Ching-Wei Lin

Project2

1. Instruction

Use python2 environment to run the program.

Run the program without argument.

Example:

python project2.py Number\_Of\_Population Number\_Of\_Max\_Iteration

1. GA parameters

In my fitness function, I evaluate each chess table by calculating how many queens are in the same row plus how many queens in the same diagonal, not column because there’s always only one queen in each column. The one who is closer to solution has a higher fitness value and the fitness value for solution is 9. In my mutation function, I generate a random number between 1 and 100 for each chess table. If the number is greater 90, randomly change a number in the table to a random number. That is to say, my mutation possibility is 10%.

1. Technique

My GA algorithm is that keep producing new generations from the current chess table until reaching max iteration or solution founded. The way I produce new generation is picking a pair of current chess tables based on their picking possibility which the one with bigger fitness value has higher possibility to be picked. Then, randomly generate a number between 1 and 7 which indicate how many number will be swapped to generate children. After, go through the mutation function and keep looping till finding the solution or reaching the maximum iteration number.

1. Findings & Conclusion

Average fitness value from 100 times running

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Max\_iteration\Population | 10 | 100 | 500 | 1000 |
| 100 | 4.587 | 4.8026 | 4.5683 | 4.47 |
| 300 | 4.844 | 4.7554 | 4.6474 | 4.468 |

Average iteration from 100 times running

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Max\_iteration\Population | 10 | 100 | 500 | 1000 |
| 100 | 93 | 39 | 11 | 7 |
| 300 | 238 | 39 | 13 | 7 |

Number of finding solution successfully from 100 times running

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Max\_iteration\Population | 10 | 100 | 500 | 1000 |
| 100 | 15 | 92 | 100 | 100 |
| 300 | 37 | 100 | 100 | 100 |

As shown above, population of 10 has bad performance both in average iteration and number of finding solution successfully because the size of population is too small to keep the good gene in order to produce “better” children. On the other hand, population of 500 and 1000 produce too many children in order to find the solution, which is wasting resources. Population of 500 produces 500\*11=5500 children in total and population produces 1000\*7=7000 in total. However, Population of 100 produce only 100\*39 = 3900 children and also solution is guaranteed.

Average iteration from 100 times running in different mutation possibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mutation Possibility | 2% | 5% | 10% | 15% |
| 100Pop, 300 max\_iteration | 48 | 33 | 39 | 46 |

Number of finding solution successfully from 100 times running in different mutation possibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mutation Possibility | 2% | 5% | 10% | 15% |
| 100Pop, 300 max\_iteration | 100 | 100 | 100 | 100 |

As shown above, 2% mutation possibility is to small that it sticks on a local maximum and spends too much time on it. On the other hand, 10% mutation possibility is too big that it removes some “good” gene, which it takes more time to find the solution.

As a result, my optimal parameter for my GA algorithm are 100 populations, 300 max iteration and 5% mutation possibility.

