# A Workload-Balancing Order Dispatch Scheme for O2O Food Delivery with Order Splitting Choice 

Ke Wang (D) Yulin Zhou (D) and Lingzhen Zhang * (D)<br>School of Management, Shanghai University, Shanghai 200444, China; ke@shu.edu.cn (K.W.); yulinzhou_98@shu.edu.cn (Y.Z.)<br>* Correspondence: shangdazlz@shu.edu.cn; Tel.: +86-6613-7696-804

Citation: Wang, K.; Zhou, Y.; Zhang, L. A Workload-Balancing Order Dispatch Scheme for O2O Food Delivery with Order Splitting Choice. J. Theor. Appl. Electron. Commer. Res. 2022, 17, 295-312. https://doi.org/ 10.3390/jtaer17010015

Academic Editors: Eduardo
Álvarez-Miranda, Hua Ke, Zhiguo Li and Zhang Zhao

Received: 25 November 2021
Accepted: 29 January 2022
Published: 9 February 2022
Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.


Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).


#### Abstract

Online-to-offline (O2O) food delivery service refers to an emerging modern business model that enables customers to order foods from local restaurants via an online platform, and then receive and enjoy them offline after the delivery, offered by couriers. Such service, discussed in this article, specifies that a customer can order from multiple restaurants in a single order and choose for them to be delivered together or separately, whereas the commonly discussed mode only permits placing an order in one restaurant at once. In this service, one crucial issue is how to dispatch these orders to couriers for offline delivery. For this, we propose a new three-stage order dispatch scheme, namely, pseudo-assign first, re-route second, and courier selection last, aiming to deliver the orders in time and balance the couriers' workload. Due to the dynamism and uncertainty inherently involved in this issue, we also take responsiveness to future demands and robustness into consideration when making the dispatch. Compared with existing approaches, the new one significantly balances the couriers' workload and, meanwhile, keeps good performance in delay rate, making the decisions more practical. Furthermore, this study analyzes the influence of customers' preference for order splitting and number of couriers on the efficiency of the distribution system, thereby interesting managerial insights for O 2 O food delivery are revealed.


Keywords: O2O service; online food delivery; split delivery; dynamic stochastic VRP; workload balance

## 1. Introduction

Thanks to the extensive applications of information technologies and their offspring, online-to-offline ( O 2 O ) businesses have recently gained great popularity by applying the convenience benefit of online services to offline reality [1]. One representative type of the O2O business model is the food delivery service, which enables the customer to order restaurant food via online (mobile) platforms [2]. As a new booming business model, it has been widely applied in many countries, showing a bright development prospect [3,4]. In the last decade, many renowned O2O food delivery companies have grown and expanded rapidly, such as UberEats and Grubhub, based in America, as well as Meituan and Eleme in China. According to Statista (https:/ /www.statista.com/outlook/dmo/eservices/online-food-delivery/worldwide, accessed on 25 November 2021), the worldwide online food delivery revenue is estimated to be 270 billion USD in 2021, with a predicted annual increase of $10.39 \%$ from 2021 to 2025. Additionally, due to the COVID-19 pandemic, online food orders increased rapidly in recent years [5]. For instance, Deliveroo, a European online food delivery platform, received an increase of more than $100 \%$ in the first half of 2021.

Generally, in the commonly discussed O2O food delivery service, an order is only associated with one restaurant, implying that several orders have to be placed if the customer wants to enjoy foods made by different restaurants. Taking a step back, even if the customer places multiple orders in a short period, there is no option for the customer to choose whether to have them delivered at one time. These orders are usually delivered
by several different couriers, and the customer has to receive the ordered foods separately. Consequently, with the increasing demand for customization, the mode that customers cannot participate in the selection of delivery options has been unable to gain a high level of customer satisfaction. With this consideration, Steever et al. [6] proposed the virtual food court delivery problem (VFCDP), a more flexible business model, in which multiple restaurants could be included in a customer's single order, and all the ordered foods are to be delivered together by one courier.

In real-life, some customers may prefer to receive all foods at one time, since they are inconvenient or do not like to collect these orders frequently. However, to pick up all the ordered foods and deliver them together by the same courier would cause a long led time. On the contrary, some other customers may accept order splitting and separate delivery, in order to receive their booked products as soon as possible. This delivery mode can also be observed in many e-commerce services, such as Amazon and JD.com. Besides, engaging the customers in the selection of delivery options helps improve customer satisfaction.

Therefore, we expand the O2O food delivery service with order splitting choice. Compared with the VFCDP proposed by Steever et al. [6], the situation discussed in this article gives customers more delivery choices, when several restaurants are ordered in a single order. Specifically, when a customer orders in multiple restaurants simultaneously, he/she can decide whether to have them delivered together by one courier or not. If the former is chosen, it is the same with the case discussed by Steever et al. and called non-split delivery. Otherwise, it becomes a split delivery situation. Furthermore, as a critical factor in this new delivery mode, the split delivery rate needs to be considered and analyzed as an unknown variable by the O2O food delivery companies, when customers are enabled to choose split delivery or non-split delivery if several restaurants are ordered in a single order.

Summarizing, this paper extends the literature on O2O food delivery by introducing a new service mode, which allows customers to select non-split or split delivery, and aims to develop an effective order dispatch scheme for this new service. Additionally, managerial insights are derived for the practical O 2 O food delivery. The main contributions of this paper are two-fold.

- For this delivery service case, we found out that the assignment rule of new orders proposed by Steever et al. [6] is less efficient in real-life, especially for its uneven workload distribution. Consequently, we offer a three-stage order dispatch scheme, namely, pseudo-assign first, re-route second, and courier selection last, showing an impressive performance on balancing courier's workloads, without needing to compromise on delivery efficiency. In addition, this new scheme attempts to handle the dynamic and uncertain issues by considering the responsiveness to future demands and robustness to uncertainty.
- To illustrate the feasibility and applicability of this new service mode, we have conducted extensive experiments, which involve the influence of customers' preference of split delivery on the delivery efficiency and decision of economical employments under distribution-known demands, casting several vital managerial policies. These conclusions would make food delivery platforms more adaptive and profitable, if they implemented this new O 2 O business model, as competition in O 2 O food delivery gets fiercer, and higher customer satisfaction is badly needed.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work regarding this problem. Section 3 introduces the definition of the problem. To get a balancing assignment policy for couriers, we articulate the new order dispatch scheme in Section 4, and its performance, compared with existing methods, is presented through some experimental analyses in Section 5. Besides, some valuable managerial insights are drawn from the experimental results. Finally, the summary and limitations of this research are provided in Section 6.

## 2. Literature Review

Online-to-offline service [7] is an emerging business model that developed rapidly in recent years and has gained much attention in academic research, as well. By effectively connecting suppliers to customers, business process to distribution logistics, and online service to offline reality, it provides considerable merits [1] to all participants in this business, as well as the performance of whole supply chain [8], but also presents challenges to the traditional supply chain management in many aspects, such as supplier selection [9], supply chain procedure [10], and network design [11,12].

