# Assignment 1 Report

Shengzhe Liang (sl1487)
Shangda Wu (sw828)

02.23.2020

## Task 1. Problem Formulation

Environment: A matrix of maze
State Space: 5*5,7*7,9*9 or 11*11 array(int)
Actions: Visualize by number of moves(either move up ,right ,left or down according to the number in the current position)
Perception/Observations: state of numbers in the matrix
Transition Function: add steps in the matrix
Evaluation Metric: positive numbers for visited states
X for unvisited states
0 for start state
G for target state

## Task 2. Puzzle Representation

The software is able to receive as input the size of the puzzle n and then generate a random puzzle that contains only legals moves, and the “G” is the goal. For example:

```
Enter 5, 7, 9, or 11
5
The original matrix is:
2  4  1  1  1
1  3  3  1  1
1  1  2  1  2
3  2  3  3  3
3  3  4  2  G
```

```
Enter 5, 7, 9, or 11
7
The original matrix is:
2  6  6  1  1  3  2
2  2  2  3  4  3  6
4  1  2  1  4  5  2
4  1  4  1  4  1  3
6  4  1  3  1  4  6
2  4  4  5  3  1  3
5  3  6  3  4  4  G
```

## Task 3. Puzzle Evaluation

In this task we compute and visualize on our GUI the minimum distance to each cell from the start cell. On one hand, if there is no path, the evaluation function value will be -k, where k is the number of cells not reachable from the start. On the other hand, the evolution function value will be the value of the puzzle to the minimum distance from the star to the goal.

Example 1_1: (n = 5, solvable)

```
Enter 5, 7, 9, or 11
5
Task2: The original matrix is:
1  2  4  3  3
4  3  1  1  3
1  3  1  1  4
3  3  1  3  3
4  1  4  1  G


Task 3: Function evaluation
0  1  X  2  X
1  3  X  X  2
X  2  X  X  3
X  X  X  3  X
X  X  X  X  3
The value function is 3
```

Example 1_2: (n = 5, not solvable)

```
Enter 5, 7, 9, or 11
5
Task2: The original matrix is:
1  4  2  3  1
4  1  1  3  4
4  1  1  1  4
1  1  2  3  1
4  4  4  3  G


Task 3: Function evaluation
0  1  X  X  X
1  X  X  X  2
X  X  X  X  X
X  X  X  X  X
X  2  X  X  X
The value function is -20
```

Example 2_1: (n = 7, solvable)

```
Enter 5, 7, 9, or 11
7
Task2: The original matrix is:
2  1  6  3  4  4  4
2  2  2  2  4  4  1
3  3  2  2  3  5  4
4  2  4  3  1  4  6
5  2  3  1  3  1  2
1  3  2  5  2  4  6
4  2  4  1  4  5  G


Task 3: Function evaluation
0  X  1  3  X  X  4
X  X  X  X  X  X  X
1  3  3  2  4  3  X
X  X  X  4  X  X  X
3  X  4  3  4  4  X
2  3  X  4  4  X  X
3  X  2  X  4  X  3
The value function is 3
```

Example 2_2: (n = 7, not solvable)

```
Enter 5, 7, 9, or 11
7
Task2: The original matrix is:
1  6  1  3  6  5  2
2  5  2  2  1  1  1
4  1  4  4  4  1  2
4  4  3  2  2  3  4
3  1  1  3  4  3  3
5  4  5  3  4  3  4
5  2  2  4  4  5  G


Task 3: Function evaluation
0  1  4  5  4  5  6
1  5  2  4  3  4  5
5  X  X  4  4  5  6
2  4  3  5  3  4  4
4  3  4  5  X  X  6
5  4  5  6  4  5  7
5  2  4  3  5  5  X
The value function is -5
```

Example 3_1: (n = 9, solvable)

```
Enter 5, 7, 9, or 11
9
Task2: The original matrix is:
6  1  7  6  1  7  5  7  2
6  4  5  3  2  3  2  4  4
5  6  2  3  4  3  4  2  5
5  3  1  4  3  2  1  1  4
4  7  6  2  1  4  4  1  7
6  5  3  4  3  2  5  6  5
4  6  5  1  5  4  3  1  6
2  3  3  2  3  5  6  6  6
6  2  4  4  6  7  3  1  G


Task 3: Function evaluation
0  2  3  6  5  6  1  7  7
7  3  4  6  3  4  4  5  5
2  X  4  5  5  3  6  6  4
X  5  X  6  4  5  5  5  6
X  6  5  7  10  5  6  6  6
4  3  X  5  11  4  2  5  6
1  6  5  7  2  X  7  6  7
3  8  4  7  9  5  X  7  5
X  X  X  8  X  6  7  9  10
The value function is 10
```

