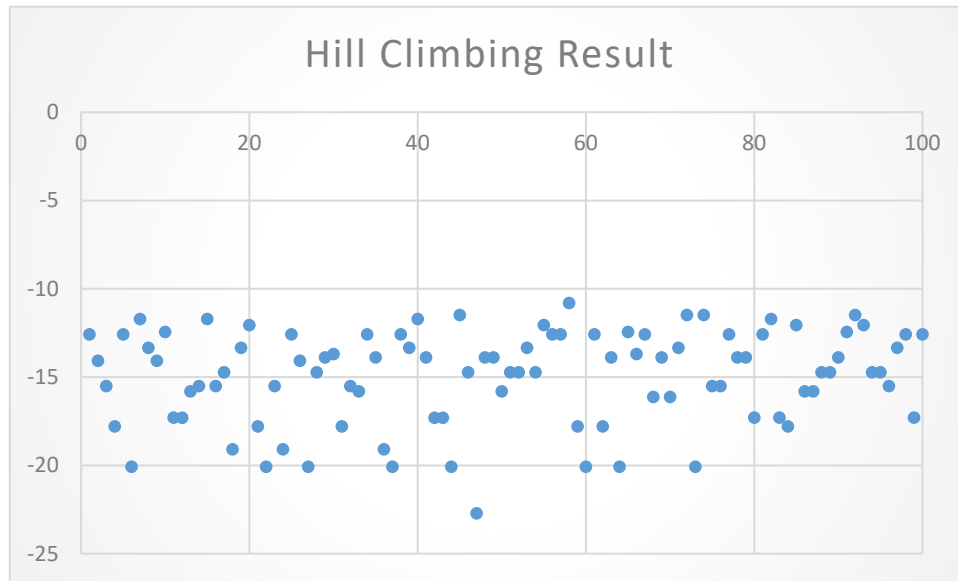


1. Optimization of a function with real parameters

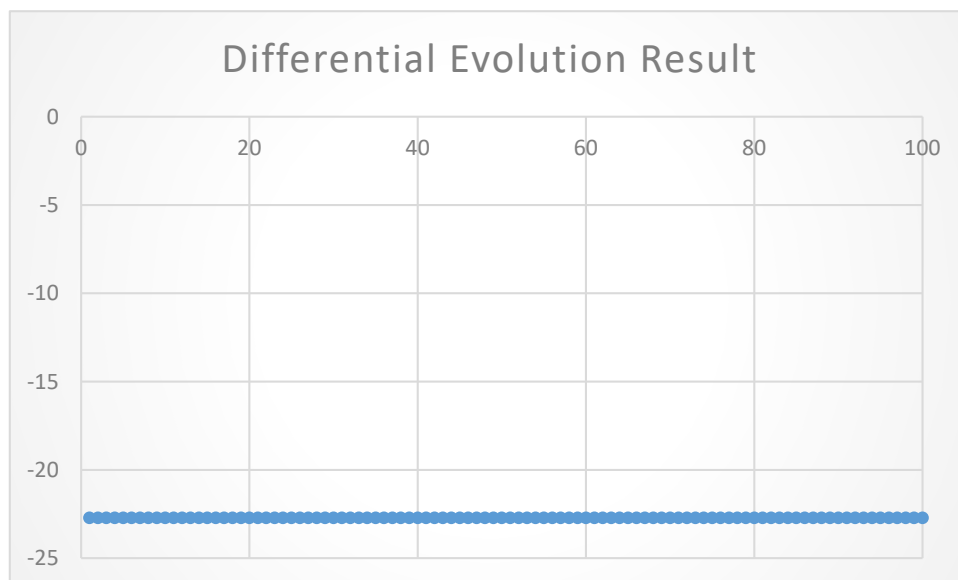
1.1 Execution

- The folder named as 'assg02-1' is for this question.
- When you run the program, the solution will be generated in the file 'output.txt'.

1.2 Analysis



As this figure shows, under the Hill Climbing algorithm, it only found the global minimum 1 time in total 100 runs.



