Jurnal Teknik Industri, Vol. 27, No. 1, June 2025
ISSN 1411-2485 print / ISSN 2087-7439 online

DOI: https://doi.org/10.9744/jti.27.1.93-104

Courier Assignment and Routing Problem Algorithm in Online Food Delivery System with Multi-Customer Delivery Patterns

Prita Meilanitasari[1*], Maulin Masyito Putri[1], Ilda Auly Sita Agustin[1], Muhammad Faisal Ibrahim[1], Dwi Sekar Arumjani[1]
[1)] Department of Logistics Engineering, Universitas Internasional Semen Indonesia
Kompleks PT. Semen Indonesia (Persero) Tbk, Jl. Veteran, Kb. Dalem, Sidomoro, Kebomas, Gresik, East Java 61122, Indonesia
Email: prita.meilanitasari@uisi.ac.id[1], maulin.putri@uisi.ac.id[1], ilda.agustin19@student.uisi.ac.id[1], muhammad.ibrahim@uisi.ac.id[1], dwi.arumjani21@student.uisi.ac.id[1]
*Corresponding author

Abstract: Online food delivery (OFD) businesses face several challenges, including the need for fast deliveries, a high volume of orders, and effective route planning to optimize service efficiency. This study employs the Meal Delivery Routing Problem (MDRP) algorithm to address issues related to courier assignment and capacity management in food delivery operations. The research focuses on scenarios involving a single courier, a single merchant, and multiple demand nodes. Two main methods were used in the study: (1) The Maximum Covering Model (MCM) algorithm, which identifies the coverage area of the courier, and (2) The Flexible Meal Delivery Assignment and Routing Problem (FMD-ARP) algorithm, which tackles routing challenges. Various scenarios were tested to validate the model based on the chosen routes. The aim of this research is to develop a new model and algorithm that reduces delivery time and increases the number of orders that couriers can handle. After processing and analyzing the numerical data, the study identified the most effective scenario that led to improved delivery times and benefits for couriers, enabling them to manage more orders and achieve faster delivery compared to existing algorithms.

Keywords: OFD, MDRP, Maximum Covering Model, FMD-ARP.

Introduction

Food delivery services reached US$96.8 million globally in 2024, indicating a significant increase in demand and growth trends for food delivery apps in the upcoming years [1]. During the pandemic, the OFD sector experienced a revenue surge of 43.2% as consumers shifted from occasional to regular usage [2]. Factors such as convenience, quality of service, price, transaction ease, and promotional strategies significantly influence consumer satisfaction and usage frequency of OFDS [3], [4], [5]. OFD services provide quick access to food, catering to the needs of busy individuals, especially students and working professionals [5], [6]. OFDS provides a streamlined purchasing process, allowing users to access various food options with minimal effort [7], [8]. The average number of meals ordered via the OFD System has nearly doubled, indicating a global trend towards this mode of purchasing meals [9]. The operational challenges of online food delivery, particularly for perishable items, are significant in meeting the increasing demand [10].

The operational challenges faced by online food delivery (OFD) businesses primarily revolve around the need for fast delivery, managing high delivery volumes, and optimizing route selection [11], [12]. Some research indicates that delivery delays can diminish consumers' willingness to pay, prompting platforms to implement on-time delivery (OTD) services with compensation to mitigate negative effects [13], [14]. [15] research indicates that consumers prioritize delivery speed alongside food safety and meal pricing when evaluating service quality. Service quality will affect customer satisfaction with the supply of the restaurant and the delivery service compared to the ideal standard promised to them [16]. Optimizing delivery times is essential for enhancing operational efficiency; for instance, batching orders can improve delivery time estimates by approximately 6% [17]. Another challenge to OFD's business operations is the high volume of demand management. Massive order volumes and ensuring timely deliveries can lead to inevitable delays due to the complexity of the logistics involved [18]. Poor management in high-demand scenarios can undermine the efficiency of the online food delivery ecosystem [19]. The pressure on delivery personnel to complete numerous orders within limited timeframes can also increase stress and a higher likelihood of traffic violations, further exacerbating delays [20].

Optimized delivery routes can reduce waiting time and ensure food quality upon arrival, significantly enhancing customer satisfaction [21], [22]. Dynamic route selection is essential for optimizing delivery times and reducing costs by minimizing fuel consumption [12], [23], [24].

The Meal Delivery Routing Problem (MDRP) algorithm can significantly address operational challenges in the online food delivery business [25]. The Meal Delivery Routing Problem (MDRP) algorithm can optimize delivery routes for sustainable food delivery, enhancing service levels and efficiency in online food delivery businesses [26]. Muralidharan et al., [27]. Mentioned a novel clustering approach dynamically creates restaurant and delivery clusters, optimizing order batching and reducing delivery times. Kim and Chung, [28]. Optimized order assignment and routing for single and multiple deliveries, demonstrating improved efficiency in food delivery. This research focuses on optimizing delivery routes for a single courier serving a single merchant while managing multiple demand nodes. While concentrating on optimizing delivery efficiency, challenges remain in ensuring equitable workload distribution among couriers. The Meal Delivery Routing Problem (MDRP) algorithm addresses equitable workload distribution and minimization to balance efficiency and fairness [29].

The methods used for the MDRP algorithm are the Maximum Covering Model (MCM) algorithm and the Flexible Meal Delivery Assignment and Routing Problem (FMD-ARP) algorithm. The Maximum Covering Model (MCM) algorithm is a combinatorial optimization approach aimed at selecting a subset of resources to maximize coverage of a given set of points or requirements [30]. The MCM can enhance operational efficiency by allowing the bundling of multiple orders and determining courier coverage area, which reduces the overall delivery time and improves service quality [31]. Furthermore, integrating adaptive strategies, such as dynamic rerouting based on real-time data, allows the MCM to respond to fluctuating demand and time-sensitive customer needs [32]. The Flexible Meal Delivery Assignment and Routing Problem (FMD-ARP) algorithm addresses the complexities of meal delivery services by optimizing the assignment of couriers to meal orders while considering various constraints such as time sensitivity and rider wage balance [25], [33]. The FMD-ARP utilizes a rolling horizon strategy to bundle multiple orders, optimizing routes based on spatial and temporal distributions, significantly enhancing delivery efficiency [31], [34].

