exampleTSP

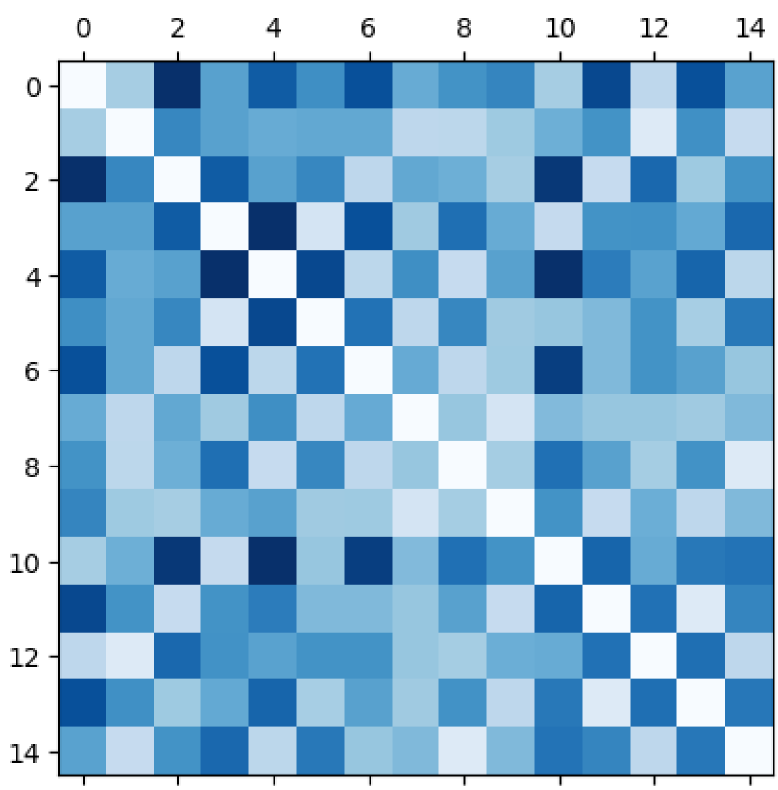
December 10, 2021

1 Traveling Salesman Problem With Genetic Algorithms

[ ]:
```
from GeneticTravelModel import *
#Import Data
data15=importCoordinates("15Cities.txt")
data48=importCoordinates("48Cities.txt")
```

The next matrix is the Distance Matrix. Each column j and row i represents a city while the intersaction [i,j] represents a distance. [i,i] and [j,j] are white because of this.

[ ]:
```
DM=coordinateMatrix(*data15)
plt.matshow(DM, cmap=plt.cm.Blues)
plt.show()
```



