WORKING PAPER SERIES

Dynamic Assignment of Delivery Order Bundles to In-Store Customers

Simona Mancini/Marlin W. Ulmer/Margaretha Gansterer

Working Paper No. 12/2023

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

FACULTY OF ECONOMICS AND MANAGEMENT

Impressum (§ 5 TMG)

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

Verantwortlich für diese Ausgabe:
S. Mancini, M. W. Ulmer, M. Gansterer
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

Bezug über den Herausgeber
ISSN 1615-4274

Dynamic Assignment of Delivery Order Bundles to In-Store Customers

Simona Mancini
University of Palermo, Department of Engineering, 90133 Palermo, Italy, simona.mancini@unipa.it
University of Klagenfurt, Department of Operations, 9020 Klagenfurt, Austria. simona.mancini@aau.at

Marlin W Ulmer*
Otto-von-Guericke Universität Magdeburg, 39016 Magdeburg, Germany, marlin.ulmer@ovgu.de, *Corresponding Author

Margaretha Gansterer
University of Klagenfurt, Department of Operations, 9020 Klagenfurt, Austria. margaretha.gansterer@aau.at

Many larger grocery stores offer home delivery services. However, the delivery cost is usually high and such services are rarely profitable. One way of reducing cost is by outsourcing some orders to in-store customers for a compensation. While initially single orders were dynamically assigned to customers, companies started exploring the assignment of order bundles instead to reduce per-order compensation and exploit consolidation potential. We investigate the value of dynamic assignment of bundles in this work. To this end, we consider a setting where all orders are known and, over time, unknown in-store customers enter the system for a short time and offer transportation of bundles of orders for compensation. The store decides dynamically which bundle to assign to which in-store customer (if any). At the end of the time horizon, the remaining orders are delivered by a dedicated fleet of store employees. The goal of the store is to minimize the compensation prices together with the delivery cost. We propose a threshold based policy tuned by a stochastic lookahead procedure. Popularity and compensation price thresholds are determined a priori by solving a set of perfect information scenarios. In every state, bundles are only assigned if they are popular enough and the compensation is comparably low. The thresholds are adapted over time to account for the decrease in assignment opportunities. We show the effectiveness of our policy in a comprehensive computational study and highlight the value of bundle assignments compared to assigning individual orders. We further show that our strategy not only reduces the compensation paid to in-store customers but also the final routing cost.

Key words: Crowdsourcing, Order bundling, Sequential decision making, Approximate dynamic programming

1. Introduction

Many retailers such as Walmart or Home Depot offer same-day home delivery to customers. Often, customers can order until a cutoff time of day, e.g., until noon and then delivery takes place afterwards, e.g., in the afternoon. This delivery is often done by company employees (Boysen et al. 2022). As customers can be widely spread around the service area with only a few customers in some areas, the travel is often long and the cost for delivery is high. Thus, it might be valuable to outsource some parts of delivery, for example, to in-store customers that visit the corresponding areas anyway. Following Archetti et al. (2016), we call these in-store customers *occasional drivers* (ODRs). Such ODRs can log into the store's app and indicate their willingness to participate. They also enter the conditions under which they are willing to participate, i.e., their destination, the maximum detour they are willing to take, and their minimum compensation per bundle. Based on the information, the platform can then decide to assign a bundle of orders to an ODR and pay the corresponding compensation. The orders in the bundle are then served by the ODR and do not need to be served in the employees' tours.

Even when knowing all the ODRs, the underlying optimization problem is quite rich with bundle generation, ODR assignment, and final employee routing. However, in reality, ODRs appear spontaneously over time. Thus, the retailer has to decide repeatedly what bundles to assign to newly arrived ODRs without knowing about future ODRs and their preferences. The consideration of bundles further increases the complexity of decision making and, as the recent survey by Savelsbergh and Ulmer (2022) summarizes, has not been studied in a stochastic dynamic setting in the literature yet. Savelsbergh and Ulmer (2022) further state that "Creating the right bundles at the right time and offering them to the right couriers [ODRs] is extremely challenging." In contrast to the assignment of individual orders, bundle assignments are interdependent. Assigning one bundle to an ODR may prohibit assigning another bundle to a second ODR since both bundles may share orders. Further, a comparably cheap assignment now may break up other attractive bundles and therefore reduces the likelihood of the remaining orders to be assigned to future ODRs. In all these considerations, the final routing cost of the own fleet must be considered as, for example, a comparably expensive assignment may reduce the routing cost substantially. All this must be done in real time before ODRs leave the system again. This prohibits extensive calculations during the process. To tackle these challenges, we propose an a priori stochastic lookahead method. Before the start of the process, a larger subset of promising bundles is generated. Our methods then solves the perfect information problem for a set of ODR-scenarios to identify structure in bundle popularity and compensation (including approximations of the final routing of the company's employees). Based on the observed structure, time-dependent thresholds for popularity and compensation are derived. These thresholds then guide decision making during the process.

We compare our method to eight benchmark policies with respect to business concept, problem, and methodology. We also provide a perfect information bound. Our policy outperforms the benchmarks and reaches comparably small gaps of 15%-20% to the bound. We derive the following managerial insights:

- Dynamically assigning bundles of orders to ODRs can reduce delivery cost significantly compared to conventional delivery by employees. In cases where the delivery cost of of the company fleet is small, adding ODRs is not beneficial. Here, companies should carefully balance potential cost savings with operational simplicity.

- Only offering individual orders to ODRs instead of bundles may also render delivery via ODRs unsustainable. Offering carefully designed bundles instead of individual orders reduces both compensation and routing cost.

- Even assigning bundles myopically can save cost compared to conventional delivery, however, it may not only cause unnecessary high compensation cost but also breaks up bundles attractive for future ODRs. The anticipatory assignment of bundles is crucial for effective decision making and yields substantially larger savings.

- For the same reason, trying to clear all orders as fast as possible may not only increase compensation paid, but also the final routing cost, as many dispersed orders must be served by the company fleet. Here, anticipating final routing cost in the assignment of bundles is very valuable.
- An effective strategy should consider the decreasing number of opportunities over the time horizon, being more selective in the beginning of the decision horizon and becoming decreasingly less selective towards its end.

We contribute to the literature with respect to both problem and methodology. Our contributions are as follows. We are the first investigating the business concept where bundles of orders are assigned dynamically to ODRs. Our work presents an effective anticipatory policy that allows instant decision making. We perform a comprehensive analysis to derive valuable managerial insights. With respect to methodology, we are among the first to use an a priori stochastic lookahead method for tuning a threshold-based policy. Furthermore, this paper is among the first integrating a cost function approximation of the routing cost in the stochastic lookahead method. Both approaches work quite effectively, and, as general concepts, might prove valuable beyond the specific problem in this work.

The paper is outlined as follows. We discuss related literature in Section 2. The problem is defined and modeled in Section 3. Our solution method is presented in Section 4 followed by the computational study in Section 5. Section 6 concludes the paper with a summary and an outlook.

2. Literature

In the following, we present the related literature on dynamic crowdsourced delivery operations.

Crowdsourced deliveries are either conducted by freelance drivers or by ODRs, where the former conduct several tasks over a longer time horizon typically being active on different platforms, while the latter extend their trip only slightly (Kaspi et al. 2022). Both are non-regular drivers used by companies to conduct cost efficient last mile deliveries (Wang et al. 2023). Extensive literature reviews on crowdsourced deliveries including matching mechanisms is provided by Alnaggar et al. (2021) and Savelsbergh and Ulmer (2022). In the following, we focus on ODRs performing deliveries.

Typically, ODRs are in-store customers with free capacity in their cars that they use to ensure deliveries once leaving the store. To the best of our knowledge, Archetti et al. (2016) were the first to present and computationally analyze the underlying decision problem, where companies include ODRs into their route plans. The authors, however, assume that all ODRs and their preferences are known and that each ODR can cover only a single delivery. Different model formulations for this deterministic problem and the impact of different compensation schemes are analyzed in Dahle et al. (2019).

In the following years, several extensions have been proposed in the literature. For instance, Macrina et al. (2017) assign more than one delivery to each ODR and also time windows are imposed. However, no dynamic or stochastic components are included in the problem setting. Solutions are generated by a hybrid algorithm combining a genetic algorithm with a local search heuristic. Di Puglia Pugliese et al. (2022) use greedy randomized adaptive search procedures and Ahamed et al. (2021) use deep reinforcement learning for crowdsourced urban delivery. Both studies assume that ODRs can serve bundles of customers, but stochasticity is not considered. Macrina et al. (2020) allow for transshipment points in a crowdsourced delivery system. Again, the problem is assumed to be deterministic and static and only single orders can be assigned to ODRs.

The number and location of transshipment points in this context is analyzed by Voigt and Kuhn (2022). The proposed model is based on static deterministic information. The reduction of fuel consumption and resulting emissions, particularly related drivers' behavior, is in the focus of Al Hla et al. (2019). ODRs are assumed to serve more than one order on their routes, but again no dynamics or stochastics are considered.

The assignment of ODRs to pickups and deliveries is studied by Yu et al. (2023). The authors assume that drivers differ in their flexibility to accept detours, but they are assumed to serve only one single order. The authors formulate the problem as a static and deterministic mixed integer linear programming model and develop a simulated annealing heuristic to generate solutions.

The dynamic version of the problem is, for instance, tackled by Arslan et al. (2019). They assume a service platform which assigns delivery tasks to so called ad hoc drivers and a dedicated fleet. The authors propose a rolling horizon framework and an exact solution approach for the matching problem. Considerable cost reductions of about 37% compared to traditional delivery systems are shown. A dynamic and stochastic variant in which online orders as well as ODRs arrive throughout the day is addressed by Dayarian and Savelsbergh (2020). The authors propose a dispatching approach that takes probabilistic information into account as well as a myopic one. A deep reinforcement learning approach for the dynamic stochastic problem is developed in Silva et al. (2023). In none of these studies, however, ODRs have the possibility to indicate preferences (i.e., bids) on offered bundles of orders.