As for the O2O food delivery service [1], by virtue of various mobile devices and smart terminals with seamless access to Internet, it gathers individual customers, local restaurants, and delivery couriers together on an online platform, on which customers can search and order various foods; then, the orders are prepared and delivered offline by the corresponding restaurants and couriers. Similar service modes that involve ordering online and delivering offline may be called different things in the literature, such as restaurant meal delivery problem [13,14], on-demand food delivery [15], on-demand meal delivery [16], online food delivery problem [2], and virtual food court delivery problem [6]. Here, more generally, we name them O2O food delivery, following [1].

In general, this problem is related to the dynamic vehicle routing problem (DVRP) [17,18], particularly the DVRP with stochastic travel times and demand (or customer requests) [19-21], since this service should be operated in an environment inherent with high uncertainty. For example, we do not know when and how many food orders the next customers will place, as well as the time needed for the corresponding restaurants to prepare these orders. However, considering the specific O2O food delivery mode discussed in this paper, the approaches to DVRP cannot be directly applied to our problem. Therefore, we here clarify some solutions for dealing with O 2 O (online) food delivery services in the most related articles.

In order to deliver the requested foods to the customers, assigning couriers to pick up the foods (i.e., order dispatching) and planning the couriers' routes are two crucial decisions in this problem, since they directly affect the service quality and customer satisfaction. So, in response to the dynamic and stochastic customer requests, the discipline of assigning new orders to the couriers and replanning the corresponding routes triggers lots of scholars' devotion.

The proactive heuristic method, proposed by Steever et al. [6], provides a feasible solution for this problem. It measures the responsiveness to future demand by two futurelooking metrics (i.e., equity and dispersion), influenced by the position of couriers in a future time. More specifically, the equity matrix considers the distance of the nearest courier to each restaurant, at a future time point, which has various weights. Beyond that, the restaurants are given different weights, according to their historical order quantity. Dispersion measures how dispersed couriers are at each future time point. They believe that the smaller the equity and greater the dispersion is, the faster couriers can respond to future orders, meaning the more the decision-maker tends to consider future demand, rather than current need. Besides, the bid selection is used to make a trade-off between the current optimal solution, according to the objective function value, as well as other solutions considering the future demand. Differently, in this study, we only use one indicator to measure responsiveness to future demand, simplifying the decision process. Besides, unlike Steever et al. [6], using the exact positions of couriers, we apply the spots that the courier has to stay (i.e., the positions of the courier's unfinished customers and restaurants). As a result of it, the workload-balancing is improved greatly.

Apart from the work of Steever et al., the anticipatory customer assignment policy (ACA), proposed by Ulmer et al. [13], also deals with uncertainty in the decision-making process. ACA attempts to reduce the occurrence of late orders by delaying dispatching orders to the courier, introducing time buffers, and using order-bundling appropriately. Instead, the method proposed in this paper will dispatch orders as soon as possible.

As for deciding which courier to deliver the new requested order, a significant, but often neglected, consideration is the fairness, regarding the workload, among couriers. It has been pointed out, by Abbatecola et al. [22], that if we only focus on the shortest delivery
time or distance when giving the routing scheme, the solution will result in an impractical situation, i.e., a problem where some couriers need to bear a lot of workloads or take a long journey, while some others are idle. In other words, although such an "optimal" solution looks excellent, in terms of the shortest time or distance, it is useless in practice. Actually, more and more attention is paid to the issue of workload balance in some related research fields, such as on-demand ride-sourcing [23], humanitarian relief logistics problems [24], and waste collection vehicle routing problems [25]. Some corresponding methods have been proposed in succession, in order to achieve workload balance in the delivery [26]. For example, a genetic algorithm meta-heuristic approach, applied by Keskinturk et al. [27], for solving a capacitated vehicle routing problem with a load-balancing objective. Nevertheless, the revealed studies on workload balance in O2O food order dispatch are still scarce.

In summary, although some existing methods could be applied to the similar food delivery problems, they are only suitable for some specific situations and goals. Considering the gaps mentioned above, in this study, we propose a real-time order dispatch scheme for the O2O food delivery with dynamic customer requests and order splitting choice, which maintains workload balance among couriers and keeps good performance in delivery efficiency, as well.

## 3. Problem Definition

### 3.1. Problem Statement

In the O2O food delivery service, an online platform directly connects customers, restaurants, and couriers via various smart mobile terminal devices. It is obliged to provide efficient food delivery to meet customers' current demands. With the emerging of a new order, the whole operation process for the O2O food delivery service is presented in Figure 1.


Figure 1. Schematic diagram of the completion of orders.
The platform permits the customer to order food from multiple restaurants in a single order. If more than one restaurant is in one order, the customer can choose non-split or split delivery. The former means that the order must be delivered by one courier at one time, implying less disruption to the customer. The latter, signifying the customer is willing to receive food separately, depends on the distribution system towards the assignment of couriers.

Once a customer places an order, the information will be transmitted to the platform immediately, and the corresponding restaurant will receive the order requirements. Meanwhile, the platform will access the current couriers' status, including the location and task process at present, for the sake of dispatching the new order properly. After the assignment, the route of the chosen courier will be re-planned. Then, the delivery platform will give the customer a promised arrival time, meaning that it will get low customer satisfaction if failed. Couriers need to complete the tasks of picking up and delivering orders, in turn, according to the planned route. If the courier arrives at a restaurant that hasn't prepared the food, the courier needs to wait until the restaurant has the food packed up. The whole service process of an order, from the placement of the order to the fulfillment of the delivery
includes two service intervals, besides the traveling on routes. One is the time spent at the restaurants for the couriers to pick up the ordered food (not including waiting time), the other is the time needed to hand the food over to the customer, after arriving at the customer's location. During the whole process, after the platform completes the assignment, customers can check the position of their corresponding couriers at any time. For the sake of making better decisions, the platform may access each restaurant's historical demand data.

To define the problem more explicitly, we give some assumptions as follows.
(1) The courier can not refuse the order assigned by the delivery platform. In other words, once a courier is assigned to the new order, this assignment does not change anymore.
(2) The courier has no fixed initial position, as well as the end position. When the courier finishes the current orders, he will wait in suit for the new order.
(3) Order delay is allowed. Specifically, it can be later than the promised delivery time.

Considering the dynamic feature of this problem, we are supposed to make sequential decisions, triggered by the generation of a new order (i.e., a new customer request). Specifically, we have to determine which courier should finish the new order and how would the new route be set, according to a series of successive decision points.

### 3.2. Desired Order Dispatch Scheme

As is the problem declared above, after a new order comes, the platform needs to select a proper courier and re-plan the corresponding route. To support it, this study attempts to develop an effective order dispatch scheme, which aims to deliver the orders in time and, meanwhile, balance the couriers' workload. Here, we illuminate the desired order dispatch scheme, with each side in consideration.