Existing studies on online food delivery (OFD) primarily focus on optimizing operations in multi-courier and multi-merchant settings, addressing delivery speed, high order volumes, and route efficiency [26]. Many studies explore methods to enhance order assignment, batching, and routing but often treat these as separate optimization problems. However, these approaches are typically designed for large-scale food delivery systems with multiple couriers and do not directly address the specific constraints of a single courier serving a single merchant with multiple demand nodes. The key gap this research aims to fill is the lack of an integrated approach that connects order assignment, pick-up, and delivery with route optimization in a single-courier model. While previous studies have successfully optimized aspects such as batching and delivery time reduction, they do not provide a comprehensive solution that simultaneously considers assignment and routing decisions for a single courier. Additionally, existing algorithms, such as the Maximum Covering Model (MCM), have been applied to maximize coverage and improve service efficiency, but their application in a single-courier setting remains underexplored. Unlike previous studies that focused on general operational improvements for large-scale platforms, this study tailors its optimization strategy to small-scale food delivery businesses, where efficient courier utilization and route planning are critical to maintain service quality and minimize delays. By bridging the gap between assignment and routing in a unified framework, this research contributes novelly to food delivery logistics, ensuring delivery efficiency and fair workload distribution for couriers. This research used the gap analysis above to deal with online food delivery businesses that use single couriers, single merchant nodes, and multi-demand nodes, using Indonesia as a case study for OFD. The manuscript is structured as follows: an introduction, followed by the methodology, which covers the existing food delivery concept and its conceptual model, then the results and discussion, and finally, the conclusion.

Methods

The method used in this study consists of three main components: the existing model, the conceptual model, and the scenario model. The OFD uses the existing model in Indonesia. The conceptual model is the proposed model, and the scenario model will be used to compare and analyze the proposed model.

Existing Model

This section analyzes the current food delivery model used in Indonesia, which involves three main users: food customers, restaurant owners, and couriers. These users connect through an order platform that facilitates communication among them. As illustrated in Figure 1, customers place orders by selecting items from the restaurant menus available on the platform. Once an order is received, the restaurant owner can choose to accept or reject it. When the food is nearly ready, the restaurant activates the platform to locate the nearest courier. The courier then picks up the order from the restaurant. Both the restaurant and the courier notify the platform once the order is completed and picked up. Afterward, the courier is responsible for delivering the order to the customer's address. Typically, each courier delivers to one customer at a time, but they can also take multiple deliveries if needed. The courier decides the order of deliveries based on their own judgment, which can lead to inefficiencies. Our findings indicate that the system does not optimize delivery routes, resulting in couriers generally delivering orders based on proximity or the sequence in which the orders were received, rather than following a more efficient routing strategy. This limitation ultimately contributes to inefficiencies in delivery times and resource utilization.

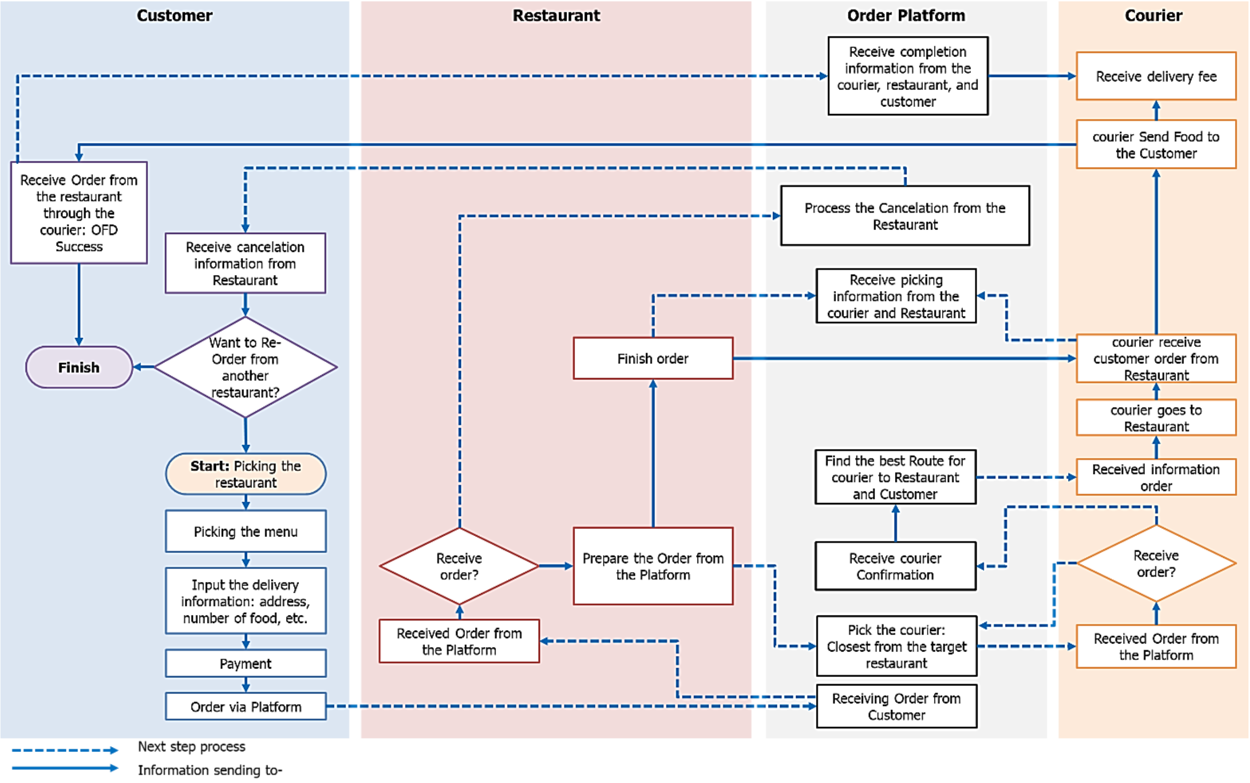

Figure 1. Existing system model for online food delivery

Conceptual Model

In response to the limitations of the existing food delivery model, a new conceptual model has been developed to improve the system's efficiency and effectiveness as seen in Figure 2. This model is organized around three core components or processes: Customer Clustering, Courier Allocation and Order Acceptance Decision, and Routing Optimization, each component is designed to address specific inefficiencies in the existing process.

Customer clustering is a method used to minimize the distance couriers travel when delivering orders to multiple drop-off points. It involves dividing customer locations into clusters to make it easier to assign couriers. This clustering process aims to divide customer areas in a way that allows couriers to handle the maximum number of orders, even when they have time constraints. By grouping customers close to one another, the system can minimize the distance a courier needs to travel between deliveries. This clustering reduces delivery times and helps decrease operational costs by cutting fuel consumption and increasing the number of deliveries a courier can handle within a given timeframe. Furthermore, customer clustering can also improve service quality, as deliveries can be done faster and more accurately.

