

Research Article

An Auction-Based Multiagent Simulation for the Matching Problem in Dynamic Vehicle Routing Problem with Occasional Drivers

Chung-Wei Shen ^{1,2}, Che-Cheng Hsu,¹ and Kuan-Hua Tseng¹

¹Department of Transportation and Communication Management Science, National Cheng Kung University, 1 University Road, Tainan City, Taiwan

²Center for Innovative FinTech Business Models, National Cheng Kung University, 1 University Road, Tainan City, Taiwan

Correspondence should be addressed to Chung-Wei Shen; tsungweishen@gmail.com

Received 4 April 2022; Revised 17 August 2022; Accepted 8 October 2022; Published 27 October 2022

Academic Editor: Michela Le Pira

Copyright © 2022 Chung-Wei Shen 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.

This research incorporated an auction mechanism into the vehicle routing problem with occasional drivers and produced simulations in an agent-based environment. Auctions were used to match online orders with potential occasional drivers. While a centralized system optimizes system performance under global objectives, the novel decentralized approach presented here illustrates emergent phenomena resulting from the interaction of individual entities in highly dynamic cases. In the simulations, the auctions were executed after a fixed time interval called a rolling time horizon. Our results suggest that the appropriate rolling time horizon produces a lower average unit compensation cost because better matches can be found when the accumulation of online orders and occasional drivers is maintained at a certain level. The simulation results also indicate that the use of an auction mechanism instead of simple nonauction rules can improve the average unit compensation cost by up to 25.1%.

1. Introduction

E-retail, with its revolutionary business model, is radically changing the way people shop. One of the key success factors of e-retail is satisfaction with last-mile delivery service [1]. With the ever-growing need to expedite last-mile delivery to meet consumers' expectations, retailers are increasingly eager to adopt innovative delivery service mechanisms.

One such innovative delivery service is “crowdsourced delivery,” in which orders can be picked up and delivered throughout a region by ordinary people using their own vehicles. Internet-enabled mobile technology facilitates last-mile crowd delivery services such as Kanga, Deliv, Door-Dash, and GoLocal [2]. The ordinary people who become couriers are called occasional drivers (ODs), and their interaction with such technology has yielded a new variant of the vehicle routing problem (VRP) called the vehicle routing problem with occasional drivers (VRPOD). ODs can be

categorized into two main types: (1) in-store customers and (2) ad-hoc drivers. These two types of ODs are similar in that they are both willing to make deliveries along a route from a specific origin to a destination. Thus, both are willing to detour from their original trips to deliver goods. The difference between the two is that in-store customers all start from the same origin (e.g., the retail store) and ad-hoc drivers have different origins (e.g., home or wherever they were when they chose to participate in the delivery service). For instance, Walmart asks in-store customers to help deliver online orders on their way home from shopping, and thus they belong to the first type [3].

The present research explores a delivery service platform that retailers can use to identify in-store ODs willing to make deliveries using their own vehicles when on their way home from shopping. The participating in-store ODs offer prices for each delivery task if the extra travel distance required is within a certain percentage of the direct travel distance from

store to home. While this is an effective way to deliver online orders, supplementary company-based drivers and vehicles are also necessary to collect and deliver online orders that have been waiting beyond a specific time threshold and have not yet been matched with ODs.

In crowdsourced delivery services, one core issue is how to match online orders with ODs [4]. There are static and dynamic versions of the matching problem. If all the information is known and will not change before the service starts, the problem is static. However, if new information may be revealed during the service time, the case is dynamic; the latter is more common in the real world. Some studies have pursued optimal solutions for crowdsourced delivery under static conditions, while others have investigated the dynamic context. In both, centralized decision-making is the most common method adopted. In the highly dynamic environment of crowdsourced delivery, purely centralized optimization requires longer computation times, and thus, some information can change during the matching process. Therefore, the optimal solution at a certain time point may not be valid even a few seconds later.

Agent-based modeling is another research stream evaluating service performance under different configurations of the dynamic crowd-based shipping issue. Agent-based simulations are modeled as individual decision-making entities called agents [5]. Agents execute their own behaviors inspired by the system they represent and the results of interactions of individual agents. Therefore, they can be used to investigate the macro-level phenomena emerging from microlevel behaviors affected by constraints or local objectives. This bottom-up simulation method has been widely used for phenomena such as traffic flow, the spread of disease, and the emergence of social norms. We used agent-based modeling to explore the possibility of incorporating auctions into crowdsourced delivery, and studied the interactions among individual agents and their effects on the patterns of practical-sized problems.

The contributions of this research can be summarized in the following three aspects. First, we considered a purely dynamic operational environment where ODs and online orders are not known in advance. Second, from the methodological point of view, we developed a decentralized agent-based model to capture the emergence of system-level features in practical problems. The system-level features considered included cost, service level, and utilization rate. The problem being investigated was a matching problem that paired the demand (i.e., online orders) with the supply (i.e., occasional drivers) for a crowdsourced delivery service. Third, this work is the first to propose an auction-based matching mechanism for ODs and online orders within the context of the dynamic VRPOD.

The remainder of this research is structured as follows. In Section 2, we summarize the relevant literature. In Section 3, we describe the dynamic VRPOD setting and explain the planning horizon. In Section 4, we present an agent-based simulation model. In Section 5, we focus on understanding the model's performance under different parameter settings. Finally, in Section 6, we summarize the main insights and discuss directions for future research.

2. Literature

Crowdsourced delivery applies the concept of crowdsourcing to logistics [6]. According to the structure proposed by Nourinejad and Roorda [7], there are two types of methods for dealing with the evaluation of crowdsourced delivery in last-mile logistics: centralized and decentralized. The former uses a central controller, to which all information is provided. This central controller makes decisions based on the global objective function of the system. In contrast, in the decentralized agent-based setting, decisions are made locally with incomplete information limited by time or space. Each agent behaves according to their default principles, according to a process initiated once the necessary criteria are met. In Sections 2.1 and 2.2, we discuss the recent literature relevant to this study as divided according to these two categories.

2.1. Centralized System Optimization. The VRPOD was formally introduced and formulated by Archetti et al. [8]. These researchers used a traditional centralized optimization approach to solve the VRPOD. Their objective function was to minimize the travel cost of regular vehicles (in proportion to the travel distance) and compensation paid to ODs. They considered the static case of the VRPOD, and no time window was proposed. Subsequently, Macrina et al. [9] included time windows for both customers and ODs and allowed for multiple deliveries. In addition to successfully incorporating the time window into the VRPOD, they also verified that multiple deliveries could increase cost savings. Triki [10] proposed two heuristic methods for solving the static case of incorporating combinatorial auctions into the VRPOD. In the proposed centralized decision-making tool, a global objective function is included to minimize the sum of the company vehicles and winning bid costs. The author suggested that the dynamic planning of the problem's resolution would be a promising future avenue of research on this topic.

As mentioned above, a natural extension of the static case is the introduction of a dynamic setting. The dynamic component of the VRPOD can include not only the demand (i.e., online orders) but also the supply (i.e., ODs). An online vehicle routing problem with occasional drivers was explored by Dayarian and Savelsbergh [3]. In their dynamic version of the VRPOD, the authors developed a heuristic called a Tabu search to rapidly generate a solution at each decision epoch. Their method was centralized because choices regarding whether the online orders would be served by available ODs or regular vehicles were made by collecting all the information and minimizing the total lateness of deliveries and overall cost induced by each option. Also examining a dynamic environment, Archetti et al. [11] separated orders into two classes: static and online. They proposed a centralized online re-optimization approach in which re-optimization was triggered when new online orders arrived. It is worth noting that the researchers assumed that the ODs were known in advance. In the dynamic ridesharing problem, matching drivers and riders is similar

to matching ODs and online orders in the VRPOD. In the dynamic ridesharing problem, in which a service provider matches thousands of potential drivers and passengers with similar itineraries, the matching must be fast enough to be of use in practice [12]. Thus, designing a selection-based matching process is a nontrivial endeavor [12].

Recently, several studies have explored centralized matching between carriers and tasks. Allahviranloo and Baghestani [4] investigated the impacts of crowdshipping on travel behavior. They formulated a mathematical model to find the optimal allocation of tasks to each carrier. Le et al. [13] designed several pricing and compensation schemes and developed a matching model to maximize the crowdshipping platform providers' benefits. The above two studies both focus on optimizing matching of the service. Matching between ODs and online orders can be seen as a crowd-sourcing contest. Using game theory, Hou and Zhang [14] modeled the game between a contest seeker and participants and derived the equilibrium results.