Delivery in time. The first and foremost goal is timely delivery. Since the delivery time seriously affects the quality of customer service, and that meeting customers' needs is the foundation of the existence of this O2O business model, we need to do our utmost to guarantee on-time delivery. Therefore, delay rate (DR), the ratio of delayed orders to the total number of orders, is used as the primary measure to evaluate the performance of a designed order dispatch scheme. To further measure the customer satisfaction, we can also apply the average late time (ALT) of delayed orders and average early time (AET) of orders delivered early. However, since the negative effect of late delivery is much stronger than the positive effect of an early delivery, the importance of AET is much lower than that of ALT [28].

Workload balance. Since a large variety in the workload will cause some ill consequences (e.g., low job satisfaction and high employee turnover), we are supposed to balance the couriers' workload. Although there are many indicators to measure the workload balance, such as the standard deviation of the driving distance [25] and lexicographic minimax approach [29], the courier pays more attention to his salary, which is directly related to the number of orders completed. Therefore, we utilize the standard deviation of the numbers of assigned orders of the couriers to represent the level of unbalance of the couriers' workload in this study, denoted as the unbalance index-standard deviation (UIsd) in the following discussions. The smaller it is, the more balanced workload is, and the designed order dispatch scheme is more practical in real-life applications.

Furthermore, due to the dynamic and uncertain nature of this problem, responsiveness to future demand and robustness to uncertainty should also be taken into consideration when making the order dispatch.

Responsiveness to future demand. When making the order dispatch decisions, we should consider satisfying the current demand quickly enough, as well as its impacts on the response to future orders. In other words, we do not expect the order allocation to always be optimal at the current decision point, but has a good performance (such as lower DR and UIsd) in the long-term.

Robustness to uncertainty. Various uncertain factors exist in the whole operation process of the O 2 O food delivery service, such as the time required by the restaurants to prepare the food and for the couriers to travel on the road. Although we can take advantage of historical data or predicted information when making the order dispatch, there is constantly a gap between them and what happens in real-life. Therefore, we hope that our solutions are robust to these uncertainties. In other words, when the actual time required in these operations changes in a small range, the designed order dispatch scheme still performs well.

## 4. Proposed Order Dispatch Scheme

To clarify the proposed order dispatch scheme, we use Figure 2 to explain how to choose a proper courier when a new order emerges. It contains three steps, including pseudo-assign, re-route, and courier selection, as introduced in detail below.


Figure 2. Framework of the proposed order dispatch scheme.
When a new order emerges, if it only contains one restaurant or the customer chooses non-split delivery, it will be treated as a single order. If it includes multiple restaurants and the customer selects split delivery, the initial order will be split into multiple orders. Additionally, the number of orders is identical to the number of restaurants contained in the initial order. As for each order, we will try to assign it to every courier one-by-one (namely, pseudo-assign) to see how it would be delivered by the couriers (re-route), as well as the corresponding outcome of the whole delivery system. Subsequently, a trade-off between the current optimality and responsiveness to possible future demand should be made, in order to assign a proper courier to this order (courier selection). Then, the order is to be delivered by this courier, along with his/her rerouted optimal routes.

### 4.1. Pseudo-Assign and Re-Route

To clarify the process of pseudo-assign and re-route, as well as the underlying optimization problem for re-routing, some necessary parameters and notations are introduced in Table 1 first.

The set of available couriers and restaurants who provide services on this platform, as well as the unserviced customers, are denoted by $V, R$, and $C$, respectively. When a new customer ( $c^{\text {new }}$ ) comes, he/she may place an order ( $I_{c^{n e w}}$ ) to one or multiple restaurants $(R)$ and choose non-split or split delivery if multiple restaurants are ordered. If non-split delivery is selected, then denote $c^{\prime}=c^{\text {new }}$, otherwise the order will be split into $\left|I_{c^{\text {new }}}\right|$ orders, implying $c^{\prime}=\left\{c_{i}^{\text {new }}\left|i=1, \cdots,\left|I_{c^{\text {new }}}\right|\right\}\right.$. Then, $C$ is updated by $\left\{C \cup c^{\prime}\right\}$.

Whenever a new order $\left(s c \in c^{\prime}\right)$ arrives, we will assume that the new order is assigned to each courier. Then, we can replan each courier's optimal route (re-route) and get the corresponding performance by solving a sub-problem.

Table 1. Parameters and notations of the online food delivery problem.

| Parameter | Explanation |
| :--- | :--- |
| $V$ | Set of available couriers. |
| $R$ | Set of available restaurants. |
| $C$ | Set of customers whose orders have not been serviced. |
| $C_{v}$ | Set of unserved customers who are assigned to courier $(v)$. |
| $I_{\mathcal{C}}$ | Set of restaurants to which customer $(c \in C)$ placed the order. |
| $c^{n e w}, I_{\mathcal{C}^{\text {new }}}$ | A new customer and the set of restaurants ordered by this customer. |
| $\Delta_{v}^{0}$ | Initial node (position) for courier $(v \in V)$. |
| $\gamma^{\max }$ | A penalty coefficient $(\gamma \geq 0)$. |
| $t_{c}^{\text {min }}$ | Promised delivery time for customer $(c)$. |
| $t_{i}^{\max }$ | Estimated time for restaurant $(i \in R)$ to finish the preparation of food. |
| $\delta^{\max }$ | Maximum tolerable delay time. |

In this sub-problem, the courier $(v)$, located at $\Delta_{v}^{0}$, has a set of unserved customers $\left(C_{v}\right)$. Each customer $\left(c \in C_{v}\right)$ expected to receive his/her ordered food before the promised delivery time $\left(t_{c}^{\max }\right)$. However, a delivery delay may happen. In this system, a delay not exceeding $\delta^{\max }$ is allowed, but should be punished with a penalty coefficient $(\gamma)$. In addition, the food ordered from restaurant (i) can be picked up by couriers until $t_{i}^{\text {min }}$, i.e., the time that the restaurant finished the preparation of food. The service interval $\left(s_{i}\right)$ at the restaurants and customers' location, mentioned before, is considered to make the problem more realistic. Then, the sub-problem is to replan the courier's route to make the his/her performance maximized, which is formulated as:

$$
\begin{equation*}
\epsilon_{v}=\max \left(\sum_{c \in C_{v}}\left(t_{c}^{\max }-t_{v c}\right)-\gamma\left(t_{v c}-t_{c}^{\max }\right)^{+}\right) \tag{1}
\end{equation*}
$$

where $t_{v c}$ is the scheduled time for the courier $(v)$ to arrive at the customer (c). This objective function aims to minimize the delayed time and maximize the early delivered time, in order to obtain higher customer satisfaction.

Here, the sub-problem determines how the new route should be, if the new order is assigned to the courier (v), and $\epsilon_{v}$ measures the performance of the courier (v) in such a situation. Actually, the sub-problem that decides the specific route for each courier can be modelled as a mixed-integer linear program (MILP), which is similar to that proposed by [6]. A more detailed formulations of this model is provided in the online supplementary materials. This MILP can be directly solved by some well-known commercial solvers, such as Lingo, Cplex, and Gurobi.

