MODELLING, SIMULATION & OPTIMISATION

SHREYANSH GUPTA
StudentId: 21239347

MSc in Data Analytics

Abstract—**In the pursuit of simulating delivery services, his study addresses the operational challenges faced by 'We-Doo,' a pioneering start-up aiming to transform suburban townships' delivery landscape. By modeling evening delivery runs in a small township, the simulation shows the impact of customer parcel frequency (p), cargo bike range limits (r), and associated operational constraints. The outcomes are evaluated through various performance metrics, encompassing daily tour length, working time, left-over parcels, and delays. In the end, this simulation-based analysis equips 'We-Doo' with data-driven insights to inform optimal cargo bike selection and strategic decision-making for efficacious last-mile delivery services.**

I. INTRODUCTION

In an intensively interconnected world, efficient and zero latent delivery services are necessary and have become of utmost priority; in an effort to reach this goal, 'We-Doo,' a pioneering start-up, wants to initiate last-minute delivery services across the nation. This innovative business model will use a secure local distribution point in a safe, friendly neighborhood where items from ordinary delivery providers can be delivered. A fleet of dedicated local drivers will then use electric cargo bikes to make evening deliveries to end customers. The actual motive behind this start-up is to provide the neighborhood with a sustainable, environment-friendly, and customer-oriented delivery service. Before further implementing this delivery service in real-world scenarios 'we-Doo needs to establish and validate its business model strategy. In this report, I will be simulating the typical evening delivery runs for this small township, taking into account various parameters and constraints. The company's success lies in selecting the most suitable electric cargo bike for this purpose.

A. *KEY PARAMETERS:*

*1) **CUSTOMER PARCEL FREQUENCY:**:* The average number of parcels a customer will receive is not known, but we will be assuming it to be within 0.2 to 0.3. This variability necessitates exploring multiple values for a thorough analysis. This frequency could be a critical factor in the simulation. If the average number of parcels a customer receives is too low, the driver may be unable to deliver all the packages in the allotted time. On the other hand, if the average number of parcels per customer is too high, the driver may only be able to complete some deliveries within the range of cargo bikes.

*2) **KEY DEDICATED RANGE OF CARGO BIKES:** :* The company has offered the option of choosing electric cargo bikes with ranges – 30km, 35km, 40km, and 45km. Remember that the company is cash-strapped and is willing to start with the cheapest bike available. While also considering the long-term benefits of using bikes with a 45km range. The range of the cargo bike is an essential factor because if it's too short, then the driver may not be able to reach the customers in the allotted time, whereas if the range is too long, then the company may be spending more money on a bike than they need to.

*3) **WORKING HOURS:** :* The drivers have a planned, estimated working limit of 3 hours per day, from 18:00 to 21:00. This is an essential factor, and this constraint needs to be factored into the simulation for better and realistic results. If the driver works more than 3 hours, then the company will have to overpay them, and if they work less than the planned hours, they might not be able to deliver all the parcels to the customers.

B. *OBJECTIVES –*

The primary purpose of this simulation is to provide the most suitable cargo bike ranges for 'Wee-Doo's' trial run in a town. To achieve this, we will conduct extensive research on different customer parcel frequencies, i.e., p-values, within and beyond the expected range of 0.2 – 0.3. For each p-value, we will evaluate the capacity and performance of the cargo bikes with different ranges (r values). We will get our hands on the crucial factors that influence the variability and sustainability of the model. By presenting a broad report on daily tour length, daily working time, number of left-over parcels, and parcel delay times, we want to give "wee-doo" the data-driven information needed to make intelligent and reliable decisions. Ultimately, this simulation will help create an efficient & successful rollout plan for the 'Wee-doo' in the new commuter township and set its place as a game changer in the last-mile transport market.

