9-2022

Choice-based crowdshipping: A dynamic task display problem

Alp ARSLAN

Firat KILCI

Shih-Fen CHENG
Singapore Management University, sfcheng@smu.edu.sg

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Choice-based Crowdshipping: A Dynamic Task Display Problem

Alp M. Arslan*

Lancaster University Management School

Firat Kilci

University of Texas at Austin

Shih-Fen Cheng, Archan Misra

Singapore Management University

Abstract

This paper studies the integration of the crowd workforce into a generic last-mile delivery setting in which a set of known delivery requests should be fulfilled at a minimum cost. In this setting, the crowd drivers are able to choose to perform a parcel delivery among the available and displayed requests. We specifically investigate the question: what tasks should be displayed to an individual driver, so as to minimize the overall delivery expenses? In contrast to past approaches, where drivers are either (a) given the choice of a single task chosen so as to optimize the platform's profit, or (b) allowed full autonomy in choosing from the entire set of available tasks. We propose a dynamic, customized display model, where the platform intelligently limits each driver's choice to only a subset of the available tasks. We formulate this problem as a finite-horizon Sequential Decision Problem, which captures (a) the individual driver's utility-driven task choice preferences, (b) the platform's total task fulfilment cost, consisting of both the payouts to the crowd-drivers as well as additional payouts to deliver the residual tasks. We devise a stochastic look-ahead strategy that tackles the curse dimensionality issues arising in action and state spaces and a non-linear (problem specifically concave) boundary condition. We demonstrate how this customized display model effectively balances the twin objectives of platform efficiency and driver autonomy. In particular, using computational experiments of representative situations, we exhibit that the dynamic and customize display strategy significantly reduces the platform's total task fulfilment cost.

Keywords: Crowdsourced Delivery, Drivers' Autonomy, Choice models, Last-mile Logistics

*Corresponding author:
Email address: a.arslan1@lancaster.ac.uk (Alp M. Arslan)

Choice-based Crowdshipping: A Dynamic Task Display Problem

Alp M. Arslan*

Lancaster University Management School

Firat Kilci

University of Texas at Austin

Shih-Fen Cheng, Archan Misra

Singapore Management University

Abstract

This paper studies the integration of the crowd workforce into a generic last-mile delivery setting in which a set of known delivery requests should be fulfilled at a minimum cost. In this setting, the crowd drivers are able to choose to perform a parcel delivery among the available and displayed requests. We specifically investigate the question: what tasks should be displayed to an individual driver, so as to minimize the overall delivery expenses? In contrast to past approaches, where drivers are either (a) given the choice of a single task chosen so as to optimize the platform's profit, or (b) allowed full autonomy in choosing from the entire set of available tasks. We propose a dynamic, customized display model, where the platform intelligently limits each driver's choice to only a subset of the available tasks. We formulate this problem as a finite-horizon Sequential Decision Problem, which captures (a) the individual driver's utility-driven task choice preferences, (b) the platform's total task fulfilment cost, consisting of both the payouts to the crowd-drivers as well as additional payouts to deliver the residual tasks. We devise a stochastic look-ahead strategy that tackles the curse dimensionality issues arising in action and state spaces and a non-linear (problem specifically concave) boundary condition. We demonstrate how this customized display model effectively balances the twin objectives of platform efficiency and driver autonomy. In particular, using computational experiments of representative situations, we exhibit that the dynamic and customize display strategy significantly reduces the platform's total task fulfilment cost.

Keywords: Crowdsourced Delivery, Drivers' Autonomy, Choice models, Last-mile Logistics

*Corresponding author:
Email address: a.arslan1@lancaster.ac.uk (Alp M. Arslan)

Preprint submitted to Transportation Research Part B: Methodological *September 12, 2022*

1. Introduction

The explosive growth of e-commerce (Young, 2019) has resulted in substantial demands for field operation workforce. While technologies such as drones, droids, or autonomous vehicles may well underpin the delivery infrastructure in the future, many e-tailers are presently adopting a "crowd workforce" model. In crowdshipping, the last-mile delivery tasks are delegated, with the help of online platforms, to a time-varying pool of willing individuals. Sampaio et al. (2019) show that crowdshipping has the potential to reduce the overall delivery cost by lowering the barriers for individual "workers" to voluntarily utilize their own resources (time and vehicles) to deliver such packages. Such a crowdsourced workforce also provides a more elastic labor supply that can efficiently respond to demand variations (e.g., during holiday season peaks) (Einav et al., 2016). As many crowd drivers choose to perform delivery tasks that align well with their regular commuting journeys, they may be willing to perform delivery tasks for rates that are lower than the dedicated workers.

The role of the online platform is critical to the efficiency and viability of this crowd workforce model, as its task assignment mechanisms directly affect the participation rate of the crowd workforce (Chen et al., 2014; Kandappu et al., 2016). Very broadly speaking, the task assignment mechanism can either be centralized (where the platform *assigns* tasks to available individual workers) or decentralized (where individual workers *select* available tasks independently from a pool). The two strategies illustrate a broad tension between *efficiency and autonomy.*

In practice, the choice of the task assignment mechanism depends heavily on the nature of the tasks. For domains such as on-demand transportation or meal delivery, where tasks have short expiration times, most online platforms (e.g., Uber, GrubHub) centralize task assignment, pairing drivers and delivery requests quickly without elaborate driver consultation. In such scenarios, because assignments are driven by the worker's current (instantaneous) location, the chance of worker rejection of centralized assignments is lower. The situation is, however, reasonably distinct for **overnight logistics tasks** (a dominant fraction of the e-commerce market, Joerss et al. 2016), where delivery tasks are known in advance, and the task-worker allocation usually happens over a longer time window (e.g., tasks are posted by midnight, and workers have until 8 am to select tasks). In such a scenario, the decentralized assignment mechanism could be more appealing as it allows workers to choose preferred tasks based on their anticipated itinerary.

This paper focuses on an overnight last-mile delivery platform and explores how to design the decentralized task assignment mechanism so that it possesses centralized-mechanism-like performance. Our key idea is to have a mechanism capable of presenting a carefully curated subset of available tasks (e.g., displayed on a worker's mobile App) to crowd drivers, with the driver retaining the autonomy of accepting a task from this subset (or optionally, declining any of the displayed tasks). We hope to combine the best features of centralized and decentralized mechanisms.

We refer to this model of 'centralized customization, autonomous selection' as *choice-based crowdshipping*. The selection of tasks displayed in such choice-based crowdshipping involves two competing objectives. First, the platform should maximize the likelihood that a driver selects a task—this likelihood is maximized if the platform displays all tasks. Second, the platform should perform display customization to minimize its operation cost (e.g., to *fully* serve as many regions as possible with crowd workers, so that most remaining tasks not selected are clustered around the same neighborhood and can thus be completed by fewer dedicated platform employees than without crowdshipping). The central theme of this paper is thus to develop and quantify the significance of such display policies, which strategically incorporate both the choice behaviors of individual drivers and the platform's cost-minimizing objective.

To analyze choice-based crowdshipping in last-mile logistics, we introduce the Dynamic Task Display Problem (DTDP). In DTDP, (a) there is a finite duration (Selection Period) over which individual drivers arrive randomly and request the platform/App to display tasks from which to make a selection, and (b) the platform dynamically determines the subset of displayed tasks for each individually arriving driver. We make the following key contributions:

- We develop a model for selective customization of tasks displayed to a specific driver that takes into account (i) the platform's objectives in minimizing the total cost of delivery, including the additional fleet & driver costs associated with fulfilling the tasks not selected by crowd drivers; (ii) an individual driver's likelihood to choose a specific task from a given displayed list, presumed as and computed via a multinomial logistic choice model; and (iii) the statistical arrival process of workers, with different spatial/route preferences, within the selection period.

- Given that an exact solution to the DTDP is computationally intractable for large instances, we propose a stochastic look-ahead strategy constructed on two main pillars: (i) Value Function Approximation and (ii) Efficient Display Sets. With this approximation strategy, we not only tackle the curse of dimensionality issue but also propose an effective way to solve sequential decision problems with non-linear boundary conditions.

- We also numerically study the effectiveness of the proposed display strategies under varying operating conditions, particularly the cases where optimal display policy differs from either the full display or single display policies. We demonstrate that the customized display strategy decreases the total fulfillment cost of the platform by shortlisting tasks more ideal to drivers and the platform, even though the number of tasks selected by crowd drivers is lower than the decentralized full display policy.

In summary, our core contribution is to introduce the concept of *customized task display* as a mechanism to induce desirable (cost-reducing) crowd driver behaviors in crowdsourcing-based overnight last-mile logistics operations. The remainder of this paper is organized as follows. In Section 2, we summarize the relevant literature for the choice-based crowdshipping problem. Section 3 begins by describing the conceptual business model for overnight crowdshipping with a motivating example. In the sequel, the choice-based crowdshipping problem is formulated as a Sequential Decision Problem, and the choice behavior of drivers is modeled as an attraction model. In Section 4, we present the solution method for the optimal display policy. Sections 5 and 6 consist of computational experiments to quantify the effectiveness of our proposed curated display policy. Section 7 closes the paper with concluding remarks and some directions for future research.

2. Related Literature

In a broad classification, dynamic task display problem (DTDP) is associated with two research streams: crowd workforce in transportation and logistics and assortment and display optimization models. The former stream aims to take advantage of the under-utilized resources in passenger and/or parcel transportation (Wang and Yang, 2019; Le et al., 2019; Alnaggar et al., 2019; Savelsbergh and Ulmer, 2022). Assortment, or display optimization models, however, leverage human choice behaviors when (mainly) a customer faces multiple options at a moment of decision (Kök et al., 2008). In this review, we present the features of the DTDP that are linked to these two research areas.

