



Print ISSN: 1738-3110 / Online ISSN 2093-7717  
 JDS website: <http://accesson.kr/jds>  
<http://doi.org/10.15722/jds.22.09.202409.85>

## Bundle System in the Online Food Delivery Platform\*

Tae Joon PARK<sup>1</sup>, Myoung-Ju PARK<sup>2</sup>, Yerim CHUNG<sup>3</sup>

Received: July 6, 2024. Revised: August 8, 2024. Accepted: August 15, 2024.

### Abstract

**Purpose:** Online food delivery platforms face challenges to operational efficiency due to increasing demand, a shortage of drivers, and the constraint of a one-order-at-a-time delivery policy. It is imperative to find solutions to address the inefficiencies in the food delivery industry. Bundling multiple orders can help resolve these issues, but it requires complex computations due to the exponential increase in possible order combinations. **Research design, data and methodology:** This study proposes three bundle delivery systems—static, dynamic, and hybrid—utilizing a machine learning-based classification model to reduce the number of order combinations for efficient bundle computation. The proposed systems are analyzed through simulations using market data from South Korea's online food delivery platforms. **Results:** Our findings indicate that implementing bundle systems extends service coverage to more customers, increases average driver earnings, and maintains lead times comparable to standalone deliveries. Additionally, the platform experiences higher service completion rates and increased profitability. **Conclusions:** This suggests that bundle systems are cost-effective and beneficial for all stakeholders in online food delivery platforms, effectively addressing the inefficiencies in the industry.

**Keywords :** Food Delivery, Platform Business, Bundle Delivery, Autonomous Decision-Making, Machine Learning

**JEL Classification Code :** C55, L87, L91

### 1. Introduction

The Online Food Delivery Platform (OFDP) industry has experienced rapid growth since COVID-19. With a valuation of USD 221.65 billion in 2022, the global online food delivery market is anticipated to grow at a compound annual growth rate (CAGR) of 10.3% from 2023 to 2030 (Grand View Research, 2022). Food delivery services through online platforms are now viewed as essential rather than optional within the restaurant industry. The surge in demand has led to challenges in meeting immediate delivery requirements, especially during mealtime hours when 60%

of orders are placed. To reduce costs, OFDPs work with freelance drivers who can choose orders based on profit maximization, which can sometimes compromise customer satisfaction. Drivers often bundle multiple orders to save time and distance, but this practice can lead to decreased food quality. Despite some platforms banning order bundling to maintain food quality, the shortage of drivers during peak hours highlights the need for bundle delivery systems that increase the number of delivered orders while maintaining high service quality.

To achieve efficient bundle delivery, it is essential to respect drivers' control over their schedules and routes,

\* This work was supported by Yonsei Business Research Institute.

1 First Author: Doctor, Yonsei School of Business, Yonsei University, South Korea. Email: [xoxoqkr@naver.com](mailto:xoxoqkr@naver.com)

2 Second Author: Professor, Department of Industrial and Management Systems Engineering, Kyung Hee University, South Korea. Email: [pmj0684@khu.ac.kr](mailto:pmj0684@khu.ac.kr)

3 Corresponding Author: Associate Professor, Yonsei School of Business, Yonsei University, South Korea. Email: [yerimchung@yonsei.ac.kr](mailto:yerimchung@yonsei.ac.kr)

© Copyright: The Author(s)

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted noncommercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

known as autonomy, while ensuring high service quality. Platforms must consider the impact on customer satisfaction, guarantee timely deliveries to maintain food freshness and optimize computational resources to avoid system overload. To address these challenges, we propose a bundle construction system that can reduce computation time and customize bundles for each driver, preserving their freedom to choose.

The paper is structured as follows: Section 2 provides related works on online food delivery platforms and existing bundle delivery systems. Section 3 introduces the framework for the bundle delivery system. Section 4 describes the specific bundle delivery systems proposed in this study. Section 5 analyzes the results of the experiments conducted. Section 6 concludes the paper and suggests directions for future research.

## 2. Related Works

Online Food Delivery Platforms (OFDPs) operate within the on-demand economy, where both customers and drivers use the platform. Customers and drivers connect with various service or goods providers through the platform (Chakravarty et al., 2014). Drivers have autonomy within their relationship with the platform; they can easily switch between different OFDP apps by deleting one app and installing another. Therefore, OFDPs must consider drivers' preferences, needs, and autonomous behavior in their operational strategies. Drivers' autonomy (Tilly & Tilly, 1998), defined as control over time, place, and tasks, is critical in the gig economy, including OFDPs, where workers self-schedule and accept or reject jobs (Shapiro, 2018). This autonomy also enhances drivers' intrinsic motivation, sense of competence, and feeling of relevance (Deci & Ryan, 2013; Ryan & Deci, 2000). From a legal standpoint, drivers' autonomy is regarded as a fundamental principle of the platform business model.

While bundling has proven effective in enhancing the efficiency and profitability of food delivery platforms (Reyes et al., 2018b; Steever et al., 2019), increasing bundle size can lead to longer computational times and longer customer wait times. Consequently, previous studies often focus on bundles comprising two (Yildiz & Savelsbergh, 2019; Wang et al., 2021; Reyes et al., 2018b; Steever et al., 2019) or three orders (Li et al., 2021). In addition, in the literature, bundles are typically categorized as either composed of orders from the same restaurant (Reyes et al., 2018a), or orders from multiple restaurants (Steever et al., 2019). These studies indicate that bundling reduces the total distance traveled compared to Point-to-Point (P2P) delivery method. However, creating bundles is relatively straightforward when orders are all from the same restaurant

(Reyes et al., 2018a). Additionally, although Steever et al. (2019) considered bundles with orders from multiple restaurants, they assumed that these orders are part of a single customer's order.

The literature primarily employs mathematical programming or machine learning. Mathematical programming involves developing mathematical models and solution techniques to efficiently find solutions. Yildiz and Savelsbergh (2019) proposed a model using a column and row generation algorithm to reduce overall driver compensation costs. Steever et al. (2019) aimed to decrease delivery delays and increase early deliveries in an auction-based system. Reyes et al. (2018b) used a rolling-horizon algorithm to dynamically select the optimal order bundles over time.

