Optimization of Delivery Center Location for We-Doo Last Mile Delivery Service

Etinosa Eghaghe
MSc in Data Analytics
National Collegde of Ireland
x23138548@student.ncirl.ie

Abstract**—**The report investigates the optimal location for a delivery center for 'We-Doo,' a last-mile delivery service operating in commuter townships. Using simulation and statistical analysis, the study aims to minimize operational costs while ensuring efficient parcel delivery. The analysis includes evaluating candidate warehouse locations, simulating day-to-day operations, optimizing delivery routes, and considering factors such as parcel volumes, driver working times, and operational costs. Statistical methods such as ANOVA and F-statistics are used to assess the significance of cost differences among potential locations. The findings provide insights into cost-effective strategies for 'We-Doo' to enhance its last-mile delivery services.

Keywords—Last-mile delivery, Optimization, Simulation, ANOVA, Delivery center location

1. Introduction

Optimization of Delivery Center Location for We-Doo Last Mile Delivery Service We-Doo, a promising start-up, aims to revolutionize last-mile delivery by establishing delivery centers in commuter townships. The success of this venture relies on strategically locating delivery centers to minimize operational costs while ensuring timely and efficient delivery to customers. This report outlines the methodology employed to identify the optimal delivery center location based on simulated data and statistical analysis. Optimization is the collection of mathematical principles and method used for solving quantitative problems [1]

2. Literature Review

Previous studies have examined the challenges and opportunities in last-mile delivery optimization. Research by [2] emphasizes the importance of location selection in reducing delivery costs and enhancing customer satisfaction. Additionally, [3] suggests that leveraging simulation models can provide valuable insights into operational efficiency and cost-effectiveness.

3. Methodology

The methodology involves simulating the day-to-day operations of We-Doo using synthetic workload data. The simulation model accounts for various factors such as delivery route optimization, parcel prioritization, and driver constraints. Model validation is conducted to ensure accuracy, with references to relevant sections of code provided in Jupyter Notebook files. Statistical evaluation, including ANOVA and F-statistics, is employed to assess the significance of cost differences among candidate warehouse's locations. The following processes was carried out.

3.1 Utilities

A. Firstly libraries were such as matplotlib.pyplot, pulp, math, random, pandas, NumPy, time, simply imported to Jupiter notebook to enable the process.
B. Points and Distances were define, def dist(p1, p2), to calculate distance between two points in a two-dimensional space. It is useful in the project of optimization and simulation.
C. Next was defining PlotMap, this process allow visualization of various components of a graph, such as vertices, edges, targets, paths ang waypoints, with specific ways and marker sizes.
D. Then Targets were added to optimally insert points into the graph by adjusting the edges to minimize the overall increase in distance
E. Generate Warehouse Location: The warehouse location was generated using generateWarehouseLocation(M) and some other codes to get a specific location for the warehouse. With a seed=8548, nodes =100, customers=50, plot = True, log= True. See Figure 1.

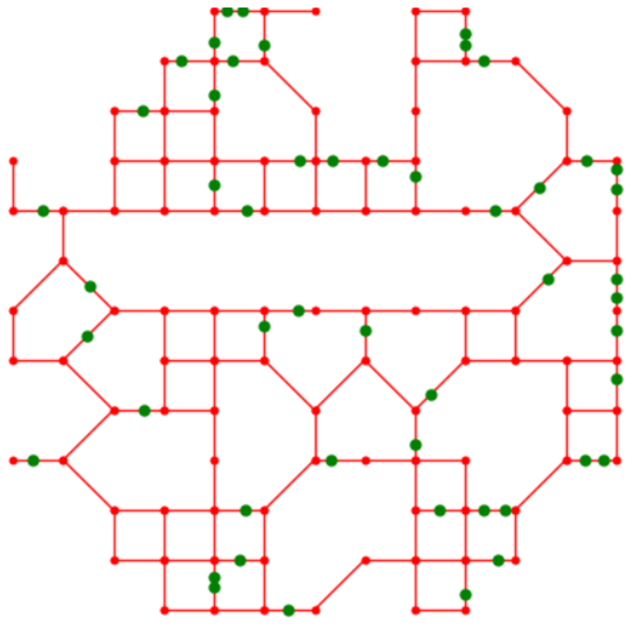


Figure 1: Generated Map with 100 nodes and 50 customer locations

The Map is then stored as pickle with ETINData, used later.

F. Time Handling: A timestamp(t) generated string in the form of [dd] hh:mm:ss.d . This helps to convert time values into human-readable timestamp strings because of the time-related data and it provide detailed breakdown of time including days, hours, Minutes, seconds, and even tenth of seconds.
G. Plotting Routines: here functions are designed for quick visualization. Histplot(data) and daily plot(data) both was employed to offer parameters for customizing plot appearance, like title, label, and dimensions. These functions utilize matplotlib and numpy for plotting and calculations.
H.