1. Plots

10 population, 300 max iteration, 5% mutation

50 populations, 300 max iteration, 5% mutation

100 populations, 300 max iteration, 5% mutation

300 populations, 300 max iteration, 5% mutation

1. Examples
2. Initial population

[[0, 2, 2, 4, 0, 4, 0, 7], [6, 4, 0, 2, 1, 2, 5, 3], [7, 4, 7, 4, 3, 0, 0, 3], [2, 6, 0, 6, 4, 4, 4, 0], [5, 6, 7, 5, 5, 1, 1, 4], [5, 2, 4, 4, 4, 5, 1, 4], [3, 2, 7, 1, 0, 2, 6, 7], [0, 2, 1, 2, 4, 6, 1, 3], [3, 3, 3, 7, 6, 4, 3, 7], [0, 7, 1, 4, 0, 0, 0, 6], [5, 6, 7, 1, 1, 0, 4, 0], [6, 3, 1, 7, 5, 4, 6, 3], [5, 0, 5, 1, 5, 7, 6, 6], [6, 0, 4, 5, 1, 0, 1, 2], [4, 1, 5, 0, 3, 7, 1, 3], [5, 5, 1, 7, 4, 2, 4, 3], [1, 2, 1, 6, 4, 6, 1, 7], [2, 1, 2, 3, 5, 2, 3, 4], [0, 5, 5, 6, 7, 6, 6, 4], [6, 3, 0, 5, 5, 7, 3, 0], [4, 0, 1, 2, 7, 7, 7, 4], [4, 4, 7, 2, 4, 4, 1, 4], [0, 1, 3, 2, 1, 6, 2, 5], [4, 6, 5, 3, 1, 2, 4, 6], [7, 0, 0, 0, 5, 1, 2, 2], [4, 7, 0, 7, 2, 0, 3, 1], [1, 1, 7, 2, 7, 7, 5, 7], [2, 1, 4, 7, 6, 4, 7, 1], [1, 6, 2, 7, 3, 7, 3, 3], [1, 5, 2, 1, 5, 3, 2, 5], [0, 2, 0, 1, 2, 3, 2, 1], [5, 1, 6, 3, 7, 7, 4, 1], [7, 2, 7, 1, 5, 7, 0, 3], [1, 5, 2, 3, 7, 0, 4, 0], [5, 7, 6, 6, 0, 0, 4, 1], [0, 6, 3, 7, 0, 4, 4, 6], [6, 4, 3, 4, 2, 7, 2, 4], [6, 0, 1, 2, 7, 5, 6, 2], [5, 4, 2, 3, 2, 4, 2, 7], [7, 7, 4, 2, 3, 7, 7, 2], [1, 0, 5, 1, 3, 1, 1, 0], [1, 1, 2, 7, 4, 2, 7, 6], [3, 5, 2, 6, 7, 4, 0, 3], [7, 0, 5, 0, 7, 4, 3, 6], [4, 4, 1, 1, 4, 1, 3, 5], [1, 0, 0, 