Courier allocation decision model and allocation decision process: The orders have been entered into the system at this stage, and couriers will be assigned accordingly. It is necessary to allocate couriers by grouping or clustering restaurant locations with the hub (courier gathering point) that can cover those orders. The courier allocation algorithm will select the courier location closest to the restaurant. The allocation algorithm uses the concept of a set covering area problem, which is intended to minimize the time it takes for the courier to reach the restaurant. The allocation algorithm uses a set covering area problem to minimize the time it takes for the courier to reach the restaurant. After the courier is assigned, the courier will be shown options to accept or reject the assignment. An order acceptance decision will be explained in the sub-chapter Scenario Model below.

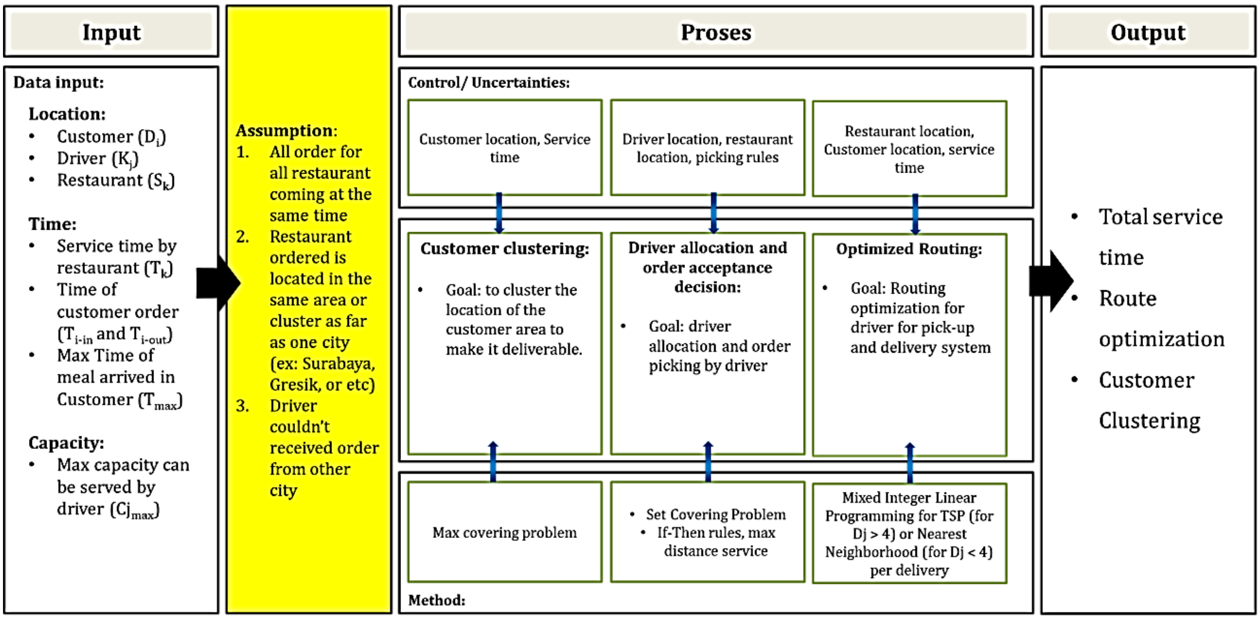

Figure 2. Conceptual model

The third stage of the process involves optimized routing. This means that once the courier has identified the customer nodes for delivery, the delivery route can be optimized to minimize delivery time. In the current model, the courier's decision-making process is the key, but it often leads to delays in delivery times for customers. The proposed model utilizes mixed integer linear programming (MILP) to address the travelling salesman problem (TSP) or nearest neighborhood (NN), depending on the number of customers served. The optimized route is crucial for helping couriers minimize delivery time. In the case of Online Food Delivery (OFD), time is of the essence as it determines the freshness of the food upon arrival to customers.

Scenario Model

There are three scenarios in the scenario model as illustrated in Figure 3. The first scenario represents the current condition of the OFD (Online Food Delivery) model, with the following limitations: The maximum number of orders a courier can take is two and Additional orders ($p2\ldots n$) may appear within the time range TS1 or outside TS1.

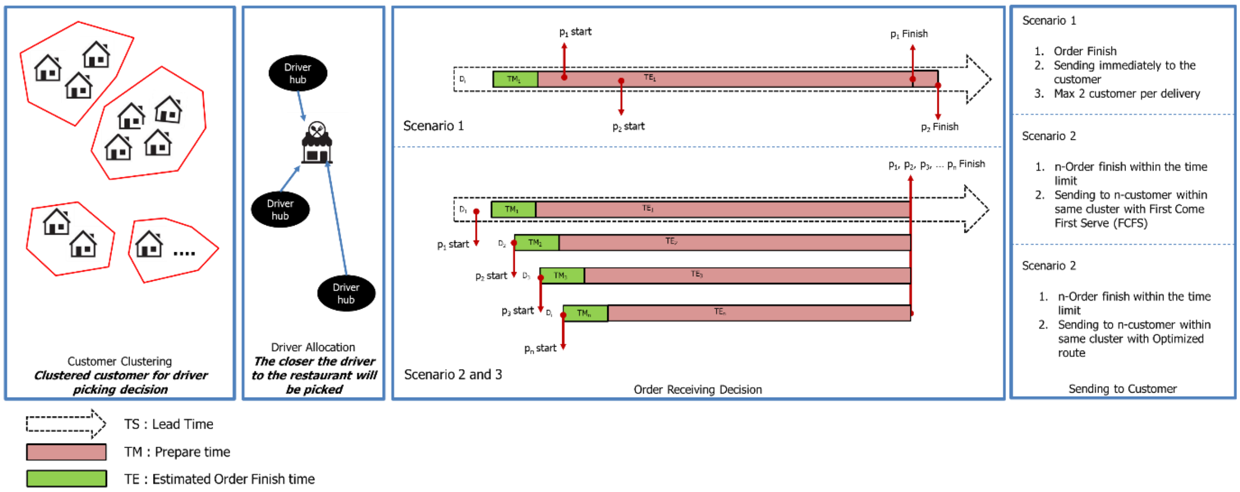

Figure 3. Scenario model

The second scenario is a proposed model that allows couriers to accept more than one order from the same or different customers. The constraints for the scenario are: There is no limit to the number of orders, Additional orders (p2...n) may appear within the time range TS1 or outside TS1, and Orders received outside TS1 will be rejected. Then, Additional orders (p2...n) will be accepted if they appear within the time range TMn-1, At the final stage of the algorithm, the optimal route will be determined, Route determination and the calculation of total trip time will be recalculated each time a new order comes in and Orders must appear within the time range TS1, and the number of accepted orders is based on the estimated time of delivery for the first order.