A variant of the VRPOD called the pickup and delivery problem with occasional drivers (PDPOD) considers situations in which multiple pickup and delivery operations are allowed for a single OD. Arslan et al. [15] considered a dynamic PDPOD and presented a rolling horizon framework that repeatedly solved the problem of matching online orders to ODs. ODs might not accept an assignment if the compensation is too low. Thus, compensation schemes were included and discussed in the context of the PDPOD by Dahle et al. [16]. In the same research stream as the PDPOD, the impact of allowing for task transfer between ODs was investigated by Voigt and Kuhn [6]. Their numerical results showed that the introduction of trans-shipment points increased the utilization of ODs and significantly reduced total cost.

Our study focused on the effects of including a rolling horizon in the matching problem instead of improving the vehicle routing problem. Here, we elaborate upon how our research contributes to auction-based matching problems. The auction-based matching problem in crowdsourced delivery mainly has two types of users of the platform: a requester who asks for the service and a worker who helps to fulfill the request [17]. In existing studies of crowdsourced delivery, matching problems attempt to minimize total routing costs [15, 18]; maximize utility, such as the total number of assigned tasks [19]; or maximize the platform provider's profits [13]. An excellent review of the matching problem was offered by Tong et al. [17]. To the best of our knowledge, during the matching process, requesters' concerns (e.g., the cost per task for the service) have yet to be investigated. The inclusion of a rolling horizon into the matching problem potentially offers benefits to requesters, but the quantitative evidence is lacking. Consequently, there is a need for a systematic evaluation of the incorporation of a rolling horizon into dynamic matching problems.

2.2. Decentralized Agent-Based Approach. The decentralized approach is comprised of a collection of autonomous decision makers in the system called agents [7]. The agent-

based approach is an ideal test environment for evaluating new crowdshipping proposals [20]. The VRPOD is suitable for the agent-based approach due to the heterogeneity of ODs. Each OD may have different individual preferences when offering their service to deliver online orders. For example, the minimum compensation amount for the same extra travel distance may be different for two ODs if they have different compensation rates. Another characteristic of the agent-based approach is that all agents independently make rule-based decisions as they interact with other agents and their simulated environment [21].

Only a few studies have explored crowdsourced delivery performance using an agent-based approach. Chen and Chankov [22] considered the crowdsourced last-mile delivery problem and developed an agent-based simulation model to evaluate the factors influencing performance. The results showed that a higher supply (i.e., couriers) to demand (i.e., packages) ratio helped to increase the percentage of delivered packages but decreased crowd utilization. In addition, higher maximum detour times increased the number of packages delivered but also increased the chance of competition among couriers, resulting in a negative impact on crowd utilization.

The literature most relevant to our study was conducted by Castillo et al. [23]; who considered the same-day delivery problem and combined company-owned vehicles with crowdsourced drivers. The results of their agent-based simulations showed that the low and high compensation could negatively affect cost performance, and thus a moderate compensation amount for crowdsourced drivers was important for the success of the crowdsource-based resolution to the last-mile delivery problem.

Although the abovementioned two studies provided important insights into the factors influencing crowdsourced delivery, two significant issues have yet to be thoroughly investigated. First, the competition between ODs when there are multiple online orders and ODs has not yet been considered. Competition between ODs will affect the probability of acceptance as well as have consequences for the last-mile performance. The impact of competition under different time horizons has also yet to be investigated. The second primary issue is that the performance results of matching with competition have not been compared to those without competition. Matching with competition means that the final matching can be postponed to the last minute. Conversely, matching without competition exemplifies a grab-and-go policy. We propose an agent-based model to provide such comparisons.

3. Problem Description

Figure 1 shows the basic structure of the same-day delivery problem considered here. We assume a retail store in the center of an area with two types of customers: in-store (C_I) and online (C_O). Some of the in-store customers are willing to deliver online orders on their return trip home. These customers are referred to as ODs.

Once the ODs arrive at the store, they use an app on their smartphone to announce their willingness to participate. At

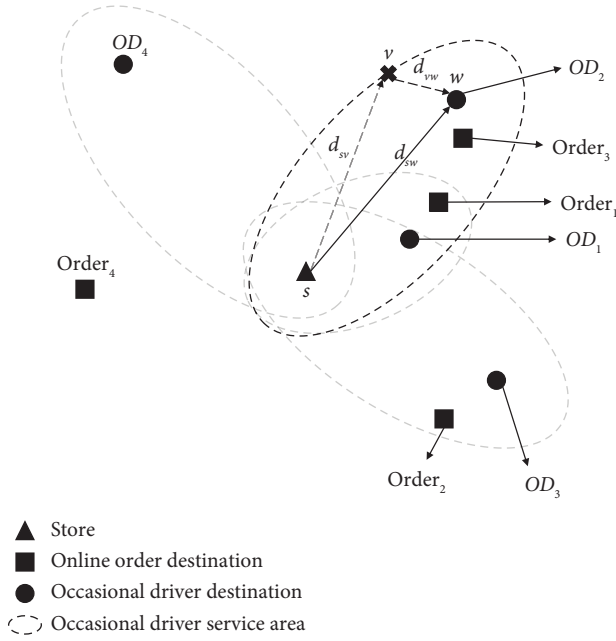


FIGURE 1: Vehicle routing problem with ODs.

that time, they also identify their earliest and latest departure times from the store. The minimum shopping duration, representing the time gap between the arrival time and the earliest departure time, is denoted by D_{OD}^{\min} . The maximum shopping duration, representing the time gap between the arrival time and the latest departure time, is denoted by D_{OD}^{\max} . The possible matching duration for the OD, D_{OD}^{match} , is the time difference between the maximum and minimum shopping hours. Because the OD also needs to shop, they are not included on the OD list if the current time is within their minimum shopping hours. Once the time passes beyond their earliest departure time, the OD is included on the OD list and they are allowed to offer a bid to the matching platform. The system automatically identifies unmatched online orders whose destinations fall within the geographical coverage of that OD. They are informed by the app if the OD receives the match decision during the possible match time. We assume five minutes for picking up the online order at the exit of the store and beginning delivery.

Each OD has a geographical coverage area determined by detour rules. Different detour rules can be made to determine the coverage area. It has previously been shown that ODs are often more willing to deliver online orders to the periphery of the direct path between the store and their home location [3]. In other cases, each driver may have specific coverage areas in which delivery tasks are near their home. This is likely because the OD is more familiar with that neighborhood's destinations. We adopted the first rule, in which detours are allowed along the original route.

Following the detour rule described above, each OD can define its service area. Take OD_2 in Figure 1 as an example, OD_2 starts from the store s and ends at destination w . Let d_{pq} represent the direct distance between nodes p and q and τ_{OD} denote the detour ratio, where $\tau_{OD} \geq 0$. It is important to

note that in practice, each OD could have different maximum detour rates. The point v must satisfy $d_{sv} + d_{vw} = (1 + \tau_{OD})d_{sw}$. The collection of all points v produces an ellipse. Therefore, the online orders 1 and 3 falling within the service area of OD_2 can be served by this OD_2 .

Having found potential online orders, each OD_j then, bids a price Bid_{OD_j} on those desired. This price is calculated using a given compensation function. An OD's compensation is composed of two parts: fixed compensation c_i^f and extra distance compensation c_i^e . The fixed compensation is paid to any OD delivering at least one order. The extra distance compensation, also called the compensation parameter, is proportional to the extra distance they need to travel to reach their destination. Our setting avoids situations in which routes correspond too well (i.e., the extra distance is close to zero). In such cases, the compensation offered would be too small for an OD to be interested. The OD receives the compensation once the order is received.

Online orders enter the platform once the customer sends the purchase request through the Internet. This is called the announcement time of the online order (see Figure 2). The request indicates the ordered items and destination. The store must then spend a certain amount of time collecting the ordered items. This is called the lead-time for preparation. Once the order is collected, it is ready to be sent out; therefore, we used this ready time as the earliest pickup time. Each online order remains on the online order list until the latest matching time. Using the latest match time implicitly guarantees the minimum service level. The time between the earliest pickup and latest the match times forms the time window for matching $TW_{O_i}^M$ for each online order.

