Project Report

Drone Delivery Systems Using Graph Theory

Group – 8
Siva Rama Krishna Kakarla – 11659343
Naga Vara Pradeep Yendluri – 11646461
Venkata Varuna Sri Budidi – 11640985
Sai teja Anguilla – 11545193
Anooshma Vainala – 11551414

***Abstract*—The objective of this project is to implement the last mile delivery systems using drones with the help of graph theory concepts. The system consists of multiple warehouses, charging stations, and customers located in a geographical region, and a fleet of drones that can transport packages from warehouses to customers. Graph Theory concepts are used to model the system as a graph, where nodes represent warehouses, charging stations, and customers and edges represent the flight paths of drones between them. We are assuming that there are charging stations to decrease the number of drones required to complete multiple packages delivery without reaching the warehouse for charging. The system employs an algorithm – Open TSP – Dynamic Programming algorithm to compute the shortest and fastest routes.**

Keywords—drones, last mile delivery system, graph theory, Open TSP

I. INTRODUCTION

The last mile delivery of goods and packages is a critical component of many online shopping sites. Traditional delivery methods can be inefficient and expensive, often resulting in delays and increasing costs. In recent years, drones are increasingly used for last mile deliveries to decrease the fuel costs and combat the worker shortage.

Drones became popular as a last mile delivery option due to their low operating costs and speed of delivery. However, drones efficient and effective usage requires intelligent systems to be used for optimizing delivery routes, manage multiple packages and handle battery life and charging issues. The system also considers drone payload capacity and battery life to determine the optimal delivery sequence. The inclusion of charging points and the ability for a single drone to carry multiple packages also enhances the efficiency and effectiveness of the delivery system.

II. GRAPH THEORY CONCEPTS

Here, we are going to briefly discuss the concepts of Graph Theory that are required to solve the delivery system.

Weighted Graphs: The delivery network is represented as a weighted graph, where nodes represent delivery locations, and edges represent the distance between locations.

Shortest Path Algorithms: To determine the most efficient delivery route, shortest path algorithms such as Dijkstra's algorithm or Open Traveling Salesman Problem algorithm can be used. These algorithms compute the shortest path between two nodes in a weighted graph, considering the weight of each edge.

Directed Graphs: A directed graph can be used to represent the delivery sequence, where nodes represent delivery locations, and edges represent the order in which the packages need to be delivered.

Capacitated Vehicles Routing Problem (CVRP): CVRP is a combinatorial optimization problem that involves finding the most efficient way to deliver packages using a limited number of vehicles with limited carrying capacity. This concept is essential for determining the optimal delivery sequence for a drone with limited payload capacity.

Charging Points as Nodes: The use of charging points in the delivery network can be represented as additional nodes in the graph, where drones can recharge their batteries. The location and capacity of these nodes can be optimized to minimize the number of charging stops required.

Minimal Hamiltonian Cycle: The Minimum Hamiltonian Cycle (MHC) problem is a well-known combinatorial optimization problem in graph theory. A Hamiltonian path or traceable path is a path that visits each vertex of the graph exactly once. MHC asks for the shortest possible cycle that visits each vertex exactly only once.

By applying these graph theory concepts, the proposed last mile drone delivery system can efficiently and effectively optimize delivery routes, manage multiple packages, and handle battery life and charging issues.

III. FLOW IN DELIVERY SYSTEM SCENARIOS

There are several scenarios that the last mile delivery system can be used can include for. First, we take package weights into account and try to fit more packages in a single route. We sort the packages by weight first and then take the first ‘n’ packages that fill the maximum load of the drone. Here are some scenarios:

A. Single Package Delivery

Suppose there is a package that needs to be delivered from warehouse to a customer. The delivery network is represented as a weighted graph, where nodes represent delivery locations, and edges represent the distance between locations.

The algorithm starts by identifying the shortest path between warehouse and the customer using geopy package to calculate the distance between the two locations. This step computes the shortest distance between two nodes.

The algorithm then checks the payload capacity and battery life of the drone to determine if the package can be delivered in one go. If the payload capacity or battery life is insufficient, the algorithm will add charging points to the path, where the drone can recharge its battery. The location and

capacity of these nodes can be optimized to minimize the number of charging stops required. Once the optimal delivery sequence is determined, the drone takes off from warehouse and follows the directed graph to deliver the package to customer.

