Finding Optimal Warehouse Location using Multiple Criteria Decision Analysis

Ankit Kumar
Master of Science, Data Analytics
National College of Ireland
Dublin, Ireland

Abstract* – *People have been transporting goods since the ancient times. In current times, due to globalization, it has become easier for people to order the best products from different countries through various e-commerce platforms and expect delivery at their doorsteps in the least time amount of time possible. Companies are spending a lot in last mile delivery and that's why a company named We-Doo is helping these big e commerce firms to deliver the products to the doorsteps by having a delivery location at hundreds of towns. This project will explore one such town and then have 5 warehouses at random and then decide which warehouse is the best for the company We-Doo that minimizes the travelling time, operations cost and finds an optimal solution to the TSP. In order to find the best warehouse location, Multiple Criteria Decision Analysis (MCDA) is used where weights are given to various parameters and accordingly the best warehouse is chosen.

Keywords **– Delivery at customer doorstep, TSP, Multiple Criteria Decision Analysis (MCDA)**

I. INTRODUCTION

Since ancient times, people have been transporting goods from one town to another. We have seen advancements in technology which led to globalization where countries are producing goods and selling to different other countries. With the growing challenge of delivering the goods, it becomes very important to deliver goods in right condition within a specific amount of time. Companies like Amazon are spending millions of dollars to do the last mile delivery. According to Amazon, last mile delivery is expensive, and 50 percent of the cost is the overall fulfilment cost [1]. The lowest cost to transport the product to a customer's location has a trade-off between the travel time, cost and finding the shortest path. According to eMarketer [2], there are benefits of effective last mile delivery namely.

- Faster delivery
- Increase in sales and revenue
- Increase in efficiency

One such problem is the travelling salesman problem which works on minimizing the cost of travelling salesman. The goal of a salesman is to lower the cost of both distance travelled and cost. Many companies are providing last mile deliveries like FedEx, USPS, UPS etc. For our analysis, we will look at one such startup which managed to raise substantial seed funding.

We-Doo company is working on the last mile delivery solution by establishing one delivery location per town to cater to all the customers in that town. As companies are spending a lot of money in tackling the TSP and Last Mile Delivery, We-Doo extends their helping hand by rolling out warehouses in hundreds of small towns by charging a small fee from the customers so that the package can be delivered to their doorstep in time. This paper discusses and analyses this company and finds a solution for them to cater to one such town.

In the late section, this paper will look at the literature review, Methodologies used, Results and interpretations and finally conclude with future work and references.

II. LITERATURE REVIEW

The key issue that this paper will address is to minimize the distance travelled by a delivery person to reduce the operations cost and maximize profits and at the same time help customers receive the products/ordered parcels on time. One way is to deliver the parcel at customer's doorstep and another method is to have a shared reception boxes where customers and pick their package up. In one such research paper [3] the authors talk about how by having shared reception boxes, the cost to the last mile delivery can be reduced to 55% to 66%. There are various search algorithms that can help reduce the time and complexity for the salesman.

One such algorithm is the A* Algorithm. A* is an algorithm that finds the minimum distance between two points A and B[4]. It maintains a tree of paths originating from the start node and it extends until it finds the end location. The function related to A* algorithm is below –

$$F(n) = g(n) + h(n)$$

In this, the n represents the next node, g(n) is the cost from the n and h(n) is the heuristic that estimates the shortest path.

In Modelling, simulation and Optimization course, Professor Christian Horn (2024) [5] discusses about the Greedy search algorithm and heuristic search algorithms which we have used in this research paper to arrive to a conclusion. As the name suggests, greedy algorithm is short sighted, and it makes series of decision which is best at the time without looking for the global best [5]. With greedy algorithm, comes the heuristic algorithm which is the basic rules of thumb. They look at the problem as a whole and tries to find the optimal solution which is approximate. In short, generally, all greedy algorithms are heuristics, but all heuristics are not greedy.

After gaining a better understanding of various techniques, let's jump into the methodology and understand how this program is finding optimized solution and give best warehouse location.

III. METHODOLOGY

In this section, we will look at the methodology of finding warehouse location that serves all its customers by spending less time each day and travels shortest path and return to the warehouse. This section is divided into two parts Optimization and Simulation. Let's dive into them now.

A. Optimization

In the python file, the optimization part is divided into 2 sections – Section 2 and 3. Section 2 encapsulates all the Functions related to optimization and it has a total of 16 sub sections that have functions in them. Section 3 then has a logical coding that finds the optimized warehouse for our town. Section 3 has 7 sub sections which logically helps the company We-Doo find an optimal warehouse location.