Example 3_2: (n = 9, not solvable)

```
Enter 5, 7, 9, or 11
9
Task2: The original matrix is:
3  1  8  1  1  7  1  6  1
1  1  3  1  5  6  6  5  3
3  4  3  2  6  1  1  3  5
8  5  1  1  5  3  4  2  8
6  6  1  5  3  3  4  5  5
3  6  4  4  4  1  4  2  7
6  7  5  6  6  2  4  5  1
7  4  5  6  7  7  3  5  8
5  3  8  7  8  7  7  4  G


Task 3: Function evaluation
0  9  2  1  2  3  9  10  X
7  8  3  2  3  4  6  8  X
8  4  6  3  5  4  5  6  X
1  6  5  6  7  5  6  8  2
X  5  4  4  X  7  8  6  5
9  8  5  10  9  8  6  7  7
X  5  8  7  4  6  X  7  6
8  X  9  8  X  4  7  8  7
10  7  3  X  6  7  9  X  X
The value function is -12
```

Example 4_1: (n = 11, solvable)

```
Enter 5, 7, 9, or 11
11
Task2: The original matrix is:
2  10  7  6  6  9  9  1  5  8  7
10  5  4  5  4  3  3  9  2  4  5
9  7  4  2  6  4  4  7  1  2  9
9  7  5  5  5  3  4  4  6  2  2
4  5  1  1  1  1  4  6  5  7  1
2  2  8  6  6  4  6  2  1  8  6
2  4  3  4  2  3  4  4  8  8  6
8  3  4  5  7  2  3  1  6  4  8
7  6  5  5  8  3  1  3  6  9  6
8  9  6  5  8  2  5  6  3  2  4
5  1  10  3  6  5  5  9  6  1  G


Task 3: Function evaluation
0  3  1  6  7  X  5  X  X  2  8
5  X  8  8  X  7  7  8  6  6  6
1  6  X  5  X  6  X  3  7  2  7
7  8  3  5  7  8  X  4  6  5  6
X  5  4  5  6  7  4  X  X  3  5
6  7  5  6  7  6  5  6  X  6  6
X  X  8  7  8  7  X  6  8  X  7
7  8  2  4  X  5  3  5  5  4  X
4  X  4  6  6  6  5  5  7  3  6
X  5  X  7  X  6  6  4  7  X  6
5  4  5  8  X  6  4  7  X  X  7
The value function is 7
```

Example 4_2: (n = 11, not solvable)

```
Enter 5, 7, 9, or 11
11
Task2: The original matrix is:
1  4  5  2  1  8  5  7  2  2  4
5  9  1  9  1  7  4  7  7  2  6
1  9  8  4  6  2  3  3  3  2  4
3  5  4  3  5  7  1  2  5  6  5
5  9  4  6  3  5  5  6  4  8  5
10  7  5  3  5  3  2  2  8  4  7
3  4  3  4  6  2  5  4  1  5  7
6  3  4  7  6  6  5  5  7  8  2
6  5  1  8  4  7  3  4  4  7  4
6  8  2  2  3  1  5  5  3  3  8
6  4  2  8  9  7  9  10  3  9  G


Task 3: Function evaluation
0  1  9  5  7  2  9  6  8  8  9
1  6  13  7  6  2  12  7  X  8  5
9  9  14  4  6  8  6  5  9  7  6
3  12  7  4  X  6  5  6  X  7  X
5  2  X  X  8  4  5  8  9  8  3
9  X  8  8  8  7  7  6  8  7  7
2  8  X  3  7  9  X  4  X  7  7
X  8  6  X  9  X  7  7  7  X  6
8  11  10  9  7  3  9  8  8  10  X
3  5  5  X  6  5  4  7  X  6  4
10  7  9  4  9  6  11  5  X  X  X
The value function is -20
```

## Task 4. Hill Climbing

In this task, we use hill climbing algorithm to compute the hardest n*n puzzle(n=5,7,9,11). In this report, we will report the statistics of how evaluation function changes as the number of iterations increases (ps: based on the same original generated random matrix from task2, and just change the number of iterations)
Example: n = 5 (5*5 matrix)

```
Enter 5, 7, 9, or 11
5
Task2: The original matrix is:
2  1  2  2  1
4  1  1  2  1
3  1  2  2  2
3  1  2  2  3
4  4  2  3  G


Task 3: Function evaluation
0  4  1  3  2
5  4  5  4  3
1  3  2  2  3
5  4  5  5  X
4  5  3  3  4
The value function is 4
```

```
Number of iterations: 50
4  4  2  4  1
4  3  3  2  1
3  2  2  2  1
1  3  3  3  2
1  4  2  3  G


0  3  6  2  1
X  4  X  3  2
3  5  7  4  3
2  3  5  4  4
1  2  8  3  9
The value function is 9
```

```
Number of iterations: 100
2  2  4  2  3
1  2  1  1  1
3  2  1  2  2
1  1  1  1  2
1  1  1  4  G


0  4  1  3  7
X  6  5  6  6
1  3  4  2  7
5  4  3  4  5
4  3  2  3  8
The value function is 8
```

```
Number of iterations: 200
4  4  1  1  3
4  2  3  3  3
4  1  2  3  3
1  3  2  2  3
1  1  3  3  G


0  2  7  8  1
8  6  4  7  9
3  5  6  X  4
2  3  X  X  2
1  2  3  8  10
The value function is 10
```

```
Number of iterations: 400
3  1  3  4  4
2  1  3  3  1
1  2  2  1  1
3  2  3  3  4
3  1  1  3  G


0  10  8  1  14
4  9  5  3  13
13  10  12  11  12
1  8  7  2  13
3  7  6  2  15
The value function is 15
```

```
Number of iterations: 800
4  1  2  4  2
2  3  3  1  4
3  3  2  2  3
3  3  1  1  2
2  3  3  3  G


0  12  2  4  1
7  13  3  5  6
2  3  3  3  2
8  11  10  9  10
1  14  2  4  15
The value function is 15
```