Overall, this scenario shows how the implemented last mile drone delivery system can efficiently and effectively deliver a single package and optimize delivery route by adding charging points and managing battery charge.

B. Multiple Package Delivery

Suppose there are multiple packages that need to be delivered from point A, with delivery locations at points B, C, D, E, F, and G. The delivery network is represented as a weighted graph, where nodes represent delivery locations, and edges represent the distance between locations. To have a safe distance we assume that drone has a 5 miles power backup. This helps in mitigating low battery situations and reaching charging points if a charging point lies within 5 miles radius of the delivery location.

The algorithm starts by identifying the route using the Open Traveling Salesman Problem (asymmetric TSP) using Dynamic Programming method. It computes the shortest path such that it visits each node only once in a weighted graph.

Once the shortest path is identified, the algorithm determines the optimal delivery sequence for the packages. This is done by creating a directed graph, where nodes represent delivery locations, and edges represent the order in which the packages need to be delivered. The algorithm considers the payload capacity and battery life of the drone to determine the optimal delivery sequence for the packages.

We take the sequence and the distance that can be traveled by drone, in each iteration we decrease the travelable distance of the drone by the charge consumed for the distance already traveled. If the upcoming delivery distance is less than the distance travelable by drone, then we can continue to the next iteration, else we add a charging point to the sequence between the current and next delivery location. The charging point is located such that the drone can travel to it with its current charge and the location is also nearest to the upcoming location. This minimizes the distance between delivery locations.

To optimize the delivery process, a key consideration is finding the nearest warehouse after completing each delivery. This step ensures that subsequent deliveries can be efficiently initiated, minimizing the overall delivery time, and maximizing customer satisfaction. We search for the nearest warehouse in the same way as for delivery location, and if we are not able to reach it in a single go, we add charging points to reach.

Once the overall sequence of locations along with charging points is determined, we can start the delivery.

ALGORITHM

Multiple Package Delivery:

1. Use the Open Traveling Salesman Problem (asymmetric TSP) -Dynamic Programming algorithm to compute the shortest path between all pairs of nodes in the weighted graph.
2. Select packages from the available list of packages, such that when their combined(sum) total weights should be within the load capacity of the drone. Only locations of the selected packages are passed for the path calculation.
3. From the TSP solution, determine the optimal delivery sequence for the packages. Create a directed graph, where nodes represent delivery locations and edges represent the order in which the packages need to be delivered.

   $L(S,j) = \min L(S-\{j\}, i)+d(i,j)$ where $i \in S, i \neq j$,

   $S \in \{C1, C2, C3, \ldots, Cn\}$ and $L(S, j)$ represents the shortest distance required to traverse all the nodes in the set S exactly once, starting from node 1 and concluding at node j.
4. For each delivery location in the sequence, if the drone's travelable distance is less than the upcoming delivery location distance, then compute the shortest distance to the nearest charging point such that drone is able to reach with its current charge and distance from chosen charging point location to the next delivery location is minimal.

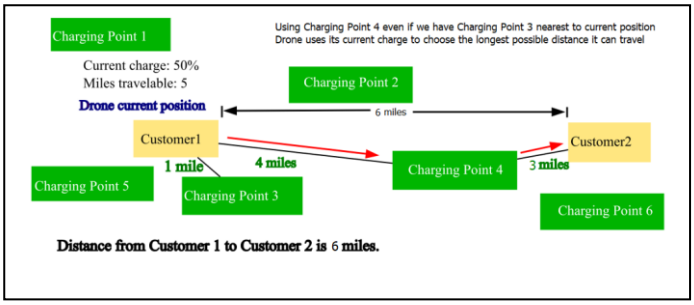

Fig 1: Finding the shortest path between delivery points via charging points.

5. Initialize the drone's travelable distance to its maximum range and start at the first delivery location in the sequence.
6. In each iteration, subtract the distance already travelled from the drone's travelable distance, and check if the upcoming delivery location is within the remaining distance.
7. If the upcoming delivery location is within the remaining distance, continue to the next iteration.
8. If the upcoming delivery location is beyond the remaining distance, add a charging point to the sequence between the current and the next delivery location such that the drone can utilize its current charge to the maximum to reach the charging point and the distance from the charging point to next delivery location is minimal.
9. Update the drone's travelable distance to its maximum range and continue with the next iteration.
10. Repeat steps 6-9 until all delivery locations have been visited.
11. Once all packages have been delivered, return to the nearest warehouse.
12. Find the nearest warehouse from the last delivery location. Use charging points to reach the nearest warehouse if the current charge is not sufficient.

