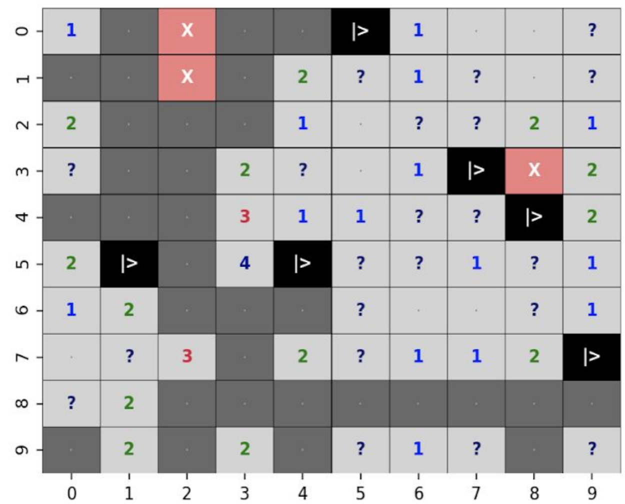
- Total mines are known
- Random picking uses average probability for a cell being a mine from the knowledge base
- Algorithm uses:

```
baseline logic + inference(subset logic) + advanced inference(intersection
logic)
```

New parameter:
Clue revelation probability ***clue_prob*** -> the probability with which we reveal the clue when safe cell is queried.
If clue_prob = 1 -> Normal case

Sample game board:





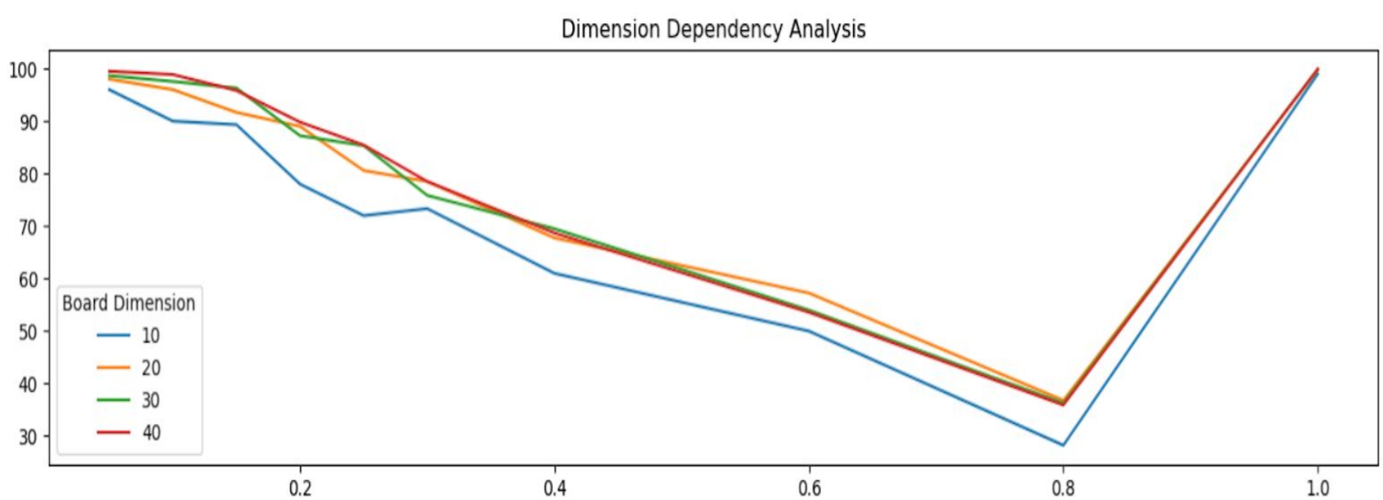
Left - Original game board

- ‘X’ are the mine cells
- Rest are the safe cells with proper clue values

Right - Agent board at some point in the game

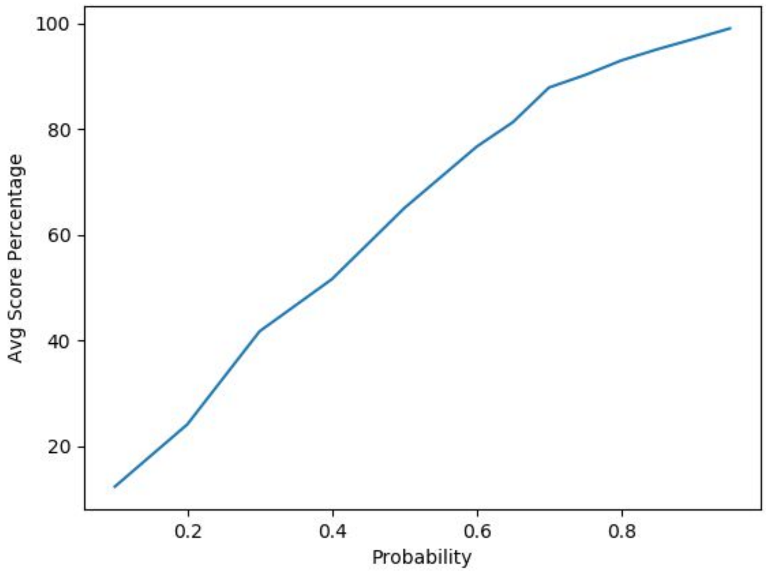
- |> - flagged mine cells (score is incremented for these cells)
- ? - safe cells whose clues haven’t been revealed
- ‘X’ - mine cells opened while random exploration
- Rest are the safe cells with correct clue values

Analysis 1: Analysis of varying dimension over a constant revelation probability.
[Below is the plot with the probability of revealing the cell clue = 0.75]
The mine density has been varied over the range of 0 to 1 for different dimensions.



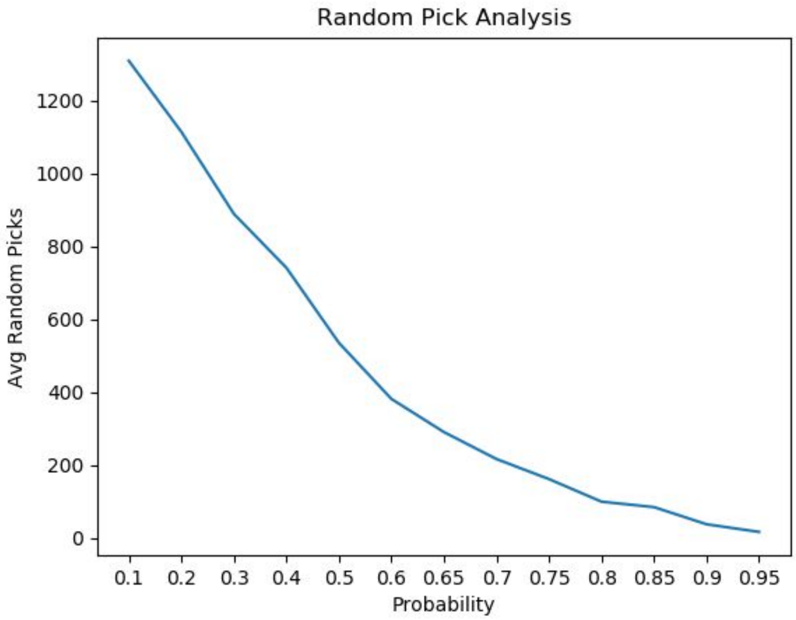
For the same density and revelation probability, the accuracy is the lowest for lower dimension.

Analysis 2: Variation of average score with respect to revelation probability



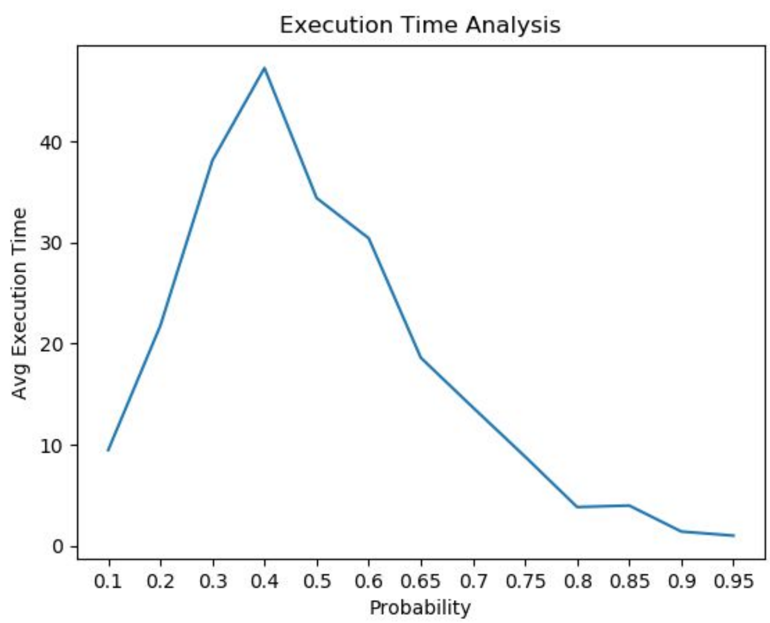
As we increase the probability for revealing the clue value at mine density 0.2, the average score increases as expected.
When the probability is 1, it behaves like a normal game where we always get the clue from safe cell.