After seeing the candidate orders, the OD proposes bid prices for all desired. The price is computed based on the OD's compensation parameter c_i^e and requires extra travel distance to return to their ultimate destination. On the platform side, if the offers made by the OD are less than the maximum amount that the store is willing to pay C^{\max} , those offers are considered by the platform as potential matches and added into the bid list for the online order. All potential bids are sorted in the order of increasing compensation so that the first match has the lowest amount. Based on the example provided in Figure 1, the bid list for each online order is shown in Table 1.

In sum, the OD bid price is established for each online order if the following two conditions are sequentially met. First, the compensation parameter of the OD needs to be less than the compensation parameter provided by the store. Then, the platform allows online orders to be bid upon if the orders are within the OD's coverage area.

In our problem, the ODs and online orders are revealed over time. This dynamic nature makes matching the two a nontrivial issue. In a dynamic environment, a rolling horizon approach is often adopted in which plans are made using all known information within the planning time horizon. After each match, the planning horizon is "rolled" forward and the process continues [24]. Therefore, a key decision when implementing a rolling horizon approach is how frequently, and specifically when, to execute the

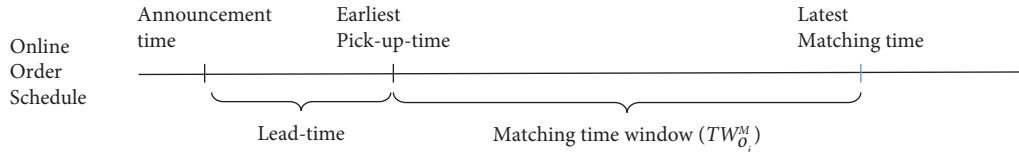


FIGURE 2: Matching time window for online orders.

TABLE 1: Example bid list for the online order in Figure 1.

| Online order list | Bid list |
|--------------------|---|
| Order ₁ | Bid _{OD₁} Bid _{OD₂} |
| Order ₂ | Bid _{OD₃} |
| Order ₃ | Bid _{OD₂} |
| Order ₄ | None |

matching procedure. At each iteration q of the rolling horizon, we determine the matches based on the current information available to the system at that point in time. Decisions are made at every δ time during the planning horizon.

We developed Figure 3 based on the case from Figure 1. In Figure 3, there are three predefined match times: 10:40, 11:20, and 12:00. Each OD had a 20-minute minimum shopping duration D_{OD}^{\min} and a 40-minute possible matching duration D_{OD}^{match} . The time window of matching $TW_{O_i}^M$ for each online order was 120 minutes. For the sake of simplicity, we did not include online order 4. In this case, at the predefined first match time of 10:40, three available ODs and three online orders were *active*, and thus, they could be considered in the match decision.

Concerning the match decision, there are two types of commitment strategies for each OD: earliest and latest. In our base case, we applied the latest commitment strategy for the match. Using this strategy, the potential matches found via auction are not finalized until the last possible match time. The advantage of the latest commitment strategy is that it increases the chance of finding better matches because better online orders might still appear; however, the latest commitment can also increase the delivery time (which is the time between the announcement and the actual receipt).

Besides in-store customers' help delivering online orders, a dedicated fleet was also included to deliver online orders not assigned to in-store customers. At the designated departure time of the regular vehicle, the unmatched online orders are ranked according to the remaining time. The remaining time is calculated between the current time and the end of the time window. Those orders with less than the predefined remaining time are then delivered by the dedicated fleet. More specifically, to better consider the urgency of the orders, all orders were divided into two categories [3]. For each order i , we define θ_i^{LT} as the latest departure time from the store. If the time gap between the current time t and latest departure time θ_i^{LT} is less than the predefined remaining time θ^{\min} , the order is designated an urgent order ($|t - \theta_i^{LT}| < \theta^{\min}$ if $t < \theta_i^{LT}$). When the predefined departure time of a regular vehicle is reached, the urgent orders are

selected for delivery. The route of the vehicle is then constructed using the greedy insertion method.

4. Agent-Based Model

The proposed agent-based model was designed to simulate the dynamic VRP with ODs. The model functioned using four main types of active actors (or agents): online orders, ODs, regular vehicles, and a bid-matching platform. Online orders and ODs had heterogeneous characteristics such as different destinations (for online orders) and compensation parameters (for ODs). Regular vehicles were comprised of the company-owned fleet and we assumed that these regular vehicles were homogeneous (though we could easily modify them for heterogeneous cases). Lastly, the bid-matching platform arranged the allocation of online orders to ODs. The interaction of the individual OD, rolling horizon, and regular fleet comprised the main themes of the proposed simulation model.

The entire simulation procedure began with time t_0 , which was the start of the run time (see Figure 4). The online orders and ODs were subsequently revealed over time. Every time a new online order or OD entered the system, the lists of online orders and ODs were updated accordingly. Next, for each new entering OD, the bid for each qualified order was added to the current bid list. At every scheduled auction match time (or epoch) an auction-matching procedure is initiated. Next, the regular vehicle selection procedure selected urgent online orders and arranged the delivery route. Finally, the simulation time was advanced by a time lapse of Δt ticks. If the time at the end of the simulation was reached, it stopped; otherwise, new online orders and ODs were generated and the above procedure was repeated.

The flowchart in Figure 4 illustrates the concept of the rolling horizon approach along with the auction procedure. The corresponding pseudocode is provided by Algorithm 1. In the simulation, a fixed-increment time advance approach (i.e., rolling horizon) was used instead of a next-event time advance approach so that a certain number of ODs and online orders could be accumulated [7].

From the system perspective, the auction procedure made matches between online orders and ODs. The auction dynamics presented in Figure 5 are described below and the corresponding pseudocode is presented in Algorithm 2.

- (1) At the time the predefined matching time arrives, the auction procedure is initiated by retrieving the current bid list. Setting i to 1 means the first online order on the list is reviewed.
- (2) For current online order i , the first (or best) item (i.e., OD) on the list of potential offers is assigned to the

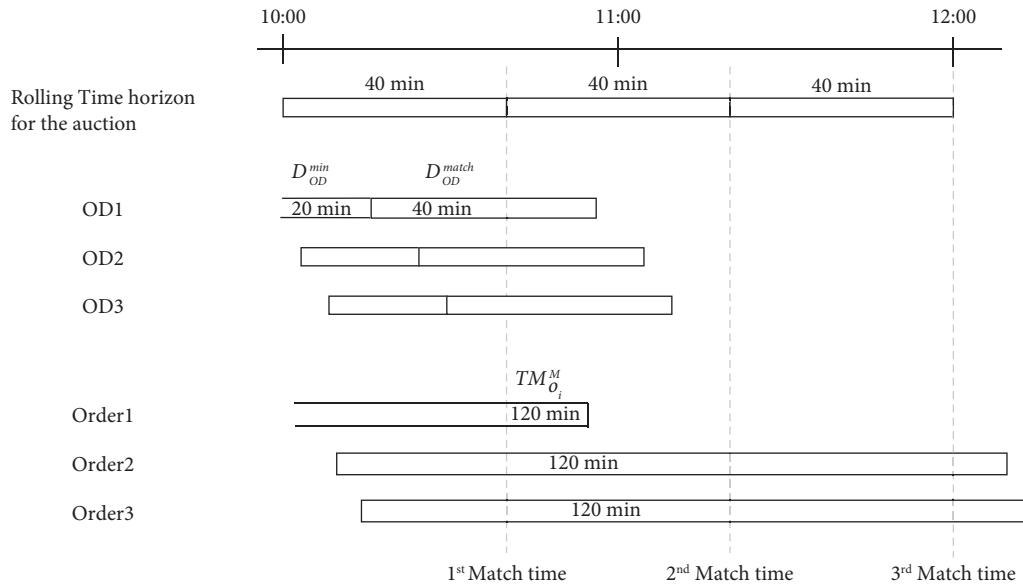


FIGURE 3: Relationships among rolling time horizons, ODs, and online orders at one match time point.