1.1 Genetic Algorithms Application

1.1.1 Model 1

1.1.2 15 Cities

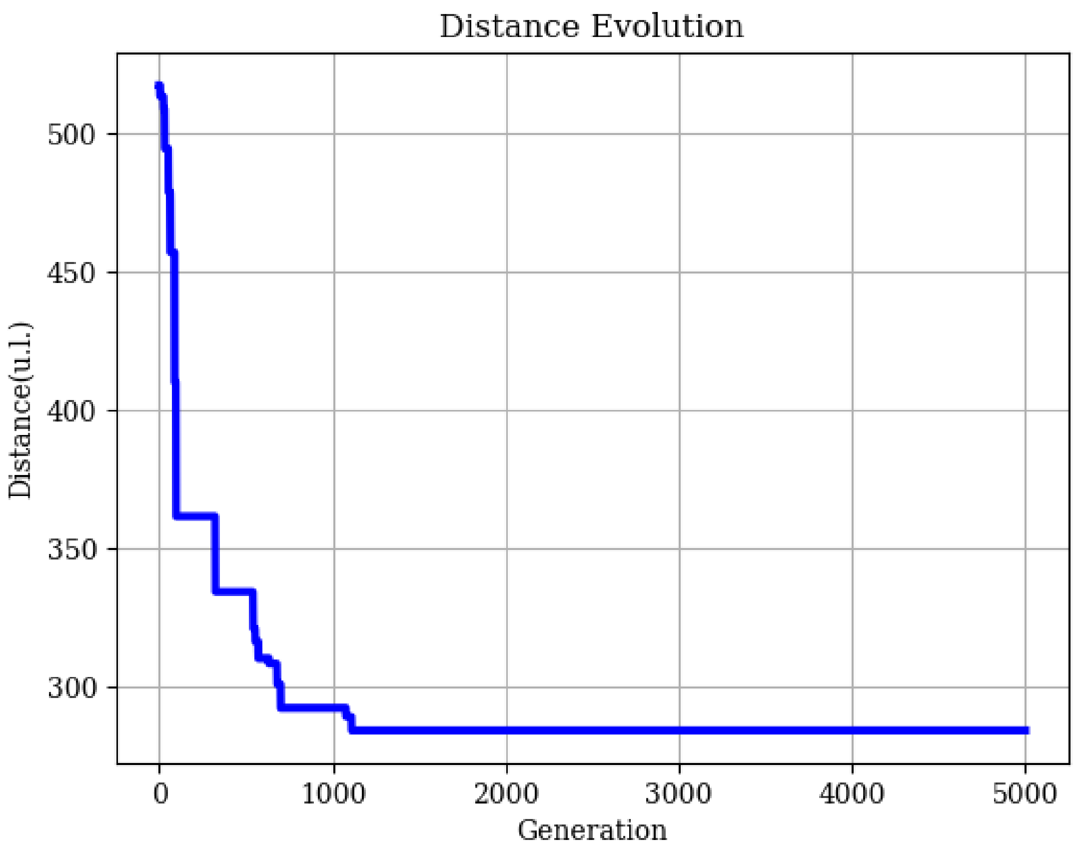
Execution time: 1.2s

[ ]:
```
popSize=30; iters=5*10**3
histPOP, histDist, pop=GeneticModel01(*data15, popSize, iters, 1, True)
print("Model 1 Done")
```

```
Model 1 Done
```

[ ]:
```
print("Distance", histDist[-1])
plotDistance(histDist)
plotTrayectory(*data15, histPOP[-1])
```

```
Distance 284.38086286247795
```



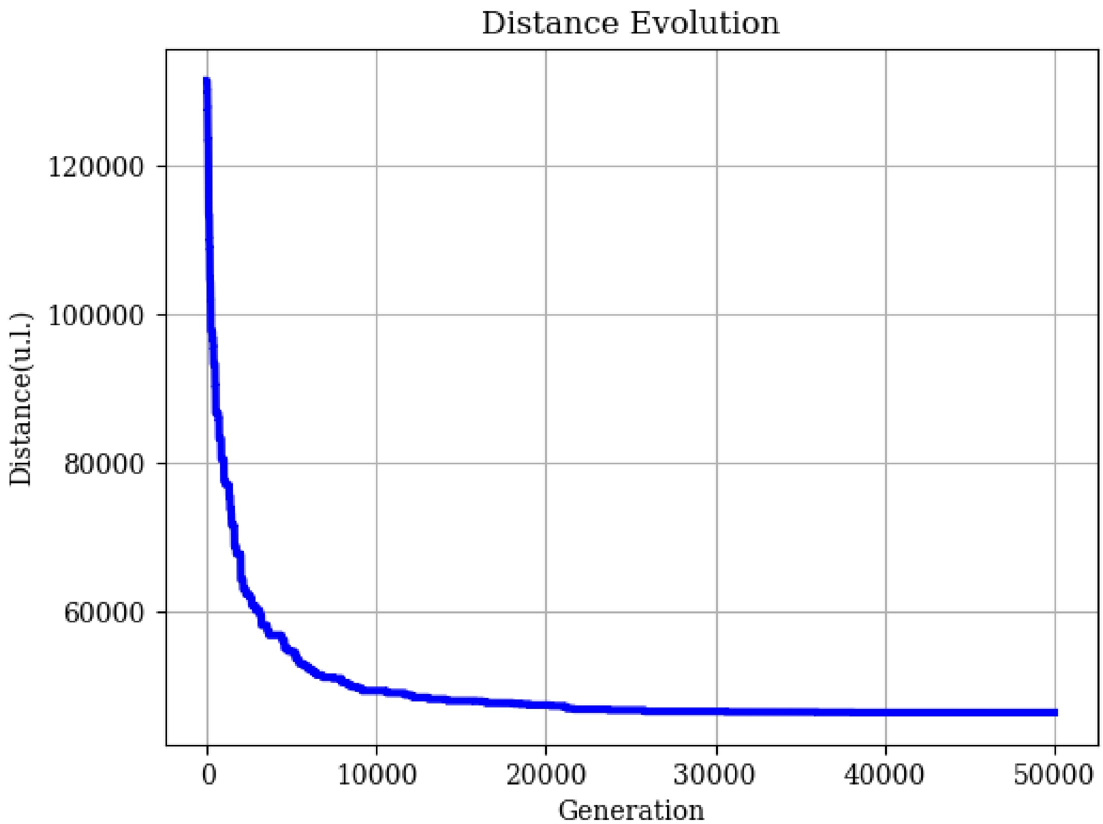
1.1.3 48 Cities

```
[ ]: popSize=50; iters=5*10**4
     histPOP2, histDist2, pop2=GeneticModel01(*data48, popSize, iters, 1, True)
     print("Model 1 Done")
```