The multi-depot version of the problem is formulated by Tao et al. (2023), who use an event-based rolling horizon solution approach with an iterative re-optimization procedure to generate solutions. A bi-level methodology for matching and routing of ODRs and private fleets is presented in Gdowska et al. (2018). On the first level, the probability of ODRs' rejection decisions is explicitly taken into account. Cheng et al. (2017) consider ODRs' trajectory uncertainties by allowing multiple routine tours to be probabilistically associated with each worker. Managerial insights based on real-world data from Singapore are provided.

Another component of uncertainty is added by Di Puglia Pugliese et al. (2023). The authors consider uncertain travel times and a penalty for each missed delivery. A chance-constrained stochastic model imposing a probability on the maximum number of missing deliveries is formulated and two exact solution approaches are proposed.

Torres et al. (2022) assume that drivers choose routes based on compensations and on other routes being offered. A two-stage stochastic model along with an exact as well as a heuristic solution approach are presented. To mitigate acceptance uncertainty in peer-to-peer transportation, Ausseil et al. (2022) propose to let platforms generate menus of orders to choose from, which they do by a multiple scenario approach for the two-stage decision problem. A similar problem is considered by Çınar et al. (2023), where instead of menus, individual compensation is optimized to balance acceptance probability with cost. Acceptance might also depend on flexibility and/or willingness of drivers.

All aforementioned work considers individual orders instead of bundles. Mancini and Gansterer (2022) propose a bundle generation approach that aims at generating attractive packages of orders for drivers with different flexibility and willingness levels. They propose a combinatorial auction-based framework, where ODRs bid on bundles depending on their characteristics. Also Triki (2021) develop a combinatorial auction for the problem of integrating ODRs in traditional delivery systems. A real world study covering door-to-door deliveries in Oman show the validity of the proposed method as well as considerable cost savings of up to 30%. Shen et al. (2022) consider a similar problem where dynamically arriving ODRs bid on individual orders and can be assigned one or more orders for delivery. They propose an auction-based multi-agent simulation to analyze the value of delivery via ODRs. The algorithm aims on clearing as many orders as possible as cheap as possible in every state, thus, ignoring future developments. In our computational study, we design the benchmark-policy "Clearance" accordingly. Alnaggar et al. (2023) tackle the problem of dynamically matching ODRs with incoming orders if compensation guarantees have to be considered. The authors propose a value function approximation for anticipatory decision making. Finally, Dayarian and Pazour (2022) utilizes in-store customers not for delivery but for dynamically picking orders in the store. The authors propose a stochastic lookahead to calculate the value of assigning an order to an in-store customer.

In all the studies mentioned above, we observe that either (i) only single orders are assigned and/or carriers do not have the option to reveal preferences (bids) for offered orders, or (ii) the problem is assumed to be static or deterministic. To the best of our knowledge, we are the first to address the dynamic stochastic problem, where bundles of orders are assigned to ODRs based on their submitted bids. In Table 1, we categorize the related literature, indicating problem characteristics (dynamic, stochastic) and whether a bidding mechanism and/or anticipatory planning are included. The chronological summary in Table 1 shows that initial work assumed deterministic settings to focus on the theoretical value of ODRs for delivery operations. Later, stochastic and dynamic aspects were added with an increasing focus on anticipation of ODR appearances and

Reference	stochastic	dynamic	bundling	anticipation
Archetti et al. (2016)				
Macrina et al. (2017)			X	
Cheng et al. (2017)	X			
Gdowska et al. (2018)	X			X
Al Hla et al. (2019)				
Arslan et al. (2019)	X	X		
Dahle et al. (2019)				
Dayarian and Savelsbergh (2020)	X	X		X
Macrina et al. (2020)				
Ahamed et al. (2021)	X	X		X
Triki (2021)			X	
Ausseil et al. (2022)	X	X		X
Dayarian and Pazour (2022)	X	X		X
Di Puglia Pugliese et al. (2022)				
Mancini and Gansterer (2022)			X	
Shen et al. (2022)	X	X		
Torres et al. (2022)	X			X
Voigt and Kuhn (2022)				
Alnaggar et al. (2023)	X	X		X
Çınar et al. (2023)	X	X		X
Di Puglia Pugliese et al. (2023)	X			
Silva et al. (2023)	X	X		X
Tao et al. (2023)	X	X		X
Yu et al. (2023)				
This work	X	X	X	X

Table 1 **Overview of problem characteristics addressed in the literature.**

their behavior. As the table also reveals, studies on the assignments of bundles to ODRs is very limited and our work is the first that considers the creation and dynamic assignment of bundles to stochastically appearing ODRs, where we propose an anticipatory solution method for decision making.

3. Problem setting

In this section, we present the problem. We give a problem description, model the problem a sequential decision process and provide a small example for illustration.

3.1. Problem description

We assume a home delivery problem where during a capture phase customers order delivery from the company's store to their home. In a subsequent fulfillment phase, the deliveries of the orders take place (compare Fleckenstein et al. 2023). Our problem focuses on the fulfillment phase, thus, all orders are known and now need to be delivered until the end of the day. For fulfillment, the company can access two sources, their own fleet and occasional drivers (ODRs). The ODRs are in-store customers, thus, they are unknown initially and appear dynamically over the course of the fulfillment phase. They indicate via app that they are generally willing to deliver bundles of orders for some compensation. The app also contains the ODR's individual preferences indicating if a bundle is acceptable (e.g., not too much travel) and the required compensation the individual ODR expects.

At equidistant time steps (e.g., every ten minutes), the company now checks the bids of the ODRs currently in the system. The bids are generated automatically by the app and indicate what bundles are acceptable for the ODRs and what compensation price they expect per bundle. Thus, the ODRs can bid on all possible bundles. Based on the bids, the company assigns bundles to ODRs and compensates them accordingly. Then, the ODRs deliver the corresponding goods. We assume that each ODR is only available in one time step and leaves the system either to deliver goods or unassigned afterwards. At the end of the fulfillment phase, the remaining orders are delivered by company drivers, e.g., company workers that agreed to deliver goods in the last hours of their shift (Boysen et al. 2022). The goal of the company is to minimize the total cost, which is the sum of the routing cost of the company drivers and the cost associated with compensation prices paid for the accepted bids.

3.2. Sequential decision process

The problem is a sequential decision process as decisions are made repeatedly over a time horizon (compare Powell 2022). A sequential decision process describes sequences of states, decisions and corresponding cost, revelation of stochastic information, and transition to new states. A special feature in our problem is that the the final decision state, the routing of the own fleet, differs in decisions and cost. In the following, we describe the components for our problem in detail. We start with the introduction of global notation.

Global notation. The set of orders is denoted I, where customer i requests a given quantity q_i. The depot (store, etc.) is denoted by 0. The set of nodes involved in the network is $I^0 = I \cup 0$. For each pair of nodes $i, j \in I^0$, we assume a company vehicle travel cost c^v_{ij}, which is known in advance. The capacity of both a company and ODR's vehicle is Q_{max}. The company fleet is composed of a sufficiently large number of m identical vehicles. Each vehicle in the company fleet starts from the depot and must return to it.

States. Decision are made in equidistant time steps $t = 0, 1, \dots, T$. Decision states in time t are indicated by S_t and include information about the orders still to serve $I_t \subset I$ and the n_t ODRs in the system with their bids and compensation, $O_t = (B_t, C_t)$. Let $P_t = \mathcal{P}(I_t)$ be the ordered power set of I_t. Then, B_t is a binary matrix of size $n_t \times |P_t|$ with $B_{opt} = 1$ if ODR o is willing to serve bundle p and zero else. The compensation is modeled similarly as a $n_t \times |P_t|$-matrix C_t with positive compensation values C_{opt} if $B_{opt} = 1$ and zero else. A state can therefore be summarized as $S_t = (I_t, n_t, O_t)$

The initial state is $S_0 = (I, 0, -)$ with all orders in the system and no ODRs arrived yet.

Decisions. A decision x_t in time $t = 1, \dots, T-1$ assigns bundles to ODRs. A decision can also be represented by a $k_t \times |P_t|$-matrix with entries $x_{opt} = 1$ if bundle p is assigned to supplier o. Let b^x_t be the set of selected bundles by decision x_t. A decision x_t is feasible if $x_{opt} \leq B_{opt} \, \forall o = 1, \dots, n_t, p = 1, \dots, |P_t|$ (no assignment of bundle p if no bid by ODR o), if $\sum_{p=1,\dots,P_t} x_{opt} \leq 1$, $\forall o = 1, \dots, n_t$ (each ODR is assigned at most one bundle), and, if the set of assigned bundles b^x_t is disjoint, i.e, an order is not assigned to two ODRs.

The cost $C(S_t, x_t)$ of a decision x_t in state S_t is the overall compensation paid:

$$C(S_t, x_t) = \sum_{o=1,\dots,n_t} \sum_{p=1,\dots,P_t} C_{opt} \times x_{opt}. \tag{1}$$

Post-decision states. The combination of state S_t and decision x_t leads to a post-decision state $S_t^x = (I_k^x)$ containing the remaining orders to serve $I_k^x = I_k \setminus \bigcup_{b \in b_t^x} b$.

Stochastic information and transition. The stochastic information $\omega_{t+1} = (n_{t+1}, O_{t+1})$ comprises a set of n_{t+1} new ODRs with their bids O_{t+1}. The new state in $t+1$ is then $S_{t+1} = (I_t^x, n_{t+1}, O_{t+1})$.