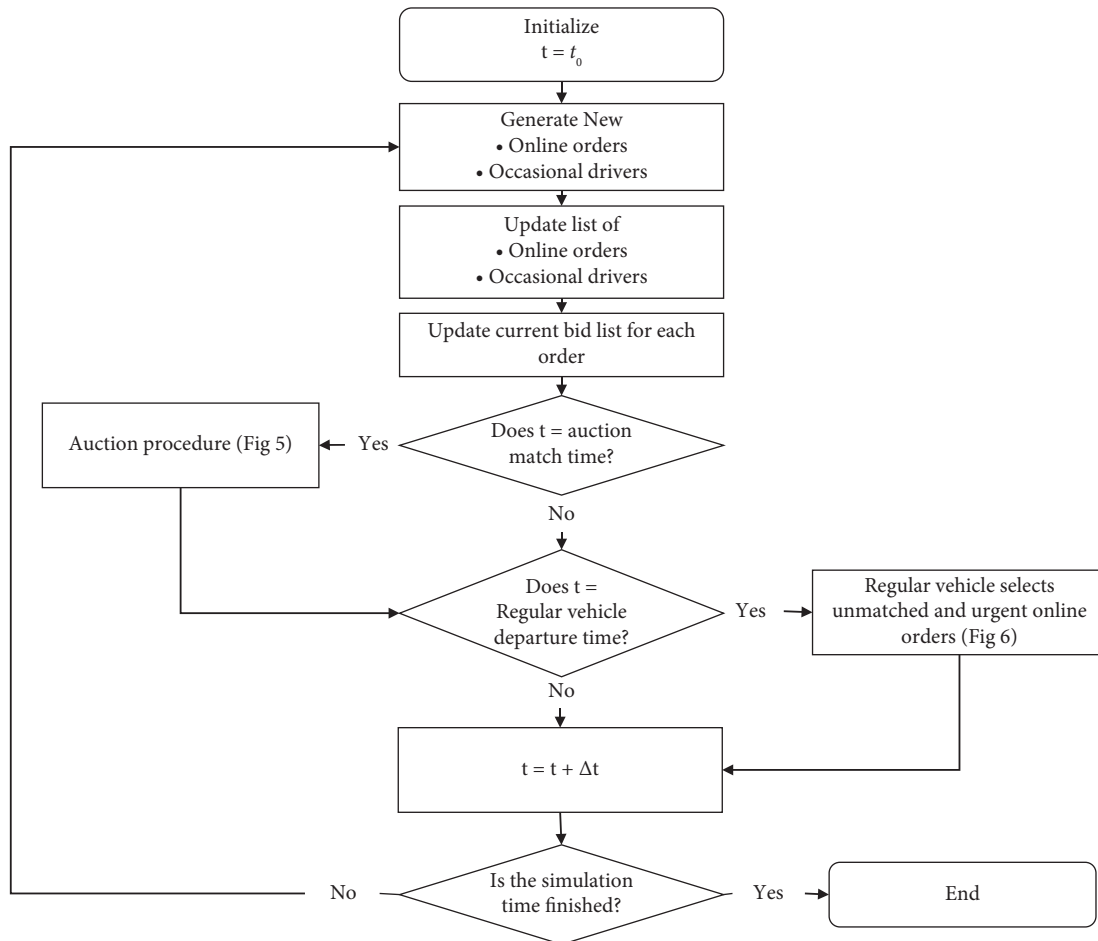


FIGURE 4: Flowchart of the simulation procedure.

```

(1) Initialize and let  $t = t_0$ 
(2) while the clock time  $t$  is not greater than the end time do
(3)   Generate new online orders and occasional drivers
(4)   Update the list of online orders and occasional drivers
(5)   Update current bid list for each online order
(6)   if  $t =$  scheduled auction match time then
(7)     Run auction-based matching procedure
(8)   end if
(9)   if  $t =$  scheduled regular vehicle departure time then
(10)    Run regular vehicle routing procedure
(11)  end if
(12)   $t = t + \Delta_t$ 
(13) end while

```

ALGORITHM 1: Whole simulation procedure.

online order. The best offer must then, be checked regarding whether the price is lower than the total compensation limit. The best offer is not accepted if the cost of the offer is more than the compensation limit.

- (3) If there are multiple ODs with the same bid prices, the OD with the fewest potential offers is chosen.
- (4) Next, the capacity of the chosen OD is checked. If the capacity has not reached the permitted limit, the assignment is finalized; otherwise, the process proceeds to the next-best item.
- (5) The next unmatched online order i on the online order list is checked. The process stops when all orders are checked.
- (6) The matched ODs exit the OD pool, and the unmatched ODs remain in the ODs pool.
- (7) The auction terminates if all online orders are checked.

A regular vehicle was still needed for cases in which no OD could be matched to certain online orders. Consequently, company-owned drivers and vehicles were used to complement the ODs. A flowchart describing the selection of urgent online orders to be served by regular vehicles can be found in Figure 6 and the pseudocode is shown in Algorithm 3. Urgent online orders are defined as follows. If an online order is currently unmatched and the remaining time available for matching is less than the predefined remaining time, or the current time is greater than the latest departure time from the store, the order is defined as an urgent online order. All urgent online orders are considered to be delivered by the regular fleet. Once the urgent orders were identified, they were dispatched using regular vehicles. The routes of the regular vehicles are generated using a greedy insertion method to solve the routing problem. The objective of the routing problem is to minimize the total travel distance that regular vehicles will travel. Here, we did not consider any other constraints, such as deadlines for tasks or vehicle capacity. The greedy insertion method was used to build the regular vehicle routes. The greedy insertion method includes the

following steps. First, we generate an empty route that start and ends at the depot. Second, we select the online order with the least extra travel distance to the existing route and insert it into the route. Third, we repeatedly execute the second step until no urgent online orders remain. The cost of each regular driver's route was the sum of the lengths of the arcs for the route. Note that even though we specified the latest departure time for each online order, the actual departure time might have been later because if the online order was not categorized as urgent, it was not selected for delivery by a regular vehicle. The online order remained in the system. However, it was still possible that the online order would not later be matched with an OD. Therefore, it would then be served by the next departure of a regular vehicle at an actual departure time later than the latest departure time.

In addition to the auction matching and regular vehicle dynamics shown in Figures 5 and 6, another important agent dynamic of the ODs is reflected in Figure 7 (the pseudocode is given in Algorithm 4). ODs have the autonomy to decide their compensation rate, which is a function of the normal distribution. In cases in our simulations in which an OD's compensation parameter was lower than the maximum compensation rate, the OD bid for online orders. If the OD received match confirmations after the scheduled auction match time, the OD could then pickup and deliver those orders; otherwise, the OD waited at the retail store until the latest leave time.

5. Numerical Experiments

As mentioned in the introduction, the purpose of this study was to gain quantitative insights into the potential benefits of auction-based decentralized agent-based responses to the VRPOD. In order to accomplish this goal, we evaluated a set of randomly generated instances. Even though we did not use real-life data, we feel that the results could be considered general, at least in terms of trends. In the remainder of this section, we introduce how we generated our instances, describe what we gained from the auction mechanism, and discuss what we learned regarding the sensitivity to problem characteristics such as the flexibility of the ODs, their vehicle