2.1. Crowd workforce in transportation and logistics

The DTDP is essentially a crowdshipping problem as its primary purpose is to efficiently integrate willing crowd drivers to serve a set of transportation requests. Initial studies in the crowdshipping literature have explored the benefits of accommodating crowd drivers into existing delivery fleets, assuming that these drivers accept any task as long as it is feasible in their stated preferences. Archetti et al. (2016) and Arslan et al. (2019) contemplate platforms giving centralized decisions to routing, assignment, and scheduling problems in a setting in which drivers' journey and time flexibility are informed in advance or dynamically, respectively. In a similar vein, Kafle et al. (2017) considers a centralized job assignment model for crowdshipping where delivery of parcels is done via the synchronization of trucks and crowd drivers in a relay-like mechanism. Dayarian and Savelsbergh (2020) generalizes this stream of research by integrating future information into the planning. A group of related studies, Rai et al. (2018), Paloheimo et al. (2016), Simoni et al. (2019), and Mancini and Gansterer (2022) investigate the impact of crowdshipping on sustainability and environmental considerations under similar centrally modeled business model presumptions.

After a rapid success of crowdsourced based transportation and logistics services, a stream of research explores tactical strategies for self-scheduled drivers (Yildiz and Savelsbergh, 2019; Ulmer and Savelsbergh, 2020; Gurvich et al., 2019; Dai and Liu, 2020). In contrast to crowd drivers, self-scheduled drivers form a semi-independent workforce, where they work according to their own schedules by performing tasks without choice autonomy. Therefore, these models primarily examine decisions like setting compensation schemes to coordinate the tension between transportation demand and driver supply.

The efficiency of crowd-based transportation platforms depends on keeping the correct number of drivers available. Therefore, understanding the determinants of crowd driver behaviors that attract or lay off them under different circumstances is crucial (Taylor, 2018). Empirical and experimental studies have shown that the drivers' willingness to work depends primarily on the detour distance, the amount of compensation paid for task completion, and demographic factors such as age (Miller et al., 2017; Le and Ukkusuri, 2018). In particular to the choice autonomy, past works (e.g., Chen et al. (2014) and Kandappu et al. (2016)) have shown that centralized task assignment allows the platform to explicitly maximize the chosen performance metrics (e.g., overall task completion rate or worker efficiency). However, as centralized planning is one-sided, some drivers may find the centrally allocated assignment undesirable and choose not to accept them. In the long run, this could lead to significant driver attrition. On the other hand, the decentralized paradigm allows individuals to have complete control over their choices and thus usually fosters higher worker satisfaction. Yet studies suggest that in such a paradigm, a small percentage of committed drivers (the super agents) could dominate the task pool, choosing the most desirable tasks when they become available immediately, thus leading to the marginalization of less active drivers (Musthag and Ganesan, 2013). This again would lead to the deterioration of the pool of crowd workers and less efficient workforce utilization.

Centralized task assignment mechanisms are widely used for real-time operations with a few exceptions. As a result, there is limited attention to customized task display/assignment mechanisms in the operations research literature. However, we are aware of the papers by Mofidi and Pazour (2019), Horner et al. (2021), and Ausseil et al. (2022) studying the concept that platforms offer a menu of task options to drivers. Mofidi and Pazour (2019) explore a bi-level optimization problem in the case a driver chooses a task from the menu. However, the driver's task selection behavior is considered to be deterministic. That is, the platform is assumed to know drivers' preference rankings among the revealed tasks and considers that the driver will always choose the top one. In the study by Horner et al. (2021), authors explicitly consider the driver's task choice autonomy by introducing a new mechanism. Following Stackelberg game principles, the platform first displays a curated menu for each driver in a batch. Then each driver responds to the menu by revealing which task she would like to perform (with the possibility of rejecting all). At the last stage, the platform matches drivers to their preferred tasks with the objective of maximizing its own utility. Finally, Ausseil et al. (2022) enrich the menu of task options by analyzing the potential conflicts that may arise due to the same task being offered to different drivers.

In contrast to these past studies, we are the first ones to study crowdshipping in the overnight delivery setting. The overnight delivery problem differs substantially from the real-time task assignment problems, especially in the way the unassigned/not selected orders/tasks are handled at the end of the selection period. Also, we explicitly model the stochastic driver choice model and incorporate the discrete choice model in our formulation with the possibility of rejecting the delivery.

2.2. Assortment and display optimization

The DTDP is closely related to assortment/display optimization problems as the platform's only tool to influence drivers' task selections is to decide which tasks to display. When the platform offers options, drivers' task-selecting behavior should be incorporated into the decision process. In our problem formulation, we use a discrete choice model to reflect drivers' decisions as a function of a set of displayed tasks and their relative preferences to each other. For the assortment optimization problem where customers face multiple products while browsing for their needs, such as flight and accommodation searches, Multinomial Logit models (MNL) are dominantly used (e.g., see Talluri and Van Ryzin (2004) and Gallego et al. (2018)). In the most general form, these problems aim to determine what product group to offer to each customer segment to maximize the overall expected revenue of the seller. In a dynamic setting, which is the most similar setting to our problem, these product groups are determined considering finite inventory and stochastic customer arrivals (see Bernstein et al. 2015). On the other hand, this study explores and generalizes the analysis to the case where the leftover (unsold) inventory has a nonlinear valuation depending on the volume.

Display problems have also received attention lately in the last-mile logistic domain with the attended home delivery services. In these services, customers choose desired time windows to receive their parcels among the displayed time windows. Yang et al. (2016) introduce the choice-based demand management models by using time-window displays and fees as tools to minimize the operational cost of home delivery. Along with the popularity of time-slot management, a similar approach is implemented in different areas such as repair jobs, patient home visits, or steering consumers toward green choices (Ulmer and Thomas, 2019; Demirbilek et al., 2019; Agatz et al., 2021).

In conclusion, to our knowledge, this paper is one of the first to consider explicit stochastic task selection in the crowdshipping domain. Even though some features of the problem are individually studied in the different literature streams, this study distinguishes itself from them by formulating drivers' task selection as an attraction model. This modeling allows us to explore the various no-choice behaviors and their impact on the platform's delivery expenses. In addition and more importantly, with the proposed model, we quantify the benefits of dynamic customization of display tasks over the fully centralized and decentralized job allocation mechanisms.

3. The Dynamic Task Display Problem

In this section, we describe the Dynamic Task Display Problem (DTDP). We first provide an overview on exactly how our proposed paradigm of choice-based crowdshipping would work in practice. After the problem definition, we provide a Markov decision process (MDP) formulation for the DTDP. The objective of the DTDP is to minimize the platform's total cost of order fulfillment, which is the sum of the total reward paid to crowd drivers and the expenses for engaging contract drivers (who are hired to deliver packages not selected by crowd drivers).

3.1. Choice-based crowdshipping system architecture

Our research problem is motivated by a real-world overnight package delivery platform that utilizes both crowdsourced and contract drivers. Consider a platform engaging crowd drivers by asking them to browse and select tasks on a smartphone App daily. Such browsing and selection are enabled during a *Selection Period*, typically spanning several hours in the evening, before the commitment deadline for the deliveries to be performed the next day. During this selection period, individual crowd drivers arrive at the platform randomly. When each driver arrives, the platform decides which unassigned tasks should be displayed. Manipulating the set of displayed tasks allows the platform to affect the collective selection behavior of crowd drivers.

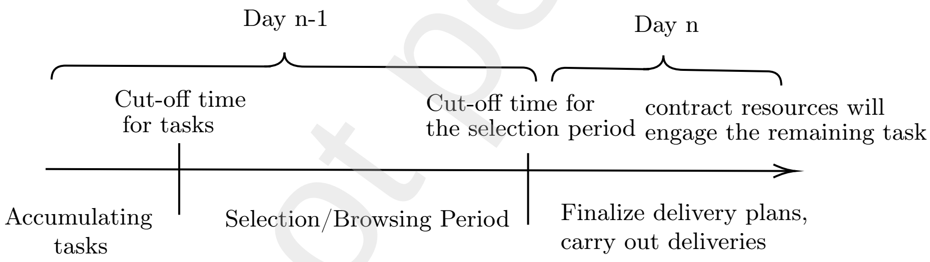

Figure 1: Operational workflow for an overnight delivery platform that utilizes both crowd and contract drivers.

Figure 1 visualizes the operational workflow of this platform. All tasks to be delivered in day n should already arrive in day $n-1$ before the first cut-off time, after which the platform decides which tasks should be displayed to an individual crowd driver whenever such a driver interacts with the platform. The platform's decision should depend on the remaining tasks and crowd drivers' self-declared preferences. After the termination of the selection period, the platform will separately determine how many (hours of) contract drivers to hire to deliver the remaining tasks. From the platform's perspective, the objective is to *minimize the total cost required to engage both crowd and the contract drivers*.

Since it is expensive to engage contract drivers, the platform would naturally aim to reduce the needs for such drivers. Two major factors affect the number of required contract drivers: 1) the number of remaining tasks and 2) the spatial distribution of remaining tasks.

Therefore, if possible, the platform should aim to not only reduce the number of residual, unselected tasks, but also avoid having these tasks scattered across a large number of zones. Since crowd drivers perform their selections in an uncoordinated fashion, the distribution of these residual tasks is thus outside the platform's direct control and dependent on the actions of the crowd drivers. However, the platform can *indirectly* control a crowd driver's task selection by curating a personalized list of tasks displayed to an individual driver to suit her self-reported spatial preferences.

An illustrative example in Figure 2 demonstrates how a customized task display can help to reduce the platform's fulfillment cost. Let the incoming driver be the last driver to come in before the cut-off time. Assume that this driver has a capacity of 1 and prefers Zone 1 over Zone 2. Assume that there are three remaining tasks in Zone 1 and one remaining task in Zone 2. We can choose one of the three display options for this driver: D_1) show tasks in both zones, D_2) shows tasks in Zone 1, and D_3) show the task in Zone 2.