Final state and decision. In the final state $S_T = (I_{T-1}^x)$, the remaining orders are served by the company fleet. Thus, the decision x_T is S_T determines the solution of a vehicle routing problem with cost $C(S_T, x_T)$ being the overall routing cost.

Solution and objective. The solution of the problem is a policy $\pi \in \Pi$. A policy assigns a decision $x_t = X^\pi(S_t)$ to every state S_t. The overall set of policies is defined as Π. An optimal solution $\pi^* \in \Pi$ minimizes the expected cost:

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}}\, \mathbb{E}\left[\sum_{t \in T} (C(S_t, X^\pi(S_t)) | S_0)\right], \tag{2}$$

starting from state S_0.

3.3. Example

In the following, we give a small example to illustrate the process. Figure 1 shows a state in decision epoch 5. The left portion of the figure depicts the orders remaining to be served after epoch 4. The middle portion depicts the state with bids and prices of the ODRs. The right portion shows a potential decision. In the example, after epoch 4, six orders remain in the system (A,B,C,D,E,F). The decision state in the middle of Figure 1 now depicts the arrival of two ODRs and their bids. Occasional driver 1 is less flexible and less costly compared to ODR2. Driver 1 bids on only one bundle, (A,E), and offers to serve orders A and E for a compensation of 3. Occasional driver 2 bids on two bundles, (B,E) and (C,D). For serving orders B and E, driver 2 requests compensation of 4 and a compensation of 5 for serving orders C and D. We observe that not all orders are covered and that order E is covered by two bundles. For this example, there are five different decisions:

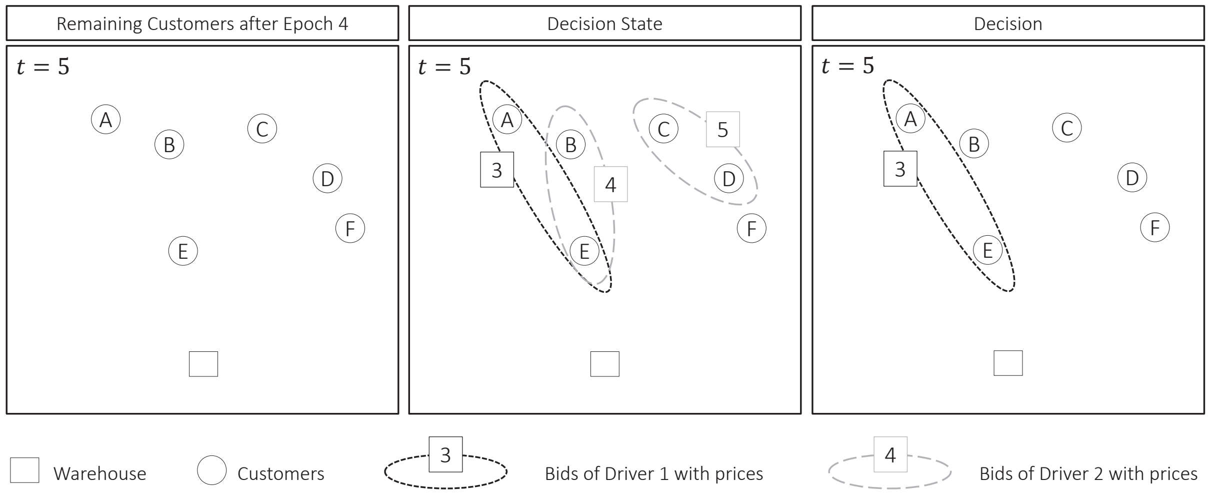

Figure 1 **Example for a decision state and a decision**

Decision	ODR1	ODR2	Cost	Remaining Orders
1	-	-	0	A,B,C,D,E,F
2	(A,E)	-	3	B,C,D,F
3	-	(B,E)	4	A,C,D,F
4	-	(C,D)	5	A,B,E,F
5	(A,E)	(C,D)	8	B,F

The first decision is to assign no bundle at all, leading to no cost and all orders remaining in the system. Decision 2 to 4 assign exactly one bundle and decision 5 assigns one bundle to each driver, bundle (A,E) to ODR1 and bundle (C,D) to ODR2. We note that decision (A,E) to ODR1 and (B,E) to ODR2 is infeasible as each order can maximally be assigned to one ODR. Decision (B,E) and (C,D) to ODR2 is infeasible as each ODR can be assigned at most one bundle.

The quality of a decision depends on two factors, the immediate compensation price a decision incurs and the expected cost to serve the remaining orders. Decision 1, no assignments, does not lead to any immediate cost, but all orders remain in the system with only a limited number of opportunities to assign them to future ODRs. Decision 5, (A,E) to ODR1 and (C,D) to ODR2, leads to only two orders, B and F, remaining in the system, however, at high cost. Further, the two orders are relatively far apart making a joint service of both via one ODR unlikely and a delivery by the company fleet expensive. A potentially reasonable decision, decision 2, is shown on the right side of Figure 1. This decision assigns bundle (A,E) to ODR1 at a comparably small compensation price of 3 and no bundle to ODR2. The remaining four orders B,C and D,F are relatively close to each other, thus, potential future ODRs may bid on the two bundles, (B,C) and (D,F). Alternatively, the routing cost of the company fleet for serving the four orders is manageable.

Even the small example already shows the challenges of decision making. In every state, an assignment needs to be found that not only is cost-efficient now but that ensures flexibility with respect to future ODRs and their assignments as well as an efficient final routing of the company fleet. Furthermore, while early in the fulfillment phase, assignments can be made in a rather selective manner in anticipation of future more cost-efficient options, in later phases, clearing the number of orders in the system becomes more important to avoid high routing cost in the end. Finally, when ODRs arrive, decisions need to be made in real-time. We incorporate these considerations a heuristic solution method.

4. The bundle price consensus policy

For our setting, a method is required that assigns bundles to ODRs in real-time while anticipating future bundle assignment opportunities as well as fulfillment cost of the own fleet.

The idea of our method is to pre-generate a large but limited set of popular bundles, i.e., only a subset of the computationally expensive power set. This is done based on an established procedure presented in Mancini and Gansterer (2022). It considers the expected preferences of ODRs to satisfy deliveries along their way to their destination and creates longer-stretched bundles similar to the bundles in Figure 1, which have been proved to be more attractive for the ODRs and more convenient for the company,

Now, for each bundle, we determine a popularity threshold and a maximum compensation price threshold a priori. Then, in a state, we only consider bids which respect both (i.e., before we admit a bid in the auction process, we seek the *consensus* of both popularity and compensation price.). These thresholds are not static, but time-dependent, and more precisely become less selective at each time-step. The idea is that, at the beginning of the time horizon, when we still have several future unknown opportunities in front of us, we can be more selective and accept only very good offers for very convenient bundles, while, as the end of the time horizon approaches and the future opportunities decrease, we become less restrictive in our bid selection in order to avoid orders remaining unassigned and requiring delivery by the company fleet. This procedure also avoids the issue raised in the example of myopically breaking up prominent bundles.

To determine popularity and price thresholds of bundles, a detailed and holistic view on the system is required. Our method pre-evaluates the bundles by a set of ODR-arrival scenarios related to the idea presented in Baty et al. (2023) for a dynamic dispatching problem. Each scenario is solved as a deterministic perfect information problem, in which all the ODRs future arrivals are known at the beginning of the time horizon. The method then counts how often a bundle was selected and to what average compensation. It also approximates the final routing cost of each bundle when not assigned. This allows identifying popular bundles that are likely effective and estimating if a bid for a bundle is cost-efficient. It also moves the computational burden to the beginning of the fulfillment phase and allows instant decisions when ODRs arrive in the system. Then, at each time step in our assignment optimization, we only consider bids addressing bundles above the popularity (bundle) threshold, and for which the price offered is below a (compensation price) threshold. Since the calculation based on perfect information values is likely too optimistic in prices and does not account for decreasing opportunities over time, we adapt both the popularity and price thresholds over time, decreasing the first and increasing the second.

All the bids passing both threshold checks are admitted in the eligible bids set. Then, a static problem is solved considering only bids in the eligible sets, aiming at minimizing the prices of accepted bids plus the proxy cost for orders not included in any accepted bid. The proxy cost reflects the approximate routing cost of the company fleet. If some orders are still in the system at the end of the time horizon, then a vehicle routing problem is solved to minimize the overall routing cost. We denote our policy the bundle price consensus (BPC) policy.

In the framework of Powell (2022), our method can be seen as a state-dependent policy function approximation tuned by an a priori stochastic lookahead with integrated cost function approximation. A policy function approximation represents a strategy based on practical or analytical considerations, like our thresholds. A stochastic lookahead uses sampled information to derive decisions (Soeffker et al. 2022). Our stochastic lookahead is different from the literature, since we apply it a priory to allow for instant decision making during the process. Also, in our stochastic lookahead, decisions are made with respect to an adapted cost function including the proxy cost. This is known as cost function approximation. In the following, we give details on how the thresholds are generated, how the set of bundles is determined, how the routing cost per bundle is approximated, and how the perfect information problem is solved.

4.1. Threshold generation

In this section, we first illustrate the process of threshold generation and then present the algorithmic details.

Process illustration. To illustrate the process, we rely on the example setting introduced in Section 3. The process is shown in Figure 2. The figure contains three main parts indicated by roman numbers: I. The initial bundle generation, II. The sampling of scenarios and their solution, and III. the calculation of the thresholds. We now discuss each part for the example:

I. Bundle Generation. For the set of initial orders, bundles are generated. In the example, we assume the same six orders as before. For the purpose of presentation, we assume that the bundle generation step provides four different bundles: (A,E), (B,E), (C,D), and (D,F). Evidently, the number of potential bundles is substantially higher in our experiments.