The third scenario is also a proposed model, however, it differs in the decision-making process for route determination. While the second scenario decides routes based on the time the orders are received, the third scenario is based on distance optimization. The constraints for the third scenario are: There is no limit to the number of orders Additional orders (p2...n) may appear within the time range TS1 or outside TS1, Orders received outside TS1 will be rejected, Additional orders (p2...n) will be accepted if they appear within the time range TMn-1, In the final stage of the algorithm, only the decision to accept or reject the order is made by calculating the estimated completion time for the entire trip of accepted orders. Another one is that orders must appear within the time range TS1, and the number of accepted orders is based on the estimated time of delivery for the first order.

The scenario model differs in the order of acceptance made by couriers. Couriers can only accept the order if the time limit is in the frame. In the final stage of this algorithm, the decision to accept or reject an order is made by calculating the estimated completion time for the entire trip of accepted orders.

Results and Discussions

Numerical Analysis

For numerical analysis, our study use data such as: time order for each customer (t_i), Customer coordinate (x, y), Courier and Restaurant coordinate (x, y), fix time (TM = 10 minutes, TS = 45 minutes, TE = 35 minutes and vehicle speed (v) = 45 km/hour). In this study, we used 50 nodes of customers (C1, C2, C3, …. C50), 20 nodes of restaurants (R1, R2, R3, … R20), and four locations of node hub courier (D1, D2, D3, D4). Those data were then compiled and used for the model as it is for the initial representation for a larger set of nodes.

Customer Clustering

The first step in the proposed algorithm is to cluster customer locations by grouping them based on the nearest distance between points, with a maximum range of 5 km. The method used was the MCM, calculated in Excel using the solver. By exploring various possibilities, clusters of the 50 customers were generated, resulting in 50 cluster combinations as outlined in Table 1.

Table 1. Example of the result for 50 clusters

Cluster 1	C1, C17, C14
Cluster 2	C2, C8, C13, C16, C30, C36, C39, C41, C43, C46, C49
Cluster 3	C3, C20, C21, C24, C28, C33, C35, C40, C42
….	
Cluster 50	C1, C9, C10, C14, C,15, C17, C23, C29, C31, C43, C47, C48, C50

Courier Allocation

Table 2. Result of courier allocation

	R1	R2	R3	R4	….	R20
D1	0,00	0,00	0,00	0,00	….	0,00
D2	0,00	**1,00**	**1,00**	**1,00**	….	0,00
D3	**1,00**	0,00	0,00	0,00	….	0,00
D4	0,00	0,00	0,00	0,00	….	**1,00**

Courier allocation determines which courier will be assigned to a restaurant when a delivery request is made. The first step in this process is to create a distance matrix that calculates the distances from restaurant locations to courier gathering points. Once the distance matrix is established, the next step is to set constraints based on the following details: a single restaurant can be covered by multiple couriers gathering points, and a single courier gathering point can serve multiple restaurants. Using the maximum covering area problem algorithm, the results of the courier allocation can be found in Table 2.

Based on Table 2, Courier 3 (D3) will cover Restaurant 1 (R1) then, Courier 2 will cover R2, R3, R4. Meanwhile Courier 4 (D4) will cover order in R20.

Order Acceptance and Routing

Order acceptance refers to the process of a courier receiving and picking up an order. There are three scenarios to consider: Scenario 1: In this scenario, a courier can accept a maximum of two orders. This limit is enforced without considering customer clustering, courier allocation, or other factors. Scenario 2: This scenario does not focus on optimal routing when determining whether to accept or reject an order. Instead, it processes deliveries in the order in which they are received. three. Scenario 3: Here, optimal routing is considered when deciding whether to accept or reject an order. Deliveries are made in the sequence of the earliest orders that were received. This clearer rendition ensures that the distinctions between the scenarios are more easily understood.

The algorithm used for order acceptance and routine depends on a mathematical model that can be seen in Figure 4.

The algorithm for making order acceptance decisions begins with data initialization. In this study, the maximum delivery time for each courier (TE) is set at 35 minutes, while the maximum cooking time for each dish (TM) is 10 minutes. These times represent the estimated duration that a dish remains fresh by the time the customer receives it. When an order is placed, information about the customer, customer cluster, and restaurant location is immediately provided to the courier. The algorithm then calculates the distance between the restaurant and Customer-i (d_{ki})using the Euclidean distance formula, as shown in equation (1). Next, it estimates the delivery time based on a courier speed of 45 km/h, which is detailed in equation (2).

