CS520 Assignment3 – Probabilistic Search and Destroy

Group Member: Xinghao Wang(xw354), Tianfang Zhang(tz203) , Meng Ye (my389)

In the following work, in order to make it easier to describe, we first describe the events briefly as

T1 = 'Cell-i is flat'
T2 = 'Cell-i is hilly'
T3 = 'Cell-i is forested'
T4 = 'Cell-i is a complex maze'

X = 'Target in cell-x'
Y = 'Target not found in cell-x'

And we can immediately get

P(T1)=0.2, P(T2)=0.3, P(T3)=0.3,P(T4)=0.2 (1)
P(Y|X, T1)=0.1, P(Y|X, T2)=0.3, P(Y|X, T3)=0.7, P(Y|X, T4)=0.9 (2)
P(nY|nX)=0 (3)

In which, nY is not Y and nX is not X.

Also, we know that X and Ti (i=1,2,3,4) are independent. So

P(X, Ti)=P(X)P(Ti) (i=1,2,3,4) (4)

A Stationary Target

Q1)

We can use the Bayesian network over time (Figure 1) to model the searching for the target process.

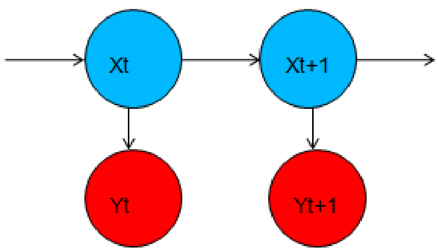


Figure 1. Bayesian network over time

Use the temporal estimation, we can derive the Belief as following

$$\mathrm{Belief}_{t+1}(x) = P(X_{t+1} = x|Y1, Y2, \dots, Yt+1) = \beta' \sum_{x'} \mathrm{Belief}_t(x')P(X_{t+1} = x|X_t = x')P(Yt+1|Xt+1 = x) \quad (5)$$

At initial state, we have

$$\mathrm{Belief}_0(x) = \frac{1}{\#\ \text{of cells}} \quad (6)$$

And

$$P(X_{t+1} = x|X_t = x') = \begin{cases} 1, \text{if} \quad x = x' \\ \quad 0, \text{else} \end{cases} \quad (7)$$

$P(Yt + 1|Xt + 1 = x)$ could be determined by looking up the table of (2) according to Ti (i=1,2,3,4) of cell x. In this way, we could update $\mathrm{Belief}_t(x)$ iteratively.

Q2)

For the probability of P(nY=x| $Y1, Y2, \ldots, Yt$), we have

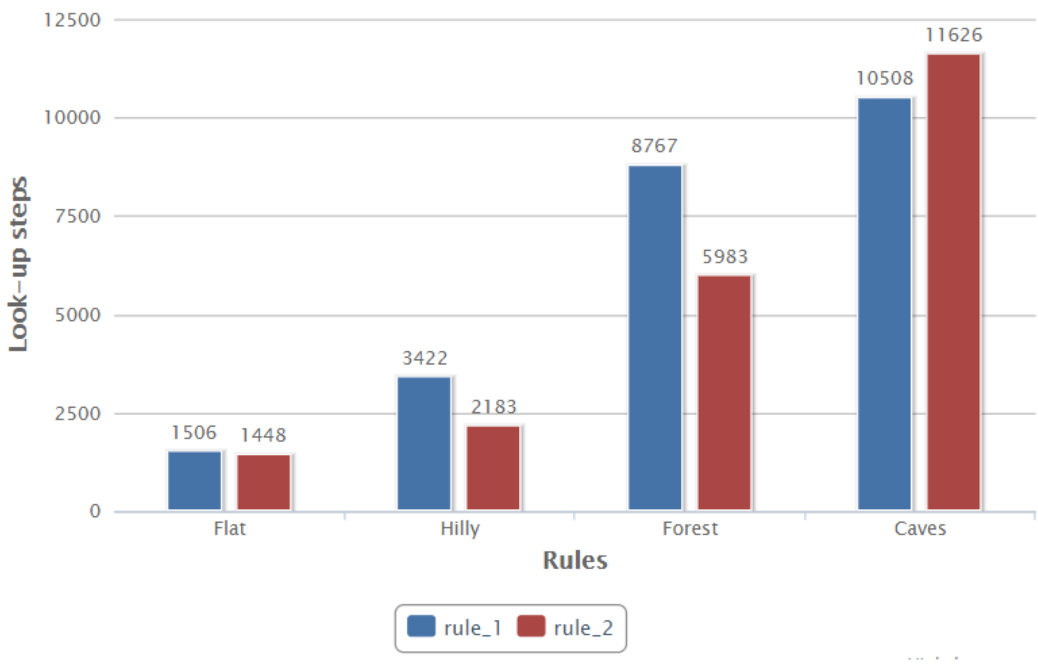
$$P(nY{=}x|\ Y1, Y2, \ldots, Yt) = 1 - P(Y|\ Xt + 1 = x, Ti)P(Xt + 1 = x)\ P(Ti)$$
$$=1 - P(Y|\ Xt + 1 = x, Ti)\ \mathrm{Belief}_{t+1}(x)\ \ P(Ti)\ (i{=}1,2,3,4) \quad (8)$$