2, 2, 6, 7, 7], [0, 5, 1, 1, 6, 4, 5, 6], [2, 3, 0, 5, 1, 1, 6, 0], [6, 7, 5, 3, 3, 0, 2, 3], [5, 4, 5, 3, 7, 0, 2, 2], [0, 1, 5, 3, 4, 0, 6, 1], [6, 7, 0, 7, 0, 3, 6, 0], [6, 7, 1, 4, 6, 7, 2, 0], [5, 6, 0, 6, 7, 6, 5, 2], [3, 3, 1, 4, 3, 5, 4, 4], [7, 2, 2, 3, 3, 6, 0, 3], [1, 3, 6, 7, 6, 4, 2, 0], [5, 6, 3, 2, 3, 0, 5, 5], [6, 2, 2, 4, 0, 4, 2, 7], [3, 2, 5, 1, 4, 6, 2, 1], [5, 3, 1, 6, 4, 6, 4, 6], [4, 2, 1, 3, 5, 2, 3, 6], [4, 3, 2, 6, 0, 5, 5, 6], [7, 5, 5, 7, 1, 7, 1, 4], [1, 0, 3, 4, 1, 1, 3, 5], [7, 0, 5, 6, 4, 2, 1, 5], [3, 2, 3, 3, 2, 1, 3, 7], [7, 5, 6, 4, 2, 4, 5, 1], [5, 6, 2, 4, 1, 1, 0, 1], [0, 4, 3, 5, 6, 4, 2, 1], [6, 4, 2, 1, 7, 3, 5, 5], [2, 7, 5, 4, 2, 5, 6, 6], [0, 0, 2, 6, 2, 7, 1, 5], [5, 3, 6, 2, 6, 7, 0, 4], [0, 2, 2, 1, 1, 7, 6, 4], [3, 1, 3, 5, 5, 7, 7, 4], [3, 7, 6, 5, 1, 3, 1, 1], [0, 0, 0, 7, 6, 5, 2, 2], [3, 4, 7, 3, 2, 7, 5, 4], [7, 5, 6, 3, 6, 6, 0, 4], [5, 0, 7, 4, 1, 5, 6, 6], [1, 6, 3, 1, 5, 6, 1, 4], [2, 5, 2, 6, 7, 4, 1, 1], [4, 5, 5, 2, 4, 6, 6, 2], [5, 3, 0, 3, 6, 4, 0, 5], [2, 7, 2, 0, 0, 4, 1, 3], [3, 0, 7, 4, 3, 3, 2, 0], [1, 6, 3, 4, 4, 3, 7, 4], [2, 5, 4, 2, 6, 7, 6, 2], [3, 4, 2, 5, 7, 6, 6, 4], [4, 7, 5, 5, 1, 6, 5, 1], [0, 5, 3, 7, 1, 3, 3, 6], [2, 3, 2, 1, 6, 2, 5, 2], [0, 4, 6, 6, 6, 6, 3, 7], [2, 7, 3, 7, 0, 1, 6, 1], [0, 3, 0, 5, 4, 7, 2, 0], [4, 1, 6, 3, 6, 2, 7, 2], [4, 4, 1, 1, 0, 2, 6, 5], [4, 0, 5, 1, 1, 3, 6, 5], [7, 2, 2, 7, 2, 4, 3, 2]]