II. **LITERATURE REVIEW:**

In one of the papers, [1] The authors have discussed the escalating challenges posed by the rising demand for Courier, Express, and Parcel (CEP) services, resulting from increased online shopping and high product return rates.They presented a Munich-based project as a case study where a CEP company implements last-mile package delivery using cargo bikes and eBikes. The study also highlights the analysis of delivery data, the introduction of an optimization scheme for determining container locations, and the simulation of cargo bike routes. The findings indicate the viability of this approach for densely populated cities, significantly reducing the daily vehicle mileage covered by diesel trucks from 180 km to approximately 45 km.

In another paper, [2] the authors explored the emergence of delivery robots as a novel technology for last-mile logistics, highlighting their autonomous nature and ability to offer customer-selected delivery time slots. The review presents a simulation model for optimizing urban parcel delivery networks, incorporating both traditional delivery vehicles and parcel robots. Mathematical optimization techniques, including solving heuristic extensions of the Traveling Salesman Problem, are employed to enhance decision-making processes within the simulation.

III. METHODOLOGY:

In this simulation, we are using Discrete-Event Simulation. This is a type of simulation where the system is modeled as a sequence of events occurring at specific points in time. Each event thus, in fact, represents a change of state in the system and controls the upcoming behavior of the system. On a larger scale, this simulation is based on agent-based modeling, which involves creating virtual entities and simulating their interactions to understand the behavior of the parcel delivery system. It focuses on the customers, parcels, and a delivery center, all of which work dynamically to mimic real-world scenarios. Events such as customer arrivals, cargo bike departures, parcel handovers, and other activities occur at discrete time points.

We used some vital attributes to build the parcel delivery simulation, like customer parcel frequency, range of cargo bikes, working hours, etc. Here the customer parcel frequency, denoted by p-values, is necessarily kept between 0.2 - 0.3 to maintain the impact of the model in real-world scenarios. The electric cargo bikes are divided into 4 categories concerning the range, i.e., r = 30, 35, 40, 45. The range of these electric cargo bikes is crucial since we need to argue and recommend the best bike for delivery based on the simulation results.

In order to achieve this simulation model, we first generate a map, a warehouse, and a set of customers using the 'generateDATA' function; the function takes several parameters, such as the seed value, the number of nodes in the map, and the number of customers which is stored into a new variable "MyData" [1]. The fig below shows the map with 50 nodes and 100 customer locations per our seed value.

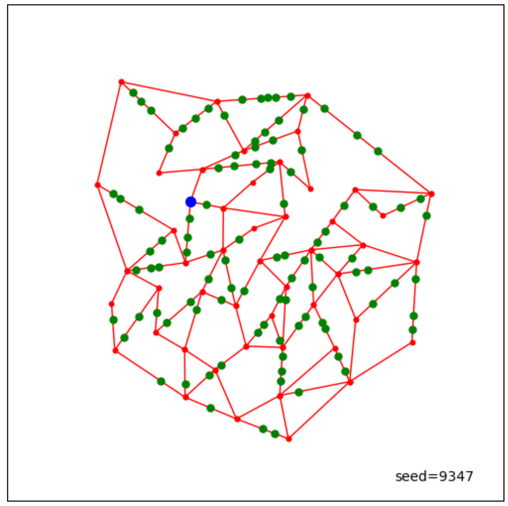

Fig. 1. MAP

We then import the 'Pickle' module. This module is used to serialize and deserialize objects. The 'MyData.pickled' is opened for writing binary data.

```
import pickle
with open('MyData.pickled', 'wb') as f:
    pickle.dump(MyData, f)
```

```
M, W, C = MyData
```

Fig. 2. MyData.pickled

The next step of our simulation is to generate a list of parcels. This is done by calling the 'generateParcels' function (section 2.1). This function takes several parameters, such as the number of days to simulate, the number of customers, the probability of a parcel being generated per day per customer, and whether to plot the results. It starts with creating a list of lists, where each inner list contains the list of parcels for a particular day. The function then generates a number of parcels for each customer, using an exponential distribution. It further adds the parcels to the list of parcels for the appropriate day. Finally, the function plots the number of parcels per customer and the number of parcels per day if the plot flag is set.