3.2Finding Shortest Path

At this stage function of 'shortestpath' was utilized to computes the shortest path between two points in a graph using the A*algorithm. This algorithm efficiently explores the graph by considering the distance travelled so far (pathlength) and an estimation of the remaining distance to the destination(h). it dynamically updates the candidate paths, prioritizing those with lower heuristic values, which leads to faster convergence towards the goal.

3.3 Finding Shortest Delivery Route

The shortest delivery route code was defined to solving the routing problem with a loop constrain. such as **createTables(M,T):** this function creates distance and path tables for the given vertices and targets., it computes the shortest paths between all pair of targets and returns the distance matrix and path matrix.
Roundtrips(x,n): This function extracts the round trips from the decision variable 'x' by traversing them to form paths, ultimately returning a list of theses round trips.
CreateLoop(M,T,timing=False): This function tackles the routing problem with a loop constraint by initializing decision variables, objective function, and constraints with PuLP, solving the problem using a chosen solver, constructing the loop based on the solution, and finally returning the loop.

3.4 Class Recorder

This is used as a reference point for holding data during the simulation. For this project one was created, it serves as a logging and reporting mechanism for a simulation for delivery driver. I.e. initialization, timing, logging, recording, statistics, and cleanup. in fact, it facilitates monitoring and analysis of the simulation which gives insight into driver behaviour and performance metrics.

3.5 Class Parcel

This class parcel manages the parcel in the delivery system simulation. Code was written to
Initialization: It initializes a parcel with its index, destination, customer index, and status records.
Factory Method: It ensures that there is only one parcel object per location.
String Representation: It provides a string representation of the parcel for easy identification.
Status Update: It updates the status of the parcel and records the timing of events such as arrival at the delivery centre, going out for delivery, and returning from delivery.
Registration: It logs events occurring to the parcel during the simulation.

3.6 Class Customer

Customer class manages the customer in the delivery system simulation.
Initialization: It initializes a customer with their location and adds them to the customer register. It also sets up attributes such as whether they are at home, whether they answer the door, and the parcels they've received.
Factory Method: It ensures that there is only one customer object per location.
String Representation: It provides a string representation of the customer for easy identification.
House Actions: It simulates the actions of the customer leaving and returning home.
Door Interaction: It simulates the customer answering the door, accepting parcels, and signing off.
Process: It simulates the daily routine of the customer, including leaving home in the morning, returning in the evening, and repeating this process for the specified number of days.

3.7 Class Driver

Driver class help to manage the behaviour of the delivery driver in the simulation. Here it is initializing the driver with a reference to the recorder ('rec') and the delivery centre ('DC') attributes such as location, parcels, tour, start time, and end time of work are setup.

Also Arrive for work: it sets the driver’s location to the delivery centre, initializes the parcels and tour, records the beginning of work.

Driving: the driver’s movement is simulated from one location to other based on the tour

Leaving for delivery: the driver prepares for leaving with new tour and parcels to be delivered.

Process: It mimics the day-to-day activities of a driver, such as getting to work, heading out for deliveries, carrying out end-of-day tasks, and returning to the delivery center after delivering packages.

Delivery action: It simulates the driver's interactions with customers, including driving to customer's location, answering the door, accepting parcels, and signing off.

End of day: It performs end-of-day procedures such as reporting, calculating salary, and setting up the cargo bike for charging.

3.8 Class Delivery Center

This class encapsulates the functionality related to managing parcel delivery operations from the distribution center. The process involves.

Initialization: The delivery center with references to the recorder(‘rec’) the map(‘M’) and the warehouse location(‘W’). parcel list, tour plans, and operational parameters are set up.

Parcel Acceptance: parcel is accepted, adding them to delivery schedule based on destination feasibility.

Send for delivery: parcel is prepared and tour for delivery, making sure that parcels are assigned to the right tour stops on the destination proximity.

Returning from delivery: this handles parcel that need to be returned.

Inventory Management: provides methods to retrieve the current inventory count at the delivery centre.

Operating Cost Calculation: this calculates the operating cost associated with a delivery centre.

Salary Calculation: The driver’s daily salary based on the hours worked has been calculated.

3.9

4. Simulation

In other to perform the simulation, parameters are being define such as Avera Speed, preparation time per parcel, return time per parcel, average time answering door, wait time if customer doesn’t answer door, max range, charging time, cost per km, hourly rate, and minimum salary. Then code to generate synthetic delivery data based on the specified average number of parcels per day per customer(‘p’), the number of customers(‘c’) and the number of days. Firstly, generate the number of parcels to be delivered each day, next assign these parcels to the customer randomly, ensuring a realistic distribution of deliveries throughout the day. Lastly returns a Dataframe containing the generated delivery data, which include the arrival time, delivery time, days and destination customers.