1) The city: Figure 1 Shows the map of the city. This city map is randomly generated by giving seed – 3061 (last 4 digits of student number – x23123061) and then stored in the pickle file. This process was done once in the beginning and now have been stored to – realdata.pickled file. There are a total of 100 nodes and 50 customers in that town.

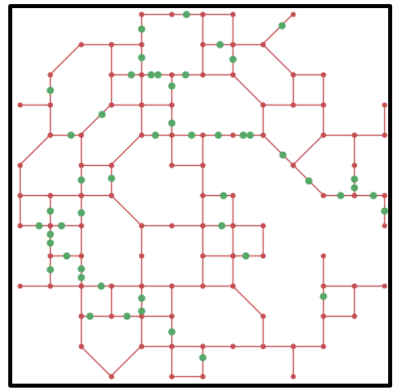

Fig. 1. Map of the Town with 100 Nodes and 50 Customers

2) The Warehouses: Figure 2 shows the map of the city with 5 different warehouse locations that was chosen at random on the first run and now has been hardcoded and locked-in for future simulation and optimization process.

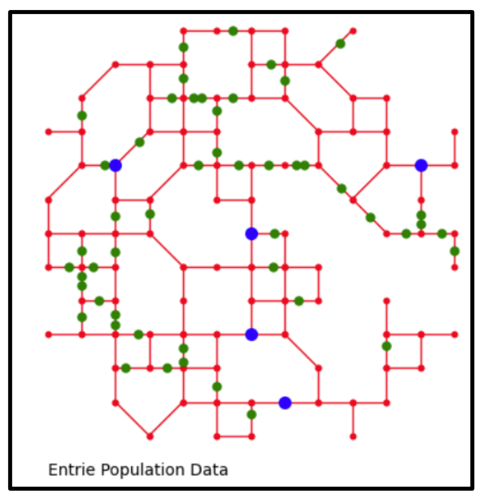

Fig. 2. Map of the town with 5 potential warehouses

The warehouse locations can be found at:

TABLE I. POTENTIAL WAREHOUSE LOCATIONS

Warehouse Number	**Warehouse Location**
Warehouse 1	4000,2320
Warehouse 2	4000,4000
Warehouse 3	1760,5120
Warehouse 4	4560,1200
Warehouse 5	6800,5120

[a] *These are the sample warehouse locations. Through the optimization process, we will choose one*

Now, in order to find the optimized warehouse locations, these functions were used as follows:

3) Greedy Algorithm: This algorithm makes a series of decisions by only looking at the next best step. It doesn't take into consideration the better path. For these cases to choose the optimal warehouse location, greedy search is not the best-case scenario. The code was optimized to include heuristic search as well.

4) Heuristic Search: Greedy search function was enhanced and rule 2 and 3 of heuristic search was added in the code. **Refer Python section 2.1.15.**

a) Heuristic Search Rule 2: The rule 2 of Heuristic Search works by reversing the order of two non-consecutive edges iteratively to check if the distance is reduced. This graphical explanation can be seen in the figure 3

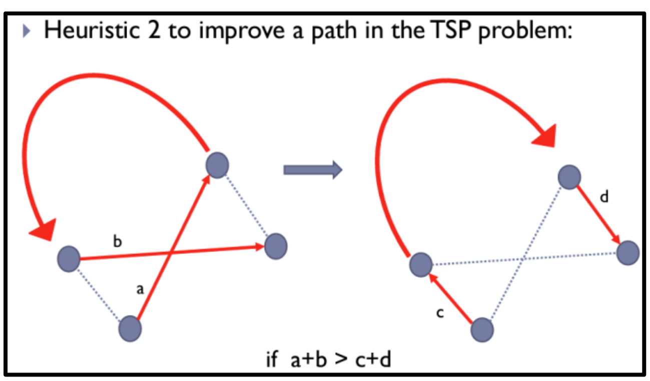

Fig. 3. Heuristic 2 to improve path in TSP Problem

b) Heuristic Search Rule 3: For heuristic Rule 3, refer Figure 4. There are four cities labelled a, b, c, and d. The heuristic considers edges connecting cities a and b, and c and d. If the total distance travelled by going from a to b and then c to d (a+b) is greater than the total distance travelled by going from a to c and then b to d (c+d), then reversing the order of the edges ab and cd would reduce the overall path length.