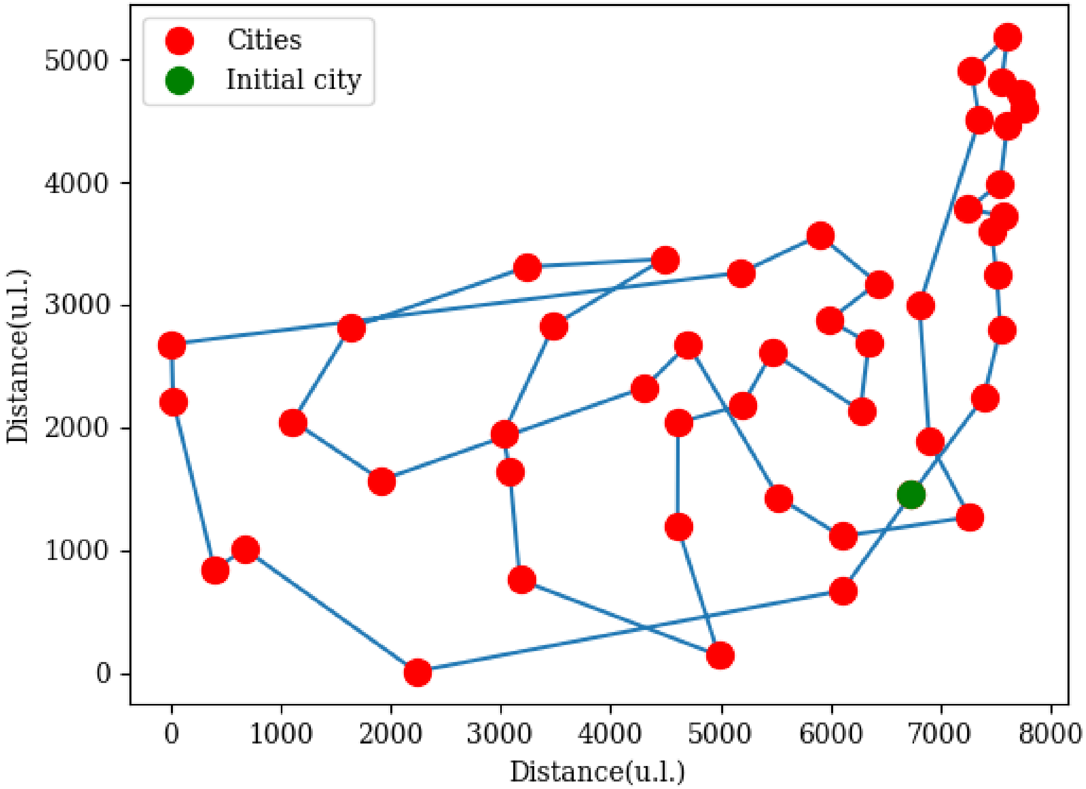
```
Model 1 Done
```

```
[ ]: print("Distance", histDist2[-1])
     plotDistance(histDist2)
     plotTrayectory(*data48, histPOP2[-1])
```

```
Distance 46421.06074219379
```


Distance Evolution
Distance(u.l.)
120000
100000
80000
60000
0
10000
20000
30000
40000
50000
Generation

Map of cities and trayectory travelled by the fittest individual.



1.2 Model 2

1.2.1 15 Ciudades

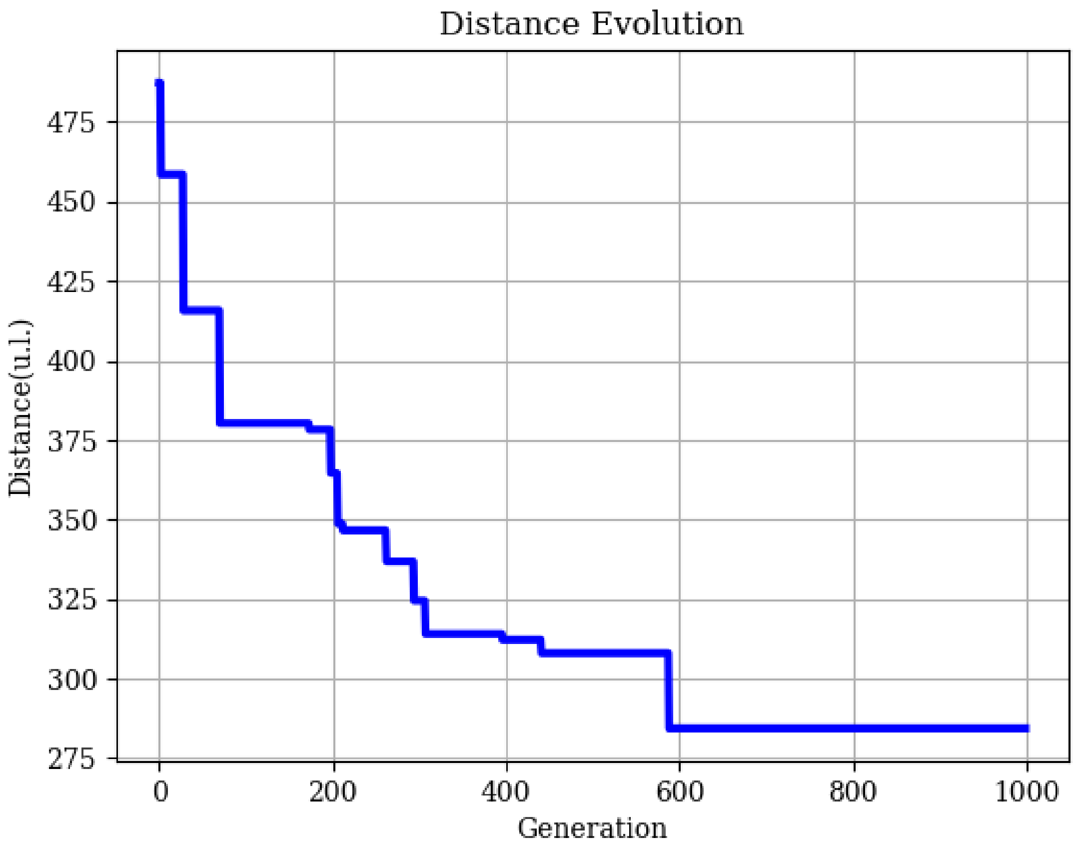
[ ]:
```
popSize=50; iters=10**3

histPOP21, histDist21, pop21=GeneticModel02(*data15, popSize, iters, 10, 2)
print("Model 2 Done")
```

```
Model 2 Done
```