By solving this MILP, the objective function value $\left(\epsilon_{v}\right)$ and current optimal route of courier (v), denoted as $N R_{v}$, are gained. Subsequently, they are utilized to select the optimal courier in the next stage.

### 4.2. Measurement of Responsiveness to Future Demand

Since the local optimal solution may become sub-optimal, or even poor, after a while, we can not leave out the considerations of future demand. With this consideration, this paper proposes a new index, which is different from the equity and dispersion proposed by Steever et al. [6], to measure the courier's responsiveness to future orders. Compared to [6], the index represented here is easier to calculate and can lead to better workload balance. The following represents the calculation process of this measurement.

First, calculate the couriers' expected position and status in several future time points. Here, the determination of the future time point is same to that of Steever et al. [6], as shown below:

$$
\begin{equation*}
\widehat{t_{i}}=\left[t_{\text {now }}+\mu_{a}+\frac{\sum_{r \in R}\left(t_{r} * P_{r}\right)}{|R|}+i * \sigma_{a}\right] \tag{2}
\end{equation*}
$$

where, $\widehat{t_{i}}$ is the future time point to be considered, $i \in[-2,-1,0,1,2] . t_{n o w}$ is the current time, $\mu_{a}$ is the mean customer interarrival time, and $t_{r}$ is the average order preparation time of restaurant $(r) . P_{r}$ is the proportion of the customer placing an order in the restaurant $(r)$, $|R|$ is the number of all restaurants, and $\sigma_{a}$ is the standard deviation associated with $\mu_{a}$.

Based on the mean speed and route, the expected location and status of each courier, at the given future time point, can be estimated. For the arrival of a new order, we can imagine that the couriers may not take this new order as the next task immediately, but more likely will insert it to his/her task list as a middle one or even the last task. Therefore, we use the intermediate nodes (customers' locations and restaurants' positions) that should be visited by the courier in his/her previously planned route, rather than the position anywhere the courier may arrive at $\left(\widehat{t}_{i}\right)$, which is applied by Steever et al. [6] to represent the courier's expected location.

We take the intermediate nodes in the route of each courier (v), whose estimated arrival time is between $\widehat{t_{i}}$ and $t_{c}^{m a x}$, as $\varphi_{i}^{v}$, where $c^{n e w}$ is the new customer. Then, at each future time point, find $\eta$ couriers nearest to each restaurant and calculate the average distance, representing the responsiveness of all couriers to the new order in the corresponding restaurant. To clarify it, we show it in Algorithm 1.

```
Algorithm 1: Measurement of responsiveness of couriers to the new order.
    for \(v\) in \(V\) do
        Assign new order \(s c\) to \(v\);
        Set \(\varphi_{i}^{v}=\varnothing\);
        for \(i\) in \([-2,-1,0,1,2]\) do
            Calculate \(\widehat{t_{i}}\), according to Equation (2);
            for node in \(\left\{C_{v} \cup I_{c, c \in C_{v}} \cup \Delta_{v}^{0}\right\}\) do
                Obtain the time \(t_{v, \text { node }}\) that the courier (v) arrived at node;
                if \(t_{v, \text { node }} \in\left[\widehat{t_{i}}, t_{c^{n n e w}}^{\text {max }}\right]\) then
                        Add node to \(\varphi_{i}^{v}\)
            for \(r\) in \(R\) do
                Find the nearest node \(n o d e_{r, \widehat{t_{i}}}^{v} \in \varphi_{i}^{v}\) and corresponding distance \(d i s_{r, \overparen{t_{i}}}^{v} ;\)
    for \(i\) in \([-2,-1,0,1,2]\) do
        for \(r\) in \(R\) do
            Find \(\eta\) closest nodes which belong to \(\left\{\right.\) node \(\left.e_{r, \widehat{t_{i}}}^{v}, v \in V\right\}\) to \(r\);
            Calculate the mean distance \(d i s_{r, \bar{t}_{i}}^{\eta v}\);
        Standardize \(d i s_{r, \widehat{t_{i}}}^{\eta v}\) to get StdDis \(r_{r, \bar{t}_{i}}^{\eta v}\);
    return \(M^{v}\), according to Equation (3)
```

In Algorithm 1, node $e_{r, \hat{t}_{i}}^{v}$ is the node where the courier (v) is closest to restaurant $(r)$ after time $\widehat{t_{i}}$, and $d i s_{r, \widehat{t_{i}}}^{v}$ is the distance between node $e_{r, \widehat{t_{i}}}^{v}$ and restaurant $(r)$. The mean distance of $\eta$ nodes to $r$, denoted as $d i s_{r, \hat{t}_{i}^{\prime}}^{\eta v}$ is used to indicate the responsiveness at time $\widehat{t_{i}}$, when a new order appears at restaurant $r \in R$. The smaller the value, the faster the response speed. $\eta$ is a positive integer. StdDis $\left.r, \hat{t}_{i}\right)$ is the standardized value of $d i s_{r, \hat{t}_{i}}^{\eta \eta}$, according to $d i s_{r, \hat{t_{i}}}^{\eta v} / \max _{r \in R}\left(d i s_{r, \widehat{t_{i}}}^{\eta v}\right)$.

Then, we propose an index $\left(M^{v} \in(0,1)\right)$ to measure the responsiveness to future orders, if the new order is assigned to courier $(v)$. The larger $M^{v}$ is, the faster the response to future orders.

$$
\begin{equation*}
M^{v}=\frac{\sum_{i=1}^{5} \sum_{r \in R}\left(1-S t d D i s_{r, \widehat{t}_{i}}^{\eta v}\right) * P_{r}}{5} \tag{3}
\end{equation*}
$$

For the sake of the efficiency of delivering current orders unfinished, but with less harm to future demand, we use the following method, expressed by Algorithm 2.

```
Algorithm 2: The trade-off between current optimal solution and future demand.
    Input: \(\lambda_{i}, i \in\{1,2\}, \epsilon_{v^{\prime}}, M^{v^{\prime}}, \epsilon_{v}, M^{v}, v \in V\)
    Output: Candidate v
    Set Candidate \(v=\{ \}\);
    for \(v\) in \(V\) do
        for \(i\) in \(\{1,2\}\) do
            if \(\frac{\left(\epsilon_{v^{\prime}}-\epsilon_{v}\right)}{\epsilon_{v^{\prime}}} \leq \lambda_{i} \leq \frac{\left(M^{v}-M^{v^{\prime}}\right)}{M^{v^{\prime}}}\) then
                    Add \(v\) to Candidate \(v\).
```

In Algorithm 2, $v^{\prime}$ is the courier with largest $\epsilon_{v}, v \in V$, and $\lambda_{i}$ is used to make a tradeoff between the maximization of the objective function and consideration of future orders. Here, we use $\lambda_{1}$ and $\lambda_{2}\left(\lambda_{2} \geq \lambda_{1}\right)$ to choose $v$, for the reason that only one selection range is not large enough. Through adding one more parameter, we will get enough candidate couriers who are more favorable to future demand. Through the consideration of future demands, if there is no courier candidate (v), we will choose the courier according to the maximization of the objective function value.

