Ryan Kennedy

**Design Architecture:**

I designed my program very similarly to how I designed my program for lab 3. I have a factory pattern as well as a strategy pattern. I chose to use a factory pattern since it allows me to add another type of algorithm, like a sorting algorithm, to my program without having to change a lot of code. I implemented the factory pattern with the virtual Algo.h class that is implemented by the TSPAlgo class. The TSPAlgo class is where the file is loaded in through a filemanifest.txt, output stats are printed to the console and saved to the algorithms .csv file, the desired algorithm is selected, and the selected algorithm is executed. I chose to create a strategy pattern for the TSP algorithms since both algorithms had similar functionality that could be abstracted behind a class. I chose to create TSP.h that is implemented like a strategy pattern. Both my tabu and genetic algorithm class implement the TSP class. This will also allow me to add more TSP algorithms to the program without having to change a lot of code. My UML diagram is below.



**All Algorithms Compared:**

**A screenshot of a social media post

Description automatically generated**

In this graph I am comparing all 4 algorithm types. Since brute force is significantly slower than the other 3 algorithms it is hard to visualize on the graph, so I stopped adding data to this column after 10 nodes. From the data table I discovered that the dynamic programming approach was faster than all approaches up until 14 nodes are in the graph. At this point the dynamic programming approach starts to slow down and tabu and genetic algorithm are faster now. Even though tabu and genetic algorithm find solutions faster than the dynamic programing approach, these solutions are not optimal solutions but instead are relatively small paths. The equation for the dynamic programming approaches trendline is y = 0.0074e0.7697x with a R² = 0.9815. The equation for tabu is y = 0.8356x2 - 7.3583x + 13.527 with a R² = 0.9695. From these equations I can tell that tabu will be faster since it is a big(O) = x2  while the dynamic programming approach big(O) = ex. I was not able to figure out a trendline for the genetic algorithm since the timing curve for this algorithm is nearly straight and does not seem to depend on the number of nodes in the graph.

**Results:**  A screenshot of a map

Description automatically generated From this graph we can tell that the timing of tabu relates directly to the number of nodes, while the genetic algorithm does not seem to relate to the number of nodes in the graph. We can tell that tabu follows the polynomial trendline with a close R² value meaning that it is a close approximation to the true equation of the timing of this algorithm. In tabu I included two ways to select neighborhoods, I had random selection and a selection for every neighbor. I found that the selection that iterates through all neighbors preformed much more consistently than the random selection. In the genetic algorithm, I included 3 types of selection techniques, random, elitist, and roulette. The random technique selects two random parent nodes each generation. The elitist technique selects the 2 best parents in each generation. Finally, the roulette technique is a random technique that gives better fit parents a higher probability to be choosen. I had a variable that would control the population size, generation size, and mutation chance so that I could better tune my genetic algorithm to produce better results. I found that having a population over about 200 made the algorithm output mostly randomized results. I also noticed that having a high mutation chance leads to more randomized data, so I left the chance to be a 20% chance of mutation.

A screenshot of a cell phone

Description automatically generated

In this graph I am comparing the size of the path to the number of nodes in the graph. At about 6 nodes in the genetic algorithm starts to deviate off of the optimal path and create worse results. The tabu algorithm stays relatively optimal until a graph size of 10 nodes and then the search outputs less optimal solutions. The tabu path stays much smaller than the genetic algorithms path, which shows this algorithm to be more precise. The two algorithms have drawbacks to each of them, while the genetic algorithm may give a suboptimal path, it can find the path much faster than tabu. Tabu on the other hand will give a shorter path, but will take more time to complete the search depending on the size of the graph.