II. Scenario Sampling and Solution. Next, different ODR-scenarios are sampled. In the example, we rely on only two sampled scenarios shown in the upper and lower right part of the figure. In the first scenario, we assume bids from three different ODRs, indicated by the different lines (dotted, dashed-dotted, solid). The dotted-line ODR bids 3 for bundle (A,E). The dashed-dotted line ODR bids for two bundles, 5 for bundle (B,E) and 2 for bundle (C,D). The solid line ODR bids 3 for

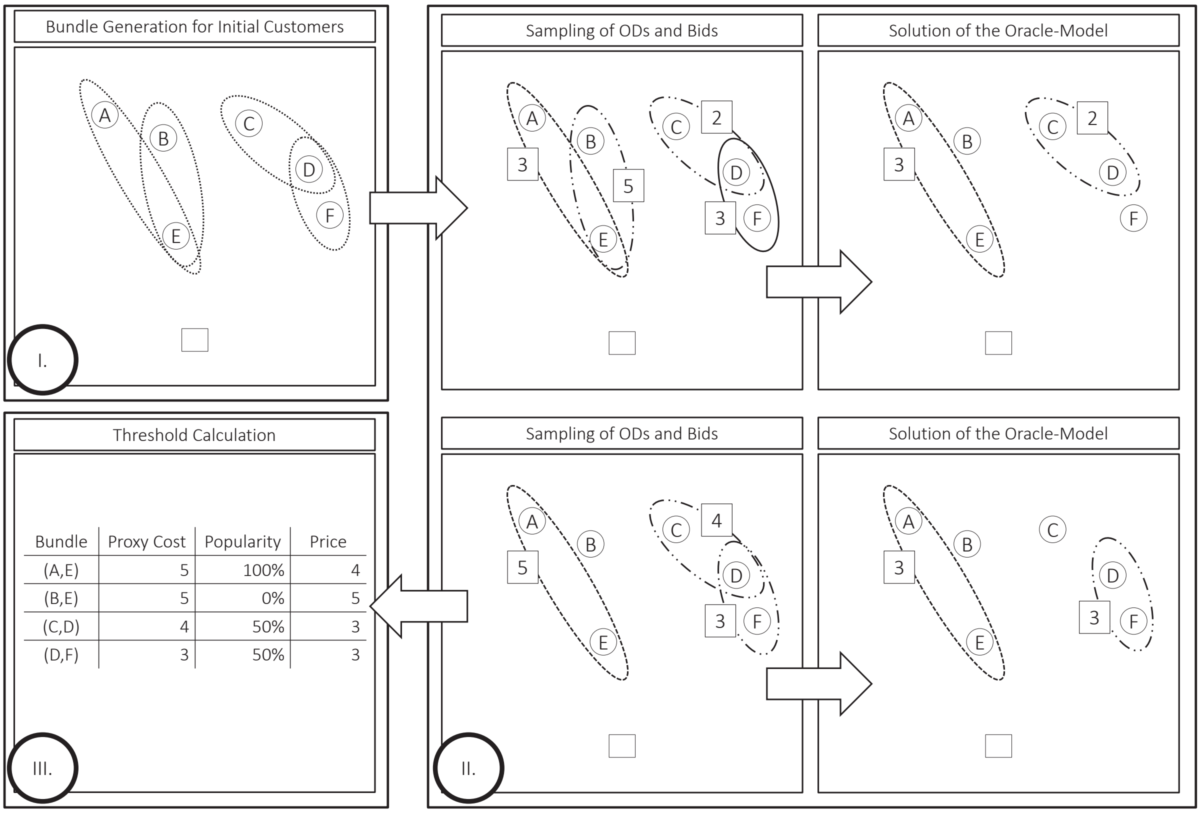

Figure 2 **Illustration of the Threshold Generation Process**

bundle (D,F). The deterministic scenario is now solved by an oracle model. The solution is shown on the top right. Bundle (A,E) and bundle (C,D) are assigned to the corresponding ODRs with compensation prices of 3 and 2. Orders B and F are served by the company fleet. The second scenario on the bottom right of the figure provides different ODRs with different bids. It is treated similarly and the oracle solution is that bundles (A,E) and (D,F) are both assigned at price 3 and orders B and C are served by the company fleet.

III. Threshold Calculation. In the final step, the thresholds for popularity and compensation prices are calculated. The popularity is covered by the percentage of selections over all scenarios. Bundle (A,E) was selected in all scenarios and therefore has a popularity of 100%. Bundle (B,E) was never selected, its popularity is 0%. The other two bundles, (C,D) and (D,F), were selected once leading to a popularity of 50%. The average popularity over all bundles is 50% as well. Calculating the average compensation prices only based on the scenarios with successful assignments will likely lead to instability and overly optimistic prices. Thus, the average compensation prices are a combination of the observed prices and proxy cost for the routing in case the bundle was not assigned in a scenario. In the example, the proxy costs for bundles (A,E) and (B,E) are 5 since the respective two orders are comparably far apart. The proxy cost for (C,D) is 4. For (D,F), it is 3. This leads to compensation price thresholds of 4 for (A,E), $(100\% \times 4 + 0\% \times 5)$, 5 for (B,E) (only proxy cost), and 3 for both (C,D) and (D,F) (half proxy cost, half compensation prices).

Going back to the example in Section 3, the derived thresholds will likely induce the selected decision of only assigning bundle (A,E) at price of 3 and not assigning bundle (B,E) for price of 4 or bundle (C,D) for price of 5. Bundle (A,E) is very popular and the price of 3 is below the average compensation price. Bundle (B,E) may have a smaller compensation price than the average price of 5, however, it is very unpopular with respect to the scenario solutions. And bundle (C,D) may have average popularity, however the compensation price is higher than the average price of 3 (We note that this might change in later time steps as the thresholds are softened over time).

Algorithm. The algorithmic procedure for generating the thresholds is shown in Algorithm 1. Inputs are the set of orders I, the pre-generated set of bundles $\hat{P}$ and a set of K scenarios of ODR-arrivals $\Omega_1, \ldots, \Omega_K$. Inputs are further the proxy cost π_p per bundle p in case it is not selected (we will define the proxy cost later in this section). Outputs are the average popularity σ_p^B and the average price σ_p^C for each bundle p. The algorithm iterates over the scenarios and solves the deterministic model with function *SolveModel*$(I, \hat{P}, \Omega)$. The obtained solution X is then used to update popularity and price over functions *IfSelected*(p, X), indicating if the bundle p was part of the solution and *Comp*(p, X) the corresponding compensation. Else, the proxy cost π_p is used.

Once we have obtained σ_p^B and σ_p^C for all the bundles in $\hat{P}$, we compute the average popularity μ^B as the ratio between the sum of popularity of all the bundles and the number of bundles:

$$\mu^B = \frac{\sum_{p \in \hat{P}} \sigma_p^B}{|\hat{P}|}.$$

Algorithm 1 Threshold generation process

Input: Orders I, Bundles $\hat{P}$, Scenarios $\Omega_1,\ldots,\Omega_K$, Proxy Cost π.

Output: Popularity σ_p^B, Price σ_p^C.

for $p \in \hat{P}$ **do**
    $\sigma_p^B \leftarrow 0$
    $\sigma_p^C \leftarrow 0$
end for
for $\Omega = \Omega_1,\ldots,\Omega_K$ **do**
    Solution $X \leftarrow \mathit{SolveModel}(I,\hat{P},\Omega)$
    for $p \in \hat{P}$ **do**
        if *IfSelected(p,X)* $==$ true **then**
            $\sigma_p^B \leftarrow \sigma_p^B + 1/K$
            $\sigma_p^C \leftarrow \sigma_p^C + \mathit{Comp}(p,X)$
        else
            $\sigma_p^C \leftarrow \sigma_p^C + \pi_p$
        end if
    end for
end for
return σ_p^B, σ_p^C

Whereas, the average (compensation or proxy) price per bundle, μ_C is computed as the sum of the price of the bundle over all the scenarios, divided by the number of scenarios:

$$\mu_p^C = \frac{\sigma_p^C}{K}.$$

At each state t, we consider only bids addressing bundles whose popularity is higher than $\mu^B - \epsilon^B(t-1)$, and with a price lower than $\mu_p^C + \epsilon^C(t-1)$. This linear change allows the threshold restrictions to become less tight as the states go by, allowing the system to be more restrictive at the beginning of the time horizon, in which several potential alternative options are yet to come and less restrictive in the last states of the time-horizon, in which the remaining opportunities are limited. After preliminary tuning tests, we set the number of scenarios to $K = 10$, the popularity threshold gradient to $\epsilon^B = 0.2$, and the compensation threshold gradient $\epsilon^C = 0.2$.

4.2. Bundle generation

Since ODRs are in-store customers willing to perform deliveries on their way back home, attractive bundles may contain not only orders close to each other but also those potentially very distant from each others but close to the direct path between the depot and ODRs' destinations. Such long-shaped bundles cannot be generated by means of classical clustering algorithms, which aim at minimizing intra-cluster distances. To generate them, we exploit the corridors-based approach, introduced by Mancini and Gansterer (2022). In order to make the paper self-contained, we briefly report in the following, how this procedure works.

This approach follows the idea of a Sweep-algorithm. We initially generate a sector defined by the smallest angle α, for which all the orders are included. This sector is further split into n_s identical small sectors characterized by an angle of size α/n_s. Each small sector is then explored separately. If the total demand of the orders included in the sector is smaller than the standard ODR capacity, we generate a single bundle containing all of them. Conversely, if it exceeds the vehicle capacity, a clustering-based approach is applied to these orders, to generate feasible bundles. The corridor-based approach is iteratively repeated with different values of n_s, namely, 10, 20 and 30. All the bundles obtained so far, excluding duplicates, constitute the bundles in $\hat{P}$.

