

IN3050

Name: Dilixiati Yimuran

Username: Dilixiay

How to run:

- Exhaustive Search:
 - No action needed except for just running the code
 - To change the amount of cities to run, change value in variable “nrOfCities” in line 22. (set to 10)

- Hill Climbing:
 - No action needed except for running the code, unless you want to change the numbers it's going to run, which is set to 20.

- Genetic:
 - At the bottom of the file, there is a function call for “main” function. Which has parameters.
 - N=times it runs (set to 20)
 - PopulationSize= Size of the population you want (set to 60)
 - AmountOfCities= How many cities you want (set to 10)
 - ParentSize= Size of the parent (set to 16) !must be even numbers!
 - ImprovementCount= Times of generations it runs after not being able to find any better solutions. (set to 100). With higher numbers of cities, the higher ImprovementCount is needed (better). With 24 cities, 1000-2000 gives us a alright distance.

Question – Answer

What is the shortest tour (i.e., the actual sequence of cities, and its length) among the first 10 cities (that is,

the cities starting with B,C,D,H and I)? How long did your program take to find it? Calculate an approximation of how long it would take to perform exhaustive search on all 24 cities?

Time: 4.4400414

Score: 7486.309999999999

Route: ['Copenhagen', 'Hamburg', 'Brussels', 'Dublin', 'Barcelona', 'Belgrade', 'Istanbul', 'Bucharest', 'Budapest', 'Berlin']

Report the length of the tour of the best, worst and mean of 20 runs (with random starting tours), as well as the standard deviation of the runs, both with the 10 first cities, and with all 24 cities.

20 Cities:

Minimum Route:

```
['London', 'Paris', 'Brussels', 'Hamburg', 'Copenhagen', 'Stockholm',  
'Saint Petersburg', 'Moscow', 'Kiev', 'Bucharest', 'Istanbul', 'Sofia',  
'Belgrade', 'Budapest', 'Vienna', 'Warsaw\n', 'Berlin', 'Prague',  
'Munich', 'Milan', 'Rome', 'Barcelona', 'Madrid', 'Dublin']
```

Min Distance: 12287.07

```
Maximum Route: ['Barcelona', 'Vienna', 'Budapest', 'Belgrade', 'Kiev',  
'Warsaw\n', 'Prague', 'Berlin', 'Copenhagen', 'Stockholm', 'Saint  
Petersburg', 'Moscow', 'Hamburg', 'Brussels', 'Paris', 'Milan', 'Rome',  
'Sofia', 'Istanbul', 'Bucharest', 'Munich', 'London', 'Dublin', 'Madrid']
```

Max Distance: 16549.000000000004

Execute time: 1.5532814436592162

10 Cities:

```
Minimum Route: ['Copenhagen', 'Berlin', 'Budapest', 'Bucharest', 'Istanbul', 'Belgrade', 'Barcelona',  
'Dublin', 'Brussels', 'Hamburg']
```

Min Distance: 7486.3099999999995

```
Maximum Route: ['Berlin', 'Belgrade', 'Bucharest', 'Istanbul', 'Barcelona', 'Dublin', 'Brussels',  
'Hamburg', 'Copenhagen', 'Budapest']
```

Max Distance: 8843.220000000001

Execute time: 0.027980100000000008

Among the first 10 cities, did your GA find the shortest tour (as found by the exhaustive search)? Did it come close?

For both 10 and 24 cities: How did the running time of your GA compare to that of the exhaustive search?

How many tours were inspected by your GA as compared to by the exhaustive search?

Among 6 and 10 cities it easily found the best route.

Time to execute for 10 cities took almost no time.

Problems occurred when numbers of cities got higher.

I could not find the best route for 24 cities.

For whatever reason my improvementCount has to be 1000+ to be able to get a distance for 24 cities around 14000-18000, otherwise minimum distance is between 19 and 30 thousand.

Explanations of the code is documented as docstrings in each py file.