### 4.3. The Whole Process of Courier Selection

In the proposed order dispatch scheme, we first suppose the new order is assigned to each courier in turn, and then determine the courier's optimal route, after being assigned this new order, as well as the corresponding performance. The final assignment will be made after balancing the current optimality and responsiveness to future demand. The whole decision-making process is integrated in Algorithm 3. For the convenience of reading, we summarize the notations of the other parameters (and variables) involved in the solution algorithms in Table 2.

Table 2. Notations involved in the solution algorithms.

| Notations | Explanation |
| :---: | :---: |
| $\widehat{t_{i}}$ | The future time point to be considered, $i \in[-2,-1,0,1,2]$. |
| $t_{\text {now }}$ | The current time when a new order places. |
| $\mu_{a}$ | The mean customer interarrival time. |
| $\sigma_{a}$ | The standard deviation associated with $\mu_{a}$. |
| $t_{r}$ | The average order preparation time of restaurant ( $r$ ). |
| $P_{r}$ | The proportion of the customer placing an order in the restaurant (r). |
| $\varphi_{i}^{v}$ | Set of intermediate nodes in the path of each courier (v), whose estimated arrival time is between $\widehat{t_{i}}$ and $t_{s c}^{\max }$. |
| $\eta$ | A positive integer. |
| node $e_{r, \widehat{t_{i}}}^{v}$ | The node where the courier (v) is closest to restaurant (r) at time $\widehat{t_{i}}$ |
| $d i s_{r, \widehat{t_{i}}}^{v}$ | The distance between node $r_{r, \hat{t}_{i}}^{v}$ and restaurant ( $r$ ). |
| $\operatorname{dis}_{r, \widehat{t_{i}}}^{\eta v}$ | Mean distance of the $\eta$ nearest distances between the node in $\left\{\right.$ node $\left.e_{r, \hat{t}_{i}^{\prime}}^{v} v \in V\right\}$ and restaurant $(r)$. |
| $\text { StdDis } s_{r, \hat{t}_{i}}^{\eta v}$ | The standardized value of $\operatorname{dis}_{r, \widehat{t_{i}}}^{\eta v}$ by $\operatorname{dis}_{r, \widehat{t_{i}}}^{\eta v} / \max _{r \in R}\left(\operatorname{dis}_{r, \widehat{t_{i}}}^{\eta v}\right)$. |
| $M^{v}$ | The responsiveness of courier (v) to future orders. |
| $\lambda_{i}$ | Constant, $\lambda_{i} \in(0,1)$. |
| $\mu_{3}^{c}$ | The maximum emerging rate of customer orders. |
| $\Delta t$ | The promised consumption time, from the time order was placed to the time the order arrived at customer destination. |
| $\tau$ | The difference between the actual order preparation time and estimated order preparation time. |

```
Algorithm 3: The whole process of order dispatch scheme.
    A new customer ( \(c^{\text {new }}\) ) places an order at the restaurant(s) \(I_{c^{\text {new }}}\) at \(t_{\text {now }}\);
    if \(c^{n e w}\) choose non-split then
        \(c^{\prime}=c^{\text {new }} ;\)
    else
        \(c^{\prime}=\left\{c_{i}^{\text {new }}, i=1, \cdots,\left|I_{c^{n e w}}\right|\right\} ;\)
    for sc in \(c^{\prime}\) do
        for \(v\) in \(V\) do
            \(C_{v} \leftarrow C_{v} \cup s c\); // Assign sc to courier (v)
            \(\epsilon_{v}^{s C}, N R_{v}^{s c} \leftarrow\) Solve MILP; // Obtain objective function \(\epsilon_{v}^{s c}\) and new
                route \(N R_{v}^{s C}\) for courier (v)
        \(v^{\prime}, \epsilon_{v^{\prime}} \leftarrow \operatorname{argmax}\left(\left\{\epsilon_{v}^{s c}, v \in V\right\}\right) ; / /\) Find the courier with maximum \(\epsilon\) and
        record corresponding value of \(\epsilon_{v^{\prime}}\) for \(S C\)
        Use Algorithm 1 calculate \(M^{v}\);
        Use Algorithm 2 to get Candidate \(v\);
        if Candidate \(v==\{ \}\) then
            \(v^{*} \leftarrow v^{\prime}\)
        else
            \(v^{*} \leftarrow \operatorname{argmax}\left(\left\{\epsilon_{v}^{s c}, v \in\right.\right.\) Candidate \(\left.\left.v\right\}\right)\)
        return optimal courier ( \(v^{*}\) ) for new order ( \(s c\) ) and new route \(\left(N R_{v^{*}}^{s c}\right)\)
```


## 5. Experimental Analyses and Managerial Insights

In this section, to measure the performance of the proposed order dispatch scheme, as well as derive managerial insights for practical online food delivery operations, we conducted extensive numerical experiments. All the algorithm involved were coded in Python. By the way, we used Gurobi 9.0.3, under the Python environment, to solve the MILP, mostly taking less than one second for each courier. Besides, all the experiments were executed on a Dell Vostro 5468, with 12 GB of RAM and an Intel Core i5-7200U processor, under the Windows 10 environment.

### 5.1. Parameter and Dataset Settings

To elaborate the performance of the order dispatch scheme, random data sets are generated in this article. This subsection presents the parameter and dataset settings. It is noted that the setting of data is as consistent as possible with that of Steever et al. [6] for a better comparison.

The service area, where the restaurants are evenly located and customers appear randomly, is assumed as a square, with a side length of 5 km . In other words, the locations of restaurants and customers are uniformly distributed in $(0,5) \mathrm{km}$. The maximum number of restaurants in a single order is assumed to be no more than 3 . The distance between two points are calculated as $1.4 *$ Euclidean distances [13,30]. Additionally, the average speed of the courier $\bar{v}=25 \mathrm{~km} / \mathrm{h} ; \eta$ in Algorithm 1 is set as 3 . The service intervals $s_{i}$ $(i \in\{R \cup C\})$ are assumed as a fixed value of 3 min .

The time range is set as 4 h , and the time $t_{v}^{\prime}$ when the couriers start to work is equal to 0 ; and thus the delivery ends at $t_{v}^{\prime \prime}=4$. The customer arrival rate is set to the same with [6]. That is, as displayed by Figure 3 of [6], in the middle one-hour time zone $1.5 \leq t_{\text {now }} \leq 2.5$, customer arrival rate reaches the maximum emerging rate $\mu_{3}^{c}$, and in the first and last one hour, i.e., $0 \leq t_{\text {now }} \leq 1$ and $3 \leq t_{\text {now }} \leq 4$, the customer requests emerge at a rate that is half of the maximum rate. At other times, it is $75 \%$ of the peak demand. Each restaurant has a proportion for the order quantity of the food delivery company, as long as $\sum_{r \in R} P_{r}=1$. To facilitate the analysis of the problem, we set the order quantity of each restaurant similarly.