Q3)

Using Rule-1 to search the target, we actually update $\mathrm{Belief}_t(x)$ iteratively and choose the highest probability location x as the search point. If there are several equal high probability locations, we randomly choose one location from them.

Using Rule-2 to search the target, we first update $\mathrm{Belief}_t(x)$ then compute P(nY=x| $Y1, Y2, \ldots, Yt$) to get the highest probability location x as the search point. If there are several equal high probability locations, we randomly choose one location from them.

Results:



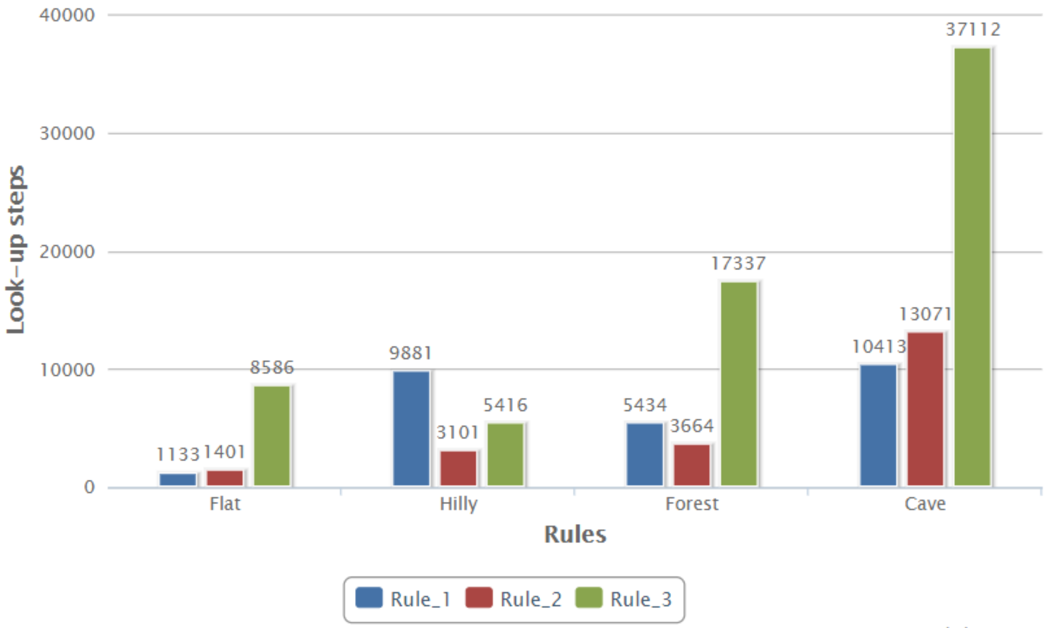
Q4)

We use the following Utility function

$$U(x) = \mathrm{Length}(x' \to x) + \frac{1}{\mathrm{Belief}_t(x)} \quad (9)$$

Where $\mathrm{Length}(x' \to x)$ is the path length from current location x' to each location x on the map. Every step, we update U(x) and find the smallest Utility location x as the next search location.

