Modeling and Optimizing Food Delivery Operations around UPLB Using Discrete Event Simulation and Binary Programming

Dorado, Mikkel A.[1] and Jetomo, Cristian B.[2]

[1] Institute of Agricultural and Biosystems Engineering, College of Engineering and Agro-industrial Technology, University of the Philippines Los Baños, Los Baños, Laguna

[2] Institute of Mathematical Sciences, College of Arts and Sciences, University of the Philippines Los Baños, Los Baños, Laguna

1. Introduction

Online ordering and delivery platforms have grown in recent years, and with them, the complexity of routing, dispatching, and time-sensitive pickup and drop-off operations. As noted by The Meal Delivery Routing Problem (Reyes, Erera, Savelsbergh, Sahasrabudhe, & O'Neil, 2018), "without exaggeration, meal delivery is the ultimate challenge in logistics: a typical order is expected to be delivered within minutes, thus reducing consolidation opportunities and imposing the need for more vehicles operating simultaneously and executing short routes." (Reyes, Erera, Savelsbergh, Sahasrabudhe, & O'Neil, 2018) Their work introduced the "Meal Delivery Routing Problem (MDRP)" to formalize the dynamic pickup and delivery systems (Reyes, Erera, Savelsbergh, Sahasrabudhe, & O'Neil, 2018).

Recently, Meal Delivery Routing Problem with Stochastic Meal Preparation Times and Customer Locations (Kancharia, Woensel, Waller, & Ukkusuri, 2024) extended the MDRP by including uncertainty in meal preparation times and in future order locations, thereby bridging the gap between deterministic modelling and real-world stochasticity. (Kancharia, Woensel, Waller, & Ukkusuri, 2024).

In the setting of the University of the Philippines Los Baños (UPLB) area, where the road network is characterized by one-way segments, varying traffic density, and varying food store locations, there is an opportunity to adapt the MDRP framework. By utilizing open source road network data, specifically from OpenStreetMap, that accurately captures directed roads, and by computing shortest-path distances using Dijkstra's algorithm on a directed network, we can assign drivers to orders with high accuracy.

This study aims to develop a binary-programming model which assigns riders to orders that (i) computes costs of shortest-path distances, (ii) optimally assigns active riders to orders, and (iii) extends the assignment to allow up to two orders from the same restaurant to be bundled to a single rider, reflecting real-world practices in meal delivery. With that, the model mimics the principles of the MDRP while remaining tractable.

2. Objectives of the Study

This project aims to develop and implement a dynamic, simulation-based optimization framework for food delivery systems that efficiently assigns riders to orders, individually and in bundles, in order to minimize the total travel distance of riders while incorporating real-world structural constraints on orders, riders, and logistics.

Specifically, it aims to:

1. construct a discrete-event simulation (DES) framework that captures the dynamic behavior of food delivery operations over a 24-hour period, including order arrivals, preparation times, rider activity states, and delivery completion events;
2. model and represent the road network, restaurants, riders, and order drop-off points in the vicinity of Los Baños, Laguna;
3. formulate a binary programming (BP) model that determines the optimal rider-order assignment by minimizing the total travel distance across the road network;
4. integrate the BP model into the DES framework to establish a periodic computation of optimal assignment; and
5. evaluate the performance of the model using various food delivery operations-related indicators.

3. Methodology

3.1. General Framework

The framework of this project starts with a fixed road network G, a fixed set of restaurants F, and a fixed set of riders R. Each restaurant f has a corresponding location l_f and order preparation length a_f. Each rider r_i also has a corresponding location l_r and activity status a_i. A set of orders O will be initialized with each order $o \in O$ having a placement time a_o, ready time e_o (equivalent to $a_o + a_f$ where a_f is the order preparation length of the restaurant where the order is placed), and a drop-off location l_o. We assume that all information about F and R is known *a priori*, but information about any particular order o is revealed only at its placement time a_o. The focus of the food delivery assignment problem is to determine the optimal rider-order assignment that minimizes the total travel cost across all riders and orders. To refine this objective, assumptions are defined in Section 3.1.1. Furthermore, the details of the framework are provided in Section 3.1.2. Model development is described in Section 3.1.3 as required in the framework, and the model evaluation process is elaborated in Section 3.1.4.

3.1.1. Structural assumptions

Assumptions are set based on three (3) categories: Orders, Riders, and Logistics.

Orders:

- Order arrives following an exponential interarrival time with mean λ.
- Order preparation time per restaurant is invariant over time. Each restaurant has a fixed amount of preparation time regardless of order content.
- Up to 2 orders from the same restaurant can be combined into *bundles* with multiple drop-off locations.

Riders:

- Riders can be *active* or *inactive*.
- Only active riders can be assigned to an order.
- Riders who are in the process of delivering an order are considered inactive. They will return to being active upon reaching the drop-off location.
- A rider's location can only change during a delivery. Thus, a rider's location can only be: their initial location, a restaurant's location, or a drop-off location.
- Riders do not execute any autonomous decisions. They don't have the option to choose which orders to deliver.

Logistics:

- Travel distance between any pair of locations is invariant over time.
- Each rider's travel speed is set to a constant 20 kph. This implies that the travel time between any pair of locations is invariant over time.

3.1.2. Framework Details

To capture the dynamic nature of food delivery systems, the framework will follow a discrete event simulation (DES) approach with a 24-hour timeframe. This framework is illustrated in Figure 1. DES have been used extensively for modeling operational-level behaviors, including order processing (Tako and Robinsons, 2012), food reclamation centers (Mohan *et al.*, 2013), home food waste dynamics (Kandemir et al., 2020), warehouse performance against major events (Abd Ghafar *et al.,* 2023), and more. It is one of the most useful techniques to simulate behaviour and performance of a real-life process, often wherein an entity with events exists (Pooch and Wall, 2024). For our purposes, the environment will be our road network, the entities are the riders and orders, and the events are the placement of orders, order pickups and drop-offs, and rider mobility. By using DES, we can examine the interaction between these entities against the events in the environment.

At time $t = 0$ in our DES, initialization of the restaurants, riders, and road network and their corresponding properties is implemented. After this, the simulation will commence. At time $t = a_o$, order o is introduced to the system and the associated restaurant will start the preparation. The order will be ready at time $t = e_o = a_o + a_f$ provided that the order is placed from restaurant f. This will be done to each additional order placed. After some time length T, the computation of the optimal rider-order assignment will be implemented. Only ready orders are incorporated in this model. Dijkstra's algorithm and a binary programming (BP) model will be used for this computation. Details of which are provided in Section 3.1.3. The model results with the optimal rider-order assignment. The riders will then proceed to their assigned restaurant's location and drop-off points. Upon completion of delivery, the rider is then ready for another order assignment.

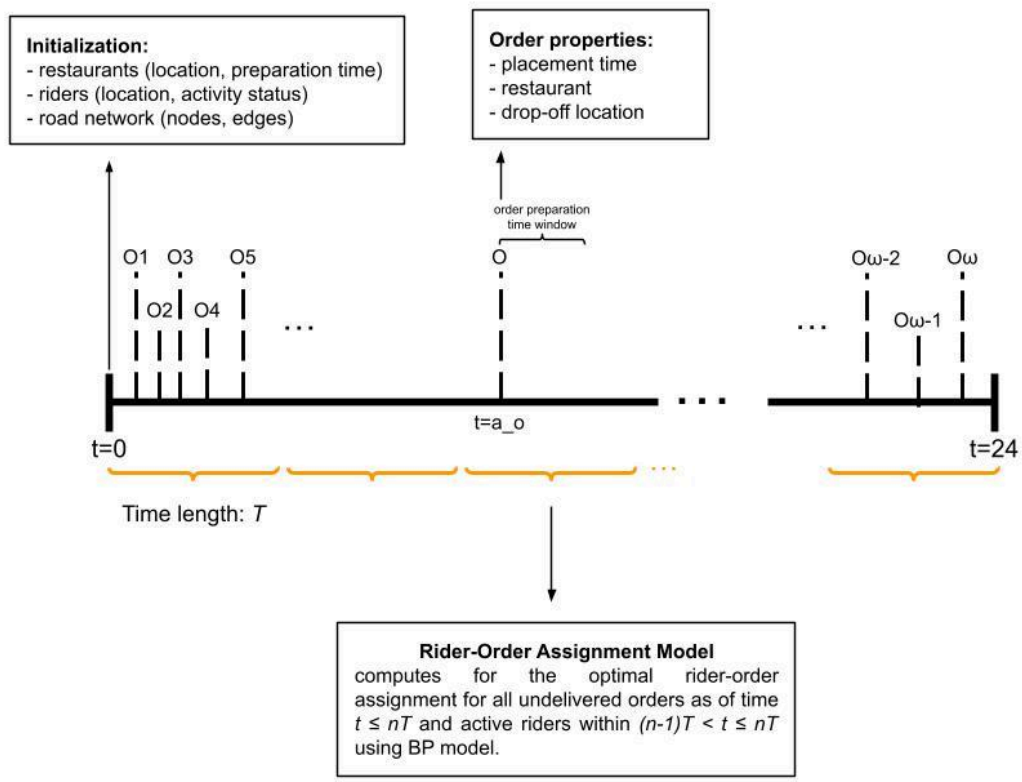

Figure 1: The discrete event simulation framework for the food delivery system.

3.1.3. The Rider-Order Assignment Model (a Binary Programming Model)

The goal of this model is to find the optimal rider-order assignment that minimizes the total distance travelled by riders. This objective requires information on a road network and corresponding distances, locations of riders, restaurants, and drop-off points, and optimal routes between any pair of locations in the network. The road network and locations can easily be represented as data. But the optimal route requires an additional layer of computation. This is where Dijkstra's algorithm can be useful.

Dijkstra's algorithm (Dijkstra, 1959) (also known as the shortest path algorithm) is one of the most used and fastest methods for determining the shortest path between any pair of nodes in a weighted network. The algorithm has been used extensively for traffic route planning, logistic distribution, network optimization, robot navigation, and more (Bing and Lai, 2022). Given two nodes, the result of the algorithm is the shortest path between them and the corresponding shortest path length (or distance).

We can utilize the algorithm to compute the distance between any pair of locations in our food delivery network. That is, suppose rider r_i is assigned to order o_j, we compute the shortest distance from rider r_i to the corresponding restaurant of order o_j and from the restaurant to the corresponding drop-off location of order o_j. We represent the total distance as the *cost* c_{ij} associated with assigning rider r_i to order o_j. We can implement this for each pairwise combination of riders and orders and come up with a *cost matrix* of shortest path lengths from Dijkstra's algorithm.

This approach, while effective, is restricted only to one order delivery for each rider at a time. This is contrary to the usual instances where riders can carry multiple orders from the same restaurant and deliver these orders to their drop-off locations sequentially. Hence for this project, we extend our model above to include the assumption that a rider can bundle up to 2 orders from the same restaurant at a time. The bundle is only limited to 2 orders to balance between capturing scenarios of real food delivery systems and model complexity.

Given a set of orders $O = \{o_1, o_2, ..., o_M\}$, we define its cross-product $P = O \times O = \{(o_1, o_1), (o_1, o_2), ..., (o_M, o_M)\} = \{p_1, p_2, ... p_k, ..., p_{M^2}\}$ as the set of all pairwise orders. Similar with the approach above, we can compute for the shortest distance between a rider r_i to the restaurant associated with the pairwise orders $p_k = (o_n, o_m)$, from the restaurant to the drop-off location for order o_n, and from here to the drop-off location for order o_m. Doing so, we can create a cost matrix with each entry c_{ik} representing the shortest distance associated with assigning rider r_i to pairwise orders p_k.

Note that when rider r_i is assigned to the pairwise orders (o_j, o_j), the total distance will essentially be the distance associated with assigning rider r_i to order o_j, indicating no bundled orders. This means that our extension of the model is valid and still captures our base model above.

Additionally, note that not all orders can be bundled. Hence, we can define a binary variable s_k indicating whether pairwise orders p_k are from the same restaurant. This variable can serve as an activation value that contributes only to the total cost when its value is 1, indicating that bundling the orders is possible.

Finally, we require an indicator for active and inactive riders. We can apply a similar approach by defining a binary variable a_i associated to each rider r_i. This variable only contributes to the model whenever $a_i = 1$ (i.e., when rider r_i is active).

From this setup, our model can now be summarized as follows. Let

- N be the number of riders.
- M be the number of orders.
- R be the set of riders given by $\{r_1, r_2, ..., r_i, ..., r_N\}$
- O be the set of orders given by $\{o_1, o_2, ..., o_j, ..., o_M\}$
- P be the set of pairwise orders given by $O \times O = \{(o_1, o_1), (o_1, o_2), ..., (o_M, o_M)\} = \{p_1, p_2, ..., p_k, p_{M^2}\}$

The parameters are

- c_{ik}: travel distance (cost) associated with assigning r_i to p_k computed from Dijkstra's algorithm.
- $s_k \in \{0, 1\}$: $s_k = 1$ if pairwise orders p_k came from the same restaurant, otherwise $s_k = 0$.
- $a_i \in \{0, 1\}$: rider activity indicator (1 if active, 0 if inactive).

The decision variables are

- $x_{ik} \in \{0, 1\}$: 1 if rider r_i is assigned to pairwise order p_k, 0 otherwise.

The objective is to minimize total cost. That is,

$$Minimize\ z = \sum_{k=1}^{M^2} \sum_{i=1}^{N} c_{ik} \cdot x_{ik}$$

subject to:

(1) Each order is assigned only once

$$\sum_{i=1}^{N} \sum_{k: o_j \in p_k} x_{ik} = 1, \text{ for } j = 1, 2, ..., M$$

(2) At most 2 orders per rider

$$\sum_{k=1}^{M^2} x_{ik} \leq 1 \text{ , for } i = 1, 2, ..., N$$

(3) Bundling feasibility

$$x_{ik} \leq s_k \text{ for } i = 1, 2, ..., N \text{ and } k = 1, 2, ..., M^2$$

(4) Rider activity feasibility

$$x_{ik} \leq a_i \text{ for } i = 1, 2, ..., N \text{ and } k = 1, 2, ..., M^2$$

(5) Binary Constraints

$$x_{ik}, a_i, s_k \in \{0, 1\} \text{ for } i = 1, 2, ..., N \text{ and } k = 1, 2, ..., M^2$$

3.1.4. Model Evaluation

Assessment of the BP model is crucial for determining its applicability for food delivery systems. Hence, in this section, we discuss how we evaluate our model using metrics related to its purpose, specifically on operational efficiency.

One of the metrics we'll look into is the **average delivery time**. This represents the total time from order pickup to customer delivery. That is,

$$average\ delivery\ time\ = \frac{1}{M}\left(\sum_{o\in O}(o_{delivery\ time}\ -\ e_o)\right)$$

Another metric we'll compute is the **average distance travelled**. This allows us to assess how well the model allocates the rider to a specific order. In notation, this is given by

$$ave.\ distance\ travelled\ =\ \frac{1}{N}\left(\sum_{r\in R} r_{total\ distance\ travelled}\right)$$

Finally, we'll look into the **average bundled-unbundled variance**. Here, we compare the added impact of bundling 2 orders from the same restaurant. Specifically, we compute the difference between the travel distances associated with assigning rider r_i to pairwise orders $p_k = \left(o_n, o_m\right)$ and the optimal travel distances of assigning a rider to order o_n alone and order o_m alone, delivering the orders sequentially. For a comprehensive evaluation, these metrics are taken across the 24-hour timeframe of the DES framework.

3.2. Study Site

The simulated food delivery service is limited to food places and customers within Los Baños and Bay, Laguna. The boundary and the road network used in the study are shown in Figure 2.

3.3. Data Description and Collection

3.3.1. Road Network

The initial road network data is obtained from the OpenStreetMap database (OpenStreetMap contributors, 2015) through QGIS (QGIS.org, 2025). Speed limits and other road-use policies (e.g., one-way restrictions, time-dependent access, vehicle-based restrictions) are not included due to the limited amount of this information in the dataset.

Nodes and edges for the simulation network are derived from the initial road network using QGIS. Both nodes and edges are exported as shapefiles for subsequent processing in Python.

The final simulation road network is generated using the NetworkX Python library (Hagberg et al. 2008). Edges are assumed to be bidirectional, and their weights are assigned based on road length (in meters).

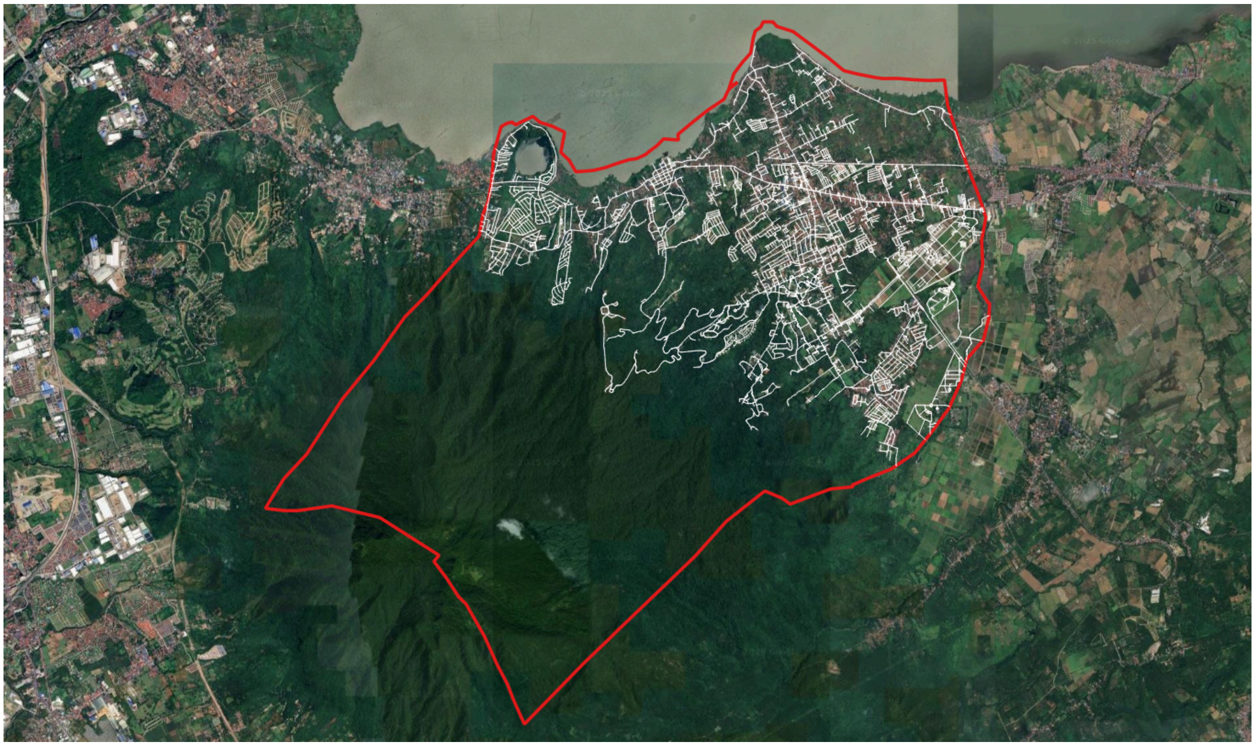

Figure 2: The boundaries of Los Baños, Laguna (red) and its road network (white).

3.3.2. Restaurants

The Restaurants used in this project were obtained from Google Maps. It was then downloaded in a GeoJSON format. We used 54 randomly picked restaurants around the vicinity of UPLB. Figure 3 presents these 54 restaurants and their corresponding locations.

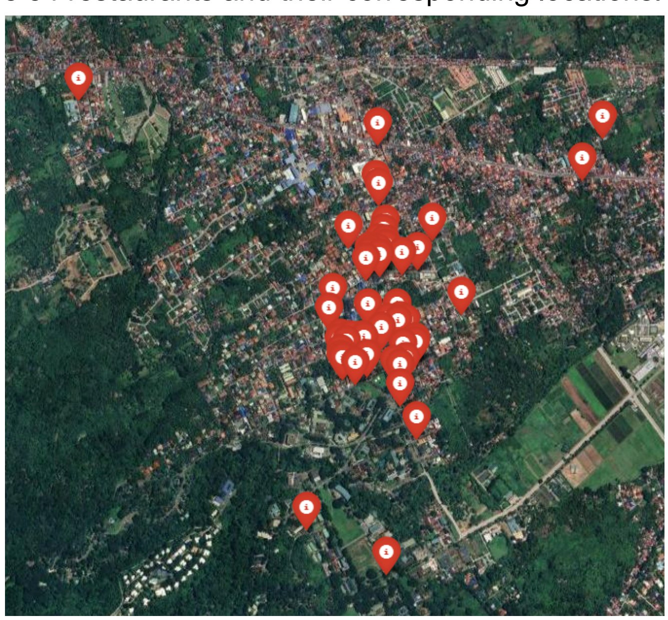

Figure 3: The 54 restaurants used in the simulation mapped in their corresponding locations.

4. Results

The goal of the project is to produce a functional dynamic simulation-optimization framework that captures a simplified operational behaviour of a food delivery system in Los Baños, Laguna. This will be done through the integration of a BP rider-order assignment model to a DES framework, capturing the essential food delivery system dynamics.

To gain a comprehensive analysis of the BP model's performance, we change the values of the number of riders N and the mean order interarrival time λ in the system and observe the corresponding changes in the metrics, gaining some insights into whether the developed system aligns with the actual nature of food delivery operations.

4.1. Varying the Number of Riders N

In analyzing the effect of varying N, we kept the order interarrival time λ fixed at 3 minutes. We tested N = 5, 10, 15, 20, and 25 riders, running 10 simulations for each setting with a different RNG seed each time.

Figure 4a shows that the total number of orders remained relatively constant, which is expected since λ was held fixed.

Figure 4b illustrates that the number of pending orders at the end of the simulation decreased as N increased, reflecting the system's improved ability to accommodate more orders with a larger rider fleet.

Figure 4c indicates that the number of delivered orders per rider decreased with higher N, which follows naturally as the workload is distributed among more riders.

Figure 4d shows a similar pattern: the total distance traveled per rider declined as N increased, again due to workload sharing.

Figure 4e demonstrates that the bundled-unbundled distance variance decreased with larger rider fleets, suggesting that bundling becomes particularly valuable when riders are limited.

Finally, Figure 4f shows that the average delivery time per order decreased as N increased, consistent with the system's improved capacity to serve orders more quickly when additional riders are available.

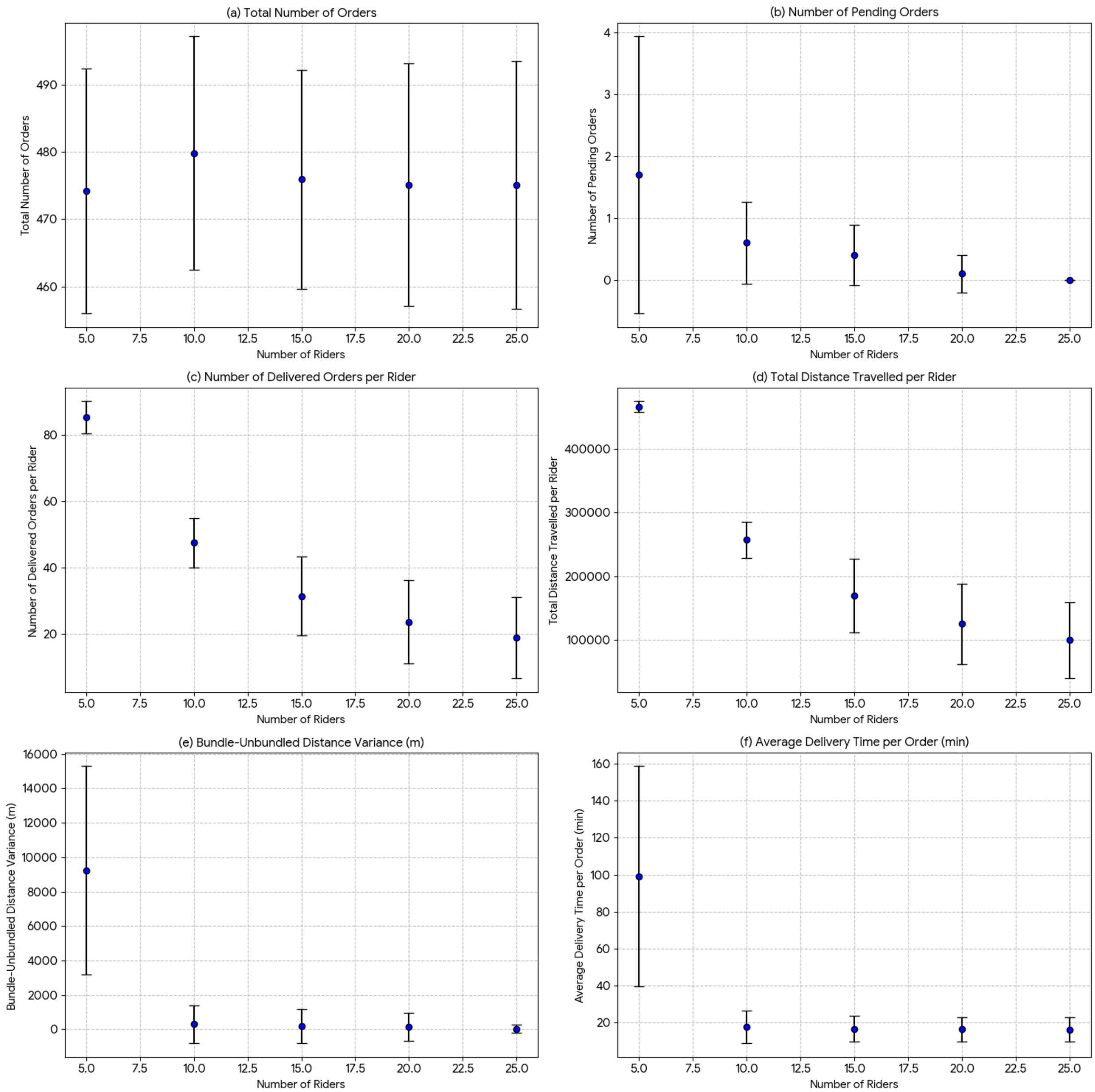

Figure 4: Operational performance outcomes for different rider fleet sizes in the delivery simulation.

4.2. Varying the Order Interarrival Time λ

In analyzing the effect of varying λ, we kept the number of riders fixed at $N = 5$. We tested $\lambda = 3$, 5, and 7 minutes, running 10 simulations for each setting with a different RNG seed for each run.

Figure 5a shows that the total number of orders decreased as λ increased, which is expected since a larger λ corresponds to less frequent orders.

Figure 5b indicates that the number of pending orders at the end of the simulation also decreased with higher λ, as fewer incoming orders reduce overall system load.

Figure 5c shows that the number of delivered orders per rider declined as λ increased, again reflecting the reduced volume of orders.

Figure 5d illustrates that the total distance traveled per rider fell with higher λ, consistent with having fewer orders to deliver.

Figure 5e demonstrates that the bundled–unbundled distance variance decreased as λ increased, suggesting that bundling is especially beneficial when rider capacity is tight. Lower order volumes reduce this constraint.

Finally, Figure 5f shows that the average delivery time per order decreased with higher λ, since fewer concurrent orders increase the likelihood that a rider is immediately available when an order arrives.

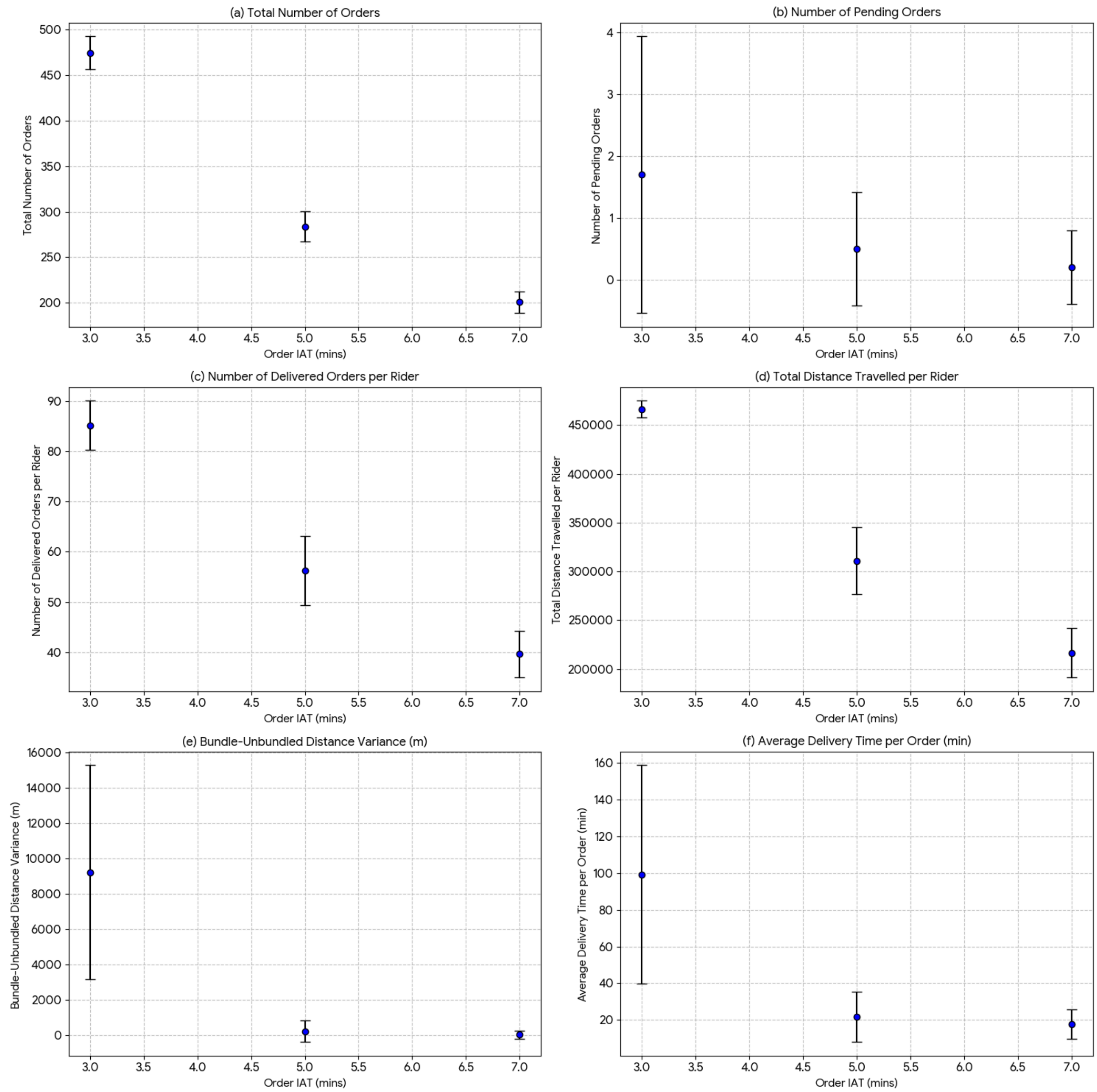

Figure 5: Operational performance outcomes for different order interarrival times in the delivery simulation.

5. Discussion

By examining the parameters we measured: total number of orders, pending orders at the end of the simulation, delivered orders per rider, total distance traveled per rider, bundled-unbundled distance variance, and average delivery time per order, we can derive several key evaluation metrics.

These metrics include:

System Overload: Based on the number of pending orders at the end of the simulation, this metric reflects how many orders the system is unable to fulfill on time.

Rider Utilization: Derived from the number of delivered orders per rider, this measures how many orders are assigned to each rider and how evenly the workload is distributed.

Workload: Using the total distance traveled per rider, this metric captures the level of physical effort required from each rider during a working period.

Timeliness: Based on the average delivery time per order, this metric indicates whether the delivery system can meet the target delivery time under the current configuration.

Order Volume: Derived from the total number of orders, this represents the total demand the system must fulfill within a given period.

Using these metrics, we can assess the delivery system’s performance, identify operational bottlenecks, and evaluate how effectively resources are utilized under different conditions. For example, if System Overload is high, we may need to increase the number of riders. If Rider Utilization shows underused riders, we could reduce the number of riders or reassign orders. High Workload per rider suggests adding more riders to distribute effort evenly. If Timeliness falls short of targets, increasing rider capacity can help meet delivery deadlines. Similarly, rising Order Volume may require more riders to handle the demand. By monitoring these metrics, we can fine-tune the system to achieve optimal performance and resource utilization.

6. Conclusions and Recommendations

The parameters currently being measured help assess the operational performance and reliability of the delivery process. This evaluation system also provides a reference point for the expected daily operational metrics.

To enhance the simulation realism, time-varying order inter-arrival rates, rider break periods, shift schedules, rider quotas, and store operating hours should be incorporated. The road network model should also be improved by integrating data on speed limits, traffic volumes, and road-usage policies.

7. References

Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., & O'neil, R. (2018). The Meal Delivery Routing Problem.

Kancharia, S. K., Woensel, T. V., Waller, T., & Ukkusuri, S. (2024). Meal Delivery Routing Problem with Stochastic Meal. *Springer Nature*, 997-1020.

Tako, A. A., & Robinson, S. (2012). The application of discrete event simulation and system dynamics in the logistics and supply chain context. Decision support systems, 52(4), 802-815.

Mohan, S., Gopalakrishnan, M., & Mizzi, P. J. (2013). Improving the efficiency of a non-profit supply chain for the food insecure. International Journal of Production Economics, 143(2), 248-255.

Kandemir, C., Reynolds, C., Tom, Q., Fisher, K., Devine, R., Herszenhorn, E., … Evans, D. (2020). Using discrete event simulation to explore food wasted in the home. Journal of Simulation, 16(4), 415–435. https://doi.org/10.1080/17477778.2020.1829515

Pooch, U. W., & Wall, J. A. (2024). Discrete event simulation: a practical approach. CRC press.

Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959). https://doi.org/10.1007/BF01386390

Bing, H., & Lai, L. (2022). Improvement and application of Dijkstra algorithms. Acad. J. Comput. Inf. Sci, 5(5), 97-102.

OpenStreetMap contributors. (2015) Planet dump [Data file from 2025-11-15]. Retrieved from https://planet.openstreetmap.org

QGIS.org, 2025. QGIS Geographic Information System. QGIS Association. http://www.qgis.org

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart, “Exploring network structure, dynamics, and function using NetworkX,” Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), Pasadena, CA, USA, pp. 11–15, Aug. 2008.