CS 520: Assignment 3 – Probabilistic Search (and Destroy)

Group members:

Name	netID	Work
Yan GU	yg369	Analysis/Write-up
Yunfan LI	yl1269	Probabilistic Search ALG
Xueyu WU	xw318	Theoretical Derivation/Write-up
Junlin LU	jl2364	Environment Setup/Locating Targets ALG

1. Program Specification:

1) **Environment setup**
Maps are generated according to probabilities of different terrain types (flat with probability 0.2, hilly with probability 0.3, forested with probability 0.3 and caves with probability 0.2). In order to test the correctness of Probabilistic Search ALG, we establish 5 by 5 grids at first. Then we construct a number of 50 * 50 maps to analyze performances of different decision strategies.

2) **Locating a moving target**
In section 2, the target can move between neighboring cells. It is worthy to speculate where the target might be, rather than search a cell randomly which may cost more actions to reach the goal. We can infer the position of the target according to surveillance reports, which give us certain patterns of target's moving path. After a few of times, we may get the exact location of the target or several possible locations to enlighten our search algorithm.

3) **Search algorithm**
We will discuss the theory and implement of Probabilistic Search ALG in detail in the following part.

2. A Stationary Target

1) **Given observations up to time t ($\mathbf{Observations}_t$), and a failure searching $\mathbf{Cell}_j$ ($\mathbf{Observations}_{t+1} = \mathbf{Observations}_t$ ^ $\mathbf{Failure\ in\ Cell}_j$), how can Bayes' theorem be used to efficiently update the belief state, i.e., compute:**

$$\mathbf{P(Target\ in\ Cell}_i \mid \mathbf{Observations}_t \text{ ^ } \mathbf{Failure\ in\ Cell}_j\mathbf{)}:$$

Initially, because likelihoods of each cell containing the target are equal, so we assign the probability for every cell is $1/(50*50) = 0.0004$. After we searching the target in a random cell (Cell_i) once and not finding the target, we update the current belief of this cell (which is not 0 because of the false negative)

$$\text{Current P(Target in Cell}_\text{i}\text{)} = \text{P(Target in Cell}_\text{i}\text{ ^ Failure in Cell}_\text{i}\text{)}$$

which is

$$\text{Current P(Target in Cell}_\text{i}\text{)} = \text{P(Target in Cell}_\text{i}\text{)} * \text{P(Failure in Cell}_\text{i}\text{ | Target in Cell}_\text{i}\text{)}$$

In addition, we can also infer new beliefs of other cells through a failure of search. Because

$$\text{Current P(Target in Cell}_\text{i}) < \text{P(Target in Cell}_\text{i}\text{)}$$

for $\text{P(Failure in Cell}_\text{i}\text{ | Target in Cell}_\text{i}) < 1$, we can conclude that the belief of all other cells should increase in order to match $\sum \text{P(Target in Cell}_\text{j}) = 1$. In particular, we scale up each probability of Cell_j based on its original belief, which means that the belief of a cell with higher probability will increase more than others:

$$1 + \frac{\text{P(Target in Cell}_\text{j})}{1 - \text{P(Target in Cell}_\text{i})} * [\text{P(Target in Cell}_\text{i}) - \text{Current P(Target in Cell}_\text{i})]$$

so the current probability of Cell_j is

$$\text{Current P(Target in Cell}_\text{i}) = \text{P(Target in Cell}_\text{j}) * \{1 + \frac{\text{P(Target in Cell}_\text{j})}{1-\text{P(Target in Cell}_\text{i})} * [\text{P(Target in Cell}_\text{i}) - \text{P(Target in Cell}_\text{i}) * \text{P(Failure in Cell}_\text{i}\text{ | Target in Cell}_\text{i})]\}$$

By modify beliefs of all cells, we can get a new belief state according to observations from the past, which is the current status for the next search.