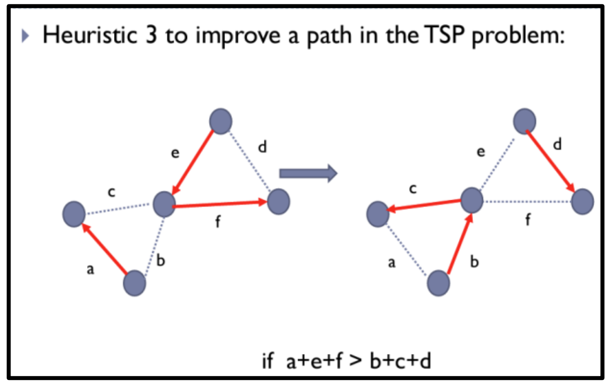

Fig. 4. Heuristic 4 to improve path in TSP Problem

5) Testing the 5 Warehouses for optimized path length: To find the optimized warehouse location that serves all the customers well and gives the least path length, firstly, it was tested on 10 customers, then 20 customers, then on all the customers (Total 50). Tables II, III and IV on page 6 compares the optimal, greedy and heuristic path lengths for all the 5 warehouses and customer locations. Optimal Path Length is just the sum of distances between 2 customer locations. This doesn't take into consideration that the delivery person must return to the warehouse. That's the reason the path length is least among the other two path lengths.

The highlighted cell in the tables below shows the warehouse locations where Heuristic Path Length is less than or equal to greedy path length. There are multiple cases where Greedy path is less than or equal to Heuristic path and that is because Greedy algorithm finds the next best route without contemplating the overall map. Greedy algorithms work on the limited or local information only. After we have implemented the rule 2 and 3 of the Heuristic search, it looks at the overall solution and even if the customers increase in future, the global minima will not change. In order to find the best warehouse, we then called the Monte Carlo Function.

The Monte Carlo function performed multiple iterations and found that the optimal warehouse is at the location – 4000,2320. The Montecarlo best warehouse serving all the customers is shown in the figure 5.

TABLE II - DELIVERY TO 10 CUSTOMERS

Warehouse Location	Optimal Path Length	Greedy Path Length	Heuristic Path Length
(4000, 2320)	27536	29315	28575
(4000, 4000)	27536	30446	32807
(1760, 5120)	28206	28206	29326
(4560, 1200)	28327	30106	29366
(6800, 5120)	28194	29326	28194

TABLE III – DELIVERY TO 20 CUSTOMERS

Warehouse Location	Optimal Path Length	Greedy Path Length	Heuristic Path Length
(4000, 2320)	36219	43342	45447
(4000, 4000)	35489	41006	40636
(1760, 5120)	35489	40626	40887
(4560, 1200)	35489	36563	42512
(6800, 5120)	36147	41006	40174

TABLE IV – DELIVERY TO 20 CUSTOMERS

Warehouse Location	Optimal Path Length	Greedy Path Length	Heuristic Path Length
(4000, 2320)	46847	52555	55713
(4000, 4000)	45727	48822	48822
(1760, 5120)	45727	54830	54428
(4560, 1200)	45727	53360	57572
(6800, 5120)	45727	54833	54833

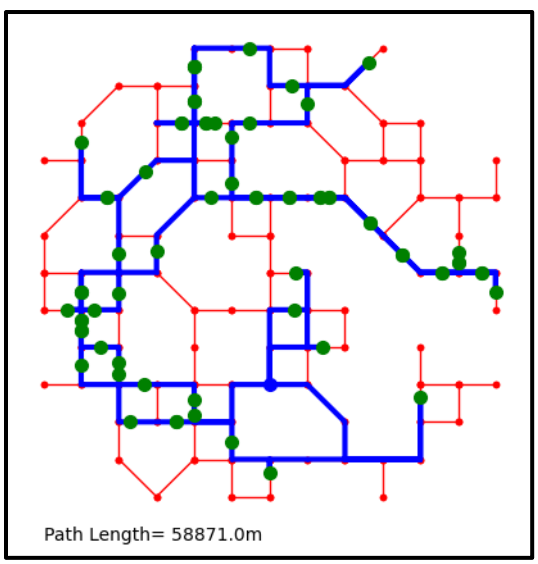

Fig. 5. Optimized path length according to Monte Carlo at – Warehouse (4000, 2320)

B. Simulation

After finding the optimized warehouse location at 4000, 2320, now we are to simulate all the warehouses to prove that our optimized solution is the best solution. In the python file, the simulation runs from section 4 to section 6.