$$d_{ki} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \tag{1}$$

$$TE_{ki} = \left(\frac{d_{ki}}{45}\right) \times 60 \tag{2}$$

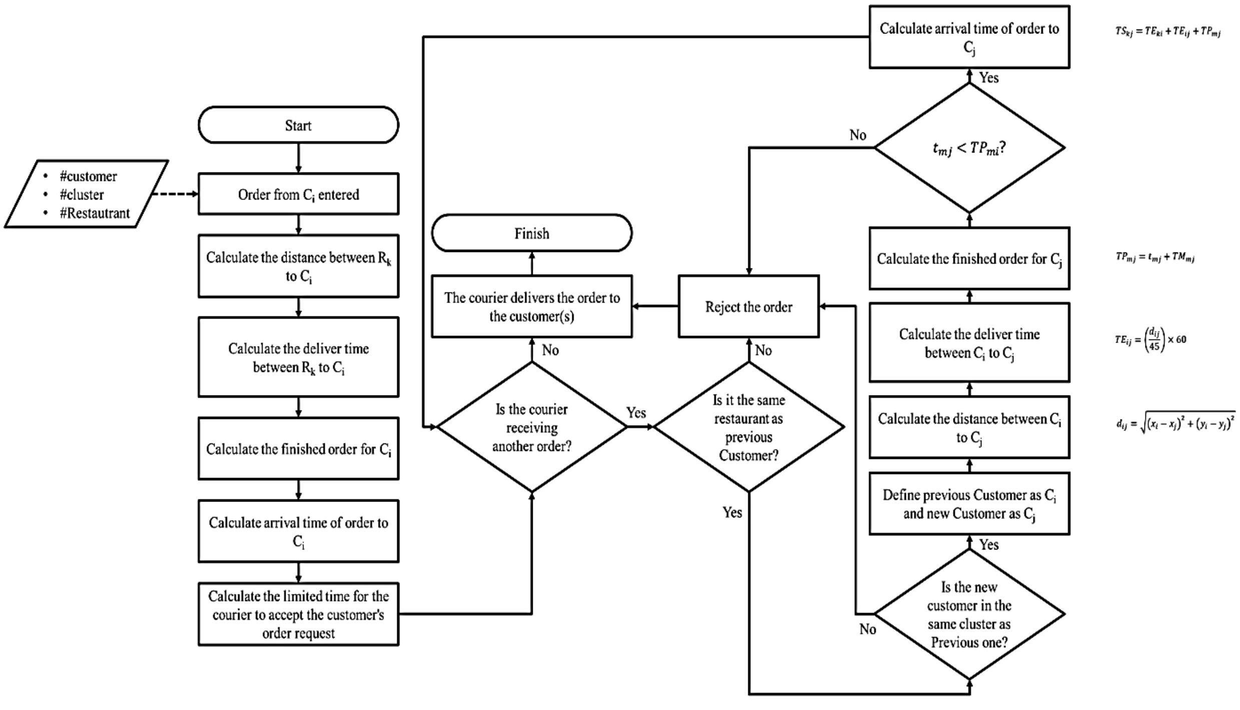

Figure 4. Algorithm for order acceptance decisions

Equation (3) calculates the finished order for C_i (TP_{mi}) by considering the order receiving time (t_{mi}) for the $\mathrm{R_k}$ from the C_i which considers the cooking time for the dish ordered by C_i is TM_{mi} using equations (2) and (3). Then we calculate the arrival time of the order to C_i using equation (4). Equation (4) shows that if there is more than one customer order that arrived, the cooking finish time will consider the max cooking of finished order. But, if the order is only one, then the finished order considered is the incoming order. The limitation of courier for the next order can receive the customer order is equal to the TP_{mi} as shown in the equation (5).

$$TP_{mi} = t_{mi} + TM_{mi} \tag{3}$$

$$TS_{ki} = TE_{ki} + TP_{mi} \tag{4}$$

$$t_{mi} = TP_{mi} \tag{5}$$

After calculating the limited time for the courier to accept the customer's order request, the algorithm uses an IF-ELSE scenario to determine whether the courier has received another order request from a different customer (C_j). If there is another order request, the algorithm will check whether the order from C_j is from the same restaurant (R_k), as the study is limited to a single restaurant with multiple orders. The algorithm will then proceed to the next step if the customer's location is in the same cluster as the previous order (Cluster of C_i = Cluster of C_j). If C_j is not in the same cluster as C_i, the order will be rejected; otherwise, the algorithm will continue to the next calculation.

Assuming the new order request is coming from the same restaurant R_k and the same cluster as C_i, the new order will be declared as C_j and the previous order as C_i. The algorithm will calculate the distance (d_{ij}) using equation (6) and the delivery time (TE_{ij}) between C_i and C_j using equation (7). Next, the finished order for C_j (TP_{mj}) will be calculated using equation (8). An IF-ELSE scenario will then ensure that the finished order time of C_j or (TP_{mj}) is still less than the finished order time of C_i or (TP_{mi}). If the receiving order time of C_j is less than estimated finish cooking order of C_i or $t_{mj} < TP_{mi}$ as in the equation (9), the order of C_j can proceed; if not, it will be rejected. The last calculation is to calculate the arrival time of order to C_j using equation (10).

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{6}$$

$$TE_{ij} = \left(\frac{d_{ij}}{45}\right) \times 60 \tag{7}$$

$$TP_{mj} = t_{mj} + TM_{mj} \tag{8}$$

$$t_{mj} < TP_{mi} \tag{9}$$

$$TS_{kj} = TE_{ki} + TE_{kj} + TP_{mj} \tag{10}$$

Notation for the equation above can be seen at Table 3.

Table 3. Notation for equation (1) to equation (10)

Initial Symbols:		
$\boldsymbol{i}$	:	Initial for the previous (from) Customer as in C_{ij} (i = 1,2, 3…, n)
$\boldsymbol{j}$	:	Initial for the next/other (to) Customer as in C_{ij} (j = 1,2, 3…, n)
$\boldsymbol{k}$	:	Initial for Restaurant-k
$\boldsymbol{m}$	:	Initial for Product order-m
Notations:		
$\boldsymbol{C}$	:	Customer ($C \in i, j$)
$\boldsymbol{R}$	:	Restaurant
$\boldsymbol{P}$	:	Food order
$\boldsymbol{d_{ki}}$	:	Distance between R_k and C_i (kilometers)
$(\boldsymbol{x_k}, \boldsymbol{y_k})$	:	Restaurant location ($R \in k$)
$(\boldsymbol{x_i}, \boldsymbol{y_i})$ or $(\boldsymbol{x_j}, \boldsymbol{y_j})$	:	Customer location ($C \in i, j$)
$\boldsymbol{TM_{mi}}$ and $\boldsymbol{TM_{mj}}$	:	Estimated cooking time for P_m for C_i and C_j (minutes)
$\boldsymbol{TE_{ki}}$ and $\boldsymbol{TE_{kj}}$	:	Estimated delivery time from R_k to C_i and C_j (minutes)
$\boldsymbol{TS_{ki}}$ and $\boldsymbol{TS_{kj}}$	:	Estimated arrival time of P_m from R_k to C_i and C_j (minutes)
$\boldsymbol{TP_{mi}}$ and $\boldsymbol{TP_{mj}}$	:	Estimated finishing order time for P_m for C_i and C_j (minutes)
$\boldsymbol{t_{mi}}$ and $\boldsymbol{t_{mj}}$	:	Receiving order time for P_m for C_i and C_j (minutes)
$\boldsymbol{t_{ij}}$	:	Deliver time from C_i to C_j (minutes)
$\boldsymbol{t_{ki}}$	:	Deliver time from R_k to C_i (minutes)
$\boldsymbol{t_{mi}}$	:	Max order received time for P_m from C_i (minutes)

Numerical Study

Using the algorithm shown in Figure 4, this research used a data model for numerical study, where each condition was generated from a different data model, as shown in Table 4.