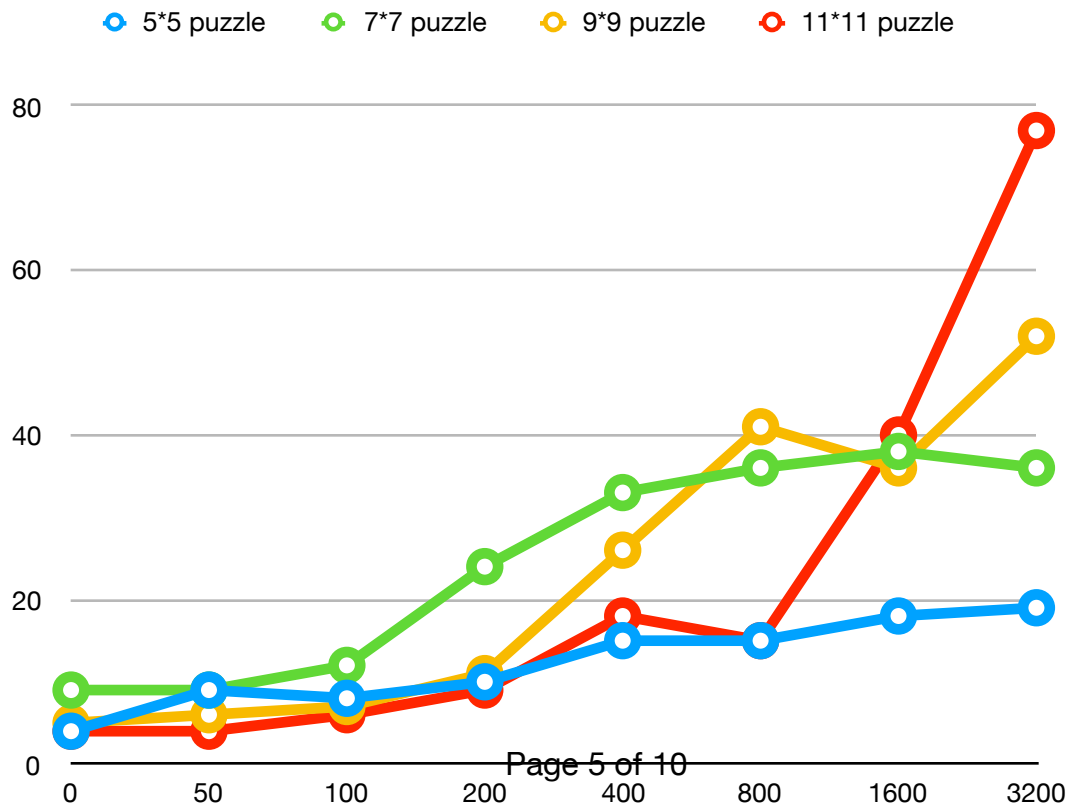
```
Number of iterations: 1600
4  1  2  4  1
4  3  2  1  4
3  3  1  3  1
3  3  3  2  4
3  1  2  3  G


0  14  6  2  1
2  15  4  3  2
5  8  7  4  9
11  13  5  12  10
1  16  17  2  18
The value function is 18
```

```
Number of iterations: 3200
3  3  1  4  2
4  3  2  3  2
2  1  1  3  1
3  3  3  3  4
3  1  2  3  G


0  9  8  1  10
4  16  6  3  5
13  15  14  12  11
1  10  7  2  6
3  17  18  2  19
The value function is 19
```

For a 5*5 puzzle example shows above, the number of iterations changes to 50, 100, 200, 400, 800, 1600, 3200 numbers of iterations. The function value increase from 4 to 19.
The statistic graph and chart below shows 4 lines indicates that 5*5, 7*7, 9*9 and 11*11 puzzles. The x axis is the number of iterations and y axis is the value of difficulties (evaluation function value)



| | 0 | 50 | 100 | 200 | 400 | 800 | 1600 | 3200 |
|---|---|---|---|---|---|---|---|---|
| 5*5 puzzle | 4 | 9 | 8 | 10 | 15 | 15 | 18 | 19 |
| 7*7 puzzle | 9 | 9 | 12 | 24 | 33 | 36 | 38 | 36 |
| 9*9 puzzle | 5 | 6 | 7 | 11 | 26 | 41 | 36 | 52 |
| 11*11 puzzle | 4 | 4 | 6 | 9 | 18 | 15 | 40 | 77 |

Basically, as the number of iteration increase, the puzzle will become more difficult to solve since the function value is also increasing respectively. The interesting thing is as the dimensions of the puzzle change, the original function value will not become harder since the increase of the dimensions. But, since the number of iterations increase in task 4 : hill climbing, more dimensions the puzzle is, more difficult to solve. For example, when the number of iterations is 0, the function values of 5*5, 7*7, 9*9, 11*11 are 4, 9, 5, 4 and when the number of iterations is 3200, the function values become 19, 36, 52, and 77 respectively.