The average order preparation time of each restaurant $\left(t_{r}\right)$ is generated from a normal distribution. The difference between the actual and estimated order preparation time, indicated by $\tau$, is considered to simulate the uncertainty of reality. Consequently, the actual order preparation time equals to $\tau+t_{r}$. The $\tau$ is also generated from a normal distribution. As we know, for normal distribution, $99.7 \%$ of the possible values are included in the range around the mean, with a deviation not exceeding three times its standard deviations. However, theoretically, we may still encounter the situation of $\tau+t_{r}<0$ or $t_{r}<0$ with extremely low probability. Therefore, we add the constraints of $\tau+t_{r} \geq 0$ and $t_{r} \geq 0$ here. If a sample with $\tau+t_{r}<0$ or $t_{r}<0$ appears, we set $\tau+t_{r}=0$ or $t_{r}=0$, indicating that the food is ready to take without processing.

For the setting of penalty coefficient $\gamma$ of unit delay time, to compare with the proactive algorithm proposed by Steever et al. [6], we choose the same value with $\gamma=1$. To determine the two parameters of $\lambda_{1}$ and $\lambda_{2}$, we conducted some preliminary experiments where $\lambda=\{0.05,0.1,0.2,0.5\}$. Three pre-test data sets, shown in Table 3, were used to complete this preliminary experiments. The split probability (split prob.) equal to 1 means that the situation is totally split delivery. For each pre-test dataset, we generated ten instances to avoid contingency. Thus, there are $30(3 \times 10)$ instances for each $\lambda$.

Table 3. Parameter settings of pre-test datasets.

|  | Pre-Test Dataset 1 | Pre-Test Dataset 2 | Pre-Test Dataset 3 |
| :--- | :--- | :--- | :--- |
| $\|R\|$ | 10 | 10 | 20 |
| $\|V\|$ | 6 | 6 | 10 |
| $\mu_{3}^{c}$ | 10 | 10 | 24 |
| $S p l i t ~ p r o b . ~$ | 1 | 1 | 1 |
| $t_{v}^{\prime}$ | 0 | 0 | 0 |
| $t_{v}^{\prime \prime}$ | 4 | 4 | 4 |
| $\Delta t(\min )$ | 45 | 45 | 45 |
| $t_{r}(\min )$ | $N(15,0)$ | $N(15,2.5)$ | $N(15,2.5)$ |
| $\tau(\min )$ | $N(0,5)$ | $N(0,1.5)$ | $N(0,1.5)$ |

The purpose of developing this order dispatch scheme is to ensure delivery in time and workload balance, delay rate (DR), and unbalance index-standard deviation (UIsd) are applied to evaluate the influence of various $\lambda$ values on the scheme's performance. For each $\lambda$ value, the average value of DR and UIsd of 30 instances are depicted in Figure 3.


Figure 3. The influence of the $\lambda$ value on performance indicators.
From Figure 3, we can see that the DR shows an increasing trend when $\lambda$ (could be $\lambda_{1}$ or $\lambda_{2}$ ) increases from 0.1 to 0.5 . When $\lambda$ equals to 0.05 and 0.1 , the value of DR is relatively small, implying better performance of the proposed scheme. A similar conclusion can
be found in UIsd. Therefore, $\lambda_{1}=0.05, \lambda_{2}=0.1$ are chosen for implementing all of the following experiments.

For the following experiments, we build six data sets, considering two aspects referring to the size of the problem and difference in the preparation time of orders, between the estimated and actual time. The settings of the other parameters and requirements were as aforementioned. Table 4 outlines the parameter setting. Datasets 1 to 3 belong to small-scale problems, and datasets 4 to 6 are medium-scale problems. Due to the uncertain nature of this problem, 30 instances were generated for each data set. All instances used in this paper can be found in the online supplementary materials.

Table 4. Parameter settings of experimental datasets.

|  | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 | Dataset 5 | Dataset 6 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\|R\|$ | 10 | 10 | 10 | 20 | 20 | 20 |
| $\mu_{3}^{c}$ | 10 | 10 | 10 | 24 | 24 | 24 |
| $t_{v}^{\prime}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $t_{v}^{\prime \prime}$ | 4 | 4 | 4 | 4 | 4 | 4 |
| $\Delta t(\min )$ | 45 | 45 | 45 | 45 | 45 | 45 |
| $t_{r}(\min )$ | $N(15,2.5)$ | $N(15,2.5)$ | $N(15,2.5)$ | $N(15,2.5)$ | $N(15,2.5)$ | $N(15,2.5)$ |
| $\tau(\min )$ | $N(0,6)$ | $N(0,1.5)$ | $N(0,1.5)$ | $N(0,6)$ | $N(0,1.5)$ | $U(5,15)$ |

### 5.2. Performance of the Proposed Scheme

This subsection focuses on comparing the proposed scheme with the two algorithms proposed by Steever et al. [6]. One of the two algorithms is "Proactive", the other is "Myopic". "Proactive" attempts to seek the solution that is effective not only at present but also for an extended period. While, "Myopic" dispatches orders directly, according to the current optimal solution (maximum $\epsilon_{v}$ value), without considering the future demand.

For completing this comparative experiment, we consider six scenarios that the split order ratio takes value from $\{0,0.2,0.4,0.6,0.8,1\}$. For each scenario, we solve every instance once by using each algorithm. The performance of each algorithm is measured by DR, ALT, and UIsd. The average results are shown in Figures 4-6. The "D1V6" in Figure 4 indicates the result obtained using Datasets 1 and 6 couriers. The proposed scheme is abbreviated as "New".


Figure 4. Comparison of three algorithms in DR under different split order ratio (Datasets 1 to 4).


Figure 5. Comparison of three algorithms in ALT under different split order ratio (Datasets 1 to 4).
To avoid a cumbersome presentation, we only present part of the results, due to the trends of others that are similar. In the Figures 4 and 5, we can see that when the split order ratio is equal to 1 , the values of DR and ALT, given by the three algorithms, are the same, and the value is the smallest. Under other scenarios, the largest difference among the DR values, given by the three algorithms, does not exceed $2 \%$. It can be concluded that the values of the DR of three algorithms are slightly different. For ALT, the largest range among three algorithms, found in "D2V4" of Figure 5, when split order rate equals to 0.6, is not more than one minute. Thus, the performance of the three algorithms, in terms of ALT, also comes out even.





Figure 6. Comparison of three algorithms in UIsd under different split order ratios (Datasets 1 to 4).

Through Figure 6, it can be found that, compared to the myopic and proactive algorithms, the new algorithm, proposed in this paper, always obtained the smaller UIsd, remarkably so in datasets 1,2 , and 4 . Thus, the performance of the new algorithm in UIsd is better than that of the other two algorithms.

According to above discussion, the new algorithm outperforms the myopic and proactive algorithms in workload balance and is not inferior to them in delivery or time.