Total generations

13

Solution

[6, 4, 0, 2, 5, 7, 1, 3]

1. Initial population

[[3, 5, 1, 3, 2, 5, 6, 7], [3, 3, 4, 3, 1, 6, 4, 0], [3, 4, 2, 4, 7, 5, 7, 1], [4, 3, 4, 1, 1, 6, 5, 5], [5, 1, 3, 6, 0, 5, 2, 3], [2, 0, 6, 6, 2, 4, 7, 2], [0, 6, 3, 2, 7, 0, 3, 0], [4, 5, 3, 2, 6, 6, 5, 7], [6, 6, 5, 5, 1, 2, 6, 5], [0, 7, 7, 5, 0, 2, 7, 7], [7, 2, 3, 6, 3, 0, 0, 0], [4, 7, 4, 2, 4, 0, 7, 7], [2, 4, 2, 1, 1, 7, 6, 1], [1, 4, 2, 3, 1, 3, 5, 7], [2, 2, 2, 7, 5, 5, 0, 4], [5, 6, 5, 0, 1, 0, 4, 5], [2, 4, 1, 2, 0, 7, 3, 2], [3, 4, 2, 2, 4, 0, 5, 5], [4, 6, 5, 2, 0, 7, 2, 1], [0, 3, 1, 4, 5, 4, 4, 2], [1, 4, 4, 6, 2, 3, 2, 0], [2, 5, 3, 6, 6, 7, 6, 5], [5, 0, 0, 5, 2, 5, 7, 5], [6, 3, 3, 5, 1, 1, 7, 6], [3, 5, 4, 5, 2, 5, 5, 7], [5, 6, 3, 1, 1, 6, 0, 2], [0, 7, 6, 2, 0, 0, 0, 4], [6, 3, 4, 3, 0, 1, 0, 7], [3, 3, 6, 4, 3, 5, 7, 2], [0, 5, 2, 4, 6, 2, 3, 2], [5, 7, 0, 2, 3, 6, 6, 0], [0, 1, 1, 6, 0, 5, 3, 5], [5, 0, 5, 1, 7, 5, 3, 3], [7, 3, 4, 0, 1, 2, 5, 2], [1, 5, 5, 5, 7, 1, 4, 3], [7, 0, 5, 6, 5, 6, 6, 0], [1, 1, 6, 0, 1, 2, 2, 4], [0, 2, 3, 0, 5, 3, 2, 3], [0, 3, 5, 5, 7, 1, 3, 4], [0, 7, 3, 3, 5, 3, 3, 0], [6, 0, 1, 6, 5, 7, 7, 3], [0, 1, 6, 6, 2, 7, 5, 5], [4, 4, 7, 2, 3, 0, 2, 0], [1, 5, 6, 0, 4, 5, 3, 6], [7, 7, 0, 3, 4, 3, 1, 6], [6, 4, 5, 1, 4, 3, 3, 0], [5, 5, 2, 0, 0, 3, 6, 2], [4, 1, 0, 3, 7, 4, 4, 0], [0, 3, 4, 1, 5, 6, 1, 0], [1, 7, 7, 0, 4, 3, 3, 5], [4, 6, 3, 5, 7, 5, 7, 5], [3, 1, 2, 6, 0, 7, 4, 7], [1, 5, 2, 2, 4, 3, 1, 6], [2, 7, 5, 3, 1, 5, 7, 1], [2, 3, 0, 0, 2, 6, 2, 7], [5, 0, 0, 2, 1, 5, 3, 2], [0, 3, 1, 7, 3, 6, 0, 0], [1, 7, 6, 5, 4, 6, 4, 3], [6, 7, 4, 1, 3, 7, 3, 7], [6, 1, 3, 7, 5, 0, 4, 6], [0, 3, 3, 2, 6, 3, 6, 6], [1, 2, 0, 5, 3, 7, 7, 3], [6, 2, 0, 2, 3, 4, 3, 0], [1, 1, 2, 5, 5, 7, 7, 5], [6, 2, 1, 2, 2, 5, 3, 3], [5, 7, 7, 2, 3, 4, 2, 4], [2, 5, 4, 7, 6, 6, 5, 0], [5, 7, 5, 6, 3, 3, 2, 1], [0, 6, 2, 7, 2, 7, 3, 4], [5, 5, 7, 4, 2, 3, 1, 6], [6, 7, 6, 4, 3, 7, 5, 0], [0, 7, 2, 6, 6, 2, 3, 1], [0, 6, 4, 0, 3, 4, 4, 5], [3, 1, 5, 5, 5, 3, 3, 7], [6, 4, 1, 0, 4, 5, 7, 7], [2, 0, 0, 0, 1, 3, 0, 2], [2, 0, 3, 6, 5, 0, 7, 6], [3, 7, 0, 4, 7, 5, 3, 1], [2, 2, 1, 5, 4, 0, 3, 4], [5, 2, 1, 2, 0, 6, 5, 2], [6, 5, 5, 1, 3, 2, 4, 7], [3, 1, 4, 3, 3, 1, 7, 4], [5, 7, 1, 0, 5, 0, 2, 1], [0, 5, 0, 7, 3, 5, 4, 5], [2, 5, 7, 3, 0, 6, 3, 3], [1, 1, 5, 5, 2, 6, 5, 4], [5, 3, 1, 3, 1, 0, 7, 1], [4, 1, 6, 3, 3, 7, 6, 6], [5, 2, 2, 1, 3, 3, 7, 4], [3, 7, 7, 2, 7, 5, 6, 6], [2, 5, 1, 2, 2, 6, 7, 6], [5, 5, 5, 1, 1, 3, 2, 3], [2, 0, 3, 7, 2, 5, 6, 3], [7, 6, 4, 0, 0, 5, 4, 5], [5, 1, 4, 2, 1, 5, 1, 0], [0, 5, 6, 1, 2, 7, 1, 6], [6, 6, 1, 2, 7, 6, 1, 3], [3, 0, 4, 2, 7, 4, 3, 6], [0, 2, 5, 0, 4, 4, 6, 5], [3, 0, 1, 1, 5, 3, 1, 7]]