Analysis 3: Variation of number of random picks with respect to revelation probability



When clues are not frequently revealed, we need more number of cells to be randomly explored to actually conclude something from the board.
Thus, with increase in the revelation probability, the average number of random picks reduces.

Analysis 4: Variation of execution time with respect to revelation probability

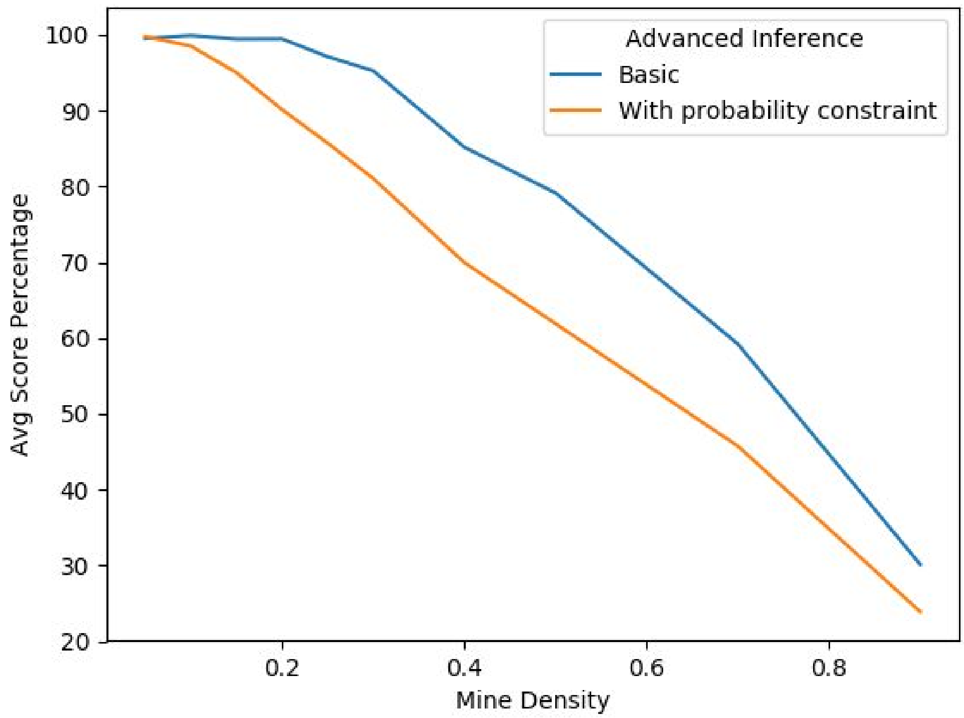


- The execution time of the algorithm increases initially as we keep increasing the revelation probability because initially when the probability of revealing clues is very low, the algorithm always picks cells randomly as it cannot infer anything. So the algorithm takes lesser time in exploring all the cells through inferences.

- As the probability keeps increasing, the number of clues it gains also increases. But the number of clues is still less when compared to the original algorithm without the revelation probability constraint. Therefore, the algorithm tries to infer as much as it can from the clues available, but fails in most of the cases due to insufficient knowledge and hence resorts to random picking again. Hence, the execution time also increases with increasing probability up until 0.4 to 0.5.
- Once the revelation probability is greater than 0.5, above 50% of the safe cells reveal clues and as a result, we have more knowledge available to make inferences and these inferences are also made faster due to sufficient amount of information. As a result, the execution time starts decreasing when revelation probability starts increasing after 0.5.