IMPLEMENTATION

Input: Collect the necessary input data for the drone's operation. This includes the maximum load capacity of the drone, information about available packages (weights and delivery locations), the locations of warehouses, the locations of charging points, and the drone's battery capacity and current charge level.

Determine Package Selection: Calculate the total weight of all available packages and sort them based on their weights in ascending order. Iterate over package weights from the above sorted list and add them to total weight and compare the total weight of the added packages with the drone's maximum load capacity to determine which packages can be accommodated. Select the packages that can fit within the drone's load capacity for the current delivery cycle.

Determine Route: Identify the delivery locations corresponding to the selected packages and warehouse as the starting point. We use routing algorithm - Open TSP using dynamic programming, to find the optimal route that visits all the delivery locations and returns to the nearest warehouse.

Battery Management: Estimate the battery usage for the determined route by considering the distance and energy requirements of the drone. Check if the drone's battery charge level is sufficient to complete the route without recharging. If the battery charge is insufficient, identify the nearest charging point along the route. Adjust the route to include a charging stop at the identified charging point.

Delivery Execution: Start executing the route, following the final path. Deliver the packages to their respective locations. If a charging stop is included, navigate to the charging point, and initiate the charging process. Monitor the battery charge level throughout the delivery process.

EVALUATION

We evaluated the performance of the Open TSP dynamic algorithm and the Greedy algorithm for calculating the shortest path in the TSP problem.

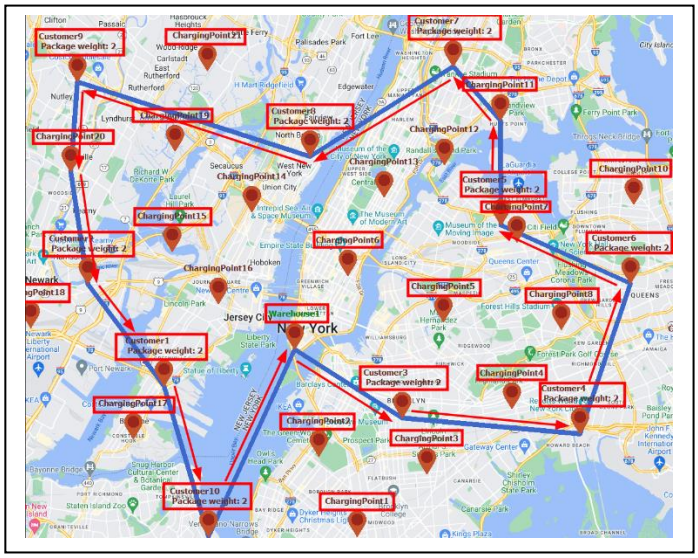

Fig 2: Evaluating Open TSP dynamic & Greedy algorithms.

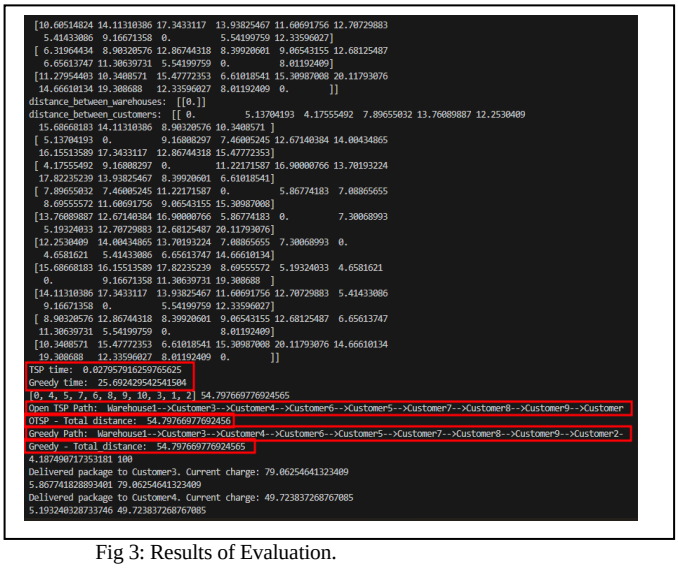

Fig 3: Results of Evaluation.