Table 4. Data model for numerical study

Number of customers	Number of clusters	Number of restaurants	Receiving order time (t_{mi})	Customer coordinate (x, y)
C1	14	R1	0	(28, 35)
C14	14	R1	2	(27, 32)
C10	14	R1	5	(30, 32)
C7	14	R1	16	(27, 28)

Table 4 shows the receiving order time for each customer. The first customer for R1 is assigned t_{11} = 0. All four customers belong to the same cluster, which is cluster number 14. The first rule, requiring orders to come from the same cluster, is satisfied. Next, we need to calculate the finishing time for each customer to determine whether the second rule, $TP_{mj} < TP_{mi}$, is satisfied. This will determine whether the order request is accepted or rejected by using three scenarios: Scenario 1 (S1), Scenario 2 (S2) and Scenario 3 (S3).

Table 5. Decision for the request order from C_i to R_k for scenario 2 and scenario 3

Number of customers	Customer initial (C_i)	Receiving order time (t_{mi})	Distance between R_k to C_i (d_{ki})	Finishing order time (TP_{mi})	Estimated arrival time from R_k to C_i (TS_{ki})	Order acceptance decision
C1	C_1	0	12.21	10	26.3	S1, S2, S3
C14	C_2	2	10.63	12	32.5	S1, S2, S3
C10	C_3	5	8.60	15	39.5	S2, S3
C7	C_4	16	8.54	$TP_{13} < t_{14}$ (15 < 16)		-
Delivery method for Scenario 1: First come first serve with limited order						
Number of customers allowed				2 customers		
Delivery sequence				R1 – C1 – C14		
Total Distance				d_{ki} (R1 to C1) + d_{ij} (C1 to C14) = 12.21 + 4.2 = 16.41 Km		
Total Delivery Time (Courier speed = ±40 km/hour)				TE_{ki} (R1 to C1) + TE_{ij} (C1 to C14) = 24.62 minutes		
Delivery method for Scenario 2: First come first serve						
Number of customers allowed				3 customers		
Delivery sequence				R1 – C1 – C14 – C10		
Total Distance				d_{ki} (R1 to C1) + d_{ij} (C1 to C14) + d_{ij} (C14 to C10) = 12.21 + 4.2 + 3.0 = 19.41 km		
Total Delivery Time (Courier speed = ±40 km/hour)				TE_{ki} (R1 to C1) + TE_{ij} (C1 to C14) + TE_{ij} (C14 to C10) = 29.12 minutes		
Delivery method for Scenario 3: Nearest Neighbor						
Number of customers allowed				3 customers		
Delivery sequence				R1 – C10 – C14 – C1		
Total Distance				d_{ki} (R1 to C10) + d_{ij} (C10 to C14) + d_{ij} (C14 to C1) = 8.60 + 3.0 + 3.16 = 14.76 km		
Total Delivery Time (Courier speed = ±40 km/hour)				TE_{ki} (R1 to C10) + TE_{ij} (C10 to C14) + TE_{ij} (C14 to C1) = 22.14 minutes		

Table 5 presents the numerical study for Scenario 1, Scenario 2, and Scenario 3 results which are constructed using the algorithm outlined in Figure 4. Scenario 1 can accommodate only two orders at a time. In contrast, Scenarios 2 and 3 differ in the method of food delivery by the courier. In Scenario 2, the courier delivers the food in the sequence that the customers place their orders, starting with the first customer and moving to the next. However, in Scenario 3, the courier prioritizes deliveries based on proximity to the restaurant, beginning with the nearest customer and continuing to the next using an optimization approach. Table 5 illustrates how the algorithm determines whether each order will be accepted or rejected.

Results and Discussion

The results indicated significant differences based on the parameter data obtained from numerical testing. This testing involved randomly generated merchant and customer coordinates, as well as order times across ten different conditions in three test scenarios. Two tests were conducted: one manually and the other using MATLAB software.

Table 6 illustrates the number of demands that can be fulfilled in each scenario. In scenario 1, the maximum number of orders a courier can handle is limited to just two. Conversely, in scenarios 2 and 3, the courier can take on more than two orders. This confirms that scenarios 2 and 3 allow for a greater number of orders to be fulfilled compared to scenario 1. The limitation in scenario 1, which represents the current operating model, restricts the number of orders a courier can carry on a single trip. In contrast, the two proposed scenarios enable couriers to take more orders, provided that other constraints, such as time, are met.

Table 6. Order accumulation

Condition	Fulfilled demand (Number of Customer/Route and Courier allocation/Total time in minutes)		
	Scenario 1	**Scenario 2**	**Scenario 3**
Condition 1	2 customers	3 customers	3 customers
	R1 – C1 – C14 – D4	R1 – C1 – C14 – C10 – D4	R1 - C10 - C14 - C1 – D4
	29.1 minutes	34.1 minutes	29.3 minutes
Condition 2	2 customers	4 customers	4 customers
	R8 – C18 – C5 – D2	R8 – C18 – C5 – C10 – C31 – D2	R8 – C5 – C10 – C31 – C18 - D2
	30.7 minutes	37.9 minutes	28.9 minutes
Condition 3	2 customers	3 customers	3 customers
	R13 – C28 – C22 – D2	R13 – C28 – C22 – C32 – D2	R13 – C22 – C32 – C28 – D2
	26.0 minutes	29.0 minutes	22.0 minutes
Condition 4	2 customers	2 customers	2 customers
	R17 – C44 – C40 – D2	R17 – C44 – C40 – D2	R17 – C40 – C44 -D2
	12.4 minutes	12.4 minutes	9.3 minutes
Condition 5	2 customers	3 customers	3 customers
	R20 – C1 – C17 –D4	R20 – C1 – C17 – C50 – D4	R20 – C1 – C50 – C17 – D4
	27.0 minutes	31.0 minutes	26.1 minutes
Condition 6	2 customers	3 customers	3 customers
	R13 – C3 – C8 – D2	R13 – C3 – C8 – C13 - D2	R13 – C13 – C8 – C3 – D2
	35.1 minutes	35.11 minutes	33 minutes
Condition 7	2 customers	3 customers	3 customers
	R2 – C43 – C47 – D2	R2 – C43 – C47 – C48 – D2	R2 – C43 – C47 – C48 – D2
	21.24 minutes	21.52 minutes	21.52 minutes
Condition 8	2 customers	3 customers	3 customers
	R20 – C13 – C19 – D4	R20 – C13 – C19 – C29 – D4	R20 – C29 – C19 – C13 – D4
	29.07 minutes	29.47 minutes	19.9 minutes
Condition 9	2 customers	4 customers	4 customers
	R10 – C14 – C47 – D4	R10 – C14 – C47 – C50 – C9 – D4	R10 – C9 – C47 – C50 – C14 – D4
	17.22 minutes	23.41 minutes	16.6 minutes
Condition 10	2 customers	5 customers	5 customers
	R3 – C22 – C26 – D2	R3 – C22 – C26 – C28 – C32 – C25 – D2	R3 – C25 – C32 – C22 – C28 – C26 – D2
	21.21 minutes	42.43 minutes	30.7 minutes

The analysis of Table 6 reveals that Scenario 1 is fundamentally distinct from Scenarios 2 and 3, as it only accommodates two orders from two different customers. This makes direct comparison invalid. Thus, the focus of the analysis is on Scenarios 2 and 3, where the number of customers served remains consistent across both cases. In all 10 conditions examined, 100% of the instances demonstrate that the number of customers served in Scenario 2 and Scenario 3 is the same. However, the key difference lies in travel time and the routes taken.