[ ]:
```
print("Distance", histDist21[-1])
plotDistance(histDist21)
plotTrayectory(*data15, histPOP21[-1])
```

```
Distance 284.38086286247795
```



Distance Evolution
Distance(u.l.)
475
450
425
400
375
350
325
300
275
0
200
400
600
800
1000
Generation

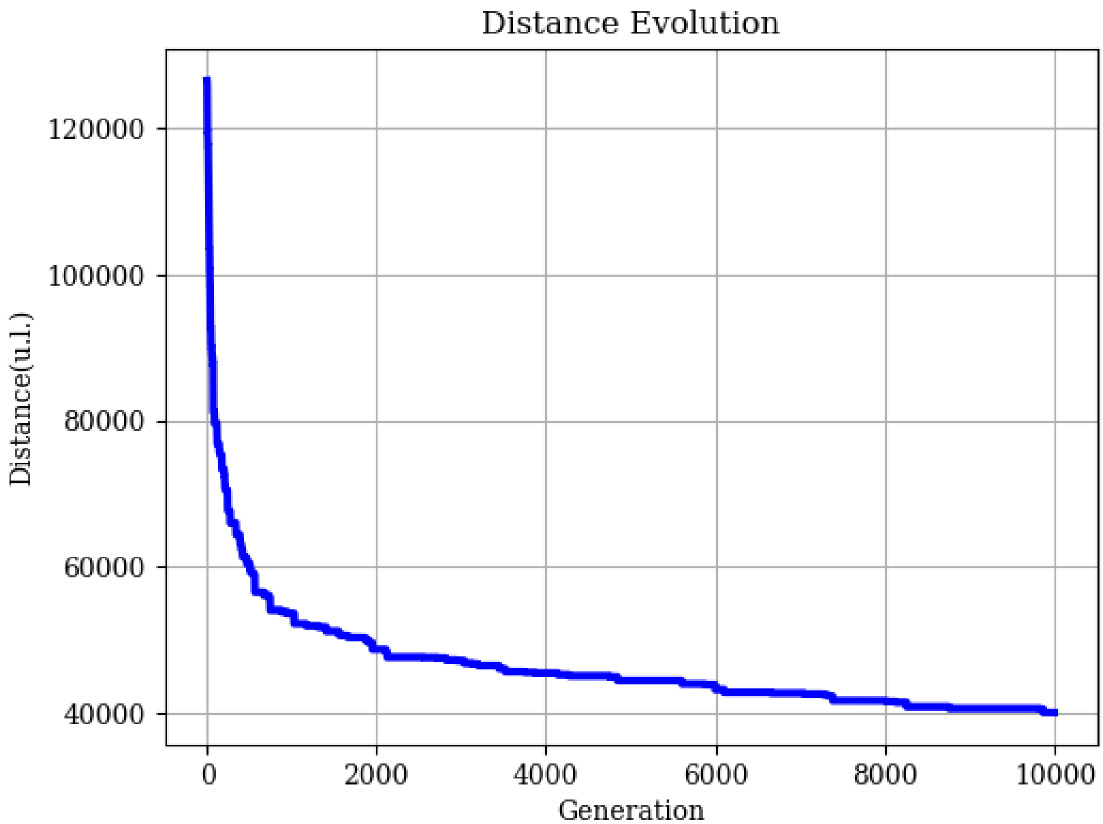
1.2.2 48 Ciudades

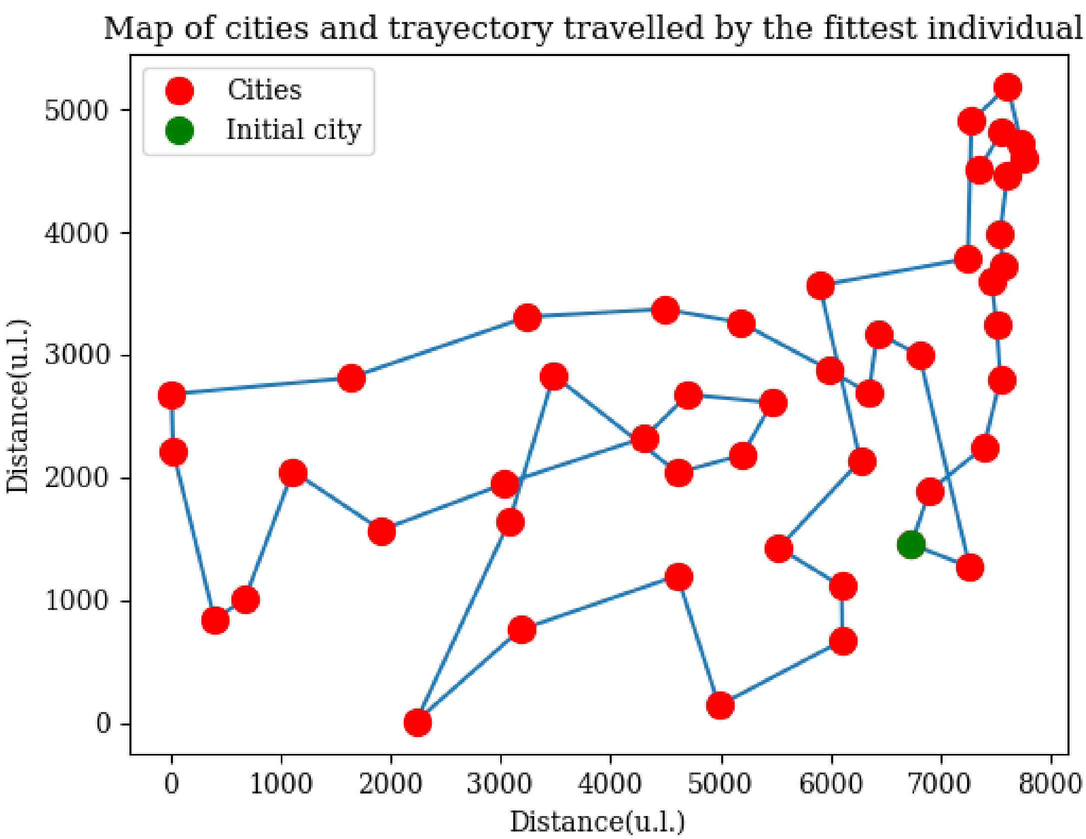
[ ]:
```
popSize=100; iters=10**4
histPOP22, histDist22, pop22=GeneticModel02(*data48, popSize, iters, 50, 2)
print("Model 2 Done")
```

```
Model 2 Done
```

[ ]:
```
print("Distance", histDist22[-1])
plotDistance(histDist22)
plotTrayectory(*data48, histPOP22[-1])
```

```
Distance 39971.3835921356
```


Distance Evolution
Distance(u.l.)
120000
100000
80000
60000
40000
0
2000
4000
6000
8000
10000
Generation