Let $P_i(D_j)$ be the probability that the driver would choose a task from Zone i, given a display set D_j. If the driver decides not to serve, set $i = 0$. For example, for D_1, the probability that the driver will choose one task from Zones 1 and 2 are 0.6 and 0.3, respectively, while the probability of choosing nothing is 0.1.

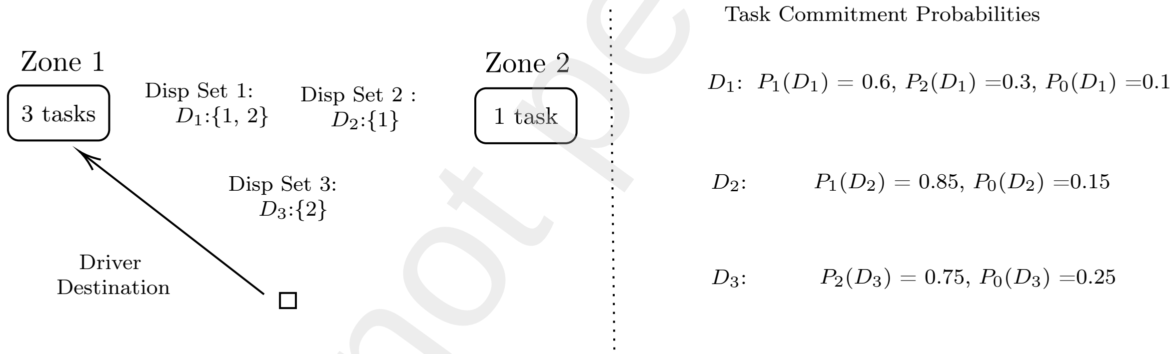

Figure 2: An example of personalized display set.

The platform aims to minimize the total cost, which includes the cost of engaging contract drivers. For this example, we assume that the fixed cost of engaging a contract driver is \$100 for any zone with residual, unselected tasks, and the variable cost for serving each additional task in the same zone is \$10/task. The expected costs of contract workers for the three display options are computed below:

- *Display Set 1*: If the driver picks a task from Zone 1, there are 2 and 1 remaining tasks in Zones 1 and 2 respectively, resulting in a cost of $(100+2\cdot10)+(100+10) = 230$. Similarly, if the driver picks a task from Zone 2, the resulting cost will be 130. Finally, if the driver chooses not to serve, the resulting cost will be 240. The expected cost is thus: $0.6\cdot230+0.3\cdot130+0.1\cdot240 =$

201.

- *Display Set 2*: The expected cost is: $0.85 \cdot 230 + 0.15 \cdot 240 = 231.5$.
- *Display Set 3*: The expected cost is: $0.7 \cdot 130 + 0.3 \cdot 240 = 157.5$.

In the above example, the use of display set D_3 (i.e., showing the crowd driver tasks only from Zone 2) results in the lowest expected cost. This is even though the driver strongly prefers Zone 1 and thus has a significantly higher probability of not choosing any task in display set D_3. This is caused by the high fixed cost of engaging a contract driver. Consequently, whenever possible, the platform would desire to adopt a display set that helps to reduce the number of zones with residual, unselected tasks.

Of course, this is an overly simplified example designed only to demonstrate the benefits of display set customization. The complete model should capture the complexity of having more zones and heterogeneous driver preferences. Moreover, we also have to consider the sequential nature of the decision-making process; i.e., drivers make their selection asynchronously and sequentially, and thus earlier display decisions impact the pool of tasks available for selection by future drivers. Lastly, our model should also explicitly incorporate how drivers choose tasks, given their preferences and the display set.

The subsidiary benefit of selective display policies is to quantify their impact under varying boundary conditions , i.e., the cost regime for the contract backup drivers. The motivating example contemplates a specific dedicated cost function as it considers the spatial costs of all unassigned tasks and the total number. To examine different cost models, we draft a generic cost function in the following model capable of addressing the spatial feature of the unassigned tasks.

3.2. DTDP as a Markov decision process

The challenge of DTDP is the uncertain nature of drivers' choices, which can be affected by the set of tasks displayed to them. This implies that what the platform displayed to the earlier drivers might affect the tasks left in the later stage. However, these impacts of earlier display sets can be summarized by X_t, the set of remaining tasks at period t. This means that the planning problem can be made Markovian if we include X_t as part of the state variables.

Based on this insight, we formulate the DTDP as a Markov decision process (MDP) with the following crucial MDP components. For easier reference, we summarize the notations used in our formulation in Table 1.

3.2.1. Selection horizon and decision epochs

Crowd drivers arrive within the selection horizon of $[0, T]$. The selection horizon is discretized in equally-sized time intervals denoted by set $\mathcal{T} = \{0, 1, \cdots, T\}$, in which the likelihood of more than

Table 1: Notations for DTDP.

Notation	Description
$\mathcal{T} = \{1, 2, \ldots, T\}$	Periods/Slots
Z	Set of zones/clusters
$tt(i, j)$	Movement cost between zone z_1 and z_2
X_t	State at time t
λ_{tk}	Arrival rate of driver designated to zone k at time t
r_z	Reward for serving a task at zone z
$d_{t,k}$	Decision vector for drivers designated to k at period t
z	Index shows delivery task location in zone z
k	Index shows a driver's destination to zone k
π	Decision policy
$C_z(x)$	Cost of having x tasks remaining at the end of the browsing period in zone z
α	Normalizing parameter for the choice model
ef_z	Driver's effort of making delivery in zone z

one driver arriving in a single period is negligible. Each driver-arriving epoch $t \in \mathcal{T}$ is a decision epoch.

3.2.2. States

The state of the system at epoch t is represented by a vector of $X_t = [x_1, \cdots, x_n]$, in which $x_i,\ i \in Z$, represents the number of unselected tasks in zone i. Set Z contains all geographical zones that are mutually exclusive.

3.2.3. Actions and costs

Assume that at epoch t a driver who prefers zone $k \in Z$ arrives (we call them type-k drivers). From the set of zones that still have remaining tasks, the decision to be made by the platform is which zones to display to this driver (we assume that from each chosen zone, a task from that zone will be chosen at random). We denote this decision as a vector $D_t = [d_1, \cdots, d_n]$, where d_i is a binary variable indicating whether zone i is shown to the driver. Upon seeing this display set, we assume that the driver will choose a task from one of the displayed zones; more specifically, let p_i be the probability of choosing from zone i, and $r_i > 0$ be the reward associated with this choice. The driver may also choose to select no task and receive no reward, which is with probability p_0 (the probabilistic choice model is described later in Section 3.2.4).

Let $R_t(D)$ be the expected reward paid to a driver with a chosen display vector D. At the end of the selection horizon, or epoch $0 \in \mathcal{T}$, the set of unselected tasks, X_0, will incur additional cost to the platform according to the function $C : X_0 \mapsto \mathbb{R}_+$. In other words, the boundary condition of our MDP is $V_0(X_0) = C(X_0)$. We assume that $C(X_o)$ has the following properties:

- $C(X_0) = \sum_{z \in \mathbb{Z}} c(x_{z0})$.
- $c(x_{z0}) = 0$, if $x_{zo} = 0$.
- $c(x_{z0})$ is increasing and concave.

With the above definition, we implicitly assume that each zone with remaining unselected tasks needs to be served by one independent contract driver.

3.2.4. Transitions

At each epoch $t \in \mathcal{T}$, let λ_{tk} be the arrival rate for the type-k drivers. The state at the end of the period changes with the displayed zones to the driver and, consequently driver's choice. We use a classical approach in handling how drivers choose among multiple alternatives, given a well-defined utility function taking the distance of the driver's destination and a task's location into account.

Formally speaking, for a given display set d_{tk} that is shown to a type-k driver, we denote the probabilities of choosing nothing and a task from zone z as $P_k(0|d_{tk})$ and $P_k(z|d_{tk})$ respectively. We assume that a driver's zone choice probability depends only on the set of zones displayed and not on the number of tasks shown for individual zones (the driver should be indifferent to tasks within the same zone as all tasks in the same zone are equivalent and the driver is limited to selecting only one task).

A classical approach in handling how individuals choose among multiple alternatives given a well-defined utility function is the Multinomial Logistic (MNL) model (see Gallego et al. (2018)). With the MNL model, a driver would choose a zone probabilistically according to a monotonic function depending on their utility function. Let $u_k(z)$ be the utility of choosing zone z for the type-k drivers. According to the framework by Ben-Akiva and Bierlaire (1999), the choice probability is defined as:

$$\begin{aligned} P_k(z|d_{tk}) &= \frac{e^{\alpha u_k(z)}}{e^{\alpha u_k(0)} + \sum_{l \in d_{tk}} e^{\alpha u_k(l)}}, \\ P_k(0|d_{tk}) &= \frac{e^{\alpha u_k(0)}}{e^{\alpha u_k(0)} + \sum_{l \in d_{tk}} e^{\alpha u_k(l)}}. \end{aligned} \tag{1}$$

The implication of defining the choice probability using (1) is that the likelihood of not serving increases when the display set size decreases. This is consistent with our modeling assumption and the empirical studies mentioned in the introduction.

The utility function $u_k(z)$ can be calculated as:

$$u_k(z) = r_z - ef_z - tt(z, k), \tag{2}$$

where r_z and ef_z refer to the reward and effort of serving a task in z (independent of type k), and

$tt(z,k)$ is the movement cost for a type-k driver to move from zone z (where the task is) to her/his preferred zone k.

In practice, it is not realistic to know or estimate the true utility of a driver. The probabilistic nature of the MNL model allows us to make up for the inaccuracy in the estimation due to tangible and intangible cost components not included in (2). By setting the value of the parameter α, we can further control how drivers would choose their actions. When $\alpha \to 0$, drivers would just randomly choose actions (utility values play no role in driver's choice). When $\alpha \to \infty$, drivers would choose the action with highest utility value (this setting implies that (2) captures the true utility).

