

## Assignment #6 – Genetic Algorithm

### **Project Introduction**

As we saw in the last problem, the runtime of the traveling salesman problem increases dramatically as we add new nodes to the system. This tells us that we need a different approach if we are to get solutions to problems with a relatively large amount of nodes. The genetic algorithm aids us in finding an optimal solution when we have too many nodes to solve using the brute force method. The idea behind using a genetic algorithm is to quickly give an approximation of an optimal solution to a problem. This algorithm's model borrows its idea from nature by taking two parent solutions and create a child solution that may contain the “best” features from each parent. To add more complexity to the system we introduce mutations to the children which helps us avoid solutions that are local minimums.

### **Code Overview**

The client will bring up the GUI class which will instantiate a copy of the GeneticAlgorithm class. The GeneticAlgorithm class will return a solution to the Traveling Salesman Problem using a genetic algorithm. Users will be able to select the mutation rate (0.00, 0.02, 0.04, and 0.06) and selection method (order and distance). When the user clicks "Run" it will run the algorithm and print the solution (Determined after 10,000 iterations). The user is then able to change the options and solve the problem again using the same data points. When the user hits reset, the GeneticAlgorithm class is re-instantiated which also generates a new set of data points.

### **Tours**

Tours are represented by a Tour object that consists of an integer array and a double value. Each tour tells us the order in which cities are visited and the distance of the tour. When sorting tours in our population data structure, we sort by the distance of the tour (The tour with the shortest distance is the best). I found the best data structure to hold the Tour objects was an ArrayList. The ability to use Collections.sort() on the List made the selection process much more streamlined.