Homework #4 – (50 points) **Due:** Oct. 30, 2019 Wednesday

Purpose: This homework is to get you familiar with Genetic Algorithms.

Group Assignment (You may form groups up to two individuals in one group)
Submit your report and source code files (may be zipped) to the Blackboard. Also, submit all the files to the K drive: **K:\Courses\2019-Fall\CPS480-ugur1a\22371842\yourCMUid** folder.
One submission per group is OK. The other member of the group needs to create a text file called "**yourCMUid-myhw4notes.txt**" and put info about the name of his/her project partner in the file, and submit the **yourCMUid-myhw4notes.txt** only (no other files) into both the BlackBoard and K drive.

1. Write a program in a language of your choice to perform a Genetic Algorithm solution for the Travelling Salesman Problem (TSP).

2. Use Distance Matrix – initialized from a data file – to calculate the total distance for a particular path (i.e. the fitness function). Data files: ***MI-part-19-miles.csv*** and ***DE-all.csv*** (and ***MI.xlsx***).

3. Write a report comparing the performance of the GA
Performance measure: the fitness value of the best solution found, and the actual time it takes to find it.

Source code documentation:

Your source should be well-documented.
It must include the following: (in the main program header as comments)

- Course number, your name, assignment number, problem number (if any).
- Group member's names, task distribution – who did what,
- Brief description of what your program does, name of the algorithm used, and
- Instructions how to compile and run your program.

You need to submit:

a) Well documented source code.
b) Execution results saved as a text file.
c) Your own data file(s) if any – no need to include the content of the data files provided.
d) Report of Part 4, including the summary of genetic operators used, the stopping criteria, the evaluate-select-reproduce phases of your GA, the initial best fitness, the final best fitness, and the percent improvement.

4. Problem encoding:

An array of n+1 cities (i.e., an individual chromosome), an $n_x n$ distance matrix.

5. GA Pseudocode

```
T=0;  // generation time
Initialize the population P randomly;
T = 1; done = 0;
While (T <= MAXGEN) and Not done
{
      Evaluate P using fitness function;
      If  reached_the_goal()
          done = 1; exit the loop;
      Select BestP based on fitness values of P
      Reproduce P // keep BestP, replace the Rest_of_P with genetically
                  // modified version of BestP
      T++;
}
```

Initialize: Initialize every individual in the population randomly

Goal test: Skip for this problem.

Genetic operators:

Crossover: The process by which two parent population members are combined to form a child. Specifically, crossover must produce only valid solutions. You are free to implement any crossover function that you wish, but keep in mind that crossover is to improve an intermediate solution.
Crossover is optional for this assignment.

Mutation: A population member may be mutated by swapping the positions of two randomly selected positions (of an individual). Mutated version must be a valid path of the TSP.
Any other mutation strategy can be used**.**

6. Inputs to GA:

a)	**RANDSEED**	the seed for random number generator
b)	**MAXGEN**	the maximum number of generations
c)	**POPSIZE**	the population size
d)	**BESTRATE**	the percent of population kept as best individuals (20%) – use the round() function
e)	**MUTRATE**	the probability of mutation

Code a simple GA to solve the problem described above. To do this you need to encode the problem, you may use single point crossover and/or two-position flip mutation. Then run your GA on both the .csv data files. Also, try the Challenge data file: ***MI.xlsx*** to see how much improvement your GA can make for a solution.

GA Tune Up: Perform the following changes on your GA code (one by one) and compare the results:

Change the MAXGEN twice	{10,000, 50,000}
Change the RANDSEED twice	{1000, 4321}
Change the mutation probability twice	{0.1, 0.3}
Change the population size twice	{20, 40} (implication: changing the starting point – initial solutions)

Add your tune-up results to the final report.

Data files:

[All comma separated files, city names are also at the top, some include space character(s) in the name.]

MI-part-19-miles.csv	Distances (in miles) of first 19-cities (in alphabetical order) in Michigan
DE-all.csv	Distances (in miles) of all 51-cities (in alphabetical order) in Delaware

Challenge data file: ***MI.xlsx*** Distances (in miles) of all 440 cities (in alphabetical order) in Michigan
(raw data in Excel, need to convert to a csv file with diagonal entries being all zero.)