06.09.2018

Navn: Torstein gombos

Username: Tagombos

Travelling Salesman Problem

Biologically Inspired Computing

# Introduction

The travelling salesman problem is an optimization problem about finding the shortest route between cities around the world. I will in this report implement various optimization methods and test performance on time and result.

# Tools

I program the methods using python 3.6. The data used comes from “European\_citites.CSV”.

# Exhaustive search

## Questions

1. Question: What is the shortest route and what is the distance?

Answer: Implementing exhaustive search for 10 cities yielded following route:

***The shortest route using exhaustive search:***

*Barcelona Belgrade Istanbul Bucharest Budapest Berlin Copenhagen Hamburg Brussels Dublin Barcelona*

*The total distance is 7486.31km*

*Code execution: 3.715876340866089s*

1. Question: How long did it take the program to find it?

Answer: The code used about 3.7s when finding optimal route for 10 cities

1. Question: How long would you expect it take with all 24 cities?

Since it does not matter what the starting point is as long as the sequence of cities is the same. One can therefore do permutations

|  |  |  |  |
| --- | --- | --- | --- |
| Number of cities | Distance(km) | Time(s) | Permutations |
| 6 | 5018.81 | 0.00203 | 120 |
| 7 | 5487.89 | 0.00697 | 720 |
| 8 | 6667.49 | 0.04188 | 5040 |
| 9 | 6678.55 | 0.36299 | 40320 |
| 10 | 7486.31 | 3.54444 | 362880 |
| 11 | 8339.36 | 39.1216 | 3628800 |

(converted answer from seconds to years)

# Hill Climbing

## Questions

1. Question: How well does hill climber perform on the same first 10 cities?

Answer: Implementing hill climbing for the first 10 cities yielded:

***The shortest route:***

*Hamburg -> Copenhagen -> Berlin -> Budapest -> Bucharest -> Istanbul -> Belgrade -> Barcelona -> Dublin -> Brussels -> Hamburg ->*

*The total distance is 7486.31km*

*Code execution: 0.013934135437011719s*

However, it does not always reach global minimum,.

1. Question: Report the best and worst as well as standard deviation and mean of 20 runs. Both for 24 and 10 cities with random starting tours.

Answer:

### Performing 20 hill climbs on random sequence of 10 cities:

*The shortest route was 7486.31km:*

*Istanbul -> Belgrade -> Barcelona -> Dublin -> Brussels -> Hamburg -> Copenhagen -> Berlin -> Budapest -> Bucharest -> Istanbul ->*

*The longest route was 9410.61km:*

*Belgrade -> Brussels -> Dublin -> Barcelona -> Istanbul -> Bucharest -> Copenhagen -> Hamburg -> Berlin -> Budapest -> Belgrade ->*

*The mean was: 7998.818500000001*

*The standard deviation was: 549.2403150215758*

*Code execution: 0.31919145584106445s*

### Performing 20 hill climbs on random sequence of 24 cities:

*The shortest route was 13483.66km:*

*Barcelona -> Madrid -> Dublin -> London -> Brussels -> Paris -> Milan -> Munich -> Prague -> Vienna -> Budapest -> Belgrade -> Sofia -> Istanbul -> Bucharest -> Warsaw -> Berlin -> Hamburg -> Copenhagen -> Stockholm -> Saint Petersburg -> Moscow -> Kiev -> Rome -> Barcelona ->*

*The longest route was 17955.05km:*

*Saint Petersburg -> Barcelona -> Madrid -> Paris -> Brussels -> Copenhagen -> Stockholm -> Moscow -> Kiev -> Hamburg -> Dublin -> London -> Berlin -> Prague -> Munich -> Milan -> Rome -> Vienna -> Belgrade -> Sofia -> Istanbul -> Bucharest -> Budapest -> Warsaw -> Saint Petersburg ->*

*The mean was: 15190.461*

*The standard deviation was: 1045.0609307640743*

*Code execution: 1.2655680179595947s*

# Genetic Algorithm

The genetic algorithm follows simple GA structure:

|  |  |
| --- | --- |
| Initialize Population | An *x* amount of random generated routes |
| Evaluate Population (Fitness) | Total distance for each route |
| Select Parents | Based on a fitness proportionate selection  Uses windowing to scale probabilities |
| Create Offsprings | Uses partially mapped crossover between selected parents |
| Mutate Offspring | Random swap on a small selection of offsprings to keep some diversity |
| Replace population | If eliteism: 10% of the best of the population will join the next generation  If not: Replace entire pop with offspring (Works better so far) |