The simulation is structured using classes and objects, representing different entities that interact through messages & methods. Uncertainties such as customer response time, parcel arrivals, and handling duration are calculated through randomness. Advanced algorithms such as, A* and Floyd-Warshall [2] optimize the driver routes and calculate the distances efficiently.

A* is a greedy algorithm; it always chooses the next node that seems to be the closest to the destination. This algorithm is often used to locate the shortest paths for a vehicle to travel between two points. On the other hand, Floyd-Warshall is an exhaustive algorithm, [3] which means that it considers all possible paths between two nodes. This makes it guaranteed to find the shortest path, but it can be prolonged for large graphs. We then test these algorithms to find the quickest way from two random nodes, A and B.

The simulation records data such as working time, tour length, and parcel delays using the recorder class, which are kept under vision to evaluate model performance. Visualization

[1] Section 2.2 Of jupyter notebook 'Generate Data'

[2] Section 3.1 & 3.2 of jupyter notebook 'Combining simulation

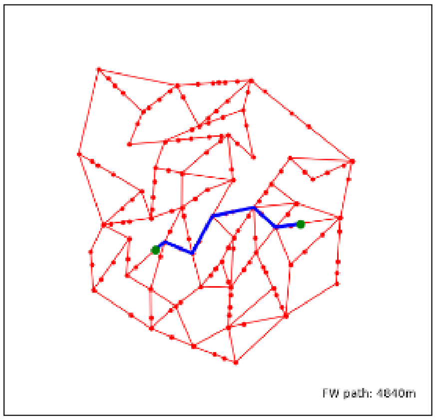

Fig. 3. Floyd-Warshall algorithm

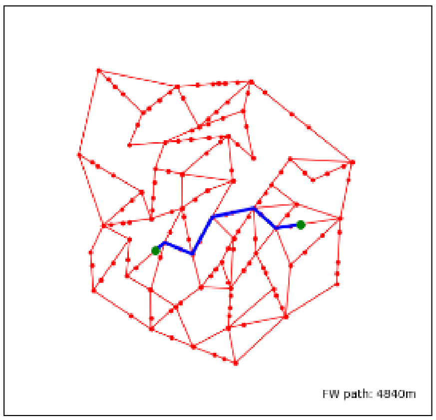

Fig. 4. A* Algorithm

aids, such as Matplotlib, help visually depict the delivery network and other data.

A. *SIMULATION PARAMETERS-*

*1) **Average time - :*** The time required for driving is based on the distance between waypoints at an average speed of 15km/h.

$$\text{AVERAGE\_SPEED} = \frac{15}{3.6}$$

*2) **Preparation time per parcel - :*** The cumulative preparation time (route planning and sorting of the parcels in the delivery order and packing the cargo bike) is assumed to be 50 sec per parcel to be delivered.

PREP_TIME_PER_PARCEL = 50

*3) **Return Time - :*** The time to process returned parcels in the delivery center is 30 sec per parcel.

RETURN_TIME_PER_PARCEL = 30

*4) **Average customer time - :*** The average customer time to answer the door, accept a parcel, or sign off.

AVERAGE_TIME_ANSWER_DOOR = 40

AVERAGE_TIME_HANDOVER = 10

AVERAGE_TIME_SIGNOFF = 10

*5) **End Time - :*** The time for end day closing procedure.

END_OF_DAY_TIME = 600

B. *SIMULATION ROUTINE –*

Firstly, we define a simulation that takes a number of parameters, including map geometry, the number of days to simulate, the number of parcels per day per customer, the bike range limit, the probability that the customer is not at home, the maximum wait time for the customer to answer the door, whether to use a heuristic for routing, whether to log the simulation, whether to plot the simulation, and whether to show progress ticks. After generating a list of parcels, the function creates a simulation environment and a recorder, which in turn is used to track the progress of the simulation.