Section 4 has all the functions and classes needed for simulating the routine for multiple delivery locations to all the customers over a period of n number of days. Section 4 also contains the constraints or parameters that is given in the problem which are also stated below:

TABLE V. CONSTRAINTS

CONSTRAINTS	VALUE	UNIT
Average Speed	15/3.6	Meter/ Second
Prep Time per parcel	50	Seconds
Return time per parcel	30	Seconds
Average time to answer door	40	Seconds
Wait time if customer doesn't answer door	60	Seconds
Operation cost per meter	0.00008	Euros/meter
Driver Payment Per Hr	30	Euros/Hr
Minimum driver pay	60	Euros
Day end procedure	10	Minutes

Once we have defined all our constraints, now is the time for simulation and taking all these constraints into consideration to find the best warehouse location that minimizes the cost and increases the productivity. To test the simulation process, 1 test scenario for 10 days with 0.15 probability per customer of receiving a parcel was conducted and second simulation was run for 30 days for 50 customers with probability of receiving a parcel at 0.15. Table VI on page 9 depicts the simulation for 30 days and 50 customers.

IV. REUSLTS AND INFERENCE

In the Python code, section 7 shows statistical measures to finalize the warehouse. Section 7.1 talks about the Multiple Criteria Decision Analysis(MCDA) ranking criteria and section 7.2 plots the same. This gives a good picture of which is the best warehouse for us to consider.

After successfully running the simulation for 30 days for 50 customers, it is now important to find the best warehouse which is not just the local minima but the global minima. In other words, we want to minimize Average Working time, Mean Distance, stats working time, operational costs and daily driver payment. There are various methods to find the most optimal warehouse. Some statistical methods include – t-tests, ANOVA, Regression, Correlation Analysis etc. but for this specific travelling salesman problem one such research paper have considered Multiple Criteria Decision Analysis (MCDA). This idea was taken from [5] where the authors talk about how they conducted MCDA to rank different features like Safety, economy, time for delivery etc for bicycles and autonomous robots to find the best solution for the last mile delivery. For this project, MCDA will be used to rank the parameters to find the best warehouse location which will serve as the optimal solution. Table VI Below shows the ranking criteria that was used-

TABLE VII. MCDA WEIGHTS ALLOCATION

Criteria	Rank	Weight
Operational Cost	1	6
Minimum Distance	2	5
Daily Driver Payment	3	4
Standard Deviation	4	3
Mean Distance	5	2
Average Working Time	6	1

The above table indicates that Operational Cost is the most important criteria for us and the least important is the Average Working time and accordingly the weights are assigned. After tunning the MCDA code, fig. 6 shows the results for the best warehouse for our town:

	Warehouse_Coordinates	Composite_Score
2	(1760, 5120)	5130.942179
0	(4000, 2320)	5238.178607
1	(4000, 4000)	5489.688090
3	(4560, 1200)	5995.169733
4	(6800, 5120)	6173.815108

Fig. 6. MCDA Composite Scores of Warehouse

TABLE VI. SIMULATION RESULT FOR 30 DAYS FOR 50 CUSTOMERS WITH PROBABILITY – 0.15 OF RECEIVING PARCELS

Warehouse	**Average Working Time (Avg per day)**	**Mean Distance (Avg per day)**	**Median Distance (Avg per day)**	**Minimum Distance (Avg per day)**	**Std Dev Distance (Avg per day)**	**Maximum Distance (Avg per day)**	**Stats Working Time (Avg per day)**	**Avg Operational Cost (Per Day)**	**Driver Daily Payment (Avg per day)**
UNIT ->	Minutes	Meters	Meters	Meters	Meters	Meters	Minutes	Euros/Day	Euros/Day
4000,2320	102.96	20212.10	19653.0	10327	5475.9	32867	112.9	1.61	60
4000,4000	102.73	20091.16	19431.5	9930	5504.5	32137	112.7	1.60	60
1760,5120	102.30	20027.50	19633.0	7126	6062.5	33257	112.3	1.60	60
4560,1200	114.23	23048.20	22794.5	14478	4820.1	32137	124.2	1.84	62
6800,5120	111.20	22247.20	21148.5	13471	4512.2	32795	121.2	1.77	60

The results show that warehouse 1760,5120 is the best warehouse in our case to consider with the least composite score. To minimize our variables, the weights were assigned in negative. If we want to maximize something, then we would provide a positive weight. Below is the graph that plots the composite scores side by side.

