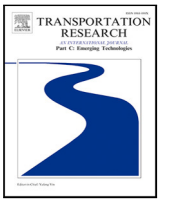
Transportation Research Part C 160 (2024) 104533

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc



Task assignment, pricing, and capacity planning for a hybrid fleet of centralized and decentralized couriers☆

Adam Behrendt *, Martin Savelsbergh, He Wang

School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

ARTICLE INFO

Keywords:
Crowdsourced delivery
Same-day delivery
Capacity planning
Pricing
Online marketplaces
Repositioning

ABSTRACT

Crowdsourced delivery platforms operate as an intermediary between consumers who require delivery tasks and couriers who make these deliveries; both of which are uncertain. The main challenge of a crowdsourced delivery platform is to meet a service level for their customers (e.g., 95% on-time delivery) by serving dynamically arriving delivery tasks with time windows. The two critical courier management decisions for a platform are how to schedule couriers and how to assign delivery tasks to couriers. These two decisions can be centralized (i.e., decided by the platform) or decentralized (i.e., decided by the couriers). Centralizing these decisions produces a more reliable workforce while decentralizing them may come with cost savings to the platform and allows more freedom to couriers in deciding when and where to work. Crowdsourced delivery platforms have begun to utilize both courier types simultaneously (i.e., a hybrid system) with the hope of reaping the advantages of each. In this paper, we address the challenge of capacity planning for a crowdsourced delivery platform that utilizes both centralized (committed) and decentralized (ad-hoc) couriers. We present fluid models for delivery systems using either type of courier, and a hybrid formulation. Our theoretical, numerical, and simulation results establish the superiority of a hybrid system over each pure system in a majority of real-world scenarios.

1. Introduction

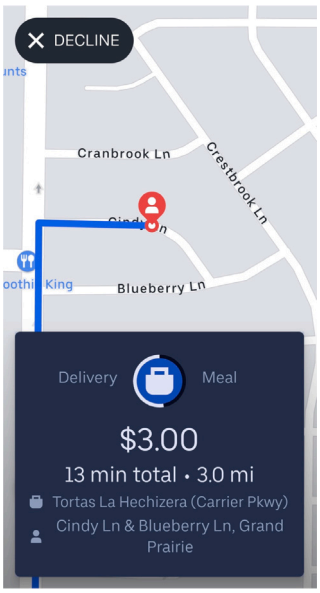
Last-mile delivery has seen sustained and rapid growth in the past few years; consumer participation in, and reliance on, e-commerce has been the dominant driver, with the market size for last-mile e-commerce delivery slated to grow from $56.25 billion in 2021 to $136.12 billion in 2030 (Spherical Insights LLP, 2022). At the same time, the emergence of the gig economy has given rise to crowdsourced delivery platforms using flexible, non-employee couriers known as *crowdsourced couriers* to provide last-mile delivery capacity (Cosgrove, 2022). A crowdsourced delivery platform operates as an intermediary between consumers who place delivery tasks and couriers who make deliveries. Delivery tasks arrive dynamically to the platform and often have to be performed shortly after they arrive in the case of same-day item delivery like Amazon (Chen, 2023) and grocery delivery like Instacart (2023). The platform seeks to achieve a predetermined service level (e.g., 95% on-time delivery, guaranteed delivery by end-of-day, etc.) using crowdsourced couriers while profiting on the margins.

The two critical courier management decisions for a platform are how to schedule couriers and how to assign delivery tasks to couriers (referred to as scheduling and matching decisions by Alnaggar et al. (2021)). These decisions can be centralized (i.e., decided by the platform) or decentralized (i.e., decided by the couriers). Essentially, the platform is deciding the level of control, or lack

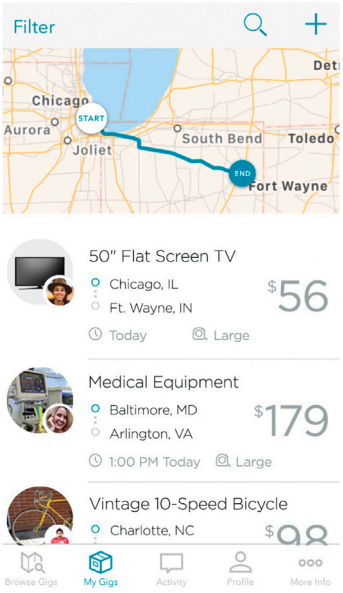
☆ This article belongs to the Virtual Special Issue on IG005587: VSI: Transport Logistics.

* Corresponding author.
E-mail address: adam.behrendt@gatech.edu (A. Behrendt).

https://doi.org/10.1016/j.trc.2024.104533
Received 20 March 2023; Received in revised form 7 February 2024; Accepted 14 February 2024
Available online 4 March 2024



(a) A platform-assigned delivery task assignment scheme where the platform "pushes" delivery tasks to couriers. Screenshot from the Uber Eats application (Financial Anatomy, 2022).



(b) A self-assigned delivery task assignment scheme where couriers "pull" delivery tasks from the platform. Screenshot from the Roadie application (Murdocca, 2017).

Fig. 1. Different delivery task assignment schemes employed by crowdsourced delivery platforms in practice.

thereof, it has over the couriers' choices of "when" to work and "what" to deliver. Two main types of scheduling methods are commonly used in practice: (1) platform-scheduling, where the platform controls the creation of shifts before the start of an operating period, and (2) self-scheduling, where the couriers arrive in an ad-hoc manner, announce their availability to the platform dynamically, and serve delivery tasks until they decide to depart. Note that traditional employee couriers fall into the category of platform-scheduled in its most restrictive definition (e.g., specific couriers are assigned specific shifts). However, the platform-scheduled category also encompasses crowdsourced couriers who sign up for shifts (created by the platform) with some probability ahead of time, according to their preference. Regardless, platform-scheduling guarantees that when an operating period begins, the platform has full information on its workforce. Due to this, platform-scheduling allows platforms to better match the temporal nature of supply and demand, whereas self-scheduling allows couriers more freedom in deciding when and where to work, while simultaneously reducing the chance of over-scheduling couriers relative to demand. As this additional freedom benefits couriers, it also makes it easier for platforms to classify couriers as independent contractors in the wake of recent legislation like Assembly Bill 5 in California (Woodruff Sawyer, 2020).

In addition to courier scheduling, we observe two main types of delivery task assignment schemes used by delivery platforms: (1) platform-assignment, where the platform "pushes" delivery tasks to couriers, and (2) self-assignment, where couriers "pull" delivery tasks from the platform; see Fig. 1. With platform-assignment (centralized delivery task assignment), the platform pushes delivery tasks to couriers, who may then accept or reject the delivery tasks.[1] Similar to platform-scheduling, the most restrictive case of platform-assignment serves to model traditional employee couriers who must accept all delivery tasks assigned. Platform-assignment reduces the uncertainty surrounding which delivery tasks are served and when, making it easier to meet a desired service level. However, platform-assigned couriers have little choice on which specific deliveries they make while the flexibility provided by gig work is viewed as a benefit of the job in many cases (Pierbridge, Inc, 2018). This improved flexibility helps to drive participation in self-assignment schemes and can mitigate the effects of traditional employee courier shortages. With self-assignment (decentralized delivery task assignment), the platform posts delivery tasks on a virtual bulletin board, and couriers select their preferred delivery task from among the posted delivery tasks. There may be varying degrees of self-assignment, as the platform may curate a different menu of delivery tasks for different couriers. When couriers can select the delivery task that perfectly matches their preferences (which is often unknown to the platform), they may be willing to accept a smaller payout on average per delivery task (e.g., a delivery task is able to be served on the way to a courier's pre-determined destination).

Based on the scheduling and assignment decisions, there are four potential driver types in general: platform-scheduled/platform-assigned, platform-scheduled/self-assigned, self-scheduled/platform-assigned, and self-scheduled/self-assigned. To the best of our

[1] However, rejecting too many delivery tasks may result in a courier receiving fewer assignments in the future.

knowledge, there are no real-world instances of platform-scheduled/self-assigned couriers, therefore, we omit them from our discussion. The platform-scheduled/platform-assigned category encompasses traditional employee couriers and more centralized versions of crowdsourced couriers. Take Amazon for example, which uses platform-scheduled/platform-assigned couriers in two forms. Amazon hires traditional employee couriers (e.g., contracted couriers) and crowdsourced couriers via their program Amazon Flex (Amazon, 2023). Amazon Flex couriers sign up for time blocks (created by Amazon) ahead of time and drive delivery routes (created by Amazon). From an operational perspective, these crowdsourced couriers are equivalent to their direct employee counterparts, as they do not have the option to reject deliveries. As such, when we describe the benefits of drawbacks of platform-scheduled/platform-assigned couriers we are referencing this general class of couriers, not just the crowdsourced variant. An example of self-scheduled/platform-assigned couriers is that of the platform Uber Eats (Uber, 2023), where couriers can elect at any time to begin work, which indicates to the platform that it may push orders to that courier. Finally, an example of self-scheduled/self-assigned couriers being used is Roadie (2023), where couriers can self-schedule and then select which orders to deliver off of a bulletin board. In our paper, we are concerned with platform-scheduled/platform-assigned and self-scheduled/self-assigned couriers. Hereafter, we will refer to platform-scheduled/platform-assigned couriers as *committed couriers* and self-scheduled/self-assigned couriers as *ad-hoc couriers*. Committed couriers are typically compensated on a per-hour basis, while ad-hoc couriers are typically compensated on a per-delivery task basis.

We refer to a platform that relies on multiple types of couriers as a *hybrid delivery system*. In contrast, a *pure delivery system* utilizes a single type of couriers. In a hybrid delivery system, we refer to the courier types as *subsystems* or *delivery channels* (interchangeably). Pure delivery systems tend to have trade-offs. For example, a pure committed system elects for a deterministic courier base paid for their time. When demand patterns are predictable, the results in an optimized and efficient fleet of couriers that consistently meets delivery promises. With the stochastic nature of order arrivals, however, the large fixed cost of hourly shifts becomes a liability when demand forecasts are over-estimated. Pure ad-hoc systems, on the other hand, forego certainty in the courier base (and many times, on-time guarantees for customers) for efficient operating costs under uncertainty. Pure ad-hoc systems can hedge against periods of unexpectedly low demand, while leveraging ad-hoc couriers' delivery task preferences and desire for flexible working hours to create cost savings on a per-order basis. However, when there is a mismatch of supply of ad-hoc couriers and demand, surging payouts to ad-hoc couriers (in order to incentivize their participation) can quickly lead to losses.

A hybrid system employing both ad-hoc and committed couriers strikes a balance, managing cost by alleviating the strain on the ad-hoc courier pricing channel while granting the platform explicit control over certain delivery assignments. Additionally, decisions made by each subsystem, such as task assignments and relocation for committed couriers, and pricing for ad-hoc couriers, can optimize the spatial distribution of demand for the other. This synergy enhances efficiency, leading to lower average prices for ad-hoc couriers and improved utilization of committed couriers.

While utilizing a hybrid delivery system has the potential to utilize both delivery channels synergistically, it also introduces a new challenge in task assignment and capacity planning for the platform. Specifically, *how can crowdsourced delivery platforms jointly solve for an optimal fleet size and delivery task assignment strategy for committed couriers and a pricing strategy for ad-hoc couriers?* In this paper, we address that over-arching research question. The main contributions of this paper are as follows:

- We model a hybrid delivery system that utilizes committed (centralized) and ad-hoc (decentralized) couriers in a manner that jointly accounts for fleet sizing, delivery task assignment and empty repositioning for committed couriers and delivery task pricing for ad-hoc couriers.
- We analyze these models and conduct numerical experiments to argue the benefits of a hybrid delivery system over pure committed or ad-hoc systems in a realistic crowdsourced same-day delivery setting.
- We provide insight that can be used by real-world crowdsourced delivery platforms in the long-term design of their platforms and mid-term planning problems.

The remainder of this paper is organized as follows. Section 2 details the relevant literature related to hybrid delivery systems. Section 3 describes the problem setting. Section 4 formally describes our models and assumptions. Section 5 provides our analysis of a hybrid system, as well as numerical experiments investigating the structure of solutions and validating our hypotheses. Finally, Section 6 presents the implications of our findings in terms of recommendations for crowdsourced delivery platforms and future research directions.

2. Literature review

This work is at the intersection of many research areas with vast collections of literature (crowdsourced delivery, ride hailing, pricing, and marketplace design, to name a few). For an overview of crowdsourced delivery, we direct the reader to Le et al. (2019), Alnaggar et al. (2021), and Savelsbergh and Ulmer (2022). One of the main contributions of our work is that we propose a joint model for a crowdsourced delivery platform that employs two types of couriers: (1) platform-scheduled and platform-assigned (committed or centralized couriers); and (2) self-scheduled and self-assigned couriers (ad-hoc or decentralized couriers). Thus, Table 1 provides a collection of recent related work surrounding platforms that employ crowdsourced drivers (i.e., package delivery or ride hailing) and categorizes the types of couriers studied.

The reader may notice that aside from this paper, there are four distinct applications in the table. The first application is a platform that employs two types of drivers, the first being platform-scheduled/platform-assigned and the second being self-scheduled/platform-assigned. The papers we included (first 7 rows of the table) view task assignments as an exclusively centralized decision (i.e., by the platform, although in many cases drivers can reject assignments). These papers tend to focus on the uncertainty

Table 1
Crowdsourced driver literature by courier types.

Random choice behavior	Courier types	Platform-Scheduled	Platform-Assigned	Self-Scheduled	Self-Assigned	
					No	Yes
Lee and Savelsbergh (2015)	2	✓	✓	✓		
Archetti et al. (2021)	2	✓	✓	✓		
Arslan et al. (2019)	2	✓	✓	✓		
Ulmer and Savelsbergh (2020)	2	✓	✓	✓		
Yildiz and Savelsbergh (2019)	2	✓	✓	✓		
Castillo et al. (2022)	2	✓	✓	✓		
Santini et al. (2022)	2	✓	✓	✓		
Cao et al. (2020)	2	✓	✓	✓	✓	
Torres et al. (2022)	2	✓	✓	✓	✓	
Behrendt et al. (2022)	2	✓	✓	✓	✓	
Cachon et al. (2017)	1		✓	✓		
Özkan (2020)	1		✓	✓		
Banerjee et al. (2022b)	1		✓	✓		
Bimpikis et al. (2019)	1		✓	✓		
Hu et al. (2022)	1		✓	✓		
Kleywegt and Shao (2021)	1		✓	✓		
Cao et al. (2022)	1			✓		✓
This paper	2	✓	✓	✓		✓

associated with the availability of self-scheduling couriers, and many are extensions of the dynamic vehicle routing problem. This specific type of hybrid system is an improvement over using either type of driver exclusively. Self-scheduled/platform-assigned drivers assist with load balancing in times of uncertain and inconsistent demand, avoiding the issue of over or under-hiring, while platform-scheduled/platform-assigned drivers can be used to meet consistent demand more efficiently. Thus, these papers are similar to our own in that they answer questions about hybrid systems, but fail to address systems in which drivers have explicit choice over which tasks they serve. Our paper specifically addresses the probabilistic choice of self-scheduled/self-assigned couriers in conjunction with managing a fleet of platform-scheduled/platform-assigned couriers.

The second application is similar to the one studied in this paper, with one key difference. The papers we included study a hybrid system with platform-scheduled/platform-assigned and self-scheduled/self-assigned drivers, however, the self-assigning behavior of the drivers is assumed to be deterministic. Specifically, in Cao et al. (2020) the self-assigned drivers are assumed to arrive at the platform with exactly which tasks they will serve in mind. Furthermore, the practicality of Cao et al. (2020) as a study on a hybrid system is questionable, as it is assumed that the two types of drivers operate at different times. In Torres et al. (2022) self-assigned drivers select the task with the most compensation, thus it is possible for the platform to rank which tasks have the highest likelihood of being selected. In Behrendt et al. (2022), self-assigned couriers select the task from the nearest pickup location with the maximum compensation (similar to Torres et al. (2022), except with multiple pickup locations). Our paper, on the other hand, explicitly models the choice probability of self-assigned couriers as a function of the price, lead time, and proximity of a task to their location.

The third application does not focus on the notion of a hybrid system, but a single type of driver that is self-scheduled/platform-assigned. The papers we included are in the ride-hailing and ride-sharing domains, and view the problem as a two-sided marketplace between customers looking to be transported and drivers. While the problem is fundamentally similar to the first two types of papers we discussed, it is viewed through a different lens. These papers heavily focus on dynamic pricing policies and matching decisions to balance the two sides of the marketplace. In the previous two applications, pricing is modeled in a more simplistic fashion and less emphasis is placed on balancing the system due to the presence of a base of centralized drivers that can guarantee some level of service. Our paper takes inspiration from this application, using similar pricing techniques and balancing the two sides of the market, but we extend the area to include pricing for self-assigned couriers as opposed to platform-assigned, as well as fleet sizing and task assignment for platform-scheduled/platform-assigned couriers.

Finally, the fourth application, highlighted by the recent work of Cao et al. (2022) focuses exclusively on pricing delivery tasks for self-scheduled/self-assigned couriers in two-sided marketplaces where the self-assignment is based on a random choice model. These pricing models based on random choice behavior are commonplace in online marketplace research, where a customer may be selecting from multiple products to buy. In our application, the random choice comes from the driver selecting which customer to serve. As we highlighted in the introduction, this self-assignment behavior is already being implemented at crowdsourced delivery companies like Roadie. We extend the model in Cao et al. (2022) to account for multiple pickup and delivery locations as opposed to just lead time, while adding platform-scheduled/platform-assigned couriers to create a model that solves capacity planning, delivery task assignment, and pricing for a hybrid system that utilizes both types of couriers. As Table 1 highlights, our work is the first to study a system that utilizes these two types of couriers. As such, this paper expands upon the various applications we illustrated in Table 1 by proposing new models and generating new insight for this novel application.

3. Problem description

In this section, we provide a high-level and informal description of the problem setting, before presenting the detailed mathematical models in the next section. We consider a capacity planning problem faced by a crowdsourced delivery platform. The

Table 2
Some notations used in the model.

Notation	Meaning
V	The set of vertices in the network
A	The set of arcs in the network
c_{ij}	Travel time on arc (i,j)
ℓ	Relative lead time of a delivery task (with range $L := [L_{\min}, L_{\max}]$)
	Ad-hoc couriers
k	Current location of an ad-hoc courier
γ	Average number of orders selected by an ad-hoc courier
θ_ℓ	Penalty cost for a task at lead time ℓ
$\lambda_{ij\ell}$	Arrival rate of new tasks for arc (i,j) at lead time ℓ
μ_k	Exogenous arrival rate of ad-hoc couriers at location k
d_{kij}	Cost for a courier at k to serve a task on (i,j)
$u_{kij\ell}$	Choice probability for couriers at k to serve a task (i,j) with lead time ℓ
$p_{kij\ell}$	Price for couriers at k to serve a task (i,j) with lead time ℓ
	Committed couriers
f	Guaranteed hourly wage
M	Fleet size of committed couriers
ρ	Multiplier converting fleet size to service rate
$r_{ij\ell}$	Rate at which delivery tasks on arc (i,j) with lead time ℓ are assigned
s_i	Aggregate assignment rate at location i
z_{ij}	Empty repositioning rate from i to j

platform is partnered with enterprise customers (e.g., online retailers or grocery store chains) as a supplier of delivery capacity. As such, delivery tasks arrive dynamically over an operating period. Each task is characterized by an origin, destination, and delivery deadline. Delivery tasks may be delivered late, in which case they incur a penalty cost that may increase with the lateness. Additionally, there is a maximum lateness value where the delivery task is said to expire and be removed from the system (incurring an expiration penalty).

The crowdsourced delivery platform hires both *committed* and *ad-hoc* couriers to serve customer demand. We assume that the order assignments for committed couriers are fully controlled by the platform for the entire operating period, and the committed couriers are compensated for their time with an hourly wage. As such, the platform must decide the fleet size of committed couriers to hire, along with its specific spatial operations, e.g., delivery task assignment and empty repositioning. Unlike committed couriers, ad-hoc couriers are not directly controlled by the platform but rather are indirectly controlled by the platform's delivery task pricing decisions and are paid per order they serve. Our assumption of different payment schemes for committed and ad-hoc couriers is motivated by our experience collaborating with several crowdsourced delivery platforms. Ad-hoc couriers arrive exogenously (see Section 4.2 for further discussion on this) to the platform and select orders according to their own preferences. Ad-hoc courier choices are functions of the prices the platform sets for each delivery task and the distance from the location of the ad-hoc courier to the origin plus the distance from the origin to the destination. In practice, the platform may hire exclusively committed couriers, completely rely on ad-hoc couriers, or utilize both, i.e., employ a hybrid fleet of couriers. In this paper, we model the capacity planning problem to decide the relative capacities of each subsystem (committed or ad-hoc) to be used to serve demand at minimum cost. In this manner, we are jointly solving for the optimal pricing strategy for the ad-hoc system and the optimal fleet size, delivery task assignments, and repositioning of the committed couriers. Solutions to this problem can be used by platforms to plan their employment strategies, and analyses of our models highlight the mechanisms by which a hybrid system outperforms either pure delivery system in most realistic scenarios.

4. Model

In this section, we formulate the mathematical model for a crowdsourced delivery platform (see Table 2 for notation). We consider the time-homogeneous case where the rates at which delivery tasks (demand) and ad-hoc couriers (supply) arrive are constant in time. We also assume the travel time on any arc does not change over time. In practice, the demand and supply rates may be time non-homogeneous, and the travel time can vary during the time of the day. In that case, the decision-maker of the platform can divide the time horizon into smaller time segments in which the rates and travel time are approximately constant, and then solve a separate decision problem for each time segment. Therefore, our model below can also be used in time non-homogeneous settings.

4.1. Demand for delivery tasks

Let $G = (V, A)$ be a directed graph, where V is a vertex set representing locations and $A = \{(i,j) : i,j \in V\}$ is a directed arc set. Demand is defined on the arcs, as a delivery task has an origin and a destination (e.g. a package must first be picked up and then delivered). Delivery tasks have a placement time t_p and a delivery deadline t_d. For current time t, the "relative lead time" $\ell(t)$ is defined as $t_d - t - c_{ij}$, where c_{ij} is the expected travel time on arc $(i,j) \in A$. When the relative lead time is less than zero,

the delivery task will be delivered late. Therefore, the relative lead time is a measure of how late or how early a delivery task will be, if delivery begins at time t. We assume that the relative lead time at the placement time of a delivery task $\ell(t_p)$ is randomly distributed on $(0, L_{\max}]$, not necessarily homogeneously, where $L_{\max}$ is the longest possible relative lead time at a delivery task's arrival. Our model is in *discrete-time* and $\lambda_{ij\ell}$ is the rate at which delivery tasks arrive to arc (i,j) with relative lead time ℓ. Hence, delivery tasks are characterized by their associated arcs (i,j) and their remaining relative lead time ℓ at the current time t. When a delivery task's relative lead time is less than zero, the platform incurs a penalty cost. We let the penalty $\theta_\ell > 0$ and $\theta_{\ell-1} \geq \theta_\ell$ for all $\ell = L_{\min}, L_{\min}+1, \dots, -1$, where $L_{\min}$ is the smallest negative lead time (e.g., maximum delay) allowed. If a task reaches the relative lead time $\ell = L_{\min}$, it becomes expired and is removed from the system; in that case, the delivery platform pays a penalty cost $\theta_{L_{\min}}$ to the customer for not delivering the task. In practice, expired orders are often canceled by the customers or reassigned to other delivery channels, so we assume they are removed from the system. Alternatively, if an expired order is rescheduled for delivery at a later time, we assume it is first removed from the system and then added to the system as a new order. For ease of notation, we denote $L := \{L_{\min}, L_{\min}+1, \dots, L_{\max}-1, L_{\max}\}$.

4.2. Ad-hoc couriers

As we described previously, ad-hoc couriers are those that self-schedule and self-assign delivery tasks. For a given location (vertex) $k \in V$, ad-hoc couriers arrive with rate μ_k. The arrival rate μ_k is exogenous to the system, and can be thought of as the rate at which ad-hoc couriers start interaction with the platform (i.e., open the mobile application looking for work) from location k. The assumption that the arrival rate μ_k is exogenous to the platform's pricing decisions is a reasonable one, as it simply captures couriers' behavior of opening the app to begin searching for work, whereas the prices can only be observed by couriers after they open the app. The rate μ_k may be dependent on many extrinsic factors, e.g., the number of couriers perceived to be in an area, the long-term history between the couriers and the platform, and competitor platforms. The literature on ride-hailing, for example, models the arrival rate to the system as endogenous and dependent on the price. This is because those studies are solving a fundamentally different problem than we are. In our model, the behavior of ad-hoc couriers can be thought of in two stages, beginning with: (1) the rate at which they open the mobile application (exogenous of the price); and (2) the rate at which they select (or do not select) delivery tasks to serve (endogenous of the price). Platforms that use ad-hoc couriers typically estimate this exogenous rate through the "interaction" statistics of their mobile application. The rate at which the platform *capitalizes* on this interaction, however, is endogenous, and clearly dependent on the prices that the platform sets for delivery tasks.

After arriving at vertex k, an ad-hoc courier will select a delivery task to serve, or serve none. The courier makes this choice in order to maximize their personal utility, given by the function

$$U_{kij\ell} = \beta p_{kij\ell} - d_{kij} + \varepsilon_{kij\ell} \;\; \forall k \in \mathcal{V}, (i,j) \in \mathcal{A}, \ell \in L, \tag{1}$$

where $p_{kij\ell}$ is the price for couriers at location k of a delivery task on arc (i,j) with relative lead time ℓ, β is a universal price sensitivity parameter, d_{kij} is the cost of a courier at location k serving a delivery task on arc (i,j) (for example, the cost d_{kij} may be approximated by the travel distance from vertex k to i for pick-up, plus the distance from vertex i to j to delivery the item), and finally, $\varepsilon_{kij\ell}$ is the portion of the utility that is unknown to the platform. Under the assumption that $\varepsilon_{kij\ell}$ follows a Gumbel distribution, we have the well-known multinomial logit (MNL) model. The MNL model has been widely used in the transportation and pricing literature for modeling individual choice behavior thanks to its flexibility, interpretability, and computational efficiency (McFadden, 1973; Ben-Akiva and Lerman, 1985; Talluri and Van Ryzin, 2004; Train, 2009). Under the MNL model, the price sensitivity parameter The price β can be estimated from ad-hoc courier selection data using multinomial logit regression (Barbosa et al., 2023).

Letting $v_{kij\ell} := \exp(\beta p_{kij\ell} - d_{kij})$ be the preference weight, we have the choice probability of ad-hoc couriers at location k selecting a delivery task i, j, ℓ as

$$u_{kij\ell} = \frac{x_{ij\ell} v_{kij\ell}}{\sum_{(i,j)\in A} \sum_{\ell \in L} x_{ij\ell} v_{kij\ell} + 1}, \tag{2}$$

and the probability that they select no delivery task as

$$u_{k0} = \frac{1}{\sum_{(i,j)\in A} \sum_{\ell \in L} x_{ij\ell} v_{kij\ell} + 1}. \tag{3}$$

Notice that the choice probabilities above are weighted by $x_{ij\ell}$, which is the number of delivery tasks with attributes (i,j,ℓ). This is to ensure that a task with attributes (i,j,ℓ) is only selected if there is at least one such delivery task present.

The above multinomial logit (MNL) choice model allows for at most a single delivery task to be chosen per arrival of an ad-hoc courier. An important behavior in practice, however, is the ability for an ad-hoc courier to select multiple delivery tasks upon arrival and thus create a bundle. That is, each exogenous arrival of ad-hoc couriers (i.e., starting a mobile app session) may result in multiple selections. In order to approximate this behavior with our current model, we make a simplifying assumption that for each arriving ad-hoc courier, these multiple selections share the same choice probabilities. In other words, the courier's preferences for tasks are not dependent on the choice order. As such, we can simply scale the arrival rate μ_k for each location by a multiplier γ which represents the average number of delivery tasks selected per arrival, which the platform can estimate with historical data.

Next, we consider the platform's decision to choose prices for ad-hoc couriers. The standard approach is to formulate the decision problem as a Markov Decision Process (MDP) that incorporates the dynamics of tasks and couriers; however, solving the MDP

model is computationally intractable because the state space of the MDP grows exponentially with the size of the network ($|V|$ and $|A|$) and the length of the lead time (L). Therefore, we will formulate the problem using a fluid model, which can be viewed as an approximation of the MDP in large-scale systems (Dai and Harrison, 2020). The fluid approximate model is a single-stage optimization problem to find the optimal set of prices $\mathbf{P} = \{p_{kij\ell} : \forall\ k \in V, (i,j) \in A,\ \ell \in L\}$.

$$\min\ \gamma \sum_{k\in V} \sum_{(i,j)\in A} \sum_{\ell\in L} \mu_k u_{kij\ell} p_{kij\ell} + \sum_{\ell\in L} \theta_\ell \sum_{(i,j)\in A} x_{ij\ell} \tag{4a}$$

$$\text{s.t. } x_{ij\ell} - x_{ij\ell-1} = \gamma \sum_{k\in V} \mu_k u_{kij\ell} - \lambda_{ij\ell-1} \qquad \forall (i,j) \in A,\ \ell \in L \tag{4b}$$

$$x_{ijL_{\max}} = \lambda_{ijL_{\max}} \qquad \forall (i,j) \in A \tag{4c}$$

$$\gamma \sum_{k\in V} \mu_k u_{kij\ell} \le x_{ij\ell} \qquad \forall (i,j) \in A,\ \ell \in L \tag{4d}$$

$$\sum_{(i,j)\in A} \sum_{\ell\in L} u_{kij\ell} + u_{k0} = 1 \qquad \forall k \in V \tag{4e}$$

$$x_{ij\ell} \ge 0 \qquad \forall (i,j) \in A,\ \ell \in L \tag{4f}$$

$$u_{kij\ell} \ge 0 \qquad \forall k \in V,\ (i,j) \in A,\ \ell \in L \tag{4g}$$

The objective function (4a) seeks to minimize the total payout to ad-hoc couriers and the cumulative late and expired penalty, and is non-linear because the price for delivery task (i,j,ℓ) is

$$p_{kij\ell} = \frac{1}{\beta}\left(\ln\left(\frac{u_{kij\ell}}{x_{ij\ell} u_{k0}}\right) + d_{kij}\right)\ \forall k \in V,\ (i,j) \in A,\ \ell \in L, \tag{5}$$

as a consequence of $u_{kij\ell}/u_{k0} = x_{ij\ell} \exp(\beta p_{ij\ell} - d_{kij})$. Constraint (4b) is a flow balance constraint, which ensures that the rates at which delivery tasks arrive and are selected for a given (i,j,ℓ) tuple are balanced with the rate at which delivery tasks experience a reduction in lead time. Constraints (4c) and (4d) guarantee this balance holds on the boundary of the maximum relative lead time. Constraint (4e) ensures that each of the $k \in V$ ad-hoc choice models, where k is the location of an ad-hoc courier when it arrives, have selection probabilities and a null probability (i.e. selecting no delivery task) that sum to 1. Finally, constraints (4f) and (4g) are non-negativity constraints.

4.2.1. Conic reformulation of the ad-hoc system

The objective function of formulation (4) is convex in the decision variables $\mathbf{x}$ and $\mathbf{u}$ by the preservation of convexity described in Boyd et al. (2004), but this formulation is not readily solvable by standard optimization software. We now present a conic reformulation of formulation (4), which is an extension of the reformulation given by Cao et al. (2022), so that the large-scale problem instances can be solved by optimization solvers such as MOSEK (MOSEK ApS, 2023). Firstly, we provide some preliminaries to the transformation. For $x_{ij\ell},\ u_{kij\ell} > 0$, let $-t_{kij\ell} \ge u_{kij\ell} \ln(u_{kij\ell}/x_{ij\ell})$ which holds if and only if $x_{ij\ell} \ge u_{kij\ell} \exp(t_{kij\ell}/u_{kij\ell})$. Let $-w_{k1} \ge u_{k0} \ln(u_{k0})$, which holds if and only if $1 \ge u_{k0} \exp(w_{k1}/u_{k0})$. Finally, let $-w_{k2} \ge -\ln(u_{k0})$, which holds if and only if $u_{k0} \ge \exp(w_{k2})$. The definition of an exponential cone in $\mathbb{R}^3$ is $\mathcal{K}_{\exp} := \mathrm{cl}\{(y_1,y_2,y_3) : y_1 \ge y_2 \exp(y_3/y_2), y_2 > 0\}$. Therefore, we have the following reformulation of (4) with a linear objective in a convex feasible region.

$$\min\ \frac{\gamma}{\beta} \sum_{k\in V} \mu_k \sum_{(i,j)\in A} \sum_{\ell\in L} (-t_{kij\ell} + d_{kij} u_{kij\ell}) - \frac{\gamma}{\beta} \sum_{k\in V} \mu_k \sum_{(i,j)\in A} (w_{k1} + w_{k2}) + \sum_{\ell\in L} \theta_\ell \sum_{(i,j)\in A} x_{ij\ell} \tag{6a}$$

$$\text{s.t. } x_{ij\ell} - x_{ij\ell-1} = \gamma \sum_{k\in V} \mu_k u_{kij\ell} - \lambda_{ij\ell-1} \qquad \forall (i,j) \in A,\ \ell \in L \tag{6b}$$

$$x_{ijL_{\max}} = \lambda_{ijL_{\max}} \qquad \forall (i,j) \in A \tag{6c}$$

$$\gamma \sum_{k\in V} \mu_k u_{kij\ell} \le x_{ij\ell} \qquad \forall (i,j) \in A,\ \ell \in L \tag{6d}$$

$$\sum_{(i,j)\in A} \sum_{\ell\in L} u_{kij\ell} + u_{k0} = 1 \qquad \forall k \in V \tag{6e}$$

$$(x_{ij\ell}, u_{kij\ell}, t_{kij\ell}) \in \mathcal{K}_{\exp} \qquad \forall k \in V,\ (i,j) \in A,\ \ell \in L \tag{6f}$$

$$(1, u_{k0}, w_{k1}) \in \mathcal{K}_{\exp} \qquad \forall k \in V \tag{6g}$$

$$(u_{k0}, 1, w_{k2}) \in \mathcal{K}_{\exp} \qquad \forall k \in V \tag{6h}$$

$$x_{ij\ell} \ge 0 \qquad \forall (i,j) \in A,\ \ell \in L \tag{6i}$$

$$u_{kij\ell} \ge 0 \qquad \forall k \in V,\ (i,j) \in A,\ \ell \in L. \tag{6j}$$

We have the following theorem.

Theorem 4.1. *For every optimal solution to formulation (6), there exists an optimal solution to (4) with equivalent* $\mathbf{x}^*$ *and* $\mathbf{u}^*$ *and the same objective value.*

For completeness, the detailed proof for Theorem 4.1 is provided in Appendix B. However, we would like to point out that this reformulation is an extension of the one presented by Cao et al. (2022), and our extension preserves their core results without having to fundamentally change their proof. Theorem 4.1 is important because it allows us to efficiently solve formulation (6) as a substitute to formulation (4).

4.3. Committed couriers

Unlike the ad-hoc couriers, committed couriers are part of a *closed system* as they do not enter or leave the system. In addition, we assume that committed couriers are not compensated with a per-task pricing scheme but rather with a guaranteed hourly wage f. Let M be the fleet size of committed couriers to hire, which is a decision variable. The rate at which the platform assigns delivery tasks from arc (i,j) with relative lead time ℓ to committed couriers is denoted by $r_{ij\ell}$, which is analogous to $\sum_{k\in V}\mu_k u_{kij\ell}$ in the case of ad-hoc couriers. Intuitively, $\sum_{\ell\in L} r_{ij\ell}$ can be interpreted as the rate at which committed couriers travel on arc (i,j) while actively carrying a delivery task emanating from i and going to j. On the other hand, we let z_{ij} be the empty-miles or repositioning rate of committed couriers, that is, the rate that committed couriers move from i to j without carrying a delivery task specific to (i,j).

For a system that only utilizes committed couriers, we can treat $\sum_{\ell\in L} r_{ij\ell}$ as an exogenous parameter for all (i,j), as it is given by the demand rate of delivery tasks on the arc (i,j) (assuming the penalty cost for expired tasks is large enough so that no tasks are dropped). In order to find the minimum number of committed couriers needed to serve the demand, we can minimize the total empty miles, subject to balancing assigned flows arising from $\sum_{\ell\in L} r_{ij\ell}$ for all (i,j). First, let $s_i = \sum_{j\in V}\sum_{\ell\in L}(r_{ij\ell} - r_{ji\ell})$ $(\forall i \in V)$ be the aggregate rate of assigned flow out of (or into) location i. Let $V^+ := \{i : s_i > 0\}$ and $V^- := \{i : s_i < 0\}$. We then have the following formulation:

$$\min \sum_{i\in V^-}\sum_{j\in V^+} z_{ij}c_{ij} \tag{7a}$$

$$\text{s.t.} \sum_{j\in V^+} z_{ij} = -s_i \quad \forall i \in V^- \tag{7b}$$

$$\sum_{i\in V^-} z_{ji} = s_j \quad \forall j \in V^+ \tag{7c}$$

$$z_{ij} \geq 0 \quad \forall i \in V^-,\ j \in V^+. \tag{7d}$$

The objective (7a) is to minimize the total time spent traveling empty miles where c_{ij} is the travel time on arc (i,j) (not to be confused with d_{kij} of the previous section, the utility cost of distance to ad-hoc couriers), constraints (7b) and (7c) ensures that the amount of empty mile flow entering or leaving a vertex is equal to the cumulative amount of assigned flow entering or leaving it and constraint (7d) ensures non-negative empty miles flow. formulation (7) is equivalent to the following simpler formulation which explicitly references $r_{ij\ell}$ outside of set notation:

$$\min \sum_{i\in V}\sum_{j\in V} z_{ij}c_{ij} \tag{8a}$$

$$\text{s.t.} \sum_{j\in V}\Big(\sum_{\ell\in L} r_{ij\ell} + z_{ij}\Big) = \sum_{j\in V}\Big(\sum_{\ell\in L} r_{ji\ell} + z_{ji}\Big) \quad \forall i \in V \tag{8b}$$

$$z_{ij} \geq 0 \quad \forall (i,j) \in A. \tag{8c}$$

Objective (8a) is the same as before, and constraint (8b) is a flow balance equation, ensuring the total rate of committed couriers leaving a location (both assigned rate and empty miles rates) is equal to that which is entering a location. We have the Theorem 4.2, which is sufficient to prove that the formulations are equivalent.

Theorem 4.2. *In an optimal solution to formulation (8), at most one of the following conditions are true for any vertex $i \in V$:*

1. $\sum_{j\in V} z^*_{ij} > 0$,
2. $\sum_{j\in V} z^*_{ji} > 0$.

We provide the formal proof of Theorem 4.2 in Appendix B. Theorem 4.2 implies that formulations (7) and (8) are equivalent, as for any $i \in V^-$, we have $\sum_{j\in V^+} z^*_{ij} \geq 0$ and $\sum_{j\in V^+} z^*_{ij} = 0$, and the reverse is true for $i \in V^+$. Therefore, we are able to use the main idea of formulation (8) to describe the committed couriers in terms of the capacity constraint which is employed in conjunction with (8b); we have

$$\sum_{(i,j)\in A} c_{ij}\cdot\Big(\frac{1}{\eta}\sum_{\ell\in L} r_{ij\ell} + z_{ij}\Big) \leq \rho M. \tag{9}$$

To accommodate the fact that a committed courier can perform multiple delivery tasks at once (a major advantage of committed couriers is that the platform has control over their routing), we introduce a factor $\frac{1}{\eta}$ with $\eta \geq 1$ and use $\frac{1}{\eta}\sum_{\ell\in L} r_{ij\ell}$. Lastly, we have ρ which is the multiplier converting the fleet size M into the capacity of the fleet in terms of the total rate at which each courier can be assigned delivery and relocation tasks.

We can formulate a pure committed system as:

$$\min\ f M + \sum_{\ell \in L} \theta_\ell \sum_{(i,j)\in A} x_{ij\ell} \tag{10a}$$

$$\text{s.t. } x_{ij\ell} - x_{ij\ell-1} = r_{ij\ell} - \lambda_{ij\ell-1} \qquad \forall (i,j) \in A,\ \ell \in L \tag{10b}$$

$$x_{ijL_{\max}} = \lambda_{ijL_{\max}} \qquad \forall (i,j) \in A \tag{10c}$$

$$\sum_{(i,j)\in A} c_{ij} \cdot \Big(\frac{1}{\eta} \sum_{\ell \in L} r_{ij\ell} + z_{ij}\Big) \le \rho M \tag{10d}$$

$$\sum_{j\in V} \Big(\sum_{\ell \in L} r_{ij\ell} + z_{ij}\Big) = \sum_{j\in V} \Big(\sum_{\ell \in L} r_{ji\ell} + z_{ji}\Big) \quad \forall i \in V \tag{10e}$$

$$M \in \mathbb{Z}_+ \tag{10f}$$

$$z_{ij} \ge 0 \quad \forall (i,j) \in A \tag{10g}$$

$$r_{ij\ell} \ge 0 \quad \forall (i,j) \in A,\ \ell \in L \tag{10h}$$

where the objective (10a) is to minimize the total hourly wage paid to committed couriers and the sum of the late penalties, constraints (10b) and (10c) are flow balance constraints similar to formulation (4), constraint (10d) is the capacity constraint which ensures that the operations of committed couriers must be feasible for the fleet size, constraint (10e) ensures the total rate of flow of committed couriers in and out of locations are equal, and constraints (10f), (10g), and (10h) constrain the decision variables. At optimality, the term $\sum_{j\in V} z^*_{ij} c_{ij}$ for all $i \in V$ with be at its minimum for the associated set of $r^*_{ij\ell}$, and thus the solution corresponds to the optimal solution to formulations (7) and (8).

4.4. Hybrid systems with ad-hoc and committed couriers

Combining the fluid models for ad-hoc couriers in Section 4.2 and committed couriers in Section 4.3, we now present the full formulation for a hybrid system below. This formulation can be converted to a conic optimization problem in the exact same way as in Section 4.2.1 so that large-scale problem instances can be readily solved by convex optimization solvers.

$$\min\ \gamma \sum_{k\in V} \sum_{(i,j)\in A} \sum_{\ell \in L} \mu_k u_{kij\ell} p_{kij\ell} + f M + \sum_{\ell \in L} \theta_\ell \sum_{(i,j)\in A} x_{ij\ell} \tag{11a}$$

$$\text{s.t. } x_{ij\ell} - x_{ij\ell-1} = \gamma \sum_{k\in V} \mu_k u_{kij\ell} + r_{ij\ell} - \lambda_{ij\ell-1} \qquad \forall (i,j) \in A, \ell \in L \tag{11b}$$

$$x_{ijL_{\max}} = \lambda_{ijL_{\max}} \qquad \forall (i,j) \in A \tag{11c}$$

$$\gamma \sum_{k\in V} \mu_k u_{kij\ell} + r_{ij\ell} \le x_{ij\ell} \qquad \forall (i,j) \in A,\ \ell \in L \tag{11d}$$

$$\sum_{(i,j)\in A} \sum_{\ell \in L} u_{kij\ell} + u_{k0} = 1 \qquad \forall k \in V \tag{11e}$$

$$\sum_{(i,j)\in A} c_{ij} \cdot \Big(\frac{1}{\eta} \sum_{\ell \in L} r_{ij\ell} + z_{ij}\Big) \le \rho M \tag{11f}$$

$$\sum_{j\in V} \Big(\sum_{\ell \in L} r_{ij\ell} + z_{ij}\Big) = \sum_{j\in V} \Big(\sum_{\ell \in L} r_{ji\ell} + z_{ji}\Big) \qquad \forall i \in V \tag{11g}$$

$$M \in \mathbb{Z}_+ \tag{11h}$$

$$z_{ij} \ge 0 \qquad (i,j) \in A \tag{11i}$$

$$x_{ij\ell} \ge 0,\ r_{ij\ell} \ge 0 \qquad \forall (i,j) \in A,\ \ell \in L \tag{11j}$$

$$u_{kij\ell} \ge 0 \qquad \forall k \in V,\ (i,j) \in A,\ \ell \in L. \tag{11k}$$

Notice that the constraint set of formulation (11) is the combination of formulations (4) and (10). *It is entirely possible that the optimal solution to the hybrid formulation is one that uses one type of courier exclusively.* Specifically, the optimal solutions to formulations (4) and (10) are feasible solutions to formulation (11). By construction, the optimal cost of the hybrid formulation can be at most as costly as the best performing pure solution $c^*_H \le \min(c^*_A, c^*_C)$, where c^*_H, c^*_A, and c^*_C are the optimal costs to formulations (11), (4), and (10) respectively.

5. Analysis and numerical experiments

In this section we will analyze and discuss the models presented in Section 4 as well as perform numerical experiments on instances designed to reflect real-world conditions (motivated by crowdsourced same-day delivery). Furthermore, we provide some computational results on the scalability of our models showing their usefulness in making planning decisions for practically sized problems. This section is organized as follows. Section 5.1 discusses the fundamental benefits of a hybrid system over pure committed

and ad-hoc systems, and analyzes the conditions under which these benefits may be observed. Section 5.2 highlights the behavior discussed in Section 5.1 with numerical experiments that are set in the crowdsourced same-day delivery domain. We construct instances that are inspired by real crowdsourced delivery platforms and similar work which has leverage real-world demand data in order to perform this analysis. We study the effect of varying committed courier wages (Sections 5.2.1 and 5.2.3), the ad-hoc supply-to-demand ratio (Section 5.2.2), and the spatial imbalances of demand flow and ad-hoc arrivals (Section 5.2.4). Finally, in Section 5.3 we provide instance characteristics and solve times for instances up to 36 locations, and argue that our model can be feasibly implemented for many practical region sizes.

5.1. Fundamental benefits of a hybrid system

In this section, we analyze market conditions under which a hybrid delivery system using both ad-hoc and committed couriers can outperform any pure delivery system that uses only one type of courier. We say that the courier capacity planning problem (formulation (11)) generates a *hybrid solution* if the delivery demand is served by both types of couriers, i.e., $M^* > 0$ and $\sum_{k\in V}\sum_{(i,j)\in A}\sum_{\ell\in L} u^*_{kij\ell} > 0$. The main mechanisms by which hybrid solutions reduce the cost over pure systems are:

1. Using committed couriers to relieve the pressure on the ad-hoc subsystem when the ad-hoc subsystem is utilized near its maximum capacity;
2. Selectively allocating tasks between the committed and ad-hoc subsystems to reduce the spatial mismatch between courier supply and delivery demand;
3. In the case of time-varying demand and ad-hoc supply — allocating delivery tasks between the two subsystems to manage the temporal imbalance between supply and demand.

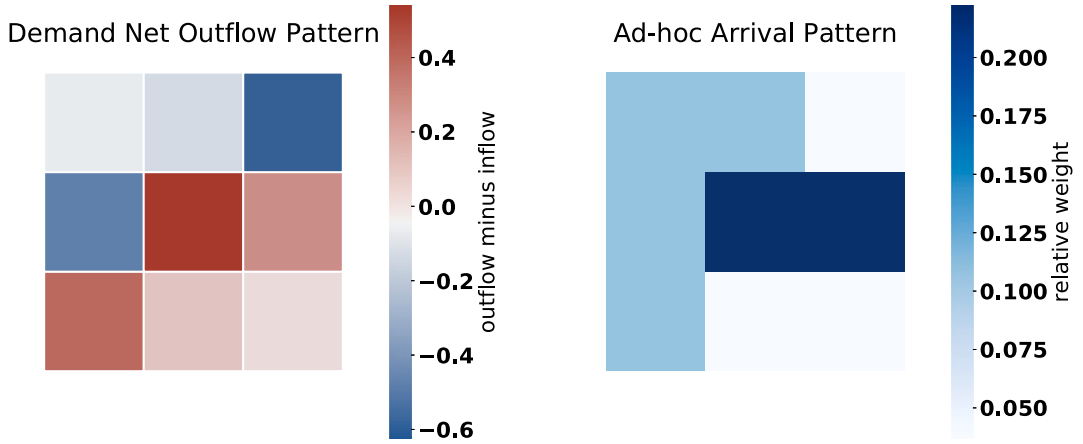
Because we assume demand and supply are time-homogeneous (see the discussion in Section 4), we will focus on the first two points. To point (1), consider a scenario where the total demand for delivery tasks is approaching the total supply of ad-hoc couriers in a pure ad-hoc system, so almost all ad-hoc couriers become busy. In this case, the optimal null probability $\sum_{k\in V} u^*_{k0}$ to the pure ad-hoc system approaches 0 as the total capacity $\gamma\sum_{k\in V}\mu_k$ approaches $\sum_{(i,j)\in A}\sum_{\ell\in L}\lambda_{ij\ell}$ from above (assuming a sufficiently large expired penalty cost). Furthermore, we recognize that the prices for ad-hoc couriers will reveal limiting behavior as $\sum_{k\in V} u^*_{k0} \to 0$. This is due to the fact that as u_{k0} decreases $p_{kij\ell}$ grows to infinity. Of course, we can imagine a case where $\gamma\sum_{k\in V}\mu_k \gg \sum_{(i,j)\in A}\sum_{\ell\in L}\lambda_{ij\ell}$ and all demand can be served by the ad-hoc couriers without incurring the problematic behavior that occurs when the ad-hoc subsystem reaches its capacity. In reality, such a situation is unlikely, as a large over-supply of ad-hoc couriers is not sustainable: Over time ad-hoc couriers would cease to arrive in the system as they do not find work, thus re-balancing it. As such, we believe it is reasonable to assume that if $\mu > \lambda$, then both values are of similar magnitude, which makes it unlikely that a pure ad-hoc system is able to avoid this limiting price behavior. In a hybrid system, using committed couriers avoids this by using committed couriers to remove the burden from the ad-hoc system at a fixed cost per courier f.

Next, we have point (2), which can take different forms based on the instance considered. Imagine a case where the committed courier wage f has a magnitude that is large in comparison to the average price of delivery tasks in the ad-hoc system, but not so high that committed couriers are not hired at all. The optimal solution to the hybrid formulation can set prices in such a way, that the demand allocated to the committed couriers is nearly balanced and thus $\sum_{(i,j)\in A} z_{ij}$ is close to zero. In this manner, the hybrid formulation uses ad-hoc pricing as a mechanism to "smooth out" the demand allocated to committed couriers, which reduces their empty miles (i.e. utilizes them more efficiently, as their time is on average more expensive). We can also imagine the reverse case, where f is small in comparison to the average price for ad-hoc couriers, but not so that $\sum_{k\in V}\sum_{(i,j)\in A}\sum_{\ell\in L} u_{kij\ell} = 0$. Then, the hybrid formulation can utilize the empty-miles flow variables to reposition committed couriers in a manner that makes it more likely that ad-hoc couriers serve delivery tasks that are close to them (i.e., delivery tasks they are more incentivized to take for a lower price, based on distance aspect of their utility). In both cases, one type of courier is used to improve the cost-effectiveness of the other, reducing the total cost operating cost.

5.2. Numerical experiments

The purpose of this section is to illustrate the utility of our model in a real-world same-day crowdsourced delivery setting. We were in contact with an actual crowdsourced delivery platform and constructed this setting to best mirror what we witnessed in practice. Although our simulations do not use the exact parameters used by the company (for confidentiality reasons), these settings were chosen so that the per delivery task compensation for ad-hoc couriers was reflective of the real world (given appropriate committed courier wages, see Fig. 3(a)). Additionally, our travel times are inspired by our previous work (Behrendt et al., 2022), which used real-world last-mile delivery data to construct instances. Thus, our travel times are on the order of 20 min to an hour and represent same-day delivery orders with relatively short time windows. We chose a 3×3 grid (i.e., a square map with 9 locations) to represent locations as we believe this is a reasonable discretization of an area pertaining to our travel distances (which is supported by the service area partitioning instances of Banerjee et al. (2022a)). However, we explore how the computation time of our model is affected by larger instances. We first describe the creation of the instances in terms of the demand arrival and ad-hoc arrival patterns that we use in the experiments for the remainder of this section.

Fig. 2(a) shows the net outflow (total outflow minus the inflow) of demand from each location i given by $\sum_{j\in V}\sum_{\ell\in L}\lambda_{ij\ell} - \lambda_{ji\ell}$. As such, we can see that there are locations that, on average, are sources or sinks. This is not to say that locations that have net



(a) Demand Pattern. (b) Ad-hoc Arrival Pattern.

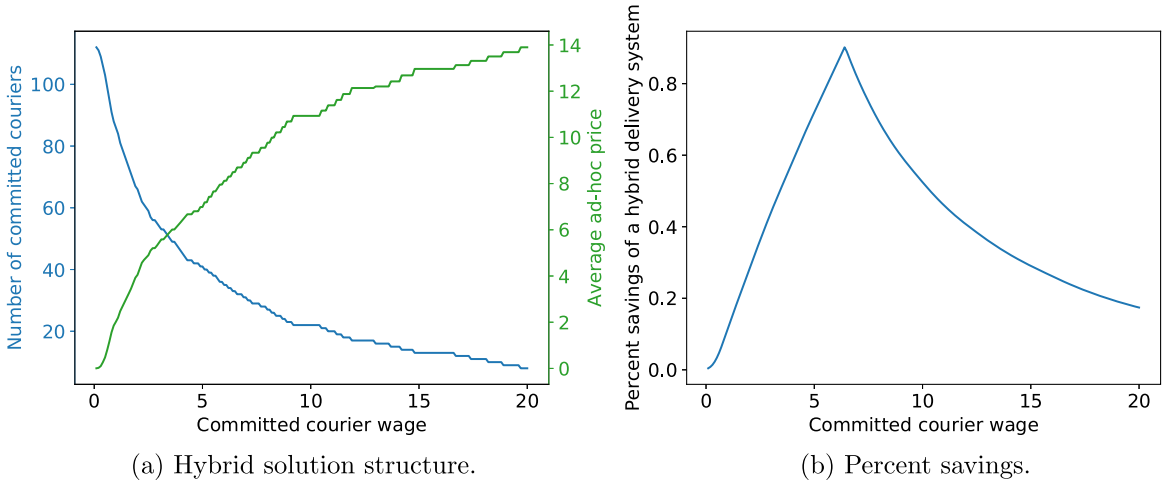
Fig. 2. Demand outflow ($\sum_{j\in V}\sum_{\ell\in L}\lambda_{ij\ell}-\lambda_{ji\ell}$) and ad-hoc couriers' arrival (μ_k) patterns.

positive demand flowing from them do not have any arcs with demand arriving to that location, and vice versa. For information on the exact arcs present in Fig. 2(a), as well as the algorithm we use to randomly generate a set of arcs, please see Appendix A. Fig. 2(b) illustrates the spatial distribution of ad-hoc courier arrivals. Note that while these spatial distributions have some commonalities, in the sense that there are locations that see a high demand outflow as well as ad-hoc courier arrivals (i.e. a good match between supply and demand), this is not always the case, meaning ad-hoc couriers may have to select orders outside of their arrival location. Finally, the most important thing to note is that what we have depicted here are *patterns*; the underlying distributions are weighted such that $\sum_{k\in V}\mu_k=1$ and $\sum_{(i,j)\in A}\sum_{\ell\in L}\lambda_{ij\ell}=1$. As such, we are able to multiply these patterns with scale parameters that increase the magnitude of arrivals but maintain the spatial distribution for our examples.

Specifically, we let $\beta=0.5$. $\eta=1.5$, $\gamma=1.5$, and $\rho=1$. As for relative lead times, we have the expiration time as $L_{\min}=-5$ with penalty $\theta_{L_{\min}}=100$. Additionally, our late penalties are $\theta_\ell=4,3,2,1$ for $\ell=L_{\min}+1,\dots-1$. That is, we have increasing late penalties as the lateness increases, with a large expiration penalty. Finally, $L_{\max}=5$ is the longest lead time a delivery task can arrive with (however, delivery tasks can arrive with relative lead times [1, 5]). As we have mentioned before, we re-scale the lead times so $L_{\min}=0$, but this does not affect the interpretation of our results. Let $d(i,j)$ be the Euclidean distance from location i to j in our 3×3 grid example. Then, we let the utility cost of distance for the ad-hoc couriers arriving at location k be $d_{kij}=5+1\cdot[d(k,i)+d(i,j)]$. In terms of the actual travel times for locations for committed couriers, we have $c_{ij}=20+10\cdot d(i,j)$. As we see, both of these terms have a constant term referring to the inherent utility cost/travel time associated with serving a delivery task/traveling within a particular location on the grid. The travel time is then at least 20 min, which we can think of as pickup and delivery times and/or intra-location travel, with an additional 10 min per unit distance traveled. As such, our travel times are always less than one hour. Similarly, the base utility cost for ad-hoc couriers serving an order is 5 with an additional 1 per unit distance. In the following subsection, we investigate the selection for the committed couriers' wage f, and in turn, the compensation ratio between ad-hoc and committed couriers given that the ad-hoc pricing parameters are held constant.

5.2.1. Ad-hoc to committed wage ratio effect on solution structure and savings

In Fig. 3 we vary the committed wage from .1 to 20, in order to highlight how the solution structure of formulation (11) changes in terms of the relative capacities allocated to each subsystem, and the cost savings associated with using a hybrid system. Let, $\mu=\gamma\sum_{k\in V}\mu_k$ and $\lambda=\sum_{(i,j)\in A}\sum_{\ell\in L}\lambda_{ij\ell}$. Let, $\lambda=3$ and $\mu=4$ be the multipliers for the demand and ad-hoc supply patterns. Specifically, this means that the ad-hoc courier supply is larger than the demand, and a pure ad-hoc system is a feasible solution. In this manner, we are able to view the benefits of a hybrid system in the case where both pure committed and pure ad-hoc systems are feasible. Fig. 3(a) showcases the optimal solution to formulation (11), where the left axis is the number of committed couriers hired M^* and the right axis is the average price of delivery tasks selected by ad-hoc couriers, $\sum_{k\in V}\sum_{(i,j)\in A}\sum_{\ell\in L}u_{kij\ell}p_{kij\ell}$. When the committed wage is negligible the optimal solution to the hybrid formulation is a pure committed system. As the wage increases the number of committed couriers to hire decreases. However, we see that even with a small committed courier wage (resulting in a large fleet of committed couriers), employing ad-hoc couriers still provides benefits. A platform can price orders lower than the wage to incur savings in the (low probability) case that an ad-hoc courier selects the order. This is represented in practice when there is an ad-hoc courier for which a delivery task results in very little deviation from their pre-planned route. Interestingly, the decay rate of the optimal fleet size and the growth rate of the ad-hoc prices are decreasing. This highlights that even when the committed courier wage is large in comparison to the ad-hoc prices, it remains optimal to hire a small fleet of committed couriers (for example,



(a) Hybrid solution structure. (b) Percent savings.

Fig. 3. The left figure depicts the optimal number of committed couriers and the optimal average ad-hoc price in a hybrid system, while the right depicts the average percent savings of a hybrid system, both as a function of the committed courier wage.

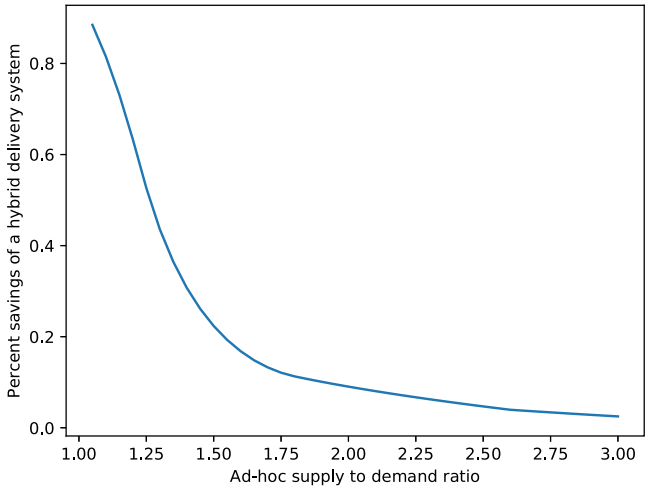


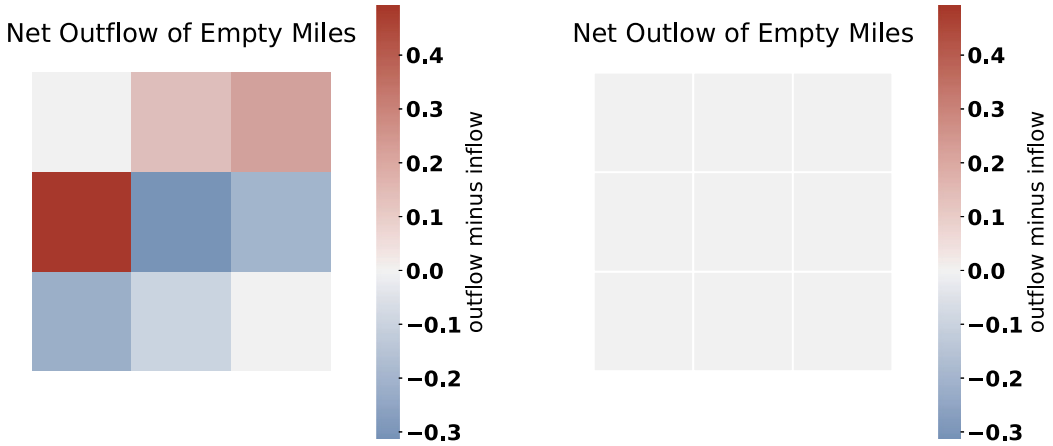
Fig. 4. Average savings of a hybrid delivery system over the best performing pure delivery system for varying ad-hoc capacity to demand ratios ($\gamma \sum_{k\in V} \mu_k / \sum_{(i,j)\in A} \sum_{\ell\in L} \lambda_{ij\ell}$).

$M^* = 2$ at $f = 20$), even though hiring zero is a feasible solution. In practice, this translates to the ability of a small, efficient fleet of committed couriers to operate on the margins of the ad-hoc capacity which removes the expensive nature of operating a pure ad-hoc system (we explore this notion further in Section 5.2.2).

In Fig. 3(b), we show the magnitude of savings of a hybrid system on the y-axis, associated with the solutions depicted in Fig. 3(a). The savings of a hybrid system is calculated as $(\min(z_C^*, z_A^*) - z_H^*)/z_H^*$ which is the percent improvement of a hybrid system over the best performing pure system. Until $f \approx 6.5$, the best performing pure system is a committed system, and beyond that, it is a pure ad-hoc system; this fact is responsible for the sudden reverse in the sign of the gradient. For this specific example, savings of more than 80% may be experienced in situations where the committed courier wage is low. In actuality, committed couriers tend to be more expensive than ad-hoc couriers on a per-delivery task basis. As such, the results for wages 15 to 20 represent a more realistic scenario. Even though the marginal decrease of the savings decreases as the committed wage grows, we still see potential savings of around 20% at $f = 20$. Of course, as these results are generated using our fluid model, these values (and most of the numerical results in this section) should be interpreted as *upper bounds* on the potential improvements of a hybrid system in actual operation, with similar parameter values.

5.2.2. Ad-hoc supply to demand ratio effect on savings

We now investigate the magnitude of savings that occur due to our first point, that hybrid systems use committed couriers to avoid the limiting behavior of the ad-hoc pricing channel. Here, we let the base multiplier for the ad-hoc supply pattern be $\mu = 2$ and



(a) $f = 1$ (b) $f = 10$

Fig. 5. Figures depict the empty-mile outflow minus the inflow ($\sum_{j \in V}(z^*_{ij} - z^*_{ji})$) of empty miles for each location $i \in V$.

for the demand $\lambda = 3$. The x-axis for Fig. 4 is a scale multiplier to μ, say ϕ. Specifically, we have $\gamma \sum_{k \in V} \mu_k / \sum_{(i,j) \in A} \sum_{\ell \in L} \lambda_{ij\ell} = \phi$, meaning that ϕ is the ratio of the ad-hoc supply to demand. When $\phi = 1$, the capacity of the ad-hoc system is equal to the amount of demand (recall that $\gamma = 1.5$). When $\phi = 2$, the ad-hoc capacity is double that of the demand (and it is the exact parameters used in the creation of Fig. 3). We only have values where $\phi \geq 1$, as those are the only cases where a pure ad-hoc solution is feasible. Here we let the committed wage $f = 15$ and, unlike Fig. 3(b), we have the best performing pure system being the ad-hoc system across the entire x-axis domain. A hybrid solution has the largest benefit over a pure ad-hoc system when the ad-hoc system is near the margins of capacity. The decrease in savings as ϕ increases is due to the fact that the cost of a pure ad-hoc system also decreases as ϕ increases. Interestingly, even when the ad-hoc supply is three times that of the demand, we still see savings of around 2.5%. This indicates that there are benefits to a hybrid system beyond avoiding the limiting behavior of the ad-hoc pricing channel which we illustrate with the following example.

5.2.3. Committed wage effect on repositioning (empty-miles)

This subsection relates to our second point, that allocating delivery tasks based on the spatial distribution of ad-hoc supply and demand reduces the operating cost of a hybrid system. In this example, we use the exact same parameters as Section 5.2.1 and show results for committed wage $f = 1$ and $f = 10$. Fig. 5 illustrates the cumulative outflow of empty miles from each location for the committed subsystem under different committed wages. In Fig. 5(a) we see a substantial amount of empty relocation of committed couriers when the wage f is relatively low compared to the pricing of ad-hoc couriers. The distribution in Fig. 5(a) is almost the direct inverse of the cumulative outflow of demand shown in Fig. 2(a). It is optimal for a hybrid system to relocate committed couriers for a low wage penalty in order to reduce the average prices in the ad-hoc channel. In Fig. 5(b), we see the opposite. The optimal cumulative outflow of empty miles is negligible when the committed wage is sufficiently high. This indicates that the ad-hoc subsystem is used as a mechanism to smooth out the demand allocated to the committed couriers, reducing the empty relocation to zero in order to avoid "wasted" committed wage payout.

5.2.4. Sensitivity to demand and ad-hoc patterns

In this section we present how the cost gap between a hybrid system and pure committed and ad-hoc systems changes with respect to varying levels of imbalance in the spatial parameters of instances. First, we start by investigating the effect of an imbalanced demand pattern, that is, the imbalance between positive net outflows and negative net outflows, as depicted in 2(a). Firstly, we generate 100 instances with varying levels of imbalance. We use the same pricing parameters as the rest of the sections, $f = 15$, $\mu = 4$, and $\lambda = 3$. Additionally, we set the ad-hoc arrival pattern to be uniform (no longer using 2(b)) and hold it constant over all generated instances. We then randomly generate instances by the manner described in Appendix A. Next, we describe the level of imbalance for a given demand pattern quantitatively with the following procedure. Let N be the set of all locations with positive net outflow and M be the set of all locations with negative net outflow. Let, $\mathbf{w}$ be the absolute value of the net outflow for all $N \cup M$. We then solve the following simple linear program:

$$\min \sum_{i \in N} \sum_{j \in M} x_{ij} c_{ij} \tag{12a}$$

$$\text{s.t.} \sum_{j \in M} x_{ij} = w_i \quad \forall i \in N \tag{12b}$$

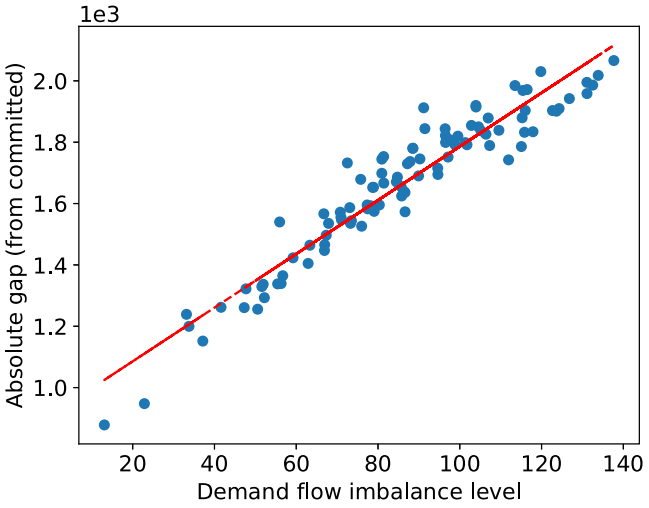


Fig. 6. Absolute gap between a hybrid system and pure committed system with respect to the demand flow imbalance level.

$$\sum_{i \in N} x_{ij} = w_j \quad \forall j \in M \tag{12c}$$

$$x_{ij} \geq 0 \quad \forall i \in N,\ j \in M. \tag{12d}$$

The parameter c_{ij} is the travel time from location i to j from previous sections, and here $\mathbf{x}$ is a decision variable representing the flow from location i to j. Thus, the optimal solution to this problem is the minimum weighted travel time to match all positive net outflows with negative net outflows, which we call an *imbalance level*. True to its name, a larger imbalance level corresponds to a larger weighted travel time between positive net outflows and negative net outflows, while a value of 0 corresponds to a perfectly balanced demand pattern. We then calculated the imbalance level for 100 instances, and solved for the optimal cost of a hybrid system (z_H^*) and the optimal cost of a pure committed system (z_C^*). We define the absolute gap as $z_H^* - z_C^*$ (note that we do not include the pure ad-hoc system in this calculation because the ad-hoc system is independent of this type of imbalance). Fig. 6 depicts the results, and we see that the absolute gap is increasing as the demand flow balance level increases. This is indicative that a hybrid system is more resilient to the imbalance in demand patterns than its committed counterpart.

We perform a similar procedure to investigate the imbalance between demand outflow and ad-hoc arrivals. We use the same parameter values, but now set the demand outflow (not net outflow) of each location to be equal and hold it constant. As for the ad-hoc arrival patterns, we randomly generate 100 samples according to the discussion in Appendix A. We calculate an imbalance level in a similar manner as before. Let N be all locations on the grid, let $\mathbf{w}^d$ be the vector of the total outflow of demand at each location, and let $\mathbf{w}^a$ be the vector of the ad-hoc arrival rate at each location. We have the following formulation:

$$\min \sum_{i \in N} \sum_{j \in N} x_{ij} c_{ij} \tag{13a}$$

$$\text{s.t.} \sum_{j \in M} x_{ij} \leq w_i^a \quad \forall i \in N \tag{13b}$$

$$\sum_{i \in N} x_{ij} = w_j^d \quad \forall j \in N \tag{13c}$$

$$x_{ij} \geq 0 \quad \forall i \in N,\ j \in M. \tag{13d}$$

The optimal solution to formulation (13) then represents the minimum weighted travel time to match the ad-hoc arrival weights with the demand outflow. Note constraint (13b) comes from the fact that we have $\mu > \lambda$, and all of the demand can be served with a portion of ad-hoc arrivals. Again, we calculated this ad-hoc to demand imbalance level for out 100 new instances and calculated the absolute gap between a hybrid system (z_H^*) and a pure ad-hoc system (z_A^*). In this case, we leave out the pure committed system because it is independent of this type of imbalance. The results are depicted in Fig. 7, where we observe that as the ad-hoc to demand imbalance level increases, so does the absolute gap between a hybrid system and a pure ad-hoc system. Again, this reinforces our intuition that hybrid systems are more resilient in the face of spatial imbalances that arise in pure committed and pure ad-hoc systems.

5.3. Computational scalability

In this section, we explore the computational scalability of our models beyond the case with 9 locations. For each $n \times n$ instance, we let there be n^3 arcs present. Furthermore, we use all of the same parameters as the previous sections, the only thing that changes

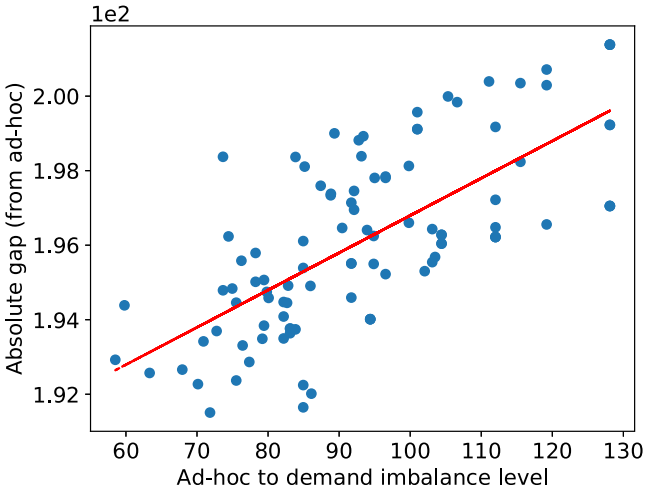


Fig. 7. Absolute gap between a hybrid system and pure ad-hoc system with respect to the ad-hoc to demand imbalance level.

Table 3
Instance characteristics and solve times for increasing grid size.

Grid size	Variables	Linear constraints	Cones	Solve time (s)
2 × 2	2298	1637	648	2.22
3 × 3	23,708	16,327	7308	10.25
4 × 4	128,610	87,377	40,992	158.49
5 × 5	482,652	325,751	156,300	625.95
6 × 6	1,428,410	960,517	466,632	3364.33

is the travel times (and naturally, the prices) are larger on average, as we have the same distance between each cell as the 3 × 3 case, but more cells. Each of these instances is solved with the MOSEK solver (MOSEK ApS, 2023) to optimality.

Table 3 shows our results. Going from our smallest instance (4 locations) to our largest (36 locations) we see that the number of decision variables moves from the thousands to the millions. This large increase in dimensionality is responsible for increased solve times, but while solve times increase rapidly, our largest instance is still solvable in under an hour. Recall that we are solving a planning problem that does not need to be solved during the operating period. As such, computation times less than a day are acceptable (e.g., planning the day before operations). As we noted before, the work of Banerjee et al. (2022a) was focused on service partitioning for same-day delivery and used 16 locations to model an area of 194.5 square miles. Thus, the manner in which our model scales is promising, as it indicates that we are able to make high-level task assignment, repositioning, and fleet sizing decisions for committed couriers along with ad-hoc pricing decisions for markets the size of a large city in a matter of hours.

6. Discussion

In this paper, we have provided strategic capacity planning models for crowdsourced delivery platforms, specifically for the fleet sizing of a centrally scheduled committed courier fleet and pricing policies for decentralized ad-hoc couriers. We then studied a hybrid delivery system that utilizes both types of couriers. We propose an optimization model for the hybrid system that allows for the joint planning of the committed and ad-hoc couriers. By analyzing this model, we found that introducing committed couriers to a pure ad-hoc system can mitigate the limiting behavior of prices when the ad-hoc system is utilized to its maximum capacity and avoid higher prices arising from tight relative lead times. Furthermore, the introduction of ad-hoc couriers to a pure committed system provides the opportunity to increase the efficiency of committed couriers by allowing more flexibility in which orders are assigned to them, thus reducing their empty miles. Through numerical examples, we illustrated that a hybrid delivery system has the potential to outperform either pure system along a large range of compensation ratios between ad-hoc and committed couriers. That is, even in cases where one type of courier may be significantly cheaper to use on average, the two types of couriers can be used in a complementary fashion to see further savings; either by using committed couriers to alter the spatial demand patterns allocated to ad-hoc couriers or vice versa.

Crowdsourced delivery platforms can utilize the insight this work has yielded. Specifically, cost savings could be experienced by platforms that operate with a pure ad-hoc system that is operating at the margins of its delivery capacity. By sourcing a small fleet of committed couriers in an urban area through crowdsourced shift sign-ups, platforms can maintain their main business model of non-employee couriers (and all the benefits and risks associated) while taking pressure off of their ad-hoc channel for a fixed cost. On the other hand, platforms that currently operate a pure committed system (in the extreme case, fleets of employee couriers) must first build a healthy market of ad-hoc couriers, which may take months or years. For large companies with name recognition and a

good reputation, this is less of a hurdle and such an initiative could pay itself off in the mid-term horizon. Of course, acquisitions of existing ad-hoc platforms offer the opportunity to immediately capitalize on an existing base of ad-hoc couriers. Ultimately, the hybrid utilization of committed and ad-hoc couriers has the potential to improve the profit of crowdsourced delivery platforms significantly. However, because our paper uses a stylized fluid model, more work needs to be done to verify that these theoretical improvements could be achieved in actual operations.

As we have mentioned, the related literature on crowdsourced delivery typically characterizes decentralized couriers as self-scheduling and assumes that orders are platform-assigned. As a result, self-assigning couriers have been understudied and underutilized. Self-scheduling couriers prefer to set the times in which they work at their whim; it is only natural that they would have a proclivity for choosing which orders they serve. An ad-hoc system supports these types of couriers naturally, while a platform-assigned system must learn courier preferences through trial and error. In platform-assignment, however, it is ultimately impossible to remove the dependencies arising from the manner in which orders are offered to couriers. For example, three orders may be offered and subsequently rejected, while the courier selects the fourth order offered. The assumption that this is reflective of their preference is inappropriate, as the courier's choice may be influenced by the three previous alternatives, may simply be an avoidance of rejection penalties, or may have been made for any number of reasons. An ad-hoc system, on the other hand, can learn individual courier preferences by presenting all available options. The information learned can be used to curate personalized menus of orders with custom prices, which can benefit both the couriers and the platform. As such, an ad-hoc system can still have influence over self-assigning couriers, in the same way a platform-assigned system may, in order to prioritize urgent orders. This problem setting is a promising area for future work, as marginally increasing the level of control allocated to the platform while preserving courier freedom may be able to improve profit for the platform while providing value to couriers. Another interesting question surrounds the optimal allocation of delivery tasks to the ad-hoc or committed subsystems within a hybrid system. Finding a partition of delivery tasks (even for deterministic demand) between the two systems is not trivial, as the optimal pricing policy will change based on which orders are present. Overall, there is ample opportunity for further research into the management of self-assigning couriers alone and the interplay between them and platform-assigned couriers in the context of a mobile delivery platform.

CRediT authorship contribution statement

Adam Behrendt: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Martin Savelsbergh:** Conceptualization, Methodology, Writing – review & editing. **He Wang:** Conceptualization, Methodology, Supervision, Writing – review & editing, Funding acquisition.

Acknowledgements

This work was primarily funded by the National Science Foundation, award number 2145661.

Appendix A. Instance generation

In this section, we will provide information on the main instance used in Section 5.2 as well as the functions used to create it and the others in Section 5.2.4. Firstly, for an $n \times n$ grid, we let there be n^3 arcs. For our 3×3 example this means we have 27 arcs. We define what we call an arc weight array (`arc_weight`) as an array of length n^2 that sums to 1, where each entry represents a location and the value is the proportion of the total arcs that emanate from that location. The arc weight array we used in Section 5.2 was `arc_weight` =[3/27, 1/27, 1/27, 3/27, 6/27, 6/27, 3/27, 3/27, 1/27]. Given an arc weight array, we generate that proportion of arcs emanating from the location and randomly assign the arcs to end locations according to a sink probability array defined for each of the n^2 locations (`sink_prob`). The `sink_prob` array sums to 1 and represents the probability that an arc will be assigned to each location as an endpoint in our arc creation algorithm. For the entirety of the paper we construct the `sink_prob` array from the `arc_weight` array as `sink_prob` = (1 − `arc_weight`)/sum(1 − `arc_weight`), which makes the probability of a location being assigned as an endpoint for any given arc inversely related to the amount of arcs emanating from that location. We now present an algorithm that randomly generates values for $\lambda_{ij\ell}\ \forall(i,j),\ \forall\ell$ that sum to 1 as required by Section 5.2.

Algorithm 1: Generate arc and lead time arrival distribution.

Data: `arc_weight`, `sink_prob`, *total_arcs*
Init: $\boldsymbol{\lambda} = \text{zeros}(n^2, n^2, L)$
for $i \in 0, 1, ..., n^2 - 1$ **do**
  $num\_arcs = \text{round}(arc\_weights_i \cdot total\_arcs))$
  $\texttt{arrival\_weights} = \text{dirichlet}(\text{ones}(num\_arcs)) \cdot arc\_weights_i$
  $\texttt{sink\_list} = \text{choice}(\text{range}(n^2), num\_arcs, \texttt{sink\_prob})$
  $\boldsymbol{\lambda} = \text{update}(\texttt{arrival\_weights}, \texttt{sink\_list})$
return $\boldsymbol{\lambda}$

Algorithm 1 loops through each location, and generates a vector `arrival_weights` with length equal to the number of arcs that emanate from that location. The arrival rate vector `arrival_weights` is created with a Dirichlet distribution with a parameter vector

of **1**, to randomly distribute the arrival rate amongst the arcs in a non-uniform fashion. Then, `sink_list` is generated with equal length, assigning the endpoints of these arcs according to `sink_prob`. Finally, $\boldsymbol{\lambda}$ is updated with these arrival weights with he `update` function. Specifically, `update` assigns the weights to the proper arcs, and uniformly distributes the weights amongst the positive lead times (no delivery tasks can arrive and already be expired or late). As such, we are returned λ which describes the arrival rates over all demand parameters, and sums to 1.

The exact output relating to Fig. 2(a) is lengthy, and thus we will simply list the 27 non-zero arcs present (the locations are labeled from 0 to 8, where the bottom left of the 3×3 grid is 0 and the top right is 9). The arcs are: {(0, 7), (0, 5), (0, 0), (0, 3), (0, 2), (0, 8), (1, 0), (2, 3), (2, 8), (3, 7), (4, 1), (4, 8), (4, 4), (4, 0), (4, 3), (4, 2), (4, 7), (4, 6), (5, 5), (5, 4), (5, 6), (5, 7), (6, 1), (6, 6), (7, 3), (7, 7), (8, 4)}.

In Section 5.2.4 we create new demand patterns and ad-hoc arrival patterns for our experiments. For demand patterns, we first need to generate a new `arc_weight` vector and then run Algorithm 1. We do this with Algorithm 2.

Algorithm 2: Generate random array.

Init: $k = n^2$, $\mathbf{w} = \mathbf{0}$
while $k > 0$ **do**
  $b = \text{uniform\_choice}([1, 2, ...k])$
  $k = k - b$
  $i = \text{uniform\_choice}([0, 1, ..., n^2 - 1])$
  $w_i = w_i + (b/n^2)$
return w

Algorithm 2 simply creates a random array of length n^2 that sums to 1 and is non-uniform. The purpose of this algorithm is that it can generate arrays that have varying distributions, which help to represent areas of high-density demand outflow. Additionally, we use 2 to construct ad-hoc arrival patterns that sum to 1, as they are represented as an array of length n^2 as well.

Appendix B. Proofs

Proof of Theorem 4.1. This proof will show that a feasible solution to formulation in Eq (4) $(\mathbf{x}, \mathbf{u}, \mathbf{u_0})$ corresponds to a feasible solution to Eq (6) with the same objective value, and that an optimal solution to formulation (6) $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{u_0}^*, \mathbf{t}^*, \mathbf{w}^*)$ corresponds to a feasible solution to formulation (4) with the same objective value. By showing these two conditions we will have completed the proof. The first direction is the trivial direction, as is true by the construction of formulation in Eq (6). Simply take the same values for $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{u_0}$ and set $t_{kij\ell} = -u_{kij\ell} \ln(u_{kij\ell}/x_{ij\ell})$ if $x_{ij\ell}, u_{kij\ell} > 0$ and 0 if $x_{ij\ell}, u_{kij\ell} > 0$ (note that we only consider these two cases because the fact that $v_{kij\ell} > 0$ implies that $u_{kij\ell} = 0$ if and only if $x_{ij\ell} = 0$), $w_{k1} = u_{k0} \ln(u_{k0})$, and $w_{k2} = -ln(u_{k0})$. Then, $(\mathbf{x}, \mathbf{u}, \mathbf{u_0}, \mathbf{t}, \mathbf{w})$ is a feasible solution to formulation (6).

As for the other direction, let $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{u_0}^*, \mathbf{t}^*, \mathbf{w}^*)$ be an optimal solution to formulation (6). Due to the objective and definitions of the cones, we have $w^*_{k1} = u^*_{k0} \ln(u^*_{k0})$ and $w^*_{k2} = -ln(u^*_{k0})$. The difficult part comes from $\mathbf{t}$, which we will deal with in cases like the previous direction. If $x^*_{ij\ell}, u^*_{kij\ell} > 0$ then $t^*_{kij\ell} = -u^*_{kij\ell} \ln(u_{kij\ell}/x_{ij\ell})$ and if $x^*_{ij\ell}, u^*_{kij\ell} > 0$ then $t^*_{kij\ell} = 0$. In both of these cases, $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{u_0}^*)$ is an optimal solution to formulation (4) with the same objective. We cannot use the same definition of $v_{kij\ell}$ to rule out the cases $x^*_{ij\ell} > 0, u^*_{kij\ell} = 0$ and $x^*_{ij\ell} = 0, u^*_{kij\ell} > 0$ like we did previously. However, constraint (6d) guarantees that $x^*_{ij\ell} = 0, u^*_{kij\ell} > 0$ is not feasible. Therefore, all that is left to show is that there cannot exist an optimal solution for formulation (6) where $x^*_{ij\ell} > 0, u^*_{kij\ell} = 0$ and the proof is complete. For any $x > 0$, let $f_x : (0, \infty) \to \mathbb{R}$ be given by $f_x(u) := u \ln(u/x)$. We note that $f'_x(u) = \ln(u/x) + 1$ approaches $-\infty$ as u approaches 0. Without loss of generality, we let (i, j) and k be the indices for which $\hat{\ell} = \min\{\ell : x_{ij\ell} > 0, u_{kij\ell} = 0\}$ exists. We then have $t_{kij\hat{\ell}} = 0$.

