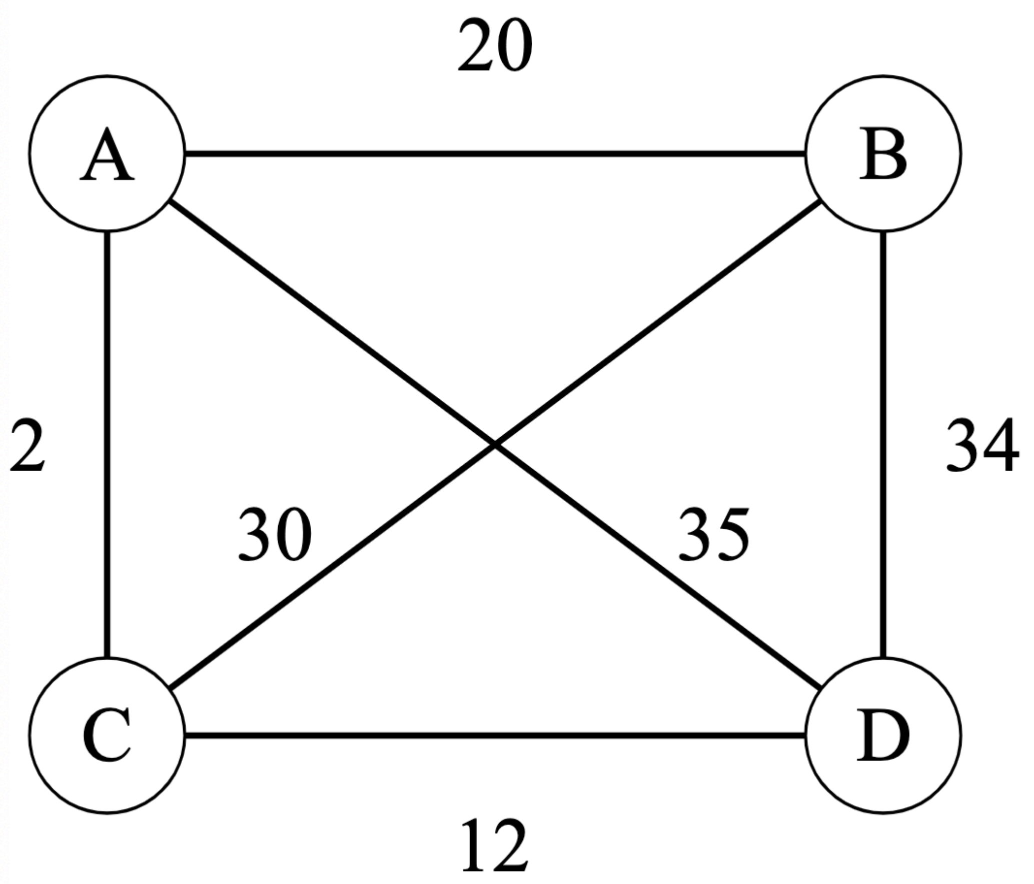
Genetic Algorithm
TSP-Traveling Salesman Problems

Team 529: Wenlang Liu
Qifeng Zhou

Problem Description

The Travelling Salesman Problem (often called TSP) is a classic algorithmic problem in the field of computer science and operation research. It is focused on optimization. In this context *better solution* often means *a solution that is cheaper*. TSP is a mathematical problem. It is most easily expressed as a graph describing the locations of a set of nodes.

Like right graph shows, our goal in this problem is to find the shortest path to visit all cities(ABCD points) exactly once.



Methodology

- **Initial Population Size(50)** : Randomly generated individuals , filled with different Route.
    - Population: { [MA,CA,NY,MI],[CA,MA,NY,MI],[NY,CA,MA,MI]......}

- **Genotype:** Each Route consists with City objects(Array of Cities).
    - Single Route: [MA,CA,NY,MI]
- **Gene Expression:** Cities are randomly distributed within the array.
    - The index of city is random. [MA,CA,NY,MI] and [CA,NY,MI,MA] are in different gene expression

Methodology

- **Selection Process → Tournament Selection**: Randomly pick 5 individuals from entire population as tournament array and individual/chromosome with highest fitness is chosen for breeding
- **Fitness Score** is the *Sum of the distance of all city in a route/array in consecutive order*. Hence, the total distance is the fitness score instead of 1/distance typically used. It was done this way, solely on better illustration during run time as the system evolved.
  - **Best fitness score** = the least distance, worst fitness score = the greatest distance, the lower the score, the shorter the path, the better fit.
- **Distance:** Calculated as the Euclidean distance on 2D space between two cities' **x** and **y** coordinate

Methodology

- **Parents**: Select two best fitness score individuals as parents using **tournament selection**

1	2	3	4	5	6	7	8	9

9	8	7	6	5	4	3	2	1

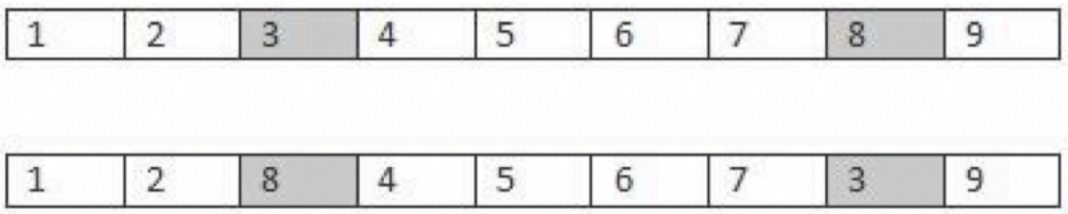
- **CrossOver:** Randomly generate a START and END point. Pick START and END point element from Parent A. Pick rest of element from Parent B keeping in check there is no duplicate gene

					6	7	8	

9	5	4	3	2	6	7	8	1

Methodology

- **Mutation (0.03 Possibility – 3%):** after crossover, child is produced. The child will have certain possibility to mutate. Swap city by index to mutate.