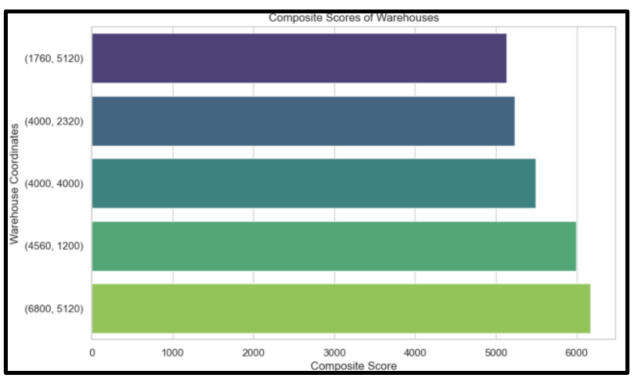

Fig. 7. Composite Score Bar Graph

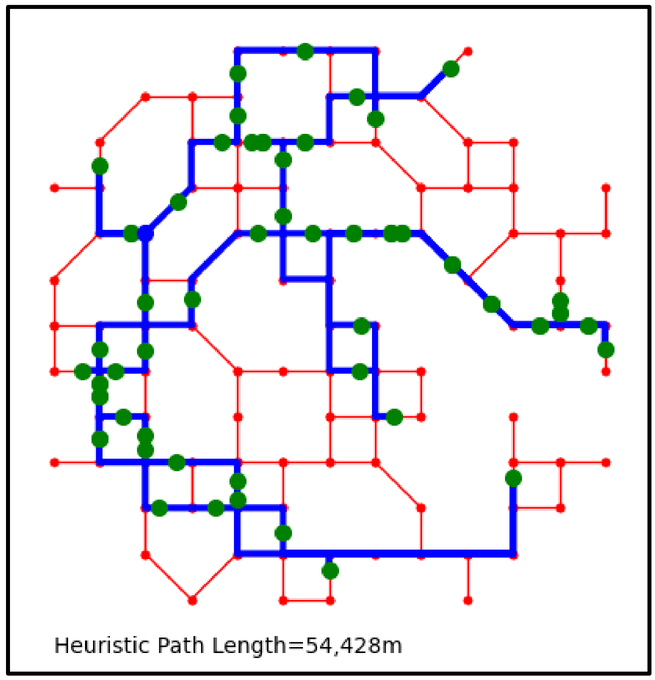

Fig. 8.Best Warehouse Heuristic Path length at (1760,5120)

V. FUTURE WORK

After doing MCDA to rank the parameters, we found out the best warehouse for this town to be at location 1760, 5120. This was calculated by taking into consideration only one probability for all the customers. If we think logically, every customer has a different probability of expecting a customer. If a customer who is located at a far end of the map orders more often and has a parcel expectancy of let's say around 40 percent, then the delivery person will clock.

more time and distance 40 percent of the times. Initial survey must be conducted with the customers and approximate parcel expectancy probability must be taken into consideration customer wise. This decision is based only on MCDA, but multiple other statistics models must be taken into consideration like f stats, ANOVA etc and should be considered as ensemble method. There are various other parameters that affect the distance and warehouse location like rain, snow etc. If an area is prone to flood and heavy rains, then the delivery might get delayed, and the electric scooter might not work. All these things must be considered for future work.

VI. REFERENCES

[1] Amazon Web Services (AWS), "AWS Last Mile Solution for Faster Delivery, Lower Costs, and a Better Customer Experience," AWS Blog, 2024.

[2] https://www.emarketer.com/insights/last-mile-delivery-shipping- explained/#:~:text= Last%2Dmile%20delivery%20is%20the,the%20last%2Dmile%20of%20delivery.

[3] Punakivi, Mikko, Hannu Yrjölä, and Jan Holmström. 2002. “Solving the Last Mile Issue: Reception box or Delivery Box?” International Journal of Physical Distribution Logistics Management 31 (6): 427–439. doi:10.1108/09590550210445362.

[4] Hart, Peter. “A* search algorithm.” Wikipedia, 1968

[5] C. Horn, “Greedy and Heuristic Algorithms” Dublin: 2024

[6] M. Kostrzewski, Y. Abdelatty, A. Eliwa, and M. Nader, “Analysis of Modern vs. Conventional Development Technologies in Transportation—The Case Study of a Last-Mile Delivery Process,”