Scenario 2 operates on the First-Come, First-Served (FCFS) principle, while Scenario 3 employs route optimization using the Nearest Neighbor (NN) method. As a result, in the 10 conditions with the same customer count, the travel time in Scenario 3 is notably shorter than that in Scenario 2. From a complex standpoint, Scenario 3 can identify customer locations nearer to the restaurant first and then determine subsequent delivery locations based on the NN concept. This approach reduces both time and distance, leading to greater efficiency.

However, when considering freshness, Scenario 2, which utilizes the FCFS approach, may be preferable. This is because it logically prioritizes the delivery of the first order as quickly as possible, even if this results in increased time and cost. Ultimately, in terms of overall efficiency, Scenario 3 proves to be more advantageous for couriers by optimizing time and distance.

It is important to note that these findings are specific to the conditions and parameters outlined in this study. The results could vary in different contexts, influenced by factors such as customer distribution, order characteristics, and routing constraints. Therefore, the conclusions drawn here should not be generalized beyond the scope of this research.

Conclusions

The meal delivery algorithm influences the courier, restaurant location, customer location, travel time, and food processing time. The conceptual model involves three scenarios. Scenario 1 is a scenario that exists in existing conditions where the courier can take the second order after the restaurant has processed the first order. Scenario 2 is a scenario that uses the first customer as a reference for decision-making for the next customer with a time limit, namely the completion of the first order with delivery adjusted to the order entered by the courier first. Scenario 3 is a scenario that estimates the optimal delivery route. Of the three scenarios, scenario 3 is the most effective in terms of the number of orders that can be taken by the courier or the total travel time. It is because scenario 3 considers the closest distance the courier can travel from the restaurant to all customers taken.

Numerical trials show that this flexible meal delivery algorithm is quite effective. However, this study still uses many assumptions, including whether the type of food taken is considered the same or requires the same preparation time. This is because the algorithm created cannot predict customers' numbers and waiting times well. Another thing that we did not consider is the change in the initial data time like cooking time, travel speed, and the number of nodes. It is possible that those data are changing flexibly. Future research must simulate the flexible meal delivery system algorithm to obtain a better picture by combining it with flexible learning theory to accommodate the flexible data changing.

Acknowledgment

This research is funded by Universitas International of Semen Indonesia under the internal research funding Research Grant for Expertise Group Study (HRKK 2023-2024).

References

[1] A. Sharma, N. Sharma, and A. Baiswar, “Food delivery app: trend, features and technologies used for app development,” *International Journal of Scientific Research in Engineering and Management IJSREM*, vol. 06, no. 04, Apr. 2022, doi: 10.55041/IJSREM12569.

[2] M. Elsiana and A. F. Maradona, “Service quality and customer loyalty in the online food delivery service industry,” *Quantitative Economics and Management Studies*, vol. 5, no. 1, pp. 197–203, Feb. 2024, doi: 10.35877/454RI.qems2407.

[3] C. A. R. Arias and R. I. Bayna-Mariano, “Online food delivery service satisfaction, food choices, and usage frequency among UPLB students aged 18-24 during the covid-19 pandemic,” *Journal of Human Ecology and Sustainability*, vol. 2, no. 1, p. 4, Feb. 2024, doi: 10.56237/jhes23012.

[4] N. Idris, M. N. F. Abu Bakar, Y. Yakub, D. Arabi, and M. N. Almunawar, “Online food delivery services during covid-19: A case study in Brunei Darussalam,” in *Advances in Finance, Accounting, and Economics*, P. Ordóñez De Pablos, X. Zhang, and M. N. Almunawar, Eds., IGI Global, 2022, pp. 335–365. doi: 10.4018/978-1-6684-5849-5.ch019.

[5] M. H. Jalis, N. S. C. Deraman, A. N. Ayub, N. ‘Aqilah Hazlan, and N. F. Y. A. Latib, “Influential factors of utilising the online food delivery (ofd) service among students at Dungun Campus, UiTM Terengganu Branch,” *E-Academia Journal*, vol. 12, no. 1, pp. 47–61, May 2023, doi: 10.24191/e-aj.v12i1.21540.

[6] N. J. A. Maalouf, E. Sayegh, W. Makhoul, and N. Sarkis, “Consumers’ attitudes and purchase intentions toward food ordering via online platforms,” *Journal of Retailing and Consumer Services*, vol. 82, p. 104151, Jan. 2025, doi: 10.1016/j.jretconser.2024.104151.

[7] S. Sujatha and G. Phaneendra, “A study on customer’s satisfaction and perception towards online food delivery apps,” *International Journal for Multidisciplinary Research*, vol. 6, no. 3, p. 23326, Jun. 2024, doi: 10.36948/ijfmr.2024.v06i03.23326.

[8] M. Keeble, T. Burgoine, and J. Adams, “P14 investigating frequent online food delivery service use: a qualitative study in UK adults,” in *SSM Annual Scientific Meeting*, BMJ Publishing Group Ltd, Aug. 2022, p. A54.3-A55. doi: 10.1136/jech-2022-SSMabstracts.113.

[9] A. Gupta *et al.*, “Use of online food delivery services among adults in five countries from the International Food Policy Study 2018–2021,” *Preventive Medicine Reports*, vol. 43, p. 102766, Jul. 2024, doi: 10.1016/j.pmedr.2024.102766.

[10] J.-F. Chen *et al.*, “An imitation learning-enhanced iterated matching algorithm for on-demand food delivery,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18603–18619, Oct. 2022, doi: 10.1109/TITS.2022.3163263.

[11] N. B. T. Nguyen, G.-H. Lin, and T.-T. Dang, “Fuzzy multi-criteria decision-making approach for online food delivery (OFD) companies’ evaluation and selection: A case study in Vietnam,” *Processes*, vol. 9, no. 8, p. 1274, Jul. 2021, doi: 10.3390/pr9081274.