4.3. Perfect information oracle model

In this section, we present a perfect information model which represents the perfect information decision problem if we know the entire set of bids we receive across all the time-horizon in advance. This model is needed for solving the scenarios. We refer to this model as the *oracle* as common in the literature on dynamic optimization problems. We note that for the oracle model, we do not require any time steps and can assume that all ODRs occur at once. The optimal solution of the oracle is a lower bound for a realization of our problem. We will use the oracle to solve the scenarios in our method and as a lower bound benchmark in our experiments.

Before we present the mathematical formulation of the problem, let us introduce some supplementary notation. In order to make the model more compact, we define the set of submitted bids as

$L = 1..N_l$. For each bid l we assume to know the bundle to which it is associated, τ_l, the ODR who placed it, $\hat{o}_l$, and the price offered, C_l. It is worth noting that only pairs (o,p) for which ODR o is willing to serve bundle p, generate a feasible bid l. Given the fact that, generally, the ODRs/bundles compatibility matrix is quite sparse, the number of variables involved in the compact formulation is much lower with respect to a more standard formulation in which variables are defined for all the (o,p) pairs.

The decision variables are reported in the following:

Z_i binary variable that takes value 1 if order i is visited by a company-owned vehicle and takes 0 otherwise

U_{ij} binary variable that takes value 1 if node j is visited by a company-owned vehicle just after node i and takes 0 otherwise

Y_l binary variables that take value 1 if bid l is accepted and take 0 otherwise

Q_i non-negative variables representing the cumulative load at node i, expressed as the total quantity of demand delivered by a vehicle along its route, when leaving node i

The model is then as follows:

$$\min \sum_{i \in I^0} \sum_{j \in I^0} c_{ij} U_{ij} + \sum_{l \in L} C_l Y_l \tag{3}$$

s.t.

$$\sum_{j \in I} U_{0j} \leq m \tag{4}$$

$$Z_j + \sum_{l \in l | j \in \tau_l} Y_l = 1 \qquad \forall j \in I \tag{5}$$

$$\sum_{l \in L | \hat{o}_l = o} Y_l \leq 1 \qquad \forall o \in O \tag{6}$$

$$\sum_{i \in I} U_{ij} = Z_j \qquad \forall j \in I^0 \tag{7}$$

$$\sum_{i\in I} U_{ij} = \sum_{i\in I} X_{ji} \qquad \forall j \in I^0 \tag{8}$$

$$Q_j \geq Q_i + q_j - 2Q_{max}(1 - U_{ij}) \qquad \forall i \in I \ \forall j \in I \tag{9}$$

$$0 \leq Q_j \leq Q_{max} \qquad \forall j \in I \tag{10}$$

$$U_{ij} \in 0,1 \qquad \forall i \in I^0 \ \forall j \in I^0 \tag{11}$$

$$Y_l \in 0,1 \qquad \forall l \in L \tag{12}$$

$$Z_i \in 0,1 \qquad \forall i \in I \tag{13}$$

The objective (3) aims to minimize the total cost for the company, expressed as the sum of the prices of accepted bids plus the routing cost for the orders served by the owned fleet. The number of vehicles exploited by the company cannot exceed the number of available vehicles in the owned fleet, as imposed by Constraints (4). Constraints (5) and (6) ensure that each order is either served by the company or assigned to one of the ODRs and that at most one bid per ODR can be accepted. If an order is served by the owned fleet, it must be visited only once, as stated in Constraints (7). Constraints (8) ensure routes continuity. Constraints (9) track cumulative load at the nodes and works as sub-tour elimination constraints. Constraints (10) ensure that vehicle capacity is respected. Finally, Constraints (11), (12), and (13) specify variable domains.

To strengthen the formulation, the following valid inequalities can be added to the model. To do that, we need to introduce a new set of variables R_{ij} that represent the load carried on arc (i, j).

$$\sum_{j\in I^0} R_{ji} - \sum_{j\in I^0} R_{ij} = q_i Z_i \qquad \forall i \in I \tag{14}$$

$$\sum_{j\in I^0} R_{j0} - \sum_{j\in I^0} R_{0j} = -\sum_{j\in I} q_j Z_j \tag{15}$$

$$R_{ij} \leq Q_{max} U_{ij} \qquad \forall i \in I_0 \ \forall j \in I^0 \tag{16}$$

$$R_{i0} = 0 \qquad \forall i \in I^0 \tag{17}$$

Constraints (14) state that the quantity delivered to each order exactly corresponds to its demand. Constraints (15) ensure that the total delivered quantity is equal to the sum of the demands of the orders directly served by the company. Constraints (16) ensure that the load carried by a vehicle never exceed its capacity. Finally, Constraints (17) forces vehicles to return empty to the depot. The effectiveness of exploiting these valid inequalities in terms of computational times reduction has been shown in Mancini and Gansterer (2022).

4.4. Proxy cost calculation

If a bundle has not been selected in a specific scenario, the compensation price is assumed to be equal to a proxy cost, π_p, representing the cost of serving the orders belonging to bundle p by the own fleet. Such cost is not trivial to estimate, since it does not only depend on the order's location, but also on the location of the other orders that must be served by the own fleet. We compute this proxy cost depending on the order's location and on the number of orders located nearby, according to the following procedure. We split the service area in identically sized non-overlapping squares. For each square s, we refer to the number of orders located in s as N_s. We also identify primary and secondary neighborhood squares with respect to s, depicted in Figure 3 in light blue and turquoise, respectively. The number of orders located in primary and secondary neighborhood squares are identified as $N_i'^s$ and $N_i''^s$, respectively. The distance between the depot and the center of the square s is indicated as d_s. The proxy cost for a single order i is then determined as:

$$\pi_i = \frac{2 \times d_{s_i} \times \gamma}{N_i^s + \alpha' N_i'^s + \alpha'' N_i''^s}$$

where $0 \leq \alpha'' \leq \alpha' \leq 1$.

Then π_p can be obtained as the average of π_i among all orders i belonging to bundle p, divided by the cardinality of bundle p, i.e. the number of orders belonging to it. After preliminary tests, we split the customer area in 16 squares (4×4 grid) and set $\alpha' = 0.1$ and $\alpha'' = 0.001$.

5. Computational Study

In this section, we present our computational study. We first describe the test instances and the benchmark policies. Then, we present the objective values and analyze the decision making of the different policies. Finally, we illustrate the effectiveness of our method's individual components.

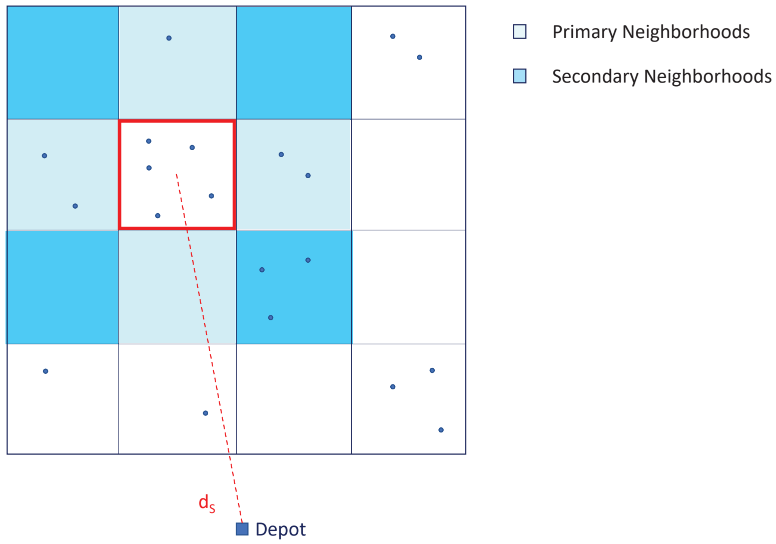

Figure 3 **Computing the proxy distance**

5.1. Instances

The instances are derived from those introduced by Mancini and Gansterer (2022). They are composed of 20 orders. Because we assume private vehicles for the ODRs and the employees of the company fleet, we draw the required capacity for orders from a uniform distribution in the range of 10% to 30% of the overall vehicle capacity, i.e., a vehicle can carry about five orders. We consider 10 time steps. At each time step, the number of occasional drivers available is uniformly distributed between 1 and 6. As with the orders, the ODRs' home locations are randomly distributed within the 10km times 10 km service area. The depot is located in the south, outside of the service area. This resembles customers traveling from work shopping on their way home.

The bids and compensation of ODRs are modeled as follows. The flexibility of ODRs is modeled via a maximum detour threshold that is randomly selected among 1,2 and 3 kilometers. The ODRs' willingness to participate is modeled by a parameter that represents the relative cost rate compared to average market prices. We randomly sample the parameter among 0.8, 1 and 1.2 for each ODR. A value of one means that the bid is truthful with respect to the detour, 0.8 means that a 20% discount is applied, 1.2 that the bid is 20% above market price. For each bundle, we then compute the detour required, and, if feasible, calculate the truthful bid using a travel cost (1.5 EUR per kilometer) for the detour plus a fixed cost per stop (1.5 EUR) and a fixed participation cost (2 EUR). We then multiply the truthful price for the willingness multiplier to obtain the actual bid price. Vehicle speed of the company's fleet is set to 20km per hour. The number of vehicles is $m = 5$. This allows serving all orders with the fleet. We performed tests with different unitary routing cost for the owned fleet, namely *low* cost (0.5 EUR per kilometer), *normal* cost (1 EUR), and *high* cost (1.5 EUR).

5.2. Benchmark policies

In total, we present eight benchmark policies, two problem-oriented policies, four method-oriented policies, and two business concept-oriented policies. We compare all policies to a perfect information lower bound where the perfect information instance is solved via the oracle model.