2) **Given the observations up to time t, the belief state captures the current probability the target is in a given cell. What is the probability that the target will be found in Cell_i if it is searched:**

$$\textbf{P(Target found in Cell}_i\textbf{ |Observations}_t\textbf{)?}$$

Based on our discussion in the 1st question, we can generalize $\text{P(Target in Cell}_\text{i})$ and $\text{Current P(Target in Cell}_\text{i})$ using observation in time t

$$\text{P(Target in Cell}_i\text{ |Observations}_{t-1}) = \text{P(Target in Cell}_\text{i})$$

$$\text{P(Target in Cell}_i\text{ |Observations}_t) = \text{Current P(Target in Cell}_\text{i})$$

so

$$\begin{aligned}&\text{P(Target found in Cell}_i\text{ |Observations}_t)\\ &= \text{P(Target in Cell}_i\text{ ^ Target found in Cell}_i\text{|Observations}_t)\\ &= \text{P(Target in Cell}_i\text{ |Observations}_t) * \text{ P(Target found in Cell}_\text{i}\text{ |Target in Cell}_\text{i})\end{aligned}$$

In particular,

$\text{P(Target in Cell}_i\text{ |Observations}_t)$ is the result of our calculation based on observation through time 1 to t, the base case $\text{P(Target in Cell}_i\text{ |Observations}_0) = \frac{1}{\text{\# of cells}}$, which is 0.0004 in 50*50 grid.

$\text{P(Target found in Cell}_\text{i}\text{ |Target in Cell}_\text{i}) = 1 - \text{P(Failure in Cell}_\text{i}\text{ |Target in Cell}_\text{i})$ which is 0.9, 0.7, 0.3 or 0.1 respectively if the terrain type of cell is flat, hilly, forested, or a maze of caves.

3) **Consider comparing the following two decision rules:**
Rule 1: At any time, search the cell with the highest probability of containing the target.
Rule 2: At any time, search the cell with the highest probability of finding the target.
For either rule, in the case of ties between cells, consider breaking ties arbitrarily. How can these rules be interpreted / implemented in terms of the known probabilities and belief states?
For a fixed map, consider repeatedly using each rule to locate the target (replacing the target at a new, uniformly chosen location each time it is discovered). On average, which performs better (i.e., requires less searches), Rule 1 or Rule 2? Why do you think that is?

Does that hold across multiple maps?

Rule 1 is what we state in Question 1. We choose cells always based on $P(\text{Target in Cell}_i)$, which is just the current belief state in the time t.

Rule 2 is what we discuss in Question 2. Probability of target found in certain cell is equal to the belief of target in this cell timing probability of target found in this cell if there is a target.

$$P(\text{Target found in Cell}_i) = P(\text{Target in Cell}_i) * P(\text{Target found in Cell}_i \mid \text{Target in Cell}_i)$$

After simulating 100 trials for each rule in a 50 * 50 grid, we get the following average numbers of searches for each rule. In flat and hill situations, Rule 2 performs better because the cell with the highest probability of finding the target is also the cell with high belief of containing the target. However, in cases of forest and cave, Rule 2 may have to search all less difficult types of terrain which might be less likely to have the target. On the contrary, no matter how hard it is to find the target, Rule 1 always searches the cell with the highest probability of having the target, so it costs less when the target in a difficult ares.

	Rule 1	Rule 2
Flat	3782	974
Hilly	2049	1280
Forest	4724	6121
Cave	10128	13110
Average	5170.75	5371.25

4) **Consider modifying the problem in the following way: at any time, you may only search the cell at your current location or move to a neighboring cell (up/down, left/right). Search or motion each constitute a single 'action'. In this case, the 'best' cell to search by the previous rules may be out of reach and require travel. One possibility is to simply move to the cell indicated by the previous rules and search it, but this may incur a large cost in terms of required travel. How can you use the belief state and your current location to determine whether to search or move (and where to move), and minimize the total number of actions required? Derive a decision rule based on the current belief state and current location and compare its performance to the rule of simply always traveling to the next cell indicated by Rule 1 or Rule 2. Discuss.**