## Performing genetic algorithm with three different population sizes:

**Tour length: All 24 cities, Best of: 20 runs,**

**With 10% elites**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Population** | **Best fitness(km)** | **Worst fitness(km)** | **Mean(km)** | **Standard deviation(km)** |
| 500 | 13814 | 17473 | 15187 | 861 |
| 700 | 13191 | 16679 | 15304 | 981 |
| 1200 | 13358 | 15575 | 14343 | 726 |

**With no elites**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Population** | **Best fitness(km)** | **Worst fitness(km)** | **Mean(km)** | **Standard deviation(km)** |
| 500 | 14132 | 18399 | 15187 | 861 |
| 700 | 13191 | 16679 | 15304 | 981 |
| 1200 | 13358 | 15575 | 14343 | 726 |

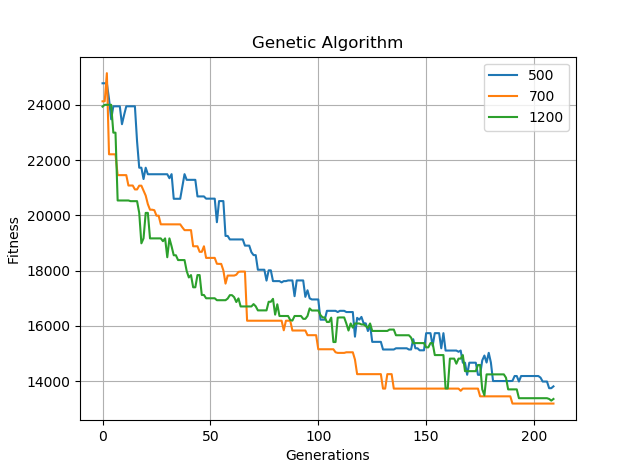
****

Figure 1 Shows the fitness over generations. Legend is population size

|  |
| --- |
| **The best route for population size 500:** |
| **Vienna -> Munich -> Prague -> Copenhagen -> Stockholm -> Saint Petersburg -> Moscow -> Kiev -> Budapest -> Bucharest -> Istanbul -> Sofia -> Belgrade -> Rome -> Milan -> Madrid -> Barcelona -> Paris -> London -> Dublin -> Brussels -> Hamburg -> Berlin -> Warsaw -> Vienna ->** |

|  |
| --- |
| **The best route for population size 700:** |
| **Munich -> Vienna -> Budapest -> Belgrade -> Sofia -> Istanbul -> Bucharest -> Kiev -> Moscow -> Saint Petersburg -> Stockholm -> Copenhagen -> London -> Dublin -> Madrid -> Barcelona -> Rome -> Milan -> Paris -> Brussels -> Hamburg -> Berlin -> Warsaw -> Prague -> Munich ->** |

|  |
| --- |
| **The best route for population size 1200:** |
| **Istanbul -> Kiev -> Warsaw -> Moscow -> Saint Petersburg -> Stockholm -> Copenhagen -> Berlin -> Prague -> Munich -> Hamburg -> Brussels -> Paris -> London -> Dublin -> Madrid -> Barcelona -> Milan -> Rome -> Vienna -> Budapest -> Belgrade -> Sofia -> Bucharest -> Istanbul ->** |

## How well does genetic algorithm perform compared to exhaustive search

Comparisons are performed with one run of GA for both 10 cities and 24 cities to check time. I’am not sure by how many tours were inspected, but the number of evaluated cities for each population should be:

So, for 100 generations which is what was used for 10 cities:

Route length: 10 cities  
Generations: 100

|  |  |  |
| --- | --- | --- |
| Population size: | Time(s) | Best distance(km) |
| 500 | 1.48 | 7486.3 |
| 700 | 2.34 | 7663.3 |
| 1200 | 4.96 | 7486.3 |

We see that genetic algorithm outperforms exhaustive search in time, even though the algorithm uses more generations than necessary. The graph below shows that the best solution already was found by the smallest population at around 70 generations. Though it managed to mutate out of this solution in the end.