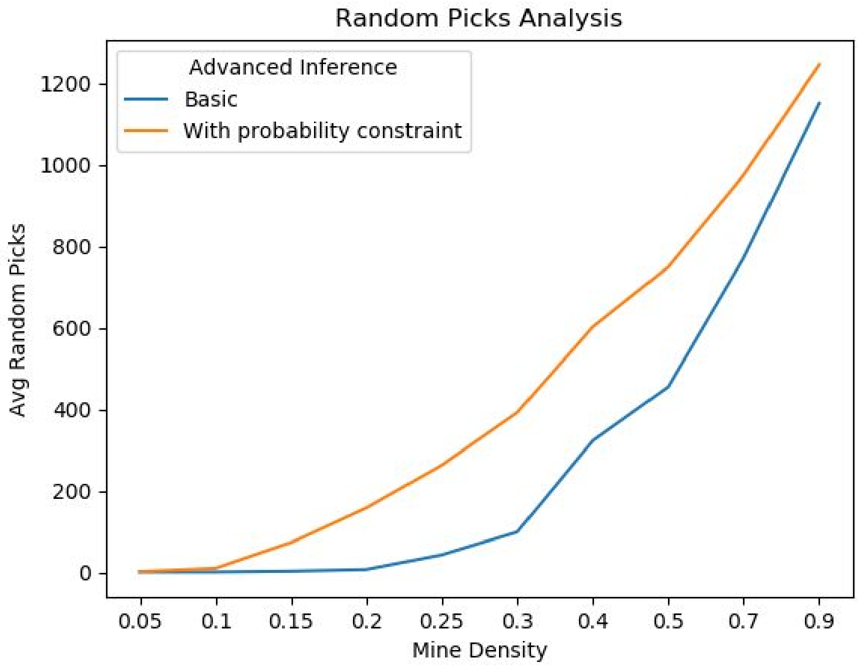
Analysis 5: Comparison of the normal algorithm with the modified algorithm
Note - For all the plots below, ***clue_prob=0.75*** has been used

5.1 Comparison of Average score over varying mine density



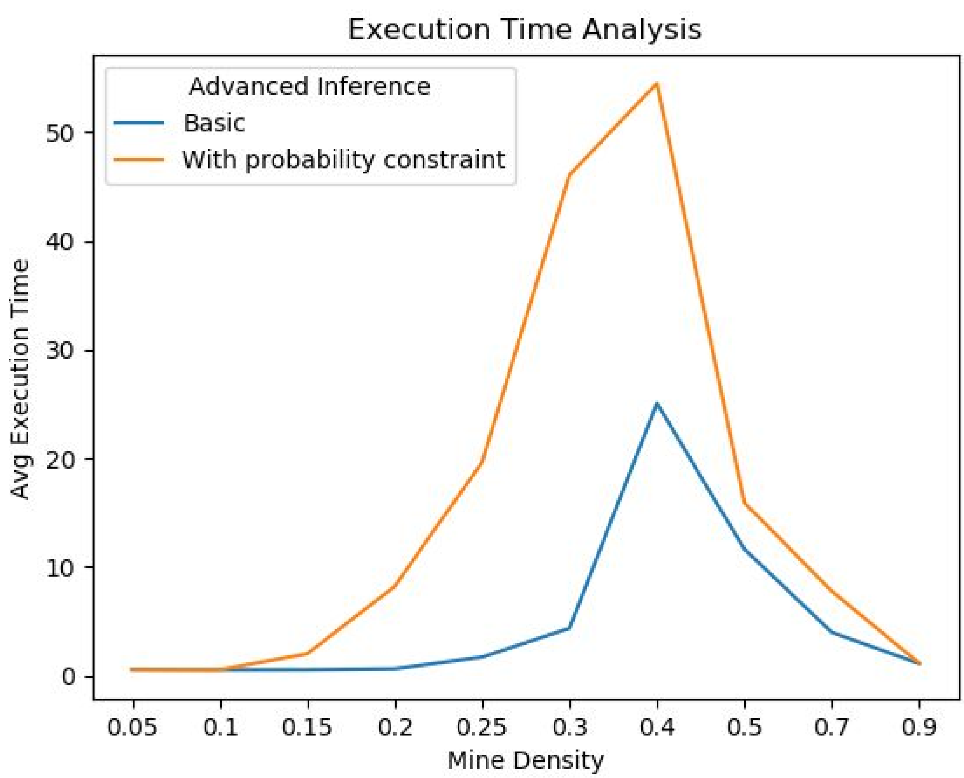
The score percentage is lesser for the modified algorithm because it has lesser number of clues as compared to the normal algorithm and it keeps picking cells randomly as it cannot make many inferences.

5.2 Comparison of random picks over varying mine density



The modified algorithm resorts to random picking more number of times than the normal algorithm due to lack of clues and inferences.