The two problem-oriented policies propose alternative ways of assigning bundles to ODRs:

Clearance: The goal of the policy is to clear all the orders as soon as possible to avoid high routing cost in the end. In each state, we search for the assignment that maximizes the number of orders served. In case of a tie, we select the decision which serves them at a lower cost. All the served orders are assigned to the correspondent ODR, while remaining ones are kept for further time slots. To serve unassigned orders after the last time slot, a vehicle routing problem is solved similarly to our main method.

Anticipatory Clearance: This policy extends the *Clearance* policy by considering potential future ODRs. In each state, this policy samples future ODR-arrivals for the remaining time slots. We solve the same optimization problem as in the *Clearance* policy, but we consider bids submitted by the real ODRs and by the sampled ODRs. All the bundles served by real ODRs are assigned to them, while all those assigned to sampled ODRs are kept in the system.

We further apply four method-oriented benchmark policies to analyze the value of the components of our *Bundle Price Consensus* (BPC) policy that uses two time-dependent thresholds on bundle popularity and compensation price and that uses a stochastic lookahead including proxy cost approximation for deriving the thresholds.

Bundle Consensus (BC): This policy is similar to BPC, however, instead of checking both the popularity and price threshold, this policy only considers the popularity threshold. Thus, in the example in Section 3, it will assign both bundles (A,E) and (C,D).

Price Consensus (PC): This policy only considers the price-thresholds. Thus, in the example in Section 3, it will be indifferent between bundles (A,E) and (B,E).

BPC-Fixed Threshold (BPC-FT): This policy is similar to our BPC-policy, however, we assume that the thresholds are fixed over the time horizon. The fixed threshold is equal to the one used in BPC at the first time step. It can be obtained by fixing ϵ_B and ϵ_C equal to 0.

BPC-No Threshold (BPC-NT): This policy is similar to BPC but does not set any thresholds. Thus, it solely relies on balancing immediate cost and proxy routing cost. It is therefore a cost function approximation and its performance is an indicator of the effectiveness of the proxy cost approximation.

The two business concept-oriented policies allow analyzing the value of offering bundles to ODRs and using ODRs for delivery:

No ODRs: For this policy, we serve all orders with our own fleet. To this end, we solve the vehicle routing problem for all orders. We set the maximum runtime per realization to one hour.

No Bundles: This policy follows exactly the procedure of BPC, but only considers bundles of size 1. It therefore allows to analyze the value of bundling orders for delivery.

5.3. Objective and decision making

First, we show the effectiveness of our policy compared to the problem-oriented benchmark policies and analyze the difference in their decision making. To this end, we calculate the objective values over ten instance realizations for each routing cost setting for the policies BPC, *Clearance*, and *Anticipatory Clearance*. We then calculate the average gap with respect to the perfect information solutions. The results are shown in Figure 4.

The x-axis depicts the different routing cost settings. The y-axis shows the average gap. We observe that our policy outperforms the benchmark policies significantly. The gaps of BPC are

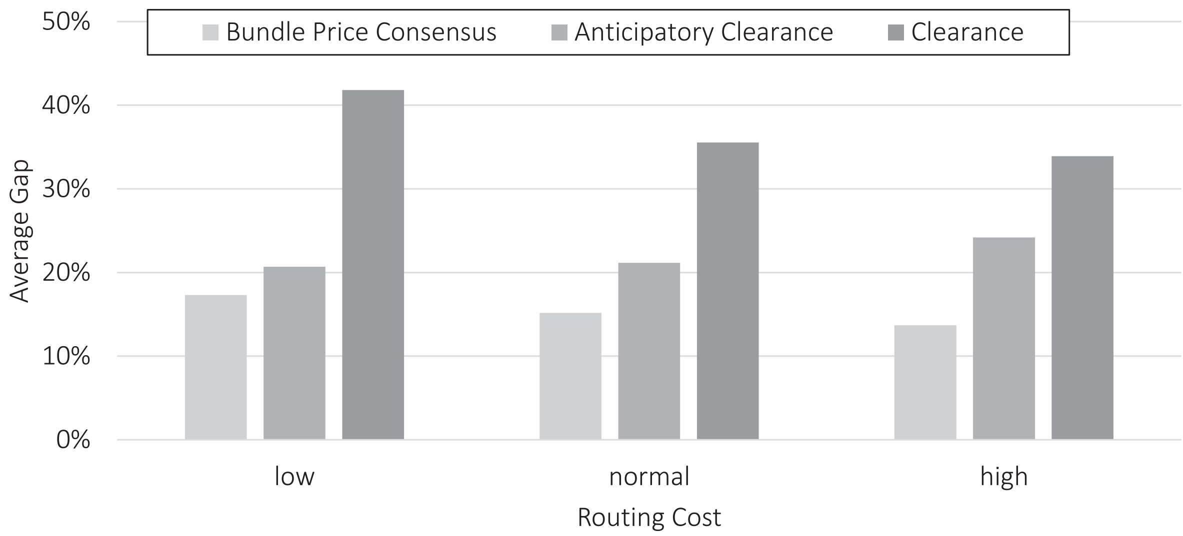

Figure 4 **Comparison of the perfect information gaps for the method-oriented benchmark policies for different routing costs.**

always between 10% and 20% while the gaps for *Anticipatory Clearance* range between 20% and 30% and the gaps for *Clearance* range between 30% and 50%. The latter indicates that greedily clearing the number of orders in the system comes at high cost of compensation, and, as we show next, does not necessarily lead to lower routing cost for the own fleet.

To analyze the cost of compensation prices and routing, we focus on the *normal* routing cost setting and calculate the average cost for compensation prices and routing for all policies and the oracle solution. For the purpose of presentation, we normalize the cost values with respect to the total oracle solution cost. The results are shown in Figure 5. The x-axis depicts the relative cost of routing and compensation prices with respect to the total oracle cost. That means that for the oracle solutions, the sum of routing and compensation is 100% overall. We note that for the oracle solution, the compensation cost is twice as high as the routing cost. We see that our BPC-policy achieves nearly the same routing cost as the oracle and slightly higher compensation cost. *Clearance* results in very high compensation cost while *Anticipatory Clearance* can reduce compensation cost even below BPC, however, at the expense of very high routing cost.

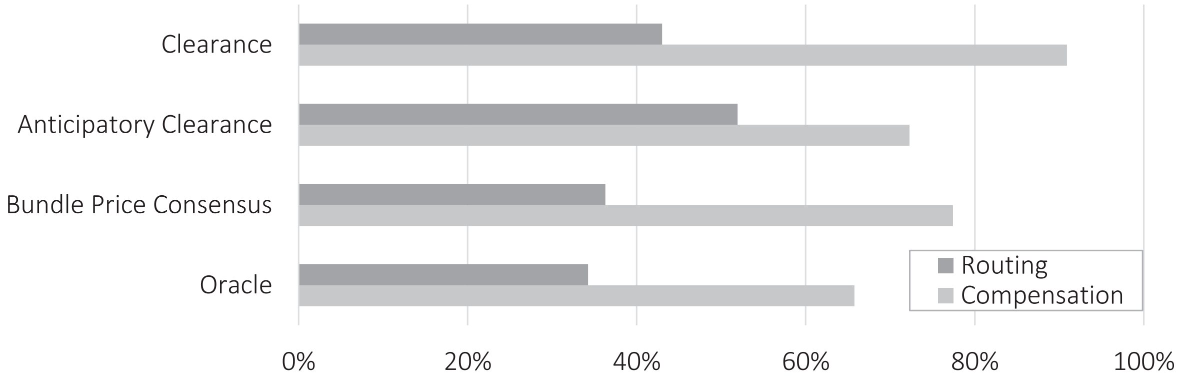

Figure 5 **Routing and ODR related costs for different policies and normal routing cost.**

It is noteworthy that the routing cost for *Clearance* is higher than for BPC even though *Clearance* aims on leaving as few orders as possible in the system to save routing cost in the end. However, it ignores that assigning some bundles now may destroy profitable bundles for the future and may leave orders stranded throughout the service area. Our BPC-policy carefully evaluates potential assignments and may decide against assigning a bundle in anticipation of future opportunities and routing cost.

This is also reflected when analyzing the average number of orders in the system of the time steps. This is shown in Figure 6. The x-axis shows the ten time steps. The y-axis indicates the average remaining number of orders after the time step. The value at time step 10 therefore indicates the average number of orders that need to be served by the company's fleet. We observe that the number of orders for the oracle solutions are usually highest while the number of orders for the *Clearance* policy are always lowest as could be expected. The first can be explained by the fact, that the perfect information oracle "knows" the future ODRs and, therefore, can postpone assignments without risk. Interestingly, our BPC-policy can catch up with the *Clearance* policy and, in the end, has nearly the same number of orders remaining to be served by the same fleet (though, at substantially lower cost as we have shown in Figure 5). Thus, with a few careful postponements in the beginning, BPC allows cheaper and more assignments later, likely, because prominent bundle opportunities are saved by not assigning "wrong" bundles earlier. We also note that while the

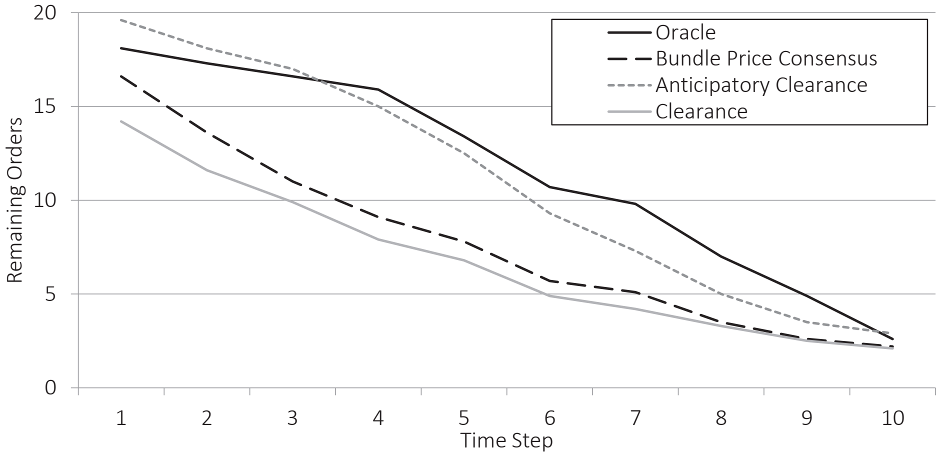

Figure 6 **Comparison of average number of orders in the system at different time steps for different policies and normal routing cost.**

average number of remaining orders are around 2 to 3 for BPC, the individual instances differ significantly. In about half the instances, about four or five orders remain, for the other half, all orders are cleared. Since vehicles have capacity for about five orders, our method either sends out one vehicle or no vehicle in the end.

