WGUPS Package Delivery – Task 2

Rafael Estrella

Western Governors University

Course: Algorithms and Data Structures II (C950)

Instructor: Amy Antonucci, Ph.D.

F. Justifying the package delivery algorithm used in the solution.

1. Strengths of the Nearest Neighbor Algorithm

One strength of the algorithm is its simplicity, as it is easy to understand and implement. The algorithm is also very efficient, as its greedy nature allows it to find an optimal solution, even more so than other more complex algorithms. The algorithm can also be adaptable, for example, if the number of queued packages changes, the algorithm would still work and follow requirements, if the dispatcher manually assigns packages to trucks appropriately.

2. Algorithm Scenario Requirements

The algorithm must adapt to changes in the number of queued packages, as packages with delayed arrivals cannot be delivered until it arrives at the hub and is loaded onto a truck. The nearest neighbor algorithm satisfies this requirement, as package changes can be done in live time, and the algorithm will adjust to the new requirements. The dispatcher would be the one in charge of assigning packages to trucks, while trucks only focus on delivering packages assigned to them. It is up to the dispatcher to assign packages to trucks in a way that maximizes the algorithm's efficiency.

One way of doing so is by assigning packages that are close to each other to the same truck, this way each truck minimizes the distance it travels. Another way is by spreading out packages with close deadlines to different trucks with drivers, so that the trucks can deliver all packages on time. If one truck has too many packages with close deadlines, it may not be able to deliver all packages on time. An alternative but less impactful way of saving mileage is by assigning delayed arrival packages to trucks that would not be on the road at the time of the package's arrival. This way, the truck would not have to travel back to the hub to pick up the package, and then travel back to the package's destination. This approach requires that one of the trucks comes back on time for the next truck with delayed packages to finish all deliveries on time. By following these constraints, the algorithm has a much higher chance of delivering all packages on time, while minimizing the total mileage traveled by all trucks.

3. Alternative Algorithms that would meet all requirements

Genetic Algorithm: this algorithm, at every step, generates a new population of solutions from the current population. The new population is generated by applying crossover and mutation operators to the current population. The fitness of everyone in the population is evaluated and the fittest individuals are selected for the next generation.

Simulated Annealing: this algorithm is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities).

a. Comparing Algorithms with Nearest Neighbor

In a genetic implementation, rather than selecting the nearest neighbor and moving to that neighbor right away, the algorithm would generate multiple routes and then select the best route based on a fitness function, allowing other more efficient routes that the nearest neighbor algorithm would have otherwise overlooked. Although this may be more efficient for delivering all packages, it would be more complex to implement, and its computational running time would be longer.

With a simulated annealing approach, the algorithm would generate a random solution and then randomly change the solution. If the new solution is better than the previous solution, it would be accepted. If the new solution is worse than the previous solution, it would be accepted with a probability that decreases as the difference between the new and old solution increases. This algorithm would also be more complex to implement, and its running time would also be longer.

G. What I would do differently

Since the algorithm benefits the most when grouping packages closest to each other, what I would have done differently is implement a way to dynamically assign packages to trucks so that the nearest neighbor algorithm may take advantage of its efficiency. By doing so, trucks would not have to travel as much. This can be done by using a clustering algorithm like K-Medoids, which would group locations together into "clusters" (H et al., 2022) based on their proximity to each other. The dispatcher can then assign each truck a group of locations to deliver to, to achieve the most optimal nearest neighbor time. Mutations can later be made to each group by swapping locations between each truck to meet the package requirements. The computational complexity of this solution would be bigger but would only require a one-time execution per delivery day, plus any outlining cases. This would greatly reduce the total mileage traveled by all trucks overall.

H. Hash Table Scenario Requirements

The hash table offers both an insert and get method, which allows the nearest neighbor algorithm to insert update pr retrieve package information. With its lookup time of O (1), the hash table is essential for the nearest neighbor algorithm to access and update package information in real time. The hash table accounts for the required average list size of 40 packages as mentioned in the assumption. The initial size of 57 hash table slots, accounts for the margin of error for any additional packages that may be added in the future.

1. Alternatives to Hash Tables

Dictionaries: Dictionaries are like hash tables in that they have key-value pairs, so they could also be used to storing and accessing package information.

Arrays: Although not the best choice, Arrays can also be used to store and retrieve package information.

a. Data Structure Comparisons

Unlike hash tables, which store a value of only one data type, like an int, string or package object, a Dictionary can store any type of data in their key value pairs. A dictionary may even be key value inside another dictionary. In terms of efficiency, they also have a lookup time of O (1), which is the same as hash tables. However, dictionaries are not as flexible as hash tables, as they cannot resize on its own.

On the other hand, arrays are not as efficient as hash tables or dictionaries, as they have a lookup time of O(n). They also cannot resize on their own, so the size of the array must be known beforehand. However, arrays are more flexible than hash tables and dictionaries, as they can store any type of data, and can be multidimensional.

References

H, M., Hegde, A., Peter, A., Shetty, N. N., & G, S. (2022). Solving MTSP (Geocode) for E-commerce Delivery using K-Medoids and Dynamic Programming. International Journal of Innovative Science and Research Technology, 7(6), 1108-1111. Zenodo. https://doi.org/10.5281/zenodo.6833647