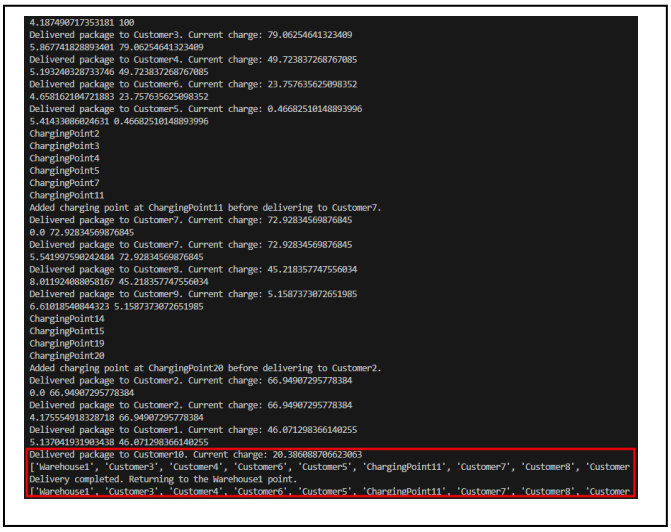

Fig 4: Results(continued) of Evaluation

To optimize the delivery process, a key consideration is finding the nearest warehouse after completing each delivery. This step ensures that subsequent deliveries can be efficiently initiated, minimizing the overall delivery time, and maximizing customer satisfaction. We search for the nearest warehouse in the same way as for delivery location, and if we are not able to reach it in a single go, we add charging points to reach.

Based on the obtained results above, The Open TSP algorithm took significantly less time **(0.03 seconds)** compared to the Greedy algorithm **(25.69 seconds).** This suggests that the Open TSP algorithm is much faster and more efficient than the Greedy algorithm for solving TSP problems of this size.

In terms of the results obtained from the two algorithms, both the Open TSP and Greedy algorithms produced the same shortest path with a total distance of 54.80 units.

Both the Open TSP algorithm and the Greedy algorithm were able to identify the shortest path for the TSP problem with a total distance of 54.80 units.

However, it's important to note that the Greedy algorithm is a brute force approach, which may not always guarantee the optimal solution for larger TSP problems.

In contrast, the Open TSP algorithm relies on more advanced optimization techniques, making it a more efficient and reliable option, particularly for larger TSP problems.

Overall, the results suggest that the Open TSP algorithm is a superior approach for solving TSP problems compared to the Greedy algorithm.

RESULTS

Scenario 1: Single Package Delivery

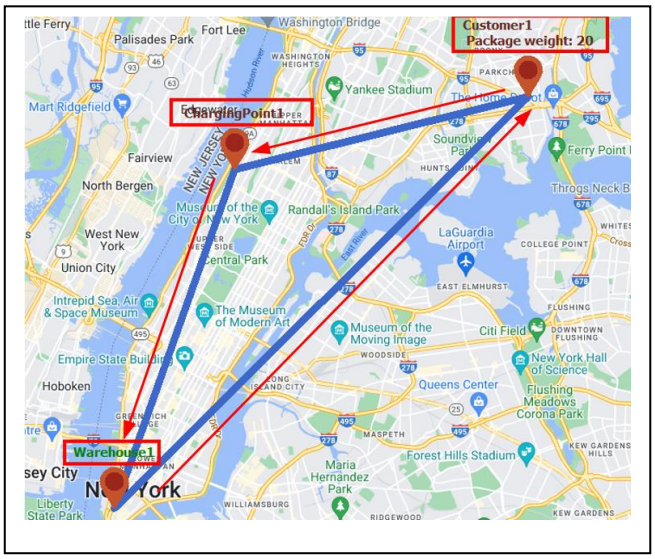

Fig 5: Optimal path for Scenario 1

Fig 6: Result of Scenario 1

The above result shows the scenario of a single package delivery.

We have added a warehouse, a customer whose package is to be delivered and charging points. The distance between the warehouse and the customer is 12.12 miles which is less than the drone's maximum travelable distance.

So, the drone directly delivers the package to the customer. Now the drone must return to the warehouse, the distance between the customer and the warehouse is 12.12 but as per our assumption the drones can travel to a maximum distance of 20 miles.