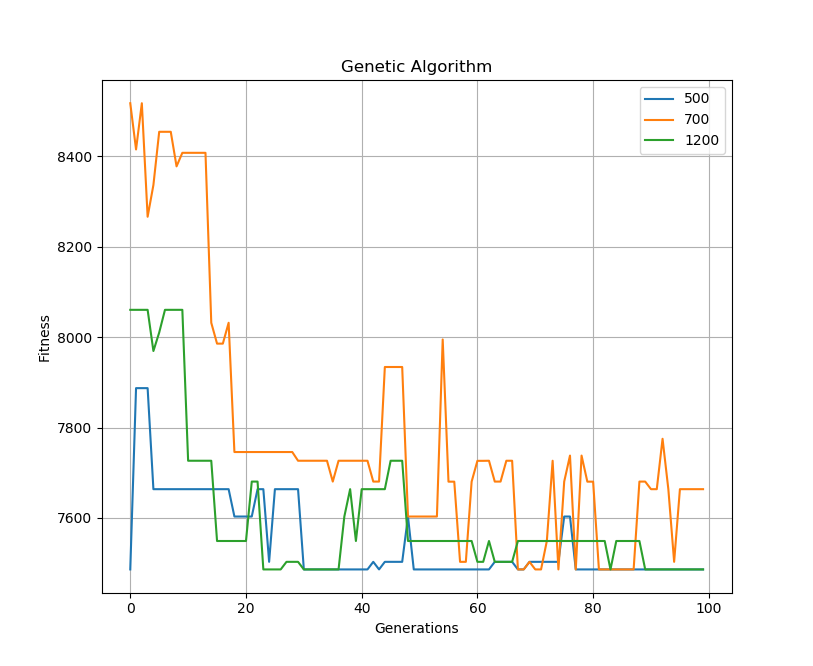


Figure 2. routes are 10 cities long

Route length: 24 cities  
Generations: 200

|  |  |  |
| --- | --- | --- |
| Population size: | Time(s) | Best distance(km) |
| 500 | 5.75 | 15526 |
| 700 | 8.64 | 15229 |
| 1200 | 18.43 | 14107 |

More generations are needed to improve the solution. Though it does what exhaustive cannot.

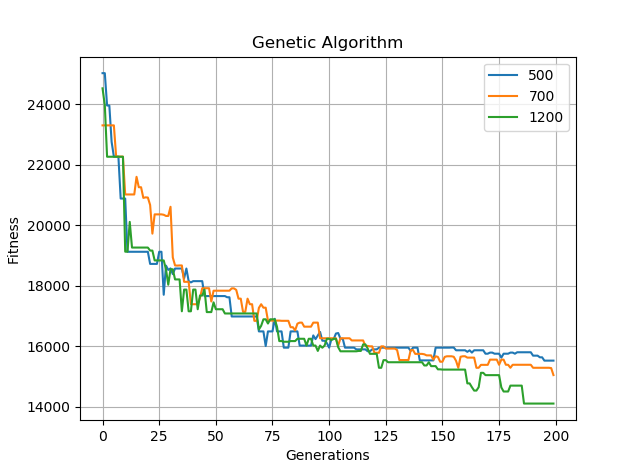


Figure 3 routes are 24 cities long

# Hybrid algorithm

Performing with 100:

Run 0 with 100

The shortest route was 12325.929999999993km:

Istanbul -> Bucharest -> Kiev -> Moscow -> Saint Petersburg -> Stockholm -> Copenhagen -> Berlin -> Hamburg -> Brussels -> Paris -> London -> Dublin -> Madrid -> Barcelona -> Rome -> Milan -> Munich -> Prague -> Warsaw -> Vienna -> Budapest -> Belgrade -> Sofia -> Istanbul ->

The longest route was 12325.929999999993km:

Milan -> Sofia -> Bucharest -> Belgrade -> Paris -> Dublin -> London -> Brussels -> Hamburg -> Copenhagen -> Berlin -> Prague -> Munich -> Madrid -> Barcelona -> Rome -> Istanbul -> Kiev -> Moscow -> Saint Petersburg -> Stockholm -> Warsaw -> Budapest -> Vienna -> Milan ->

Performing with 700:

Run 0 with 700

The shortest route was 12287.070000000007km:

Moscow -> Kiev -> Bucharest -> Istanbul -> Sofia -> Belgrade -> Budapest -> Vienna -> Warsaw -> Berlin -> Prague -> Munich -> Milan -> Rome -> Barcelona -> Madrid -> Dublin -> London -> Paris -> Brussels -> Hamburg -> Copenhagen -> Stockholm -> Saint Petersburg -> Moscow ->

The longest route was 12287.070000000007km:

Munich -> Prague -> Brussels -> London -> Dublin -> Copenhagen -> Stockholm -> Saint Petersburg -> Milan -> Barcelona -> Madrid -> Paris -> Hamburg -> Berlin -> Warsaw -> Vienna -> Rome -> Sofia -> Istanbul -> Bucharest -> Belgrade -> Budapest -> Kiev -> Moscow -> Munich ->

Performing with 1200:

Run 0 with 1200

The shortest route was 12287.070000000007km:

Hamburg -> Brussels -> Paris -> London -> Dublin -> Madrid -> Barcelona -> Rome -> Milan -> Munich -> Prague -> Berlin -> Warsaw -> Vienna -> Budapest -> Belgrade -> Sofia -> Istanbul -> Bucharest -> Kiev -> Moscow -> Saint Petersburg -> Stockholm -> Copenhagen -> Hamburg ->

The longest route was 12287.070000000007km: