Choice-based Crowdshipping for Next-day Delivery Services: A Dynamic Task Display Problem

Alp Arslan*

Lancaster University Management School

Fırat Kılcı

University of Texas at Austin

Shih-Fen Cheng, Archan Misra

Singapore Management University

Abstract

This paper studies integrating the crowd workforce into next-day home delivery services. In this setting, both crowd drivers and contract drivers collaborate in making deliveries. Crowd drivers have limited capacity and can choose not to deliver if the presented tasks do not align with their preferences. The central question addressed is: How can the platform minimize the total task fulfilment cost, which includes payouts to crowd drivers and additional payouts to contract drivers for delivering the unselected tasks by customizing task displays to crowd drivers? To tackle this problem, we formulate it as a finite-horizon Stochastic Decision Problem, capturing crowd drivers' utility-driven task preferences, with the option of not choosing a task based on the displayed options. An inherent challenge is approximating the non-constant marginal cost of serving orders not chosen by crowd drivers, which are then assigned to contract drivers. We address this by leveraging a common approximation technique, dividing the service region into zones. Furthermore, we devise a stochastic look-ahead strategy that tackles the curse of dimensionality issues arising in dynamic task display execution and a non-linear (problem specifically concave) boundary condition associated with the cost of hiring contract drivers. In experiments inspired by Singapore's geography, we demonstrate that choice-based crowd shipping can reduce next-day delivery fulfilment costs by up to 16.9%. The observed cost savings are closely tied to the task display policies and the task choice behaviors of drivers.

Keywords: Crowdsourced Delivery, Drivers' Autonomy, Last-mile Logistics, Dynamic Task Display

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

1. Introduction

The growth of e-commerce has resulted in substantial demands for field operation workforce, particularly for last-mile delivery. 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 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 labour supply that can efficiently respond to demand variations; e.g., during holiday season peaks (Einav et al., 2016).

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). 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, DoorDash) centralize task assignment, pairing drivers and delivery requests quickly without elaborate drivers' consultation. In such scenarios, the chance of worker rejection of centralized assignments is lower because the worker's instantaneous location drives assignment decisions. The situation is, however, reasonably distinct for next-day delivery tasks, a dominant fraction of the e-commerce market, where delivery tasks are known in advance, and the task-worker allocation usually happens over a longer time window. Moreover, the remaining tasks at the end of the selection period can be serviced by contract drivers, who could be recruited hourly or payroll employees of shippers to guarantee promised deliveries. In such a scenario, the decentralized assignment mechanism could be more appealing as it allows crowd workers to choose preferred tasks based on their anticipated itinerary.

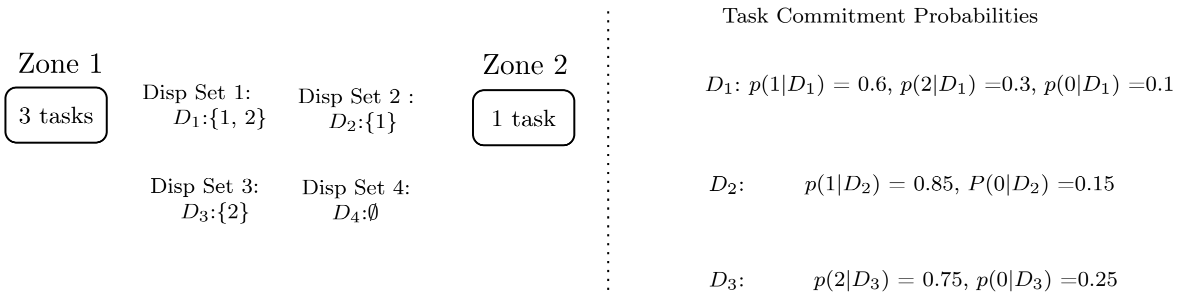
The idea of mixing crowdsourced and contract drivers for next-day delivery tasks comes from our collaboration with UrbanFox, a mid-size logistics service provider in Southeast Asia. As reported by NHK (2018), UrbanFox has been a pioneer in Singapore in utilizing crowdsourced workers to supplement their contract workers during peak seasons. However, their fully decentralized/autonomous matching has some shortcomings, and they have been searching for solutions to match crowdsourced workers and deliver tasks more effectively.

This paper, therefore, focuses on a next-day last-mile delivery platform and investigates the development of a decentralized task assignment mechanism that mimics the efficiency of a centralized approach. Our proposed mechanism grants crowd drivers autonomy in task selection while preserving the benefits of central planning. This is achieved by presenting crowd drivers with effectively curated subsets of available tasks, allowing the central planner to influence drivers towards service regions with higher clearance probabilities. This, in turn, enables task consolidation and reduces the cost of engaging contract drivers, which is proportional to the number of service regions with unselected tasks. To maintain alignment with the overarching objective of minimizing service regions with unselected tasks, we incorporate a realistic cost approximation method to estimate the expenses incurred in engaging contract drivers. The subsequent section illustrates how this approach operates in practice.

1.1. An illustrative example

Assume that we are deciding on the set of tasks to show to the last incoming driver in order to minimize the platform's fulfilment cost for Zones 1 and 2, where there are three and one remaining tasks, respectively. We also assume that this driver chooses at most one task and probabilistically chooses Zone 1 over Zone 2. There are four possible display scenarios: D_1 shows tasks from both zones, D_2 shows tasks from Zone 1, D_3 shows tasks from Zone 2, and D_4 shows no task. Let $p(i|D_j)$ be the probability that the driver would choose a task from Zone i given a display set D_j. $p(0|D_j)$ is the case where the driver chooses not to serve.

Figure 1: An illustrative example how display sets work.



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 unselected tasks remaining, 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:

- D_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 $(100+2\cdot 10)+(100+10) = 230$. 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: $0.6\cdot 230+0.3\cdot 130+0.1\cdot 240 = 201$.

- D_2: The expected cost is: $0.85 \cdot 230 + 0.15 \cdot 240 = 231.5$.
- D_3: The expected cost is: $0.7 \cdot 130 + 0.3 \cdot 240 = 157.5$.
- D_4: The cost of fulfilment is 240.

In the above example, the use of display set D_3 (i.e., showing the crowd driver a task only from Zone 2) results in the lowest expected cost. This is even though the driver prefers Zone 1 and thus has a significantly higher probability of not choosing any task in the 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 tasks.

Of course, this is an overly simplified example designed only to demonstrate the benefits of display set customization. The complete model captures the complexity of having more zones, heterogeneous driver preferences, and the non-linear cost of engaging contract drivers. Moreover, we also consider the sequential nature of the decision-making process; i.e., drivers make their selection asynchronously and sequentially. Therefore, earlier display decisions impact the pool of tasks available for selection in the latter periods, future drivers' preferences and hence, the display sets.

1.2. Summary of contributions

We refer to this model of 'centralized customization, autonomous selection' as *choice-based crowdshipping for next-day delivery.* Thus, this paper's central theme is to develop and quantify the significance of such display policies, which strategically incorporate both the choice behaviours of crowd drivers and the platform's cost-minimizing objective. To analyze choice-based crowdshipping in the next-day delivery service, we introduce the Dynamic Task Display Problem (DTDP). In DTDP, (a) there is a finite duration (Selection Horizon) over which crowd drivers arrive randomly and request the platform/App to display tasks from which to make a selection, (b) the platform dynamically determines the subset of displayed tasks for each individually arriving driver, and (c) the platform hires contract drivers to make the delivery of remaining tasks after the selection period. We make the following contributions:

- We propose a stochastic look-ahead strategy to solve the computationally intractable dynamic task display problem in realistic-size instances. Our solution approach is built upon two pillars:(i) Value Function Approximation to address the state space expansion and (ii) Efficient Display Sets to address the action space expansion.
- Instances inspired by Singapore's geographical properties, we numerically show that enabling choice-based crowdshipping decreases the fulfilment cost of overnight delivery tasks up to 16.9% by balancing the workload between the crowd and contract drivers.
- The experiments exhibit that the chosen display policy significantly influences cost savings that could be obtained from choice-based crowdshipping. The proposed customized task display policy consistently outperforms other benchmarks representing fully decentralized (display all

tasks) and centralized strategies (priority on cost-saving).

- We observe the amount of cost-savings is sensitive to the reward paid to drivers and the number of arriving crowd drivers. We also observe that the fully decentralized display policy may increase the total fulfilment cost if the reward amount of the drivers is exceptionally high or the crowd drivers become picky.
- Our method of customized display policy shows additional benefits when the crowd driver's task choice behaviour becomes less predictable. Also, we see that the cost-saving results remain robust to the parameters of the contract drivers' cost function.
- For practitioners using or planning to deploy a crowdshipping system for a next-day delivery service, we highlight the following outcomes from the experiments:
  - Setting a crowdshipping system for next-day delivery requires more sophisticated decision support in comparison to deploying in the on-demand delivery services,
  - Prioritizing the crowd driver in task delegating is not always a cost-efficient strategy,
  - Choice customization is a more valuable strategy if the reward paid to crowd drivers is neither too high nor low.

The remainder of this paper is organized as follows. In Section 2, we summarize the relevant literature for the choice-based crowdshipping. Section 3 begins by describing the conceptual business model. In the sequel, the dynamic task display problem is formulated as a Sequential Decision Problem. In Section 4, we present the solution method for the customized display policy. Sections 5 and 6 consist of computational experiments to quantify the effectiveness of choice-based crowdshipping and display policies. Section 7 ends the paper with concluding remarks and some directions for future research.

2. Literature Review

The DTDP is a crowdshipping problem as its primary purpose is to efficiently integrate willing crowd drivers to serve a set of logistics and transportation requests. Crowdshipping 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., 2021; Savelsbergh and Ulmer, 2022). 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. Furthermore, Boysen et al. (2022); Mousavi et al. (2021) consider settings where crowd drivers have priority in the assignment phase under the assumption that the cost of serving with crowd drivers will always be lower than using third-party logistic services. This assumption may hold true for on-demand deliveries; however, neglecting the consolidation opportunities in next-day delivery services would lead to suboptimal solutions.

Several extensions to the above crowdshipping problems have been studied in the literature, including relay-like mechanisms that crowd shippers finish the last leg of delivery using company vehicles (Kafle et al., 2017), integrating public transit passengers and transfers between crowd drivers (Yıldız, 2021; Macrina et al., 2020; Kızıl and Yıldız, 2023), using in-store customers and employees (Dayarian and Savelsbergh, 2020; Dayarian and Pazour, 2022), and bundling of tasks (Mancini and Gansterer, 2022).

Stochastic crowd driver behaviour creates operational challenges, and several studies have studied the resulting issues. Gdowska et al. (2018) consider a setting where crowd drivers could decline the delivery option. Mousavi et al. (2022) integrate stochastic crowd driver participation in mobile depot location problem. Torres et al. (2022) explore a new crowdshipping variant in which crowd drivers may not show up after the task allocation phase in a next-day delivery setting. Hence, they propose a two-stage approach to consider recourse action to keep the service level. Furthermore, Haferkamp et al. (2024) examines the impact of displaying heat maps where the tasks' pick-up density is highlighted, and they show the benefits of using the New York taxi data set.

Centralized task assignment mechanisms are widely used for real-time operations with a few exceptions. We are aware of the papers by Mofidi and Pazour (2019); Horner et al. (2021); Ausseil et al. (2022); Karabulut et al. (2022); Yang et al. (2024) studying the concept that platforms offer a menu of task options to drivers. Mofidi and Pazour (2019) explore a bi-level optimization problem if a driver chooses a task from a menu. However, the driver's task selection behaviour 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). In the last stage, the platform matches drivers to their preferred tasks to maximize utility. Ausseil et al. (2022) enrich the menu of task options by analysing the potential conflicts that may arise due to the same task being offered to different drivers. In both studies by Horner et al. (2021); Ausseil et al. (2022), the driver selection behaviour is modelled through a utility function taking distance time as inconvenience and driver choice is derived from the ranking per scenario of the options offered. Karabulut et al. (2022), on the other hand, specifically concentrate on quantifying the value of adaptive menu sizes in peer-to-peer platforms with a specific focus on the agent's behaviour, while learning drivers' request selection behaviours through their earlier choices. Yang et al. (2024) similarly develop the utility function per driver order combination, taking into account the compensation amount, order type (passenger or goods delivery), order location' attractiveness.

The essential difference between supplier menu design studies for on-demand setting and the menu design for the next-day delivery is the cost structure of unselected tasks. In the on-demand logistics/transportation setting, the unselected tasks by the crowd are very costly and undesirable as it is costly to find a back-up service to fulfil it. Nevertheless, in the next-day delivery and the opportunities of consolidation, tasks nearby can be delivered by a single van or a contract driver. Hence, the marginal cost of each delivery can be lower than delegating to a crowd driver. As a result, the allocation of tasks between the crowd and the contract driver is not trivial, especially since the crowd behaviour is stochastic and the cost of fulfilling not-selected or not-displayed tasks is not constant or linear in terms of task number but varies with their geographical spread.

After the 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 schedules by performing tasks without choice autonomy. Therefore, these studies examine customer pricing and driver compensation schemes to coordinate the friction between transportation demand and driver supply. This study, however, mainly considers individuals with itineraries instead of scheduled drivers.

In contrast to these past studies, we are among the first to study crowdshipping in next-day delivery services. The overnight delivery problem differs substantially from the real-time on-demand task assignment problems, especially in how the unassigned/not selected orders/tasks are handled and the cost is accounted for at the end of the selection period. Also, we explicitly model the stochastic driver choice model and incorporate the drivers' choice model in our formulation with the possibility of not making a delivery if the choices are not satisfactory. This setting enables us to consider the crowdshipping and traditional delivery resources equally without making one another a priority. Furthermore, in this study, we explicitly study the consolidated delivery opportunities for contract drivers, which leads us to examine various practical insights for practitioners and present viable strategies to integrate crowd drivers into next-day delivery services.

3. Problem Definition

This section describes the Dynamic Task Display Problem (DTDP). We first provide an overview of exactly how our proposed paradigm of choice-based crowdshipping would work in practice. After the description of the choice-based crowdshipping system, we provide a formal definition of the DTDP and formulate it as a Markov decision process (MDP).

3.1. Choice-based Crowdshipping System Architecture

This study's platform architecture is based on the business model of UrbanFox (Fai, 2019), a mid-size logistics service provider in Southeast Asia and a research project collaboration with Singapore Management University, A*STAR and Fujitsu (Fujitsu et al., 2018). The focal platform engages crowd drivers by asking them to browse and select delivery tasks on a smartphone App daily. Such browsing is enabled during a *Selection Horizon*, typically spanning several hours in the evening before a deadline. During this selection horizon, individual crowd drivers arrive at the platform randomly. The platform decides which unassigned tasks should be displayed when each driver arrives. Customizing the set of displayed tasks allows the platform to steer the collective selection behaviour of crowd drivers.

Figure 2: Operational workflow for an overnight delivery platform that utilizes both crowd and contract drivers.

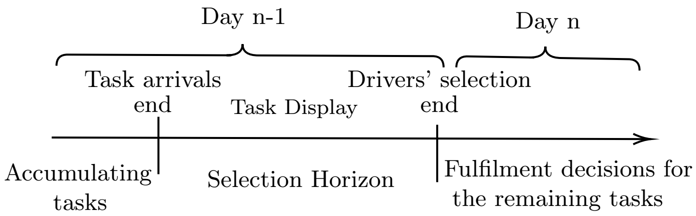


Figure 2 visualizes the operational workflow of this platform. All tasks to be delivered in day n should already arrive in day $n-1$ before the first cut-off time, after which the platform decides which tasks/clusters of tasks should be displayed to a crowd driver whenever such a driver interacts with the platform. At the selection horizon, the platform terminates delegating tasks to crowd drivers, and therefore, the remaining orders must be fulfilled by the contract drivers. The platform aims to minimize the total fulfilment cost, which depends on the crowd driver compensation and the cost of delivering the remaining tasks using contract drivers. All accumulating tasks should be fulfilled.

While computing the crowd driver compensation is straightforward due to the predetermined reward per task delivery, it is complicated to compute the cost of engaging contract drivers. Given a set of remaining tasks, finding the most cost-effective way to serve them using contract drivers is essentially a k-travelling salesman problem (k-TSP) (Xu et al., 2013), which involves solving both the subset selection problem (select subsets of tasks to be served by each driver) and the TSP for each selected subset. To focus on the crowd worker management problem, we approximate the contract driver cost using the continuous approximation (CA) by Daganzo (2005). More precisely, we partition the whole planning region into smaller zones such that a single contract driver can adequately serve each zone. Our approach is inspired by the widely adopted *cluster-first, route-second* approaches in the logistic industry; e.g., see Holland et al. (2017). Furthermore, to reflect the high fixed cost in engaging contract drivers and the economies of scale in serving tasks within each zone, we define the cumulative cost function to be concave and increasing in the number of tasks delivered for each zone. This setup reflects the impact of the number of tasks and their spatial distribution realistically in the total cost approximation by contract drivers. Therefore, the minimization of the fulfilment cost depends not only on the number of residual, unselected tasks after the selection period, but also on the geographical spread of these tasks; *i.e.,* the number of zones with tasks in our cost approximation. In the following subsection, the Dynamic Task Display Problem (DTDP) will be described formally.

3.2. Dynamic Task Display Problem

The DTDP considers a next-day delivery provider, or shortly platform, fulfilling known customer delivery requests with crowd drivers and contract drivers within a delivery area. The relevant assumptions and notations for DTDP are as follows.

Selection horizon. We denote the selection horizon when a crowd driver arrives and browses a delivery task, as $\mathcal{T}$, where $t = T$ is the start of the horizon and $t = 0$ is the end of the selection horizon.

Geography and customer orders. The DTDP covers a fixed delivery area. The platform seeks to deliver a set of customer orders, denoted by $\mathcal{O}$, which are known before the selection horizon, and all orders are picked up from a single depot. The delivery area is mutually exclusively partitioned into a set of zones, denoted by $\mathcal{Z}$ and $Z = |\mathcal{Z}|$. Each request $o \in \mathcal{O}$ is part of a single zone $z \in \mathcal{Z}$ such that set $\mathcal{O}_z$ denotes all the requests in zone z. Furthermore, each zone has area A_z and its distance from the depot β_z. Distance between two zones, z_1 and z_2 denoted by $tt(z_1, z_2)$.

Contract drivers and recruiting cost. The platform recruits a contract driver for each zone in the absence or lack of willing crowd drivers to make all deliveries. The cost of engaging a contract driver is proportional to the total duration of delivery operation including loading/unloading activities at the depot, time to travel to the zone from the depot, service time per customer, and the time between two customers in the zone. Hence, we compute the cost of a contract driver serving zone $z \in \mathcal{Z}$ with $x_z = |\mathcal{O}_z|$ as the following equation.

$$f(x_z) = \Gamma\big[\beta_z + \beta_s x_z + \beta_{tz}\sqrt{A_z x_z}\big], \tag{1}$$

where Γ represents the hourly rate of hiring a contract driver, β_z is the setup time (loading, unloading, and travelling to zone) for serving the zone, β_s is the time of serving a single order(independent of the zone type), and β_{tz} is the time coefficient for moving between two customers' location within the zone z. See the continuous approximation in (Daganzo, 2005) for the derivation of Equation (1).

Crowd drivers and rewards. Throughout the selection horizon, the arrival times of crowd drivers follow a stochastic process, denoted by $\Lambda(t), t \in \mathcal{T}$. Each crowd driver is associated with their selfdeclared destination zone $m \in \mathcal{Z}$; shortly, we call drivers going to zone $m \in \mathcal{Z}$, m type drivers. Also, let λ_{tm} be the arrival rate of a driver with destination to zone $m \in \mathcal{Z}$ such that $\sum_{m \in Z} \lambda_{tm} = \Lambda(t)$, for each $t \in \mathcal{T}$.

Let $u_m(z)$ be the utility of m type crowd driver deliver a request in zone $z \in \mathcal{Z}$. We define the utility of a driver of type m for serving zone z as the total net payoff obtained from serving that zone, which is equal to the reward received, minus the penalty of making a stop in zone z rather than travelling directly to their preferred zone m, plus a random shock that is unobserved by the decision maker and can be expressed mathematically as follows.

$$u_m(z) = r_z - \beta_{detour}(tt(d,z) + tt(z,m) - tt(d,m)) + \varepsilon_{mz}, \tag{2}$$

In Equation (2), r_z refers to the reward of serving a task in Zone z, $tt(d,z)$ is the distance from zone d to zone z and d signifies the zone where the depot is located. β_{detour} is the disutility coefficient per unit distance travelled for detour, and ε_{mz} is i.i.d. standard Gumbel random variable. Depending on the driver type m, and the zone z, the disutility of distance travelled can be larger than the reward for zone z, r_z. Therefore, we allow $u_m(z)$ to take negative values.

Our framework also allows drivers to opt in for an outside option (i.e., do not choose any of the zones/tasks displayed by the platform). We define the utility of the outside option as $u_m(0)$. A common value for the utility of the outside option is 0, which is a result of receiving zero reward by exerting no effort. However, in cases where the outside option represents switching to a competitor platform, $u_m(0)$ may take positive values.

Following the framework by Ben-Akiva and Bierlaire (1999), we model the choice probabilities when D - set of zones - is displayed to driver type m using the Multinomial Logit (MNL) model and the choice probabilities can be written as follows.

$$p_m(z|D) = \frac{e^{\alpha u_m(z)}}{e^{\alpha u_m(0)} + \sum_{l \in D} e^{\alpha u_m(l)}}, \qquad p_m(0|D) = \frac{e^{\alpha u_m(0)}}{e^{\alpha u_m(0)} + \sum_{l \in D} e^{\alpha u_m(l)}}. \tag{3}$$

The implication of defining the choice probability using (3) is that the likelihood of not serving increases when the display set size decreases. This is consistent with our modelling assumption and the empirical studies mentioned in the introduction. Also, the parameter $\alpha > 0$ enables us to quantify the choice adherence of the driver to the more favourable task. In the numerical section, we utilize α to see the impact of the predictability of driver preferences.

3.3. DTDP as a Markov Decision Process

The DTDP operates in a dynamic and stochastic environment due to the uncertain nature of driver arrivals and the task choices of each driver. Therefore, we present the DTDP as a finite-horizon stochastic problem with the following Markov Decision Process components.

Decision epochs. Each driver-arriving epoch $t \in \mathcal{T}$ is a decision epoch, after which the platform has to decide which tasks to display to the driver. We regard T as the beginning of the selection period; hence, period t represents the number of remaining periods before the selection period terminates, and $t = 0$ is the terminal epoch.

States. The state of the system at epoch t is represented by a vector of $X_t = [x_{t1}, \cdots, x_{tZ}]$, in which $x_{tz},\ z \in \mathcal{Z}$, represents the number of customer requests in zone z at period t.

Actions and costs. The platform decides which set of zones will be displayed after the driver arrives and her type (destination zone) is revealed at period t. We represent this decision as a set $D \subset \mathcal{Z}$, which includes zones the platform wants to reveal to the driver type m. Let $\mathcal{D}_t$ be a collection of all possible display sets the platform can display at period t.

The platform incurs a cost only if the driver decides to deliver a request in zone $z \in D \in \mathcal{D}_t$ in period t. In this case, the platform compensates the crowd driver with the reward of $r_z > 0$. Furthermore, as the platform is bound to deliver all customer requests, the remaining unselected customer requests after the selection horizon terminates should be delivered by the contract drivers. Hence, the cost paid for all contract drivers recruited will be given as $\sum_{z\in\mathcal{Z}} f(x_{0z})$ following Equation 1.

Transitions. A new decision epoch is triggered upon the new crowd driver's arrival. After the platform takes the action, the system changes as follows:

- System state X_t is updated if the crowd driver at period t chooses to make delivery in zone z; *i.e.,* $X_t - e_z$, where e_z is a unit vector where z^{th} element equals 1.
- The decision epoch becomes $t - 1$.

Objective function. Let Π be the set of all Markovian deterministic policies and π is a sequence of decision rules: $\pi : (\delta_T^\pi, \cdots, \delta_t^\pi, \cdots, \delta_0^\pi)$, where each decision rule $\delta_t^\pi : \mathcal{D}_t \mapsto D$ is a function that specifies the chosen action given that system state is X_t following the policy π. Hence, the DTDP seeks a policy that minimizes the expected total fulfilment cost for the platform

$$\min_{\pi\in\Pi} \mathbb{E}\big[\sum_{t\in\mathcal{T}} R(\delta_t^\pi(X_t)) + F(X_0)\big], \tag{4}$$

where $F(X_0) = \sum_{z\in\mathcal{Z}} f(x_{0z})$ and $R(\cdot)$ is the function returns the expected reward paid to crowd drivers under policy π.

The DTDP provides a framework for platforms to deploy a Customized Display (CD) strategy. That is, the platform can vary the size and contents of the subset of the task to display to the incoming crowd drivers depending on the current state (i.e., remaining task list) and arriving drivers' preferences. In Section 5 (computational study), we also experiment with a single-task display policy (SDP), a specialized version of CD in which the platform is allowed to display at most one task at a time.

4. Solution Approach

This section provides methods to solve the DTDP. Let the value function $V_t(X_t)$ denote the expected cost-to-go at period t (t periods to the termination of the selection period) given the state, X_t. Theoretically, one can solve the value function defined below to determine the optimal policy being sought in Equation (4):

$$V_t(X_t) = \sum_{m \in \mathcal{Z}} \lambda_{tm} \left[\min_{D_m \in \mathcal{D}_t} \left\{ \sum_{z \in D_m} p_m(z|D_m) \Big(r_z + V_{t-1}(X_t - e_z) \Big) + p_m(0|D_m) V_{t-1}(X_t) \right\} \right]. \quad (5)$$

When the set D_m is displayed to the driver m, and the driver chooses zone $z \in D_m$, the platform's cost is the reward paid to the driver, r_z, plus the cost of starting the next period with one less task in zone z. Therefore, the first term inside the minimization calculates the expected reward plus the cost-to-go when the set D_m is displayed to the driver m. The second term is the expected cost-to-go when the driver does not choose any of the tasks displayed, which is equal to the probability of driver m not choosing any of the tasks in D_m times the cost of starting the next period with the same set of tasks. Note that the driver index in D_m indicates that the display set depends on the arriving driver type. After obtaining the optimal cost for each driver m, we take an expectation over the arrival probability of each driver to calculate the expected cost of the starting period t with inventory vector X_t.

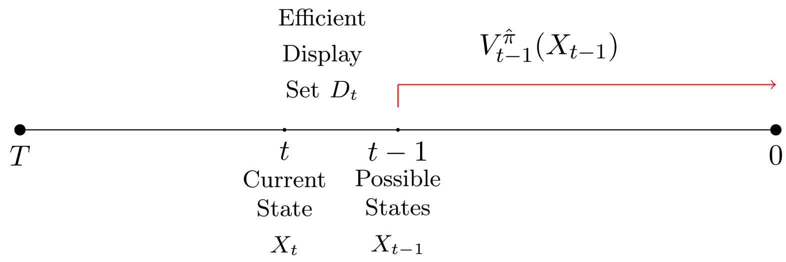
Solving the value function in Equation (5) via backward induction is not possible when the number of zones, Z, exceeds trivially tiny values due to the exponential growth of state and action space in Z. In addition, the DTDP involves two additional novel challenges: (i) uncertainty in drivers' type and their task selection behaviour during the task selection phase, and (ii) the non-linear cost structure at the terminal period for computing the fulfilment cost using the contracted drivers.

To address these challenges, we propose a stochastic look-ahead approach that balances the computational efficiency concerning 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 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. 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. Our method executes two steps at each crowd driver arrival epoch t. In the first step, the associated value function for all possible states that can be reached from the current state X_t is approximated by a preset display policy $\hat{\pi}$, which is used throughout the rest of the selection horizon. In the second step, the efficient display sets are formed considering the approximated value functions. We provide details of these two steps in the remainder of this section.

Figure 3: Solution approximation architecture: VFA & EDS.



4.1. Value Function Approximation (VFA)

In this section, we describe how to approximate the value function for a given state, which involves estimating the total expected fulfilment cost for a given number of tasks and the number of periods to go until the end of the selection period. The approximation is divided into two phases: First, we define a display policy $\hat{\pi}$. Afterwards, we use the probabilistic information of future driver arrivals and the structure of the chosen display policy to compute the expected remaining tasks at the terminal decision epoch to calculate the total expected fulfilment cost under $\hat{\pi}$.

Consider a generic single-task display policy that is independent of states, time, and drivers, and follows a predetermined zone sequence (SDPS). In the SDPS policy, each arriving crowd driver sees a single task from the highest-ranked zone. 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 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 have to be fulfilled by the contracted drivers. As the boundary condition 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 ease of exposition, we first introduce a few preliminaries.

Definition. When a display set consists of a single zone $z \in \mathcal{Z}$ in time t, we define the probability that a task is chosen as $P_{zt} = \sum_{m \in \mathbb{Z}} \lambda_{tm} p_m(z|D = \{z\})$.

This definition shows that for each time epoch in the period of $t \in \mathcal{T}$, the probability that a zone-z task will be chosen follows the Bernoulli distribution with parameter P_{zt} when only zone z is displayed.

Suppose the platform employs an SDPS policy, where the ranking order of the zones to be displayed is predetermined from the current period t to period 0, the end of the selection phase. Then, the SDPS policy tells us which zone will be displayed after $y-1$ tasks are chosen and let us denote this zone by the index $[y]$. Furthermore, let $P(y,t)$ denote the probability of having a total of y tasks chosen at the end of the selection phase or throughout the last t periods.

Proposition. For a fixed SDPS policy, $P(y,t)$ can be calculated using the following recursive equation:

$$P(y,t) = P(y-1,t-1)P_{[y]t} + P(y,t-1)(1-P_{[y+1]t}) \tag{6}$$

With the following boundary conditions:

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

$$P(y,t) = 0 \quad \forall y > t \tag{6b}$$

$$P(1,1) = P_{[1]1} \tag{6c}$$

To explain the intuition behind Equation (6), we first elaborate on the boundary conditions. Equation (6a) states that the probability of having 0 tasks completed in t epochs is possible only when the first task on the SDPS policy does not get chosen in the first t epochs. Equation (6b) implies that since at most one task can be selected in an epoch, the number of selected tasks cannot exceed the number of epochs. Equation (6c) denotes that having one task completed at the end of the first epoch implies that the first task in the SDPS policy is chosen in the first epoch. Then, to complete y tasks in t epochs, one of the following has to happen: i) $y-1$ tasks are completed in $t-1$ epochs, and y^{th} task is completed in epoch t, or ii) y tasks are completed in $t-1$ epochs and $(y+1)^{th}$ task does not get picked in t. This relationship between consecutive time epochs and completed tasks is illustrated in Equation (6).

The proposition is the core component for computing the value function approximation for state X_t, or $V_t^{\hat{\pi}}(X_t)$ using a SDPS policy as a proxy. Now, consider an arbitrary SDPS policy. Let $Y^{SDPS}(i)$ be the vector consisting of the number of tasks chosen in each zone when a total of i tasks are chosen by crowd drivers under the SDPS display policy, and $Y_z^{SDPS}(i)$ refers to the z^{th} index of the vector. Then, the expected total fulfilment cost for when the state is X_t when there are t periods remaining in the selection horizon, which will serve as an estimate for $V_t^{\hat{\pi}}(X_t)$ in our approach, can be calculated as follows.

$$V_t^{\hat{\pi}}(X_t) = \sum_{i=0}^{\min(\sum_z x_{zt}, t)} P(i,t) \left(F(X_t - Y^{SDPS}(i)) + \sum_{z=1}^{Z} r_z \cdot Y_z^{SDPS}(i) \right) \tag{7}$$

To better understand the intuition behind the derivation of Equation (7), let us consider a fixed value of i. After having i tasks completed by crowd shippers, the system terminates at state $X_t - Y^{SDPS}(i)$. Therefore the cost of having all the remaining tasks completed by contract drivers is calculated as $F(X_t - Y^{SDPS}(i))$ and $\sum_{z=1}^{Z} r_z \cdot Y_z^{SDPS}(i)$ represents the total reward paid to the crowd shippers. We then take the expectation over all possible values of i, which can vary between 0 and the minimum of the remaining time periods, and the total number of delivery tasks. The probability of having i tasks completed at the end of t periods, $P(i,t)$ can be calculated using the Proposition, the display sequence of the SDPS policy, and the choice probabilities of the drivers.

Setting an SDPS policy: The accuracy of $V_t^{\hat{\pi}}(X_t)$, and hence the cost benefit of employing a CD policy will depend on how the SDPS policy is set up. Therefore, an effective ranking of zones needs to consider the dynamics of the system, such as the arrival mechanism of the drivers, and the cost benefits of displaying each zone, which depends on the terminal cost structure, arriving drivers' choice probabilities, and the current state. Moreover, the ranking system should be computationally efficient to support on-the-fly decision making.

In our approach, given the remaining time periods t and x_{zt} remaining tasks, we rank each zone z with respect to their expected cost savings, $CS_{zt}(x_{zt})$ under the condition that only zone z is displayed from time t to the end of the selection horizon. To compute the $CS_{zt}(x_{zt})$ we employ the above Proposition to calculate the probability of having $i \in \{0, 1, \dots, x_{zt}\}$ tasks completed by crowd drivers and for each i, we arrive at the cost saving by subtracting the cost of having $x_{zt} - i$ tasks delivered by contract drivers from the cost of having x_{zt} tasks delivered, and adding the reward paid to the drivers for completing i tasks. Thus, the expected cost saving of displaying solely zone z for t periods with x_{zt} remaining tasks depends on two factors: (i) the probability distribution of the remaining tasks in zone z and (ii) the cost saving coming from crowd drivers delivering them instead of the contract driver. Given the dynamics of the system and the terminal cost function, we can pre-compute $CS_{zt}(x)$ for each zone z, remaining time periods t, and number of remaining tasks x. Further details on this procedure can be found in Section 8.2.1 of the Appendices.

Once we have the cost savings pre-calculated, we construct SDPS policies on the fly at each driver's arrival to estimate $V_t^{\hat{\pi}}(X_t)$. Given the state X_t at period t, we rank the zones according to their expected cost benefit per period in descending order, and start adding zones to the SDPS policy one by one. While adding zones to the display set, we keep track of their median completion times, which is the median number of times a zone needs to be displayed in order for all the tasks in that zone to be delivered by crowd drivers. We terminate the procedure if: (i) all remaining zones have negative cost-benefit, or (ii) the median completion time will exceed t when any of the remaining zones with a cost-benefit is added to the display set. More details on this procedure, including the pseudo-code, can be found in Section 8.2.2 of the Appendices.

4.2. Efficient Display Sets (EDS)

In this section, we present efficient display sets following Theorem 1 in Talluri and Van Ryzin (2004) and Lemma 1 in Bernstein et al. (2015) to reduce the action space. These studies examine assortment problems in airline fare products and substitutable inventory optimization in retail such that the decision-maker determines to display which air products or which sets of goods to be retailed among Z options. Similar to their setting, the platform can display $2^Z - 1$ different zone combinations when Z zones with tasks to be delivered. Nevertheless, the theorem and the lemma above show that an optimal display policy exists if the platform shows only Z ordered and nested sets, called *Efficient Display Sets*. In the following, we explain how to form Efficient Display Sets for the DTDP.

Efficient display sets are formed systemically by calculating the marginal cost contribution of having an additional delivery task in a specific zone. That is, we first define the marginal expected cost generated by the x_{zt}^{th} task in zone z at period t, *i.e.*, $\triangle_{t-1}^{z}(X_t) = V_{t-1}(X_t) - V_{t-1}(X_t - e_z)$. In other words, the expected cost difference if x_{zt}^{th} is delegated to a crowd driver or not. Using this definition and the fact that $P_m(0|D_m) = 1 - \sum_{z \in D_m} P_m(z|D_m)$, one can rewrite the optimality equation in Eq (5) as follows:

$$V_t(X_t) = \sum_{m \in \mathcal{Z}} \lambda_{mt} \left[\min_{D_m \subset \mathcal{D}_t} \left\{ \sum_{z \in D_m} P_m(z|D_m) \Big(r_z + V_{t-1}(X_t - e_z) - V_{t-1}(X_t) \Big) + V_{t-1}(X_t) \right\} \right]$$

$$= \sum_{m \in \mathcal{Z}} \lambda_{mt} \left[\min_{D_m \subset \mathcal{D}_t} \left\{ \sum_{z \in D_m} P_m(z|D_m) \Big(r_z - \triangle_{t-1}^{z}(X_t) \Big) \right\} \right] + V_{t-1}(X_t). \quad (8)$$

Let $r_t^z(X_t) = r_z - \triangle_{t-1}^{z}(X_t)$ denote the effective marginal expected cost of a task in zone z at period t, given that there x_{zt} tasks remaining in the zone. Note that $r_t^z(X_t)$ can be interpreted as the expected cost saving if a crowd driver chooses a task to deliver into zone z. Now, consider an ascending ordering of zones in effective marginal expected cost such that $r_t^{[1]}(X_t) \leq r_t^{[2]}(X_t) \leq \ldots \leq r_t^{[Z]}(X_t)$, where $[s]$ returns the zone index in order of s^{th}. Then, we can construct the efficient display sets starting with a set of a single zone consisting of only zone [1]. To construct the efficient display set sized $s \leq Z$, we consider only the ordered set in the form of $\{[1], [2], \ldots, [s]\}$. As a result, the platform will choose among at most Z display sets; each includes a unique number of zones.

It is important to note that displaying solely Efficient Display Sets, introduced in Talluri and Van Ryzin (2004), produces optimal action when the boundary condition is a linear function. Nevertheless, the DTDP generalizes the boundary conditions; therefore, the optimality of the efficient display sets does not hold for the DTDP. However, we show the effectiveness of the Efficient Display sets with numerical experiments in Appendices8.3.

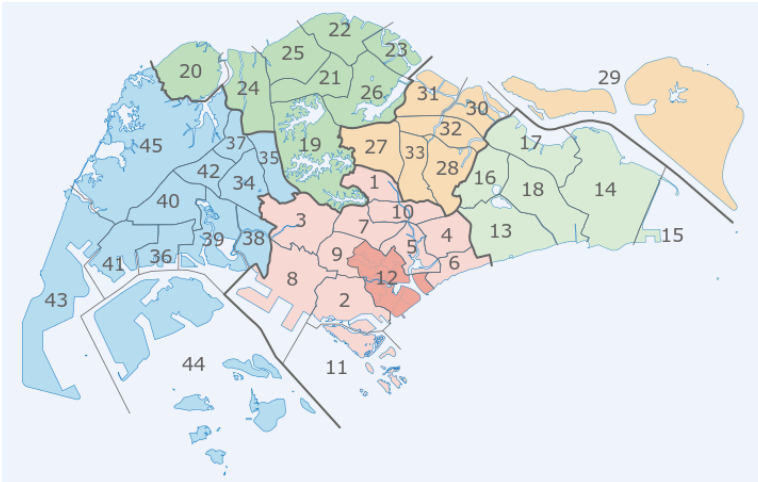
5. Experimental Setup

In this section, we describe our experimental setup to validate our solution approach outlined in Section 4, and explain how we generate the instances, including crowd drivers and delivery requests. We also present benchmark task-display policies to test the proposed customized dynamic task-display policy. Furthermore, we present the key performance metrics that quantify the performances of display policies, and we discuss the results in depth.

5.1. Network for delivery requests and crowd drivers

We design experiments that use Singapore as a test bed. We utilize Singapore's mutually exclusive planning regions as delivery zones, illustrated in Figure 4 (Wikipedia, 2023). The single depot is located in Zone 13, which is consistent with the setup of our industry collaborator. We remove zones with very low population densities (islands, natural reserves, industrial regions, etc.) and split dense zones into smaller ones to balance the average delivery tasks across zones. As a result, we proceed with our computational experiments with 36 zones. To calculate the distance between zones, we employ the Google Maps API using the coordinates of the centroid of each zone. Furthermore, we set the number of customer orders and the expected number of drivers to be 500.

Figure 4: Planning zones in Singapore.



While constructing the instances, we use the population of each zone and its distance from the depot as the primary metrics to determine the parameters for each zone. A task is assigned randomly to zone z proportionately to the population of zone z.

Each crowd driver is associated with a zone where this driver prefers to end his/her delivery trip. Similar to delivery tasks, the arrival of crowd drivers who favour a particular zone as the ending zone is proportional to that zone's population density.

5.2. Parameters Setup

It is essential to set up parameters realistically to derive insightful analysis. To account for contract drivers' expenses, we consider a cost structure that is associated with the time needed to deliver the number of parcels for a given time period. In Singapore, the working time arrangement of a contract worker varies between two and six hours, averaging four hours. The current rate of renting a van with a driver is S$56 per hour, (LCHLogistics, 2023). Therefore, we take $\Gamma = 56$. We assume the service time per delivery, β_s, to be five minutes per order (Dalla Chiara et al., 2022). To determine β_z for each zone, we use the travel time from the depot to zone z and assume a loading time of 10 minutes. Finally, we set $\beta_{zt} = 0.859$ for each zone (Franceschetti et al., 2017).

We consider that a crowd driver's zone selection preferences can be explained through their true utility function, as explained in Equations (2) and (3) in Section 3. We consider that a crowd driver receives a zone-independent flat rate of S$7.5 to deliver a parcel in parallel to the average ride-hailing income of drivers per ride (Agarwal et al., 2022). The cost associated with detour inconvenience is calculated based on the distance; *i.e.*, for each kilometre of more detour for delivery, the driver's perceived utility decreases by β_{detour} = S$0.20, which is consistent with the fuel cost per kilometre of travel.

Table 1: Parameters used in the Singapore case study.

Number of zones	Z	36
Number of customer orders	O	500
Number of expected crowd drivers	K	500
No-choice utility for each $z \in \mathcal{Z}$	$u_z(0)$	0
Reward	$r_z = r, z \in Z$	S$7.5
Cost of non-completed tasks	$f_z(x), z \in Z$	$56[\beta_z + \beta_s x + \beta_{tz}\sqrt{A_z x}]$
MNL normalization parameter	α	1

A driver chooses a task (zone) among the ones displayed on his/her mobile App; they also have the option of not selecting any of the displayed zones and simply walking away. We take the no-choice utility, $u_z(0)$, of each arriving driver and for each zone as 0. Table 1 summarizes the values of all parameters used in our case study.

5.3. Benchmark Policies

In this section, we present two main benchmark policies, namely Display-All and Clearance-L, to compare the performance of the customized display (CD) policy described in Section 4 and one special version of the CD policy, Single Best Display (SBD).

- **Display-All.** In this policy, the platform displays tasks from all zones with 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 simulation.
- **Clearance-L** In this policy, at most $L \leq Z$ number of zones will be displayed to the crowd drivers. This policy chooses L zones according to the increasing number of remaining tasks, starting with the zone with the fewest tasks.
- **Single best display (SBD).** Single best display is the restricted version of the CD policy such that the platform follows the same logic, but it only shows at most one task to each arriving crowd driver. The driver either takes the displayed option and delivers it or leaves the system.

5.4. Key Performance Indicators

We define three key performance indicators (KPIs) for evaluating competing policies.

- *Cost saving.* The cost savings are computed as percentages, reflecting costs saved compared to the case where contracted drivers handle all deliveries.
- *Matched tasks.* The ratio of delivery tasks served by the crowd drivers.
- *Reward ratio.* The proportion of the total reward paid to the crowd drivers compared to the total fulfilment cost.

6. Computational Study

In this section, we computationally compare the customized task display strategy with the Single Best Display, Display-All, and Clearance-L to quantify its benefit. We then test the choice-based crowdshipping benefits in different settings, such as varying crowd drivers' task selection pickiness.

6.1. Results and Discussion

In this section, we compare our Customized Display (CD) method to benchmark policies introduced in Section 5.3. We ran 50 instances varying in the spread of 500 tasks across zones and the number of arriving drivers. For each of these runs, we calculate the total fulfilment cost, the total number of requests served by crowd drivers and the total reward amount used to compensate the crowd drivers for the CD policy and all benchmark policies. We then compare each policy according to the KPIs introduced in Section 5.4. Table 2 summarizes comparisons by presenting the maximum, the average, and the minimum of each metric for each benchmark policy and proposed CD policy.

The results show that choice-based crowdshipping reaches the highest cost saving with CD policy, an average of 16.9%, over all runs. These results show that CD policy's cost saving is better

Table 2: Choice-based crowdshipping experiments in Singapore (50 replications).

KPI		Customized Display	SBD	Display All	Cle-1	Cle-2	Cle-3
	max	18.4%	18.0%	11.4%	15.4%	16.2%	16.1%
Cost Saving	average	16.9%	16.6%	9.0%	12.2%	14.0%	13.9%
	min	15.3%	14.8%	7.2%	10.0%	11.2%	11.8%
	max	76.1%	72.8%	92.7%	52.1%	64.5%	76.5%
Matched Tasks	average	63.6%	61.9%	91.2%	37.5%	56.0%	67.0%
	min	53.9%	53.1%	89.5%	27.1%	46.7%	60.4%
	max	75.0%	71.2%	85.1%	48.2%	61.1%	71.8%
Reward Ratio	average	63.2%	61.2%	82.7%	35.5%	54.3%	64.3%
	min	53.6%	53.0%	80.5%	25.0%	44.6%	57.4%

performing than Display-All, Clearance-1, 2, and 3, which achieve average cost savings of 9%, 12.2%, 14.0%, and 13.9%, respectively. (See Appendices for extended results from the Clearance-L policy.) A restricted version of CD, allowing the display of only a single best zone (SBD) outperforms the benchmark policies but brings less cost saving than the CD, an average of 16.6%. Furthermore, we can observe that the cost-saving performances of CD and SBD are more consistent than others as the variance of cost savings for these two policies is lower than other benchmarks across all runs. These results indicate that anticipating the behaviours of arriving crowd drivers, accounting for future drivers, and considering marginal cost savings against joint contracted driver delivery are valuable in the choice-based next-day delivery setting. Furthermore, having the flexibility of displaying multiple options to drivers brings extra cost savings.

When we look at the Display-All and Clearance policies, we see that Display-All performs worse in terms of cost saving, even though it outperforms all other policies in matched tasks and reward ratio. This is primarily caused by the cost structure of contracted drivers. Recall that the majority of savings are achieved if hiring additional contracted drivers is avoided, which is associated with clearing tasks from as many zones as possible. Display-all does not highlight any zone over another one. Consequently, this policy increases the chance of ending up with more zones with positive remaining tasks at the end of the selection period. Clearance policies, on the other hand, concentrate only on the zone(s) with fewer tasks. As a result, they prioritize clearing as many zones as possible without considering the drivers' choice behaviours. Nevertheless, results strengthen the insight that giving more than one option to drivers is beneficial.

One may expect that the Display-All policy is the most crowd-driver-friendly because an arriving driver sees all available task options under this policy. The results in Table 2 justify this logic as the number of matched tasks with crowd drivers (91%) and the total reward paid to crowd drivers (85.1%) is highest with the Display-All policy. Nevertheless, it is crucial to notice that maximizing tasks delivered by crowd drivers does not necessarily maximize cost savings; mainly because savings also depend on the cost structure of contract driver recruitment. As a result, Display-All achieves substantially less cost savings (7.9% less than CD and 5% less than Cle-2) than other policies, as the primary goal is to maximize the crowd drivers' participation.

6.2. Sensitivity to the Reward and Driver/Task Ratio

This section examines how the varying reward amounts paid to a crowd driver and the average number of arriving crowd drivers per task influence the performance of display policies. We decrease and increase the base reward amount S$7.5 to S$7 and S$8. More rewards paid means that drivers perceive higher utility for making a delivery, increasing the likelihood of choosing one of the displayed zones, and fewer rewards paid means the opposite. Furthermore, we also change the expected number of crowd drivers arriving per delivery task. The case in which 0.9 driver arrives per task represents the scarcity of crowd drivers, whereas the case in which 1.1 drivers arrive per task represents the extra crowd driver resource availability. Table 3 presents the percentage of cost savings and the ratio of tasks served by crowd drivers with varying reward, driver-to-task ratio, and display policies.

Table 3: Impact of varying reward to crowd and driver/task ratio.

		Cost Saving Driver/Task Ratio			Matched Tasks Driver/Task Ratio		
Reward	Policy	0.9	1	1.1	0.9	1	1.1
7	CD	18.5%	19.5%	20.3%	66.3%	71.6%	75.3%
	Display All	8.4%	13.0%	15.9%	84.0%	89.6%	93.5%
	Clearance-1	11.9%	12.7%	13.4%	27.6%	30.6%	33.5%
	Clearance-2	14.7%	15.5%	16.4%	42.9%	47.3%	51.8%
	Clearance-3	15.8%	16.6%	17.4%	53.7%	59.5%	65.1%
7.5	CD	16.3%	16.9%	17.5%	59.9%	63.7%	66.6%
	Display All	4.7%	9.1%	12.2%	85.3%	91.1%	95.3%
	Clearance-1	11.7%	12.3%	12.8%	34.1%	37.2%	40.4%
	Clearance-2	13.4%	14.0%	14.5%	50.4%	55.8%	61.2%
	Clearance-3	13.5%	13.9%	14.5%	61.0%	67.4%	74.0%
8	CD	14.1%	14.5%	14.8%	54.3%	57.7%	60.4%
	Display All	0.7%	4.9%	8.2%	86.3%	92.4%	96.8%
	Clearance-1	10.6%	11.1%	11.5%	40.1%	44.3%	48.8%
	Clearance-2	11.1%	11.3%	11.4%	57.6%	63.9%	70.2%
	Clearance-3	10.3%	10.4%	10.2%	67.3%	74.7%	82.4%

Table 3 shows two general patterns irrespective of the display policies: (i) decreasing the reward amount paid to the crowd drivers and (ii) increasing the expected arriving crowd driver per task increases the average cost saving from the choice-based crowdshipping in the next day delivery service.

On the other hand, the impact of changing rewards on the relative cost savings of different display policies is not uniform. Notably, the Display-All policy is the most sensitive to the reward amount. The base average cost savings of 9.1% increases to 13% if the reward amount drops S$7 and decreases to 4.9% if the reward is S$8. This is an anticipated result as the Display-All policy gives full autonomy to drivers. In other words, crowd drivers' willingness to deliver tasks depends on the attractiveness of the reward. Even though the number of matched tasks increases with more reward per delivered task, cost saving is not aligned with it. This result is because crowd deliveries become less advantageous against contract driver fulfilment.

However, the impact of changing rewards on the CD and Clearance policies is distinct compared to the effects on the Display-All policy. It is key to recognize that CD policy aims to maximize cost savings; hence, increasing the reward paid per task makes delivering tasks with contracted drivers relatively cheaper. Consequently, increased reward decreases the ratio of tasks served by crowd drivers for these policies. Furthermore, we observe that the ranking of Clearance-L policies varies with reward. For a lower reward, as expected, the Clearance-L policy with a larger menu size leads to greater cost savings.

The impact of the driver-task ratio shows that more drivers per task increase the cost savings and the tasks delivered by crowd drivers. However, the display-all policy is again the most sensitive to this parameter, similar to changing the reward amount.

The joint impact of the reward and driver task ratio variations can also be seen in Table 3. Along the lines of the isolated impacts of reward and driver/task ratio, we observe the policy Display-All magnifies the negative impacts of higher rewards and scarce driver cases. In this scenario, the choice-based cost savings approach zero if the Display-All is chosen. At the same time, they consistently provide 14% and 10% cost minimization, respectively. This result is another manifestation of the shortcomings of the decentralized Display-All policy in the next-day crowdshipping setting, particularly when the reward and expected crowd driver relations are not forecasted accurately.

Note that due to the Markovian property of the MDP, our solution methodology implicitly assumes that an arriving driver makes a choice before the next driver's arrival. In a real-world setting, this may not always be true. To quantify the impact of this assumption on the solution quality, we conduct a simulation study by assuming the driver's decision can be delayed by a certain probability and periods. The results of this experiment show that the variance in the savings is negligible. We share the details of this study in Section 8.6 of the Appendices.

6.3. Sensitivity to Contract Drivers' Cost Parameters

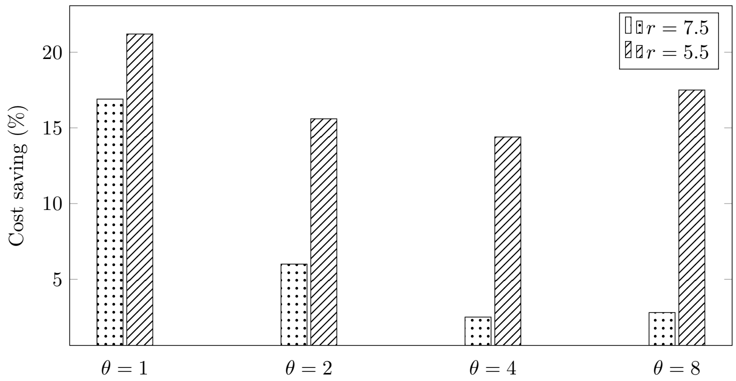
This section shows the impact of contract drivers' cost parameters on the crowdshipping cost savings. In other words, we test the influence of the cost components of serving a zone by a contract driver in Equation (1):

$$c_z(x) = \Gamma \cdot \left[\beta_z \theta^{-1} + \beta_s x + \beta_{tz} \sqrt{\theta A_z x}\,\right],$$

where θ represents the number of zones a contract driver is responsible for delivery. We vary the area coverage via θ such that a contract driver can be responsible for two, four, and eight zones. For example in the case of $\theta = 4$, or a contract driver can serve up to four zones, the total cost will be different (less) by S$$(42\beta_z - \beta_l\sqrt{A_z x})$ for zone z.

Even though this cost perturbation underestimates the total fulfilment cost without crowdshipping, it is expected to estimate the contracted driver's expenses after the crowd drivers' selection is over, and a few tasks remain to be delivered. The cost-saving percentages of the CD policy with different θ, under base rewards S$7.5 and S$5.5 are presented in Figure 5.

Figure 5: Impact of varying contract drivers' cost parameters.



In line with the analysis of varying reward amounts to crowd drivers in Section 6.2, Figure 5 shows that a relative decrease in the cost of contracted drivers without an adjustment in the reward amount leads to a setting where crowdshipping becomes less cost-effective. The cost savings of crowdshipping with CD policy shrinks to as low as 2.5%. However, with an adjusted reward of S$5.5, we observe the cost savings remain relatively stable to the changes in the contract driver cost parameters.

Figure 5 also illustrates the impact of the cost approximation on contract drivers' expense on the total fulfilment cost remains limited if the relative attractiveness of delivering tasks by crowd drivers to the contract drivers does not alter drastically.

6.4. Understanding the Customized Display Policy and Crowd Drivers' Choices

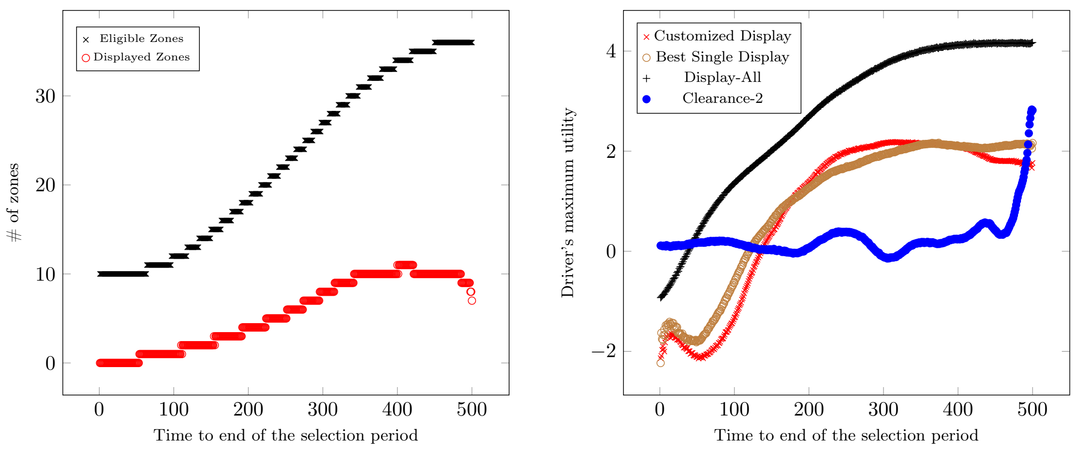
In this section, we explore the choice-based crowdshipping setting from crowd drivers' perspective and demonstrate why CD policy outperforms other policies in terms of total cost savings. Firstly, we zoom in on the CD policy and present the results showing the varying displayed task sizes throughout the selection period. Later, we look at the utility of crowd drivers for the displayed tasks.

Figure 6a presents the evolution of the median number of displayed zones to arriving drivers and the median number of eligible zones with CD policy. We see that the displayed number of

Figure 6: Impact of the customized display policy on drivers' choice.

(a) The number of eligible and displayed zone numbers.

(b) Driver's maximum utility



zones and the number of zones with a positive task number are strongly correlated. In other words, CD policy displays more options to an arriving driver when there are more eligible zones. This pattern of decreasing number of displayed zones exhibits that CD policy seeks the balance between decreasing the risk of crowd driver attrition by displaying multiple options and concentrating on zones with higher cost-saving potentials.

In cases where fulfilling tasks in some zones is more cost-efficient with contracted drivers than delegating to crowd drivers, the platform desires the task display policy to detect and act on this situation. Figure 6a also shows that CD policy has this feature, and it may opt not to display any tasks to an arriving crowd driver. For the last 50 periods before the selection period is over (periods 0-50), the CD policy considers the risks and the expected cost savings, and decides not to display tasks to arriving drivers.

Figure 6b depicts how display policies are perceived by crowd drivers throughout the selection period. We use the metric "driver's maximum utility" to measure the driver's "happiness" through what tasks are shown to her. For a given time period t in the selection horizon and displayed tasks (zones) D_t, the driver's maximum utility equals the utility of choosing a task from the driver type m's most preferred zone within the set D_t; *i.e.*, $\max\{u_m(z), z \in D_t\}$ from Equation (2). Recall that in the Display-All policy, the number of zones displayed always equals the number of zones with a positive number of tasks. In the Clearance-L policy, on the other hand, at most k zones are displayed. The CD policy, on the other hand, determines the display set by trying to appeal to individual drivers.

In Figure 6b, we see that the Display-All policy consistently produces display sets maximizing the driver's maximum utility except for the last few periods. This is an expected outcome as the number of displayed zones increases, the possibility of having zones that are favourable by the crowd drivers in the display set increases. On the other hand, the second best-performing policy w.r.t. the driver's utility changes with time or, more precisely, with the remaining number of delivery tasks. At the beginning of the selection period, when the number of tasks is higher, the customized and the best single-display policies display zone(s) relatively similar according to the driver's utility metric, and they perform better than Clearance-2. However, at the end of the selection period, drivers' maximum utility remained higher with the Clearance-2 policy. Furthermore, we also see that in all policies except Clearance-2, the driver's maximum utility gradually falls below zero as fewer tasks remain to show at the end of the selection horizon.

Figure 6b also helps us understand why CD policy outperforms Single Best Display, *i.e.*, having the flexibility of displaying multiple zones performs better than displaying at most a single zone. It is noticeable that for these two policies, a driver's maximum utility is similar throughout the selection period. Nevertheless, drivers' choices also depend on unobservable factors. From the MNL model, we know that the probability of a driver not choosing a task will be lower when more diverse task options are displayed to her. Hence, CD policy enables the steering of drivers' behaviours more efficiently by exploiting this feature of the choice model.

6.5. Crowd Drivers' Task Selection Predictability and Pickiness: The Impact of Parameter α and the No-Choice Utility

This section explores two types of crowd drivers' selection behaviours: (i) we first investigate settings in which crowd drivers adhere more or less to their observable utility behaviours when they select from the displayed set, or in short, crowd driver's *choice predictability*, (ii) we also explore crowd drivers' willingness to deliver tasks, *i.e.*, *driver's pickiness* in participation. We control drivers' choice predictability by setting α. With all else being equal, a higher α represents a higher predictability as increasing (decreasing) α will increase (decrease) the likelihood of a driver choosing zones with positive (negative) utility, with the steepest increase (decrease) happening to the zone with the most positive (negative) utility. Moreover, as $\alpha \to \infty$, a driver will deterministically choose the option with the highest positive utility, or leave the system without choosing any task if all options yield a negative utility.

We control drivers' pickiness to participate by setting the no-choice utility $u_m(0)$. As $u_m(0)$ increases, the likelihood of leaving the platform without choosing any of the displayed zones increases for all display sets.

Figure 7 presents choice-based cost-saving results for additional α values of 0.25, 0.5, 2, and 4, which represent the cases where drivers are 2 or 4 times less/more predictable than the baseline cases explored in earlier experiments. Figure 9, on the other hand, shows the cost savings when crowd drivers' no-choice utility (utility of not participating) varies from the default S\$0 to one, two, three, and four Singapore dollars.

Figure 7: Impact of drivers' task selection predictability on cost saving.

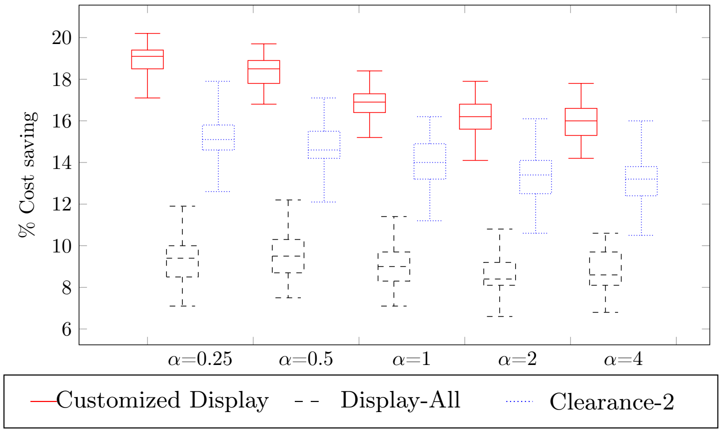


Figure 7 shows an overall trend that the cost saving from the choice-based crowdshipping system decreases as drivers become more predictable. More precisely, when drivers have a strong preferential perception between delivering tasks into different zones (α is higher), the system-wide cost-saving decreases, particularly for CD and Clearance policies. For example, the median cost saving is 19.1% when α is 0.25, but 16% when α is 4 with the CD policy. As expected, Display-All is insensitive to drivers' choice predictability. The relative ranking of cost saving remains the same for all α; CD performs the best, followed by Clearance-2 and Display-All.

Figure 8: The impact of α on customized display policy.

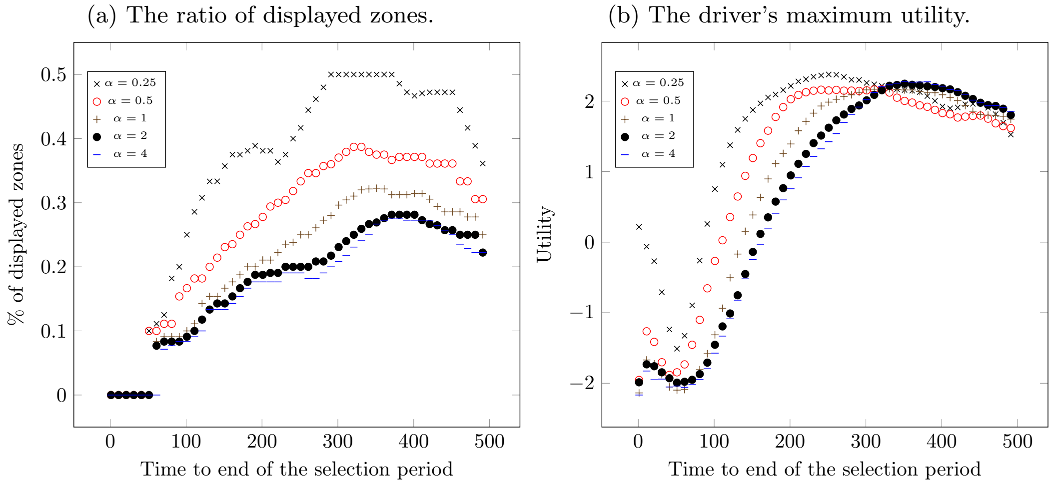


Figure 8 zooms in on how CD policy adapts its display set size to changes in drivers' behaviours. Figure 8a exhibits the proportion of zones displayed out of the zones with remaining tasks throughout the selection period. Note that $\alpha = 1$ represents the previous experiment. This figure first reveals that the driver's choice behaviour influences CD policy actions. For relatively less predictable drivers (for cases where $\alpha < 1$), the CD policy displays more options than the opposite setting. For example, CD displays half of the eligible zones when $\alpha = 0.25$ at the initial periods, but this ratio drops to a quarter when $\alpha = 4$. In other words, CD policy utilizes the driver's lack of preferences and steers them easily by showing them more options, which increases the participation rate.

Figure 8b presents the shifts of the driver's maximum utility for varying α values. Note that the drivers' observed utility remains the same, but the choice behaviours between the displayed options vary when α takes different values. In this figure, we observe that the driver's perceived utility is primarily associated with the display menu size. That is, the CD policy adapts its display menu sizes in negative correlation to α. As a result of this adaption, the maximum utility of the incoming driver stays relatively higher when drivers are less predictable (another way to see this is that drivers are more flexible with smaller α, thus allowing us to show them more options profitably).

Figure 9: Impact of drivers' task selection pickiness on cost saving.

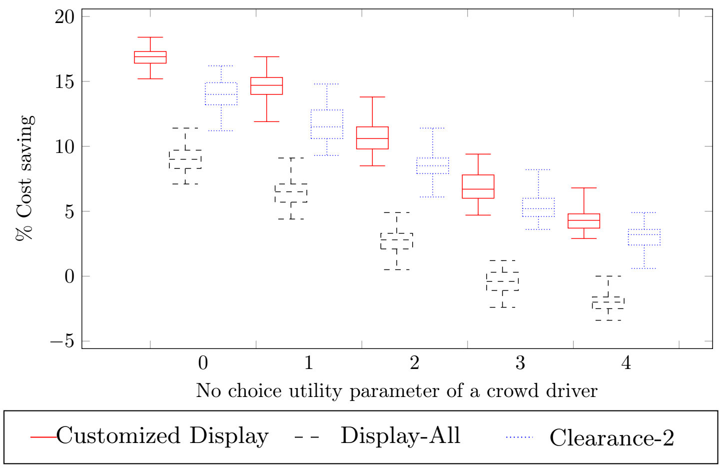


Figure 9 shows that the settings with pickier drivers, or the drivers with higher no-choice utility, reduce the cost-saving potential of choice-based crowdshipping regardless of the chosen display policy. Unlike the driver's predictability, the decentralized Display-All policy is influenced significantly by the picky drivers, as the cost savings reach negative values (choice-based crowdshipping increases the fulfilment cost). This result stems from the cherry-picking behaviours of picky drivers in a decentralized setting. In other words, displaying all options will increase participation when drivers are more selective about participating in crowdshipping. Nevertheless, in this scenario, the participating crowd drivers choose the tasks that suit their criteria, which leads to a large spread of remaining tasks to be fulfilled by contracted drivers. On the other hand, CD can regulate unfavourable selection by limiting the displayed tasks.

7. Concluding Remarks

In this work, we introduce and study the Dynamic Task Display Problem in a choice-based crowdshipping setting, where an online platform delegates delivery tasks to dynamically arriving crowd drivers. We investigate a display mechanism between the two extremes of: (a) purely centralized, which suggests only a single task to the driver, and (b) purely decentralized, which displays all delivery tasks and lets drivers choose freely. We propose a mechanism with *customized display*, where individually arriving workers are presented with a collection of carefully chosen tasks to minimize the total fulfilment cost. We overcome the curse of dimensionality stemming from the Markov Decision Process formulation by devising a stochastic look-ahead policy. We experiment the choice-based crowdshipping and various display policies on Singapore-inspired instances.

Our numerical experiments demonstrate that choice-based crowdshipping can achieve significant cost savings for the overnight delivery problem. This is achieved via carefully curated set of tasks to be shown to different drivers. The highest cost savings can be achieved by the proposed customized display policy, which considers the drivers' choice behaviour, the contracted delivery expenses, and future drivers. We also show that when drivers become less picky while selecting tasks, choice-based crowdshipping will dominate, and the customized display policy is the most effective policy to leverage this feature. It is important to note that our numerical experiments present conservative savings from crowdshipping due to the restriction of single task selection per crowd driver. Further studies can explore the impact of multi-selection or the impact of displaying bundled tasks.

Our findings and insights have practical implications for the broader crowdshipping and online task assignment/selection platform design. In particular, for scenarios where instantaneous matching of tasks and workers is not required, our work demonstrates the benefits of tailoring the list of tasks displayed to each worker by considering the spatio-temporal features of future worker arrivals. Our work also suggests the importance of designing an adaptive platform mechanism to balance the desire for worker autonomy and assignment efficiency, especially for scenarios where task fulfilment is achieved through a combination of an elastic crowdsourced worker pool and contracted resources (delivery workers, trucks, etc.).

References

Agarwal, S., Charoenwong, B., Cheng, S.F., Keppo, J., 2022. The impact of ride-hail surge factors on taxi bookings. Transportation Research Part C: Emerging Technologies 136, 103508.

Alnaggar, A., Gzara, F., Bookbinder, J.H., 2021. Crowdsourced delivery: A review of platforms and academic literature. Omega 98, 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 56, 1304–1326.

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.

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

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.

Daganzo, C.F., 2005. Logistics systems analysis. Springer Science & Business Media.

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

Dalla Chiara, G., Krutein, K.F., Ranjbari, A., Goodchild, A., 2022. Providing curb availability information to delivery drivers reduces cruising for parking. Scientific reports 12, 19355.

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

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.

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

Fai, L.K., 2019. Thinking outside the box: Innovative solutions for the package delivery industry: NHK World-Japan News. URL: https://www3.nhk.or.jp/nhkworld/en/news/backstories/293/.

Franceschetti, A., Jabali, O., Laporte, G., 2017. Continuous approximation models in freight distribution management. Top 25, 413–433.

Fujitsu, A*STAR, Singapore Management University, UrbanFox, 2018. Fujitsu, SMU, A*STAR, and UrbanFox launch field trial to enhance crowdsourced delivery in Singapore. URL: https://www.fujitsu.com/global/about/resources/news/press-releases/2018/0913-02.html.

Gdowska, K., Viana, A., Pedroso, J.P., 2018. Stochastic last-mile delivery with crowdshipping. Transportation research procedia 30, 90–100.

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.

Haferkamp, J., Ulmer, M.W., Ehmke, J.F., 2024. Heatmap-based decision support for repositioning in ride-sharing systems. Transportation Science 58, 110–130.

Holland, C., Levis, J., Nuggehalli, R., Santilli, B., Winters, J., 2017. UPS optimizes delivery routes. Interfaces 47, 8–23.

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.

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.

Karabulut, E., Gholizadeh, F., Akhavan-Tabatabaei, R., 2022. The value of adaptive menu sizes in peer-to-peer platforms. Transportation Research Part C: Emerging Technologies 145, 103948.

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

LCHLogistics, 2023. URL: https://lchlogistics.com/lorry-rental/.

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.

Macrina, G., Pugliese, L.D.P., Guerriero, F., Laporte, G., 2020. Crowd-shipping with time windows and transshipment nodes. Computers & Operations Research 113, 104806.

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

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.

Mousavi, K., Bodur, M., Cevik, M., Roorda, M.J., 2021. Approximate dynamic programming for crowd-shipping with in-store customers.

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

NHK, 2018. Thinking outside the box: Innovative solutions for the package delivery industry. `https://web.archive.org/web/20190617201918/https://www3.nhk.or.jp/nhkworld/en/news/backstories/293/`. [Online; accessed 1-June-2024].

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.

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

Torres, F., Gendreau, M., Rei, W., 2022. Vehicle routing with stochastic supply of crowd vehicles and time windows. Transportation Science 56, 631–653.

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

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

Wikipedia, 2023. Planning Areas of Singapore — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Planning\%20Areas\%20of\%20Singapore&oldid=1146114363. [Online; accessed 16-June-2023].

Xu, L., Xu, Z., Xu, D., 2013. Exact and approximation algorithms for the min–max k-traveling salesmen problem on a tree. European Journal of Operational Research 227, 284–292.

Yang, Y., Umboh, S.W., Ramezani, M., 2024. Freelance drivers with a decline choice: Dispatch menus in on-demand mobility services for assortment optimization. Transportation Research Part B: Methodological 190, 103082.

Yıldız, B., 2021. Express package routing problem with occasional couriers. Transportation Research Part C: Emerging Technologies 123, 102994.

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

8. Appendices

8.1. Notation

For easier reference, we summarize the notations used in the paper in Table 4.

Table 4: Notations for DTDP.

Notation	Description
	Sets
$\mathcal{T}$	Periods/Slots, $t = T, T-1, , \cdots, 1, 0$
$\mathcal{Z}$	Set of zones/clusters, $z = 1, 2\cdots, Z$
$\mathcal{O}$	Set of delivery tasks, $o = 1, 2, \cdots, O$
$\mathcal{O}_z$	Set of delivery tasks in zone $z \in \mathcal{Z}$
	Parameters associated with geography
$tt(i, j)$	Distance between zone i and j
A_z	Area of zone z
	Contract drivers
Γ	Hourly rate of hiring a contract driver
β_z	Duration of travelling to zone z from the depot in hour
β_s	Duration of serving a single order (independent of a zone)
β_{tz}	Coefficient accounting for travel time between two orders in a specific zone
$f_z(x)$	Cost of fulfilling x tasks in zone z with a contract driver
	Crowd drivers
λ_{tm}	Arrival rate of driver type m at time t
r_z	Reward for the crowd driver delivering a task to zone z
$u_m(z)$	utility of delivering order in zone z for m type driver
β_{detour}	Coefficient accounting for disutility of detour
	Modeling
X_t	State at time t
$\mathcal{D}_t$	Collection of feasible display sets at period t
z	Index shows order location in zone $z \in \mathcal{Z}$
m	Index shows a driver's destination zone $m \in \mathcal{Z}$
π	Decision policy
α	Normalizing parameter for the choice model

8.2. SDPS Policy Generation

In this section, we detail on the calculation of the cost savings for each zone under SDPS, and present the pseudo-code of the algorithm to generate the SDPS policy on the fly while implementing the VFA.

8.2.1. Expected Cost Saving Calculation

To evaluate the SDPS Policy on-the-fly efficiently, we calculate the expected cost savings for each possible value of remaining time epoch, and remaining number of tasks for each zone, assuming that only that zone is shown throughout the remaining of the selection period

To demonstrate, suppose a zone z has x_{Tz} number of tasks at the beginning of selection period that has length T. Fix $t \in \{1, .., T\}$ and $x \in \{1, .., x_{Tz}\}$. Then, we calculate the expected benefit of solely displaying zone z, and displaying nothing afterwards, given that there are t time epochs remaining, and x tasks waiting to be delivered at zone z as follows:

$$CS_{zt}(x) = \sum_{i=0}^{x} P(i,t)(c_z(x) - c_z(x-i) + r_z i)$$

In the above expression, we take expectation over $i \in \{0, ..., x\}$, the number of tasks that can be completed until the end of the selection period, given that there are x deliveries remaining in zone z. $P(i,t)$ represents the probability of having i deliveries completed by crowd drivers in t periods, and can be calculated using Proposition 1. When i deliveries are completed by the crowd drivers, there are $x-i$ deliveries remaining in zone z and therefore, $f_z(x) - f_z(x-i) + r_z i$ represents the change in the terminal cost value for zone z plus the reward paid to the crowd shippers to complete i deliveries. If $CS_{zt}(x) > 0$, we deem zone z beneficial to display when there are x deliveries remaining in the zone, and there are t epochs remaining.

Given the initial state $X_T = \{x_{T1}, ...x_{TZ}\}$, $CS_{zt}(x)$ can be precomputed for all $z \in \mathcal{Z}, t \in \mathcal{T}$ and $x \in \{1, .., x_{Tz}\}$. For a fixed z, x and t, calculating $CS_{zt}(x)$ requires a single loop over $i \in \{0, ..., x\}$ and is $\mathcal{O}(\bar{X})$ where $\bar{X}$ is the largest element in X_T. We repeat this procedure for all zones, time periods and possible values of remaining tasks, and therefore the expected cost saving calculation for a problem instance is $\mathcal{O}(ZT\bar{X}^2)$.

8.2.2. SDPS Policy Generation Algorithm

The algorithm takes a state (X_t), the number of remaining time epochs (t), a list of median completion time of one task in each zone when that zone is displayed exclusively (M), and a 3-D array that contains the cost savings for each zone, remaining time epochs and inventory level ($CS_{zt}(.)$) as an input and returns a list of zones ($SDPS$).

We start the procedure by creating an empty list, $SDPS$, which will contain an ordered list of zones to be displayed in the SDPS policy, given the current state of the system. Then, we determine the list of candidate zones, $\mathcal{Z}_{cand}$, by identifying zones that with positive number of remaining tasks, positive cost savings and a median completion time greater than the number of remaining periods. Then, we identify the zone with the maximum savings per period until completion, z^*, and add it to the SDPS list, and update t by deducting the median completion time of zone z^* under the

Algorithm 1 SDPS Policy Generation

Input $X_t, t, M, CS_{zt}(.)$
$SDPS \leftarrow \emptyset$
$\mathcal{Z}_{cand} \leftarrow \{z : X_t(z) > 0, CS_{zt}(X_t(z)) > 0, X_t(z)M(z) \geq t\}$
while $\mathcal{Z}_{cand} \neq \emptyset$ **do**
    $z^* \leftarrow \arg\max_{z \in \mathcal{Z}_{cand}} \{CS_{zt}(z)/(X_t(z)M(z))\}$
    $SDPS \leftarrow SDPS \cup \{z^*\}$
    $t \leftarrow t - X_t(z^*)M(z^*)$
    $\mathcal{Z}_{cand} \leftarrow \{z : X_t(z) > 0, z \notin SDPS, CS_{zt}(X_t(z)) > 0, X_t(z)M(z) \geq t\}$
end while
Return $SDPS$

assumption that only zone z^* is displayed. Next, we determine $\mathcal{Z}_{cand}$ the same way described above, but we exclude zones that are already in $SDPS$ to avoid repetition. We repeat this process until there are no eligible zones to add to $\mathcal{Z}_{cand}$.

Initial construction of $\mathcal{Z}_{cand}$, and finding the zone with maximum cost savings per period requires iterating over at most Z zones. Constructing $\mathcal{Z}_{cand}$ for the second time and onwards require iterating over at most Z zones, and the $SDPS$ list, which may contain at most Z zones. All other update operations require $\mathcal{O}(1)$ time. Therefore, the computational complexity of Algorithm 1 is $\mathcal{O}(Z^2)$.

8.3. Solution Approach Validation

In this section, we test and report the goodness of the stochastic look-ahead (LA) proposed in Section 4. The proposed LA consists of both state and action space reduction techniques (termed value function approximation(VFA) and efficient display sets (EDS), respectively). To quantify the performance of these ideas, we need instances where we can use the backward recursion and compute the expected fulfilment cost stated in Equation 5. Therefore, we consider 50 randomly generated instances consisting of eight zones located along the perimeter of a circle and the depot is situated at the centroid of the circle. 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. The primary goal of these experiments is to assess the performance of the stochastic look ahead method. Therefore, the cost savings may not represent the reality due to the artificial instances.

Figure 10 presents the choice-based crowdshipping cost savings for the small instances of five task display policies: 1 Customized Display-derived optimally (CD-opt), 2 Customized Display-derived by stochastic look-ahead (CD-SL), 3 Single Best Display-derived by stochastic look-ahead, 4 Display-All, and 5 Clearance-1. We see that the median cost-saving difference between the proposed stochastic look-ahead method and the optimal policy is less than 1%.

Figure 10: Cost savings of using varying dynamic task display mechanisms, **eight zones and twenty tasks**.