1.3 Model 3

1.3.1 For 15 cities.

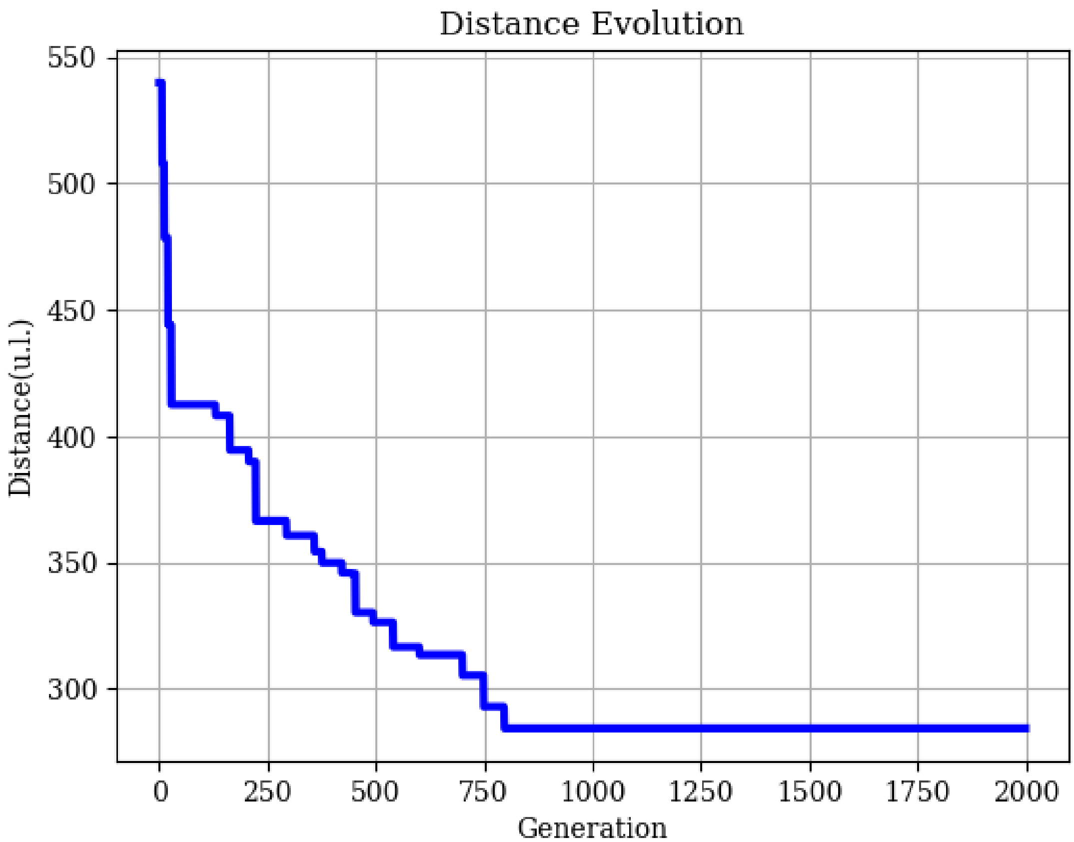
Registered time: 1.3s

[ ]:
```
popSize=50; iters=2*10**3
histPOP3_15_1, histDist3_15_1, pop3_15_1=GeneticModel03(*data15, popSize,␣
↪iters, 2, True, 2, 5)
print("Model 3 Done")
```

Model 3 Done

[ ]:
```
print("Distance", histDist3_15_1[-1])
plotDistance(histDist3_15_1)
plotTrayectory(*data15, histPOP3_15_1[-1])
```

Distance 284.38086286247807



Distance Evolution
550
500
450
400
350
300
Distance(u.l.)
0
250
500
750
1000
1250
1500
1750
2000
Generation