5.3 Comparison of execution time over varying mine density



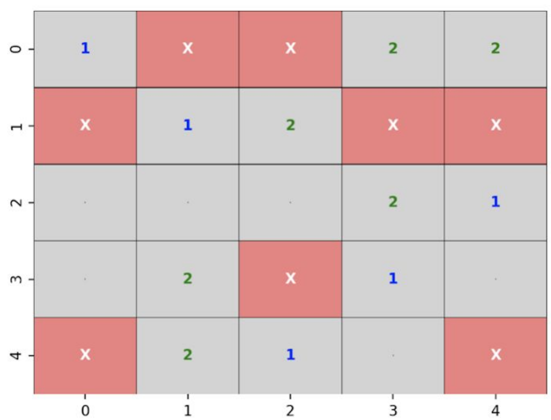
Bonus-2

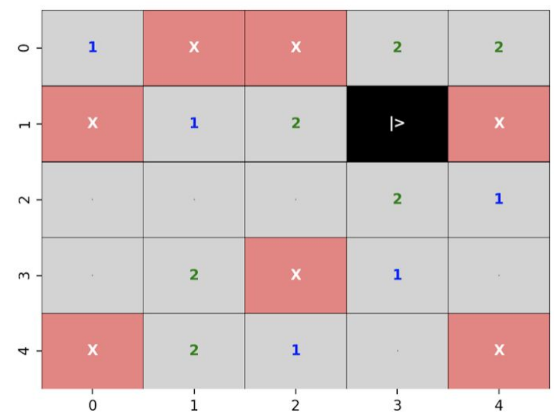
• *When a cell is selected to be uncovered, the revealed clue is less than or equal to the true number of surrounding mines (chosen uniformly at random). In this case, the clue has some probability of underestimating the number of surrounding mines. Clues are always optimistic.*

Code specifications:

- Total mines are not known to the agent (because of this, all the equations in the knowledge base are in the form of inequality)
- Random picking occurs based on the average likelihood of a cell being a mine
- The clue is equal to or less than the actual mines -- while board initialization, a value from [0, actual clue] is randomly picked with uniform probability
    - 0<= optimistic clue <= actual clue

Sample:





As the clue cells are optimistic, now we will get the equations of the form:

clue <= actual mines

A+B+C+D >= clue

[Can be seen in the figure above]

Direct inferences like we used to make in the normal case are no longer applicable here. We cannot reduce the equations using the subset check.

A+B+C+D>=2

B+C>=1

The subtraction can not be performed here.

Only mine cells can be found from these set of equations. Safe cells can never be determined with surety.

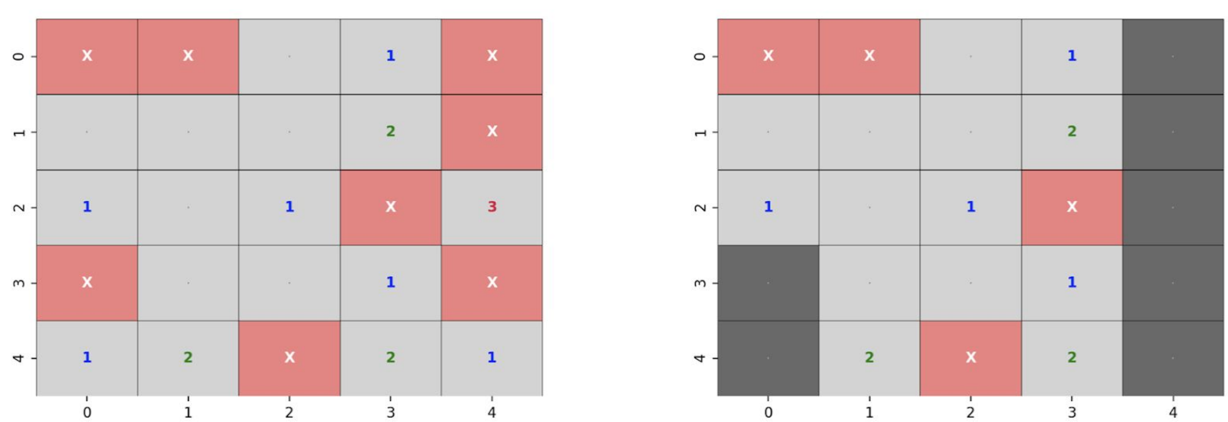
The only case where inference can be made is when optimistic clue value on the right of the equation is equal to the number of hidden cells on the left.
A+B+C+D+E>=5 means that all the 5 cells are mines.

```
Var1, Var2, ....…, VarN >= N implies that all the N variables are 1
```

One of the equation updates in this case is, if a variable set is a subset of other variable set and if the value associated with the subset is larger than the superset, we update the value of superset to the value of its subset.
Example: A+B+C+D+E>=2 and B+C+D+E>=3 implies A+(B+C+D+E)>=3

