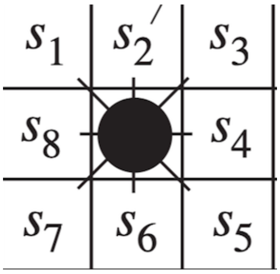
Problem 1:

Suppose the identity number of each sensor is same as the figure below and $S_i = 1$ if the corresponding cell is occupied:



Generally, the location of a robot can be divided into three categories:

I. All the sensors detect nothing ($S_i = 0$ for all sensors)

II. The robot reaches the boundary but not at the corner (one of S_2, S_4, S_6, and S_8 is 1, and the others are 0)

III. At the corner ($S_1 = S_2 = S_8 = 1$ OR $S_2 = S_3 = S_4 =1$ OR $S_4 = S_5 = S_6 = 1$ OR $S_6 = S_7 = S_8 = 1$)

For Type I, let the robot move north by default.
For Type II, let the robot go clockwise along the boundary.
For Type III, let the robot stay.

The production system is:
$S_1S_2S_8 + S_2S_3S_4 + S_4S_5S_6 + S_6S_7S_8$ -> stay
S_2 -> east
S_4 -> south
S_6 -> west
S_8 -> north
1-> north

Problem 2:
Suppose each input named X_1, X_2, X_3, X_4, X_5

The Boolean function is:
$(X_1 \cdot \neg X_3 \cdot \neg X_4) + (X_2 \cdot \neg(X_3 \cdot X_4)) + (X_1 \cdot X_2)$

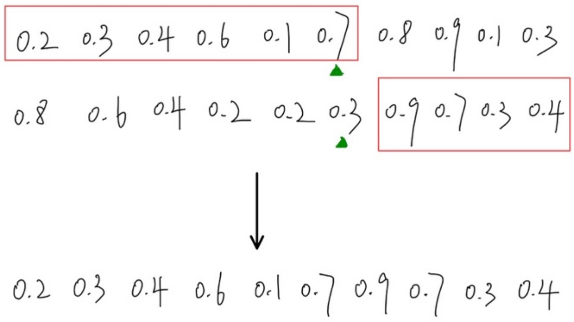
Problem 3:

1. Since it is a classification problem, I use Accuracy as the fitness function.

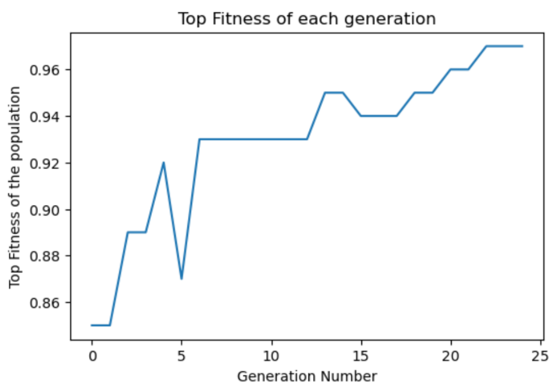
$$Accuracy = \frac{Number\ of\ True\ Prediction}{Number\ of\ Data}$$

2. The crossover operation (90%) as follows: a father perceptron and a mother perceptron are chosen from the current generation by the tournament selection process (10 perceptron are

randomly chosen from the population, and father/mother is the one with the highest/second highest fitness value. The combination of weights and threshold can be seen as a 1*10 vectors. Then, a cross over index i (the green triangle) is randomly chosen. And the 1_{st} to the i_{th} value of father perceptron's vector (the red part of the first vector) and the $i+1_{th}$ to the 10_{th} value of mother perceptron's vector (the red part of the second vector) are chosen to generate a child perceptron.



3. Copy 9% of the perceptron from generation n to generation n+1. These are chosen by the tournament selection process: 10 perceptron are randomly selected from the population, and the most fit of these ten is chosen.
4. Mutate 1% of the population. The process is: randomly choose a weight w_i, let $w_i = 1 - w_i$.
5. The size of population = 5000. In the first generation, for each perceptron, I randomly generate a 1*10 vector (all element $\in [0,1]$). The first nine elements are the weights of the perceptron and the last one is the threshold.
6. There are two conditions. I set a maximum iteration number and the threshold fitness value. For each generation, if the perceptron with the highest fitness value equal or greater than the threshold, the GP stops; OR GP stops when the number of generation no longer smaller than the maximum iteration number
7. It outputs the perceptron with the highest fitness value in the last generation. The result is 97% accuracy rate. The prediction is in file 'Q3_prediction.csv'
   Following figure demonstrates the development of the highest fitness value in each generation.



Problem 4:

1. My Perceptron:
   Weight: [1., -4., -4., 0., 0., 0., 0., 1.], threshold = 1.0
   Boolean expression : $X_1 \cdot \neg X_2 \cdot \neg X_3 + X_8 \cdot \neg X_2 \cdot \neg X_3$
2. The perceptron is located in Perceptron.py

Problem 5:

(2) No, since it is not a linearly separable function. In some circumstance, the Boolean function of the production system contains operators (e.g. XOR) more than linear operator (and, or, not).