Total generations

29

Solution

[6, 1, 3, 5, 7, 0, 2, 4]

1. Initial population

[[0, 6, 1, 2, 3, 2, 0, 6], [0, 1, 1, 1, 0, 6, 1, 7], [3, 2, 7, 0, 0, 6, 7, 0], [3, 2, 6, 3, 1, 3, 0, 1], [4, 6, 5, 1, 3, 7, 4, 1], [0, 2, 7, 7, 4, 0, 2, 4], [7, 6, 4, 1, 2, 3, 2, 4], [2, 2, 3, 5, 4, 4, 5, 4], [1, 7, 1, 4, 0, 0, 0, 6], [7, 3, 2, 7, 1, 6, 3, 1], [5, 3, 6, 7, 5, 2, 7, 7], [4, 4, 7, 0, 3, 4, 7, 6], [2, 1, 1, 2, 0, 7, 4, 2], [0, 0, 6, 6, 6, 0, 6, 3], [0, 6, 1, 5, 0, 0, 5, 4], [5, 1, 3, 2, 3, 4, 4, 5], [4, 6, 3, 7, 4, 4, 0, 5], [2, 1, 1, 3, 2, 7, 2, 7], [6, 2, 3, 1, 7, 2, 0, 0], [7, 5, 0, 0, 4, 6, 5, 1], [4, 6, 3, 7, 7, 3, 5, 0], [5, 6, 5, 4, 3, 5, 0, 6], [6, 3, 0, 6, 3, 3, 1, 5], [3, 2, 7, 7, 2, 1, 6, 2], [0, 4, 2, 6, 0, 7, 0, 5], [4, 5, 0, 3, 6, 1, 4, 3], [0, 4, 6, 5, 4, 4, 3, 3], [5, 3, 5, 4, 0, 3, 6, 5], [1, 3, 2, 0, 2, 3, 1, 5], [4, 4, 3, 0, 5, 0, 0, 2], [7, 5, 2, 7, 0, 5, 4, 5], [1, 5, 4, 3, 6, 4, 6, 3], [5, 0, 5, 3, 2, 3, 7, 1], [1, 0, 7, 2, 7, 3, 1, 3], [7, 4, 0, 2, 2, 6, 0, 3], [6, 0, 3, 0, 6, 7, 4, 7], [0, 4, 6, 0, 4, 3, 3, 4], [1, 7, 5, 4, 7, 2, 6, 1], [1, 7, 6, 6, 1, 2, 1, 1], [7, 5, 3, 0, 7, 5, 0, 6], [5, 0, 5, 1, 1, 2, 5, 1], [4, 7, 0, 6, 3, 2, 6, 5], [3, 7, 6, 0, 6, 0, 5, 5], [6, 6, 7, 4, 3, 4, 1, 1], [7, 5, 2, 7, 4, 1, 5, 6], [0, 1, 7, 1, 0, 0, 6, 4], [3, 7, 2, 6, 5, 5, 6, 7], [6, 4, 0, 3, 5, 5, 1, 6], [4, 4, 6, 4, 2, 1, 0, 3], [2, 6, 5, 4, 0, 0, 6, 6], [4, 4, 1, 5, 1, 6, 3, 5], [5, 1, 3, 7, 3, 0, 1, 7], [3, 7, 7, 1, 7, 0, 2, 6], [0, 4, 4, 0, 0, 6, 4, 0], [7, 3, 3, 4, 3, 2, 0, 1], [2, 5, 3, 0, 0, 6, 4, 0], [0, 1, 7, 5, 7, 2, 5, 2], [6, 7, 7, 5, 3, 1, 2, 4], [7, 7, 3, 4, 5, 4, 2, 2], [5, 2, 7, 0, 2, 2, 0, 4], [6, 0, 4, 4, 6, 4, 1, 3], [4, 3, 7, 4, 5, 3, 0, 0], [6, 4, 5, 0, 1, 4, 7, 6], [3, 0, 6, 1, 2, 6, 2, 2], [0, 3, 1, 0, 1, 1, 6, 3], [0, 4, 1, 3, 2, 1, 7, 5], [2, 0, 7, 5, 3, 0, 0, 7], [2, 0, 3, 1, 7, 4, 3, 3], [7, 7, 1, 3, 4, 4, 7, 0], [5, 6, 7, 3, 1, 1, 1, 7], [5, 2, 0, 2, 1, 3, 4, 4], [0, 3, 5, 0, 6, 7, 0, 3], [4, 1, 5, 2, 0, 6, 2, 6], [1, 4, 1, 5, 5, 7, 1, 4], [5, 2, 0, 4, 7, 4, 4, 3], [7, 7, 2, 2, 7, 4, 4, 7], [7, 3, 7, 0, 5, 3, 6, 3], [0, 4, 4, 7, 0, 0, 5, 7], [5, 3, 0, 4, 2, 2, 0, 6], [3, 5, 4, 1, 7, 5, 4, 0], [7, 2, 6, 6, 6, 3, 5, 0], [2, 3, 5, 1, 5, 1, 7, 3], [4, 6, 1, 5, 0, 5, 7, 7], [0, 6, 4, 6, 4, 3, 0, 1], [4, 7, 0, 4, 7, 2, 4, 5], [0, 4, 6, 1, 4, 5, 7, 6], [2, 2, 7, 1, 3, 7, 3, 5], [5, 7, 1, 4, 5, 0, 0, 5], [3, 5, 7, 7, 7, 4, 0, 7], [7, 7, 5, 1, 2, 3, 7, 7], [2, 5, 5, 3, 3, 0, 6, 5], [0, 7, 0, 0, 4, 7, 6, 4], [5, 5, 4, 5, 6, 1, 1, 7], [3, 5, 1, 7, 6, 3, 0, 0], [2, 2, 5, 0, 7, 2, 0, 6], [7, 4, 7, 2, 3, 1, 2, 4], [1, 6, 3, 4, 3, 1, 2, 3], [4, 6, 6, 7, 1, 7, 1, 1], [1, 4, 5, 2, 6, 4, 0, 7], [1, 5, 2, 4, 7, 2, 1, 6]]

Total generations

62

Solution

[7, 3, 6, 2, 5, 1, 4, 0]