New Rule: Search begins at a random cell (it will perform better from the center of grid, we apply this in the Section 2) and must travel to other cells to search with each unit distance traveled equal to 1 action. To choose which cell to search, we calculate a new probability of a potential search given the required number of actions needed to perform the search. To search the current location, 1 action is required. To search location 4 units away, 5 actions are required. So we create the rule to calculate as

$$\mathbf{P' = P * 0.99^{number\ of\ moves}}$$

By selecting the maximum probability, we ensure that our moves maximize the likelihood of finding the target per action spent. If the expected cell is too far away, we assume that it is not worth going that far, because we can easily consider that the cell too far away may need a lot cost and may not have a target we want. So, if we find the expected cell is too far away, we will still search the location we stand on until the P of that cell is bigger than the cell we are on, then we make a move. Under this rule, we expect our performance to be worse that Rule 1 and Rule 2, as we are now constrained by how far we must walk to search a location.

After simulating 10 trials (because of the limit of our hardware environment, we don't have enough time to perform more trials) for each rule in a 50 * 50 grid, we get the following average numbers of searches for each rule.

	Rule 1 search	Rule 1 move	Rule 2 search	Rule 2 move
Flat	3782	2980	974	2633
Hilly	2049	4483	1280	3324
Forest	4724	14893	6121	15267
Cave	10128	23287	13110	30478
Average	5170.75	11410.75	5371.25	12925.5

5) An old joke goes something like the following:

A policeman sees a drunk man searching for something under a streetlight and asks what the drunk has lost. He says he lost his keys and they both look under the streetlight together. After a few minutes the policeman asks if he is sure he lost them here, and the drunk replies, no, and that he lost them in the park. The policeman asks why he is searching here, and the drunk replies, "the light is better here".

In light of the results of this project, discuss.

The drunk man rule is consistent with Rule 2 in part 3. Though the man is confident that his keys are in the park, the man still chooses to search under the streetlight, where he knows a higher probability of successfully finding his keys if at all his keys were there. Though the light seems a higher probability to the man, if the keys have a minimal enough chance of being under the light to begin with, then the man's probability for finding his keys is still minimal. On the other hand, even though his keys may more likely to be in the park, the man is still discouraged by the rate of difficulty for finding keys under the dark. In common sense, this seems ridiculous, but just considering the probability in mathematics we will know this strategy make sense.

According to our results in part 3, the man's best action depends on the true probability of his keys being in the park or under the street light and the success rate of finding the keys in the park or under the street light given the keys are truly in the park or light. It is the same as the problem we solve.

So, we can conclude that if the probability of the keys being in under the light is much greater than the probability of the keys being in the park or if the probabilities are relatively equal, then it would be more efficient for the man to search under the light first. If the man loses his keys frequently, then his strategy of looking under the light first would slightly change. For one time, if he didn't find the key under street light in one search, he may consider moving on to the park or search under the street light again. Hence the ratio of cost of move is some number less than 1 and it may affect the probability of finding the key, so he can consider whether to move and how far to move, in our problem, the ratio is 0.95 accordingly. If the man wishes to also save time it takes to move to a search location and the man happens to be closer to the street light then the park, then it is more efficient for him to exhaustively search under the street light before moving on to the park that is farther away. Again, these conclusions were made under the assumption that the probability of the man losing his keys in the park or under the light are relatively equal. Therefore, we may think that the drunk man is a smart man rather than drunk in the mathematics way.

3. A Moving Target

1) **In this section, the target is no longer stationary, and can move between neighboring cells. Each time you perform a search, if you fail to find the target the target will move to a neighboring cell (with uniform probability for each). However, all is not lost - whenever the target moves, surveillance reports to you that the target was seen at a Type1 x Type2 border where Type1 and Type2 are the cell types the target is moving between (though it is not reported which one was the exit point and which one the entry point.**
Implement this functionality in your code. How can you update your search to make use of this extra information? How does your belief state change with these additional observations? Update your search accordingly, and again compare Rule 1 and Rule 2.