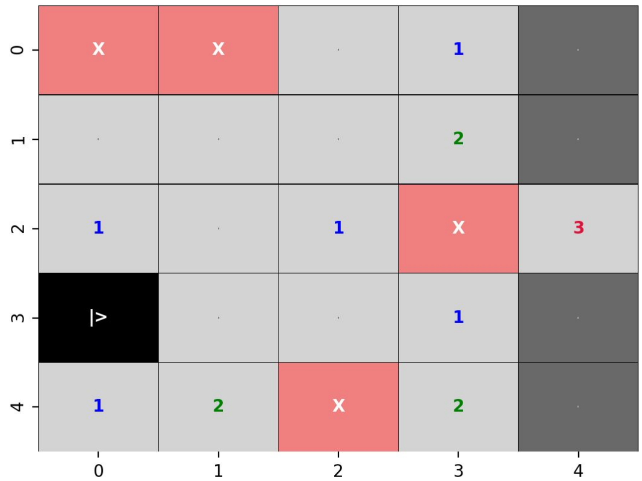
Approach:
We try to query the cells whose at least one neighbor is open. Because only querying the safe cells which are present in the knowledge base will lead to reduction of a variable from existing equations and might lead to some conclusion.
Opening a new random cell whose no neighbor is explored will lead to a conclusion only when its value reveals 8 (or the max number of neighbors it has) - this is highly unlikely.
Hence, we only open the cells whose at least one neighbor is explored.

Step by step run of a sample:

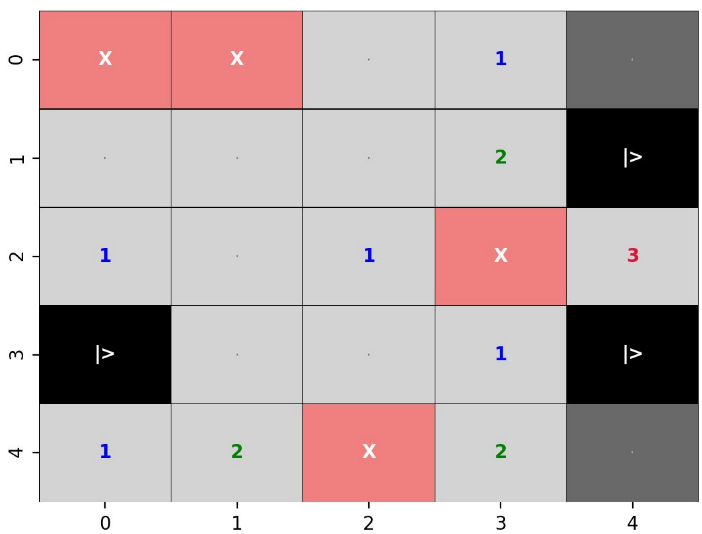


Left - Underlying game board

After this step, we have the equation (3,0) >=1 implies **(3,0)=Mine**
Hence, the algorithm flags (3,0) as mine.



At this step, we can conclude that (1,4) + (3,4) >= 2 implies both **(1,4) and (3,4) are mines.** This is what the algorithm deduces.

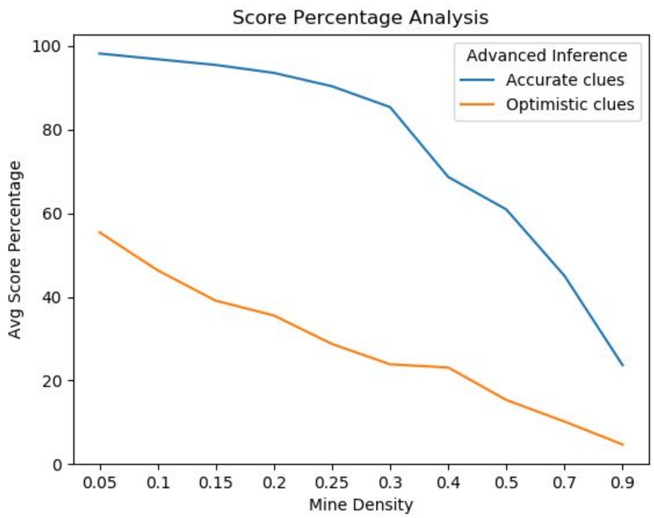


Key Observations:

- All the clues are <= original mines
- Mine cells can be determined in the cases where clue is equal to the number of unknown cells around it.
- No cell has zero unexplored neighbors. Every open cell has at least one explored neighbor.