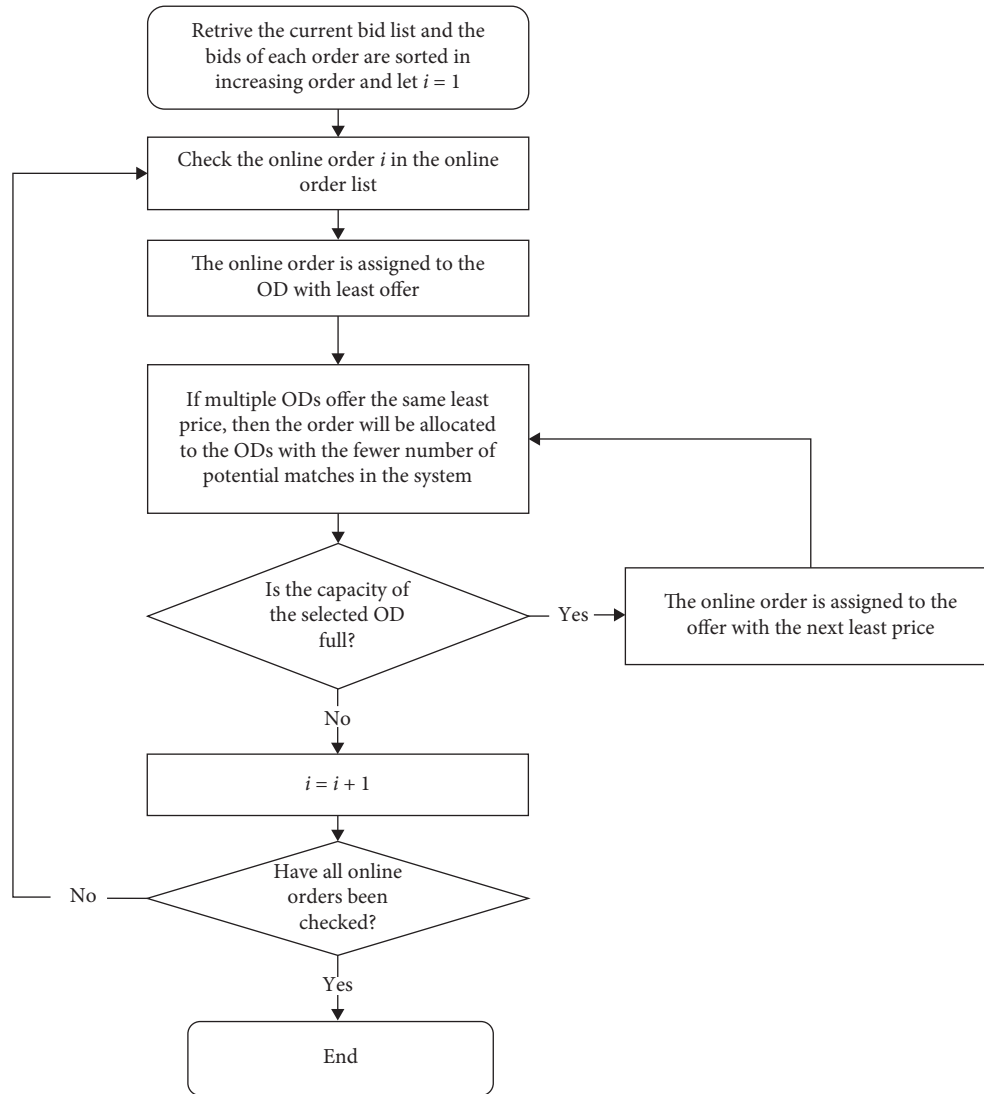


FIGURE 5: Auction-based matching dynamics.

```

(1) Input: current bid list in ascending order for each online order  $i$ 
(2) for  $i=1$  to  $n$  do
(3)   Select  $OD_j$ , the lowest bidder, and if not yet reached the capacity limit
(4)   if more than one  $OD$  offers the same bid then
(5)     order  $i$  is assigned to the  $OD_j$  with the fewer number of bids
(6)   end if
(7) end for
(8) Output: matching results
  
```

ALGORITHM 2: Auction-based matching procedure.

capacity, frequency of bidding, and compensation threshold. We also investigated the dispatching policy for regular vehicles and the effects of different geographic distributions of online orders. All experiments were implemented in NetLogo and conducted on a 3.4 GHz Intel Core i7 processor and 8 GB RAM.

5.1. Instance Generation. We generated instances by beginning with a 10 km by 10 km square region. We further assumed that a depot for the regular vehicles was situated at the center of the square region, at [5, 5]. The store was also located at the center of the region, meaning that all the ODs had the same origin.

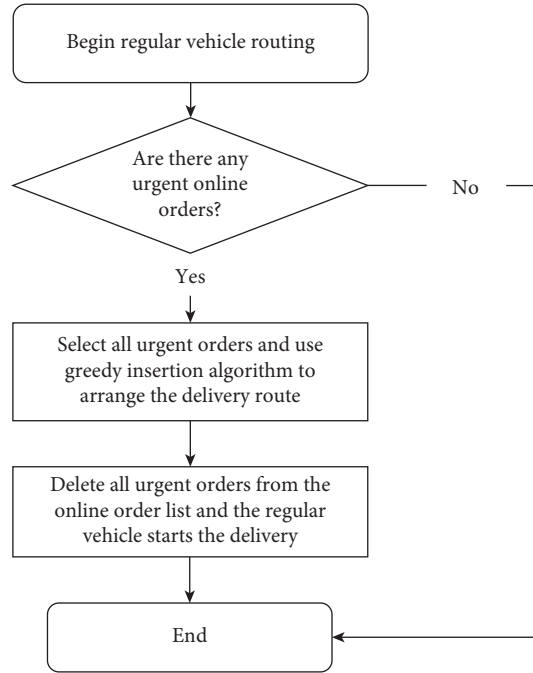


FIGURE 6: Regular vehicle dynamics.

(1) **Input:** Current list of unassigned online orders
 (2) Set θ^{\min} = remaining time threshold
 (3) **for** $i = 1$ to n **do**
 (4) Set θ_i^{LT} = latest departure time from the store for online order i
 (5) **if** $(|t - \theta_i^{LT}| < \theta^{\min} \text{ or } t > \theta_i^{LT})$ **then**
 (6) Mark order i as an urgent online order
 (7) **end if**
 (8) **end for**
 (9) Use greedy insertion algorithm to generate the route for urgent orders
 (10) Delete all urgent orders from the current order list
 (11) **Output:** Delivery route for urgent online orders

ALGORITHM 3: Regular vehicle routing procedure.

Note that all instances in our base-case simulation had 100 online orders and 100 ODs. To obtain the test instances for the VRPOD, the online order locations were identified by coordinates $[x_i, y_i]$, which were generated from Solomon instances. We generated destinations for the ODs in a uniformly random fashion in the squares at the lower left $[0, 0]$ and upper right $[10, 10]$ corners. The times of the online orders and ODs' arrivals were uniformly randomly generated within the total simulation time span. We used Euclidean distances and assumed a constant speed of 25 km per hour for the regular fleet. The speed of the ODs was 20 km per hour. The reason that the ODs were made to be slower was that those drivers would not have been familiar with the road network, a condition that usually reduces travel speed.

Here, we assume that the numbers of regular vehicles and occasional drivers are fixed. Furthermore, because the cost of a

regular vehicle and bids for occasional drivers are both only proportions of the distance traveled, the speed of regular vehicles and the number of occasional drivers will only affect the lateness ratio and average fulfillment time for an online order.

All instances were evaluated based on the following metrics:

- (1) *Cost for regular vehicles* (C_R): The total cost of the routes performed by regular vehicles; the cost is proportional to the distance traveled.
- (2) *Cost for occasional drivers* (C_{OD}): The total compensation cost for the OD.
- (3) *Total delivery cost* (TC): The total delivery cost, a combination of the above two items.
- (4) *Lateness ratio* (R_{late}): The number of late departures divided by the total number of online orders (reported as a percentage).

