

Research Article

Optimization of Rider Scheduling for a Food Delivery Service in O2O Business

Guiqin Xue,¹ Zheng Wang ,¹ and Guan Wang^{1,2}

¹School of Maritime Economics and Management, Dalian Maritime University, Dalian, China

²Technical University Bergakademie Freiberg, Akademiestraße, Freiberg 6,09599, Saxony, Germany

Correspondence should be addressed to Zheng Wang; drwz@dlut.edu.cn

Received 25 February 2021; Revised 6 April 2021; Accepted 1 May 2021; Published 25 May 2021

Academic Editor: David Rey

Copyright © 2021 Guiqin Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Services such as Meituan and Uber Eats have revolutionized the way the customer can find and order from restaurants. Numerous independent restaurants are competing for orders placed by customers via online food ordering platforms. Ordering takeout food on smartphone apps has become more and more prevalent in recent years. There are some operational challenges that takeout food service providers have to deal with, e.g., customer demand fluctuates over time and region. In this sense, the service providers sometimes ignore the fact that some riders may be idle in several periods in regions, while, in contrast, there may be a shortage of riders in other situations. In order to address this problem, we introduce a two-stage model to optimize scheduling of riders for instant food deliveries. A service provider platform expectantly schedules the least quantity of riders to deliver within expected arrival time to satisfy customer demand in different regions and time periods. We introduce a two-stage model that adopts the method of mixed-integer programming (MIP), characterize relevant aspects of the scenario, and propose an optimization algorithm for scheduling riders. We also divide the delivery service region and time into smaller parts in terms of granularity. The large neighborhood search algorithm is validated through numerical experiments and is shown to meet the design objectives. Furthermore, this study reveals that the optimization of rider resource is beneficial to reduce overall cost of the delivery. Takeout food service platforms decide scheduling shifts (start time and duration) of the riders to achieve a service level target at minimum cost. Additional sensitivity analyses, such as the tightness of the order time windows associated with the orders and riders' familiarity with delivery regions, are also discussed

1. Introduction

Ordering takeout food on smartphone apps is becoming more and more popular in China in the past several years. In a recent survey, 68% of diners order takeout food at least once a day, with 33% order at least once a week [1]. The takeout delivery business is regarded as an online-to-offline (O2O) business model applied by the food industry. In the past few years, Chinese consumers have widely accepted this model. Consumers could get rid of the hassle of going to restaurants but can get more food choices and faster services through online ordering mode. In China, online takeout food service platforms such as Meituan and Baidu provide takeout ordering services to most of the working population. Firstly, customers make orders on smartphone apps, meals

are then prepared and packaged by restaurants, and the takeout delivery platform assigns independent contracting riders to deliver the food. Compared with other foodservice providers such as McDonald's and KFC, online food ordering platforms do not hire a full-time employee to do the delivery job. Thus, a key issue faced by all takeout delivery platforms is how to formulate a timetable and route for riders to deliver food so that customers can get food as quickly as possible within the desired time [2]. Besides, taking into account the fluctuations in customer demand with regions and time periods, the scheduling of takeaway platforms will become more complicated. For example, the order arrival rate usually fluctuates more during the peak dining period (soaring during the meal times and dropping to deficient levels in other mealtimes) in the business district

than in the suburbs [3, 4]. Furthermore, to prevent or mitigate any adverse effects of the uncertainty associated with riders' delivery capacity, the service platform may also choose to have a scheduled delivery workforce that they can utilize more effectively.

Takeout service providers always try their best to fulfill customer requirements in the delivery process. Customers usually expect reliable and fast service within the expected time [5]. Therefore, in this study, we developed and evaluated a new approach to address order assignment and riders' service sequence. For the sake of simplicity, the rider starts from the central depot to the restaurant in the relevant pickup region. Generally, riders provide delivery service for customers within a radius of 3 km. Customers always place orders before each deadline, such as dividing the day into equidistant service periods, 10:00–12:00, 12:00–14:00, 15:00–17:00, and 17:00–19:00. We assume that a restaurant receives an order at 10:30 with the deadline as 11:30. In this case, the service provider observes that the service region does not have adequate rider resources, so the order is not assigned to the available rider to serve as fast as possible. The number of riders required varies in different periods in each region, and different regions have different order requests in the same period. Therefore, the delay of customer requests seems inevitable with the deadline. Furthermore, the delayed service also leads to lower service quality and more customer complaints [6]. Hence, a service platform needs to design reasonable scheduling plan to allocate riders' resources for instant food deliveries with respect to the fluctuations of customer demand over time and space. As shown in Figure 1, colors from light to dark indicate customers' ordering demand at the same ordering time. Meanwhile, it can be seen that the greater the customers demand near the central region, the smaller the demand in the edge region.

Due to the practical challenges, instant logistics is facing many difficulties, while developing rapidly. Foodservice providers face the complex problem as to control the cost of scheduling riders, while maintaining high quality of customers' service. That is, takeout service providers struggle to efficiently assign orders to riders for instant food deliveries. The issue is an inherent contradiction between the undulated distribution of order time, space locations, and available riders' stability. Several regions will have excess idle riders/riders without delivery tasks during the trough, resulting in a waste of resources. However, during peak dining periods, the order volume shows an increasing trend relative to nondining periods, and existing riders cannot satisfy the delivery demands of customers. This phenomenon is called insufficient rider resources. For example, in a day, the two peak periods of ordering takeout are usually 12:00 to 13:00 and the dinner is 17:00 to 19:00, respectively. Moreover, the number of existing riders assigned by the service platform in regions is insufficient to meet the growing customer requests. In this case, this phenomenon always causes many orders which are not delivered by the available rider resources for instant deliveries with fluctuation of customer requirements over

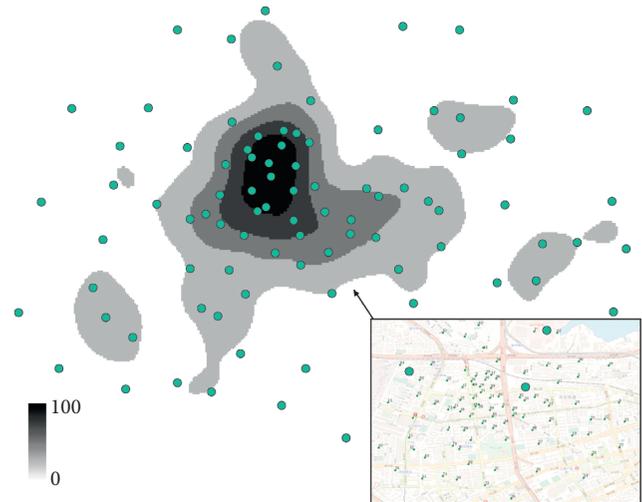


FIGURE 1: The fluctuation of customer requirements over time and space.

time and space. However, in the nonmeal periods of a day, some riders do not receive orders from the service provider. There is excess capacity in this case. The expected delivery time of orders is concise to a certain extent (the expected time refers to the customer placing the order to delivering the order). Customers have more stringent requirements than before the previous delivery system on speed, safety, and service quality. Therefore, the freshness of meals in the instant logistics delivery is becoming more significant than the traditional transportation [6].

Different types of ordering takeout have requirements for distribution conditions and ways. Hence, they also increase the difficulty of distribution processes for riders. Besides, riders will flexibly adjust their routes according to their own experience and real-time road conditions. In doing this, the rider route usually deviates from the actual routes, and thus, this phenomenon delays the customer's delivery service. A two-stage programming model is formulated to address the challenges mentioned above. In the first stage of the model, the rider's fixed cost and the rider's travel cost are considered, and the second stage of the model is to dispatch the rider to work in a specific area during a certain period. The study intends to provide a reasonable solution heuristic to optimize the rider resources of service providers. We use the time series analysis method to generate the initial route scheme according to the order quantity transmitted to the service provider. The consideration of the available riders in different periods and regions highlights the importance of punctuality [7]. Therefore, the objective of the first-stage is to minimize the total cost, including fixed cost and traveling cost of riders in the subregion periods. The objective of the second-stage is to minimize the rider scheduling times. The ordering takeout usually combines customer requests into a big order pool associated with the ordering time, space location, food category, and freshness to deal with the order requests. Specifically, a service provider adopts a way to assign orders and then schedules riders to deliver the

distribution tasks in each period and subregion. This way usually refers to distribution regions.

The rest of the paper is organized as follows. In Section 2, we review the related literature. We present the problem formally in Section 3 and formulate a multi-objective integer programming model and its related definitions in Section 3. Section 4 details the solution to the proposed problem. We use the proposed algorithm to perform the case study, of which the data comes from a takeout platform in China. Section 6 details the conclusion of this study and prospects for future research opportunities.

2. Literature Review

The delivery service providers deliver goods to customers over a given planning period [8]. They estimate the importance of transport capacity in the service regions to assign riders to deliver orders by the deadlines [5, 9]. To gain a competitive advantage, the instant delivery company tries the best to offer increasingly narrower delivery deadlines [10]. However, the trend towards shorter delivery lead-times increases transport resources cost for the instant food delivery of customer requirements over period and space, since the service provider has to recruit more riders [11] or seeks the assistance of the third logistics, such as crowdsourcing [12–16] or spare social transport capacity [2], to release the transport resources pressure. Hence, transport capacity management is significant for the instant delivery providers. [17–19]. Besides, Zhang et al. [20] analyzed the influence of the pickers' learning effects in the online integrated order picking and delivery application. They also consider the pickers' learning effect essential for the order fulfillment process's accuracy and predictability. However, the riders will be passive entities that are subject to a digital "panopticon" [7], as the effect is used to tame algorithms. Decisioners providing services need to maintain a balance by optimizing existing capacity resources, that is, the balance between existing rider resources and service quality. In the case of rider resources and no additional riders are rented, the order requests of customers are delivered by scheduling riders from other regions, while also ensuring the quality of customer service.

The problem of instant delivery is closely related to the pickup and delivery problem [21] and the dial-a-ride problem [22]. Each order has a pickup node and delivery node, respectively. Hence, the customer requires to be serviced by the rider within an hour or less. For example, all riders must deliver the orders to customers within 45 minutes to take out the delivery. Moreover, the rider must start from the depot, visit some new pickup nodes that come up as a result of new delivery nodes, and then return to the depot within the given time windows [23]. Otherwise, they may be subject to additional penalties due to delivery delays [24]. Usually, by using precise algorithms, we may work out the optimal solution, and by using the heuristic method, we may get a near-optimal solution to the delivery problem. [25] discussed the rider consistency in the dial-a-ride problem and proposed a branch-and-cut algorithm to solve two mathematical formulations. To the best of our knowledge, by exact methods, we may obtain the optimal solution. However, the computational time is too long for solving middle-size problems. Therefore, the exact

algorithm is unreasonable to address the proposed problem, because the high timeliness of instant delivery. Therefore, in the real-world scenario, the heuristics approach is adopted. Ghiani [26] introduced the order assignment and vehicle scheduling in the dynamic dial-a-ride problem. Li (2016) proposed a large neighborhood search algorithm to address the share-a-ride problem and determined the time slack. Klapp [27] formulated an arc-based integer programming model, and they also designed local search heuristics to solve a dynamic dispatch waves problem where the order arrival times are known. Gschwind [28] and Azi [29] developed a large neighborhood search heuristic to optimize the pickup and delivery problem.

Gu [30] presented the benefits of the truck-drone combination associated with the ordering takeout delivery of two advanced ant colony heuristics and a method to minimize the number of dispatched vehicles and the total travel time. Besides, the same-day delivery is another subtopic of research worthy of attention. When it comes to instant delivery, goods are usually delivered within the same-day [6, 31]. These delivery activities always occur in the form of online behavior [32]. It is worth noting that the pickup node and delivery node are usually located in the same city. Ulmer [33, 34] analyzed the drones combined with regular delivery vehicles to improve same-day delivery performance based on geographical districting. Klapp [35] discussed the request acceptance and distribution decisions in same-day delivery. They split the time horizon into four-periods. Bent [36] applied the multiple-scenario approach (MSA) to sample a set of scenarios for obtaining future customer requests, and the samples can determine a suitable route. Furthermore, in Table 1, we compare existing research works with this study to show the main differences.

The differences between this study and previous research are as follows. Firstly, we formulate a two-stage model to address the problem of configuring transport resources of delivery riders and optimize both quantity and travel routes of the riders for fluctuations of customer demand over time and space in any period of the entire region. Secondly, we design an LNS approach to generate the delivery rider's route given the determined minimum number of delivery riders in the first stage of our model. Then, the second stage of our model is addressed by the Gurobi 9.1, a commercial solver. Thirdly, the service region and the whole day are divided into some subregions and subperiods.

Our contributions are as follows. Firstly, we present the optimization of rider scheduling for a food delivery service (ORSFDS) problem based on a dataset from an online-to-offline takeout food service provider in China. Then, we formulate two MIP models and propose the LNS algorithm to generate the near-optimal solution. After that, an empirical study and extended computation are carried out on data provided by Dalian, China. Finally, this study includes the sensitivity analysis on the tightness of order time windows and the rider's familiarity with the region.

3. Problem Definition and Formulation

In this section, we first present a formal description of the ORSFDS. We then define the rider routing and rider scheduling. Suppose a set of customers $D = \{1, \dots, d\}$ that need to be

TABLE 1: Comparison between the existing literature and this study.

Reference	Transport capacity	Rider assignment	Rider scheduling	Two-stage model	Heuristic algorithm
Ulmer et al. [33]	✓		✓		
So et al. [11]	✓		✓		
Berbeglia [21]	✓		✓		✓
Cordeau et al. [22]			✓		✓
Li et al. [37]			✓		✓
Archetti [8]		✓	✓		✓
Huang et al. [5]	✓	✓			✓
Steever et al. [38]			✓		✓
Sun [7]	✓		✓		✓
Wang [39]	✓		✓		✓
Yildiz et al. [4]	✓		✓		✓
Liao [9]			✓	✓	✓
This study	✓	✓	✓	✓	✓

served by riders during time periods in regions, and they are located in a connected graph, $G = (V, E)$, where $V = \{0\} \cup N$ is the set of vertices, E is the set of edges and $N = P \cup D = \{1, \dots, n\}$ is a group of nodes including restaurants and customers. The node 0 represents the rider's origin, and nodes 1 to n represent, respectively, the locations of restaurants and customers. For each order $i = (o_i, d_i)$, we use O_s to denote the set of order sub-regions. In this study, we divide the distribution region into several sub-regions and the rider service horizon into small time periods throughout the day.

We present the fluctuation of customer demand with time in a certain service region in Figure 2. The example is depicted in Figure 2. The figure shows the fluctuation of customer demand with time in a specific service region. We discretize the working horizon of a day for riders and take every three hours as a period for riders to deliver meals. It can be seen from Figure 2 that there is excess capacity at $T1$ and $T3$ between the relationship of rider's supply and customer's demand. Excess capacity refers to customers' low demand at $T1$ and $T3$, resulting in some idle riders having no orders to deliver. However, at $T2$ and $T4$, there is insufficient capacity to meet customers' demands. Insufficient transport capacity refers to the surge of customer demand in the region at $T2$ and $T4$, and the existing riders cannot complete the delivery of orders. Similarly, this phenomenon also exists in other regions. Additionally, to make rational use of the existing resources, this study mainly considers integrating the existing regional riders' resources in a certain period to minimize the scheduling cost and meet the needs of customers and service quality. Therefore, we develop a two-stage scheduling model to optimize the rider scheduling problem.

In practice, however, due to the sharp increase of order demand in the peak period (11:00–13:00 and 17:00–19:00), a service provider (e.g., Meituan) usually assigns more riders to deliver orders within the expected time. To assign resources more reasonably and thus attain low operation costs, we have developed a large neighborhood search to solve the problem. In our study, in order to alleviate the large amount of orders in the business district, the existing rider resources cannot satisfy customers' order requests. Thus, we propose to schedule some idle riders from a subregion with a small order volume to another subregion with a large order volume to satisfy the needs of customers in a timely manner.

Throughout this study, two decisions are to be made in this problem: the number of routes of riders and the service periods and subregions of riders in the first stage of the model. For the number of riders and the traveling routes of riders' decisions, we use $x_{r0j} = 1$ to indicate that riders deliver directly from 0 to j , that is, riders start from the origin to work; and, $x_{rij} = 1$ indicates that the rider travels from i to j in the route, and $x_{r0j} = 0$ and $x_{rij} = 0$; otherwise, $i, j = 0, 1, \dots, n$. We also use $x_{rst} = 1$ to denote that the rider r provides delivery service for customer requests in the period t in the subregion s during the working horizon. For the sake of brevity, we formulate a two-stage model to optimize the rider scheduling problem. The rider scheduling model is to determine the delivery route and number of riders in each subregion during a time period and to determine the working period and working region of a rider.

In practice, we consider several reasonable assumptions to guarantee model availability. (1) For each order, it has a certain pickup quantity. Riders load the customer's ordered meals at the pickup node and unload them at the delivery node. (2) The pickup node is visited by the rider exactly once later than the delivery node. Significantly, the pickup node and the delivery node must be accessed by the same rider. (3) Each vehicle begins service at an initial position. As riders are independent contractors, each rider determines how assigned orders are sequenced and routed. (4) These routing and sequencing procedures are known to the dispatchers. Since the order requests are known to the service providers in advance, dispatchers and riders communicate via mobile phones. (5) Riders accept all orders assigned during the order horizon. The related notations and definitions used to formulate the ORSFDS are shown in Table 2.

$$\min : \sum_{r \in R_s} \sum_{j \in N \cup \{0\}} \gamma_1 x_{r0j} + \sum_{r \in R_s} \sum_{i \in N} \sum_{j \in N} \gamma_2 d_{ij} x_{rij}, \quad (1)$$

$$y_{ro_i} = y_{rd_i}, \quad \forall r \in R_s, i \in O_s, \quad (2)$$

$$\sum_{r \in R_s} y_{ro_i} = 1, \quad \forall i \in O_s, \quad (3)$$

$$t_{ij} + \alpha_{ir} \leq \alpha_{jr} + M(1 - x_{rij}), \quad \forall r \in R_s, i, j \in N \cup \{0\}, \quad (4)$$

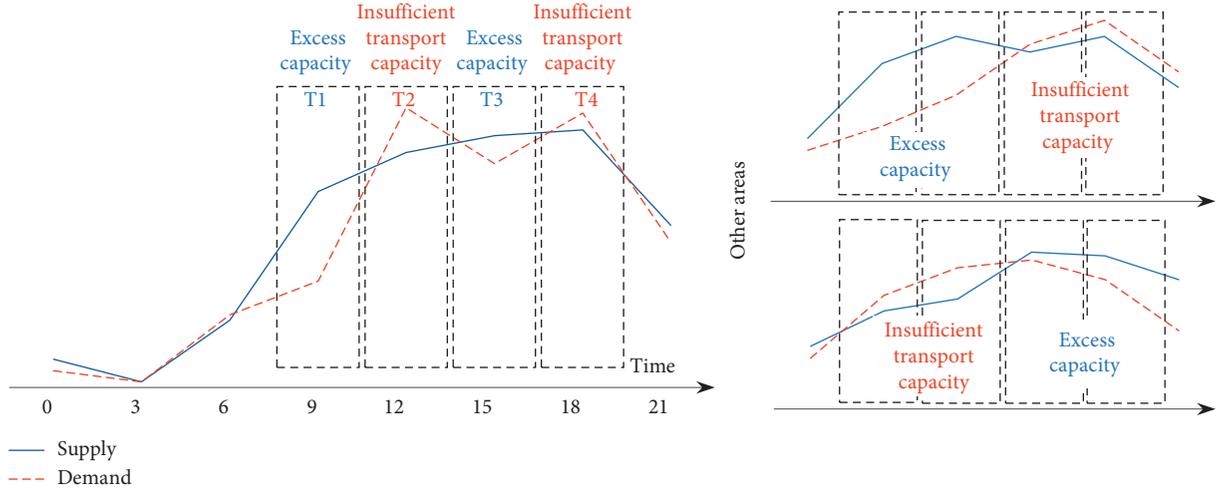


FIGURE 2: The fluctuation of customer demand with time in a certain service region.

TABLE 2: Sets, parameters, and decision variables.

Notation	Description
<i>Sets</i>	
R_s	Set of assigned riders in subregion
O_s	Set of orders in subregion, for any order i , it consists of two nodes, denoted by (o_i, d_i) .
D	Set of customers
S	Set of sub-regions serviced by riders, $S = 1, \dots, s$
$N = P \cup D \cup O$	Set of all nodes including pickup nodes (restaurant's locations) and delivery nodes (customer's locations) and the depot (the starting node of the rider)
<i>Parameters</i>	
a_i, b_i	The order placed time by customer i and the expected delivery time by customer i , respectively
u_{ir}	The load quantity of rider r arriving at node i
α_{ir}	Departure time of rider r at node i
q_i	The demand at node i
c_r	The capacity of the vehicle owned by the rider r
t_{ij}	Each arc (i, j) is associated with a travel time
d_{ij}	Distance of the arc (i, j) , unit: meter
γ_1	Fixed costs for each rider, unit: CNY
γ_2	The traveling cost, unit: CNY
<i>Decision variables</i>	
x_{rij}	Binary decision variable equals 1 if the rider r travels from node i to node j ; 0 otherwise.
x_{rst}	Binary decision variable equals 1 if rider r is assigned to deliver food delivery by the service platforms in the period t in the subregion s
y_{ir}	Binary decision variable equal 1 if the rider r services node i

$$\alpha_{ir} \leq b_i, \quad i \in D, \quad (5)$$

$$\alpha_{ir} \geq a_i, \quad \forall i \in P, \quad (6)$$

$$\alpha_{o_i r} + t_{o_i d_i} \leq \alpha_{d_i r}, \quad \forall i \in O_s, \quad r \in R_s, \quad (7)$$

$$x_{rd_i o_i} = 0, \quad \forall i \in O_s, \quad r \in R_s, \quad (8)$$

$$u_{ir} - q_i \leq u_{jr} + M(1 - x_{rij}), \quad \forall i \in D, \quad j \in N \cup \{0\}, \quad r \in R_s, \quad (9)$$

$$u_{ir} + q_i \leq u_{jr} + M(1 - x_{rij}), \quad \forall i \in P, \quad j \in N \cup \{0\}, \quad r \in R_s, \quad (10)$$

$$0 \leq u_{ir} \leq C, \quad \forall i \in N, \quad r \in R_s, \quad (11)$$

$$\sum_{i \in N} \sum_{j \in N} x_{rij} = \sum_{i \in N} y_{ri}, \quad \forall r \in R_s, \quad (12)$$

$$\sum_{r \in R_s} \sum_{j \in N} x_{rij} = 1, \quad \forall i \in N, \quad = \sum_{i \in N} y_{ri}, \quad \forall r \in R_s, \quad (13)$$

$$\sum_{r \in R_s} \sum_{i \in N} x_{rij} = 1, \quad \forall j \in N, \quad (14)$$

$$\sum_{r \in R_s} \sum_{j \in N} x_{r0j} = \sum_{r \in R_s} \sum_{j \in N} x_{r0j} \leq |R_s|, \quad (15)$$

$$\sum_{j \in N} x_{r0j} = \sum_{j \in N} x_{rj0} \leq 1, \quad \forall r \in R_s, \quad (16)$$

$$\alpha_{ir} > 0, \quad \forall i \in N, r \in R_s, \quad (17)$$

$$x_{rij} \in \{0, 1\}, \quad \forall r \in R_s, i, j \in N \cup \{0\}, i \neq j, \quad (18)$$

$$y_{ri} \in \{0, 1\}, \quad \forall i \in N, r \in R_s, \quad (19)$$

The objective function (1) seeks to minimize the total costs, including fixed cost and riders' travel cost [40]. Constraints (2) and (3) ensure that each order in the subregion is serviced exactly once by the same rider from the pickup node to the delivery node. Constraint (4) satisfies the travel time requirement between two adjacent nodes. Constraint (5) guarantees that customer requests are delivered before the order expected delivery time. The rider arrives at the restaurant to pickup food or leaves the restaurant to deliver food after placing the order is presented in constraint (6). Constraint (7) presents the time relationship between the same rider visiting the pickup node and delivery node; That is, the delivery time when the rider leaves the delivery node cannot be earlier than the total time. It is determined by the pickup time and traveling time from the pickup node to the delivery node. The arc does not travel directly from the pickup to the delivery node, as in Constraint (8). Payload capacities are presented in Constraints (9)-(11). Constraints (9) and (10) define the payload capacity of the vehicle owned by a rider. Constraint (11) determines the payload does not exceed the capacity of the vehicle. Constraint (12) shows the relationship between the node and the rider's ownership. Constraints (13) and (14) ensure the connectedness of the order nodes. Similarly, constraints (15) and (16) indicate that each rider starts from the depot, goes to the restaurant to pickup food, delivers the customer request, and returns to the same depot. Constraints (17)-(19) are decision variable definitions.

Riders' shifts are crucial for the optimization of the scheduling of riders. However, in the real-world scenario, the number of riders required is different in each period and each subregion. For example, at two peaks throughout the day, the number of order requests increased almost exponentially. In this case, the number of riders required for each period in the same region varies over time. Therefore, we propose a new way of scheduling riders. In other words, we schedule idle riders from several subregions to other subregions with less transport capacity in the same period so that the problem attaining the case of short riders in some subregions during the peak period could be solved.

For the sake of simplicity, in the second stage of the model, we suppose a set of subregions that need to be serviced by riders during the period. In our discussion, we assume a set of periods $T = \{t_1, t_2, \dots, t_{|T|}\}$, and the payload capacity of

riders at any subperiod in any subregion is denoted by q_{st} . There are $|R|$ available riders in the distribution, denoted by $R = \{1, 2, \dots, r\}$. We assume that each rider can only serve familiar subregions $S_r \subseteq S, S_r \neq \emptyset$. Because of the fluctuations of order requests in different subregions and periods, riders who may transfer among subregions are denoted by $s \in S$ to another $s' \in S$. Each arc is associated with a transfer cost by $c_{ss'}$. Riders are not allowed to serve subregions far from their original region, and the rider has the maximum transfer distance L_{\max} . For the rider's shift decision, the decision variable is denoted by $x_{rst} = 1$, and the rider r provides delivery service for customer requests at a period t in the subregion s . To satisfy the customer requirements and obtain the minimum scheduling times, the model is as follows.

$$\min = \sum_{r \in R} \sum_{s \in S} \sum_{t \in T} x_{rst}, \quad (20)$$

$$\sum_{s \in S} x_{rst} \leq 1, \quad \forall r \in R, t \in T, \quad (21)$$

$$\sum_{r \in R} x_{rst} \geq q_{st}, \quad \forall s \in S, t \in T, \quad (22)$$

$$\sum_{s \in S_r} x_{rst} = 0, \quad \forall r \in R, t \in T, \quad (23)$$

$$x_{rst} + x_{rs'(t+1)} \leq 2, \quad \forall r \in R, s, s' \in S, t \in T, c_{ss'} \leq L_{\max}, \quad (24)$$

$$x_{rst} + x_{rs'(t+1)} \leq 1, \quad \forall r \in R, s, s' \in S, t \in T, c_{ss'} > L_{\max}, \quad (25)$$

$$x_{rst} \in \{0, 1\}, \quad \forall r \in R, s \in S, t \in T. \quad (26)$$

The objective function (20) minimizes the overall scheduling time of riders [16]. Constraints (21) ensure that each rider can serve at most one subregion at a period. Constraints (22) indicate that the number of riders required by each sub-region in each period is not greater than the total number of assigned riders. Constraints (23) state that each rider has a fixed familiar region and cannot go to the unfamiliar sub-regions. Constraints (24) and (25) govern the transfer distance relationship for riders. Finally, Constraints (26) denote the decision variable.

4. Solution Heuristic

This section introduces the proposed large neighborhood search algorithm in detail. There are some metaheuristic algorithms that may have several drawbacks such as premature convergence and to tramp in local optimal or stagnation. To overcome these disadvantages, this study attempts to develop a hybrid heuristic algorithm that enables it to acquire high computational efficiency. Furthermore, heuristics and metaheuristics are widely applied for different research domains, such as online learning, scheduling, multiobjective optimization, medicine, passenger and freight terminal operations, data classification, and others.

As far as we know, the proposed rider scheduling model is an extension of the dial-a-ride problem, a well-known NP-hard problem [41]. Therefore, it is crucial to obtain high-

quality optimization results in a reasonable calculation time. A large neighborhood search algorithm is a widely verified resource scheduling algorithm [42]. The constructed initial solution improves the existing solution by iteratively executing the destroy and repair operators and generates an approximately optimal solution.

In the rider scheduling model, we optimize both the number of riders and the route of riders. Firstly, we design an insertion heuristic method to construct feasible initial solutions quickly; in the subsequent iterations, the existing solutions are destroyed by the removal operator, and then the removed orders are inserted by the repair operator. After this operation, a new neighborhood solution can be generated from the current solution. When the termination condition is satisfied, the algorithm outputs the optimal solution. Table 3 shows the pseudo-code for designing a large neighborhood algorithm.

4.1. Insertion Heuristic to Design Initial Solution. Cordeau et al. [41] constructed an initial solution using the completely stochastic method. However, the downside of this method is sometimes the infeasible solution may also be obtained. Therefore, in the proposed, we consider the time window of orders and the proximity of orders to generate a rider's feasible route quickly. First, all orders are sorted ascendingly based on the expected arrival time of the orders; then, each order is assigned to the rider's delivery route to generate $|R|$ initial sequences according to expected arrival time (from early to late). Finally, we merge the orders into the same delivery route by taking care of the four metrics listed below:

- (1) Calculate the distance between two pickup nodes.
- (2) Calculate the distance between two delivery nodes.
- (3) Calculate the distance between a delivery node and a pickup node.
- (4) Calculate the distance between a pickup node and a delivery node.

Delivery route merging is performed iteratively. In each iteration, an order is randomly selected from the remaining uninsured orders, one of the four metrics is randomly selected, the minimum distance based on these two random features is then calculated, and the order is inserted into the route with the minimum distance. During the insertion process, we also need to do a feasibility check according to the constraints of the order assignment's model to ensure the feasibility of the initial solution. A feasible initial scheduling plan is obtained until all orders are inserted into the route.

4.2. Neighborhood Search. In this section, to obtain a new neighborhood solution, the removal operation and repair operation are designed by destroying and reconstructing the current solution. In the removal operation, we use four removal operators: natural sequence removal operator, maximum time window removal operator, worst order removal operator, and cluster removal operator. As with each removal operation, the output result is only a part of the solution. Therefore, we

TABLE 3: The pseudo code of the heuristic algorithm.

Input: The order dataset
Output: The numbers of the riders and the rider's route
Step 1: generate initial route s_{ini} by using insertion heuristic
Step 2: set initial solution s_{ini} as the current solution, neighborhood solution s_{nei} , and best solution s_{best}
Step 3: implement remove operation on current solution s_{cur} to obtain partial solution s_{part}
Step 4: use repair operator to obtain neighborhood solution s_{nei} from partial solution s_{part}
Step 5: if s_{nei} outperforms s_{best} , accept the inferior solution according to the principle of simulated annealing
Step 6: check if the termination condition is fulfilled, that is, after N consecutive iterations, the value of S_best no longer changes. Then, jump to step 6 if yes, otherwise to step 3
Step 7: output the optimal solution

implement repair operators to construct a new feasible solution from partial solutions. In this study, useful repair operators are the optimal insertion operator, regret insertion operator, and maximum waiting time insertion operator.

4.2.1. Remove Operator. The purpose of the neighborhood search is to remove some orders from the current solution based on specific criteria. There are four neighborhood operations applied in this study as follows.

Natural sequence removal operator: the node in the route with "0" rider load is referred to as the natural sequence. In general, the rider route consists of two natural sequences, respectively. That is, sequence $1 \rightarrow 2 \rightarrow 1+n \rightarrow 3 \rightarrow 2+n \rightarrow 3+n$ and sequence $4 \rightarrow 5 \rightarrow 5+n \rightarrow 4+n$, namely. Randomly selects a natural sequence from the removal operator and removes it from the current rider route.

Maximum time window removal operator: select the order with the largest time window in all routes and remove them because orders with larger time windows have more time flexibility.

Worst order removal operator: for the rider route, the removed route's cost-saving value calculates one by one and removes the order with the most considerable saving value.

Cluster removal operator: randomly select a root node and threshold, and then remove the order on each route closest to the root node.

4.2.2. Repair Operator. The purpose of the repair operation is to reinsert the removed orders into the rider route. The repair operator performs iteratively in the repair process until all orders insert into a set of routes. In this study, we use the following three kinds of repair operators.

Optimal insertion operator: this way is similar to the insertion heuristics operation in section 4.2, in which the unserviceable orders insert to the location of the minimum cost increment.

Regret insertion operator: considering the optimal cost value insertion, the most challenging order usually puts to the final processing. Therefore, for all unserviceable orders, the optimal insertion location's cost value is calculated and generated by the cost value vector. Then, randomly select a

value from the vector and insert its corresponding order into the route.

The maximum waiting time insertion operator is to insert orders into the rider route with a more considerable buffer time. The waiting time refers to the time difference $\Delta t_{wait} = \alpha_{ir} - t_i, \forall i \in P$, between the rider's arrival time and the pickup node's opening time.

4.3. Selection and Stopping Criterion. The optimal and regret insertion is used to generate the initial feasible solution based on its current number. At each iteration of the algorithm, we randomly select removal and repair operators to destruct and reconstruct the current solution and obtain a new solution. Moreover, the simulated annealing criterion is used to determine if the new solution is accepted. Therefore, by adopting this criterion, we can accept a suboptimal solution for the time being to search for a universally optimal solution rather than a locally optimal solution.

5. Computational Experiments and Analysis

5.1. Benchmark Example Test. The LNS algorithm is implemented in C# and tested on a system with 64bit Windows 10 OS, Intel i7 processor (2.60 Hz), and 16 GB RAM. Then, we use the Gurobi 9.1.0 commercial solver to verify our model, but it can only handle less than ten orders. Therefore, we need to adopt the proposed heuristic to solve medium-sized problems.

The DARP has been studied for decades, associated with many benchmark instances that have been given. The standard instance provided by Cordeau [22] is used to verify the proposed mathematical model and heuristic method.

The difference between the proposed algorithm and DARP is that the pickup node and delivery node have only a one-sided time window. Therefore, the standard DARP instance can quickly be transformed into a test instance based on the first-stage of the model. We can set the time window on the right side of the pickup node and the delivery node's left side as positive infinity and 0, respectively.

We have compare the tabu search with the proposed LNS algorithm, and the results are shown in Table 4. The first column shows the instance name. The second column enumerates the order requests. Inset A, the quantity of customer requests is one unit. However, set B is subject to the uniform distribution. The third column is the capacity of the rider's vehicle. Column 4 indicates the minimal number of riders obtained from the model. Columns 5 to 6 and 7 to 8 denote the tabu search and LNS algorithm's solution and computational time, respectively.

According to Table 4, the proposed algorithm and tabu search method are two implementations of the first stage of the model. The performance of the proposed algorithm is better than the tabu search algorithm. We could obtain an optimal solution for 16 of the 24 instances by using the LNS algorithm; however, by employing the TS algorithm, we could only get an optimal solution of 13 instances. However, in terms of the computational time, the LNS algorithm is slightly longer than the tabu search algorithm. Moreover, the

TABLE 4: Summary of results on different scale instances.

Ins.	q_i	Q	The optimal solution	TS		LNS	
				Solution	Time	Solution	Time
a2-16	1	3	2	2	10	2	17
a2-20	1	3	2	2	16	2	16
a2-24	1	3	2	2	20	2	29
a3-30	1	3	3	3	26	3	30
a3-36	1	3	3	3	28	3	32
a4-40	1	3	4	4	32	4	34
a4-48	1	3	4	5	35	5	43
b3-24	U[1, 6]	6	3	3	18	3	25
b3-30	U[1, 6]	6	3	3	24	3	30
b3-36	U[1, 6]	6	3	3	27	3	29
b4-16	U[1, 6]	6	4	4	10	4	12
b4-32	U[1, 6]	6	4	4	26	4	30
b4-48	U[1, 6]	6	4	4	35	4	41
b5-40	U[1, 6]	6	5	6	32	5	32
b5-50	U[1, 6]	6	5	6	38	5	40
b5-60	U[1, 6]	6	5	5	42	6	50
b6-48	U[1, 6]	6	6	7	34	7	39
b6-60	U[1, 6]	6	6	7	40	7	49
b6-72	U[1, 6]	6	6	8	44	7	45
b7-56	U[1, 6]	6	7	8	41	7	44
b7-70	U[1, 6]	6	7	9	46	8	46
b7-84	U[1, 6]	6	7	9	55	8	60
b8-64	U[1, 6]	6	8	9	43	8	52
b8-80	U[1, 6]	6	8	9	53	9	68
b8-96	U[1, 6]	6	8	9	73	9	82

LNS algorithm is stochastic in nature, and their performance often fluctuates, so for test instance (a) and instance (b), we run the proposed algorithm for 1000 iterations. The convergence processes are shown in Figure 3; although the convergence rate of the LNS algorithm is slow, the solution approach has good stability. It should be noted that instance (a) and instance (b) are the benchmark examples, whose convergence is shown in, namely, Figures 3(a) and 3(b). In the real-world scenario, the time difference between the two algorithms can be ignored. Therefore, the LNS algorithm in this study can better solve the first stage of the model.

5.2. Data Source. This section is about ride optimization scheduling problems for on-demand food deliveries with a computational experiment from Dalian, China. Section 4.2.1 analyzes the delivery company's order data in half a month to illustrate customer demand fluctuations in time and space. In Section 4.2.2, this study carries out a specific calculation test to optimize riders' transport capacity. Section 4.3 discusses sections of discusses window tightness and riders' familiarity with distribution regions based on capacity assignment.

5.2.1. Description Of Real-World Cases. In this section, numerical experiments are performed based on data from an online-to-offline food delivery platform located in Dalian, China. To demonstrate the proposed algorithm's effectiveness in the configuration of transportation resources of delivery riders for on-demand food deliveries with the

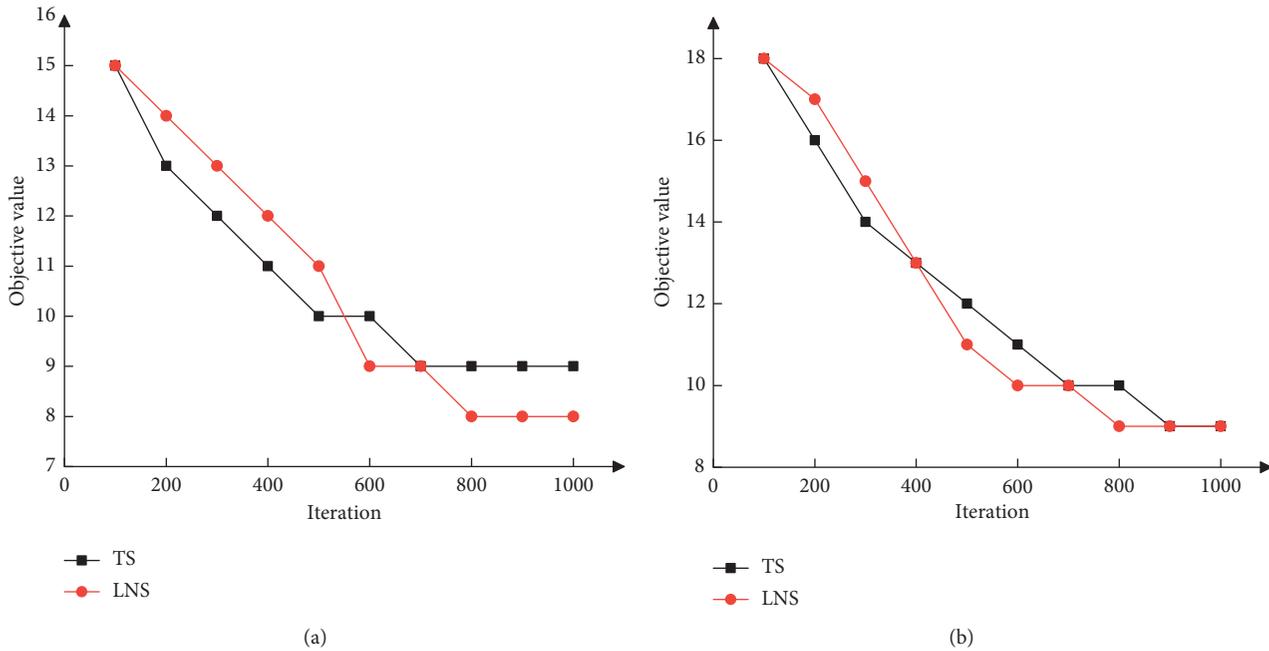


FIGURE 3: The convergence processes for instances. (a) Instance b8-64 (b) Instance b8-96.

fluctuations of customer demand over time and space, we take the customer order requests of a takeout food service provider from January 1 to 15, 2017, as the data source. Data sources include order IDs, restaurant names, customer locations, rider names, order times, delivery times, navigation distances, order regions, and rider regions. Then, we use the *Python* pandas package to process the data and get all order requests from the first 15 days of Jan 2017, as shown in Figure 4. It can be observed that the average number of daily orders is more than 5000.

Furthermore, the business districts with more than 100 orders on January 1, 2017, are selected as our research regions. Then we get 23 sub-regions, and their detailed information is shown in Figure 5. There are very few orders taken before 8:00 AM and 20:00. The peak hours are 11:00–12:00, 14:00–15:00, and 16:00–17:00. As a result, we focus on the delivery period from 8:00–20:00. We divide the whole day into six sub-periods, and each sub-period is 2 hours.

To analyze the order requests imbalance among the different sub-periods and sub-regions. We select four business districts: Luosifu, Xi'an Road, Wanda, and Shandong Road, to depict the order variations in Figure 6. We could see that the number of orders made by customers varies in regards to both time and region. Therefore, the takeout food delivery platform needs reasonably schedule riders to deliver customer requests and maintain the transportation capacity at a suitable level.

5.2.2. Optimization of Rider Scheduling. This section selects 46 orders from 10:00 to 12:00 in the Luosifu business district to determine the optimal number of delivery riders

required. The coordinates of the rider resource based on the Baidu Map are shown in Figure 7. No. 0 stands for the depot. The order pickup nodes are from No. 1 to No. 46, and the corresponding delivery nodes are from No. 47 to No. 92. The demand for each order is 1 unit, and the capacity of the rider is six units. The fixed cost of the rider is 150 CNY.

The proposed LNS algorithm is implemented in C# and run on an environment with Intel(R) Core (TM) i7-9750H CPU, 2.60 GHz, and 16 GB of RAM. The minimum number of riders is 13 in the above instance. The routes of riders are shown in Table 5 and Figure 8. Therefore, the first stage of the proposed model and the LNS heuristic are effective for the takeout food service provider in reducing the number of delivery drives required in the given period.

Moreover, in Table 6, we present the minimal number of riders obtained by the proposed heuristic and the number of dispatched riders in an original dataset in the 23 subregions from 10:00 to 12:00. The results show that, in 20 out of 23 subregions, our method outperforms the current system in use. Therefore, we conclude that the proposed model and LNS algorithm are superior to the current assignment method. Furthermore, we optimize the number of riders in the six time periods and 23 subregions. We assume that the transfer distance between two adjacent subregions is 3 km. The maximal allowed transfer distance for each rider is 9 km between two consecutive subperiods. The Gurobi 9.1 solver is used to generate the optimal solution for the second stage of the model. The corresponding results are presented in Table 6.

Table 7 presents the minimal number of riders required of the 23 subregions in the six subperiods. However, the instant delivery network needs 300 riders, and all riders schedule 1800 times. However, in the real scenario, a

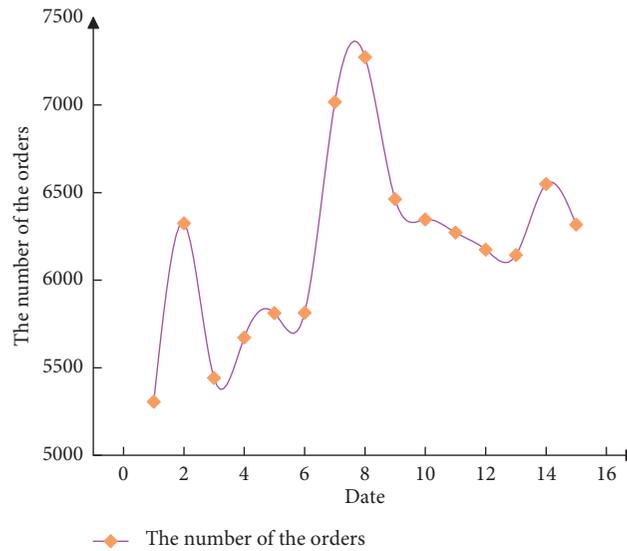


FIGURE 4: The average daily orders in the first half of January 2017 based on the historical dataset.

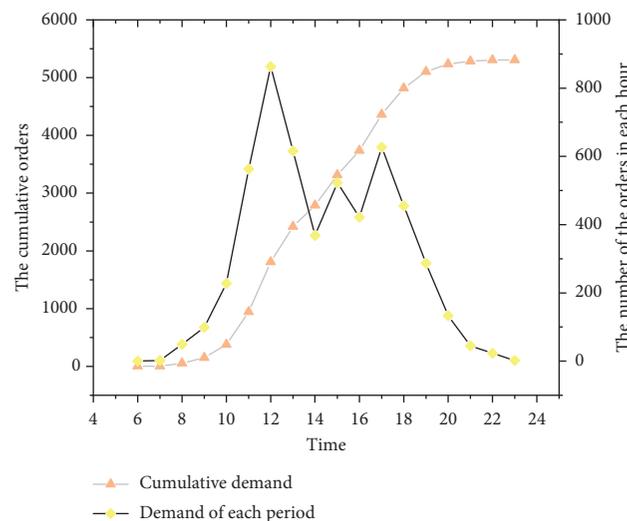


FIGURE 5: The statistics chart of the orders in January 2017.

service provider assigns 328 riders and schedules 2180 times to deliver customer requests. In order to address the fact that existing riders cannot satisfy the customer's ordering demands during the peak meal period, in our study, we proposed to schedule idle riders from some subregions to other subregions with high rider's demand. In this way, the total number of riders has been reduced by 8% compared to the previous scheduling scheme (each subregion has sufficient riders), and the cost of scheduling riders has also been saved by 4200 CNY. Hence, good capacity optimization is practical and economical for service providers. Randomly select ten riders and discuss the rider's changes in the service area in subperiods, and the results are shown in Table 8. The results show that the number of riders required in different time periods in different subregions is different. This is due to the fluctuation of orders in the same time period in different regions and different

time periods in the same region. For example, No. 1 assigns a rider to deliver orders in 6 different subregions. Hence, configuring riders' transport resources for instant deliveries with the imbalance of customer requirements over time and space is extremely critical.

5.3. Sensitivity Analysis. In this section, we conduct sensitivity analysis on the tightness of the order time windows and regional familiarity.

5.3.1. Analysis on the Tightness of the Order Time Windows. Punctuality is crucial to on-demand food delivery businesses since the freshness of meals must be preserved within the scheduled arrival time to meet the customers' expectations. Besides, by introducing an optimal delivery riders schedule,

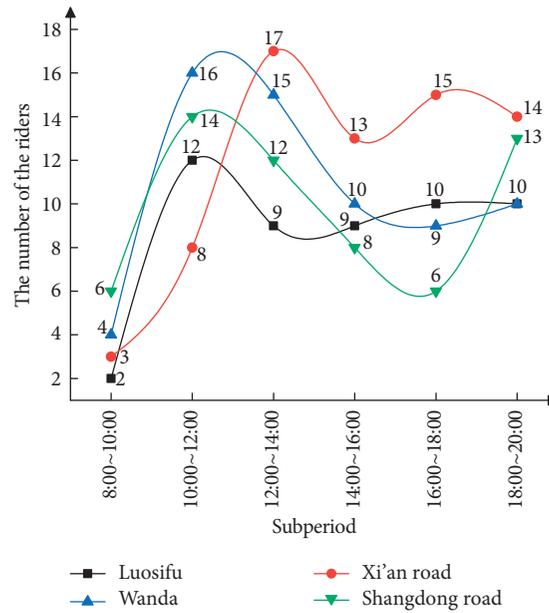


FIGURE 6: Statistical chart of transport capacity demand in each subperiod of the business circle.

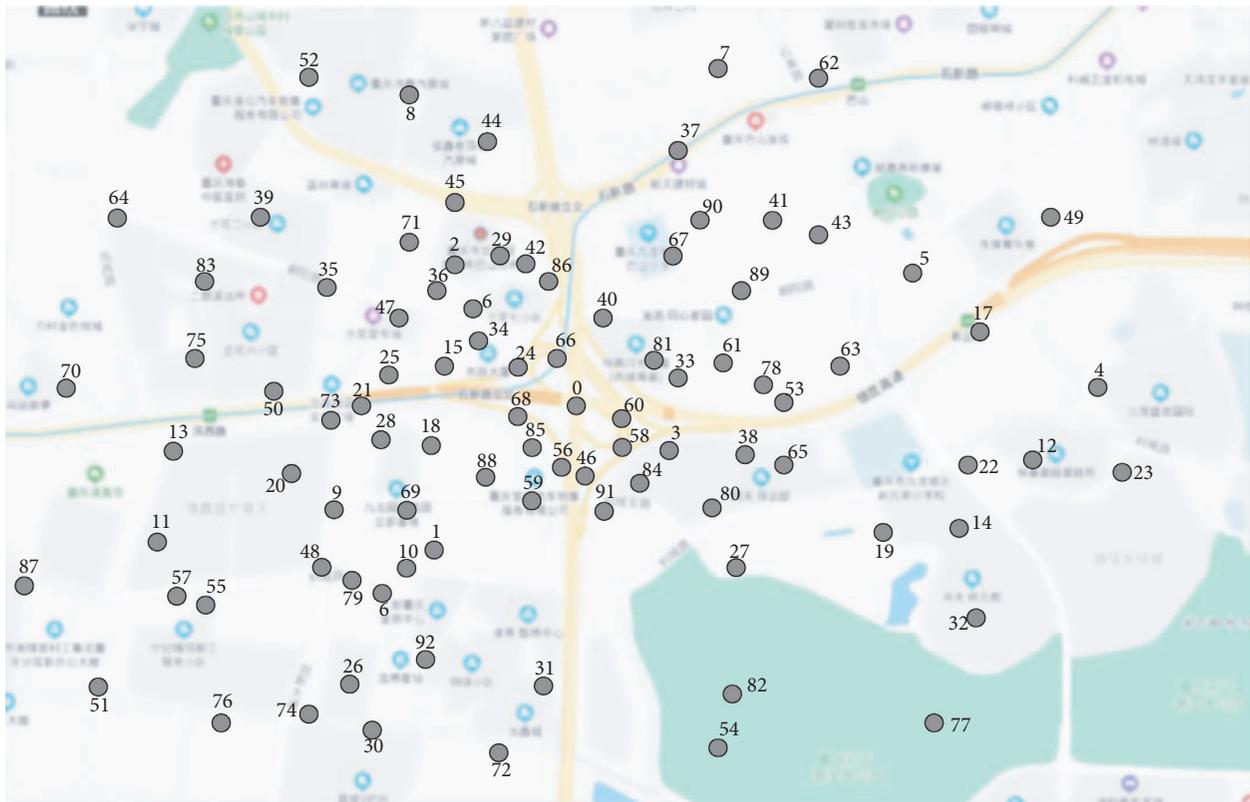


FIGURE 7: The spatial distribution of 46 orders.

more operation costs can be saved for the delivery platforms. However, in the most extreme cases, one order is assigned to one delivery rider only by the takeout service platform since

the objective is to satisfy all customer requests to obtain a good service reputation. We simulate three different order time windows and discuss the number of riders required.

TABLE 5: Optimization of rider scheduling.

Rider index	Rider routes
1	0-1-21-25-47-71-67-0
2	0-12-22-14-58-60-68-0
3	0-4-23-27-69-73-50-0;
4	0-41-5-30-76-51-87-11-57-0
5	0-24-70-13-59-20-66-0
6	0-16-45-44-62-43-90-89-91-0
7	0-37-39-83-15-85-61-0
8	0-33-46-92-6-79-18-64-52-0
9	0-29-8-75-31-54-77-32-19-65-78-0
10	0-40-86-2-36-28-48-74-82-0
11	0-34-42-35-88-80-81-0
12	0-3-7-49-17-63-53-38-84-0
13	0-9-55-26-72-10-56-0

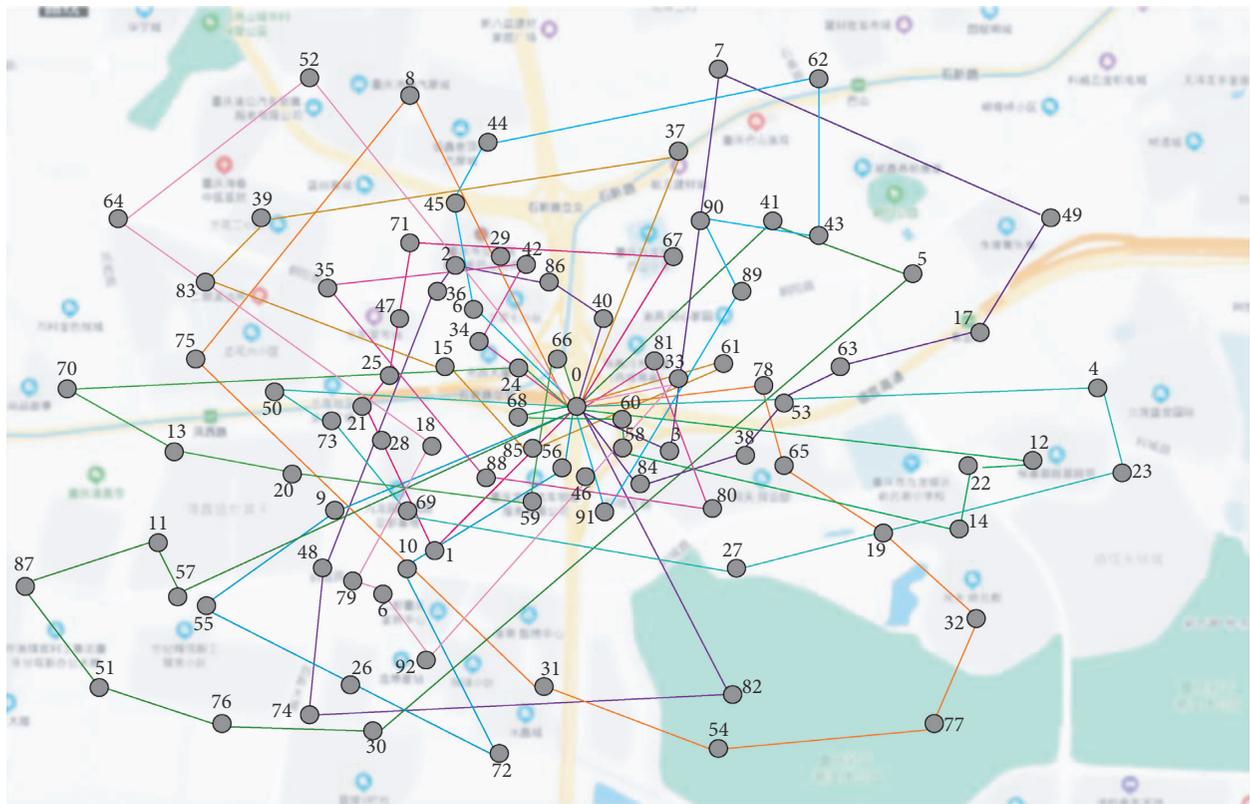


FIGURE 8: Visualization of the rider’s scheduling route.

Case 1 is the relaxed time windows, and the rider only has 60 minutes. Case 2 is the standard time window, and the rider has to go from the pickup node to the delivery node in 45 minutes. Case 3 is the tight time windows, and only 30 minutes is allowed. We test the three different time windows in the Luosifu business district, and the corresponding results are shown in Figure 9.

As the time window tightens, Figure 8 displays an opposite trend between the number of dispatched riders and each rider’s average order number. A more relaxed time window means fewer riders and more orders for each rider. Overall, the results show that the tightness of the order time windows and the

number of riders are closely related. More importantly, tighter time windows will inevitably make the distribution of transport capacity more imbalanced because the food delivery platforms have to assign more riders in peak periods.

5.3.2. *Analysis on the Regional Familiarity.* In the second-stage of the model, we prohibit the riders from visiting the sub-regions’ orders they are not familiar with regions. Because riders may spend less time to finish orders in a more familiar region, in the previous instance, we use the routes provided by Baidu Map, but there may be other shorter

TABLE 6: Number of riders in 23 sub-regions for instant deliveries.

No.	The number of the riders after optimization	The actual used riders
1	18	19
2	16	17
3	17	18
4	14	15
5	13	16
6	14	15
7	17	19
8	14	17
9	14	14
10	17	18
11	14	16
12	15	15
13	18	21
14	13	14
15	14	15
16	14	17
17	16	16
18	13	15
19	15	16
20	12	14
21	14	16
22	17	20
23	14	16

TABLE 7: The minimum number of riders required for each sub-region and each period of the whole delivery day.

Sub region number	Subperiod number					
	1	2	3	4	5	6
1	7	13	7	9	12	7
2	14	6	7	8	7	13
3	13	15	13	9	7	13
4	12	5	7	12	8	8
5	11	13	13	13	14	12
6	13	12	10	13	11	6
7	10	13	9	11	14	10
8	12	9	10	14	8	6
9	6	11	9	8	14	14
10	5	14	7	14	13	6
11	12	19	12	14	12	20
12	14	19	18	18	14	15
13	12	18	16	20	12	18
14	12	15	13	17	13	14
15	10	12	13	15	20	12
16	18	12	19	10	10	18
17	14	15	12	16	13	15
18	17	17	18	13	17	16
19	18	12	19	15	20	17
20	15	10	10	15	14	20
21	5	8	7	9	13	6
22	7	15	12	14	15	10
23	14	6	9	13	6	14

TABLE 8: Regional scheduling of some riders in each period.

Sub region number	The rider number									
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	1	1	1
2	3	1	0	2	0	1	0	0	0	0
3	6	4	1	1	2	4	2	0	0	0
4	9	1	2	3	3	6	4	1	0	0
5	11	1	2	1	1	4	6	4	0	0
6	13	4	2	3	4	4	8	2	0	1

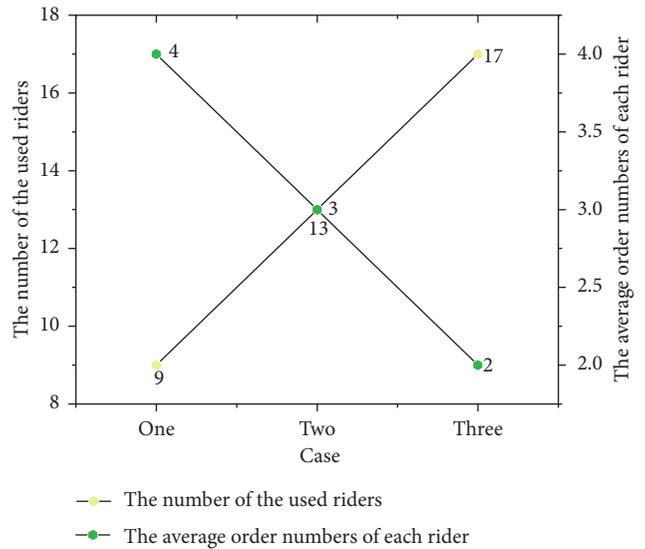


FIGURE 9: Relationship between the number of riders and the number of orders per capita under different time windows.

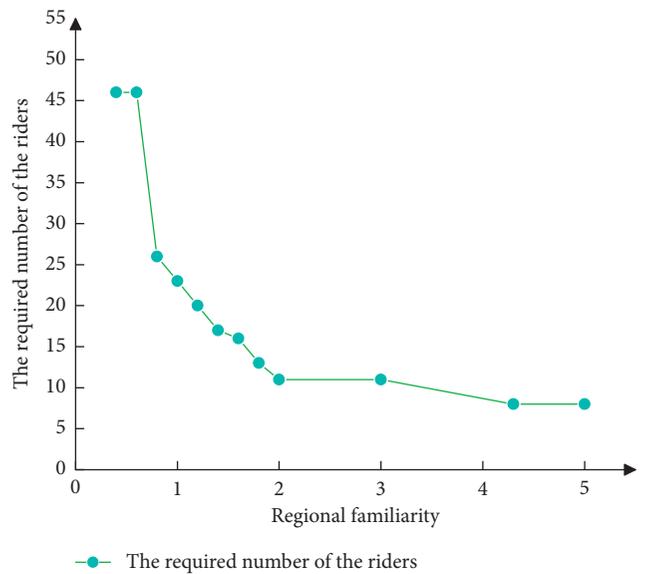


FIGURE 10: Relationship between regional familiarity and the required number of riders.

routes in practice. Therefore, we define the regional familiarity to describe the ratio between the Baidu Map distance and the rider's actual travel distance. It is evident that the smaller the familiarity, the larger the actual travel distance. We still take the Luosifu as an example and discuss the transportation capacity fluctuations in different regional familiarity. The results are depicted in Figure 10.

In Figure 10, the delivery system's number of delivery riders increases with the regional familiarity. When the rider maintains a higher regional familiarity, the required rider will be smaller, which will reduce the transportation capacity demand. Otherwise, the company assigns more riders, or the rider must travel a longer distance, which will increase the transportation capacity burden.

6. Conclusions

This study introduced a two-stage model to tackle riders' optimization problems in the on-demand food delivery business. Considering the intensively fluctuating demand of customer orders regarding period and region, the whole municipal level region is divided into smaller sub-regions while the whole period is divided into several sub-time periods. For each subregion, we formulate a dial-a-delivery rider model (first-stage of the model of our model) to minimize the delivery riders' number in each period. Furthermore, the imbalance of transportation capacity in different sub-regions and sub-periods is considered, and a transportation capacity allocation model (the second-stage of our model) is then built to minimize the time of the delivery rider's schedule. Due to the computational complexity of the first-stage of the model, we develop an ALNS heuristic to generate a near-optimal solution. In the next step, we have compared our proposed heuristic with results from other studies. Finally, the second-stage of the model is solved by Gurobi 9.1, a commercial solver. The dataset from an O2O foodservice platform in China is used to conduct the case study.

The results show that fewer delivery riders are required than the current delivery system in use by applying the first-stage of the model. The cost of saving after applying our model is 4200 CNY, and the new plan of delivery rider scheduling reduces the total delivery times by 328. Hence, the proposed model and algorithm could optimize the number of delivery riders in each subregion and improve the delivery rider scheduling. Besides, we find that the tightness of the order time windows and the number of delivery riders are closely related, and a tighter time window means more delivery riders and fewer orders for each delivery rider. Riders are found to travel significantly less distance when they are more familiar with the task subregion. However, in this study, we only focus on static order demand and ignore potential customer requirements. Moreover, in terms of riders' capacity, this paper mainly considers how to divide the whole region into small sub-regions, and separate the planning horizon into small subperiod. Furthermore, the division of rider service sub-regions depends on the area where the business circle is located, since this way is more intuitive in assessing customer demand. Meanwhile, an appropriate resource partition, with many reasonable

considerations, may outperform the proposed way in this study model, in terms of optimizing service resources.

There are several directions for further research. Firstly, we will consider the order demand mode of different delivery areas and the historical fluctuations of order demand in peak and off-peak periods, weekdays, and holidays. Secondly, appropriate resource partition methods are designed to optimize the allocation and scheduling of distribution-driven subareas. At last, the uncertainty in the delivery process may be considered (such as the uncertainty of order extraction (unexpected delay and real-time traffic conditions) to propose a more robust solution. Overall, considering more uncertainties, the on-demand food distribution-driven scheduling problem may also be worth studying.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (71971036, 71421001, and 71531002), the Major Program of Key Disciplines in Dalian (2019J11CY002), the Key R&D Project of Liaoning Provincial Department of Science and Technology (2020JH2/10100042), and the Humanities and Social Sciences Program of the Ministry of Education of China (19YJA630084).

References

- [1] Iimedia, "Research report on china's online catering takeout industry in the first half of 2017," 2017, <https://www.iimedia.cn/c400/54716.html>.
- [2] B. Kin, T. Ambra, S. Verlinde, and C. Macharis, "Tackling fragmented last mile deliveries to nanostores by utilizing spare transportation capacity-A simulation study," *Sustainability*, vol. 10, no. 3, p. 653, 2018.
- [3] Reuters, "How Amazon Is Making Package Delivery Even Cheaper," 2016, <https://fortune.com/2016/02/18/amazon-flex-deliveries>.
- [4] B. Yildiz and M. Savelsbergh, "Service and capacity planning in crowd-sourced delivery," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 177–199, 2019.
- [5] W. Huang, Z. Zhang, X. An, and G. Min, J. Li, Dynamic scheduling for urban instant delivery with strict deadlines," in *Proceedings of the ICC 2020 - 2020 IEEE International Conference on Communications*, pp. 1–6, Dublin, Ireland, 2020.
- [6] V. C. S. Yeo, S.-K. Goh, S. Rezaei, and Y. Sern, "Consumer experiences, attitude and behavioral intention toward online food delivery (OFD) services," *Journal of Retailing and Consumer Services*, vol. 35, no. 35, pp. 150–162, 2017.
- [7] P. Sun, "Your order, their labor: an exploration of algorithms and laboring on food delivery platforms in China," *Chinese Journal of Communication*, vol. 12, no. 3, pp. 308–323, 2019.

- [8] C. Archetti and M. G. Speranza, "The inventory routing problem: the value of integration," *International Transactions in Operational Research*, vol. 23, no. 3, pp. 393–407, 2016.
- [9] W. Liao, L. Zhang, and Z. Wei, "Multi-objective green meal delivery routing problem based on a two-stage solution strategy," *Journal of Cleaner Production*, vol. 258, p. 120627, 2020.
- [10] M. W. Ulmer, "Delivery deadlines in same-day delivery," *Logistics Research*, vol. 10, no. 3, pp. 1–15, 2017.
- [11] K. C. So and J.-S. Song, "Price, delivery time guarantees and capacity selection," *European Journal of Operational Research*, vol. 111, no. 1, pp. 28–49, 1998.
- [12] A. M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk, "Crowdsourced delivery-A dynamic pickup and delivery problem with ad hoc drivers," *Transportation Science*, vol. 53, no. 1, pp. 222–235, 2019.
- [13] V. E. Castillo, J. E. Bell, W. J. Rose, and A. M. Rodrigues, "Crowdsourcing last mile delivery: strategic implications and future research directions," *Journal of Business Logistics*, vol. 39, no. 1, pp. 7–25, 2018.
- [14] I. Dayarian and M. Savelsbergh, "Crowdshipping and same-day delivery: employing in-store customers to deliver online orders," *Production and Operations Management*, vol. 29, no. 9, pp. 2153–2174, 2020.
- [15] N. Kafle, B. Zou, and J. Lin, "Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery," *Transportation Research Part B: Methodological*, vol. 99, pp. 62–82, 2017.
- [16] Y. Wang, S. Peng, X. Guan et al., "Collaborative logistics pickup and delivery problem with eco-packages based on time-space network," *Expert Systems with Applications*, vol. 170, Article ID 114561, 2021.
- [17] E. P. Jack and T. L. Powers, "A review and synthesis of demand management, capacity management and performance in health-care services," *International Journal of Management Reviews*, vol. 11, no. 2, pp. 149–174, 2009.
- [18] A. F. Veinott and H. M. Wagner, "Optimal capacity scheduling-II," *Operations Research*, vol. 10, no. 4, pp. 533–546, 1962.
- [19] J. Yoon, H. Yildiz, and S. S. Talluri, "Risk management strategies in transportation capacity decisions: an analytical approach," *Journal of Business Logistics*, vol. 37, no. 4, pp. 364–381, 2016.
- [20] J. Zhang, F. Liu, J. Tang, and Y. Li, "The online integrated order picking and delivery considering Pickers' learning effects for an O2O community supermarket," *Transportation Research Part E: Logistics and Transportation Review*, vol. 123, pp. 180–199, 2019.
- [21] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "Dynamic pickup and delivery problems," *European Journal of Operational Research*, vol. 202, no. 1, pp. 8–15, 2010.
- [22] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, 2007.
- [23] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.
- [24] L. He, "Research on route optimization of takeout delivery vehicles considering occasional road congestion," *Frontiers in Economics and Management*, vol. 1, no. 10, pp. 96–102, 2020.
- [25] K. Braekers and A. A. Kovacs, "A multi-period dial-a-ride problem with driver consistency," *Transportation Research Part B: Methodological*, vol. 94, pp. 355–377, 2016.
- [26] G. Ghiani, E. Manni, A. Quaranta, and C. Triki, "Anticipatory algorithms for same-day courier dispatching," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 1, pp. 96–106, 2009.
- [27] M. A. Klapp, A. L. Erera, and A. Toriello, "The dynamic dispatch waves problem for same-day delivery," *European Journal of Operational Research*, vol. 271, no. 2, pp. 519–534, 2018.
- [28] T. Gschwind and M. Drexel, "Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem," *Transportation Science*, vol. 53, no. 2, pp. 480–491, 2019.
- [29] N. Azi, M. Gendreau, and J.-Y. Potvin, "A dynamic vehicle routing problem with multiple delivery routes," *Annals of Operations Research*, vol. 199, no. 1, pp. 103–112, 2012.
- [30] Q. Gu, T. Fan, F. Pan, and C. Zhang, "A vehicle-UAV operation scheme for instant delivery," *Computers & Industrial Engineering*, vol. 149, p. 106809, 2020.
- [31] B. Yildiz, M. Savelsbergh, and M. Savelsbergh, "Provably high-quality solutions for the meal delivery routing problem," *Transportation Science*, vol. 53, no. 5, pp. 1372–1388, 2019.
- [32] S. A. Voccia, A. M. Campbell, and B. W. Thomas, "The same-day delivery problem for online purchases," *Transportation Science*, vol. 53, no. 1, pp. 167–184, 2019.
- [33] M. W. Ulmer and B. W. Thomas, "Same-day delivery with heterogeneous fleets of drones and vehicles," *Networks*, vol. 72, no. 4, pp. 475–505, 2018.
- [34] M. W. Ulmer and S. Streng, "Same-Day delivery with pickup stations and autonomous vehicles," *Computers & Operations Research*, vol. 108, no. 8, pp. 1–19, 2019.
- [35] M. A. Klapp, A. L. Erera, and A. Toriello, "Request acceptance in same-day delivery," *Transportation Research Part E: Logistics and Transportation Review*, vol. 143, p. 102083, 2020.
- [36] R. W. Bent and H. Van, "Scenario-based planning for partially dynamic vehicle routing with stochastic customers," *Operations Research*, vol. 52, no. 6, pp. 977–987, 2004.
- [37] B. Li, D. Krushinsky, T. Van Woensel, and H. A. Reijers, "An adaptive large neighborhood search heuristic for the share-a-ride problem," *Computers & Operations Research*, vol. 66, pp. 170–180, 2016.
- [38] Z. Steever, M. Karwan, and C. Murray, "Dynamic courier routing for a food delivery service," *Computers & Operations Research*, vol. 107, pp. 173–188, 2019.
- [39] W. Wang and L. Xie, "Optimal pricing of crowdsourcing logistics services with social delivery capacity," *Journal of Combinatorial Optimization*, pp. 1–23, 2021.
- [40] P. Toth and D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Applied Mathematics*, vol. 123, no. 1–3, pp. 487–512, 2002.
- [41] J.-F. Cordeau and G. Laporte, "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," *Transportation Research Part B: Methodological*, vol. 37, no. 6, pp. 579–594, 2003.
- [42] F. Lehuédé, R. Masson, S. N. Parragh, O. Péton, and F. Tricoire, "A multi-criteria large neighbourhood search for the transportation of disabled people," *Journal of the Operational Research Society*, vol. 65, no. 7, pp. 983–1000, 2014.