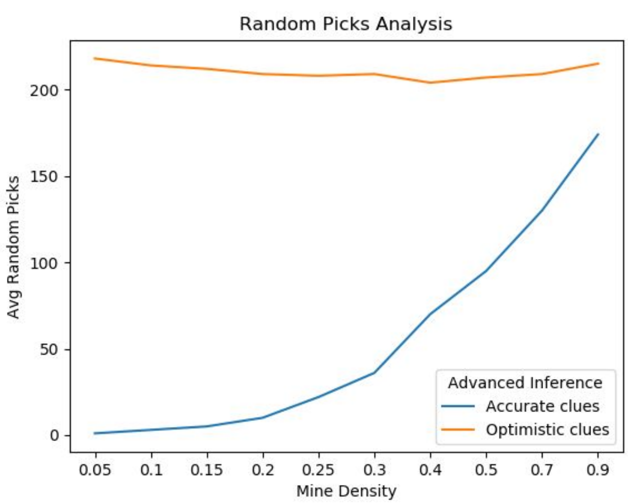
Analysis: Comparison of the modified algorithm with the normal algorithm (advanced inference using average likelihood without total number of mines) over a range of mine densities from 0 to 1.

1. **Comparison of Score Percentage**



The average score percentage of the algorithm with optimistic clues is always lesser than the normal algorithm because we are never sure if a cell is safe. Hence we need to keep picking cells which have the least likelihood to be mines to make useful inferences from the knowledge base. But this sometimes results in picking mine cells which reduces our score percentage.

2. **Comparison of Random Picks**



As the clues in the modified algorithm have less information and do not allow us to identify the safe cells, we need to keep picking cells randomly until we can deduce something useful from the knowledge we have available. This results in a higher number of random picks for the modified algorithm than the normal algorithm.

Bonus-3

• *When a cell is selected to be uncovered, the revealed clue is greater than or equal to the true number of surrounding mines (chosen uniformly at random). In this case, the clue has some probability of overestimating the number of surrounding mines. Clues are always cautious.*

Code specifications:

- Total mines are not known to the agent (because of this, all the equations in the knowledge base are in the form of inequality)
- Random picking occurs based on the average likelihood of cell being a mine
- The clue is greater than the actual mines -- while board initialization, a value from [actual clue, max neighbors] is randomly picked with uniform probability
    - actual clue <= cautious clue <= max number of neighbors for a cell

As the clue cells are cautious, now we will get the equations of the form:

clue >= actual mines

A+B+C+D <= clue

Direct inferences like we used to make in the normal case are no longer applicable here. We cannot reduce the equations using the subset check.

A+B+C+D<=2

B+C<=1

The subtraction cannot conclude anything.

Only safe cells can be found from these set of equations. Mine cells can never be determined with surety.

The only case where inference can be made is when the clue value on the right of the equation is 0.

A+B+C+D+E<=0 means that all the 5 cells are safe cells.

```
Var1, Var2, ....…, VarN <= 0 implies that all the N variables are 0
```

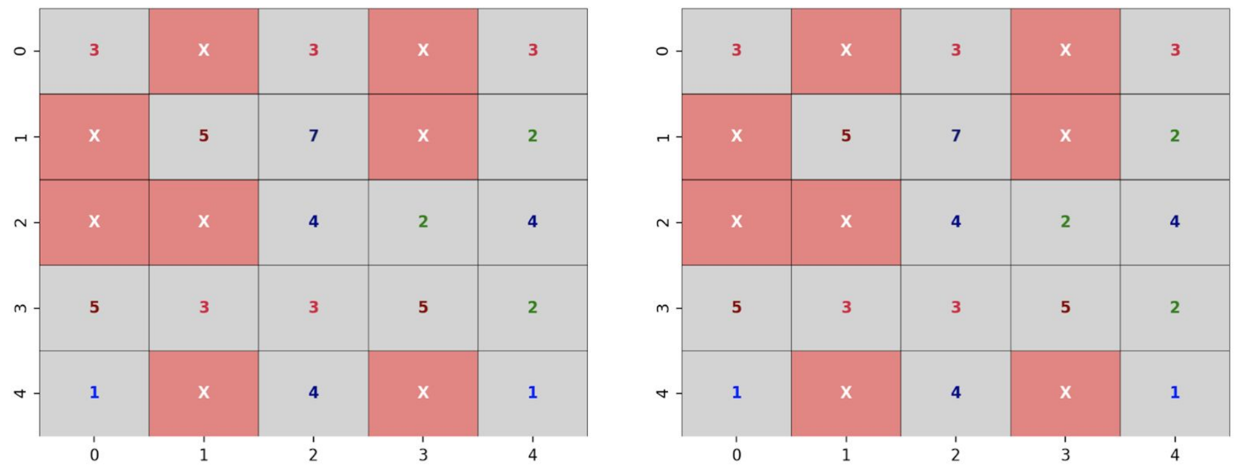
One of the equation updates in this case is, if a variable set is a subset of other variable set and if the value associated with the superset is lesser than the subset, we update the value of subset set to the value of its superset.