The function then creates a delivery Centre & a driver. This center is responsible for accepting parcels from customers and dispatching them to the driver. The driver from there on delivers the parcels to the customers. After this, the simulation enters a loop that runs for the number of days specified by the 'days' parameter. In each iteration of the loop, the function does the following:

- Checks if the simulation should be logged or plotted.
- Generates a new list of parcels for the day.
- Generates a new list of parcels for the day.
- Waits for a day to pass.

The function then returns the recorder.

C. *MODEL VERIFICATION & SMALL SIMULATION –*

To verify the model, we load the data from the file ‘MyData.pickled’ and perform a small simulation. This data is a tuple of three objects: the map, the warehouse, and the customer. We then visualize these three objects in order to have a clear understanding of the plot. (fig. 5)

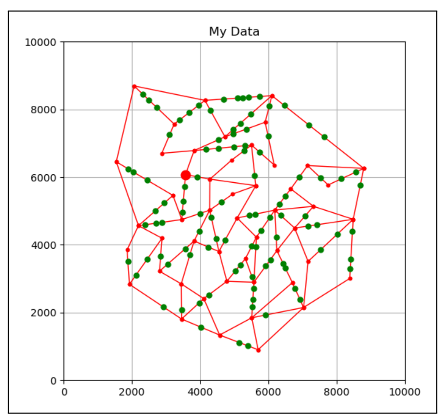

Fig. 5. Small simulation

Now we run two small sample simulations, for 4 days and 10 days, respectively, with a parcel rate of 0.2 parcels per day per customer. We made use of a heuristic for routing.

The evaluations and conclusions are made with the help of histograms of the working time, tour length, and left over parcels. Also, we print a table of the parcel delays to understand the outcomes efficiently.

D. *MULTIPLE LARGE SIMULATION-*

To begin with our actual simulation model, we start by defining two functions: ‘multiSimulation()’ and ‘loadSimulation() .’The multi simulation function runs a number of simulations with different seed values. This function first creates a directory called ’res’ if it does not exist, then it iterates over the list of the seed values. For every seed value the function creates a filename, runs a simulation, and saves the results to the file. The load simulation function takes the same parameters as the multi-simulation function; however, the files are loaded instead of running new simulations.

Now, we call the defined functions after loading the data from the ‘MyData.pickled’ file. Then we combine the function by calling the ‘combineRecorders()’ function. We assign different p-values and limits while calling these functions to compare, contrast, and select the best simulation model. In the end, we analyze our simulation results with the help of visualisations such as histograms and table for Daily working time, daily tour length, daily left-over parcels, and parcel delays.

IV. RESULTS & INTERPRETATIONS:

To evaluate the simulation and bring to a conclusion, we have two major factors, Customer parcel frequency (p-values) & Cargo bike range. For our analysis, we conducted the simulation by examining three different p-values within the range of 0.2 - 0.3, each one with four different range i.e., (30km, 35km, 40km, and 45km). The following observations were made after the simulation.

A. *Customer parcel frequency:*

For each p-value selected, 0.21, 0.25, and 0.29, the simulation was executed to calculate the performance of the delivery system. The following results were obtained:

*1) **Daily tour length**:* The daily tour length was recorded for each of the three p-values and the cargo bike range. It was observed that as the customer parcel frequency increased, the tour length approached the cargo bike range.

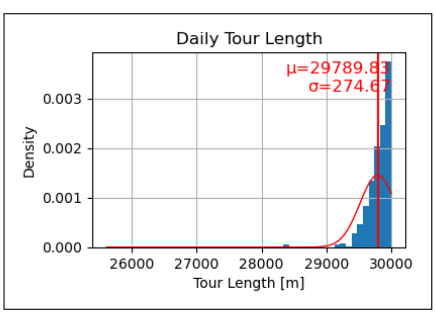

Fig. 6. Tour lenght as per 0.21 and 30km

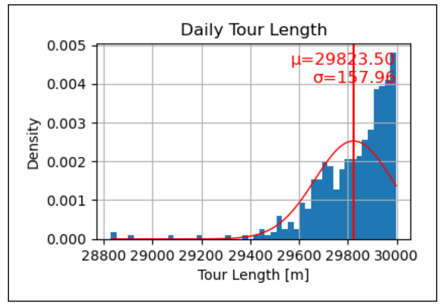

Fig. 7. Tour lenght as per 0.29 and 30km

*2) **Daily working time:** :* The daily working times for delivery drivers were calculated and compared across different p - values and cargo bike ranges. As expected, the increase in the p-values led to longer working times. For cargo bikes ranges closer to the average daily tour length, the working times were sometimes extended due to the constraint of completing deliveries within the working hours. For example, the daily working time of 177.31 minutes was observed when the p-value and limit were 0.21 and 30000, respectively. On the other hand, the same working time was increased to 260.67 minutes when the p-value and limit were set to 0.29 and 45,000, respectively.

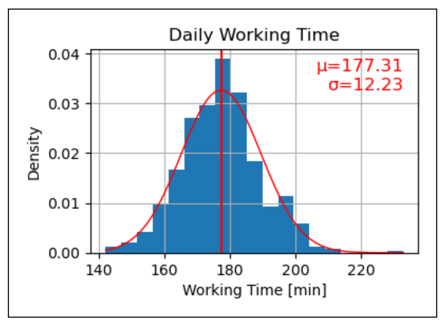

Fig. 8. Daily working time as per 0.21 and 30km

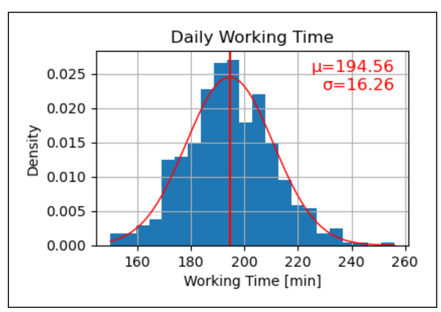

Fig. 9. Daily working time as per 0.29 and 30km

*3) **Number of left-Over parcels-** :* It was noticed that for certain combinations of p-values and limits, there was a gradual increase in left-over parcels. For example, the left over parcels each day was 47.83 when the p-value was 0.21, however, when the p-value was increased to 0.29, the left over parcels each day was also increased to 75.94.

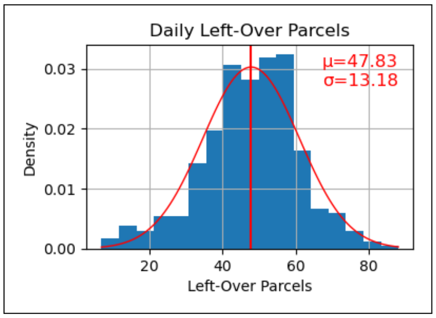

Fig. 10. Left over parcels as per 0.21 and 30km

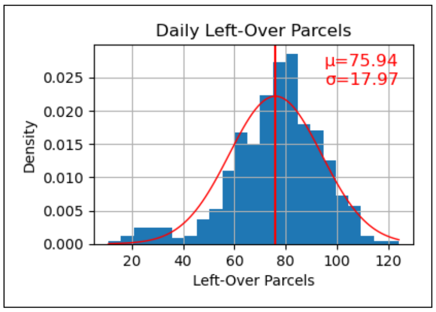

Fig. 11. Left over parcels as per 0.29 and 30km

*4) **Parcel delay times: :*** These delay times define the quality of the service. Higher p-values resulted in increased delay times as the system struggled to accommodate the higher parcels arrived within the limited delivery window. For instance, When the p-value was 0.21 and the limit was set to 30,000 the delay was up to 20 days, whereas when the p-value was increased to 0.29 with limit being constant, the delay was subsequently increased to 30 days.

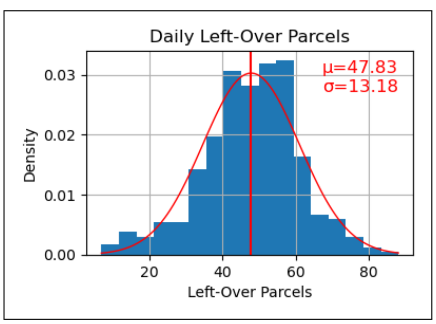

Fig. 12. parcel delays as per 0.21 and 30km

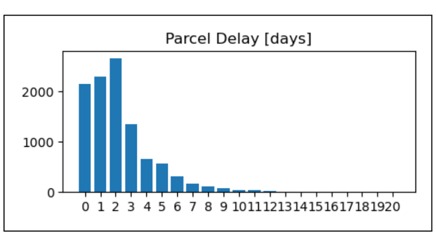

Fig. 13. Left over parcels as per 0.29 and 30km

B. CARGO BIKE RANGE ANALYSIS: :

For each cargo bike range r, the simulation was conducted using various p values to understand the impact of bike range on system performance:

*1) **Daily tour length**:* As the cargo bike range was increased with respect to p-value being constant, the system's ability to complete deliveries within the range improved. This led to fewer delays.

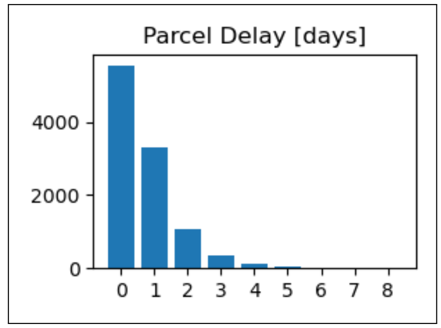

Fig. 14. parcel delays as per 0.21 and 45km

*2) **Daily working time: :*** The daily working time of the driver increased when the range of the cargo bike was increased and as a result the working time exceeded for most of the p-values.

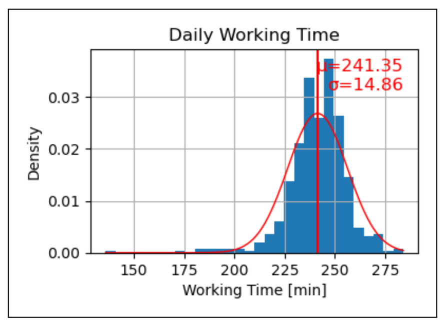

Fig. 15. Daily working time as per 0.21 and 45km

*3) **Number of left-Over parcels- :*** An increase in cargo bike range generally correlated with a reduction in the number of left-over parcels. For example, when the limit was 30km, the left-over parcels were 47.83. Whereas, when the limit was gradually increased to 35km, 40km, and 45km, the results were 33.89, 23.14, and 15.40 respectively. Fig.16 shows the reduction in number of left of parcels.

*4) **Parcel delay times: :*** The increase in the range of the cargo bikes contributed in the reduction in the delay times since the extended range allowed more efficient completion of the deliveries.

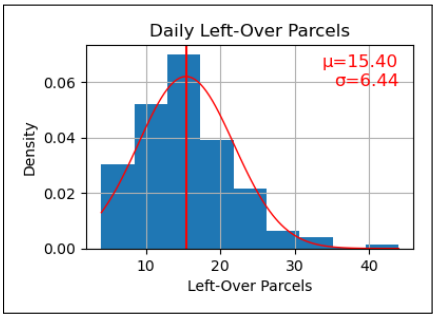

Fig. 16. number of left over parcels per 0.21 and 45km

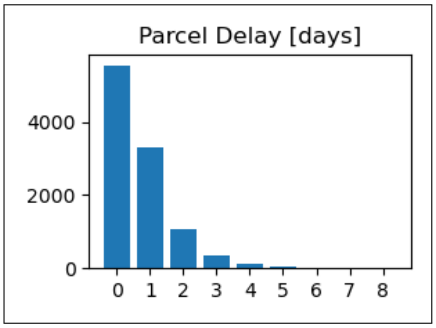

Fig. 17. Parcel delays as per 0.21 and 45km

C. *RECOMMENDATION:*

The best combination of p values and limit for the simulation problem depends on the specific goals of the simulation. However, the following combinations may be a good starting point:

The table 1 below shows the simulation results for different combinations of p-values (average number of parcels a customer receives per day) and cargo bike range limits. The other columns represent various performance metrics for the delivery system under each combination.

P-value	Limit	Working time	Tour length	Left-over parcels	Delays
0.21	30000	177.31	29789.8	47.83	20
0.21	35000	200.42	34742.18	33.89	14
0.21	40000	222.04	39641.18	23.14	11
0.21	45000	241.35	44029.45	15.40	8
0.25	30000	186.68	29803.20	63.24	21
0.25	35000	209.56	34794	45.52	18
0.25	40000	231.88	39743	32.90	12
0.25	45000	252.63	44567	22.97	10
0.29	30000	194.56	29823	75.94	30
0.29	35000	217.96	34807	56.18	19
0.29	40000	240.86	39759	41.32	16
0.29	45000	260.67	44650	28.87	13

TABLE I
TABLE CAPTION HERE.

- P-value = 0.21, limit = 35000: This combination has a relatively low working time and tour length, as well as a small number of left-over parcels. This suggests that the delivery system is able to efficiently deliver most of the parcels within the given time and budget.
- P-value = 0.25, limit = 40000: This combination also has a relatively low working time and tour length, but it has a slightly higher number of left-over parcels. This suggests that the delivery system may be able to deliver even more parcels if the budget is increased.
- P-value = 0.29, limit = 45000: This combination has the highest working time and tour length, but it has the lowest number of left-over parcels. This suggests that the delivery system is able to deliver all of the parcels, but it may take longer than the other combinations.

Ultimately, the best combination of p values and limit will depend on the specific goals of the simulation. If the goal is to deliver as many parcels as possible, then the combination with p = 0.29 and limit = 45000 may be the best choice. However, if the goal is to minimize the working time and tour length, then the combination with p = 0.21 and limit = 35000 may be a better choice.

V. FUTURE WORK AND REFLECTIONS:

The current simulation is a good starting point for studying the delivery process, but there are many possible directions for future work. One area of improvement would be to incorporate more realistic factors into the simulation, such as traffic, weather, and customer preferences. This would make the simulation more realistic and would help to identify the best delivery strategies for different situations. Another area of improvement would be to consider multiple delivery vehicles in the simulation. This would allow us to compare different routing strategies and to optimize the use of resources.

The simulation's ability to accurately represent the complexity of the delivery process in real life must be carefully considered, as must any assumptions or simplifications that could have impacted the outcomes. The accuracy of prediction abilities could be improved by addressing these issues. The examination of parameter sensitivity also reveals key inputs influencing system performance, providing tactical insights for choice-making and resource allocation.

VI. ACKNOWLEDGEMENT

I extend my sincere appreciation to professor horn for generously providing us with the valuable code for the simulation. This code not only expedited our progress but also enriched our learning experience. Professor Horn's dedication to our academic journey has been truly inspiring, and I am deeply thankful for his mentorship and contributions that have significantly enhanced the quality of our work.

REFERENCES

[1] T. Niels, M. T. Hof, and K. Bogenberger, "Design and operation of an urban electric courier cargo bike system," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2531–2537.

[2] M. Poeting, S. Schaudt, and U. Clausen, "Simulation of an optimized last-mile parcel delivery network involving delivery robots," in *Advances in Production, Logistics and Traffic*. Springer International Publishing, 2019, pp. 1–19.

[3] Ramadiani, D. Bukhori, Azainil, and N. Dengen, "Floyd-warshall algorithm to determine the shortest path based on android," *IOP Conference Series. Earth and Environmental Science*, vol. 144, p. 012019, 2018.