The initial belief state for each cell is also $\frac{1}{\#\text{ of cells}}$, which is 0.0004 in particular for 50*50 grid. After a failed search, we can get a report revealing the movement of target between 2 cells. According to this, we can infer that the belief of all other types of terrain must be 0. The target can only be in those cells with the same types as surveillance report, and their probabilities will increase proportionately. Based on this conclusion, we can update our belief state as following:

$$\text{Current P(Type of Cell}_i \text{ != Type1 or Type2)} = 0$$

$$\text{Current P(Type of Cell}_i = \text{Type1 or Type2)} = \text{P(Type of Cell}_i = \text{Type1 or Type2)} * [1 + \frac{\text{P(Type of Cell}_i = \text{Type1 or Type2)}}{\sum \text{P(Type of Cell}_j = \text{Type1 or Type2)}} * \sum \text{P(Type of Cell}_j \text{ != Type1 or Type2)}]$$

```
previous belief:                update belief:
 [[0.04 0.04 0.04 0.04 0.04]    [[0.16666667 0.16666667 0.16666667 0.16666667 0.16666667]
 [0.04 0.04 0.04 0.04 0.04]     [0.         0.         0.16666667 0.         0.        ]
 [0.04 0.04 0.04 0.04 0.04]     [0.         0.         0.         0.         0.        ]
 [0.04 0.04 0.04 0.04 0.04]     [0.         0.         0.         0.         0.        ]
 [0.04 0.04 0.04 0.04 0.04]]    [0.         0.         0.         0.         0.        ]]
```

*Figure 3-1 belief with moving target(5*5 grid)*

Like we discussed before, the performance here by using different rules is the same thing as stationary target. The only different is that: in addition to observation *target not found in* $Cell_i$, we also get the information of *the path* the target going through, which help us to pinpoint the target more efficient and easier.

	map1	map2	map3	map4	map5	map6	map7	map8	map9	map10	average
rule 1	5396	16529	2750	10746	13316	15264	4597	17605	10177	5247	10162.7
rule 2	5842	10380	10035	9469	5710	3359	5080	6084	1752	1533	5924.4

*Figure 3-2 average search for different maps(50*50 grid)*

As we can see at *Figure 3-2*, Rule 2 has a better performance in most cases. This may be a result of moving target. For Rule 1, it searches different types of terrain more evenly than Rule 2, so it more likely to select forest or cave which has higher probability of containing the target but is difficult to search. However, in this section, the target can move. Instead of searching forest or cave after completing all flat and hill like Section 1, Rule 2 now can easily locate target in less difficult terrains.

Not only on 50 by 50 maps, Rule 2 also has less searches on other maps with different dimension. Showed on *Figure 3-3*.

	dim=10	dim=20	dim=30	dim=40	dim=50
rule 1	170.6	596.1	4326.9	4614.5	10162.7
rule 2	65.4	263	3334.5	4028.3	5924.4

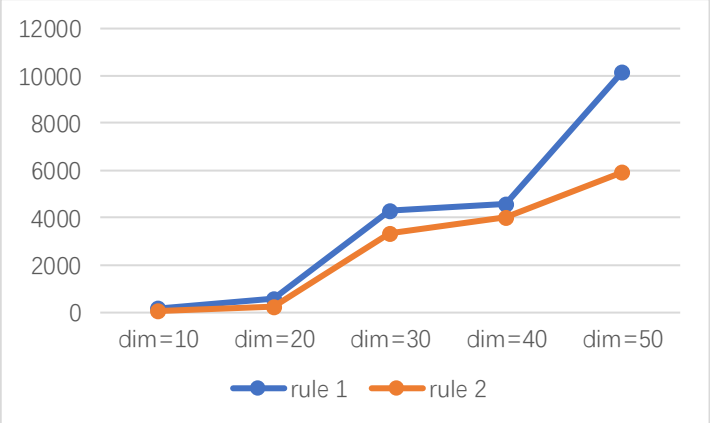

Figure 3-3 average search for different dims

2) **Re-do question 4) above in this new environment with the moving target and extra information.**

Similar with strategies used in stationary target situation, we have to decide whether we search current cell or move to others with higher probabilities based on Rule1 or Rule2. However, it is easier to make a decision: as we discussed in the Q1, if the current belief of this current cell is 0, we have no point to search it. The only choice is to move to the most promising cell.

At first, we choose to start from the center of the board because it has a relatively closest distance to all the cells from direction. We locate the cells with the highest probability, using Manhattan distance to find the closest among them, we move towards this cell. Every time we move to the next cell, if current cell has no chance of containing the target, we decide not to search, then the target will move, the current belief with change, we recompute the closest distance and decide our next move.

The total action require for the search is significantly larger than Rule1 and Rule2, in an average of 14000. However, if we only take the times of searching, we get a better outcome than Rule1 and Rule2. The average of searching times is below 5000 in a 50*50 board. Showed on Figure 3-4. We believe that it is because after several updates of belief, the program can generally determine which direction the target is located. So it will keep moving towards the target and narrow down times of searching.

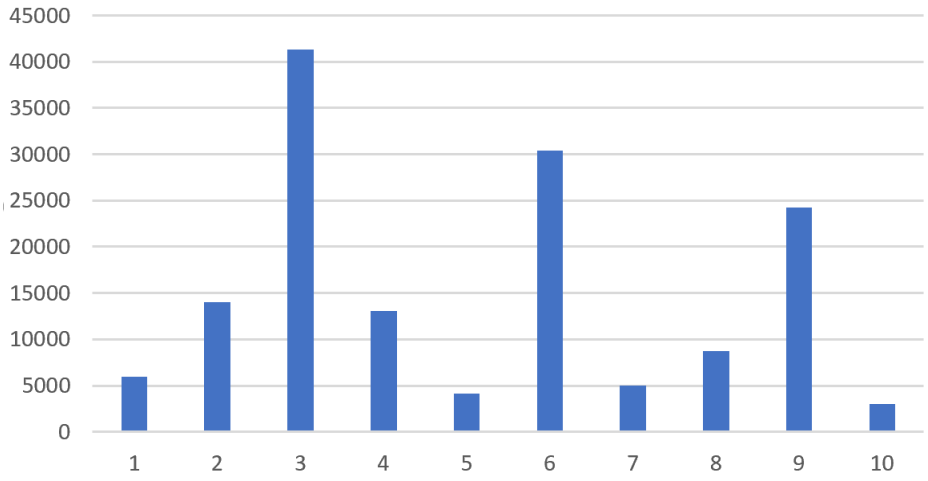

Figure 3-4 total actions in 10 different board

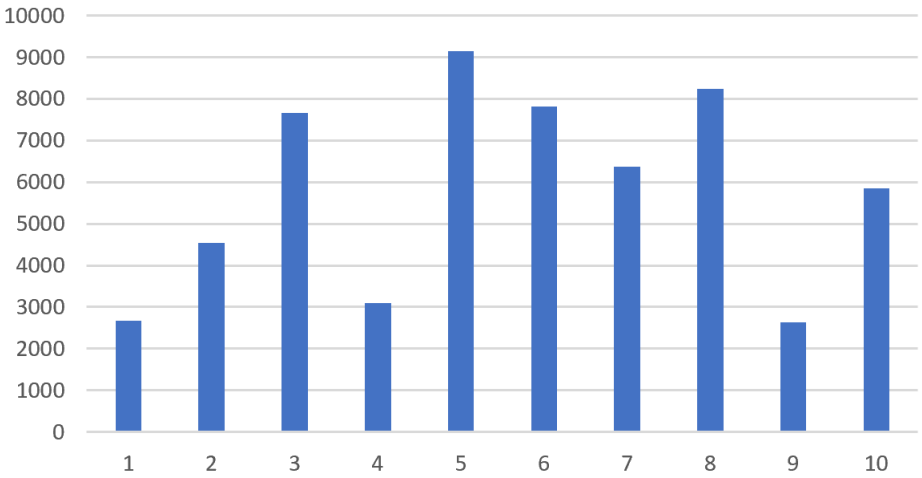

Figure 3-5 average search for 10 different board