WGU C950 DSA2

NHP2 – NHP2 TASK1: WGUPS ROUTING PROGRAM

## Identify a named self-adjusting algorithm

I used a **genetic algorithm** to determine a near ideal route by randomly guess a subset of the total permutations of routes and mix the best routes (shortest) together and repeated this for a set number of times. The more iterations of this the shorter the route, but the longer it will take to run.

## Write an over of your program

1. **Explain the algorithm’s logic using pseudocode.**

For each truck

Create a list of package ids to be loaded in a list.

Convert the list of package ids to the index of the address in distance matrix.

Randomly select 25 permutations of the route

Loop for N iterations:

Evaluate the route by taking a list of routes and calculating the total distance of the route, but the route will get a 100-mile penalty if any package is delivered after its deadline.

The shortest route will be added to the best list and to the parent list and a list of vector probabilities will be returned and will randomly be picked to add to the parent list until there are 4 routes.

Once there are 4 routes in the parent list, a mutation may be done (probably of 90%) by swapping the first part of one route with the second part of another route without repeating a stop, then possibility a swapping any of the two stops may occur (probability of 90%).

Loop

The best route will be returned along with the distance of the route.

Next truck

1. **Describe the programming environment you used to create the application.**
   1. The application was developed with Python 3.11 with a virtual environment of pipenv in VS Code.
2. **Evaluate the space-time complexity of each major segment of the program and the entire program using big-O notation.**
   1. Main.py
   2. Helper.py
   3. Genetic.py
   4. Truck.py
   5. HashTable.py
      1. Time complexity: inserts are O(1) and lookups will be O(n) once the list is full.
      2. Space complexity: O(n)
   6. Package.py
   7. ExcelToCSV.py