Example: A+B+C+D+E<=2 and B+C+D+E<=3 implies B+C+D+E<=2

Conclusion:

The accuracy we get here will always be 0. Because we never identify mine cells.

Note - We can get an accuracy >0 if we know the total number of mines on the game board.



Left - original board

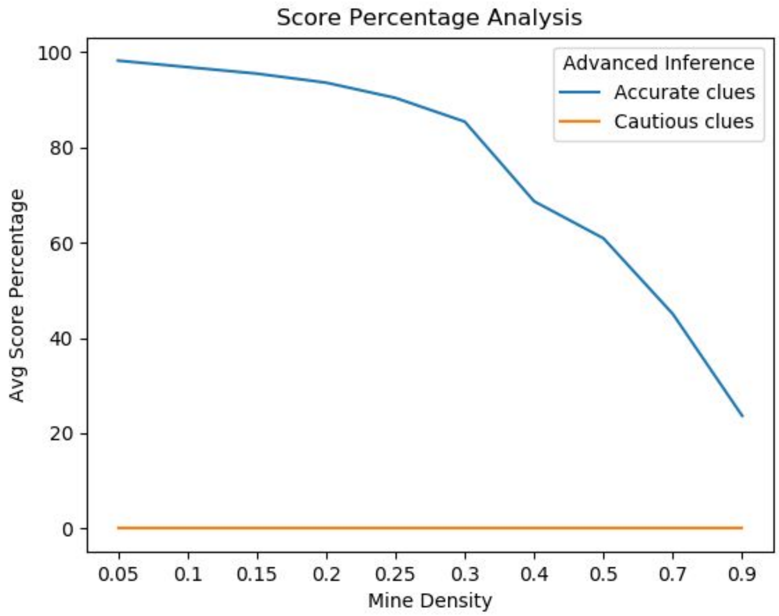
Right - solved board

Key Observations:

- All clues >= original number of mines around the cell
- None of the mine cell is marked in the final solved board -- because we cannot determine mine cells using the given information. Information about the total number of mines can help increase the accuracy.

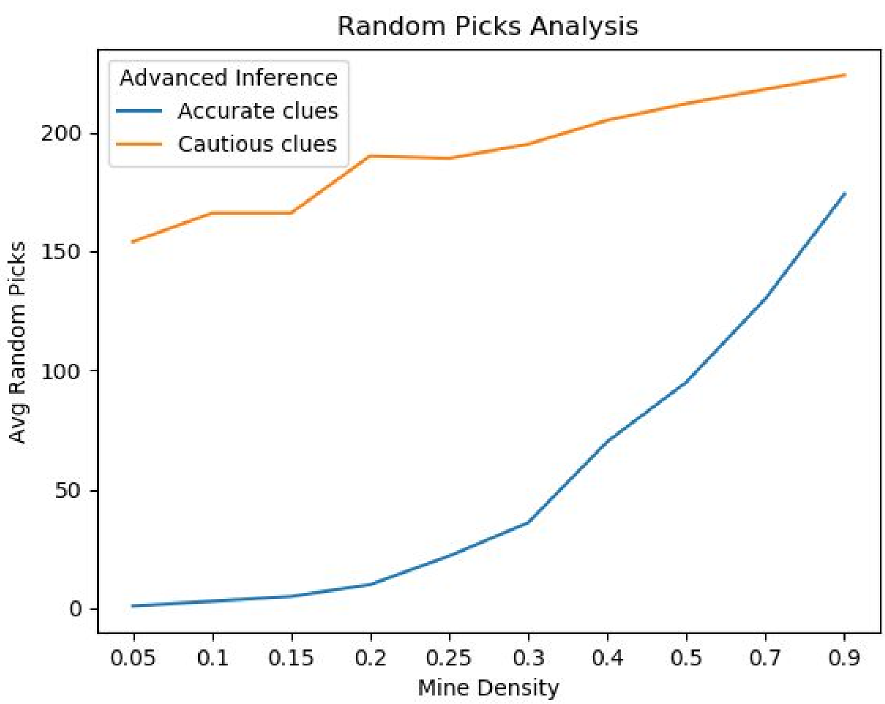
Analysis: Comparison of the modified algorithm with the normal algorithm (advanced inference using average likelihood without total number of mines) over a range of mine densities from 0 to 1.

1. **Comparison of Score Percentage**



The score percentage of the modified algorithm with cautious clues is always 0 as this algorithm can never detect mine cells unless we have prior knowledge about the total number of mines.

2. **Comparison of Random Picks**



Due to less informative clues, the algorithm picks random cells more number of times than the normal algorithm.