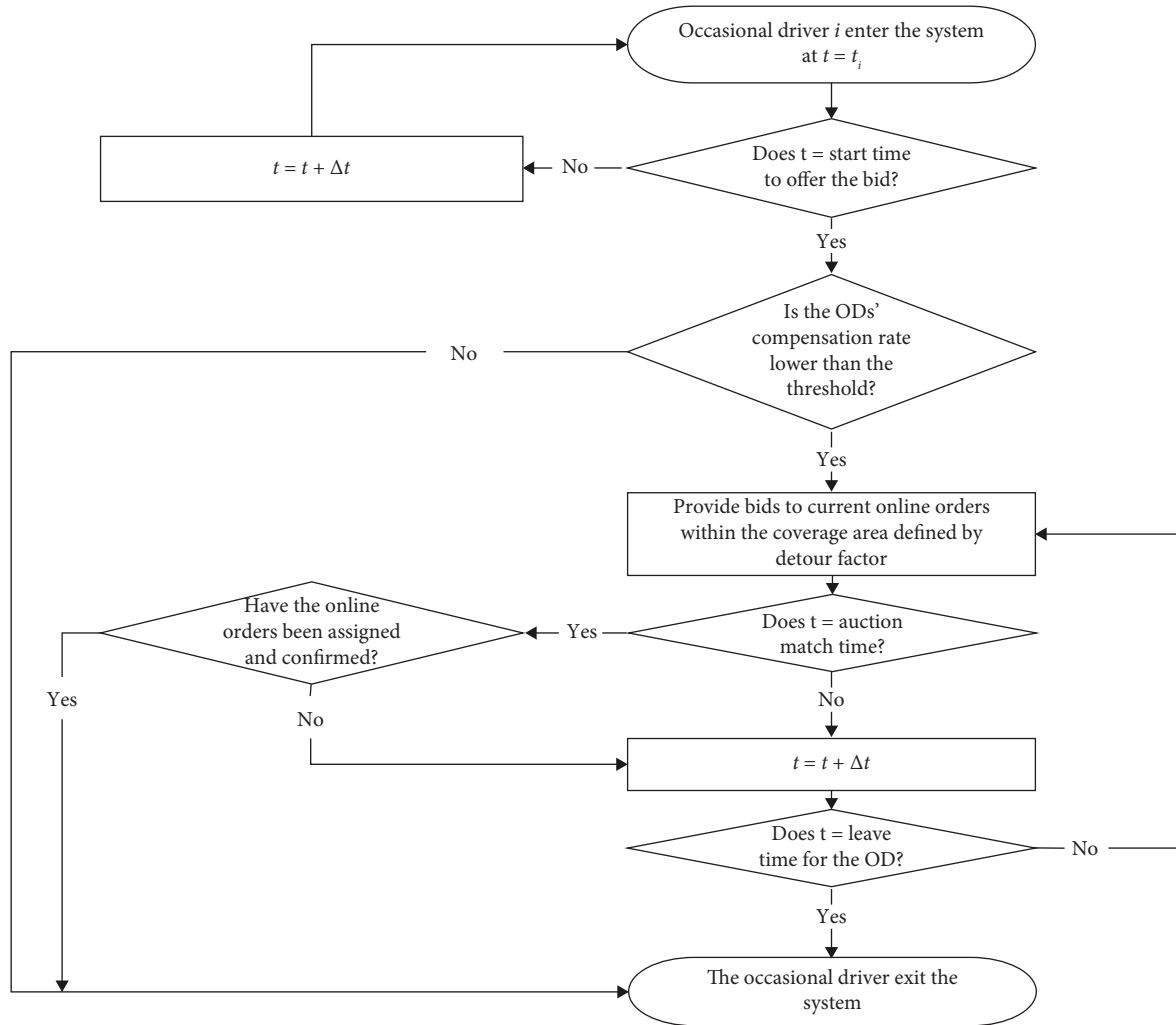


FIGURE 7: OD dynamics.

```

(1) An occasional driver  $j$ ,  $OD_j$ , enter at  $t = t_j$ 
(2) while the clock time  $t$  is not greater than the leave time for  $OD_j$  do
(3)   if  $t =$  start time to offer the bid then
(4)     if  $OD_j$ 's compensation rate is not greater than the maximum rate
(5)       Provide bids to current online orders
(6)     else
(7)        $OD_j$  exit the system
(8)     end if
(9)   if  $t =$  scheduled auction match time then
(10)    Run auction-based matching procedure
(11)   end if
(12)    $t = t + \Delta_t$ 
(13) end while
  
```

ALGORITHM 4: Occasional driver dynamics.

(5) *Average order fulfillment time* (T_f): The average time between the announcement of the online order and actual received time.

(6) *Order matching rate* (R_{order}): The number of matched online orders divided by the total number of online orders announced (reported as a percentage).

- (7) *OD matching rate* (R_{ODs}): The number of matched ODs divided by the total number of ODs (reported as a percentage).
- (8) *OD total detour travel distance* ($Dist_d$): The total detour distance traveled by the matched OD.
- (i) The following parameters were set to the default values in our base case:
 - (ii) Total simulation time span: $T = 480$ minutes
 - (iii) Time horizon between two consecutive matches: 40 minutes
 - (iv) Time for ODs to shop in the store: 20 minutes
 - (v) Time for ODs to stay at the store, excluding shopping time: 40 minutes
 - (vi) Maximum detour ratio for each OD: 0.6
 - (vii) Fixed compensation for each OD: \$1.00
 - (viii) Variable compensation per extra travel distance for each OD: \$0.60
 - (ix) Maximum number of orders an OD could deliver: 2
 - (x) Time between two consecutive fixed departure times for the regular fleet: 60 minutes
 - (xi) Departure flexibility, which was the time between the earliest and latest departure times: 120 minutes
 - (xii) Matching commitment strategy: latest commitment strategy for matches.

5.2. Base Case Results. Using the base case settings described above, we provide the simulation results in Table 2. Under the base case, we tested the effects of different time horizon settings for two consecutive matches: 5 min, 10 min, 20 min, 30 min, 40 min, 50 min, 60 min, and 70 min. Table 2 shows that the cost of ODs decreased when the time horizon increased; conversely, the cost of regular vehicles followed an opposite trend. The lowest total cost was at around 237 to 239 when the time horizon was between 20 and 50 minutes. The CPU run time is the average run time of the simulation. As the rolling horizon increases, the CPU run-time gradually increases.

The OD match rate just slightly decreased by 4.8% (44 to 42) when the rolling time horizon was increased from 5 to 40 minutes. The success matching rate of the online orders decreased by 7.1% (70 to 65). At the same time, the compensation cost for the ODs decreased by up to 15.7% (103.75 to 87.50). Therefore, we inferred that with an appropriate rolling time horizon, there would be an increased chance to find lower-cost matches. When the rolling time horizon was increased from 40 to 70 minutes, the matching rates of online orders and ODs both decreased significantly. This showed that the benefit of increasing the time horizon would eventually be offset when the time horizon was greater than the time for ODs to stay at the store (the default setting was 40 minutes). When the time between matches was greater than the time for the ODs to stay at the store (the default setting was 40 minutes), matching opportunities became more and more often missed. This was because given the maximum wait time at the store, some ODs might not have

had the chance to be matched if the matching time horizon was greater than the maximum waiting time. For example, consider an OD announced at 10:00, with the previous matching decision made at 9:55. The OD-order match could be missed if the matching time horizon was 50 minutes because the next matching decision would be at 10:45 and the OD would leave at 10:40.

5.3. Results Comparison for the Grabbing and Assignment Models. Here, we present the results for both the grabbing (i.e., without competition) and assignment (i.e., with competition) models. The grabbing model reflected ODs' immediately grabbing orders when satisfied with the extra travel distance and compensation. This model reflected cases in which orders were selected by the first OD but not necessarily the best [25]. The assignment model was proposed in Section 4. In order to highlight the advantages of including the auction mechanism in our pairing process, we compared the results of cases "with competition" and "without competition," using a test problem generalized from Solomon's VRPTW benchmark problem R101. Table 3 shows that the auction significantly outperformed the no-auction in terms of average cost per online order (\$1.59 vs. \$1.24). The average compensation for ODs improved from \$2.61 (without competition) to \$1.96 (with competition), a 25.1% savings. Not surprisingly, the bid approach generated better individual cost savings; however, the matching rates for ODs (38%) and online orders (50%) decreased. Other advantages of not having competition matching include lower lateness rates and average order fulfillment times. This was because in the absence of competition cases, once matches were found, the OD could leave the store immediately.

The benefits of including a rolling horizon are two-fold. First, in the purely dynamic Case (i.e., when online orders and occasional drivers are assumed in advance to be unknown), including the rolling horizon approach allowed for the matching quality (defined as the average cost per online order per OD) to be improved by up to 11.5% when the rolling horizon was increased to a certain level (see Table 2). In addition, testing by benchmark problem, inclusion would improve the total cost by 5%, as compared to when no rolling horizon method is included (see Table 3).

It is worth noting that agent-based simulation is an effort to incorporate individual decision-making behavior into each type of agent. Therefore, it cannot be guaranteed that the obtained solution is either optimal or near-optimal [19]. In other words, agent-based simulations capture emergent phenomena. Emergent phenomena result from the interactions of individual entities with one another. As a consequence, in agent-based models, agents are not usually able to "instantaneously" find the global optimum solution space; rather, they discover the solution space in a stepwise process of searching for better solutions [26].

5.4. Sensitivity to Problem Characteristics. In this section, we discuss the implications of sensitivity to problem characteristics, including the maximum OD detour ratio, OD

TABLE 2: Base case under different rolling horizons.

| Rolling horizon | Cost | | | Lateness ratio | Online order | | Occasional driver | | Average cost per order per OD | CPU run time (mm:ss) |
|-----------------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|-------------------------------|----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance | | |
| 5 | 142.87 | 103.75 | 246.62 | 0.24 | 68.44 | 70 | 44 | 73.73 | 1.469 | 8:14 |
| 10 | 146.57 | 97.55 | 244.12 | 0.27 | 75.42 | 67 | 43 | 66.29 | 1.442 | 9:36 |
| 20 | 146.17 | 93.79 | 239.96 | 0.28 | 77.04 | 66 | 42 | 60.55 | 1.400 | 11:26 |
| 30 | 148.46 | 89.06 | 237.52 | 0.29 | 79.23 | 65 | 41 | 55.30 | 1.347 | 12:43 |
| 40* | 151.74 | 87.50 | 239.24 | 0.30 | 81.17 | 65 | 42 | 53.14 | 1.329 | 14:01 |
| 50 | 170.09 | 68.56 | 238.65 | 0.39 | 95.09 | 52 | 33 | 39.21 | 1.300 | 14:28 |
| 60 | 176.34 | 65.69 | 242.04 | 0.45 | 103.40 | 50 | 30 | 35.71 | 1.300 | 17:03 |
| 70 | 200.70 | 45.05 | 245.75 | 0.55 | 117.56 | 34 | 21 | 27.00 | 1.301 | 17:30 |

*Base case.

