C950 Task-2 WGUPS Write-Up

(Task-2: The implementation phase of the WGUPS Routing Program).

(Zip your source code and upload it with this file)

Student Name

ID #011023634

WGU Email: afordj1@wgu.edu

Date 8/21/2023

C950 Data Structures and Algorithms II

# F1. Strengths of the Chosen Algorithm

a. Greedy Nearest Neighbor Approach: The algorithm uses a nearest neighbor approach to determine the closest package delivery order for each truck. This approach is efficient and ensures that packages are delivered to nearby locations first, reducing the overall distance traveled by the trucks. It minimizes the travel time between consecutive deliveries.

b. Real-time Time Tracking: The algorithm tracks the time and updates the delivery time for each package as it is being delivered. This real-time tracking allows the system to accurately maintain the delivery time of each package and determine its status (at hub, in route, delivered) based on the provided time input.

# F2. Verification of Algorithm

# a. Package Deadline and Status: The algorithm ensures that packages are delivered on time by considering the package deadline. If the delivery time is after the package's deadline, it updates the package status to "Delivered." If the delivery time is before the deadline, it updates the status to "In route." This verification meets the requirement of delivering packages on time and updating their status accordingly.

# b. Total Distance Limit: The algorithm calculates the total distance traveled by each truck and ensures that the combined total distance traveled by all trucks is kept under the specified limit of 140 miles. This verification satisfies the requirement of keeping the distance traveled within the specified limit.

# c. Real-time Tracking: The algorithm accurately tracks the progress of each truck and package by updating their delivery times and statuses. This real-time tracking meets the requirement of allowing the supervisor to monitor the progress of each truck and package at any given time.

# F3. Other Possible Algorithms

. Clark-Wright Savings Algorithm:

The Clark-Wright Savings Algorithm is a well-known heuristic algorithm for solving the vehicle routing problem. It is designed to determine an efficient route for a fleet of vehicles to deliver a set of customers' demands. The algorithm constructs a savings matrix based on the potential savings achieved by combining two separate routes into a single route. It then greedily selects routes with the highest savings until all packages are assigned.

. Dijkstra's Shortest Path Algorithm:

Dijkstra's algorithm is a graph-based algorithm used to find the shortest path between two nodes in a graph. It can be adapted to solve the vehicle routing problem by treating each address as a node in the graph and calculating the shortest paths from the depot to all delivery locations.

# F3a. Algorithm Differences

Differences from the Implemented Algorithm:

The Clark-Wright algorithm focuses on combining routes to minimize the total distance traveled by the trucks, rather than considering nearest neighbors individually.

It optimizes routes by creating a savings matrix and selecting routes with the highest savings, which can lead to better overall route efficiency.

Differences from the Implemented Algorithm:

Dijkstra's algorithm calculates the shortest path based on edge weights (distances) between nodes, aiming to find the most efficient route from the depot to each delivery location.

It doesn't prioritize packages' individual deadlines explicitly, but it aims to minimize the distance traveled, which indirectly contributes to meeting delivery deadlines.

Unlike the implemented algorithm, Dijkstra's algorithm might not consider the order in which packages are delivered but focuses on the shortest path between locations.

# G. Different Approach

Ant Colony Optimization (ACO):

ACO is inspired by the foraging behavior of ants and is used to find optimal paths in graphs. I would adapt ACO to create efficient delivery routes by simulating the movement of ants on the streets of the city. Ants deposit pheromones on the paths they travel, and the algorithm biases future ant movements based on the pheromone levels.

Modifications:

Convert the city's street network into a graph and apply ACO to find optimal routes for each truck.

Use pheromone updates and evaporation to adjust the attractiveness of different paths based on package deliveries, distances, and deadlines.

Experiment with parameter settings such as pheromone evaporation rate and ant movement probabilities to achieve the desired trade-off between exploration and exploitation.

# H. Verification of Data Structure

Chaining Hash Map (HashTableMap) - Used in the Solution:

Strengths:

Efficient data retrieval: Hash tables provide O(1) average-case time complexity for insertion, lookup, and deletion operations.

Collisions handling: Chaining handles collisions by storing multiple items in the same bucket, reducing the likelihood of collisions affecting performance.

Dynamic resizing: The hash table can dynamically resize itself to maintain a balanced load factor and optimize performance.

Verification of Scenario Requirements:

Efficient lookup: The hash table allows quick retrieval of package information based on package IDs.

Package management: The hash table effectively stores and manages package information, allowing for easy insertion, lookup, and modification of package attributes.

Package tracking: The hash table can store and update package delivery status and associated information.

# H1. Other Data Structures

Binary Search Tree (BST):

In a binary search tree, each node stores a package's information, and the tree is organized based on package IDs. This structure supports efficient insertion, deletion, and lookup operations, making it suitable for package management and retrieval. However, it may not provide the constant-time performance guarantees of a hash table in all cases.

Priority Queue (Heap):

A priority queue could be used to manage packages based on their delivery deadlines. Each package would be assigned a priority based on its deadline, allowing for efficient retrieval of packages that need to be delivered soon. However, this data structure may require additional handling for updating and maintaining the priority of packages as their status changes.

# H1a. Data Structure Differences

Differences from the Chaining Hash Map:

Binary Search Tree (BST):

Storage: A BST stores data in a hierarchical structure, whereas a hash table uses an array of buckets.

Performance: BST operations have an average time complexity of O(log N), which may be slightly slower than the constant-time complexity of a hash table.

Handling Collisions: A BST does not inherently handle collisions like a hash table does with chaining.

Priority Queue (Heap):

Ordering: A priority queue orders elements based on their priority value, while a hash table does not inherently impose an order on its elements.

Priority Management: A priority queue requires careful management of priorities as packages' status changes, while a hash table is simpler in this regard.

Lookup Complexity: Priority queue operations have a time complexity of O(log N), which could be slower than a hash table's O(1) average case.

# I. Sources

Dorigo, M. (n.d.). *Ant colony optimization*. Scholarpedia.

http://www.scholarpedia.org/article/Ant\_colony\_optimization

GeeksforGeeks. (2023a, April 11). *Priority queue using binary heap*. GeeksforGeeks.

https://www.geeksforgeeks.org/priority-queue-using-binary-heap/

GeeksforGeeks. (2023b, May 8). *Binary search tree*. GeeksforGeeks.

https://www.geeksforgeeks.org/binary-search-tree-data-structure/

Section 6-4-12. (n.d.). https://web.mit.edu/urban\_or\_book/www/book/chapter6/6.4.12.html

Wikimedia Foundation. (2023, August 12). *Dijkstra’s algorithm*. Wikipedia.

https://en.wikipedia.org/wiki/Dijkstra%27s\_algorithm#:~:text=Dijkstra’s%20algorithm%20(%2F%CB%88da%C9%AA,and%20published%20three%20years%20later.

Lysecky, R., & Vahid, F. (2018, June). *C950: Data Structures and Algorithms II*. zyBooks.

Retrieved March 22, 2021, from <https://learn.zybooks.com/zybook/WGUC950AY20182019/>