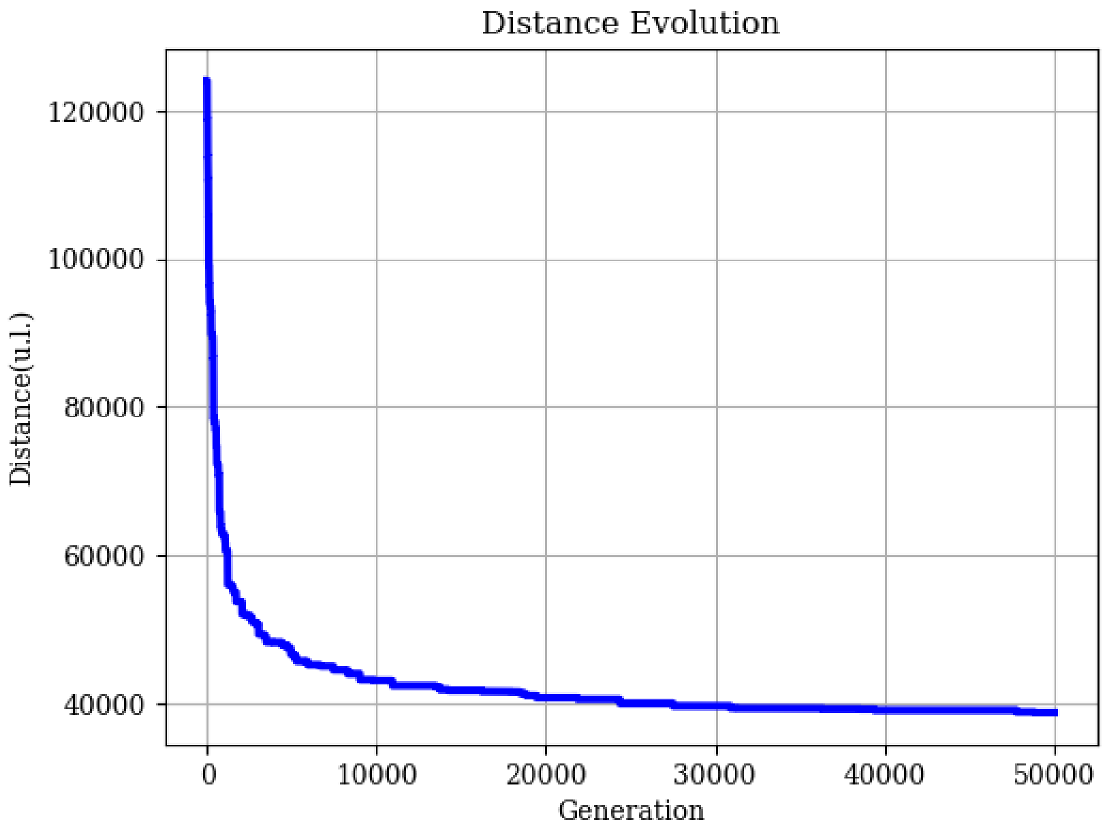
1.3.2 For 48 cities

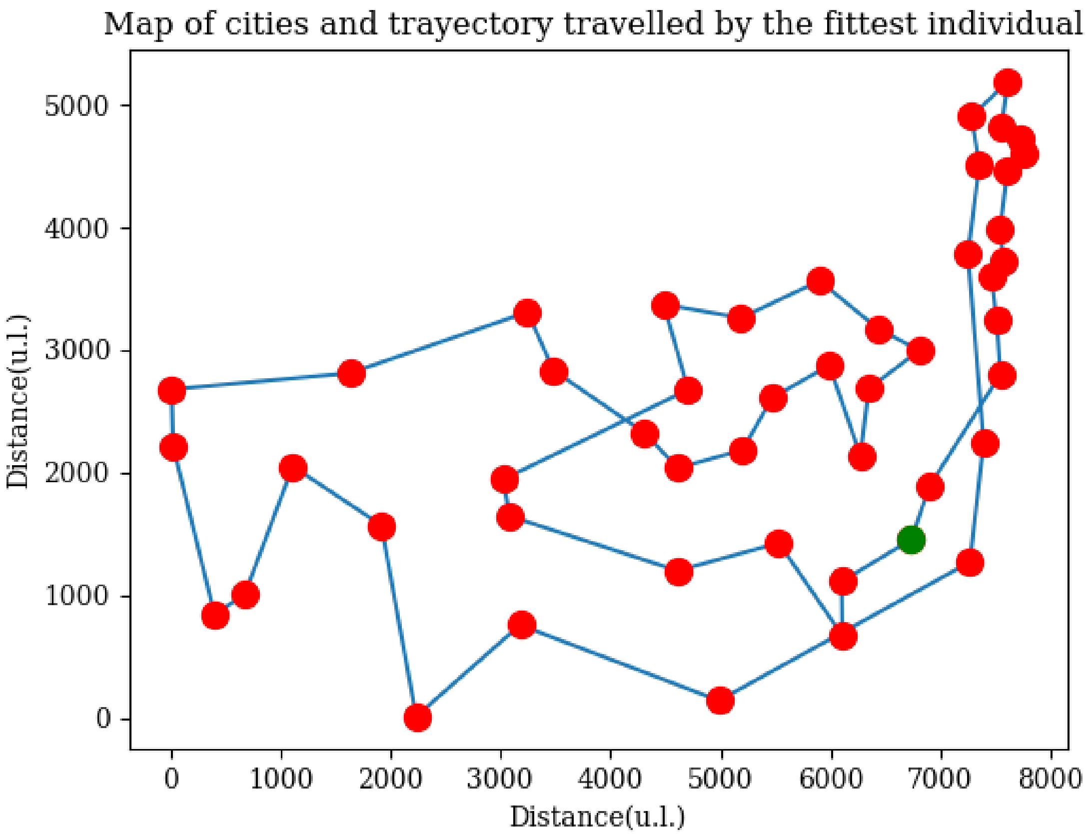
```
[ ]: popSize=50; iters=5*10**4
     histPOP3_15_2, histDist3_15_2, pop3_15_2=GeneticModel03(*data48, popSize,␣
      ↪iters, 2, True, 2, 5)
     print("Model 3 Done")
```

```
Model 3 Done
```

```
[ ]: print("Distance", histDist3_15_2[-1])
     plotDistance(histDist3_15_2)
     plotTrayectory(*data48, histPOP3_15_2[-1])
```

```
Distance 38725.18365968017
```


Distance Evolution
120000
100000
80000
60000
40000
Distance(u.l.)
0
10000
20000
30000
40000
50000
Generation