TABLE 3: Comparison of the grabbing and proposed assignment models.

| Model | Cost | | | Lateness ratio | Online order | | Occasional driver | |
|-------------------------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance |
| <i>Grabbing model</i> | 52.56 | 130.55 | 183.11 | 0.08 | 46.64 | 82 | 50 | 100.86 |
| <i>Assignment model</i> | 99.70 | 74.31 | 174.01 | 0.27 | 78.24 | 60 | 38 | 54.14 |

carrying capacity, ratio of online orders to ODs, OD compensation threshold, distributions of online orders and ODs, and regular vehicle service policy.

5.4.1. Impact of Maximum OD Detour Ratio. Table 4 shows the impacts of different maximum detour ratios for ODs. When the ratio increased from 1.2 to 1.6 (the default setting), the OD usage ratio increased by 17% (from 36 to 42). There was also a 21% increase in online orders delivered by ODs (from 53 to 64), meaning that the number of online orders delivered by each OD increased when the maximum detour ratio increased. It is also of note that the number of matched online orders and utility rate of ODs remained almost unchanged by an increase in the detour ratio from 1.6 to 2.0. This was mainly due to the number of online orders carried being dominated by the number of items allowed for each OD to carry.

5.4.2. Impact of OD Carrying Capacity. As expected, the carrying capacity positively affected the number of matched online orders (see Table 5). The percentage of matched online orders increased as the maximum capacity increased, while the matched rate of ODs slightly decreased. This implies that the average number of orders delivered per OD increased as capacity increased.

5.4.3. Impact of the Ratio of ODs to Online Orders. To better study the interaction between the numbers of ODs and online orders, we generated three sets of ratios, according to the following rules:

Ratio 1: ODs/online orders = 50/100.

Ratio 2: ODs/online orders = 100/100.

Ratio 3: ODs/online orders = 150/100.

Table 6 shows the outcome of changing the ratio from 0.5 to 1.5. Clearly, as the number of ODs increased, more opportunities to find a match emerged. Therefore, the percentage of matched online orders increased from 44 to 74. The second important finding is that increasing the ratio of ODs to online orders had a negative impact on the OD match rate (from 53% to 33%). Though the OD usage rate decreased, the total matched ODs actually increased from 26.5 to 49.5. This suggests that the delivery task was performed on more ODs. At the same time, we observed that as a result of this change, the lateness ratio decreased by more than 50%.

5.4.4. Impact of OD Compensation Threshold. In this section, we compare three different compensation thresholds. Each OD was assigned a compensation coefficient by drawing a random number from a normal distribution $N(0.6, 0.25)$ That OD was considered for crowdshipping only if the compensation coefficient was less than or equal to the compensation threshold. Note that a higher threshold meant that more ODs were allowed to enter the system, and thus, the matched rate of both online orders and ODs could be increased. Table 7 reports the results obtained from these cases. A comparison of those results revealed that allowing more ODs to participate increased the matched percentage for both online orders and ODs, but the marginal effect was diminishing.

TABLE 4: Maximum OD detour ratio.

| Detour ratio | Cost | | | Lateness ratio | Online order | | Occasional driver | |
|--------------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance |
| 1.2 | 179.97 | 62.10 | 242.07 | 0.40 | 97.09 | 53 | 36 | 18.49 |
| 1.4 | 159.89 | 79.02 | 238.91 | 0.33 | 85.52 | 61 | 40 | 38.60 |
| 1.6* | 152.78 | 87.21 | 239.99 | 0.30 | 81.15 | 64 | 42 | 52.59 |
| 1.8 | 145.77 | 91.03 | 236.80 | 0.29 | 78.91 | 66 | 42 | 62.89 |
| 2 | 146.41 | 91.37 | 237.78 | 0.29 | 78.71 | 66 | 42 | 67.79 |

*Default setting.

TABLE 5: Different OD carrying capacities.

| Carrying capacity | Cost | | | Lateness ratio | Online order | | Occasional driver | |
|-------------------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance |
| 1 | 189.80 | 56.91 | 246.71 | 0.49 | 106.57 | 43 | 43 | 28.73 |
| 2* | 150.96 | 87.79 | 238.75 | 0.29 | 80.23 | 65 | 42 | 52.38 |
| 3 | 127.45 | 102.33 | 229.78 | 0.21 | 69.47 | 74 | 41 | 66.26 |

*Default setting.

TABLE 6: Impacts of the ratio of ODs to online orders.

| ODs (online orders) | Cost | | | Lateness ratio | Online order | | Occasional driver | |
|---------------------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance |
| 50/100 | 191.90 | 63.60 | 255.50 | 0.48 | 107.70 | 44 | 53 | 40.45 |
| 100/100* | 151.38 | 87.64 | 239.02 | 0.29 | 81.02 | 64 | 42 | 52.44 |
| 150/100 | 123.70 | 98.66 | 222.36 | 0.20 | 68.00 | 74 | 33 | 56.01 |

*Default setting.

TABLE 7: Impact of the OD compensation threshold.

| Threshold | Cost | | | Lateness ratio | Online order | | Occasional driver | |
|-----------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance |
| 0.6 | 177.10 | 71.19 | 248.29 | 0.41 | 96.84 | 52 | 32 | 50.35 |
| 0.85* | 150.03 | 87.54 | 237.57 | 0.30 | 81.16 | 64 | 41 | 52.76 |
| 1 | 149.89 | 89.09 | 238.98 | 0.28 | 78.07 | 66 | 44 | 53.20 |

*Default setting.

TABLE 8: Comparison of regular vehicle dispatch policies.

| Dispatch policy | Cost | | | Lateness ratio | Online order | | Occasional driver | |
|---------------------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance |
| Fixed interval* | 154.31 | 87.04 | 241.35 | 0.30 | 81.26 | 64 | 42 | 52.87 |
| Fixed urgent orders | 107.94 | 88.62 | 196.56 | 0.31 | 89.98 | 66 | 42 | 53.96 |

*Default setting.

TABLE 9: Comparison of different geographic distributions of online orders.

| Geographic distribution | Cost | | | Lateness ratio | Online order | | Occasional driver | |
|-------------------------|-----------------|-------------------|--------|----------------|--------------------------|--------------------|-------------------|-----------------------|
| | Regular vehicle | Occasional driver | Total | | Average fulfillment time | Orders matched (%) | ODs matched (%) | Total detour distance |
| C101 | 76.48 | 44.26 | 120.73 | 0.43 | 94.59 | 38 | 22 | 16.98 |
| R101 | 86.37 | 81.93 | 168.30 | 0.24 | 75.24 | 62 | 38 | 38.41 |
| RC101 | 117.33 | 68.41 | 185.74 | 0.39 | 88.53 | 57 | 34 | 29.74 |

5.5. Impact of Regular Vehicle Dispatch Policy. We evaluated two strategies for managing the time at which regular vehicles embark to deliver online orders. The first strategy, which initiated departure once a fixed time interval of 60 minutes had elapsed. The second strategy initiated departure once the fixed number of urgent online orders reached 10. The results are shown in Table 8.

Because we only changed the company vehicle policy, the measurements referring to the ODs matched, online orders matched, and the total OD detour distance did not change. The cost of the company vehicles, however, decreased from \$154.31 to \$107.94 when switching from a fixed interval to a fixed number of urgent orders. The above savings were mainly the result of the policy of a fixed number of urgent orders; the average number of regular vehicles was three. However, for the fixed time interval policy, the average number of regular vehicles was six. The average number of urgent online orders per hour was between four and five. Under the fixed urgent orders policy, the company vehicle needed to wait (longer than one hour) until the number reached 10. It is of note that the average order fulfillment time was longer for the fixed urgent orders strategy. In sum, there was a tradeoff between the number of regular vehicles used and the average order fulfillment time.

5.6. Impact of Online Orders' Geographic Distribution. Here, we generated the instances beginning with the three Solomon VRPTW classes representing different geographic distribution patterns of destinations. Online order destinations were either:

- (1) Clustered (C101),
- (2) Randomly distributed (R101), or
- (3) Partially clustered and partially randomly distributed (RC101).