Results:



Q5)
Sometimes, the decision rule may not give a correct or expected direction for our target search due to not accurate probability estimation. It is clear that, the more accurate the probability estimation is, the less cost the search algorithm has. Rule-2 is more close to our final goal of finding the target than Rule-1 because of the false negative probability. So Rule-2 should have a better performance than Rule-1 theoretically and this is also validated by the previous experiments.

Bonus: A Moving Target

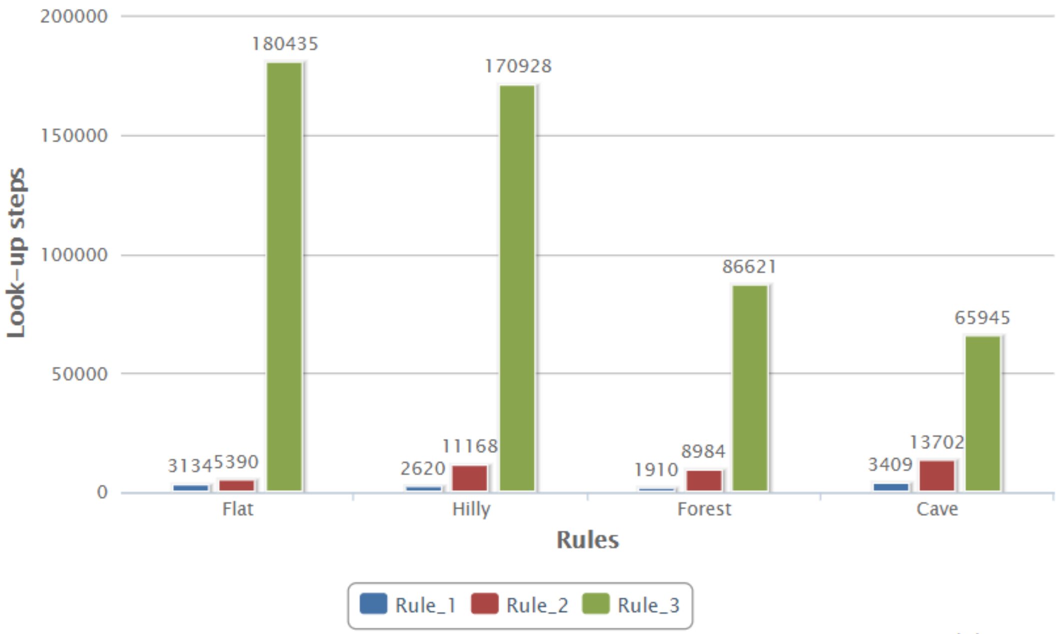
Since the target can move between neighboring cells, when update $\mathrm{Belief}_{t+1}(x)$, we have

$$P(X_{t+1} = x | X_t = x') = \begin{cases} \frac{1}{\#\ \text{neighbors of}\ x'}, \text{if} \quad x \text{ is a nrighbor of } x' \\ 0, \text{else} \end{cases} \tag{10}$$

And the feedback of the target could make us believe that

$$\mathrm{Belief}_{t+1}(x) = \quad 0, \text{if } x \text{ is of the terrain type which the feedback reports} \tag{11}$$

Now we can update $\mathrm{Belief}_{t+1}(x)$ with the moving target and the extra feedback information.
Results:



200000
150000
100000
50000
0
Look-up steps
180435
170928
86621
65945
3134
5390
2620
11168
1910
8984
3409
13702
Flat
Hilly
Forest
Cave
Rules
Rule_1
Rule_2
Rule_3