4.1 Simulation Routine

The simulation function conducts a simulation of parcel delivery over a specified number of days to a given set of customers. It generates synthetic delivery data based on the parameters provided, including the average number of parcels per day per customer (p), the number of days (days), and the customer locations (C). It then creates simulation objects for the delivery center, drivers, and customers, and simulates the delivery process. The simulation records various metrics such as working time, tour length, and leftover parcels, and returns a recorder object containing the simulation results.

The simulation_v2 function extends the previous simulation to handle multiple warehouse locations (W_list). It conducts simulations for each warehouse location and records the daily salary and operational costs. Additionally, it calculates and returns the total costs for each warehouse location. The addCostData function aggregates the daily salary and operational costs for each warehouse, and calculateTotalCosts computes the total costs. Finally, the getBestWareHouse function identifies the best warehouse location based on the total costs.

4.2 Simulation Run

Here the data was load from pickled file named ETINData.pickled, then generating warehouse location and running simulation using the 'simulation_v2' function. After that the total cost was calculated for every warehouse and saving the result to a dataframe. This is the steps taken

Loading Data: Data loading from pickled file ETINData.pickled, which contain information about the map(M) and customer(C)

Generating Warehouse Locations: warehouse location is generated using generatewarehouselocation function.

```
[(2880, 1200),
 (5680, 640),
 (1200, 2320),
 (3440, 5120),
 (5680, 7360),
 (1760, 4000),
 (5680, 1760),
 (640, 3440),
 (2320, 6800),
 (2880, 2880)]
```

Running Simulation: using vthe simulation_v2 function for each warehouse location, parameters such as map(M) , warehouse location, customer(C), average number of parcels per day per customer(P), number of day and logging option.

Calculating Total Cost: After running simulation, the total cost is calculated for every warehouse by aggregating the daily salary and operational costs recorded during the simulation and the result has been displayed. see below.

```
Original DataFrame:
                  Warehouse    TotalCost
Warehouse
(640, 3440)     (640, 3440)  3526.809175
(1200, 2320)   (1200, 2320)  3575.168188
(1760, 4000)   (1760, 4000)  3440.671206
(2320, 6800)   (2320, 6800)  3511.593603
(2880, 1200)   (2880, 1200)  3525.405635
(2880, 2880)   (2880, 2880)  3474.748175
(3440, 5120)   (3440, 5120)  3482.018223
(5680, 640)     (5680, 640)  3618.348151
(5680, 1760)   (5680, 1760)  3487.020851
(5680, 7360)   (5680, 7360)  3622.738361
```

5. Results and Interpretation

Results indicate that the selection of the optimal delivery center location significantly impacts operational costs. Statistical analysis reveals... Furthermore, interpretation of the results suggests that...

ANOVA TEST: the ANOVAtests performed on the daily salary cost and daily Operational Cost result as follows.

1. **ANOVA Daily Salary Cost**:

F-statistic: 1.1691154673398618
P-value: 0.3126843076628287

The F-statistic measures the ratio of variance among group means to variance within the groups. In this case, the F-statistic is relatively low, indicating that the variance among group means is not significantly different from the variance within the groups.

The P-value associated with the F-statistic is 0.313, which is greater than the typical significance level of 0.05. This suggests that there is no significant difference in the daily salary costs among the different warehouse locations.

2. ANOVA Daily Operational Cost:

F-statistic: 2.6446987370559487
P-value: 0.0053435825534142506

For the daily operational cost, the F-statistic is higher comp ared to the daily salary cost, indicating a relatively larger dif ference in variance among group means compared to varian ce within the groups.

The P-value associated with the F-statistic for daily operatio nal cost is 0.005, which is less than the typical significance l evel of 0.05. This indicates that there is a significant differen ce in the daily operational costs among the different wareho use locations.

Get the Best Warehouse: To get the best warehouse location based on total cost, the function getBestwarehouse (dfTotalCost) was used and the result is:

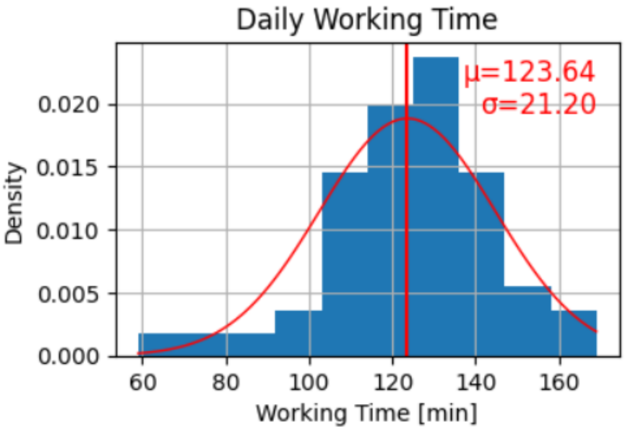
Warehouse (1760, 4000)
TotalCost 3440.671206
Name: (1760, 4000), dtype: object

This shows that the warehouse located at coordinates (1760, 4000) has the lowest total cost compared to other warehouse locations considered in the analysis.