In Solomon's benchmark problem, geographical data are randomly generated in problem set R1, clustered in problem set C1, and a mix of random and clustered structures in problem set RC1. Hence, R101 represents the first instance in problem set R1, C101 is the first instance in problem set C1, and RC101 is the first instance in problem set RC1.

We assumed that all ODs began from a central location (the store) and the destinations were uniformly randomly distributed throughout the area. The results in Table 9 show that the highest number of matched ODs was for R101 (38%) and the lowest was for C101 (22%). This was because the OD destinations were uniformly randomly generated. Therefore, if the online orders are clustered (C101), the chance that an OD can not find any matchable online orders increases. Furthermore, it was easier to increase competition among ODs if a cluster of online orders happened to be bid upon by several ODs. The above two reasons made it more likely that some ODs would not be matched with online orders in C101.

Conversely, if the online orders' destinations were randomly distributed, such as in instance R101, competition between ODs was less likely. Thus, R101 had the highest number of online order matches (62%). It is also of note that

the number of matched orders for R101 was 63% more than the number for C101, implying that a better geographic distribution alignment between the online orders and ODs would significantly increase the match ratio for the service.

6. Conclusions

In this research, an auction-based multiagent model of the dynamic VRPOD was simulated to explore the potential value of auctions in crowdsourced delivery services. Overall, we concluded that the levels of match quality depended on the values for the rolling time horizon, the time period between two consecutive matches. If the rolling time horizon was similar to the time ODs would be at the store, then, the average compensation cost for each OD was optimized. Conversely, if the rolling time horizon was much smaller than the OD stay time, the average compensation cost per OD increased because the short rolling horizon increased the likelihood of missing the chance to accumulate online orders between auction runs. However, if the rolling time horizon was longer than the OD stay time, the compensation cost per OD increased due to the reduced opportunity to be matched.

Our proposed auction-based model was also compared with the nonauction case. The advantage of the auction was that the compensation paid to each OD could be reduced by up to 23.8%, but the nonauction scenario demonstrated less lateness. We further showed that the match rate for online orders was positively related to the detour ratio, OD carrying capacity, ratio of ODs to online orders, and OD compensation threshold. Finally, waiting to accumulate a fixed number of urgent online orders, rather than waiting for a fixed time interval, decreased the cost per regular vehicle.

In the proposed simulation model, we assumed that ODs would insist on their initial compensation rate throughout the time period they remained at the store. In fact, if ODs desiring delivery tasks before leaving the store do not receive any matches, they may change their preferences as the time remaining approaches the threshold value. The current model could be modified to include changing preferences, an issue that will be considered in future work.

Auction mechanisms are a rich area of future study. How auction mechanisms affect the performance of the VRPOD and its application to realistic problems are topics worthy of further investigation.

Data Availability

The data supporting this numerical analysis are modified from Solomon's VRPTW benchmark problems. The data are available at <https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/>.

Disclosure

This paper is revised from a portion of the thesis entitled *An Auction-Based Multiagent Simulation for the Dynamic Vehicle Routing Problem with Occasional Drivers* by Che-Cheng Hsu [27].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Chung-Wei Shen conceptualized the study, developed the methodology, visualized the study, revised the manuscript, and replied the comments. Chung-Wei Shen, Che-Cheng Hsu, and Kuan-Hua Tseng performed the software and validated the results. Chung-Wei Shen and Che-Cheng Hsu did formal analysis and wrote the original draft. All the authors reviewed the results and approved the final version of the manuscript.

Acknowledgments

This research was sponsored by the Ministry of Science and Technology, Taiwan, ROC under Grant numbers MOST 109-2410-H-006-078-, MOST 110-2410-H-006-060-, and MOST 111-2410-H-006-097-.

References

- [1] Y. Vakulenko, P. Shams, D. Hellström, and K. Hjort, "Service innovation in e-commerce last mile delivery: mapping the e-customer journey," *Journal of Business Research*, vol. 101, pp. 461–468, 2019.
- [2] V. Carbone, A. Rouquet, and C. Roussat, "The rise of crowd logistics: a new way to co-create logistics value," *Journal of Business Logistics*, vol. 38, no. 4, pp. 238–252, 2017.
- [3] 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.
- [4] M. Allahviranloo and A. Baghestani, "A dynamic crowdshipping model and daily travel behavior," *Transportation Research Part E: Logistics and Transportation Review*, vol. 128, pp. 175–190, 2019.
- [5] E. Bonabeau, "Agent-based modeling: methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. 3, pp. 7280–7287, 2002.
- [6] S. Voigt and H. Kuhn, "Crowdsourced logistics: the pickup and delivery problem with transshipments and occasional drivers," *Networks*, vol. 79, no. 3, 2021.
- [7] M. Nourinejad and M. J. Roorda, "Agent based model for dynamic ridesharing," *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 117–132, 2016.
- [8] C. Archetti, M. Savelsbergh, and M. G. Speranza, "The vehicle routing problem with occasional drivers," *European Journal of Operational Research*, vol. 254, no. 2, pp. 472–480, 2016.
- [9] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, and D. Laganà, "The vehicle routing problem with occasional drivers and time windows," in *Proceedings of the International Conference on Optimization and Decision Science*, Sorrento, Italy, November 2017.
- [10] C. Triki, "Using combinatorial auctions for the procurement of occasional drivers in the freight transportation: a case-study," *Journal of Cleaner Production*, vol. 304, Article ID 127057, 2021.
- [11] C. Archetti, F. Guerriero, and G. Macrina, "The online vehicle routing problem with occasional drivers," *Computers & Operations Research*, vol. 127, Article ID 105144, 2021.
- [12] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: a review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.
- [13] T. V. Le, S. V. Ukkusuri, J. Xue, and T. Van Woensel, "Designing pricing and compensation schemes by integrating matching and routing models for crowd-shipping systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 149, Article ID 102209, 2021.
- [14] T. Hou and W. Zhang, "Optimal two-stage elimination contests for crowdsourcing," *Transportation Research Part E: Logistics and Transportation Review*, vol. 145, Article ID 102156, 2021.
- [15] 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.
- [16] L. Dahle, H. Andersson, M. Christiansen, and M. G. Speranza, "The pickup and delivery problem with time windows and occasional drivers," *Computers & Operations Research*, vol. 109, pp. 122–133, 2019.
- [17] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 217–250, 2020.
- [18] K. Gdowska, A. Viana, and J. P. Pedroso, "Stochastic last-mile delivery with crowdshipping," *Transportation Research Procedia*, vol. 30, pp. 90–100, 2018.
- [19] S. Gujar and B. Faltings, "Auction based mechanisms for dynamic task assignments in expert crowdsourcing," in *Proceedings of the Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, pp. 50–65, Springer, Istanbul, Turkey, May 2015.
- [20] J. Dötterl, R. Bruns, J. Dunkel, and S. Ossowski, "On-time delivery in crowdshipping systems: an agent-based approach using streaming data," in *ECAI 2020*, pp. 51–58, IOS Press, Amsterdam, Netherlands, 2020.
- [21] C. M. Glen, M. L. Kemp, and E. O. Voit, "Agent-based modeling of morphogenetic systems: advantages and challenges," *PLoS Computational Biology*, vol. 15, no. 3, Article ID e1006577, 2019.
- [22] P. Chen and S. M. Chankov, "Crowdsourced delivery for last-mile distribution: an agent-based modelling and simulation approach," in *Proceedings of the 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, December 2017.
- [23] V. E. Castillo, J. E. Bell, D. A. Mollenkopf, and T. P. Stank, "Hybrid last mile delivery fleets with crowdsourcing: a systems view of managing the cost-service trade-off," *Journal of Business Logistics*, vol. 43, no. 1, pp. 36–61, 2022.
- [24] N. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang, "Dynamic ride-sharing: a simulation study in metro Atlanta," *Procedia-Social and Behavioral Sciences*, vol. 17, pp. 532–550, 2011.
- [25] L. Zhen, Y. Wu, S. Wang, and W. Yi, "Crowdsourcing mode evaluation for parcel delivery service platforms," *International Journal of Production Economics*, vol. 235, Article ID 108067, 2021.
- [26] F. Wall, "Agent-based modeling in managerial science: an illustrative survey and study," *Review of Managerial Science*, vol. 10, no. 1, pp. 135–193, 2016.
- [27] C. C. Hsu, "An auction-based multi-agent simulation for the dynamic vehicle routing problem with occasional drivers," Master Thesis, National Chung Kung University, Tainan City, Taiwan, 2021.