The machine learning literature focuses on identifying bundle patterns from data. Li et al. (2021) used neural networks to generate order bundles, demonstrating high accuracy and efficiency. Wang et al. (2021) applied a decision tree model trained with Extreme Gradient Boosting (XGBoost) to optimize the assignment of bundles and orders in China's OFDP Meituan.

Our study differs in the following three aspects:

1. Despite the drawback of increased computational complexity, we consider bundles with sizes of both 2 and 3. Furthermore, we consider bundles consisting of orders from multiples restaurants for multiple customers.
2. We consider driver autonomy in the simulation experiments. While some studies acknowledge drivers' right to reject orders (Li et al., 2021; Ulmer et al., 2021), we assume that drivers have the autonomy to choose the orders or bundles that benefit them, rather than being obligated to accept the bundles assigned by the platform.
3. We introduce a machine learning-based system for efficient bundle calculation and analyze it using real-world data. Given that South Korea's OFDP market is among the most active in the world, it can provide insights not yet explored in the existing literature.

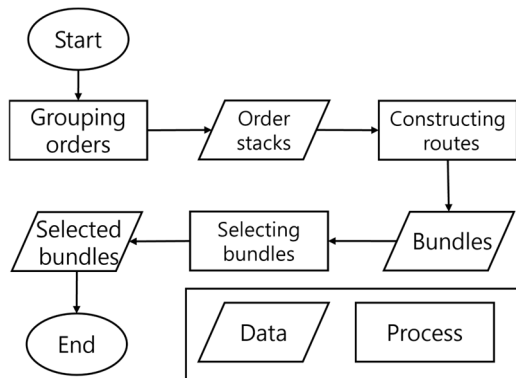
## 3. Bundle Construction and Selection System

We assume the food delivery platform with  $K$  drivers and  $R$  restaurants. Orders are received in real-time associated with a restaurant ( $o$ ), a customer ( $d$ ), and the time required to prepare the food ( $e$ ). For each order, the platform calculates the origin-destination time (ODT). ODT is the estimated travel time from the restaurant to the customer. The food lead time (FLT) is the estimated time between food pick-up and delivery to the customer. The lead time (LT) is the estimated delivery time from order placement to delivery completion.

On receipt of an order, the platform must decide whether

to publish the order immediately or calculate a bundle for it. Our policy is that an order can be included in a bundle if its FLT within a bundle does not exceed twice the ODT. As FLT increases, food quality often declines, causing customer dissatisfaction. Customers with shorter ODT—those closer to the restaurant—expect a smaller increase in FLT, while those with longer ODT are more tolerant of longer FLT. To balance these expectations and avoid overly limiting bundle creation, we set the FLT upper limit at twice the ODT. Bundle construction, which is the process of determining orders that should be delivered together as a bundle, involves exploring potential bundle candidates and calculating the associated delivery route along with the associated delivery time. Once bundle construction is complete, the platform selects bundles for drivers to choose from. Orders that cannot be delivered within 60 minutes are canceled (Ahn & Lee, 2018). The platform's objective is to maximize the total number of fulfilled deliveries.

Figure 1 outlines the bundle construction and selection system, consisting of three distinct processes: grouping orders, constructing route, selecting bundles.



**Figure 1:** Bundle Construction and Selection System.  
Source: Provided by Authors

**Grouping Orders:** To be a bundle, individual orders need to be grouped into an order stack of size 2 or 3. With  $n$  customer orders, the number of possible order stack is given by  $C_2^n + C_3^n \left( \frac{n(n-1)}{2!} + \frac{n(n-1)(n-2)}{3!} \right)$ . This process is crucial for addressing the computation issue, as considering all possible order stacks requires intensive computation.

**Constructing Routes:** For each order stack, the platform generates all possible routes, ensuring that the food lead time for each order remains within the threshold. Computation in this process is less critical because the number of possible routes of an order stack is limited due to the condition that restaurants must be visited before customers. For instance, in the case of an order stack comprising two orders,  $i$  and  $j$ , each with origin and

destination pairs  $(o_i, d_i)$  and  $(o_j, d_j)$ , there are four potential routes:  $o_i - o_j - d_i - d_j$ ,  $o_i - o_j - d_j - d_i$ ,  $o_j - o_i - d_i - d_j$ , and  $o_j - o_i - d_j - d_i$ . Similarly, for an order stack with three orders, the total number of possible routes is limited to 36.

**Selecting Bundles:** Publishing too many bundles can make it difficult for drivers to select orders and may decrease their autonomy in the selection process. Therefore, the platform selects the route that maximizes the saved travel distance when the orders are delivered as a bundle.

## 4. System Improvement

In the system discussed in Section 3, certain inefficiencies arise from the exhaustive consideration of all potential order stacks and the construction all valid routes for each order stack. This section presents improvements to the processes of grouping orders and constructing routes using an enhanced order grouping technique and a classification model (CM). The CM, which is known for its ability to capture domain-specific features and to adeptly classify data across various domains, the CM is structured as an ensemble and is trained using the XGBoost package. The enhanced order grouping technique uses parameters obtained from the CM to determine the search range for grouping orders. The route construction process includes an additional step: before constructing routes, the CM briefly classifies whether an order stack can form a bundle. Only the classified order stacks proceed to route construction.

In Section 4.1, we refine the grouping order process by focusing on relevant orders stacks and incorporating parameters extracted from the CM. This refinement results in a significant reduction in the number of order stacks to be considered, effectively addressing the inefficiencies and contributing to an improved bundle construction and selection system. In Section 4.2, we describe a classification model to determine whether an order stack qualifies as a bundle.

### 4.1. Enhanced Order Grouping Process

In this section, we enhance the grouping order process described in Section 3. Selectively focusing on the relevant order stacks associated with a pre-determined pivot order, we significantly decrease the number of order stacks requiring further process. Furthermore, we integrate parameters extracted from the classification model to augment the efficiency of the bundle construction system.