Next, we analyze the type of bundles that are assigned. To this end, we count the number of assignments for different bundle sizes for the three policies *Clearance*, *Anticipatory Clearance* and BPC and calculate the relative frequency for each size. The results are depicted in Figure 7. The x-axis depicts the relative frequency and the number on the bars indicate the corresponding bundle size. Although the maximum size of the bundles offered is six, only very few bundles contain six orders, and none of them is selected by any policy. Therefore, we discard them in this analysis. We recall that the overall number of assigned orders are about the same for all policies as seen in Figure 6. However, the bundle sizes are very different. Notably, the *Clearance*-policy and also the *Anticipatory Clearance*-policy assign substantially more bundles of size one but also more bundles of size five compared to BPC. Our policy shows a more balanced distribution, likely because too small bundles might prohibit assignments of prominent bundles in the future.

5.4. Method

Next, we analyze the components of our method, namely the value of the stochastic lookahead, the two time-dependent thresholds and the routing cost approximation. To this end, similar to Figure 4

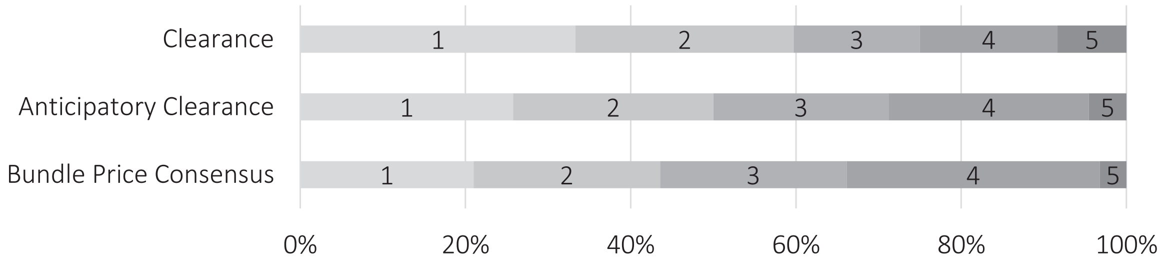

Figure 7 **Relative distribution of bundle sizes.**

we compare the average gap of our BPC with the four method-oriented benchmarks, only the bundle popularity threshold (BC), only the compensation price threshold (PC), fixed thresholds BPC-FT, and no thresholds at all (BPC). The results are shown in Figure 8.

We observe that BPC always provides the best solutions with gaps around 15%. The two policies with individual thresholds, BC (around 20%) and PC (20%-30%), perform comparably well but still significantly worse. This indicates that a joined consideration of popularity and price is required. Notably, BC outperforms PC for all settings. Thus, for our setting, bundle popularity seems to be more important than compensation prices. Policy BPC-FT performs significantly worse than the policies with time-dependent thresholds, reaching gaps of 30%-40%. Its performance for the setting with high routing cost is especially poor. The policy ignores that assignment opportunities decrease over time. Therefore, it ends up with more orders at the end to be served by the company's fleet. This is particularly expensive if the fleet's routing cost is high. The opposite development can be seen for the policy without thresholds BPC-NT. For low routing cost, it performs similarly to the greedy *Clearance*-policy with a gap of about 40%. With increasing routing cost, its relative performance increases as well with a gap of about 20%. As this policy balances compensation prices with approximated routing cost, it anticipates the cost of not assigning a bundle. Once the routing cost become more important, this anticipation becomes more valuable as well. The relatively strong performance of this policy also confirms the value of the routing proxy cost approximation.

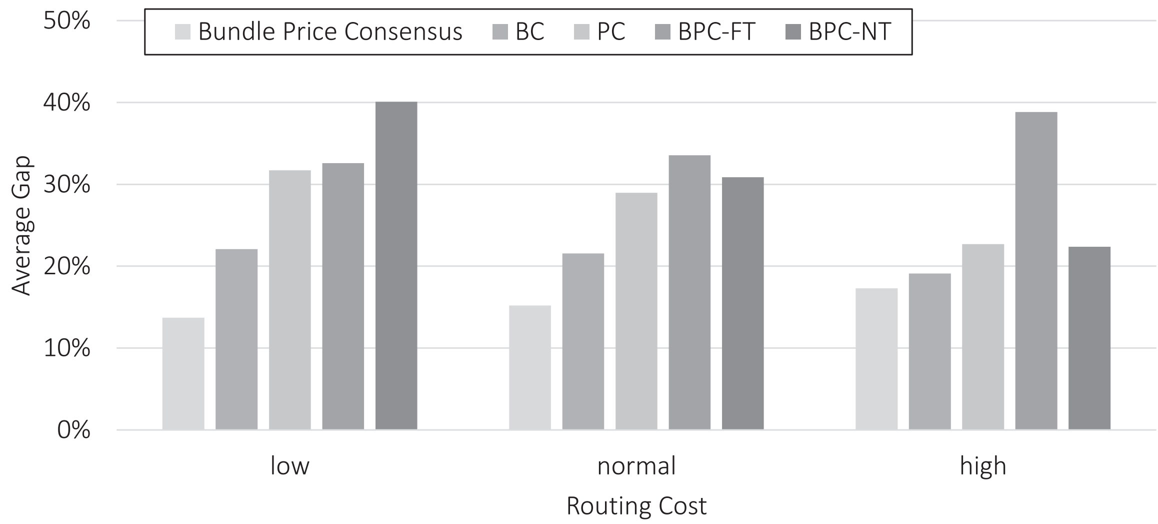

Figure 8 **Comparison of the perfect information gaps for the method-oriented benchmark policies for different routing costs.**

5.5. Business concept

Finally, we analyze the value of the business concept of assigning bundles of orders to dynamically occurring ODRs. To this end, we compare BPC to a policy only assigning single orders to ODRs and a policy not relying on ODRs at all. The results are prepared similarly to the other experiments. They are presented in Figure 9. We note that the range of the y-axis is significantly larger than for the previous policies as some experiments result in cost more than twice as high compared to the perfect information solution.

First, we observe that the No Bundles-policy is always substantially worse than BPC. This indicates that it is very valuable for a company to offer bundles instead of individual orders. The reasons for the gap are twofold. First, when only offering single orders, the cost per order as a combination of fixed cost and detour cost is higher compared to a case where orders are bundled. The participation cost is only paid once and the detour cost is smaller due to the triangular inequality. Second, the willingness to participate for ODRs is higher when bundles are offered. The reason is the same, that the suppliers earn more by traveling relatively less per delivery. When analyzing the value of using ODRs for delivery, we observe very different results with respect to the routing

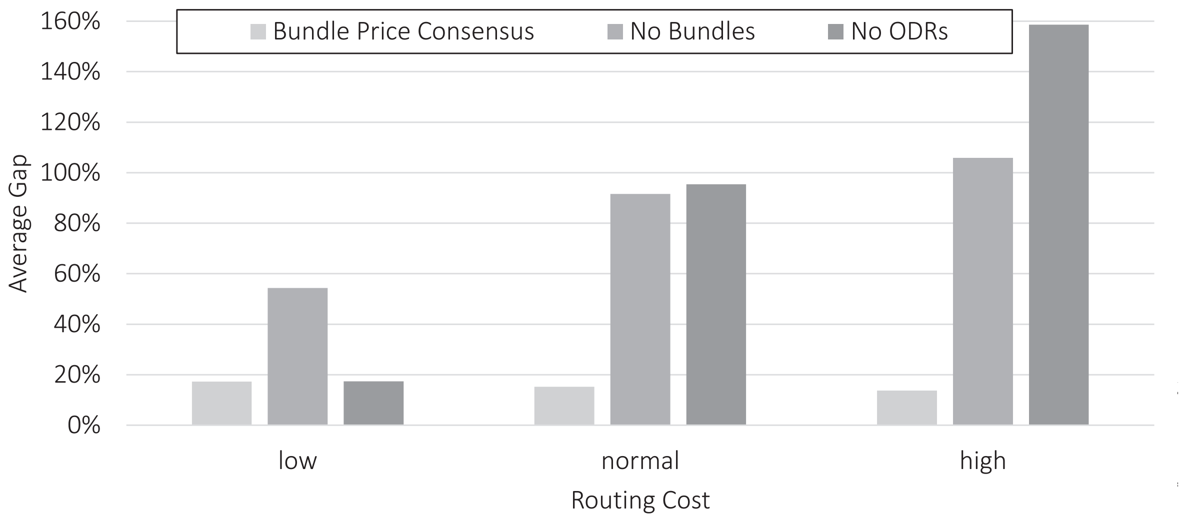

Figure 9 **Comparison of the perfect information gaps for the business concept-oriented benchmark policies for different routing costs.**

cost of the company fleet. In case routing cost is low, BPC (17.3% gap) and No ODRs (17.4% gap) perform nearly the same. The value of using ODRs is very small and companies may decide against the operational complexity of adding ODRs to their delivery operations. Once the cost increases to normal, the benefit of BPC becomes significant (15.2% vs. 95.4% gap). Companies should consider adding ODRs to their delivery concepts in case bundles are assigned in an anticipatory manner. Adding ODRs but only assigning individual orders does not yield a larger benefit though (91.6% vs. 95.4% gap). In the case of high routing cost, ODRs become particularly valuable, even assigning individual orders (105.9% vs. 158.6% gap), but especially when assigning bundles with BPC (13.7% vs. 158.6% gap).