[12] I. W. E. Gaida and M. Mittal, “Efficient supply chain delivery planning considering dynamic route selection using Ant Colony Optimization,” in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India: IEEE, Oct. 2022, pp. 1–8. doi: 10.1109/ICRITO56286.2022.9964847.

[13] Z. Du, Z.-P. Fan, and Z. Chen, “Implications of on-time delivery service with compensation for an online food delivery platform and a restaurant,” *International Journal of Production Economics*, vol. 262, p. 108896, Aug. 2023, doi: 10.1016/j.ijpe.2023.108896.

[14] R. Pan, Y. Huang, and X. Xiao, “Evaluating consumers’ willingness to pay for delay compensation services in intra-city delivery—A value optimization study using choice,” *Information*, vol. 12, no. 3, p. 127, Mar. 2021, doi: 10.3390/info12030127.

[15] W. Liu and W. J. Florkowski, “Online Meal delivery services: Perception of service quality and delivery speed among Chinese consumers,” 2018, doi: 10.22004/AG.ECON.266866.

[16] A. Putra, Z. J. H. Tarigan, and H. Siagian, “Influence of information quality on retailer satisfaction through supply chain flexibility and supplier relationship management in the retail industry,” *Jurnal Teknik Industri*, vol. 22, no. 2, pp. 93–102, Dec. 2020, doi: 10.9744/jti.22.2.93-102.

[17] R. P. Moghe, S. Rathee, B. Nayak, and K. M. Adusumilli, “Machine learning based batching prediction system for food delivery,” in *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, Bangalore India: ACM, Jan. 2021, pp. 316–322. doi: 10.1145/3430984.3430999.

[18] J. Zheng, L. Wang, L. Wang, S. Wang, J.-F. Chen, and X. Wang, “Solving stochastic online food delivery problem via iterated greedy algorithm with decomposition-based strategy,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 2, pp. 957–969, Feb. 2023, doi: 10.1109/TSMC.2022.3189771.

[19] C. Duthie *et al.*, “Online on-demand delivery services of food and alcohol: A scoping review of public health impacts,” *SSM - Population Health*, vol. 21, p. 101349, Mar. 2023, doi: 10.1016/j.ssmph.2023.101349.

[20] X. Lu, X. Guo, J. Zhang, X. Li, L. Li, and S. Jones, “Reducing traffic violations in the online food delivery industry—A case study in Xi’an City, China,” *Front. Public Health*, vol. 10, p. 974488, Oct. 2022, doi: 10.3389/fpubh.2022.974488.

[21] M. Sun, Y. Xu, F. Xiao, H. Ji, B. Su, and F. Bu, “Optimizing multi-echelon delivery routes for perishable goods with time constraints,” *Mathematics*, vol. 12, no. 23, p. 3845, Dec. 2024, doi: 10.3390/math12233845.

[22] J. Li, “Optimization of multi-objective fresh food e-commerce delivery route with fuzzy time window,” in *International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2022)*, R. Liang and J. Wang, Eds., Guangzhou, China: SPIE, May 2023, p. 134. doi: 10.1117/12.2674681.

[23] K. Thopate, T. Bang, Y. Bankar, M. Bansal, and A. Bagade, “Efficient transport optimization,” in *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, Bangalore, India: IEEE, Apr. 2023, pp. 554–557. doi: 10.1109/ICAECIS58353.2023.10170624.

[24] D. Al-Tayar and Z. Alisa, “Enhancing sustainability in logistics through stochastic network routing mechanism optimization using ant colony algorithm,” *Heritage and Sustainable Development*, vol. 5, no. 2, pp. 229–238, Sep. 2023, doi: 10.37868/hsd.v5i2.239.

[25] W. Wang and S. Gao, “Waiting strategy for the dynamic meal delivery routing problem with time-sensitive customers using a hybrid adaptive genetic algorithm and adaptive large neighborhood search algorithm,” *Systems*, vol. 12, no. 5, p. 170, May 2024, doi: 10.3390/systems12050170.

[26] K. R. Thipparthy *et al.*, “Optimizing delivery routes for sustainable food delivery for multiple food items per order,” *Discover Sustainability*, vol. 5, no. 1, p. 122, Jun. 2024, doi: 10.1007/s43621-024-00326-y.

[27] A. Muralidharan, N. Harsha, T. V. Bhat, A. Elizabeth, and K. Srinivas, “Optimal delivery positioning algorithm using clustering,” in *2024 International Conference on Circuit, Systems and Communication (ICCSC)*, Fes, Morocco: IEEE, Jun. 2024, pp. 1–4. doi: 10.1109/ICCSC62074.2024.10617140.

[28] S. H. Kim and B. D. Chung, “Integrated food delivery problem considering both single-order and multiple order deliveries,” *Computers & Industrial Engineering*, vol. 196, p. 110458, Oct. 2024, doi: 10.1016/j.cie.2024.110458.

[29] A. Martínez-Sykora, F. McLeod, T. Cherrett, and A. Friday, “Exploring fairness in food delivery routing and scheduling problems,” *Expert Systems with Applications*, vol. 240, p. 122488, Apr. 2024, doi: 10.1016/j.eswa.2023.122488.

[30] F. Tomova, S. Filipov, and A. Avdzhieva, "Covering a maximum number of points by a fixed number of equal disks via simulated annealing," *Journal of Chemical Technology and Metallurgy*, vol. 59, no. 2, pp. 457–464, Jan. 2024, doi: 10.59957/jctm.v59.i2.2024.26.

[31] L. Zhou, A. Ye, and S. Hu, "A four-stage heuristic algorithm for solving on-demand meal delivery routing problem," 2022, *arXiv*. doi: 10.48550/ARXIV.2212.03505.

[32] W. Wang and S. Gao, "An Alns-enhanced adaptive genetic algorithm for dynamic meal delivery routing problem with time-sensitive priority and waiting strategy," 2023, *SSRN*. doi: 10.2139/ssrn.4551614.

[33] Y. Xie and Z. Wu, "A genetic algorithm for meal delivery problem considering rider's wage balance," in *2024 36th Chinese Control and Decision Conference (CCDC)*, Xi'an, China: IEEE, May 2024, pp. 4873–4879. doi: 10.1109/CCDC62350.2024.10588199.

[34] M. D. Simoni and M. Winkenbach, "Crowdsourced on-demand food delivery: An order batching and assignment algorithm," *Transportation Research Part C: Emerging Technologies*, vol. 149, p. 104055, Apr. 2023, doi: 10.1016/j.trc.2023.104055.