This process includes (i) the selection of a pivot order, denoted as  $i$ , and (ii) the identification of the set  $C_i$  of orders that can comprise order stack when associated with

the pivot order  $i$ . Selecting the pivot order depends on whether the bundle construction system is dynamic, static, or hybrid. The details of this are explained into Section 4.3. For now, we assume that we have a pivot order  $i$ . To be included in  $C_i$  an order must satisfy the following conditions: the distance from its restaurant to the restaurant of pivot order  $i$  is below a specified threshold, denoted as  $u_{origins}$ , and the distance from its customer to the customer of pivot order  $i$  is below a threshold,  $u_{destinations}$ . Additionally, the orders are sorted in a non-increasing order based on their distance between their customers and the customer of pivot order  $i$ . The size of  $C_i$  is restricted by a predetermined threshold, denoted as  $u_{count}$ .

The thresholds  $u_{origins}$  and  $u_{destinations}$  are extracted from the *if-then* rules within the decision tree of the classification model described in Section 4.2. Given that the decision tree branches are designed to maximize classification performance, the *if-then* rules contain the information to classify order stacks as bundle-able or not. Specifically, we extract all the values of the distances between restaurants (and customers) from the *if-then* rules and calculate their average as  $u_{origins}$  and  $u_{destinations}$ .

## 4.2. Classification Model

In this section, we introduce a classification model that determines whether an order stack can be bundled or not. The first step of the classification model is to select the relevant features. Since bundles are related with geographical characteristics, the selected features should include information that affects the bundle construction, including location, distance, area, and time.

(i) Location feature: This includes the location coordinates of the restaurants and the customers, as well as the difference in coordinates between them.

(ii) Distance feature: This includes the distance from the restaurant to the customer, the distance between restaurants, and the distance between customers.

(iii) Area feature: This represents the area enclosed by the restaurants or customers within an order stack. A smaller value indicates that orders are densely located, resulting in a short detour when bundled together. In essence, the area feature provides insight into the spatial arrangement of the order set, providing valuable information about the potential for efficient bundling.

(iv) Time feature: This includes the time of order placement.

From the perspective of CM training, bundle-able order stacks are very rare compared to non-bundle order stacks. Specifically, bundle-able order stacks account for only 1% for size 2 bundles and 0.1% for size 3 bundles. To address this imbalance, we use the Synthetic Minority Over-

sampling Technique (SMOTE) to generate more bundle-able order stacks data and balance the dataset.

The CM uses decision trees as the base learner and is trained by minimizing the gradient. The final model includes logistic regression and random forest classifier along with the decision tree model. During training, the best combination of hyperparameters (subsample, minimum child weight, learning rate, max depth, number of estimators) is chosen to enhance performance.

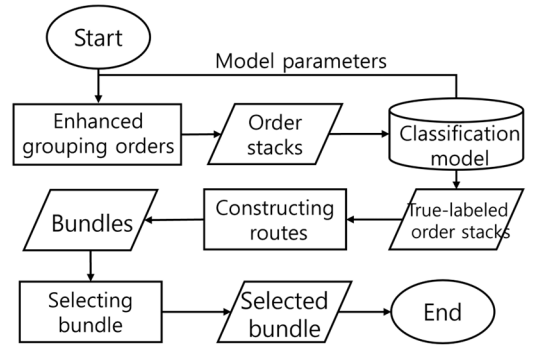
It is important to note that the CM has potential limitations, including misclassification, commonly referred to as false positives and false negatives. False positives are mitigated through route construction in the bundle system. Although false negatives are not directly addressed by the system, the CM is designed to classify order stacks during peak hours, when order volumes are very high, thereby minimizing the impact of false negatives.

## 4.3. Three Bundle Systems

In this section, we present three different bundle systems: Dynamic, Static, and Hybrid.

### 4.3.1. Dynamic System

In the dynamic system, each time a new order arises, it is considered as a pivot and the entire process, from grouping order to constructing bundle, is executed in real-time (See Figure 2). Since all bundles calculated in the dynamic system include the pivot order, only a single bundle can be selected. We select the bundle with the largest saved travel time.



**Figure 2:** Dynamic System. Source: Provided by Authors

This dynamic system ensures fast computation by focusing solely on orders proximate to the pivot (new order) and constructing potential routes for those orders. However, this limitation may hinder finding better bundles. Additionally, this greedy selection strategy favoring the largest-value bundle may lead to inefficient results. The Dynamic Vehicle Routing Problem (DVRP) also faces

similar challenges. In DVRP, delaying route construction to gather more information is a common strategy (Ferrucci, 2013).

#### 4.3.2. Static System

In the static system, we consider orders occurring during the time interval  $(t - 1, t]$  and proceed the bundle construction process at discrete time points  $t$ . As discussed in Section 3, the number of orders considered directly increases the number of order stacks. Therefore, techniques that can reduce the number of order stacks should be employed for the static system.

The goal of redefining pivot orders is to calculate the bundles that have a high probability of being selected by drivers at time  $t$ . We make two assumptions about the drivers: 1) To maximize profit, drivers will try to select the next order immediately after completing the current one, and 2) During the order selection process, drivers will only explore a certain number of orders. We call this number the “max exploring threshold”.

**Redefining Pivot Orders:** We list drivers in order of their completion times, starting with the earliest. For each driver, we arrange orders based on the distance between the driver's previous order completion location and the store of each order. From these, we select up to the max exploring threshold number of orders. This process is repeated for each driver. Orders selected in this manner may be chosen multiple times by different drivers. These selected orders undergo additional clustering based on spatial proximity. Initially, stores are clustered based on their locations using the K-Nearest Neighbor algorithm. Within each store cluster, orders are further clustered based on customer locations. Subsequently, for each cluster, we compute the centroid order, which is then redefined as a pivot order.

**Enhanced Order Grouping Process:** For each redefined pivot order, the static system employs the enhanced order grouping process described in Section 4.1 to generate order stacks. Note that the orders identified in the redefining pivot order step become pivots, and the other (non-pivot) orders can form a bundle of size 2 or 3 with the pivot order. Hence, each order stack generated in this step includes one redefined pivot. Subsequently, these clustered order stacks undergo the CM evaluation as described in Section 4.2.