We first consider the case where $\hat{\ell} = 1$. Observe that $(u^*_{k0}, 1, w^*_{k2})$ implies that $u^*_{k0} > 0$ and let $g : (0, \infty) \to \mathbb{R}$ be given by $g(u) := -(1 - u)\ln(u)$ (and note that $g'(u^*_{k0}) = \ln(u^*_{k0}) - (1 - u^*_{k0})/u^*_{k0} > -\infty$). There exists a $\delta \in (0, \min\{u^*_{k0}, x^*_{ij\hat{\ell}}/\mu_k\})$ such that $f'_{x^*_{ij\hat{\ell}}}(u) - d_{kij\hat{\ell}} - \theta_0\mu_k - g'(u^*_{k0} - u) < 0$ for all $u \in (0, \delta)$. Now, consider the new solution $(\mathbf{x}, \mathbf{u}, \mathbf{u_0}, \mathbf{t}, \mathbf{w})$ constructed from $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{u_0}^*, \mathbf{t}^*, \mathbf{w}^*)$ by increasing $u^*_{kij\hat{\ell}}$ by δ and decreasing u^*_{k0} by δ. The changes this has caused to the solution are as follows: $u_{kij1} = \delta$, $t_{kij1} = -\delta \ln(\delta/x^*_{ij1})$, $u_{k0} = u^*_{k0} - \delta > 0$, $x_{ij0} = x^*_{ij0} - \delta\mu_k \geq 0$, $w_{k1} = -u_{k0}\ln(u_{k0})$, and $w_{k2} = \ln(u_{k0})$. The change in objective value is

$$\int_0^\delta \left[f'_{x^*_{ij1}}(u) - d_{kij\hat{\ell}} - \theta_0\mu_k - g'(u^*_{k0} - u) \right] du < 0$$

which implies that $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{u_0}^*, \mathbf{t}^*, \mathbf{w}^*)$ was not an optimal solution to formulation (6).

Finally, we consider the case where $\hat{\ell} \geq 2$. First, note that $x^*_{ij\hat{\ell}-1} > 0$ from constraint (6b) and that $u^*_{kij\hat{\ell}-1} > 0$ by the construction of $\hat{\ell}$. We observe that $f'_{x^*_{ij\hat{\ell}-1}}(u^*_{kij\hat{\ell}-1}) = \ln(u^*_{kij\hat{\ell}-1}/x^*_{ij\hat{\ell}-1}) + 1 > -\infty$, therefore there exists a $\delta \in (0, \min\{u^*_{kij\hat{\ell}-1}, x^*_{ij\hat{\ell}-1}/\mu_k\})$ such that $f'_{x^*_{ij\hat{\ell}}}(u) - d_{kij\hat{\ell}} - f'_{x^*_{ij\hat{\ell}-1}}(u^*_{kij\hat{\ell}-1} - u) + d_{kij\hat{\ell}-1} - \theta_{\hat{\ell}-1}\mu_k < 0$ for all $u \in (0, \delta)$. Now, consider the new solution $(\mathbf{x}, \mathbf{u}, \mathbf{u_0}, \mathbf{t}, \mathbf{w})$ constructed from

$(\mathbf{x}^*, \mathbf{u}^*, \mathbf{u_0}^*, \mathbf{t}^*, \mathbf{w}^*)$ by increasing $u^*_{kij\hat{\ell}}$ by δ and decreasing $u^*_{kij\hat{\ell}-1}$ by δ. The changes this has caused to the solution are as follows: $u_{kij\hat{\ell}} = \delta$, $t_{kij\hat{\ell}} = -\delta \ln(\delta / x^*_{ij\hat{\ell}})$, $x_{ij\hat{\ell}-1} = x^*_{ij\hat{\ell}-1} - \mu_k \delta > 0$, $u_{kij\hat{\ell}-1} = u^*_{kij\hat{\ell}-1} - \delta \in (0, x_{ij\hat{\ell}-1}]$, and $t_{kij\hat{\ell}-1} = -u_{kij\hat{\ell}-1} \ln(u_{kij\hat{\ell}-1} / x^*_{ij\hat{\ell}-1})$. The change in the objective value is

$$\int_0^{\delta} \left[f'_{x^*_{ij\hat{\ell}}}(u) - d_{kij\hat{\ell}} - f'_{x^*_{ij\hat{\ell}-1}}(u^*_{kij\hat{\ell}-1} - u) + d_{kij\hat{\ell}-1} - \theta_{\hat{\ell}-1}\mu_k \right] du < 0$$

which implies that $(\mathbf{x}^*, \mathbf{u}^*, \mathbf{u_0}^*, \mathbf{t}^*, \mathbf{w}^*)$ was not an optimal solution to formulation (6). □

Proof of Theorem 4.2. We will prove this concisely by contradiction. Suppose that there is an optimal solution with $\sum_{j\in V} z^*_{ij} > 0$ and $\sum_{j\in V} z^*_{ji} > 0$ for location i. Let $\delta_i = \min(\sum_{j\in V} z^*_{ij}, \sum_{j\in V} z^*_{ji})$, $\sum_{j\in V} z_{ij} = \sum_{j\in V} z^*_{ij} - \delta_i$, and $\sum_{j\in V} z_{ji} = \sum_{j\in V} z^*_{ji} - \delta_i$. It is clear to see that either $\sum_{j\in V} z_{ij}$ or $\sum_{j\in V} z_{ji}$ must equal zero (or both), and constraint (8b) is not violated, as we have decreased the left and right-hand sides by the same quantity. There exists one i such that $\delta_i > 0$, but there may be multiple. Regardless, we see that the objective has strictly improved:

$$\begin{aligned} \sum_{i\in V}\sum_{j\in V} z_{ij} c_{ij} &= \sum_{i\in V}\sum_{j\in V} (z^*_{ij} - \delta_i) c_{ij} \\ &< \sum_{i\in V}\sum_{j\in V} z^*_{ij} c_{ij} \end{aligned}$$

Therefore, we have constructed a solution better than our supposed optimal solution, which is the desired contradiction. □

References

Alnaggar, A., Gzara, F., Bookbinder, J.H., 2021. Crowdsourced delivery: A review of platforms and academic literature. Omega 98, 102–139.

Amazon, 2023. https://flex.amazon.com/. Accessed: September 19th, 2023.

Archetti, C., Guerriero, F., Macrina, G., 2021. The online vehicle routing problem with occasional drivers. Comput. Oper. Res. 127, 105144.

Arslan, A.M., Agatz, N., Kroon, L., Zuidwijk, R., 2019. Crowdsourced delivery—A dynamic pickup and delivery problem with ad hoc drivers. Transp. Sci. 53 (1), 222–235.

Banerjee, D., Erera, A.L., Toriello, A., 2022a. Fleet sizing and service region partitioning for same-day delivery systems. Transp. Sci. 56 (5), 1327–1347.

Banerjee, S., Freund, D., Lykouris, T., 2022b. Pricing and optimization in shared vehicle systems: An approximation framework. Oper. Res. 70 (3), 1783–1805.

Barbosa, M., Pedroso, J.P., Viana, A., 2023. A data-driven compensation scheme for last-mile delivery with crowdsourcing. Comput. Oper. Res. 150, 106059.

Behrendt, A., Savelsbergh, M., Wang, H., 2022. A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. Transp. Sci..

Ben-Akiva, M.E., Lerman, S.R., 1985. Discrete Choice Analysis: Theory and Application to Travel Demand, vol. 9, MIT Press.

Bimpikis, K., Candogan, O., Saban, D., 2019. Spatial pricing in ride-sharing networks. Oper. Res. 67 (3), 744–769.

Boyd, S., Boyd, S.P., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Cachon, G.P., Daniels, K.M., Lobel, R., 2017. The role of surge pricing on a service platform with self-scheduling capacity. Manuf. Serv. Oper. Manage. 19 (3), 368–384.

Cao, Y., Kleywegt, A., Wang, H., 2022. Dynamic pricing for two-sided marketplaces with offer expiration. Pre-print on SSRN.

Cao, J., Olvera-Cravioto, M., Shen, Z.-J., 2020. Last-mile shared delivery: A discrete sequential packing approach. Math. Oper. Res. 45 (4), 1466–1497.

Castillo, V.E., Bell, J.E., Mollenkopf, D.A., Stank, T.P., 2022. Hybrid last mile delivery fleets with crowdsourcing: A systems view of managing the cost-service trade-off. J. Bus. Logist. 43 (1), 36–61.

Chen, C., 2023. How to get your amazon purchase delivered in just hours with same-day delivery. https://www.aboutamazon.com/news/amazon-prime/amazon-same-day-delivery. Accessed: September 19th, 2023.

Cosgrove, E., 2022. Gig economy emerges as key part of the parcel ecosystem. https://www.supplychaindive.com/news/gig-economy-parcel-ecosystem/593203/. Accessed: September 19th, 2023.

Dai, J., Harrison, J.M., 2020. Processing Networks: Fluid Models and Stability. Cambridge University Press.

Financial Anatomy, 2022. Uber eats driver tutorial in 2022 (what's NEW). https://www.financialanatomy.net/blog/ubereatstutorial. Accessed: September 19th, 2023.

Hu, B., Hu, M., Zhu, H., 2022. Surge pricing and two-sided temporal responses in ride hailing. Manuf. Serv. Oper. Manage. 24 (1), 91–109.

Instacart, 2023. How instacart works. https://www.instacart.com/help/section/4524023334676/360039161192. Accessed: September 19th, 2023.

Kleywegt, A.J., Shao, H., 2021. Optimizing pricing, repositioning, en-route time, and idle time in ride-hailing systems. arXiv preprint arXiv:2111.11551.

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. Transp. Res. C 103, 83–103.

Lee, A., Savelsbergh, M., 2015. Dynamic ridesharing: Is there a role for dedicated drivers? Transp. Res. B 81, 483–497.

McFadden, D., 1973. Conditional logit analysis of qualitative choice behavior. In: Zarembka, P. (Ed.), Frontiers in Econometrics. Academic Press, pp. 105–142, (Chapter 4).

MOSEK ApS, 2023. Introducing the MOSEK optimization suite 10.0.38. URL https://docs.mosek.com/10.0/intro/index.html.

Murdocca, P., 2017. Meet roadie - america's new peer-to-peer delivery network. https://www.droverrideshare.com/blog/roadie. Accessed: September 19th, 2023.

Özkan, E., 2020. Joint pricing and matching in ride-sharing systems. European J. Oper. Res. 287 (3), 1149–1160.

Pierbridge, Inc, 2018. The growth of the gig delivery economy. https://pierbridge.com/news/the-growth-of-the-gig-delivery-economy/. Accessed: September 19th, 2023.

Roadie, 2023. Why drive with roadie make money driving your way. https://driver.roadie.com/make-money-driving/. Accessed: September 19th, 2023.

Santini, A., Viana, A., Klimentova, X., Pedroso, J.P., 2022. The probabilistic travelling salesman problem with crowdsourcing. Comput. Oper. Res. 142, 105722.

Savelsbergh, M.W., Ulmer, M.W., 2022. Challenges and opportunities in crowdsourced delivery planning and operations. 4OR-Q J. Oper. Res. 20, 1–21.

Spherical Insights LLP, 2022. Global last mile delivery for E-commerce market size to grow USD 136.12 billion by 2030 | CAGR of 8.13. https://www.globenewswire.com/en/news-release/2022/9/27/2523224/0/en/Global-Last-Mile-Delivery-for-E-Commerce-Market-Size-to-grow-USD-136-12-Billion-by-2030-CAGR-of-8-13-Spherical-Insights-LLP.html. Accessed: September 19th, 2023.

Talluri, K., Van Ryzin, G., 2004. Revenue management under a general discrete choice model of consumer behavior. Manage. Sci. 50 (1), 15–33.

Torres, F., Gendreau, M., Rei, W., 2022. Crowdshipping: An open VRP variant with stochastic destinations. Transp. Res. C 140, 103677.

Train, K.E., 2009. Discrete Choice Methods with Simulation. Cambridge University Press.

Uber, 2023. Deliver when you want, make what you need. https://www.uber.com/us/en/deliver/. Accessed: September 19th, 2023.

Ulmer, M., Savelsbergh, M., 2020. Workforce scheduling in the era of crowdsourced delivery. Transp. Sci. 54 (4), 1113–1133.

Woodruff Sawyer, 2020. AB5 in California: A new era for independent contractors and freelance workers. https://woodruffsawyer.com/employee-benefits/ab5-california/. Accessed: September 19th, 2023.

Yildiz, B., Savelsbergh, M., 2019. Service and capacity planning in crowd-sourced delivery. Transp. Res. C 100, 177–199.