6. Conclusion

In this work we have shown how the dynamic and anticipatory assignment of order bundles to occasional drivers can reduce delivery cost substantially. There are a variety of avenues for future research. In our problem, we introduced the concept of bundles to dynamic delivery with crowd-sourced couriers. We focused on the case where all orders are known a priori and have to be picked up at the same location by in-store customers. Remaining unassigned orders are delivered by the company fleet at the end of the process. Future research may extend the problem in several dimensions. First, instead of a final routing in the end of the process, the company may decide to dynamically route delivery vehicles to ensure fast delivery or even time windows Baty et al. (2023). This increases complexity since now dynamic decisions are made about the dispatching of the vehicles and the unassigned orders to deliver. Such orders should be located close to each other and, at the same time, should not break up attractive bundles. Here, the idea of popularity might be used to filter for orders that are not part of any attractive bundle. Another extension may be that not only the arrivals of occasional drivers but also of orders are stochastic and dynamic (Liu and Luo 2023). In that case, bundles are created dynamically as well. While our threshold-concept might still apply in such a settings, alternative measures for compensation prices and popularity must be developed to account for the uncertainty in the set of orders. Also, instead of considering a single pickup location (the store), delivery might be possible from multiple stores in the city or even well-located micro hubs nearby (Mousavi et al. 2022, Kızıl and Yıldız 2023). Here, decisions about the assignment of orders to stores must be made as well as potential transportation to the hubs. The proposed model and method may be connected with the work of Dayarian and Pazour (2022) optimizing assignments of orders to in-store customers with both picking and routing in mind. Finally, the problem may be extended to a crowdsourced pickup and delivery problem as proposed in Arslan et al. (2019), or, even for crowdsourced passenger transportation (Haferkamp et al. 2023).

Acknowledgments

Marlin Ulmer's work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Emmy Noether Programme, project 444657906. We gratefully acknowledge their support. Margaretha Gansterer's work is funded by FWF Austrian Science Fund: P 34502-N.

References

Ahamed, T., B. Zou, N. P. Farazi, and T. Tulabandhula (2021). Deep reinforcement learning for crowdsourced urban delivery. *Transportation Research Part B: Methodological 152*, 227–257.

Al Hla, Y. A., M. Othman, and Y. Saleh (2019). Optimising an eco-friendly vehicle routing problem model using regular and occasional drivers integrated with driver behaviour control. *Journal of Cleaner Production 234*, 984–1001.

Alnaggar, A., F. Gzara, and J. H. Bookbinder (2021). Crowdsourced delivery: A review of platforms and academic literature. *Omega 98*, 102139.

Alnaggar, A., F. Gzara, and J. H. Bookbinder (2023). Compensation guarantees in crowdsourced delivery: Impact on platform and driver welfare. *Omega*, 102965.

Archetti, C., M. Savelsbergh, and M. G. Speranza (2016). The vehicle routing problem with occasional drivers. *European Journal of Operational Research 254*(2), 472–480.

Arslan, A., N. Agatz, A. Kroon, and R. Zuidwijk (2019). Crowdsourced delivery: A dynamic pickup and delivery problem with ad-hoc drivers. *Transportation Science 53*(1), 222–235.

Ausseil, R., J. A. Pazour, and M. W. Ulmer (2022). Supplier menus for dynamic matching in peer-to-peer transportation platforms. *Transportation Science 56*(5), 1304–1326.

Baty, L., K. Jungel, P. S. Klein, A. Parmentier, and M. Schiffer (2023). Combinatorial optimization enriched machine learning to solve the dynamic vehicle routing problem with time windows. *arXiv preprint arXiv:2304.00789*.

Boysen, N., S. Emde, and S. Schwerdfeger (2022). Crowdshipping by employees of distribution centers: Optimization approaches for matching supply and demand. *European Journal of Operational Research 296*(2), 539–556.

Cheng, S.-F., C. Chen, T. Kandappu, H. C. Lau, A. Misra, N. Jaiman, R. Tandriansyah, and D. Koh (2017). Scalable urban mobile crowdsourcing: Handling uncertainty in worker movement. *ACM Transactions on Intelligent Systems and Technology (TIST) 9*(3), 1–24.

Çınar, A. B., W. Dullaert, M. Leitner, R. Paradiso, and S. Waldherr (2023). The role of individual compensation and acceptance decisions in crowdsourced delivery. *arXiv preprint arXiv:2305.01317*.

Dahle, L., H. Andersson, M. Christiansen, and M. G. Speranza (2019). The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research 109*, 122–133.

Dayarian, I. and J. Pazour (2022). Crowdsourced order-fulfillment policies using in-store customers. *Production and Operations Management 31*(11), 4075–4094.

Dayarian, I. and M. Savelsbergh (2020). Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management 29*(9), 2153–2174.

Di Puglia Pugliese, L., D. Ferone, P. Festa, G. F., and M. G. (2022). Solution approaches for the vehicle routing problem with occasional drivers and time windows. *Optimization Methods and Software 37(4)*, 1384–1414.

Di Puglia Pugliese, L., D. Ferone, G. Macrina, P. Festa, and F. Guerriero (2023). The crowd-shipping with penalty cost function and uncertain travel times. *Omega 115*, 102776.

Fleckenstein, D., R. Klein, and C. Steinhardt (2023). Recent advances in integrating demand management and vehicle routing: A methodological review. *European Journal of Operational Research 306*(2), 499–518.

Gdowska, K., A. Viana, and J. P. Pedroso (2018). Stochastic last-mile delivery with crowdshipping. *Transportation Research Procedia 30*, 90–100.

Haferkamp, J., M. W. Ulmer, and J. F. Ehmke (2023). Heatmap-based decision support for repositioning in ride-sharing systems. *Transportation Science*.

Kaspi, M., T. Raviv, and M. W. Ulmer (2022). Directions for future research on urban mobility and city logistics. *Networks 79*(3), 253–263.

Kızıl, K. U. and B. Yıldız (2023). Public transport-based crowd-shipping with backup transfers. *Transportation Science 57*(1), 174–196.

Liu, S. and Z. Luo (2023). On-demand delivery from stores: Dynamic dispatching and routing with random demand. *Manufacturing & Service Operations Management 25*(2), 595–612.

Macrina, G., L. Di Puglia Pugliese, and F. Guerriero (2020). Crowd-shipping with time windows and transshipment nodes. *Computers & Operations Research 113*, 104806.

Macrina, G., L. Di Puglia Pugliese, F. Guerriero, and D. Laganà (2017). The vehicle routing problem with occasional drivers and time windows. In A. Sforza and C. Sterle (Eds.), *Optimization and Decision Science: Methodologies and Applications*, Cham, pp. 577–587. Springer International Publishing.

Mancini, S. and M. Gansterer (2022). Bundle generation for last-mile delivery with occasional drivers. *Omega 108*, 102582.

Mousavi, K., M. Bodur, and M. J. Roorda (2022). Stochastic last-mile delivery with crowd-shipping and mobile depots. *Transportation Science 56*(3), 612–630.

Powell, W. B. (2022). *Reinforcement Learning and Stochastic Optimization.* John Wiley & Sons, Ltd.

Savelsbergh, M. and M. W. Ulmer (2022). Challenges and opportunities in crowdsourced delivery planning and operations. *4OR 20*(1), 1–21.

Shen, C.-W., C.-C. Hsu, and K.-H. Tseng (2022). An auction-based multiagent simulation for the matching problem in dynamic vehicle routing problem with occasional drivers. *Journal of Advanced Transportation 115*, 2999162.

Silva, M., J. P. Pedroso, and A. Viana (2023). Deep reinforcement learning for stochastic last-mile delivery with crowdshipping. *EURO Journal on Transportation and Logistics 12*, 100105.

Soeffker, N., M. W. Ulmer, and D. C. Mattfeld (2022). Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research 298*(3), 801–820.

Tao, Y., H. Zhuo, and X. Lai (2023). The pickup and delivery problem with multiple depots and dynamic occasional drivers in crowdshipping delivery. *Computers & Industrial Engineering 182*, 109440.

Torres, F., M. Gendreau, and W. Rei (2022). Crowdshipping: An open VRP variant with stochastic destinations. *Transportation Research Part C: Emerging Technologies 140*, 103667.

Triki, C. (2021). Using combinatorial auctions for the procurement of occasional drivers in the freight transportation: A case-study. *Journal of Cleaner Production 304*, 127057.

Voigt, S. and H. Kuhn (2022). Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers. *Networks 79(3)*, 403–426.

Wang, L., M. Xu, and H. Qin (2023). Joint optimization of parcel allocation and crowd routing for crowdsourced last-mile delivery. *Transportation Research Part B: Methodological 171*, 111–135.

Yu, V., G. Aloina, P. Jodiawan, A. Gunawan, and T.-C. Huang (2023). The vehicle routing problem with simultaneous pickup and delivery and occasional drivers. *Expert System with Applications 214*, 119118.

Otto von Guericke University Magdeburg
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91/67-1 85 84
Fax: +49 (0) 3 91/67-1 21 20

www.fww.ovgu.de/femm

ISSN 1615-4274