Note that, in the static system, bundles generated through the enhanced order grouping process must include the redefined pivot order as a pivot. This requirement implies that the system might overlook better bundles composed solely of non-pivot orders. To address this, the static system explores neighborhoods to consider other bundles that were not considered in the enhanced order grouping process.

**Searching Neighborhoods:** During the redefinition of pivot orders, we identified pivot orders and their neighborhoods (orders belonging to the cluster of the pivot order). For each order stack with pivot order  $i$ , we select an order from pivot order  $i$ 's neighborhood set and replace the pivot order in the stack with that order. This creates a new order stack composed solely of non-pivot orders. Since the pivot order and its substitute are within the same cluster, they are spatially proximate. We repeat this process for every order belonging to the pivot order  $i$ 's neighborhood set. For these newly created order stacks, if the food lead time of each component does not exceed twice the ODT, then these order stacks are considered bundles. This straightforward approach efficiently generates good-quality bundles.

**Selecting Bundles:** Ensuring driver autonomy on the platform positively impacts driver loyalty. Therefore, when selecting bundles from generated order stacks, the platform should prioritize maximizing saved travel time without compromising driver autonomy. In the static system, only bundles without duplicate orders are published to facilitate drivers' decision-making. Displaying too many bundles on the driver's app screen can hinder their autonomous decision-making process. To calculate saved travel time, we compare how much travel time is reduced by each bundle compared to point-to-point deliveries.

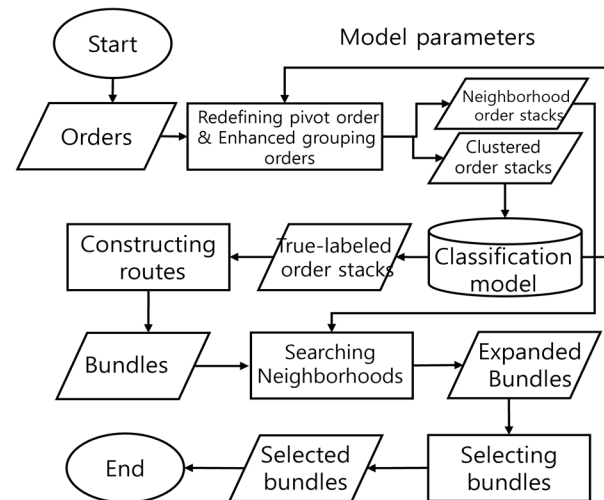


Figure 3: Static System. Source: Provided by Authors

Figure 3 outlines the static system. It begins with redefining the pivot order and then applies the enhanced order grouping process described in Section 4.1. This results in generation of clustered order stacks and neighborhood order stacks. The clustered order stacks undergo the CM evaluation, after which routes are constructed for these order

stacks. Meanwhile, the neighborhood order stacks undergo the neighborhood searching process, which will be explained below, and the bundles generated from neighborhood order stacks will be added to the bundle pool. In the final step, the static system proceeds to bundle selection.

#### 4.3.3. Hybrid System

The hybrid system combines both dynamic and static systems, leveraging the real-time responsiveness of the dynamic system alongside the accumulated information from the static system. The hybrid system consists of static and dynamic parts, each operating independently. Specifically, the dynamic part activates whenever a new customer order is received, while the static part runs periodically to handle all pending orders. During this process, a bundle published by one part may include orders that are already included in the bundle from the other part. In the case of overlapping orders in the bundles published by the static and dynamic parts, the bundle generated by the dynamic part takes priority, and the overlapping bundle from the static part is removed.

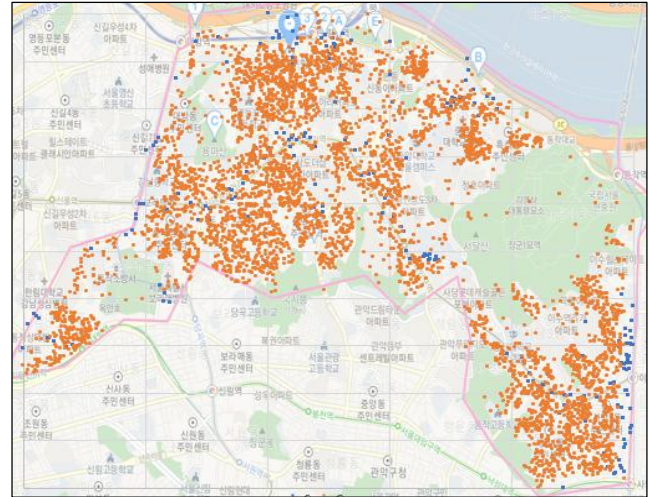
## 5. Experimental Results

### 5.1. Generation of Instance

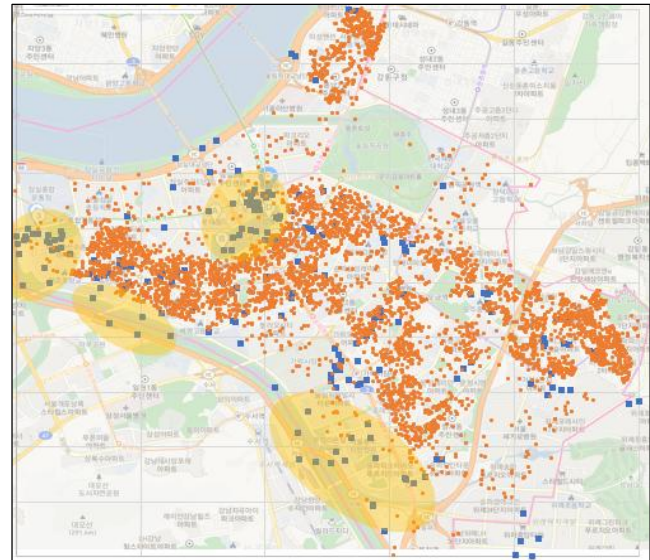
The experiment uses real data from Songpa (33.65 km<sup>2</sup>) and Dongjak (16.35 km<sup>2</sup>), each with 4800 orders from 230 restaurants. Figure 4 presents a screenshot from a commercial map service, integrated with data on restaurants and customers. As shown in Figure 4, Dongjak has approximately twice the number of restaurants and orders per unit area compared to Songpa, resulting in a higher overall order density. However, while Songpa has a relatively lower overall order density than Dongjak, it is characterized by large apartment complexes with very high order densities.