As the drone already travelled 12.12 miles to deliver the package to the customer, it has left with the charge to travel for only 7.88 miles. As it cannot reach the warehouse directly, it will add a charging point to the path after delivering the package to the customer and when the charge is full, it will return to the warehouse.

Scenario 1: Extension

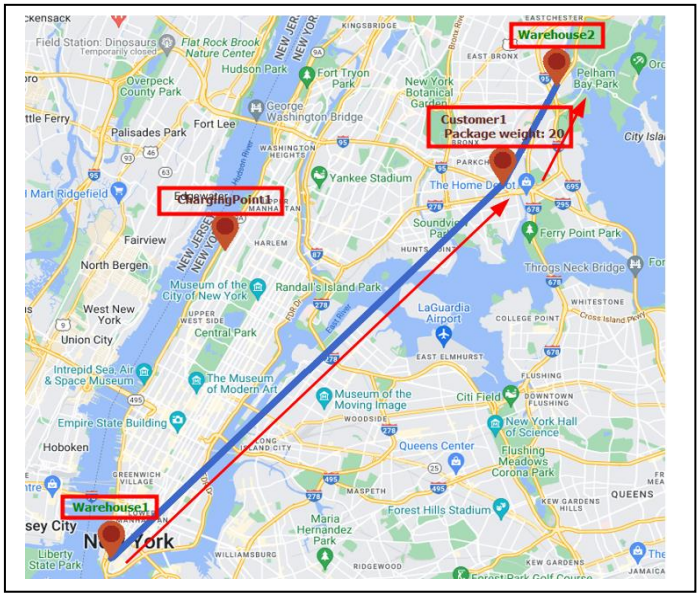

Fig 7: Optimal path for Scenario 1 - Extension

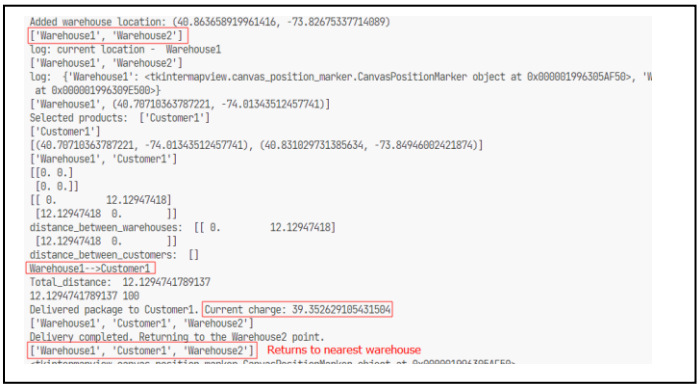

Fig 8: Results of Scenario 1 - Extension

To test the drone, to see if it will reach the nearest warehouse or not in scenario 1, we have added a new warehouse nearer to Customer1. We ran the application, and it calculated the path from customer to the new warehouse which we added.

This implies that the path is being calculated to return the drone to the nearest available warehouse from the last package delivery location.

Scenario 2: Multiple Package Delivery

To test the application to calculate the shortest path to deliver multiple packages on a single flight, we have added **three** customer locations along with the warehouse and charging points.

The algorithm picked the customer location which is near to the warehouse first among all the delivery locations. As we can see from the results, the distance between the warehouse and the Customer3 is 6.31 miles. The drone will deliver the package to Customer3 as the distance is less than the maximum distance travelable by the drone.

To test the application to calculate the shortest path to deliver multiple packages on a single flight, we have added

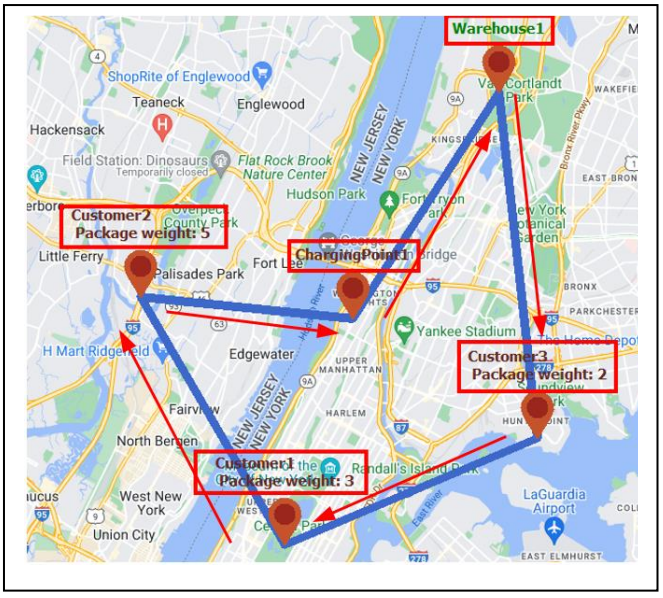

Fig 9: Optimal path for Scenario 2

three customer locations along with the warehouse and charging points.