### Task 5. Shortest path first algorithm

As the size of the matrix grows bigger, the run time gets bigger.

### Task 6. A*

In this case, we set the heuristic as “adding the current number into its row or column can reach the target’s row or column”. From the runtime we can tell that A* algorithm works better than Dijkstra in this case.

### Task 7. Propose and implement a population-based approach

In this task, we set the population set to be 10, so there will be 10 matrix to mutate together, in order to faster create a harder maze. We set 5 ways of how they combine with each other. We first create a dice, and when the dice is on

0 or 1, [A,B],[C,D],[E,F],[G,H],[I,J] will combine
2 or 3, [A,C],[D,F],[G,I],[H,J],[B,E] will combine
4 or 5, [A,D],[E,H],[B,I],[C,F],[G,J] will combine
6 or 7, [A,E],[F,J],[B,G],[C,H],[D,I] will combine
8 or 9, [A,J],[B,I],[C,H],[D,G],[E,F] will combine

the way of how they going to mutate is completely random, two dices are going to determine the number of row and column to switch each time.
As the population grows, the time to mutate decreases. However, this can still be a really random process comparing to hill climbing. Since hill climbing is more like a scientific way, to specifically go through the particle one by one, genetic algorithm is more like a mimic of natural processing, which won’t make the difficulty of the maze change that drastically. However, as the size of the matrix increases, if we are running same iteration on hill climbing and genetic algorithm, genetic algorithm actually gives a harder maze. We believe that’s because hill climbing go throw particles one by one, and it definitely will take a longer time to give a harder maze. However, since genetic algorithm change more than one particle at a time, a harder maze can be generated as the size of the matrix increases.

```
Task2: The original matrix is:
4  4  4  1  2
4  2  2  1  1
3  1  2  1  4
3  3  2  3  4
1  4  1  1  G


Task 3: Function evaluation
0  3  2  4  1
6  X  5  5  6
3  X  5  4  2
2  X  4  3  5
1  2  3  4  5
The value function is 5
```

```
Task 7: Matrix After genetic 50
3  2  2  2  4
3  1  3  2  2
2  2  2  2  3
4  2  3  2  4
1  1  1  4  G


Solve Genetic
0  2  7  1  8
7  6  7  7  X
9  3  8  2  9
1  5  6  6  2
5  4  5  3  9
```

```
Task 7: Matrix After genetic 100
2  3  4  3  1
2  2  2  3  2
3  1  2  3  4
3  2  1  3  1
2  2  1  4  G


Solve Genetic
0  X  1  4  X
7  5  8  6  9
1  4  4  2  5
5  4  3  4  10
X  3  2  3  11
```

```
Task 7: Matrix After genetic 200
3  3  3  3  2
2  2  3  1  3
3  2  1  3  2
2  1  3  3  4
3  2  1  2  G


Solve Genetic
0  7  3  1  8
2  X  3  X  X
7  6  10  6  9
1  8  2  2  X
X  5  4  5  10
```

```
Task 7: Matrix After genetic 400
2  1  1  3  3
4  3  4  3  4
1  1  1  1  1
2  1  3  3  2
2  4  2  3  G


Solve Genetic
0  2  1  2  X
2  3  2  5  3
1  2  3  4  5
2  3  3  3  6
7  4  8  6  9
```

```
Task 7: Matrix After genetic 800
3  4  3  3  1
2  2  1  1  2
1  3  4  3  2
2  3  2  3  3
1  4  4  1  G


Solve Genetic
0  5  4  1  4
2  4  3  4  5
6  7  4  5  8
1  4  2  2  3
X  6  X  X  9
```