1. Order generation involves two sampling steps: Sampling the restaurants: From a public database of Seoul, 230 restaurants are sampled.

2. Sampling customer locations: From a public building database in Seoul, customer locations are sampled. Orders are generated over two-hour period using a Poisson process, which simulates peak times for lunch or dinner. Each restaurant receives an average of 20 orders per hour, with an average distance of 1.2 km between restaurants and customers.



(a) Dongjak



(b) Songpa

Source: Provided by Authors

**Figure 4:** Generated Instances of Dongjak and Songpa with Restaurants (marked in Blue) and Customers (in Orange).

Conditions and assumptions:

- Approximately 10% of all orders include time-sensitive menus. These sensitive orders can only be included in a bundle if their expected FLT is less than ODT plus 5 minutes. For other orders, they can only be included in a bundle if their FLT does not exceed twice the ODT.
- Drivers' speed is 16.8 km/h, adjusted for narrow and congested alleyways.
- Food pickup takes one minute, and delivery takes two minutes.

- The delivery fee is based on Euclidean distance: \$2.36 for 0-675 meters, \$2.75 for 675-1900 meters, and \$2.75 plus \$0.06 per 100 meters beyond 1900 meters.

The experiment assumes:

- The platform can monitor each driver's location and task in real time. Note that task could be a single order or a bundle of size 2 or 3.
- Drivers select the most profitable task; if a bundle is less profitable, the driver will choose a single order that offers higher profitability. This illustrates the autonomous decision-making of drivers in the experiments.
- Drivers can only perform one task at a time and cannot select more than two tasks simultaneously.
- Drivers select tasks from a list sorted in ascending order based on the distance between their current location and the start point of the task (the first restaurant in the bundle).
- For bundle tasks, drivers must follow the visit sequence provided by the platform.

## 5.2. Classification Model Performance

We trained the CM with order stack sizes of 2 and 3 for the Songpa and Dongjak instances. Table 1 demonstrates that each CM can classify whether a given order stack can form a bundle or not. This classification helps reduce the number of order stacks requiring route construction. The performance of the CM is measured using three metrics: Recall, Precision, and Fall-out. Recall measures the ratio of correctly classified order stacks, with high recall indicating accurate classification. Precision measures the ratio of correctly classified bundles among those identified as bundles by the CM, with high precision indicating high accuracy. Fall-out measures the proportion of non-bundle data misclassified as bundles. A low fall-out value in Table 1 indicates the accurate classification of non-bundles, thereby achieving the goal of reducing the number of routes constructed.

Note that the following performance metrics are averages from 30 simulations. The experimental environment includes an i7-6700 CPU with a clock speed of 3.40 GHz and 16 GB of RAM, and the experiments were coded using Python 3.6.

**Table 1:** Classification Model Performance

		Measure		
		Recall	Precision	Fall-out
Classification model	B2-Songpa	0.93	0.90	0.003
	B3-Songpa	1	1	0
	B2-Dongjak	0.97	0.92	0.008
	B3-Dongjak	0.99	0.99	0.00004

To analyze the performance of our dynamic, static, and hybrid models designed using the CM model, we compare their runtime with that of the enumeration system described in Section 3. This enumeration system constructs routes for all order stacks obtained from the enhanced order grouping process, without using a CM. The enumeration and static systems are invoked every 5 minutes, the dynamic system every 0.67 seconds (simulating 40 orders per minute), and the hybrid system every 5 minutes and upon receiving new orders.

Table 2 compares computation times for the enumeration, dynamic, static, and hybrid systems in Dongjak and Songpa. The dynamic, static, and hybrid systems demonstrate shorter computation times compared to the enumeration system. Specifically, the static system shows a shorter Time per run than the enumeration system with the same running interval, attributed to the CM reducing the number of order stacks requiring route construction.

**Table 2:** Run Time of Each System

Instance	Variation	Time per Run(sec)	Run Interval(sec)
Songpa	Enumerate	84.41	300
	Static	15.37	300
	Dynamic	0.24	0.67
	Hybrid	0.30	0.67
Dongjak	Enumerate	359.45	300
	Static	36.15	300
	Dynamic	0.23	0.67
	Hybrid	0.33	0.67

In Dongjak, the Time per run for the enumeration system exceeds its running interval, indicating it cannot complete computations before the next run. In Songpa, the Time per run for the enumeration system is more than four times longer, suggesting performance variability.

The CM significantly reduces the Time per run in the dynamic, static, and hybrid systems. Conversely, the enumeration system, where CM is not used, exhibits significantly longer Time per compared to the other systems. Specifically, in the Dongjak instance, the enumeration system's Time per run even exceeds the run interval, indicating its impracticality.

In the dynamic and hybrid systems, the Songpa and Dongjak instances show minimal differences in Time per run, indicating that the computational load in these systems does not significantly favor one instance over the other. However, in the static system, Songpa takes more than twice as long time per run compared to Dongjak. This difference may be attributed to higher overall order density in Dongjak, which results in a greater number of candidate order stacks compared to Songpa. Nevertheless, even with this difference, the computation time in the static system remains under 40 seconds, making it more practical than the enumeration

system, which takes approximately 6 minutes to complete its tasks.