3.2.5. Objective function

In the DTDP, we seek a Markovian display policy Π that minimizes the total expect delivery cost for the platform:

$$\min_{\Pi} \mathbb{E}\big[\sum_{t\in\mathcal{T}} R_t^{\Pi}(D) + C(X_0)\big].$$

3.2.6. Optimality equation

Let the value function $V_t(X_t)$ denote the expected cost at period t to terminal period 0, when X_t is the set of unassigned delivery requests. The recursive value function is defined as:

$$V_t(X_t) = \sum_{k\in K} \lambda_k \left[\min_{d_{tk}\subset \bar{D}_t} \left\{ \sum_{z\in d_{tk}} P_k(z|d_{tk}) \Big(r_z + V_{t-1}(X_t - e_z) \Big) + P_k(0|d_{tk}) V_{t-1}(X_t) \right\} \right], \quad (3)$$

where e_z is a unit vector where z^{th} element equals 1, and $\bar{D}_t$ is the set containing all zones that have positive task counts in time t. The value function $V_t(X_t)$ has two sources of uncertainty: (i) the choice made by the current driver k, and (ii) the type of the next arriving driver. For (i), it is captured by $P_k(z|d_t k)$ and $P_k(0|d_t k)$ above, which indicate whether a task from zone z is chosen, or the driver chooses not to perform any task. For (ii), it is handled by accounting for λ_k over all possible types $\mathbb{Z}$.

4. Solution Approach

In this section, we provide details of the approximation approach for finding solution to our formulated DTDP instances.

4.1. Motivation and overview of the stochastic look-ahead approach

Solving the DTDP, formulated as a finite-horizon MDP in Section 3, via backward fashion is not possible when the number of zones/clusters exceeds trivially small values (less than 9 in practice). The backward induction approach is unfortunately plagued by the exponential explosion of state and action space in the number of zones/clusters. In addition, the DTDP involves two additional novel challenges: (i) uncertainty in drivers' type, arrival times, and their task selection behavior during the task selection phase, and (ii) the non-linear cost structure at the terminal period when computing the fulfillment cost of unselected tasks.

To address these challenges, we propose a tractable and forward-looking stochastic look-ahead method that balances the computational efficiency with respect to the problem size and the solution quality. The look-ahead method has two essential pillars: (i) *Value Function Approximation* (VFA) to overcome the state space explosion and (ii) *Efficient Display Sets* to reduce the action space. The role of these two pillars can be seen with the reorganization of the optimality equation as follows:

$$D_t^\pi(X_t, k) = \underset{d \in \mathcal{D}_t}{\arg\min} \Big(R(X_t, d) + \mathbb{E}\{\bar{V}_t(X_t^d) | X_t, k\} \Big), \tag{4}$$

where X_t^d is the post-decision state right after the driver k is given the display set d but before the next driver arrives. The VFA eliminates the need to pre-compute the values of each state. In other words, only states that are encountered throughout the execution will be approximated (this is similar to the reaching algorithm in dynamic programming). The impact of the Efficient Display Sets is the elimination of some display subsets intelligently without compromising the solution quality.

The overall architecture of our solution approach is illustrated in Figure 3. At each driver arrival epoch t, our method executes two steps. In the first step, the associated value function for all possible states that can be reached from the current state X_t is approximated. At the second step, the efficient display sets are formed with respect to the approximated value functions. We provide details of these two steps in the remainder of this section.

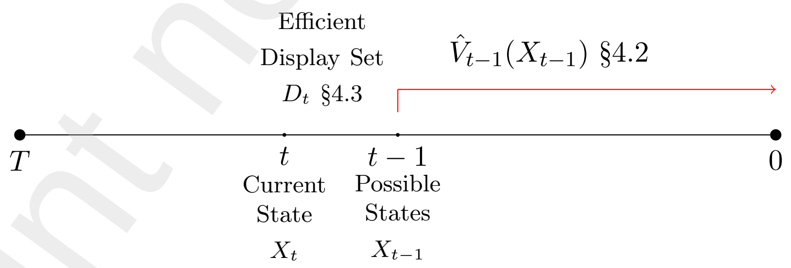

Figure 3: Solution approximation architecture: VFA

4.2. Value function approximation (VFA)

In this section, we describe how to approximate the value function for a given state, which involves estimating the total fulfillment cost for a given number of tasks and the number of periods to go until the end of the selection period. The approximation is broken down into two phases:

First, we define a set of simple yet practical display rules. Afterward, we use the probabilistic information of future driver arrivals and the structure of the chosen display policy to approximate the expected remaining tasks at the terminal decision epoch.

We start by introducing a generic state- and time-independent display policy: *a single-task display following predetermined zone sequence (SDPS)*. In this policy, each arriving driver is shown a single task from the highest-ranked zone that is still with unselected tasks. When this zone is fully served, the next zone is chosen following a predetermined sequence, which can be as simple as using zone IDs (i.e., Zones 1, 2, ..., $|Z|$).

The idea of the VFA is to employ the described the SDPS policy from period $t-1$ until the end of the selection period; i.e., period 0, and estimate the number of remaining tasks that has to be fulfilled by the dedicated workforce under the assumption that the sequence of zones is predetermined. As the boundary condition of the DTDP at the end of the selection period is not linear in the total remaining tasks, it is crucial to estimate how many tasks will remain unselected for each zone to achieve a good approximation. In the following, we explain how to compute the expected boundary condition. For the ease of the exposition, we first introduce a few properties.

Proposition 1. *When a display set consists of a single zone $z \in Z$ in time t, we define the probability that a task is chosen as: $P_{zt} = \sum_{k \in \mathbb{Z}} \lambda_{kt} p_{kz}$, where p_{kz} denotes the probability that a type-k driver chooses a task in zone z when only zone z is displayed.*

Proposition 1 shows that for each time epoch in the period of $\tau \in \{t, \cdots, 0\}$, the probability that a zone-z task will be chosen follows the Bernoulli distribution with parameter P_{zt}.

Proposition 2. *Consider a SDPS policy where the ranking order of the zones to be displayed is predetermined. Let Y denote the total number of tasks and $[y]$ denote the index of the zone that is displayed after $y-1$ tasks have been committed, and let $P(y,t)$ denote the probability of having a total of y tasks chosen at the end of period t. Then, for each $0 < y \leq t$, $P(y,t)$ can be calculated using the following recursive equation:*

$$P(y,t) = P(y-1,t-1)P_{[y]t} + P(y,t-1)(1-P_{[y]t}) \tag{5}$$

With the following boundary conditions:

$$P(0,t) = \prod_{n=1}^{t}(1-P_{[1]n}) \quad \forall t = 1..T \tag{6}$$

$$P(y,t) = 0 \quad \forall y > t \tag{7}$$

$$P(1,1) = P_{[1]1} \tag{8}$$

Then, let $X^{[y]}$ be the vector that denotes the number of tasks in each zone after y tasks have been committed. Once the above values of $P(y,t)$ are calculated using Proposition 2, the expected total cost of operating under the SDPS policy with a predetermined display order for T periods given an initial inventory vector X_T and number of total tasks Y can be obtained as follows:

$$V_T^{SDPS}(X_T) = \sum_{n=1}^{\min(Y,T)} P(n,T)\left(C(X^{[n]}) + \sum_{z=1}^{n_Z} r_z \cdot \left(X_T(z) - X^{[n]}(z)\right)\right) \tag{9}$$

4.3. Efficient display sets (EDS)

We further employ the efficient display sets following the Theorem 1 in Talluri and Van Ryzin (2004) and Lemma 1 in Bernstein et al. (2015) as the following proposition to reduce the action space significantly. More precisely, the efficient display sets decrease the size of the action space from 2^{n_Z} to n_Z.

Efficient display sets are formed systemically by calculating the marginal contribution of displaying a single zone. To form efficient sets, we first define the expected cost generated by the x_z^{th} task in zone z at period t, i.e., $z \in \bar{D}_t$, define $\triangle_{t-1}^z(X_t) = V_{t-1}(X_t) - V_{t-1}(X_t - e_z)$. If we rewrite the optimality equation in (3) as:

$$V_t(X_t) = \min_{d \subset \bar{D}_t} \left\{ \sum_{z \in d} \lambda_z P_k(z|d)\left(r_z - \triangle_{t-1}^z(X_t)\right)\right\} + V_{t-1}(X_t). \tag{10}$$

If a task from z is displayed and selected in period t, the expected total cost changes by the reward given to the driver minus marginal penalty reduction in zone z. Let $r_t^z(X_t,k) = r_z - \triangle_t^z(X_t,k)$ denote the effective marginal cost of task in zone z in period t when driver k appears and $z \in \bar{D}_t$. Consider an ordering of zones so that $r_t^{z_1}(X_t,k) \leq r_t^{z_{n_Z}}(X_t,k)$ and let a set consisting of the zones with marginal effective penalty given by $A_l(X) = \{z_1, \cdots, z_l\}$.

It is important to note that displaying solely efficient display sets introduced in the study by Talluri and Van Ryzin (2004) is a dominant action when the boundary condition is in the form of a linear function. Nevertheless, the DTDP generalizes the boundary conditions; therefore, the dominance of the efficient display sets does not hold.

5. Experimental Setup and Solution Approach Validation

In this section, we describe the settings employed in our computational study. We first introduce benchmark display policies and key performance indicators. We then validate our SDPS approach using a set of small-enough instances where exact solutions are attainable. Lastly, we quantify the impact of a dynamic and customized display policy through a real-life inspired scenario that is based on open data in Singapore (utilizing both the map topology and the population distribution).

5.1. Benchmark policies

We present two benchmark policies to compare the performance of the customized display (CD) policy in Section 4.

- **Full display (FD).** In this policy, the platform display tasks from all zones that still have tasks remaining. This policy can be considered as a fully decentralized display mechanism. Given the remaining time epochs and the number of tasks in each zone, the value function of this policy is approximated through simulations.

- **Single display (SD).** Given the earlier defined cost structure, an intuitive policy is to focus on finishing tasks from a zone before moving on to other zones. In practice, the SD policy selects the zone with fewest remaining tasks, breaking ties by zone index (choosing lower index first). The SD policy can thus be thought of as a special case of the SDPS policy. Given the remaining time epochs and the number of tasks in each zone, the value function of the SD policy is calculated analytically using Equations (6), (7), (8), and (9).

5.2. Key performance indicators

In this section, we define four key performance indicators (KPIs) used in evaluating competing policies.

- *Fulfillment cost (FC).* This is the objective function value we defined for our MDP formulation. It summarizes the costs of paying all crowd drivers and the contract drivers at the end of the selection period.

- *Matched requests (MR).* The ratio of requests served by the crowd drivers.

- *Reward ratio (RR).* The proportion of the total reward paid to the crowd drivers compared to the total fulfillment cost.

- *Drivers' utility (DU).* As the true utility of a driver is unknown to the platform, we derive this metric by defining the maximum anticipated utility that an arriving driver is seeing. In this metric, we quantify the impact of displaying more or few task types to drivers.

5.3. Solution approach validation

The stochastic look-ahead (LA) proposed in Section 4 consists of both state and action spaces reduction techniques (termed VFA and EDS respectively). To understand the goodness of these approximation ideas, we need instances that are small enough for the exact solution method.

In this section, we consider 50 randomly generated instances consisting of eight zones located along the perimeter of a circle, which are denoted as the set Ins. For each instance, 20 tasks are distributed randomly to eight zones with equal probability, implying a finite (albeit small) probability of one or more zones having no delivery tasks. We set the length of the selection horizon equal to the number of tasks, which ensures that there will always be at least one zone to display to a driver. In all instances, the eight zones are identical. The driver's reward for serving a zone is $r_z = \$30$, and the cost of having x tasks remaining at the end of the planning horizon (i.e., the payout for employing contract drivers) is equal to $c(x) = 150\sqrt{x}$.

We assume that drivers' destinations define their types. Without loss of generality, we assume that type-k driver resides in zone k. Therefore completing a task in zone k is the most preferred option. Furthermore, we assume that no additional effort is needed to complete a task. In that sense, we calculate driver k's utility for serving a task in zone z as $u_k(z) = r_z - tt(z,k)$ and the no-choice utility is $u^0 = 0$. Each driver's arrival or a non-arrival is equally likely, hence $\lambda_z = \lambda_0 = 1/9$. All parameters used in our validation are summarized in Table 2.

Table 2: Parameters used in the validation of our SDPS approach.

Selection horizon	T	20
Number of zones	n_Z	8
Number of delivery tasks	n_P	20
Probability of no-arrival	λ_0	1/9
No-choice utility	u^0	1
Reward	$r_z = r, z \in Z$	30
Cost of non-completed tasks	$c_z(x), z \in Z$	$a_z\sqrt{x}$
Cost function coefficient	$a_z, z \in Z$	150
MNL normalization parameter	α	0.1

To quantify the effectiveness and the efficiency of the look-ahead method, we calculate the gap in the total fulfillment cost and the run time using the equations displayed below.

$$Ave.Gap = \frac{\sum_{i \in Ins} FC_i(Approximation)}{FC_i(Exact)} - 1$$
$$MaxGap = \max_{i \in Ins} \frac{FC_i(Approximation)}{FC_i(Exact)} - 1$$
$$TimeSaved = 1 - \frac{CompTime(Approximation)}{CompTime(Exact)}$$

Table 3 presents the average, maximum cost gaps, and proportional time saved when one of the following approximation methods is used to compute the total fulfillment cost: (i) EDS: uses only the efficient display sets, (ii) VFA: uses only the value function approximation, and (iii) LA: uses VFA and EDS together. For completeness, we also present the performance metrics from the full display (FD) and the single display (SD) policies.

As seen from Table 3, using EDS reduces the run time by $\sim 90\%$, and the differences in the

Table 3: Average and maximum gaps in total fulfillment cost and average time saving.

Approximation	EDS	VFA	LA (VFA + EDS)	FD	SD
Avg. Gap	0.0%	3.0%	3.0%	9.1%	24.0%
Max Gap	0.1%	4.6%	4.6%	11.4%	30.5%
Time Saved	90.2%	99.6%	99.9%		

expected fulfillment cost are minimal. As stated earlier, the efficient display sets are not a dominant display strategy as the maximum gap is slightly above 0%. Nevertheless, the EDS still generates all possible state spaces, which can grow exponentially as the number of zones and tasks grows. On the other hand, the approximation that only uses VFA performs relatively poorer compared to the EDS, but the time saving is significant. Also, the cost gap remains in an acceptable range: 3% on average and 4.6% for the maximum. The Look Ahead (LA) method, which combines both the EDS and VFA methods, results in an average gap of 3%, essentially indistinguishable from the use of VFA alone. As the computations of larger instances are not possible with exact approach or only EDS, we use the LA policy in our case study that is at the city scale (using Singapore data).

In Table 3, we also observe the performances of two benchmark policies. If the platform employs FD or SD policies, the fulfillment cost would be 9.1% or 24.0% higher on average compared to the dynamic and customized display policy.

6. Case Study: A Singapore Study

Throughout this section, we present the results and analyze the behavior of the stochastic look-ahead approach presented in Section 4 and observe its benefits compared to the Full Display (FD) and Single Display (SD) policies using Singapore based case study.

6.1. Singapore case instance generation

To derive insights for realistic applications, we design experiments that use Singapore as a test bed. We utilize Singapore's 44 mutually exclusive zones as delivery zones (illustrated in Figure 4). We assume that the only depot is located in Zone 13, which is consistent with the setup of our industry collaborator. We removed 13 zones with very low population densities, and the resulting number of zones is 31 (n_Z). To calculate the distance between zones, we employed the Haversine formula using the coordinates of the centroid of the polygon that represents each zone. Furthermore, we set the number of delivery tasks (n_P) and the length of the selection horizon T to be 1000. Table 4 summarizes the values of all parameters used in our case study.

While constructing the instances, we used the actual population of each zone and its distance from the depot as the main metrics to determine the parameters for each zone. A task is assigned

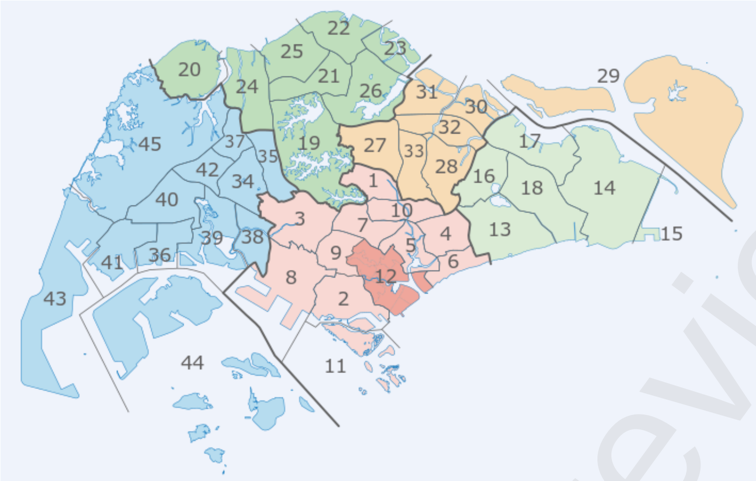

Figure 4: Singapore Planning Zones

randomly to zone i with probability ψ_i, where ψ_i is the fraction of the population that lives in zone i. Similarly, the cost of unselected tasks to the platform differs from zone to zone, and it is inversely proportional to the zone's proximity to the depot and is further explained at the end of this section.

The crowd drivers 'arrive' (i.e., open the App to browse the displayed tasks and make a selection) at different slots of the browsing period homogeneously. Similar to the validation instances, we assume that the number of driver types is equal to the number of zones, with type k driver's most preferred zone being zone k. For a period, the probability of no driver arriving in a selection period is $\lambda_0 = 0.05$, and the arrival probability for the type-k driver is proportional to the population of zone k and is equal to $\lambda_k = \psi_k/(1 - \lambda_0)$. If a driver commits to deliver a task, she receives a reward of \$30 – this reward is independent of the zone within which the selected task lies. Each additional kilometer of detour experienced by a driver translates into efforts equivalent to \$1. Therefore, we characterize the effort of serving zone z (ef_z) as the distance from the depot to zone z, that is, $tt(13, z)$.

A driver chooses a task (zone) among the ones displayed on her mobile App; they also have the option of not selecting any of the displayed zones and simply walking away. We model drivers' choice behavior using the MNL choice behavior model as outlined in Section 3.2. The no-choice utility, u^0, of each arriving driver equals \$1 for the base scenario. However, we also test the sensitivity of display policies by subsequently varying the value of this no-choice utility. It is important to note that no-choice utility affects the attractiveness of available tasks, and if a driver needs to make a significant detour to complete the delivery to a zone, they may be better off not choosing anything.

For the unselected/remaining tasks at the end of the browsing period, the platform has to incur the fixed cost $c_z(x) = a_z\sqrt{x}$ for each zone having at least one such task. The cost coefficient a_z varies between 300 and 500 for each zone, and is proportional to the square of the distance from the depot

Table 4: Parameters used in the Singapore case study.

Selection horizon	T	1000
Number of zones	n_Z	31
Number of delivery tasks	n_P	1000
Probability of no-arrival	λ_0	0.05
No-choice utility	u^0	1
Reward	$r_z = r, z \in Z$	30
Cost of non-completed tasks	$c_z(x), z \in Z$	$a_z\sqrt{x}$
Cost function coefficient	a_z	$300 + (500 - 300)(\frac{d_i}{d_{max}})^2$
MNL normalization parameter	α	0.1

to the zone, normalized by the maximum distance from the depot to a zone. This parameterization presents a challenge to the platform's operational model, as it incurs higher costs in zones that most drivers prefer less.

6.2. Base case results

In this section, we discuss the behavior and the performance of the dynamic and customized display (CD) policies in the Singapore case study. We compare the performance of CD policy to the two baseline policies, Full Display (FD) and Single Display (SD), using the KPIs described in Section 5.2.

Table 5: Base case results.

Policy	Fulfillment Cost		Matched Requests		Reward Ratio		Driver Utility	
	Ave	Max	Ave	Max	Ave	Max	Ave	Max
FD	4.8%	7.8%	4.9%	6.6%	0.0%	2.2%	32.5%	53.1%
SD	8.9%	11.1%	-31.7%	-33.3%	-37.3%	-39.0%	-43.1%	-70.2%

Table 5 summarizes the average and the maximum of the percent difference of the four comparison metrics for the two baseline policies compared to the CD policy. Figure 5 shows the range of the percent differences of the same metrics. The percent difference is calculated as the value of the baseline policy minus the value of the CD policy divided by the value of the CD policy. Therefore, a positive difference means that the metric value for the baseline policy is higher than the CD policy, and a negative difference means vice versa.

The results show that the CD policy provides substantial cost savings over the FD and SD policies. The gap between the CD and FD policies varies from 2.6% to 7.8%, with an average of 4.8%. For the FD policy, the number of tasks completed by the drivers is 4.9% higher on average; however, the reward ratio of the platform stays constant. FD policy provides drivers a greater choice autonomy, which decreases their likelihood of leaving the system empty-handed. However,

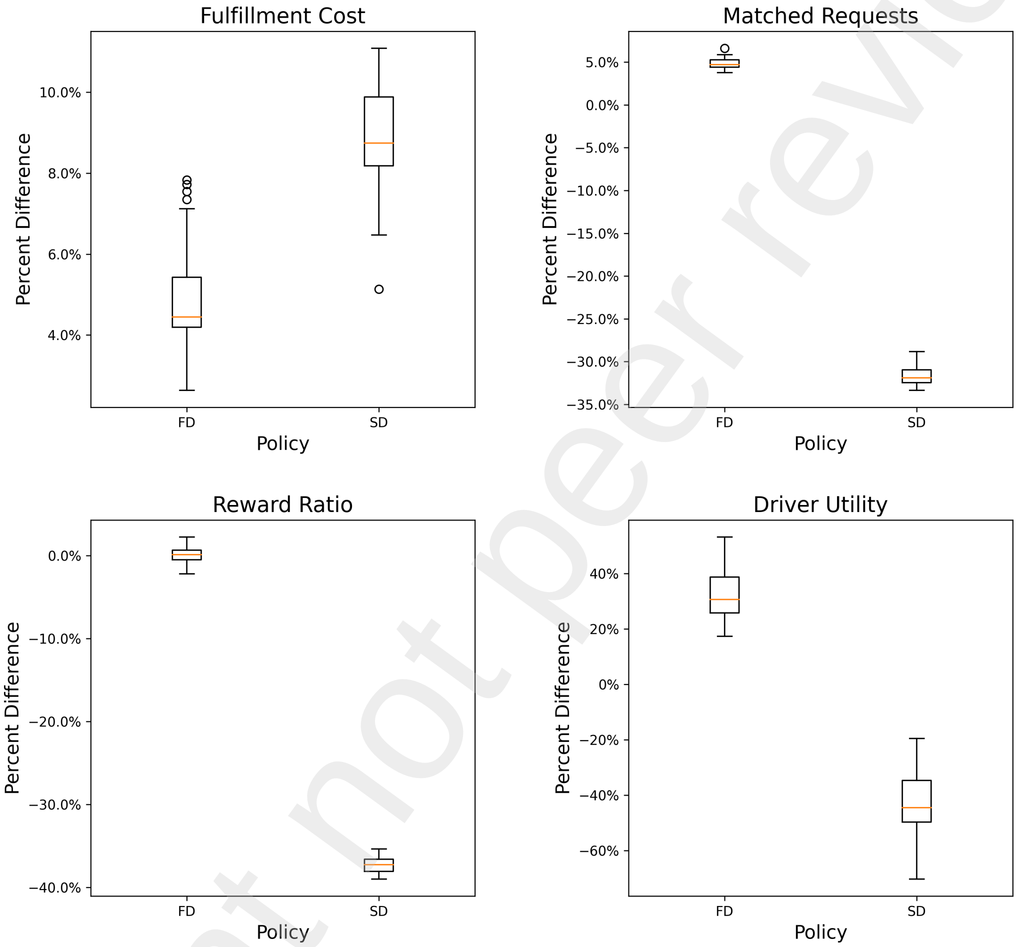

Figure 5: Percent gaps of comparison metrics: Baseline Policies vs. Customized Display.

although more reward is paid to the drivers in the FD policy, because of its decentralized nature, drivers usually end up myopically selecting tasks that do not help with fulfillment cost reduction.

Compared to the FD and CD policies, SD policy performs much worse. On average, implementing an SD policy costs 8.9% more than the CD policy, and the cost gap ranges between 5.1% and 11.1%. This is primarily caused by the decrease in the number of matched requests. In other words, while the SD policy aims to direct individual drivers towards tasks minimizing the total cost, a lack of sufficient choices results in a notable increase in drivers choosing nothing. This leads to an increased number of unselected tasks, which results in much higher cost for engaging contract drivers. This is also demonstrated in significantly lower reward ratio. Our proposed CD policy effectively mediates between the two extremes of FD and SD, increasing the number of matched tasks by providing a driver with greater number of choices but selectively, guiding drivers' choices to reduce the number of zones requiring contract drivers at the end.

As expected, the FD policy is the most beneficial for the drivers as displaying everything on hand gives them a better chance to maximize their utilities. The SD policy is the least beneficial policy for the drivers as the single zone displayed is independent of the driver's preferences. In aggregate, the CD policy scores in between these two policies. However, this is not always true when the evolution of the driver's utility is observed throughout the selection period. For the FD policy, the driver's potential utility is the highest among all policies until $T = 150$, and then decreases sharply. As drivers arriving early can choose the most preferred zones, the least desired zones are left for the drivers arriving towards the end of the selection period. The driver's maximum potential utility in the SD policy is relatively constant throughout the selection period, confirming the aim of the SD policy. The driver's utility in the CD policy is very close to the driver's utility in the SD policy at the early and mid stages of the selection period. However, the driver's utility increases towards the end of the selection horizon, surpassing the utility for the FD policy. This verifies that the CD policy saves some of the more attractive zones for the later decision periods.

In the FD policy, the number of zones displayed is always equal to the number of zones with a positive number of tasks. The SD policy always displays a single zone to the driver. The number of zones displayed in the CD policy lies somewhere in between, aiming for a more intelligently designed display set for the arriving drivers. That said, the cardinality of the display set varies over time. Figure 7 presents the number of zones with a remaining task and the number of displayed zones as a function of the remaining time epochs (left), as well as the number of displayed zones normalized to the number of zones with a remaining task (right). The CD policy generally does not display more than six zones in a given time epoch. The number of displayed zones decreases naturally over time with the decreasing number of zones with an available task. However, towards the end of the selection period, the percentage of displayed zones increases up to 50%. Moreover, as the end of the selection period approaches, the number of displayed zones averages below 1, which implies that

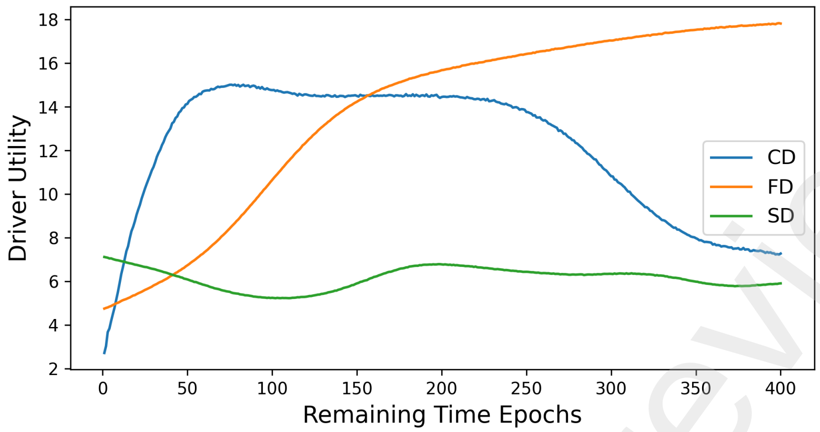

Figure 6: Evolution of driver's utility over time.

the CD policy chooses to display nothing if the expected cost benefit from displaying a zone does not cover the expected reward to be paid to the drivers.

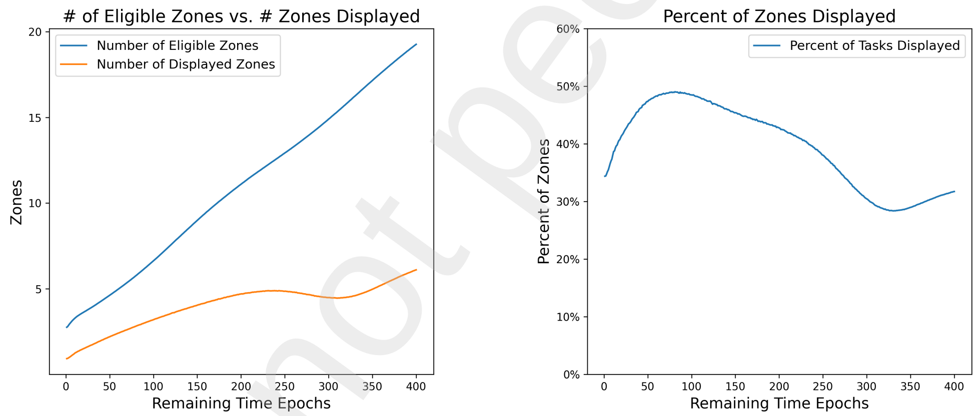

Figure 7: Number of eligible and displayed zones over time.