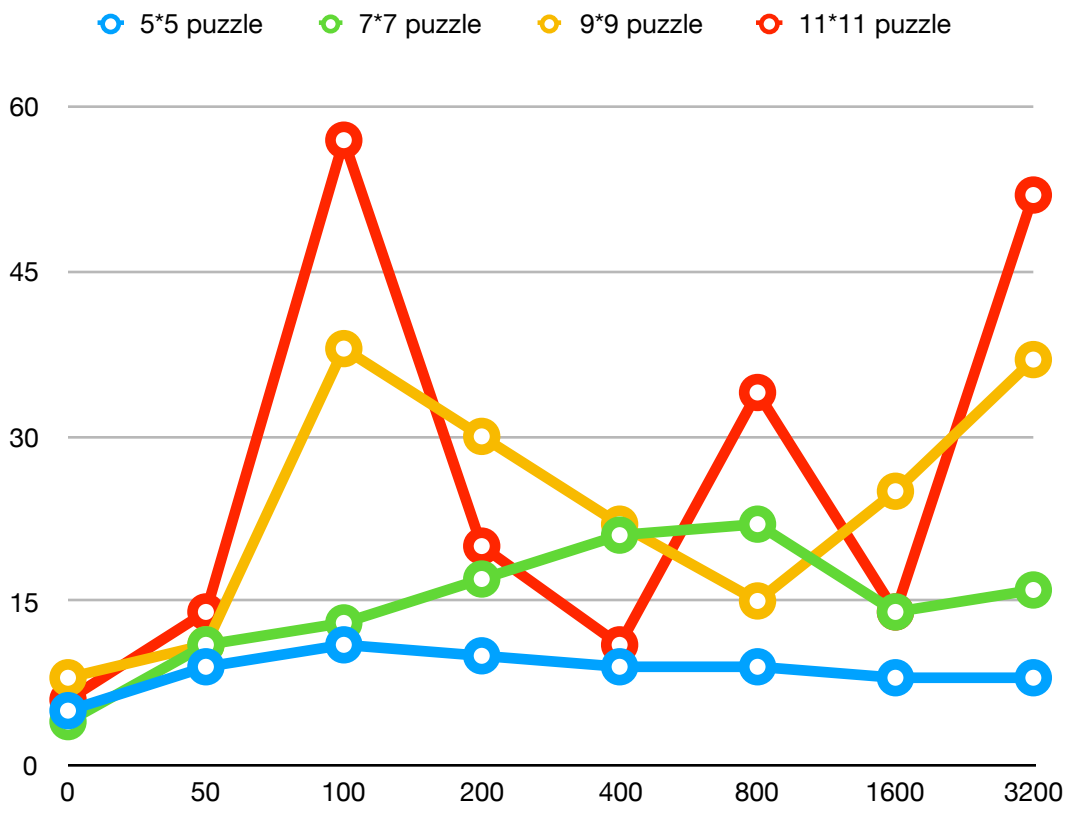
```
Task 7: Matrix After genetic 1600
3  4  2  3  4
4  1  2  1  4
4  3  1  3  1
3  2  3  2  1
4  1  4  3  G


Solve Genetic
0  5  6  1  7
5  4  4  3  4
5  5  7  4  6
1  3  5  2  7
7  6  7  X  8
```

```
Task 7: Matrix After genetic 3200
3  3  3  2  2
2  3  3  3  2
3  1  1  3  3
4  2  2  1  3
3  4  2  4  G


Solve Genetic
0  2  4  1  3
X  4  6  X  5
3  5  6  2  4
1  3  5  4  2
8  5  7  5  8
```

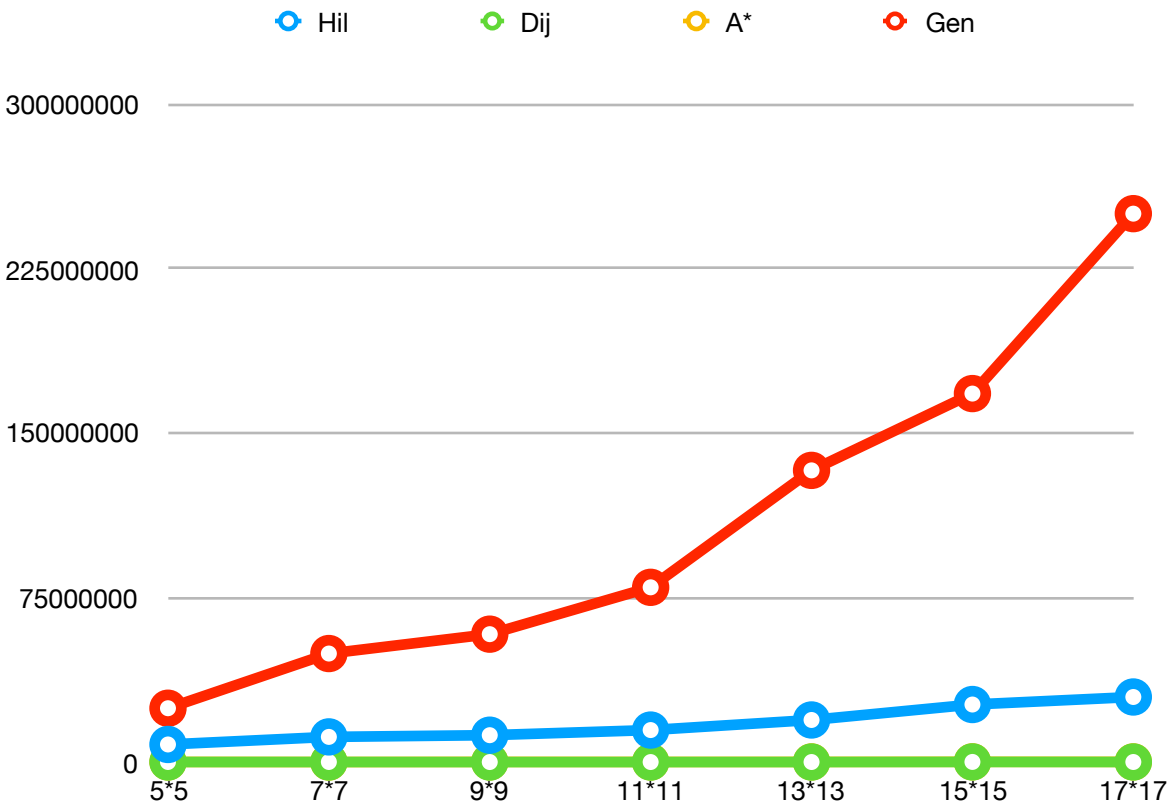
| | | 0 | 50 | 100 | 200 | 400 | 800 | 1600 | 3200 |
|---|---|---|---|---|---|---|---|---|---|
| | 5*5 puzzle | 5 | 9 | 11 | 10 | 9 | 9 | 8 | 8 |
| | 7*7 puzzle | 4 | 11 | 13 | 17 | 21 | 22 | 14 | 16 |
| | 9*9 puzzle | 8 | 11 | 38 | 30 | 22 | 15 | 25 | 37 |
| | 11*11 puzzle | 6 | 14 | 57 | 20 | 11 | 34 | 14 | 52 |