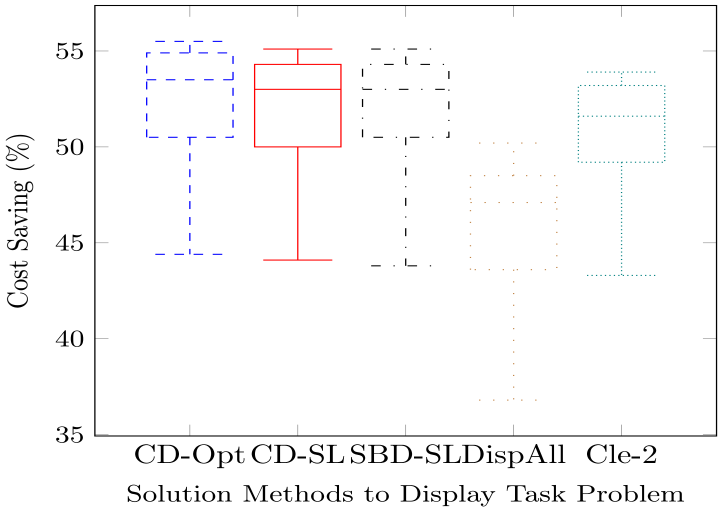
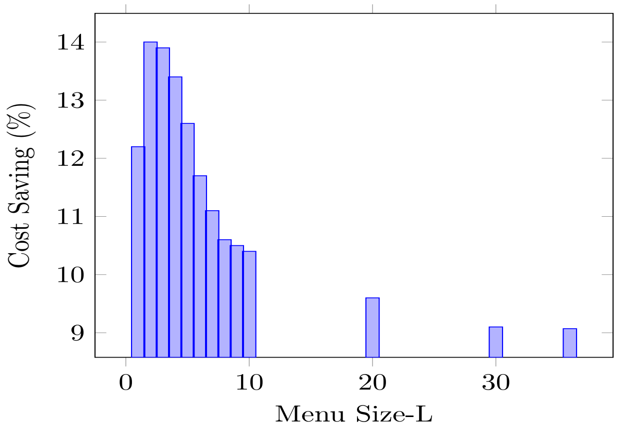


Figure 11: Cost savings for varying menu sizes for clearance policy.



8.4. Performance of Clearance-L policies

This section presents the impact of varying display menu sizes using Clearance-L. The Clearance-L policy aims to minimize the fulfilment expenses of engaging contract drivers, but it is allowed to show up to L zones as a menu size. Note that if the total number of zones with positive inventory drops below L at some point in the selection horizon, then this policy becomes equivalent to Display All after that time.

Figure 11 shows the cost savings of the Clearance-L policy for the 50 instances generated using Singapore data, with L varying from 1 to 36. The results show that for our instance set, average cost savings increase as L goes from 1 to 2, and start to decrease afterwards, with $L = 2$ being the best Clearance policy and $L = 3$ coming as a close second. Note that these results are specific to our instance set and cost function and the best-performing Clearance-k policy might differ under different termination cost functions or instance sets.

8.5. Impact of request and driver volume

This section examines how the volume of delivery requests and available crowd drivers influence the benefits of crowd shipping. To explore this aspect, we consider two experimental settings, each with varying numbers of requests and drivers: 100, 250, 500, 750, and 1000.

In the first setting, we use our original instances inspired by Singapore's geography while keeping the number of zones (which corresponds to the maximum number of contract drivers) constant. In the second or round-shaped setting, we use a fixed-sized circular area but adjust the number of zones to maintain an average of 25 delivery requests per zone. In other words, while the number of zones will be 36 in Singapore, regardless of the request volume, it will vary between 4 and 40 for the second setting.

Figure 12: Round-shaped geography. Zone size changes.

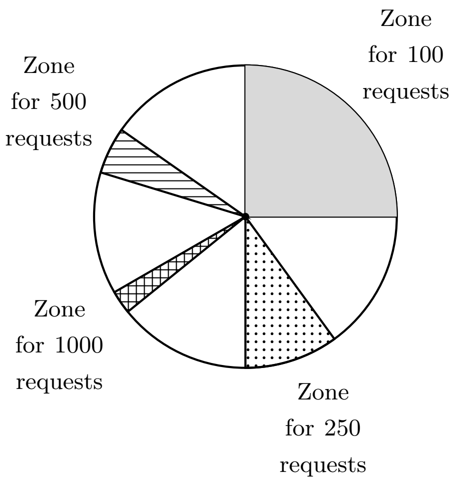


Figure 12 illustrates the idea behind the round-shaped setting. We set the depot at the centre, where all requests are to be collected, and crowd and contract drivers originate their journeys. We consider the destinations of the requests to be distributed uniformly within the area. Depending on the request volume, we set the zone sizes; *i.e.*, for 100 requests, we partition the area into four zones or for 250 requests into ten zones.

We keep the monetary compensation paid to crowd drivers and recruitment costs to contract drivers at the original levels for both settings. Hence, we concentrate on quantifying the impact of volume changes within the same areas, with and without the ability to rezone. It is crucial to note that the expenses of contract drivers are closely related to how zones are set and aligned with the expected number of requests.

Table 5 illustrates the cost-saving ratios associated with integrating crowdshipping into next-day delivery operations through various display policies. It is clear that the advantages of crowdshipping diminish as the volume increases or when delivery requests become more geographically concentrated. In other words, the marginal effort required to deliver additional requests is lessened for contract drivers due to the closer proximity of delivery destinations while compensation per crowd remains unaffected by the density.

Table 5: Impact of request and crowd driver volume.

Cost Savings(%): Singapore instances. No Re-zoning.				
Volume	# Zones	Customized Display	Display-All	Cle-2
100	36	59.8%	53.2%	49.8%
250	36	37.2%	30.4%	31.3%
500	36	16.9%	9.0%	14.0%
750	36	7.6%	-3.0%	4.5%
1000	36	3.1%	-10.8%	-1.6%
Cost Savings(%): Round-shaped geography instances. With Re-zoning.				
Volume	# Zones	Customized Display	Display-All	Cle-2
100	4	18.0%	18.0%	16.7%
250	10	11.5%	8.9%	9.2%
500	20	5.5%	2.8%	3.8%
750	30	3.2%	-0.3%	0.9%
1000	40	2.2%	-2.3%	-1.0%

We also observe that rezoning ability significantly influences cost-saving computations in Table 5. Since contract driver cost estimation depends on the initial zoning setup—the total number of zones—failing to adjust zones in response to major changes in delivery request volume can lead to over- or underestimation of contract driver expenses. Simply put, recruiting more than five contract drivers (i.e., zones) may not be ideal if the expected delivery demand is only 100. To address the shortcomings of Singapore instances in this aspect, we develop an adaptive zoning policy for round-shaped geographic instances.

Table 5 presents the volume impact on the display policies. We see that the Display-All policy performs on par with the customized display when the request density is low. At the same time, the Display-All fails to facilitate saving when the delivery density increases. Table 5 also reveals the weight of the adaptive display policy in overnight delivery services. These results contribute to the overall discussion of how crowdshipping integration into overnight delivery services requires additional attention in comparison to on-demand delivery services.

8.6. Impact of delay in crowd driver's choice

In this section, we elaborate on our experiment to quantify the impact of a crowd driver's delayed decision (i.e., making a choice after the next driver arrives) on cost savings. In this experiment, each arriving crowd driver delays their decision with probability p_d, postponing the decision of the driver by n_d periods, allowing subsequent crowd drivers to arrive at the platform before the delayed driver makes a choice. These delays occur randomly and cannot be observed by the system in advance. During the decision delay of a driver, any options displayed to the driver are temporarily blocked to prevent selection conflicts. Consequently, if a displayed zone has only one remaining task and is shown to a crowd driver, it remains unavailable for later arrivals until the driver finalizes their decision. We run our experiment by varying n_d between one and three and p_d between 0.1 and 1, with increments of 0.1.

Figure 13: The impact of drivers' delayed choice

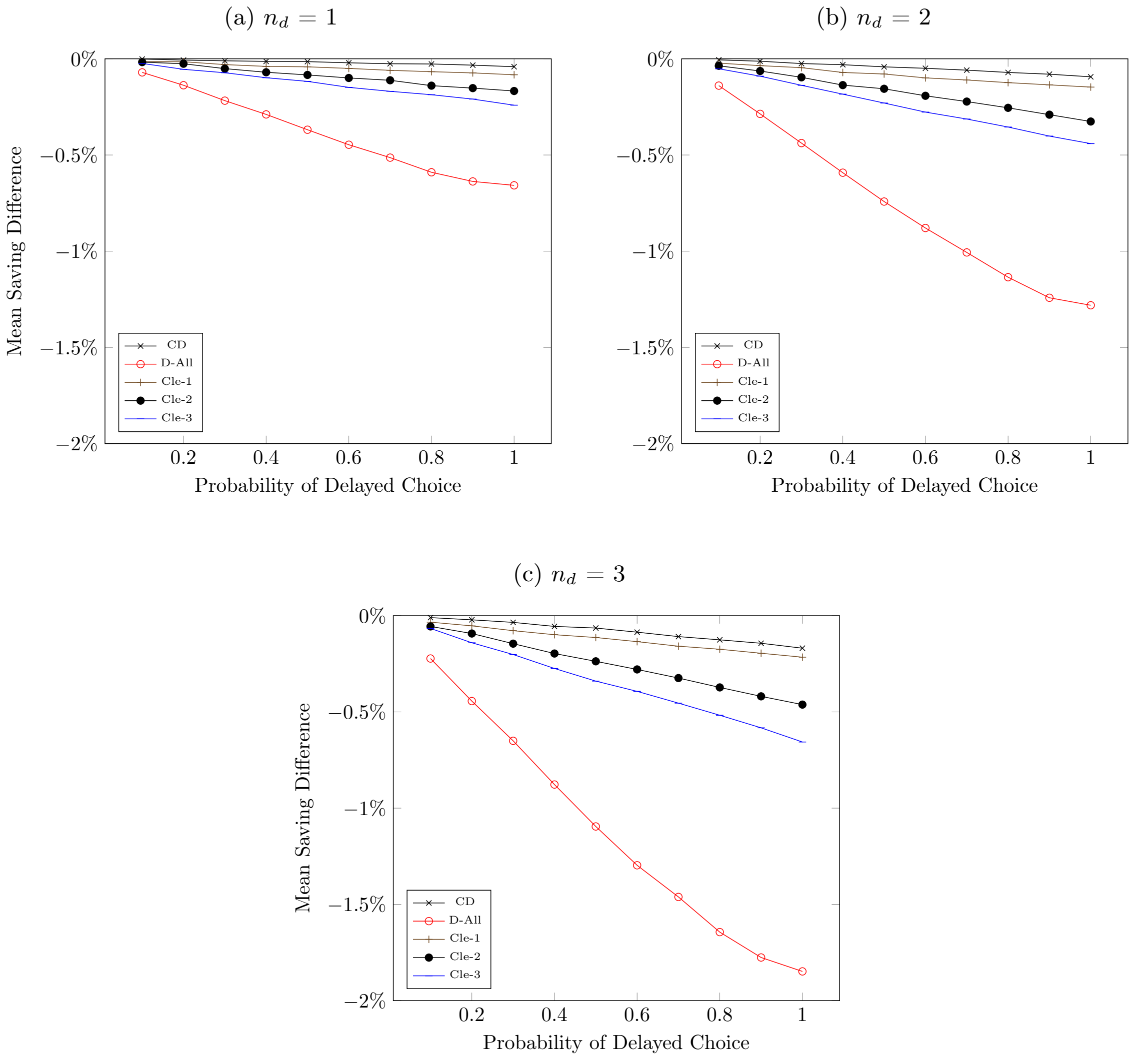


Figure 13 shows the percent point (pp) change in average cost savings with respect to the savings for the base case displayed in Table 2 for the Customized Display policy and all benchmark policies. As one would expect, cost savings decrease in the presence of delayed driver decision, and the magnitude of the decrease increases with i) probability of delayed choice, and ii) length of the delay, regardless of the policy implemented. Moreover, the impact of delayed decision is the smallest for all values of n_d and p_d when the Customized Display policy is implemented. This experiment concludes that while assuming that a driver makes their choice before the next driver arrives creates a bias in the results, this bias does not impact the results of this study.