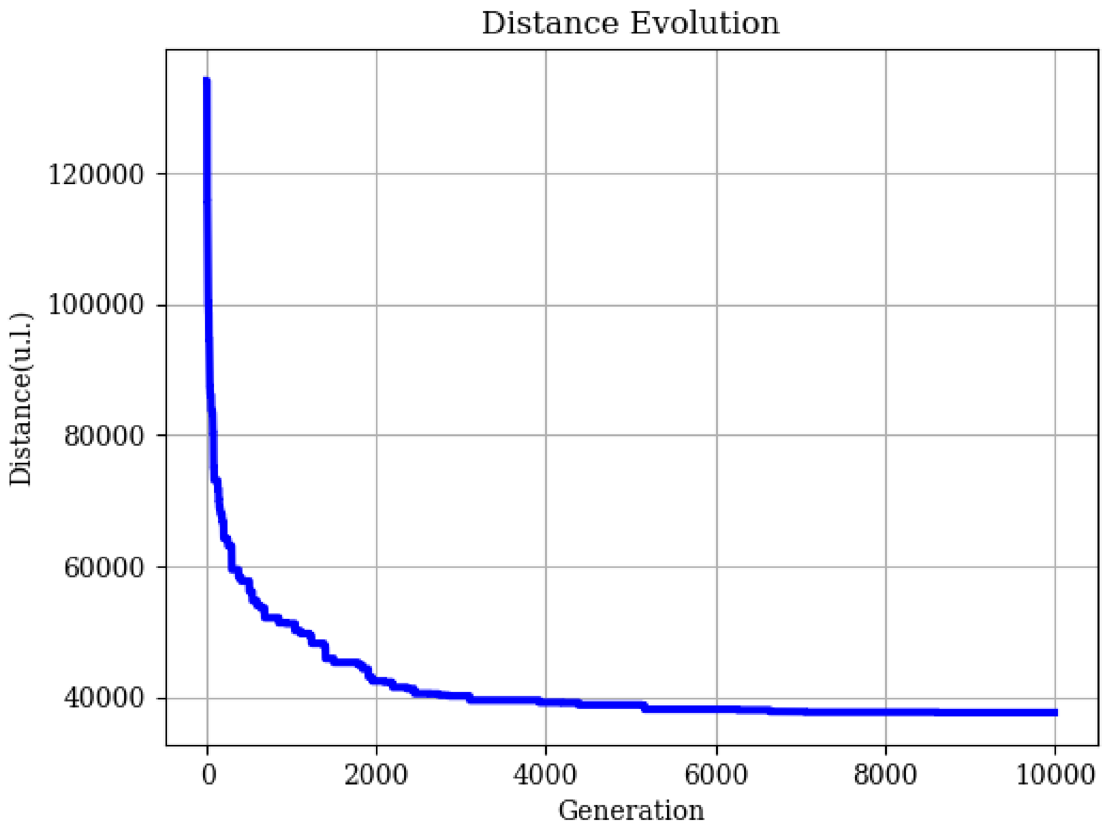


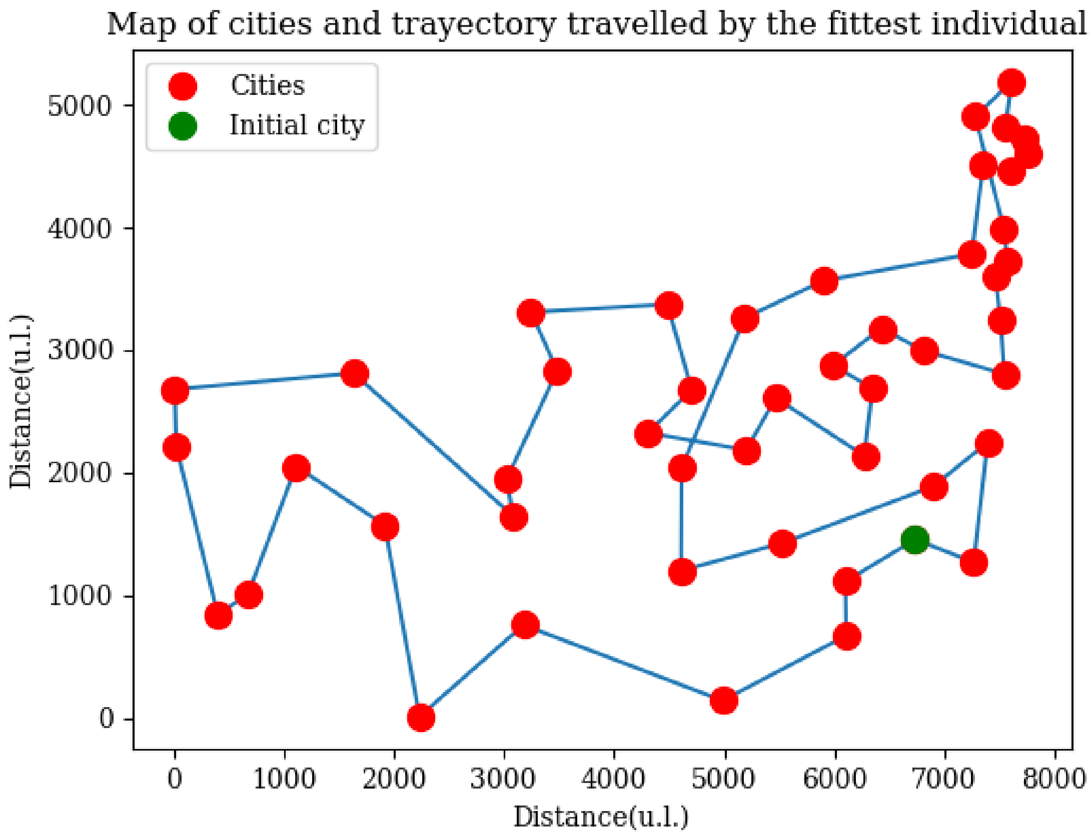

```
[ ]: popSize=100; iters=10**4
     histPOP3_15_2, histDist3_15_2, pop3_15_2=GeneticModel03(*data48, popSize,␣
      ↪iters, 2, True, 7, 50)
     print("Model 3 Done")
```

```
Model 3 Done
```

```
[ ]: print("Distance", histDist3_15_2[-1])
     plotDistance(histDist3_15_2)
     plotTrayectory(*data48, histPOP3_15_2[-1])
```

```
Distance 37615.24027529461
```


Distance Evolution
120000
100000
80000
60000
40000
Distance(u.l.)
0
2000
4000
6000
8000
10000
Generation



Map of cities and trayectory travelled by the fittest individual.
Cities
Initial city
Distance(u.l.)
Distance(u.l.)
0
1000
2000
3000
4000
5000
0
1000
2000
3000
4000
5000
6000
7000
8000