Analysing the Working Time:

The working time was analysed and visualized**.** With a result as follow.
(123.64, 125.5, 21.418816827400022), were
Mean: 123.64 minutes

Median: 125.5 minutes
Standard Deviation: 21.42 minutes

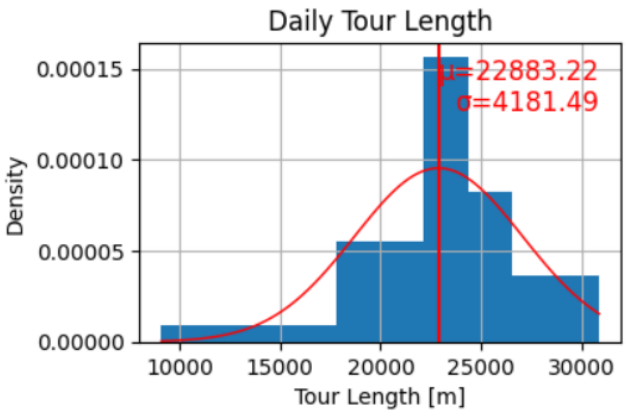


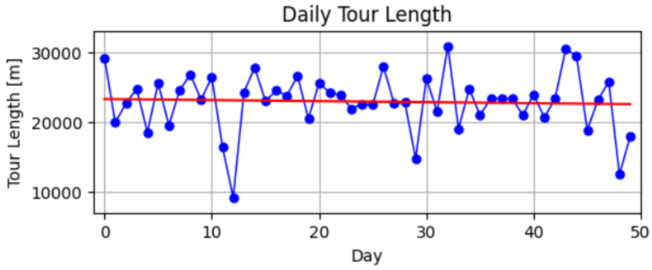
Analyzing the Daily Tour Length:

Based on the analysis of the tour length data from the best recorded simulation(‘bestrec’), statical result shows as follow (22883.22, 23317.5, 4223.945646923415)

Mean: 22883.22 meters
Median: 23317.5 meters
Standard Deviation: 4223.95 meters

Additionally, histogram is generated and plot to visualize the distribution of the tour length in the simulation scenario.

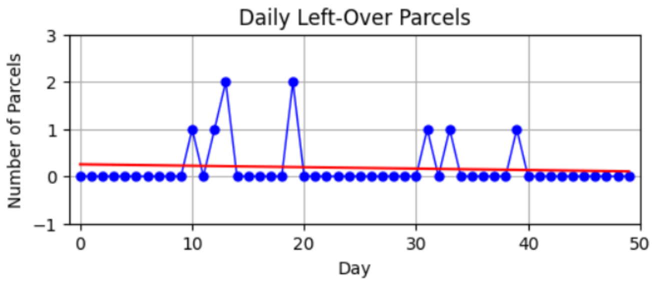




Left over Parcel

The result is as follows.
(0.18, 0.0, 0.48191793695522867)

Mean: 0.18 parcels
Median: 0.0 parcels
Standard Deviation: 0.48 parcels



Timing recorded:

The recorded timing information from the best simulation (bestRec) is as follows:

Total elapsed time: 124.27 seconds
Time spent in createLoopH: 99.62 seconds
Time spent in createLoop: 23.57 seconds
Time spent in addTarget: 1.07 seconds

6. Reflection & Future Works

While the current study provides valuable insights into delivery center location optimization, there are opportunities for further research and improvement. Future work could explore... Additionally, it is essential to consider...

Reflections on the consistency of results with common sense reveal...

7. References

[1] S. J. Wright, "Optimization | Definition, Techniques, & Facts," Encyclopedia Britannica, Apr. 05, 2024. *https://www.britannica.com/science/optimization*

[2] M. Decker, *Last Mile Logistics for Disaster Relief Supply Chain Management: Challenges and Opportunities for Humanitarian Aid and Emergency Relief*. Anchor Academic Publishing, 2013.

[3] G. Perboli, M. Rosano, M. Saint-Guillain, and P. Rizzo, "Simulation–optimisation framework for City Logistics: an application on multimodal last-mile delivery," IET Intelligent Transport Systems, vol. 12, no. 4, pp. 262–269, Mar. 2018, doi: 10.1049/iet-its.2017.0357.

[4] V. M. Yanchuk, "Simulation of package delivery optimization using a combination of carriers and Drop-Ship," Jan. 19, 2021. *https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3769041*