### 5.3. Sensitivity Analyses

### 5.3.1. The Impact of the Split Order Ratio

According to Figure 4, the DR has a remarkable and stable reduction with the increasing rate of split order. It may be because each new order can be assigned to a courier more flexibly when more customers choose split delivery. Correspondingly, the ALT roughly shows a reduction when the rate of split order varies from 0 to 1 in Figure 5.

In Figure 6, the UIsd values of Datasets 1 and 3 tend to reduce with increasing split order ratio, while in Datasets 2 and 4, there is no uniform trend. Therefore, the impact of the split order rate on workload balance cannot be generalized. However, the UIsd tends to be balanced with the increase of the split order ratio in small-scale problems.

### 5.3.2. The Effect of the Number of Couriers

For these six data sets, we analyzed the influence of the increase or decrease of the numbers of couriers on the indicators DR and ALT in two scenarios where the split order ratio is 0 and 1 , corresponding to non-split and split delivery. The results of Datasets 1, 2, and 4 are shown in Figures 7 and 8, for the reason that a complete display takes up too much space. The legend "D1_P = 0" represents the experiment running with Dataset 1 in a non-split delivery case.


Figure 7. Effect of couriers' quantity on delay rate.
Through analyzing Figures 7 and 8, in the non-split delivery case, the increase of couriers' quantity is not effective in reducing the DR and ALT. While in split delivery cases, the efficiency of the delivery system is obviously improved. However, on the whole, increasing the number of couriers can reduce the DR and ALT. Moreover, when the number of couriers is relatively scarce, the effect of increasing the number of couriers becomes obvious. When the couriers are not lacking, increasing the number of couriers has a weak positive influence.


Figure 8. Effect of couriers' quantity on the average late time for late orders.

### 5.3.3. Influence of Uncertainty on Delivery Efficiency

The uncertainty here refers to the order preparation time required by the restaurant, which we cannot accurately know in advance. Here, we discuss the influence of random variation on DR, UIsd, ALT, and AET. Dataset 4 describes the situation where there is a big gap between the estimated and actual time required to prepare the order, while Dataset 5 is the opposite. Dataset 6 expresses the case that the order preparation time that truly happened tends to be longer than the expected time. Figure 9 shows the performance of the four evaluation indicators in these three cases.


Figure 9. The impact of the difference between the expected and actual order preparation time on delivery efficiency.

As we can see, the closer the predicted value is to reality, the lower the delay rate and average late time for late orders will be. It shows that the accuracy of the prediction still plays an essential role in the effective distribution of orders. Moreover, with the increasing split order ratio, the DR and ALT will decrease relevantly. While, when the estimated time is accurate, the average early time of advanced orders will also show a relatively high value. Besides, the uncertainty has little effect on equity of the workload.

### 5.4. Managerial Insights

With the rapid development of O 2 O food delivery services, a large amount of orders are generated on the online platform every day. Efficient order dispatch and dynamic adjustment of delivery routes are critical to the success of the O 2 O platforms [31].

Following the commonly discussed O2O food delivery modes [1,6], this paper further discusses the situation that customers are enabled to order from multiple restaurants in a single order and have the higher autonomy to determine whether it be split, according to their preferences.

It is found that this business model is feasible, as Steever et al. proposed [6]; however, under the same conditions, comparing with the current split delivery form, the implementation of non-split needs to increase the number of couriers, otherwise the delay rate would be increased. However, as a service industry, the O 2 O food delivery would always put primacy on the service quality and customer satisfaction, while the demand of customers is diversified. So, the platform should recruit enough couriers to implement this business model.

Nevertheless, as illustrated by the simulation experiments, the effectiveness of increasing the number of couriers on improving customer satisfaction is relatively weak after it exceeds a certain threshold, regardless of non-split or split order delivery. Since excessive couriers not only increase the unnecessary labor cost but also has no significant effect on improving the efficiency of the distribution system, we only need an appropriate number of couriers when the business model is applied. The decision-makers can get a reasonable number of couriers with the aid of simulation analysis, combined with historical data.

If customers are given the option of non-split or split delivery, increasing the split delivery ratio of the system can effectively reduce the delivery pressure of couriers and increase customer satisfaction. That is to say, during the peak period, to cut down the volume of work for couriers and improve customers' satisfaction with low cost, without hiring more couriers, the company can induce customers to select split delivery by employing coupons, freight reduction, and other promotion policies. In short, to accurately satisfy a few customers' non-split preferences, the delivery efficiency needs to sacrifice, to a certain extent, under the conditions proposed in this paper.

## 6. Conclusions

Based on the VFCDP proposed by Steever et al. [6], this paper further discussed a new generation O 2 O food delivery service with order splitting choice. With the aim of delivering the orders on time, as well as balancing courier's workloads, we developed a three-stage order dispatch scheme, which includes pseudo-assign, re-route, and courier selection, in order to help the online platform efficiently dispatch orders and dynamically adjust couriers' delivery routes. Comparing it with the proactive and myopic algorithm of Steever et al. [6], the numerical experiments validate that the new scheme can significantly balance the couriers' workload, meanwhile keeping good performance in delivery efficiency.

Considering the O2O food delivery service is an emerging business model, there are still some issues to be addressed for its healthy development, which can be further studied in the future research, for example, introducing forecasting methods to predict customer requests and the time taken in the service process, as well as designing robust order dispatch schemes that are immune to the interruption of emergency events. Besides, considering the cooperation among couriers may be an interesting topic.

Furthermore, as mentioned in the managerial insights, customers choosing non-split delivery may bear some price increase, as a result of increased labor, compared to split delivery. It is valuable to investigate the impacts of payment structure on the customers' choice and further discuss the O2O food delivery mode by considering the influence of increasing payment for offering more flexible delivery services to customers.

Supplementary Materials: Supporting information is available online at https:/ /www.mdpi.com/ article/10.3390/jtaer17010015/s1.

Author Contributions: Conceptualization, K.W., Y.Z. and L.Z.; Data curation, Y.Z.; Formal analysis, K.W. and Y.Z.; Investigation, L.Z.; Methodology, K.W., Y.Z. and L.Z.; Software, Y.Z.; Validation, L.Z.; Visualization, Y.Z. and L.Z.; Writing-original draft, K.W., Y.Z. and L.Z.; Writing-review and editing, K.W., Y.Z. and L.Z. All authors contributed to all aspects of the study. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported, in part, by the grant from the National Key Research and Development Program of China (No. 2020YFB1708200).

Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.
Data Availability Statement: The datasets used for experimental study can be found in the online supplementary materials of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

## References

1. Roh, M.; Park, K. Adoption of O2O food delivery services in South Korea: The moderating role of moral obligation in meal preparation. Int. J. Inf. Manag. 2019, 47, 262-273. [CrossRef]
2. Chen, J.F.; Wang, L.; Wang, S.; Wang, X.; Ren, H. An effective matching algorithm with adaptive tie-breaking strategy for online food delivery problem. Complex Intell. Syst. 2021, 1-22. [CrossRef]
3. Liu, Y. An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. Comput. Oper. Res. 2019, 111, 1-20. [CrossRef]
4. Cosmi, M.; Oriolo, G.; Piccialli, V.; Ventura, P. Single courier single restaurant meal delivery (without routing). Oper. Res. Lett. 2019, 47, 537-541. [CrossRef]
5. Dsouza, D.; Sharma, D. Online food delivery portals during COVID-19 times: An analysis of changing consumer behavior and expectations. Int. J. Inov. Sci. 2020, 13, 218-232. [CrossRef]
6. Steever, Z.; Karwan, M.; Murray, C. Dynamic courier routing for a food delivery service. Comput. Oper. Res. 2019, 107, 173-188. [CrossRef]
7. Shen, C.W.; Chen, M.; Wang, C.C. Analyzing the trend of O2O commerce by bilingual text mining on social media. Comput. Hum. Behav. 2019, 101, 474-483. [CrossRef]
8. Nasrollahi, M.; Fathi, M.R.; Sanouni, H.R.; Sobhani, S.M.; Behrooz, A. Impact of coercive and non-coercive environmental supply chain sustainability drivers on supply chain performance: mediation role of monitoring and collaboration. Int. J. Sustain. Eng. 2021, 14, 98-106. [CrossRef]
9. Nasrollahi, M.; Fathi, M.R.; Sobhani, S.M.; Khosravi, A.; Noorbakhsh, A. Modeling resilient supplier selection criteria in desalination supply chain based on fuzzy DEMATEL and ISM. Int. J. Manag. Sci. Eng. Manag. 2021, 16, 264-278. [CrossRef]
10. Safari, H.; Etezadi, S.; Moradi-Moghadam, M.; Fathi, M.R. Maturity evaluation of supply chain procedures by combining SCOR and PST models. Int. J. Process Manag. Benchmark. 2021, 11, 707-724. [CrossRef]
11. Fathi, M.R.; Nasrollahi, M.; Zamanian, A. Mathematical modeling of sustainable supply chain networks under uncertainty and solving it using metaheuristic algorithms. Ind. Manag. J. 2020, 11, 621-652.
12. Jiang, G.; Wang, Q.; Wang, K.; Zhang, Q.; Zhou, J. A novel closed-loop supply chain network design considering enterprise profit and service level. Sustainability 2020, 12, 544. [CrossRef]
13. Ulmer, M.W.; Thomas, B.W.; Campbell, A.M.; Woyak, N. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. Transp. Sci. 2021, 55, 75-100. [CrossRef]
14. Yildiz, B.; Savelsbergh, M. Provably high-quality solutions for the meal delivery routing problem. Transp. Sci. 2019, 53, 1372-1388. [CrossRef]
15. Seghezzi, A.; Winkenbach, M.; Mangiaracina, R. On-demand food delivery: A systematic literature review. Int. J. Logist. Manag. 2021, 32, 1334-1355. [CrossRef]
16. Allen, J.; Piecyk, M.; Cherrett, T.; Juhari, M.N.; McLeod, F.; Piotrowska, M.; Bates, O.; Bektas, T.; Cheliotis, K.; Friday, A.; et al. Understanding the transport and $\mathrm{CO}_{2}$ impacts of on-demand meal deliveries: A London case study. Cities 2021, 108, 102973. [CrossRef]
17. Psaraftis, H.N.; Wen, M.; Kontovas, C.A. Dynamic vehicle routing problems: Three decades and counting. Networks 2016, 67, 3-31. [CrossRef]
18. Barkaoui, M. A co-evolutionary approach using information about future requests for dynamic vehicle routing problem with soft time windows. Memet. Comput. 2018, 10, 307-319. [CrossRef]
19. Hvattum, L.M.; Løkketangen, A.; Laporte, G. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transp. Sci. 2006, 40, 421-438. [CrossRef]
20. Ulmer, M.W.; Mattfeld, D.C.; Köster, F. Budgeting time for dynamic vehicle routing with stochastic customer requests. Transp. Sci. 2018, 52, 20-37. [CrossRef]
21. Ulmer, M.W.; Goodson, J.C.; Mattfeld, D.C.; Hennig, M. Offline-online approximate dynamic programming for dynamic vehicle routing with stochastic requests. Transp. Sci. 2019, 53, 185-202. [CrossRef]
22. Abbatecola, L.; Fanti, M.P.; Pedroncelli, G.; Ukovich, W. A distributed cluster-based approach for pick-up services. IEEE Trans. Autom. Sci. Eng. 2018, 16, 960-971. [CrossRef]
23. Lesmana, N.S.; Zhang, X.; Bei, X. Balancing efficiency and fairness in on-demand ridesourcing. Adv. Neural Inf. Process. Syst. 2019, 32, 5309-5319.
24. Bozorgi-Amiri, A.; Khorsi, M. A dynamic multi-objective location-routing model for relief logistic planning under uncertainty on demand, travel time, and cost parameters. Int. J. Adv. Manuf. Technol. 2016, 85, 1633-1648. [CrossRef]
25. Benjamin, A.M.; Rahman, S.A.; Hashim, F. Driver's workload analysis for waste collection vehicle routing problem. In Proceedings of the 2015 International Symposium on Mathematical Sciences and Computing Research (iSMSC), Ipoh, Malaysia, 19-20 May 2015; Publishing House: Ipoh, Malaysia, 2015; pp. 392-395.
26. Matl, P.; Hartl, R.F.; Vidal, T. Workload equity in vehicle routing: The impact of alternative workload resources. Comput. Oper. Res. 2019, 110, 116-129. [CrossRef]
27. Keskinturk, T.; Yildirim, M.B. A genetic algorithm metaheuristic for bakery distribution vehicle routing problem with load balancing. In Proceedings of the 2011 International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey, 15-18 June 2011; IEEE: Istanbul, Turkey, 2011; pp. 287-291.
28. Mao, W.; Ming, L.; Rong, Y.; Tang, C.S.; Zheng, H. Faster Deliveries and Smarter Order Assignments for an On-Demand Meal Delivery Platform. SSRN 3469015. 2019. Available online: https:/ /anderson-review.ucla.edu/wp-content/uploads/2021/03/ Mao-et-al_OnlineMealDelivery_2019_SSRN-id3469015.pdf (accessed on 25 November 2021).
29. Lehuédé, F.; Péton, O.; Tricoire, F. A lexicographic minimax approach to the vehicle routing problem with route balancing. Eur. J. Oper. Res. 2020, 282, 129-147. [CrossRef]
30. Boscoe, F.P.; Henry, K.A.; Zdeb, M.S. A nationwide comparison of driving distance versus straight-line distance to hospitals. Prof. Geogr. 2012, 64, 188-196. [CrossRef]
31. Zou, G.; Tang, J.; Yilmaz, L.; Kong, X. Online food ordering delivery strategies based on deep reinforcement learning. Appl. Intell. 2021, 1-13. [CrossRef]