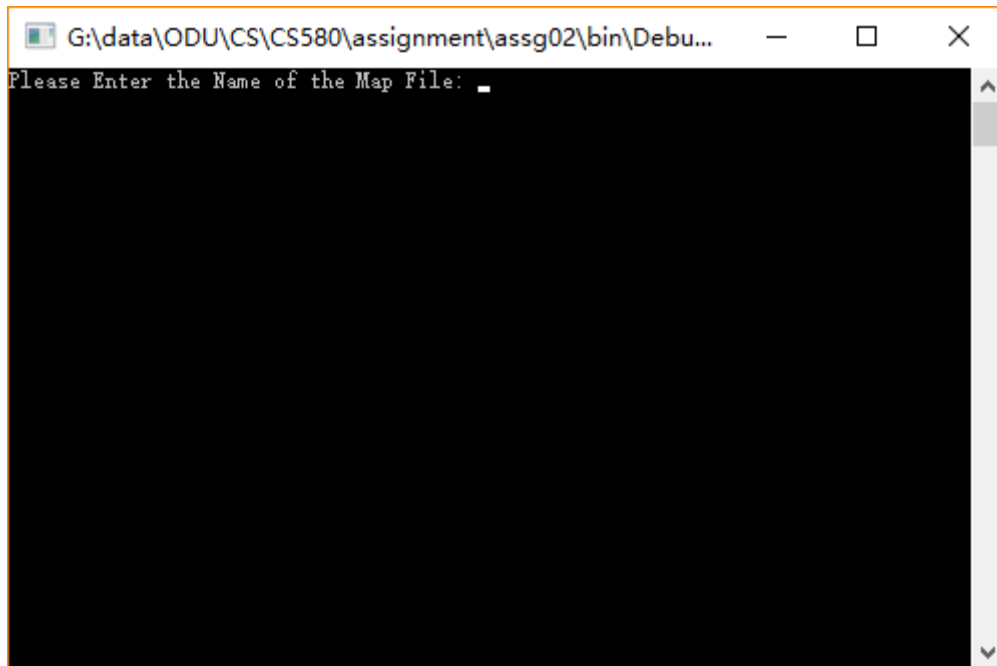
As this figure shows, under the Differential Evolution algorithm, it nearly found the global minimum every time in the total 100 runs. Although in 7 times, it only found the local minimum (not $x=0$ and $y=0$), but the value of the Ackley's function is nearly the same as the global minimum (exactly the same in 4 decimal places).

2. Traveling Salesman Problem (TSP)

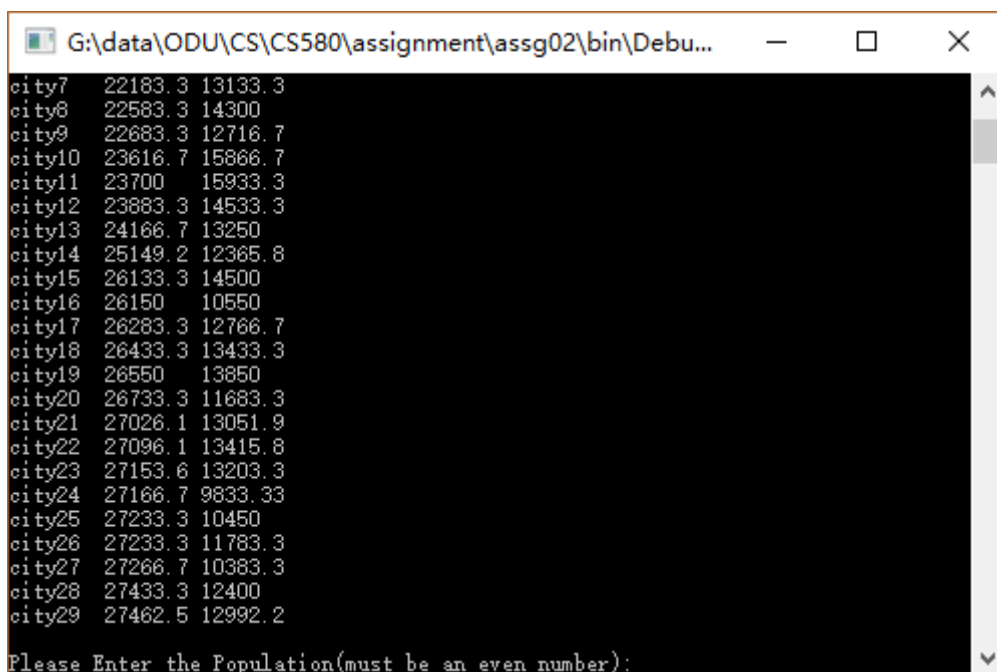
2.1 Execution

- The folder named as 'assg02' is for this question.
- When you run the program, first you will be asked to input the name of map file.

Please enter the full name of the map file (e.g. map.txt).



- Once you enter the name of the map file, it will display the coordinates of the cities. Then you will be asked to enter the population. (It is recommended that the population is 10 times of the cities.)