Finally, we discuss the implications of implementing a slight variation of the SD policy: Clearance-L. Similar to the SD policy, the Clearance-L policy aims at minimizing the potential expenses in engaging contract drivers, but it is allowed to show up to L zones instead of just 1. When $L = 1$, this policy is equivalent to the SD policy. If L is ever greater than the number of zones with remaining tasks, the policy will simply show all remaining zones. We calculate the expected total fulfillment cost of the Clearance-L policy for $L = 1$ to n_z through simulation and compared it with the total procurement cost of the CD policy in Figure 8. As seen from the chart, the percent gap of the total cost decreases as L increases, hits a turning point, and then follows a generally increasing trend. For our instances, Clearance-3 is the best performing Clearance policy, with an average total procurement cost of 2.7% higher than the CD policy. Moreover, the average procurement cost of this policy is less than both SD and FD policies. Therefore, when implementing the CD policy is not computationally feasible, one can adopt a Clearance-L policy as a cost-effective alternative to the FD or SD policies.

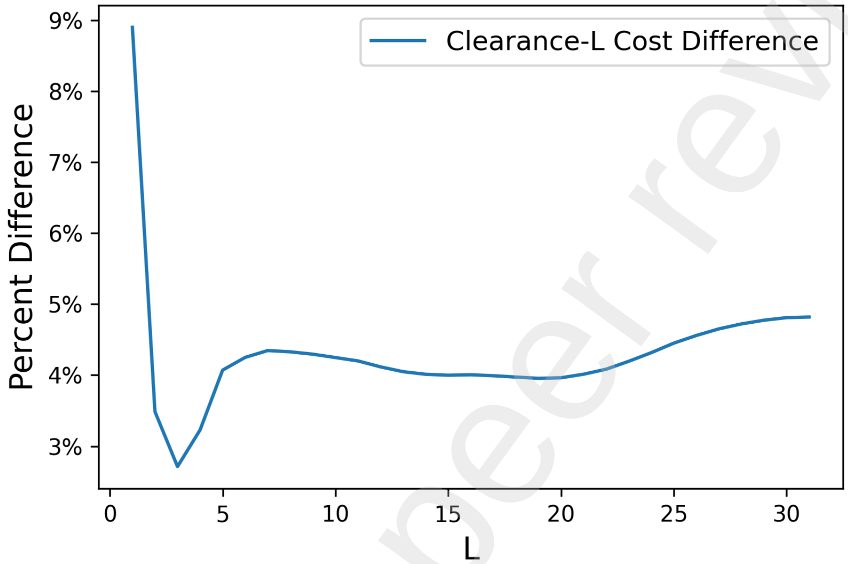

Figure 8: Cost gap of the Clearance-L policy vesus L.

6.3. Sensitivity analysis: No-choice utility

The utility that a crowdsourced driver does not serve any task displayed by the platform's App (u^0), plays a role in determining a driver's selection behavior and, thereby, the performance of different display strategies. Note that our model allows 'no choice' to be a legitimate selection outcome and thus be associated with a non-zero utility. To study the impact of this parameter, we vary u^0 while keeping the other parameters as shown in Table 4. For each driver, we vary the no-choice utility between 0 and $u^d_{\max}$, where $u^d_{\max}$ is defined as the maximum potential utility the driver d can get by completing a task. This range helps us to examine a wide variety of selection regimes. For example, when $u^0 = u^d_{\max}$, the no-choice utility is equal to that of the most-preferred task and higher than that of all other tasks, indicating a regime characterized by *low task attractiveness*. At the other extreme, $u^0 = 0$ represents a case where there is a low likelihood of a driver refusing all tasks. Note that zero utility does not imply zero choice probability, as the choice probability is calculated as a function of e^u rather than u itself.

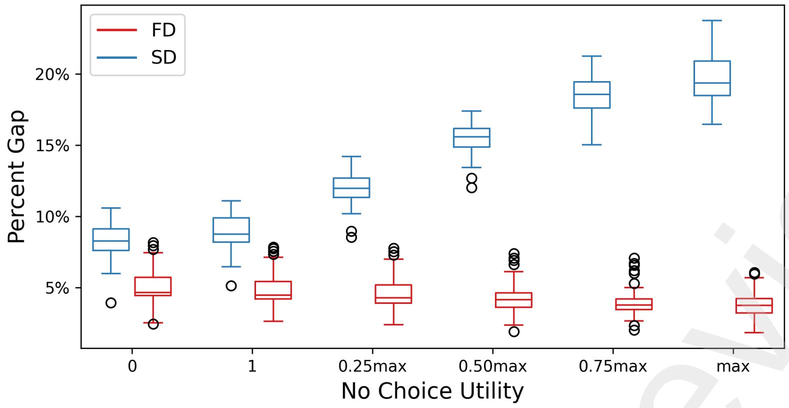

Figure 9: FC gaps of FD and SD with respect to the no-choice utility.

Figure 9 studies the relative performance of our proposed CD policy vis-a-vis the SD and FD policies. As CD always has the lowest FC value, the graph plots the percent gap (i.e., the percentage by which the FC of SD and FD policies exceed that of CD). In general, the FD policy beats the SD policy. As the no-choice utility increases, the FC gap between FD and CD decreases, whereas the FC gap between SD and CD increases. This also implies that the FC gap between FD and SD increases. Especially for the large values of the no-choice utility, the likelihood of a crowd driver leaving the platform without choosing anything increases significantly when a single zone is displayed. Because of this, the number of tasks completed is significantly less for the SD policy than the CD policy, which is the primary driver of the increasing gap in total fulfillment costs between these two policies.

However, even for very high values of no-choice utility, the FD policy suffers from the inefficiency of autonomy. As the platform does not have control over what the driver chooses under the FD policy, the contract driver costs incurred are higher than the CD policy, even though the number of tasks completed is higher with the FD policy for all values of no-choice utility. Although the cost of this inefficiency decreases with the increased no-choice utility, the fulfillment cost under the FD policy is 3% higher when $u^0 = u^d_{max}$. This shows that even when the crowd drivers have a relatively very high likelihood of leaving the platform without choosing anything, it is still beneficial for the platform to limit their options rather than providing them full autonomy.

6.4. Sensitivity analysis: MNL normalization parameter (α)

As stated in Equation (1), the relative likelihood of a crowd driver choosing a particular zone z depends on (i) driver k's utility of serving zone z, $u_k(z)$, and (ii) the MNL normalization parameter, α. In particular, α controls the amount of increase (or decrease) in the likelihood of a driver choosing a zone when the utility of the driver increases (or decreases). In other words, larger differences in driver utilities translate into larger differences in the likelihood and the choice probabilities as α increases. To study the impact of this translation, we vary α from 0.01 to 1 while keeping the other parameters as shown in Table 4. As the marginal effect of α is more significant when its value is smaller, we vary α in smaller ranges when it is closer to 0.1 and increase the step size as α grows larger.

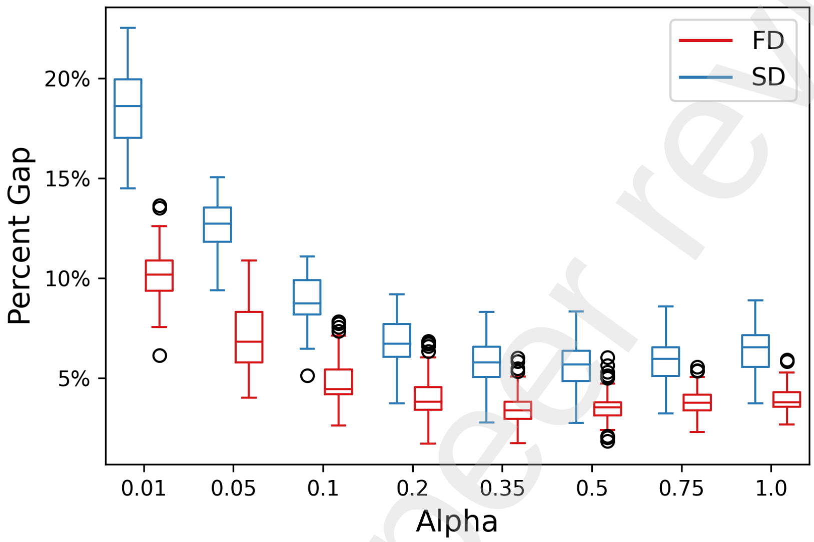

Figure 10: FC gaps of FD and SD with respect to α.

Figure 10 illustrates the relative difference in FC of our proposed CD policy compared to the SD and FD policies for various α values. For all α values, our proposed CD policy outperforms the SD and FD policies in terms of fulfillment costs (FC). Compared to the FD and SD policies, the benefit of operating under the CD policy is the highest when the drivers are the most indifferent. One thing to notice is that when $\alpha = 0.01$, the probability of a driver leaving the platform without choosing any of the provided options is the highest among all other values of α, even with the same display set. Therefore, optimizing display policy becomes even more important in these settings.

The benefit obtained by using the CD policy decreases as α increases to 0.35 and then starts to increase slightly as α grows larger. Under the FD policy, the drivers are increasingly inclined to choose the zone that yields the highest utility to them as α tends to 1. Therefore, the benefit of such a policy to the platform starts to decrease as α increases. Under the CD policy, increasing α increases the no-choice probability when the only zone that is displayed has a driver utility less than 1. This explains the slight increase in the FC gap of the SD policy.