The algorithm picked the customer location which is near to the warehouse first among all the delivery locations. As we can see from the results, the distance between the warehouse and the Customer3 is 6.31 miles. The drone will deliver the package to Customer3 as the distance is less than the maximum distance travelable by the drone.

Now, after delivering the package to Customer3, the drone can travel 13.69 miles. The distance between location of Customer3 and location of Customer1 is 4.99 miles. So, the drone will continue its flight to deliver the package to Customer1.

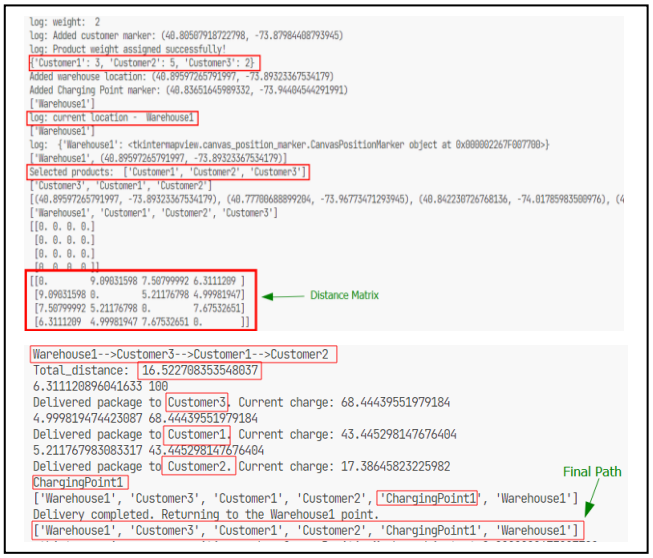

Fig 10: Result for Scenario 2

After delivering the package to customer 1, the drone can travel for 8.7 miles. The distance between Customer1 and Customer2 is 5.21 miles. As the drone can travel for 8.7 miles, it will continue to deliver the package to Customer2.

After delivering the package to all the customers, the drone is left with a charge that is sufficient to only 3.89 miles. As the drone cannot reach the warehouse back with the available charge, it will search for a charging point between the last delivery location and warehouse in such a way that the distance that needs to be travelled is minimal.

Charging point 1 is added to the path and once the drone the fully charged it will return to the warehouse.