### 5.3. Performance of Bundle Systems from the Perspectives of Platform Stakeholders

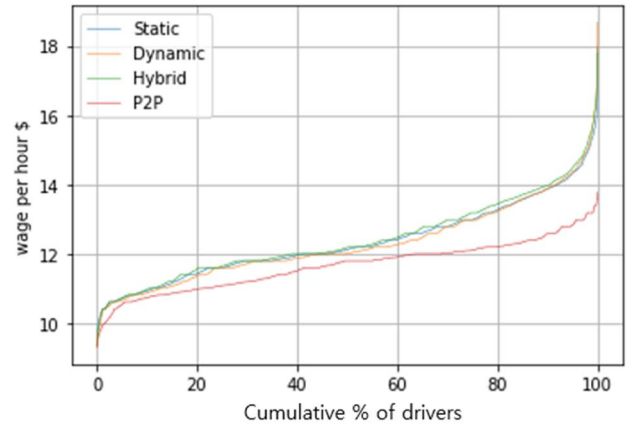
In this section, we analyze whether the dynamic, static, and hybrid bundle systems improve various performance metrics of the existing delivery platform from the perspectives of different stakeholders. For each stakeholder, we introduce appropriate metrics and examine how these metrics change with the bundle system. For drivers, we assess how average earnings have changed with the introduction of bundling and evaluate the fairness of their experience by examining the relationship between travel time and compensation. For the platform, we evaluate profit and customer service ratio. Finally, for customers, we analyze improvements in LT (Lead Time) and FLT (Food Lead Time).

As a benchmark for comparing the performance of the bundle systems, we consider the Point-to-Point (P2P) delivery system, which delivers one customer order at a time, and compare it with the three bundle systems.

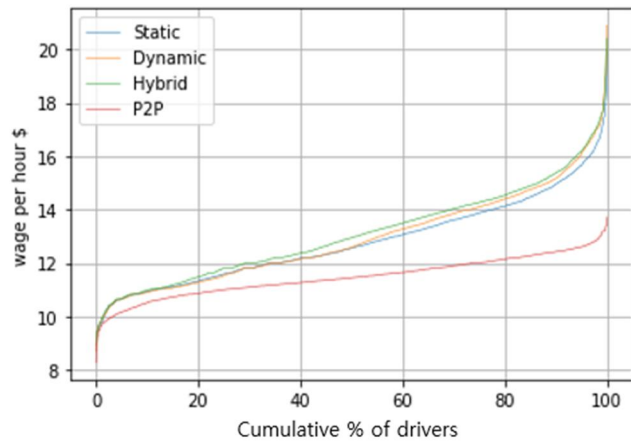
**Drivers:** Assuming all drivers have the same hourly driving cost, we analyze how their earnings improve across dynamic, static, and hybrid systems compared to the P2P method. Table 3 details drivers' average earnings and their deviations from the P2P method. In Songpa, drivers' average earnings increased in all bundle systems, reflecting an increase of 5.55% to 5.88%. In Dongjak, drivers' average earnings increased by 11.99%, 13.21%, and 12.09% across all systems compared to the P2P method, which is twice of Songpa. Notably, the differences between instances (Songpa vs. Dongjak) are more pronounced than those between different systems. These variations are due to geographical characteristics of the two areas. Since drivers' earnings depend on the distance they travel, this suggests that in Songpa, orders from large apartment complexes are bundled together, resulting in very short travel distances for drivers.

**Table 3:** Average Earning of Drivers

Instance	Type	Average Earnings per Hour (\$/h)	Increase in Average Earnings Compared to P2P (%)
Songpa	Static	12.32	5.73%
	Dynamic	12.29	5.55%
	Hybrid	12.33	5.88%
	P2P	11.65	-
Dongjak	Static	12.77	11.99%
	Dynamic	12.91	13.21%
	Hybrid	12.78	12.09%
	P2P	11.40	-



(a) Songpa



(b) Dongjak

**Figure 5:** Cumulative Frequency Graph of the Drivers' Earnings per Hour. Source: Provided by Authors

The introduction of bundles significantly impacts average earnings for most drivers. In Songpa, the percentage of drivers earning over \$12 per hour through the P2P approach increases from 35% to 60% with bundles (see Figure 5 (a)). Similarly, in Dongjak, this percentage rises from 30% to 70% (Figure 5 (b)). Given that \$12 exceeds South Korea's minimum wage, bundles have the potential to attract more drivers to the platform. Furthermore, bundles enable drivers to increase their order processing capacity, allowing them to earn \$16 or more per hour. This demonstrates how bundles not only enhance earnings for most drivers but also support higher income levels beyond basic wage thresholds.

Fairness is a crucial factor that we consider because driver satisfaction is closely tied to the fairness of the platform (Griesbach et al., 2019). For example, ride-hailing platforms such as Uber are regarded as fair when drivers' earnings correspond proportionally to their active work hours (Sühr et al., 2019). Similarly, in food delivery



platforms, fairness revolves around maintaining a balance between drivers' labor input and their earnings. We evaluate platform fairness by analyzing the regression coefficient between drivers' total traveled distances and earnings.

**Table 4:** Slopes of the Linear Regression Equation for Earnings Based on the Moving Distance

Instance	Type	Slope
Songpa	Static	19.45**
	Dynamic	18.44**
	Hybrid	25.75**
	P2P	-39.04**
Dongjak	Static	78.04**
	Dynamic	87.61**
	Hybrid	92.91**
	P2P	-26.47**

Note: \*\*:p-value < 0.01.

Table 4 presents the linear regression coefficients of earnings for total traveled distance. In the P2P approach, a negative coefficient indicates that drivers' earnings decrease as the distance between the restaurant and the customer increases, incentivizing drivers to prefer shorter-distance orders for higher earnings. This phenomenon may explain longer waiting times for long-distance orders in practice. Conversely, in the static, dynamic, and hybrid systems with bundles, the regression coefficients are positive. This indicates a direct relationship between traveled distance and earnings: drivers can earn more as they undertake longer trips compared to P2P. This highlights how bundles promote fairness by rewarding drivers for their efforts in fulfilling longer-distance orders.

**Platform:** The platform's service rate and revenue are critical metrics for evaluating bundle systems, which help assess the overall financial performance and effectiveness of the bundle systems. Assuming the platform's profit is calculated as 2% of the delivery fees paid by the customers, we obtain the following results.

**Table 5:** Platform-related Performance Metrics

Instance	System	Service Rate (%)	Increase in Service Ratio Compared to P2P (%)	Platform Profit (\$)
Songpa	Static	77.42%	3.93%	207.28
	Dynamic	77.23%	3.74%	206.46
	Hybrid	77.91%	4.42%	208.60
	P2P	73.49%		195.76
Dongjak	Static	75.07%	6.84%	214.83
	Dynamic	75.63%	7.40%	216.89
	Hybrid	76.69%	8.46%	219.87
	P2P	68.23%		191.65

As shown in Table 5, the service rate in Songpa is slightly higher than in Dongjak, which can be attributed to the bundling of orders from large apartment complexes in Songpa. In contrast, the increase in service rate with bundle systems compared to P2P in Dongjak is about twice as much as that in Songpa, indicating a more pronounced effect of the bundle systems in Dongjak.

Table 5 also shows the platform profit during the 2-hour lunch period. When calculating the platform's profit over the course of a day and a month, it is evident that using bundle systems results in a significantly higher increase in profit compared to P2P. Furthermore, given the recent trend of rising delivery fees, if the platform's profit were calculated as 5% of the delivery fee instead of 2%, the impact would be even greater.

The cost of implementing the bundle system is minimal, primarily covering server computation expenses. In contrast, the bundle systems lead to a significant increase in earnings for both drivers and the platform. This brief cost-benefit analysis clearly highlights the advantages of the bundle systems from the perspective of platform profitability.

Bundles provide a higher service rate and platform profit compared to P2P, which contributes to the long-term growth of the online food delivery platform market and enhances its network effect. Positive service experiences are known to foster customer loyalty (Frank, 2007). As more bundles can be efficiently formed and delivered, this attracts more drivers to the platform, creating a positive feedback loop beneficial for both customers and drivers. Additionally, considering customer concerns about current delivery fees, the platform might contemplate reducing these fees based on the increased profits. In the long term, this strategy could contribute to expanding the overall market for food delivery services.

**Customers:** Customer satisfaction is assessed by examining LT and FLT. Table 6 presents the average ODT, the average LT, average FLT, and the increases in LT and FLT compared to P2P for each system in Songpa and Dongjak.

The increase in LT(Lead Time) and FLT(Food Lead Time) compared to P2P for the three bundle systems are less than 0.6 minutes and 0.7 minutes respectively in Songpa, and less than 1.53 minutes and 1.38 minutes respectively in Dongjak, indicating a negligible impact on customer satisfaction. In Table 6, Dongjak saw a longer average ODT compared to Songpa, resulting in longer lead times and food lead times in Dongjak. This suggests that the same bundling policy may affect customer experience differently in Songpa and Dongjak.

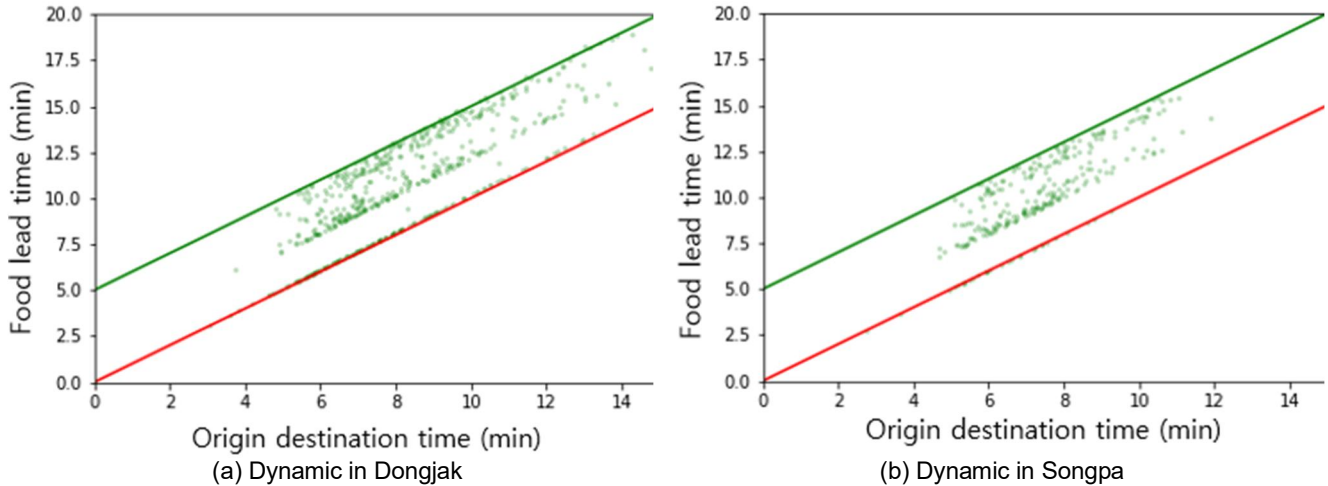
In our experimental setup, approximately 10% of orders

are considered FLT-sensitive, requiring FLT to be shorter than ODT (Origin Destination Time) + 5 minutes. Figure 6 illustrates a scatter plot of ODT and FLT for these orders. The red line represents  $FLT = ODT$ , and the green line represents  $FLT = ODT + 5$  minutes. All orders fall between

these two lines, indicating that FLT remains within acceptable limits. This demonstrates the effectiveness of the bundle systems even for sensitive orders, highlighting their ability to maintain satisfactory service levels.

**Table 6:** Customer-related Performance Metrics

Instance	System	ODT (min)	LT (min)	FLT (min)	Increase in LT Compared to P2P (min)	Increase in FLT Compared to P2P (min)
Songpa	Static	3.36	14.88	4.00	0.60	0.69
	Dynamic	3.40	14.57	3.87	0.29	0.56
	Hybrid	3.39	14.76	4.01	0.48	0.70
	P2P	3.31	14.28	3.31	-	-
Dongjak	Static	4.22	18.02	5.54	1.53	1.38
	Dynamic	4.32	17.27	5.34	0.78	1.18
	Hybrid	4.26	17.62	5.54	1.13	1.38
	P2P	4.16	16.49	4.16	-	-



**Figure 6:** Food Lead Time Scatter Plot of the Time-sensitive Customer Orders. Source: Provided by Authors

## 6. Conclusion

In this study, we introduce bundle systems designed with a classification model to efficiently compute bundles in food delivery platforms. We developed dynamic, static, and hybrid bundle systems and compared their performance against an enumeration approach. In the experiments assuming peak hours, these three systems demonstrated shorter computation times compared to the enumeration system. We conducted a comprehensive business analysis from the perspective of each stakeholder in the food delivery platform. Our findings indicate that bundling has a positive impact on both customers and drivers compared to the P2P model. Specifically, it increases drivers' income, expands service to more customers, and maintains comparable lead times to standalone deliveries. Additionally, the platform benefits from higher service completion rates, increased

commissions, and improved fairness for drivers. The bundle systems are cost-effective and beneficial for all platform participants, underscoring the value of our approach. In fact, several food delivery platforms have already adopted bundle strategies, validating the feasibility of our proposed bundle systems.

However, this study has several limitations. First, because our bundle system uses mixed integer programming, low computation times are not guaranteed. Future research should explore heuristics or alternative machine learning methods for improving bundle model. Second, although our study uses real-world data from Seoul to validate the bundle systems, the geographic and temporal scope of the experiments is somewhat limited. Future research should broaden the dataset to include multiple regions and varied timeframes to provide a more comprehensive evaluation of the system's effectiveness. Third, by assuming uniform

driver capabilities and experiences, our model does not reflect the actual diversity among drivers. Additionally, the impact of external factors such as weather conditions, traffic congestion, and seasonal demand variations has not been thoroughly explored. Future research should focus on developing a more comprehensive model that better reflects these real-world conditions. Integrating these factors would provide a more robust and realistic assessment of the bundle system. Furthermore, incorporating feedback from stakeholders could enhance the bundle delivery system, making this an important area for future research.

## References

- Ahn, Y., & Lee, C. (2018). The effect of delivery waiting time in a pizza delivery restaurant on customer satisfaction and repurchase intention. *Culinary Science & Hospitality Research*, 24(5), 131-144.
- Chakravarty, A., Kumar, A., & Grewal, R. (2014). Customer orientation structure for internet-based business-to-business platform firms. *Journal of Marketing*, 78(5), 1-23. <https://doi.org/10.1509/jm.12.0442>
- Deci, E. L., & Ryan, R. M. (2013). *Intrinsic motivation and self-determination in human behavior*. New York, USA: Springer. <https://doi.org/10.1007/978-1-4899-2271-7>
- Ferrucci, F. (2013). *Pro-active dynamic vehicle routing: real-time control and request-forecasting approaches to improve customer service*. Heidelberg, Germany: Physica Berlin. <https://doi.org/10.1007/978-3-642-33472-6>
- Frank, J. (2007). Meat as a bad habit: A case for positive feedback in consumption preferences leading to lock-in. *Review of Social Economy*, 65(3), 319-348. <https://doi.org/10.1080/00346760701635833>
- Grand View Research. (2022). *Online food delivery market size, share and trends analysis report by type, by region, and segment forecasts, 2023–2030*. Market Analysis Report, Report ID: GVR-4-68039-942-2, Horizon. Link: <https://www.grandviewresearch.com/industry-analysis/online-food-delivery-market-report>
- Griesbach, K., Reich, A., Elliott-Negri, L., & Milkman, R. (2019). Algorithmic control in platform food delivery work. *Socius*, 5, <https://doi.org/10.1177/2378023119870041>
- Li, C., Zhu, L., Fu, G., Du, L., Zhao, C., Ma, T., ... & Lee, P. (2021). Learning to Bundle Proactively for On-Demand Meal Delivery. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (pp. 3898-3905). November 1-5, Queensland, Australia. <https://doi.org/10.1145/3459637.3481931>
- Reyes, D., Erera, A. L., & Savelsbergh, M. W. (2018a). Complexity of routing problems with release dates and deadlines. *European journal of operational research*, 266(1), 29-34. <https://doi.org/10.1016/j.ejor.2017.09.020>
- Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., & O'Neil, R. (2018b). The meal delivery routing problem. *Optimization Online*. <https://optimization-online.org/?p=15139>
- Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1), 68. <https://doi.org/10.1037/0003-066X.55.1.68>
- Shapiro, A. (2018). Between autonomy and control: Strategies of arbitrage in the “on-demand” economy. *New Media & Society*, 20(8), 2954-2971. <https://doi.org/10.1177/1461444817738236>
- Steever, Z., Karwan, M., & Murray, C. (2019). Dynamic courier routing for a food delivery service. *Computers & Operations Research*, 107, 173-188. <https://doi.org/10.1016/j.cor.2019.03.008>
- Sühr, T., Biega, A. J., Zehlike, M., Gummadi, K. P., & Chakraborty, A. (2019). Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3082-3092). August 4-8, Anchorage, USA. <https://doi.org/10.1145/3292500.3330793>
- Tilly, C., & Tilly, C. (1998). *Work under capitalism: New perspectives in sociology*. Boulder, Colorado: Westview Press.
- Ulmer, M. W., Thomas, B. W., Campbell, A. M., & Woyak, N. (2021). The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science*, 55(1), 75-100. <https://doi.org/10.1287/trsc.2020.1000>
- Wang, X., Wang, L., Wang, S., Yu, Y., Chen, J. F., & Zheng, J. (2021). Solving Online Food Delivery Problem via an Effective Hybrid Algorithm with Intelligent Batching Strategy. *Proceedings of the 17th International Conference on Intelligent Computing Theories and Application, Part II* (pp. 340-354). August 12–15, Shenzhen, China. [https://doi.org/10.1007/978-3-030-84529-2\\_29](https://doi.org/10.1007/978-3-030-84529-2_29)
- Yildiz, B., & Savelsbergh, M. (2019). Provably high-quality solutions for the meal delivery routing problem. *Transportation Science*, 53(5), 1372-1388. <https://doi.org/10.1287/trsc.2018.0887>