7. Concluding Remarks

In this work, we consider the "Choice-based Crowdshipping" problem, where an online platform/mobile App is used to allocate overnight package delivery tasks to a pool of crowdsourced delivery workers. Taking advantage of the reasonably long (overnight) selection period, during which individual workers interact with the platform and commit to a specific delivery job, we investigate a mechanism that lies between the two extremes of (a) purely centralized, which is efficient in recommending tasks, but runs the risk of reducing worker acceptance rates, and (b) purely decentralized, which results in higher worker acceptance rates but can result in less favorable spatial distribution of unselected tasks. Our major contributions are the formalization of a mechanism for coming up with *customized display*, where individually arriving workers are presented with a choice of tasks from multiple carefully-chosen zones to balance the desire for a higher worker acceptance rate and a globally favorable spatial clustering of unselected tasks. The customized display process is formalized as a Markov Decision Process that dynamically selects the set of displayed tasks, taking into account (a) the spatial properties of currently-unselected tasks and (b) the anticipated future driver arrivals.

Our numerical experiments demonstrate significant benefits for our customized display mechanism, which achieves a favorable trade-off between worker selection rates and total platform cost than achievable by either purely-centralized or decentralized approaches. Under tested settings, SD policy results in overall task matched rates 31.7% lower than our CD policy; compared against the fully-decentralized FD approach, our CD approach achieves only modest 4.9% lower task matched rates. More importantly, even with a lower task matched rate, CD results in a total platform cost that is significantly (4.8%) lower than the decentralized FD approach on average, primarily by proactively reducing the contract driver cost in fulfilling the unselected tasks. (In addition, total platform costs with the CD policy are also 8.9% lower than the centralized alternative, where the lower crowdsourced task acceptance rate sharply pushes up the cost spent on contract drivers.)

We believe that our findings and insights have practical implications for the broader crowdshipping business and the design of online task assignment/selection platforms. In particular, for scenarios where the task-worker matching needs not be performed instantaneously, our work demonstrates the benefits of *dynamically* tailoring the list of tasks displayed to each worker by considering the spatio-temporal statistics of future worker arrivals. Our work also suggests the importance of designing platform behavior to balance the desire for worker autonomy and assignment efficiency, especially for scenarios where task fulfillment is achieved through a combination of an elastic crowdsourced worker pool, backstopped by contracted resources (delivery workers, trucks, etc.). There are several additional aspects and refinements possible within this broad area.

- *Non-Stationary Driver Arrivals:* Our numerical analysis in this paper assumed a time-homogeneous

Poisson process for driver arrivals. In practice, many deadline-based online platforms exhibit a non-stationary pattern, with a significantly higher rate of worker arrivals closer to the deadline. Our formulation can, however, be relatively easily modified to handle such non-stationary statistics. In particular, one potential approach is to adjust period lengths according to arrival rates (e.g., make period length inversely proportional to the arrival rate), and thereby maintaining our model's basic assumption of a negligible probability of two or more driver arrivals within any slot.

- *Multiple Task Selection:* To simplify the analysis, we assume that each driver is able to select at most one task. Thus, a natural direction for the future work is to extend this analytical framework to accommodate scenarios where individual crowd drivers can commit to multiple tasks. This will require changes to the underlying sequential decision problem, incorporating additional constraints (e.g., capacity constraints associated with each driver's vehicle) and updated preference models (e.g., higher preference to select two proximately-located tasks). In general, this will likely lead to significantly higher computational complexity and will thus require powerful heuristics.

- *Driver's Task Selection Model:* In our present work, we've modeled the driver's selection choice via a MNL model, where the utility associated with a task is a linear function of the task reward and the delivery detour. Alternative variants of utility definitions are certainly possible and would influence the resulting display choices. Empirical studies will be needed to determine whether additional behavioral effects (e.g., the tendency of users selecting only the top-most 1 or 2 responses for online search queries) manifest in a crowdsourcing mobile App.

References

Agatz, N., Fan, Y., Stam, D., 2021. The impact of green labels on time slot choice and operational sustainability. Production and Operations Management 30, 2285–2303.

Alnaggar, A., Gzara, F., Bookbinder, J.H., 2019. Crowdsourced delivery: A review of platforms and academic literature. Omega , 102139.

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

Arslan, A.M., Agatz, N., Kroon, L., Zuidwijk, R., 2019. Crowdsourced delivery: A dynamic pickup and delivery problem with ad hoc drivers. Transportation Science 53, 222–235.

Ausseil, R., Pazour, J.A., Ulmer, M.W., 2022. Supplier menus for dynamic matching in peer-to-peer transportation platforms. Transportation Science .

Ben-Akiva, M., Bierlaire, M., 1999. Discrete choice methods and their applications to short term travel decisions, in: Handbook of transportation science. Springer, pp. 5–33.

Bernstein, F., Kök, A.G., Xie, L., 2015. Dynamic assortment customization with limited inventories. Manufacturing & Service Operations Management 17, 538–553.

Chen, C., Cheng, S.F., Gunawan, A., Misra, A., Dasgupta, K., Chander, D., 2014. Traccs: Trajectory-aware coordinated urban crowd-sourcing, in: Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA.

Dai, H., Liu, P., 2020. Workforce planning for o2o delivery systems with crowdsourced drivers. Annals of Operations Research 291, 219–245.

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

Demirbilek, M., Branke, J., Strauss, A., 2019. Dynamically accepting and scheduling patients for home healthcare. Health care management science 22, 140–155.

Einav, L., Farronato, C., Levin, J., 2016. Peer-to-peer markets. Annual Review of Economics 8, 615–635.

Gallego, G., Topaloglu, H., et al., 2018. Revenue management and pricing analytics. Springer.

Gurvich, I., Lariviere, M., Moreno, A., 2019. Operations in the on-demand economy: Staffing services with self-scheduling capacity, in: Sharing Economy. Springer, pp. 249–278.

Horner, H., Pazour, J., Mitchell, J.E., 2021. Optimizing driver menus under stochastic selection behavior for ridesharing and crowdsourced delivery. Transportation Research Part E: Logistics and Transportation Review 153, 102419.

Joerss, M., Schröder, J., Neuhaus Florian, Klink, C., Mann, F., 2016. Parcel delivery. the future of last mile. URL: https://www.mckinsey.com/~/media/mckinsey/industries/travel%20transport%20and%20logistics/our%20insights/how%20customer%20demands%20are%20reshaping%20last%20mile%20delivery/parcel_delivery_the_future_of_last_mile.ashx.

Kafle, N., Zou, B., Lin, J., 2017. Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. Transportation research part B: methodological 99, 62–82.

Kandappu, T., Jaiman, N., Tandriansyah, R., Misra, A., Cheng, S.F., Chen, C., Lau, H.C., Chander, D., Dasgupta, K., 2016. Tasker: Behavioral insights via campus-based experimental mobile crowd-sourcing, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, New York, NY, USA. pp. 392–402. URL: http://doi.acm.org/10.1145/2971648.2971690, doi:10.1145/2971648.2971690.

Kök, A.G., Fisher, M.L., Vaidyanathan, R., 2008. Assortment planning: Review of literature and industry practice, in: Retail supply chain management. Springer, pp. 99–153.

Le, T.V., Stathopoulos, A., Van Woensel, T., Ukkusuri, S.V., 2019. Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. Transportation Research Part C: Emerging Technologies 103, 83–103.

Le, T.V., Ukkusuri, S.V., 2018. Selectivity correction in discrete-continuous models for the willingness to work as crowd-shippers and travel time tolerance. arXiv preprint arXiv:1810.00985 .

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

Miller, J., Nie, Y., Stathopoulos, A., 2017. Crowdsourced urban package delivery: Modeling traveler willingness to work as crowdshippers. Transportation Research Record: Journal of the Transportation Research Board , 67–75.

Mofidi, S.S., Pazour, J.A., 2019. When is it beneficial to provide freelance suppliers with choice? a hierarchical approach for peer-to-peer logistics platforms. Transportation Research Part B: Methodological 126, 1–23.

Musthag, M., Ganesan, D., 2013. Labor dynamics in a mobile micro-task market, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM. pp. 641–650.

Paloheimo, H., Lettenmeier, M., Waris, H., 2016. Transport reduction by crowdsourced deliveries–a library case in finland. Journal of Cleaner Production 132, 240–251.

Rai, H.B., Verlinde, S., Macharis, C., 2018. Shipping outside the box. environmental impact and stakeholder analysis of a crowd logistics platform in belgium. Journal of cleaner production 202, 806–816.

Sampaio, A., Savelsbergh, M., Veelenturf, L., Van Woensel, T., 2019. Crowd-based city logistics, in: Sustainable Transportation and Smart Logistics. Elsevier, pp. 381–400.

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

Simoni, M.D., Marcucci, E., Gatta, V., Claudel, C.G., 2019. Potential last-mile impacts of crowdshipping services: a simulation-based evaluation. Transportation , 1–22.

Talluri, K., Van Ryzin, G., 2004. Revenue management under a general discrete choice model of consumer behavior. Management Science 50, 15–33.

Taylor, T.A., 2018. On-demand service platforms. Manufacturing & Service Operations Management 20, 704–720.

Ulmer, M., Savelsbergh, M., 2020. Workforce scheduling in the era of crowdsourced delivery. Transportation Science 54, 1113–1133.

Ulmer, M.W., Thomas, B.W., 2019. Enough waiting for the cable guy—estimating arrival times for service vehicle routing. Transportation Science 53, 897–916.

Wang, H., Yang, H., 2019. Ridesourcing systems: A framework and review. Transportation Research Part B: Methodological 129, 122–155.

Yang, X., Strauss, A.K., Currie, C.S., Eglese, R., 2016. Choice-based demand management and vehicle routing in e-fulfillment. Transportation science 50, 473–488.

Yildiz, B., Savelsbergh, M., 2019. Service and capacity planning in crowd-sourced delivery. Transportation Research Part C: Emerging Technologies 100, 177–199.

Young, J., 2019. Global ecommerce sales grow 18% in 2018. URL: https://www.digitalcommerce360.com/article/global-ecommerce-sales/.