- d. After you enter the population, it will start to find the solution. When a better solution is found, it will be displayed on the screen including the route and the total distance.

```
G:\data\ODU\CS\CS580\assignment\assg02\bin\Debu...
city23 27153.6 13203.3
city24 27166.7 9833.33
city25 27233.3 10450
city26 27233.3 11783.3
city27 27266.7 10383.3
city28 27433.3 12400
city29 27462.5 12992.2

Please Enter the Population(must be an even number): 300
Route:
26 23 14 20 16 24 13 9 2 6 21 7 19 5 8 18 15 12 11 27 22 25 10 4 17 1 29 3 28
Total Distance: 104625

Route:
26 23 14 20 16 24 13 9 2 6 21 7 19 5 3 8 18 15 12 11 27 25 10 4 17 22 1 29 28
Total Distance: 90902.3

Route:
26 23 20 14 16 24 13 9 2 6 21 7 19 5 1 8 18 15 12 11 27 25 10 4 17 22 28 29 3
Total Distance: 90466.7

Route:
2 12 20 14 16 24 5 1 9 7 6 21 26 19 28 13 8 18 15 23 11 10 4 17 25 22 29 27 3
Total Distance: 86501.3
```

- e. Every 10 seconds, it will show how many generations have been generated.

```
G:\data\ODU\CS\CS580\assignment\assg02\bin\Debu...
Route:
20 28 21 11 12 13 19 26 15 23 17 25 27 29 5 9 18 4 6 2 1 22 7 10 3 8 16 14 24
Total Distance: 78141

Route:
20 28 21 11 12 13 19 26 15 23 17 25 27 29 5 9 18 4 2 1 6 7 10 3 8 16 14 24 22
Total Distance: 73546.5

Route:
20 28 21 11 12 13 19 26 15 23 17 25 27 29 9 18 4 2 1 6 5 7 10 3 8 16 14 24 22
Total Distance: 69690.7

Run Time: 10    Generation: 4899
Route:
29 24 25 27 21 22 13 6 2 11 18 28 17 3 10 4 14 19 23 20 26 16 15 8 12 5 1 7 9
Total Distance: 69246

Run Time: 20    Generation: 9928
Route:
20 18 22 23 17 8 4 27 19 26 29 15 21 28 7 3 9 24 25 14 5 10 12 11 6 2 1 13 16
Total Distance: 65500.3

Run Time: 30    Generation: 14901
Run Time: 40    Generation: 19836
```

- f. If there has not been any better solution found in last 300 seconds, the program will stop and display the final solution including the route and the total distance.

```

G:\data\ODU\CS\CS580\assignment\assg02\bin\Debu...
Run Time: 170 Generation: 85941
Run Time: 180 Generation: 90978
Run Time: 190 Generation: 95945
Run Time: 200 Generation: 100983
Run Time: 210 Generation: 105975
Run Time: 220 Generation: 110952
Run Time: 230 Generation: 116015
Run Time: 240 Generation: 121043
Run Time: 250 Generation: 126018
Run Time: 260 Generation: 130910
Run Time: 270 Generation: 135885
Run Time: 280 Generation: 140848
Run Time: 290 Generation: 145852
Run Time: 300 Generation: 150835
Run Time: 310 Generation: 155782
Run Time: 320 Generation: 160781

Solution:
Route:
20 18 22 23 17 8 4 27 19 26 29 15 21 28 7 3 9 24 25 14 5 10 12 11 6 2 1 13 16
Total Distance: 65500.3

Process returned 0 (0x0)   execution time : 8708.371 s
Press any key to continue.

```

2.2 Approaches

In this program, 2 crossover algorithms and 2 mutation algorithms are implemented with the crossover probability of 60% and mutation probability of 10%. The 2 crossover algorithms and 2 mutation algorithms have the same chance to be chosen.

The first crossover algorithm is PMX and the second one is Revised PMX in which I modified the mapping strategy of PMX. In PMX, it starts from the first crossover point looking for elements in that segment of P2 that have not been copied. But in Revised PMX, it starts from the left side looking for elements not in that segment of P2 that have not been copied (it's just like the last step in PMX). Then it fills the rest of the offspring with those elements that have not been dealt with from the crossover segment in P2 in the same order.

Step1:



Step2:



Step3:



The first mutation algorithm is Swap Mutation and the second one is Insert Mutation.

2.3 Analysis

Nearly every time, this program will get a solution with total distance less than 70000. The best solution I have got is 57482.5, and the route is 26, 17, 7, 4, 8, 10, 5, 1, 2, 6, 3, 9, 14, 28, 24, 16, 27, 29, 21, 18, 19, 11, 22, 20, 15, 13, 12, 23, 25.