## Task 8. Evaluation

For genetic, it works pretty good when the matrix size is big; however, its run-time can be much larger than hill climbing, since more than one item is changed every time. Hill climbing works fine when the size of matrix is relatively small comparing to the iteration time; however, since it has to run through every single particle to out put the hardest maze, it won't work that well when iteration is small corresponding to the size of the matrix. For A*, if the heuristic works for the case, it well preform better than Dijkstra. For Dijkstra, it increases run-time as the puzzle becomes larger.
The chart and graph below compare the run-time between hill-climbing algorithm(task 4), dijkstra algorithm (task 5), A* algorithm (task 6), and genetic algorithm (task 7). The x axis means different size of puzzles, and the y axis means the run-time in nanosecond.

| | 5*5 | 7*7 | 9*9 | 11*11 | 13*13 | 15*15 | 17*17 |
|---|---|---|---|---|---|---|---|
| Hil | 8087169 | 11538278 | 12315871 | 14649495 | 19248341 | 26290716 | 29722700 |
| Dij | 96846 | 100682 | 74521 | 77694 | 86452 | 130125 | 100007 |
| A* | 131229 | 84113 | 144601 | 122876 | 134456 | 172956 | 152178 |
| Gen | 24602711 | 49524341 | 58355050 | 79703038 | 133000000 | 168000000 | 250000000 |




## Task 9: The most difficult and largest puzzle you can define and slove

The hardest puzzle (25*25) is:

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 18 | 18 | 19 | 7 | 18 | 22 | 20 | 11 | 19 | 9 | 21 | 17 | 16 | 21 | 17 | 16 | 1 | 17 | 15 | 24 | 19 | 3 | 23 | 24 | 4 |
| 13 | 14 | 10 | 7 | 22 | 13 | 20 | 17 | 9 | 14 | 23 | 9 | 22 | 14 | 14 | 23 | 16 | 3 | 17 | 9 | 19 | 13 | 7 | 12 | 15 |
| 15 | 9 | 20 | 2 | 9 | 17 | 18 | 1 | 22 | 13 | 11 | 9 | 19 | 14 | 3 | 11 | 13 | 13 | 14 | 1 | 16 | 21 | 18 | 17 | 6 |
| 21 | 13 | 5 | 15 | 14 | 20 | 5 | 19 | 18 | 12 | 14 | 21 | 11 | 13 | 17 | 16 | 17 | 4 | 8 | 14 | 16 | 13 | 2 | 18 | 23 |
| 8 | 20 | 3 | 17 | 15 | 10 | 19 | 11 | 14 | 4 | 13 | 12 | 5 | 4 | 8 | 19 | 20 | 20 | 17 | 5 | 10 | 20 | 6 | 11 | 11 |
| 17 | 16 | 10 | 19 | 10 | 4 | 3 | 19 | 1 | 11 | 5 | 15 | 3 | 11 | 6 | 16 | 9 | 15 | 1 | 2 | 16 | 15 | 1 | 10 | 14 |
| 2 | 20 | 22 | 17 | 12 | 18 | 12 | 3 | 1 | 14 | 9 | 2 | 18 | 18 | 10 | 6 | 11 | 12 | 18 | 17 | 14 | 18 | 8 | 23 | 19 |
| 11 | 19 | 3 | 7 | 9 | 10 | 12 | 15 | 15 | 14 | 17 | 10 | 12 | 1 | 15 | 17 | 1 | 14 | 17 | 12 | 8 | 17 | 16 | 20 | 15 |
| 7 | 7 | 12 | 8 | 13 | 15 | 5 | 14 | 10 | 15 | 4 | 13 | 2 | 2 | 16 | 16 | 7 | 8 | 8 | 5 | 13 | 16 | 1 | 6 | 13 |
| 18 | 5 | 12 | 14 | 6 | 14 | 7 | 10 | 15 | 11 | 9 | 6 | 7 | 8 | 9 | 1 | 6 | 6 | 3 | 14 | 13 | 21 | 2 | 20 | 20 |
| 24 | 12 | 4 | 13 | 16 | 13 | 2 | 4 | 5 | 4 | 13 | 11 | 2 | 13 | 11 | 7 | 15 | 4 | 11 | 14 | 11 | 15 | 11 | 3 | 24 |
| 21 | 14 | 12 | 17 | 20 | 14 | 7 | 16 | 2 | 15 | 12 | 1 | 1 | 3 | 5 | 12 | 11 | 7 | 1 | 19 | 8 | 17 | 14 | 18 | 1 |
| 18 | 13 | 6 | 5 | 8 | 18 | 12 | 14 | 11 | 4 | 12 | 2 | 9 | 4 | 14 | 11 | 4 | 10 | 13 | 10 | 18 | 20 | 15 | 9 | 14 |
| 10 | 19 | 4 | 7 | 7 | 1 | 13 | 1 | 14 | 15 | 12 | 5 | 2 | 8 | 13 | 2 | 13 | 11 | 1 | 13 | 11 | 9 | 22 | 13 | 7 |
| 16 | 3 | 11 | 14 | 20 | 11 | 15 | 1 | 15 | 6 | 13 | 8 | 14 | 14 | 12 | 6 | 10 | 17 | 3 | 12 | 12 | 17 | 11 | 18 | 16 |
| 19 | 17 | 21 | 9 | 17 | 8 | 1 | 15 | 4 | 8 | 3 | 11 | 12 | 12 | 3 | 9 | 3 | 15 | 17 | 9 | 20 | 19 | 18 | 3 | 18 |
| 24 | 1 | 21 | 18 | 16 | 13 | 13 | 14 | 8 | 14 | 13 | 10 | 14 | 10 | 4 | 11 | 11 | 6 | 18 | 16 | 12 | 19 | 19 | 9 | 2 |
| 9 | 14 | 20 | 18 | 13 | 17 | 17 | 1 | 2 | 1 | 16 | 6 | 8 | 6 | 6 | 12 | 17 | 14 | 3 | 3 | 20 | 19 | 7 | 10 | 12 |
| 24 | 12 | 6 | 14 | 19 | 16 | 17 | 18 | 18 | 13 | 18 | 14 | 17 | 6 | 8 | 8 | 7 | 17 | 14 | 11 | 14 | 14 | 18 | 23 | 20 |
| 6 | 20 | 15 | 16 | 8 | 6 | 7 | 11 | 17 | 11 | 12 | 13 | 5 | 4 | 13 | 4 | 17 | 13 | 7 | 5 | 16 | 11 | 22 | 3 | 20 |
| 15 | 3 | 20 | 17 | 6 | 8 | 8 | 12 | 13 | 17 | 3 | 10 | 5 | 20 | 17 | 8 | 1 | 9 | 15 | 12 | 6 | 21 | 18 | 11 | 4 |
| 11 | 7 | 14 | 6 | 20 | 5 | 19 | 18 | 17 | 14 | 21 | 18 | 3 | 6 | 17 | 20 | 14 | 4 | 20 | 8 | 20 | 7 | 17 | 23 | 14 |
| 19 | 21 | 9 | 6 | 9 | 4 | 12 | 18 | 13 | 6 | 10 | 22 | 16 | 7 | 13 | 5 | 4 | 14 | 9 | 16 | 21 | 21 | 7 | 1 | 12 |
| 10 | 20 | 4 | 22 | 1 | 22 | 7 | 1 | 14 | 23 | 10 | 18 | 14 | 22 | 9 | 5 | 1 | 7 | 11 | 7 | 2 | 5 | 19 | 18 | 6 |
| 6 | 20 | 18 | 5 | 16 | 4 | 6 | 10 | 3 | 17 | 17 | 14 | 21 | 16 | 13 | 19 | 20 | 11 | 21 | 3 | 16 | 6 | 6 | 16 | 0 |

And the solution of this puzzle is:

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 479 | 475 | 2 | 212 | 330 | 109 | 26 | 136 | 243 | 3 | 436 | 93 | 143 | 499 | 209 | 208 | 184 | 1 | 104 | 478 | 476 | 78 | 412 | 477 |
| 165 | 19 | 464 | 81 | 523 | 17 | 114 | 459 | 321 | 461 | 68 | 463 | 98 | 127 | 257 | 20 | 209 | 256 | 18 | 285 | 170 | 320 | 542 | 462 | 460 |
| 564 | 561 | 505 | 560 | 119 | 333 | 116 | 458 | 138 | 273 | 562 | 504 | 95 | 120 | 503 | 565 | 559 | 182 | 118 | 284 | 173 | 563 | 274 | 415 | 117 |
| 130 | 68 | 472 | 341 | 360 | 15 | 195 | 146 | 132 | 240 | 66 | 49 | 91 | 129 | 66 | 372 | 556 | 186 | 299 | 281 | 359 | 131 | 541 | 46 | 67 |
| 164 | 21 | 468 | 343 | 362 | 367 | 192 | 148 | 134 | 366 | 176 | 235 | 178 | 365 | 191 | 368 | 8 | 179 | 20 | 363 | 175 | 22 | 54 | 177 | 364 |
| 403 | 424 | 405 | 537 | 313 | 423 | 536 | 216 | 315 | 311 | 422 | 439 | 406 | 420 | 314 | 22 | 215 | 404 | 533 | 534 | 312 | 535 | 538 | 419 | 421 |
| 163 | 292 | 159 | 294 | 205 | 161 | 296 | 290 | 316 | 238 | 291 | 237 | 89 | 122 | 204 | 375 | 206 | 254 | 297 | 158 | 295 | 293 | 539 | 162 | 160 |
| 491 | 488 | 469 | 3 | 494 | 328 | 486 | 484 | 317 | 87 | 4 | 492 | 490 | 495 | 496 | 329 | 511 | 187 | 487 | 483 | 489 | 493 | 485 | 64 | 86 |
| 164 | 197 | 473 | 82 | 521 | 167 | 196 | 165 | 198 | 358 | 200 | 83 | 407 | 124 | 201 | 125 | 512 | 386 | 199 | 280 | 168 | 166 | 543 | 414 | 84 |
| 552 | 548 | 508 | 231 | 546 | 101 | 549 | 246 | 229 | 244 | 547 | 233 | 100 | 550 | 498 | 554 | 555 | 232 | 553 | 101 | 245 | 551 | 544 | 230 | 545 |
| 36 | 263 | 470 | 261 | 391 | 287 | 390 | 289 | 322 | 393 | 409 | 51 | 408 | 125 | 260 | 371 | 262 | 388 | 288 | 286 | 392 | 389 | 52 | 410 | 35 |
| 31 | 427 | 465 | 429 | 33 | 29 | 305 | 27 | 304 | 204 | 302 | 432 | 431 | 306 | 203 | 428 | 307 | 301 | 300 | 30 | 430 | 32 | 303 | 28 | 34 |
| 42 | 40 | 516 | 379 | 377 | 44 | 111 | 38 | 380 | 382 | 36 | 433 | 378 | 383 | 41 | 376 | 514 | 384 | 43 | 381 | 515 | 39 | 37 | 45 | 35 |
| 222 | 571 | 507 | 339 | 336 | 335 | 108 | 219 | 220 | 573 | 223 | 337 | 569 | 567 | 570 | 566 | 338 | 181 | 529 | 107 | 572 | 568 | 221 | 411 | 574 |
| 6 | 426 | 471 | 4 | 10 | 14 | 8 | 218 | 12 | 394 | X | 434 | 92 | 307 | 502 | 395 | 7 | 5 | 530 | 283 | 172 | 9 | 540 | 13 | 11 |
| 156 | 3 | 153 | 345 | 151 | 327 | 326 | 149 | 323 | 241 | 158 | 234 | 324 | 128 | 258 | 374 | 513 | 183 | 2 | 157 | 155 | 152 | 150 | 154 | 325 |
| 57 | 69 | 62 | 60 | 211 | 55 | 194 | 457 | 213 | 272 | 65 | 236 | 94 | 121 | 64 | 210 | 214 | 387 | 56 | 103 | 212 | 61 | 59 | 63 | 58 |
| 450 | 70 | 74 | 72 | 361 | 329 | 453 | 456 | 452 | 451 | 67 | 437 | 448 | 455 | 500 | 71 | 207 | 185 | 531 | 282 | 449 | 73 | 75 | 454 | 447 |
| 1 | 480 | 517 | 342 | 3 | 332 | 113 | 25 | 135 | 310 | 159 | 338 | 97 | 481 | 259 | 370 | 309 | 255 | 19 | 482 | 174 | 24 | 77 | 4 | 2 |
| 354 | 350 | 467 | 347 | 251 | 334 | 355 | 247 | 137 | 357 | 352 | 249 | 252 | 356 | 349 | 373 | 558 | 253 | 248 | 348 | 358 | 351 | 353 | 416 | 250 |
| 402 | 441 | 474 | 340 | 276 | 141 | 110 | 278 | 400 | 239 | 277 | 440 | 398 | 142 | 65 | 396 | 557 | 399 | 532 | 279 | 171 | 401 | 275 | 397 | 575 |
| 47 | 425 | 509 | 344 | 522 | 54 | 115 | 145 | 133 | 88 | 4 | 48 | 96 | 144 | 190 | 23 | 510 | 188 | 21 | 106 | 169 | 189 | 53 | 46 | 85 |
| 268 | 264 | 506 | 270 | 525 | 331 | 527 | 147 | 318 | 271 | 267 | 435 | 88 | 526 | 497 | 266 | 308 | 385 | 528 | 269 | 246 | 319 | 265 | 417 | 87 |
| 223 | 442 | 466 | 80 | 524 | 16 | 193 | 227 | 228 | 242 | 224 | 438 | 99 | 126 | 501 | 369 | 444 | 445 | 226 | 102 | 225 | 443 | 79 | 418 | 446 |
| 113 | 22 | 518 | 346 | 520 | 140 | 112 | 217 | 139 | 141 | 5 | 50 | 90 | 123 | 202 | 21 | 8 | 180 | 298 | 105 | 519 | 23 | 76 | 413 | 576 |

which need at least “576” steps from start to goal.