- **Eliminate:** if new child fitness score is better than the worst fitness score individual in the original population, replace the lowest fitness score individual with this child.

- ***Repeat***

Experiment & Result

- Kept City Size at 5
  - All possible path solution is 120 (5!)
  - 10 Evolution : 254m
  - 100 Evolution : 249
  - 1000 Evolution: 249m

- Kept City Size at 20
  - All possible path solution is 2.4329E + 18 (20!)
  - 10 Evolution: 1663m
  - 100 Evolution: 1554m
  - 1000 Evolution: 1290m

City Size	Test Run	Best Fitness Score	Mutation Rate	Possible Path
5	10	254	0.015	120
5	100	249	0.015	120
5	1000	249	0.015	120
10	10	530	0.015	3628800
10	100	524	0.015	3628800
10	1000	497	0.015	3628800
20	10	1663	0.015	2.4329E+18
20	100	1554	0.015	2.4329E+18
20	1000	1290	0.015	2.4329E+18

Conclusion

- GA found useful to find ideal solution when there are many possible solution. **The larger the number of element *(n)* is the individual, the more possible combination of solution *(n!)***

- **When mutation is set to high frequency, it was observed the population array to be more chaotic and the chromosome diversity to be high**

- For a 5 city sized array, the number of possible route is 120. For a 10 city sized array, the number of possible route is 3628800. For a 20 city sized array, the number of possible route is 2.4329E +18. **With an increase number of test run, the system to had more trials to evolve the solution space into a better solutions keeping all other parameters constants**

Main Console Run Screenshot

```
{6,9,13,19,15,12,17,18,1,11,3,14,8,10,7,2,5,4,0,16,}1871.169848661801З
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,19,16,17,18,}1621.4989448613135
{15,16,6,1,17,14,7,3,11,18,13,2,9,8,10,19,5,12,0,4,}1832.7201081911169
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1713.53203210795
{5,3,0,8,6,1,7,2,4,9,10,11,12,13,14,15,16,17,18,19,}1770.812287915596
{2,1,0,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1703.0185352778801
{2,6,1,5,0,3,4,7,8,9,10,11,12,13,14,15,16,17,18,19,}1591.1002693024677
{0,1,2,3,4,5,6,7,9,10,11,8,12,13,14,15,16,17,18,19,}1787.6961107579375
{0,1,2,3,4,7,6,5,8,9,10,11,12,13,14,15,16,17,18,19,}1850.5672826419361
{19,0,1,4,2,5,6,7,8,9,10,11,12,13,14,15,16,17,18,3,}1845.090637854131
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1713.53203210795
{19,0,1,4,2,5,6,7,8,9,10,11,12,13,14,15,16,17,18,3,}1845.090637854131
{2,6,1,5,0,3,4,7,8,9,10,11,12,13,14,15,16,17,18,19,}1591.1002693024677
{19,5,3,0,8,9,6,10,1,7,2,11,12,13,14,15,16,17,18,4,}1822.8939419489886
{4,12,6,8,7,19,13,9,16,11,15,10,3,5,1,14,17,18,2,0,}1859.0535663090377
{2,1,0,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1703.0185352778801
{2,6,1,5,0,3,4,7,8,9,10,11,12,13,14,15,16,17,18,19,}1591.1002693024677
{2,1,0,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1703.0185352778801
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1713.53203210795
{7,2,9,10,1,5,6,11,0,3,8,12,4,13,14,15,16,17,18,19,}1647.0735565315233
{2,1,5,0,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1654.8178989601145
{2,1,0,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1703.0185352778801
{0,1,2,3,7,9,10,5,6,11,8,12,4,13,14,15,16,17,18,19,}1739.306426691123
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1713.53203210795
{2,1,5,0,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1654.8178989601145
{13,1,2,3,4,5,11,0,8,9,6,10,7,12,14,15,16,17,18,19,}1875.5592622056156
{17,15,13,19,5,11,3,0,8,16,9,6,10,1,7,2,18,14,4,12,}1835.1939695357298
{0,1,2,3,4,5,6,7,9,10,8,11,12,13,14,15,16,17,18,19,}1778.0194972892912
{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1713.53203210795
{13,5,3,0,1,2,6,7,8,9,10,11,4,12,14,15,16,17,18,19,}1789.951210923568
{7,2,6,1,5,0,3,8,4,9,10,11,12,13,14,15,16,17,18,19,}1577.0526851086404
{7,2,9,6,1,5,0,3,8,4,10,11,12,13,14,15,16,17,18,19,}1560.8449866182984
{0,1,2,3,4,5,6,7,9,8,10,11,12,13,14,15,16,17,18,19,}1832.4912758329067
{0,1,2,3,4,5,6,7,9,10,11,8,12,13,14,15,16,17,18,19,}1787.6961107579375
{5,11,3,0,8,9,6,10,1,7,2,4,12,13,14,15,16,17,18,19,}1707.0433976393786
Parent A - {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1713.53203210795
Parent B - {2,1,0,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}1703.0185352778801
START:4:END:10
Child Gen     -> {2,1,0,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,}:1703.0185352778801
Fittest Route -> {7,2,9,6,1,5,0,3,8,4,10,11,12,13,14,15,16,17,18,19,}1560.8449866182984
```

Unit Test Carried Out & Passed