Scenario 2: Extension

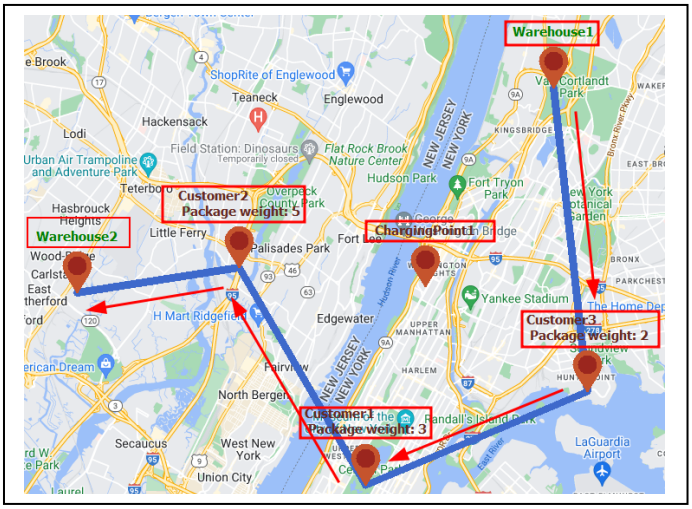

Fig 11: Optimal path for Scenario 2 - Extension

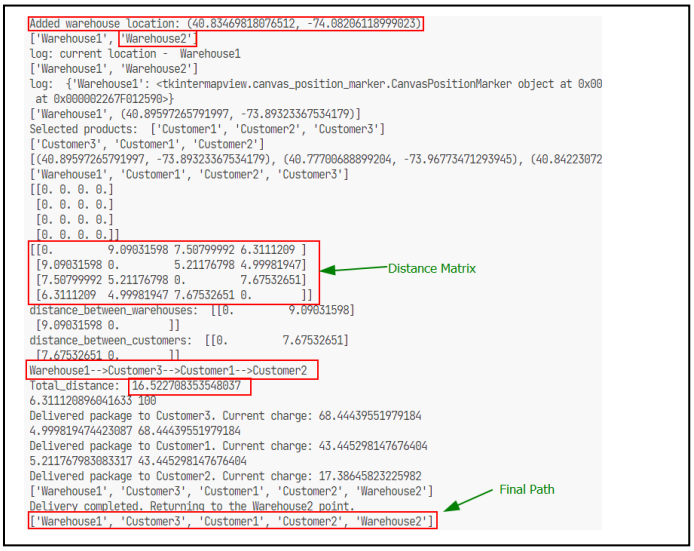

Fig 12: Result for Scenario 2 - Extension

To test the drone, to reach the nearest warehouse in scenario 2, we have added a new warehouse nearer to Customer2. We ran the application, and it calculated the path from customer to the new warehouse which we added.

This implies that the path is being calculated to return the drone to the nearest available warehouse from the last package delivery location.

RISKS

1. Inaccurate data: The Open TSP algorithm relies on accurate data to make decisions, so any inaccuracies or errors in the data could lead to incorrect decisions, such as selecting the wrong delivery route or destination.

2. Limited scope: Algorithms used in drone delivery systems may have limited scope or be unable to account for

unexpected or unusual situations, such as bad weather or restricted air spaces.

3. Training: The development and application of algorithms in drone delivery systems require specialized training and expertise, which could be a barrier to adoption or lead to a shortage of qualified personnel.

4. Operational limitations: The use of algorithms in drone delivery systems may be limited by factors such as battery life, range, or payload capacity, which could impact the system's overall effectiveness and efficiency.

5. Regulatory compliance: The use of algorithms in drone delivery systems could potentially violate regulations or laws, especially around airspace regulations. This could result in legal penalties or damage to the reputation of the company involved.

FUTURE SCOPE

1. Charging points can be used as small warehouses to pick packages if the drone has capacity.

2. Adding delivery windows selected by the customers to the system to calculate the efficient path.

3. We need to collect real-time data about weather, air traffic, and the location of other drones, and use this information to the delivery system. By doing so, drones can adjust their routes and speeds for optimal delivery times and safety.

4. Another way to improve the performance of the system is to divide the map into 4 segments around the warehouse and use different drones to deliver packages in these segments to reduce the delivery time and distance to be travelled.

5. Security and safety features can also be added in the future. For example, geofencing and collision avoidance systems can be implemented to ensure safe and secure drone operations.

6. Integrating the project with other technologies such as machine learning and IoT devices can improve the system's functionality. Machine learning can be used to predict delivery volumes and optimize drone routing, while IoT sensors can monitor drone performance and identify maintenance issues.

REFERENCES

[1] [Hamiltonian Path] https://en.wikipedia.org/wiki/Hamiltonian_path

[2] W.D.H. Pushpakumara 19001258 February 2022. [online]Available: https://www.researchgate.net/publication/358751793_Smart-Drone_Delivery_System

[3] Travelling salesman problem [online] Available: https://en.wikipedia.org/wiki/Travelling_salesman_problem

[4] https://www.tutorialspoint.com/design_and_analysis_of_algorithms/design_and_analysis_of_algorithms_travelling_salesman_problem.htm

[5] Used TSP dynamic algorithm package for finding route between selected packages and warehouse. (https://github.com/fillipe-gsm/python-tsp)

[6] Used customtkinter, TkinterMapView library for UI implementation. https://github.com/TomSchimansky/CustomTkinter, https://github.com/TomSchimansky/TkinterMapView

[7] To calculate distances between locations, we used geopy python package.(https://github.com/geopy/geopy)

[8] A Study on the Traveling Salesman Problem with a Drone - https://www.andrew.cmu.edu/user/vanhoeve/papers/tsp_drone_cpaior2019.pdf

[9] Delivery Optimization of a Logistics Network based on Drones - https://ysjournal.com/computer-science/delivery-optimization-of-a-logistics-network-based-on-drones/