Crowdsourced Same-day Delivery: Joint Planning and Coordination for Centralized and Decentralized Couriers

Adam Behrendt, Martin Savelsbergh, He Wang
School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA,
adam.behrendt@gatech.edu, martin.savelsbergh@isye.gatech.edu, he.wang@isye.gatech.edu

Crowdsourced delivery platforms operate as an intermediary between consumers who place orders and couriers who make 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 orders with time windows. The two critical courier management decisions for a platform are how to schedule couriers and how to assign orders 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 for more freedom to couriers in deciding when and where to work. Crowdsourced delivery platforms have begun to utilize multiple courier types (i.e., a hybrid system) with the hope of reaping the advantages of each. In this paper, we address the challenge of managing two types of couriers at both the planning and operational level. We present fluid models for fleet sizing and order pricing that establish the superiority of a hybrid system over each system individually and we see total cost improvements ranging from 3–20% in a realistic example. Furthermore, we study order allocation in a hybrid system in depth, and propose order pooling and splitting policies. In our online experiments we find that a look ahead splitting policy outperforms pooling and batching policies by 2.4% while being more robust to uncertainty.

Key words: crowdsourced same-day delivery, marketplace design, mobile delivery platforms

1. Introduction

Last-mile delivery has seen sustained and rapid growth in the last decade; consumer participation in, and reliance on, e-commerce has been the dominant driver. At the same time, the emergence of the gig economy has given rise to crowdsourced delivery platforms using flexible, non-employee couriers (crowdsourced) to provide last-mile delivery capacity. A crowdsourced delivery platform operates as an intermediary between consumers who place orders and couriers who make deliveries. Delivery tasks arrive dynamically to the platform and often have to be performed shortly after they arrive. The platform seeks to achieve a predetermined service level (e.g., 95% on-time delivery, guaranteed delivery by end-of-day, etc.) using the 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. 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 thereof, it has over the couriers’ choices of when to work and which orders to serve. Two main types of scheduling are commonly used in practice: (1) platform-scheduling, where the platform offers a set of shifts for couriers to sign up 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, and serve orders until they decide to depart. 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. 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 (e.g., Assembly Bill 5 in California).

![Figure 1](image_url) Different order assignment schemes employed by crowdsourced delivery platforms in practice.

In addition to courier scheduling, we observe two main types of order assignment schemes used by delivery platforms: (1) platform-assigned, where the platform “pushes” orders to couriers, and (2) self-assigned, where couriers “pull” orders from the platform; see Figure 1. With order platform-assignment, the platform pushes orders to couriers, who then accept or reject the orders (rejecting too many orders may result in a penalty). Order platform-assignment reduces uncertainty about which orders are served and when, making it easier to meet a desired service level. With order
self-assignment, the platform posts orders on a virtual bulletin board and a courier selects their preferred order from among the posted orders. There may be varying degrees of self-assignment, as the platform may curate a different menu of orders for different couriers. Order self-assignment offers couriers more flexibility in which orders they serve, which may help avoid repercussions in case of too many rejected pushed orders. Furthermore, when couriers can select the order that perfectly matches their preferences (which is unknown to the platform), they may be willing to accept a smaller payout on average per order. Recently, a few crowdsourced delivery platforms have started to employ both centralized and decentralized scheduling and order assignment simultaneously, so as to balance courier control and courier satisfaction, as some couriers prefer the flexibility of choosing when they work and which orders they serve. Hereafter, we will refer to couriers that are platform-scheduled and platform-assigned orders as committed couriers and couriers that self-schedule and self-assign orders as ad-hoc couriers. Committed couriers are typically compensated on a per-hour basis, while ad-hoc couriers are typically compensated on a per-order basis.

We refer to a platform that relies on both committed and ad-hoc 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 committed and ad-hoc components as subsystems or delivery channels (interchangeably). While utilizing a hybrid delivery system retains the advantages of both delivery channels, it also introduces a new challenge in the planning and operation of the platform, namely, how to simultaneously manage two delivery channels. In this paper, we address this challenge at both the planning and operational level.

In terms of planning, we study how to decide the relative capacities of the delivery channels, which are specified by the fleet sizing for committed couriers and order pricing for ad-hoc couriers, and investigate when it is beneficial to use a hybrid delivery system. More specifically, we present fluid models for the planning of hybrid and pure delivery systems. We show, using an illustrative example, that using a hybrid delivery system rather than a pure delivery systems can result in savings of 3–20%.

In terms of operations, we study how to coordinate the delivery channels in real time given their relative capacities. Specifically, we investigate how to allocate orders to each of the delivery channels. We present two general classes of coordination policies, depicted in Figure 2. The split class of policies refers to policies that construct two mutually exclusive sets of orders, one for each delivery channel, in which each arriving order is allocated to one of the channels based on a set of rules. A split policy may be static (e.g., with a fixed set of rules) or dynamic (e.g. with a set of rules that changes based on the state of the system). In a split policy, the two channels utilize their delivery capacity independently of one another. That is, orders on the bulletin board for self-assignment are not serviceable by committed couriers and orders assigned to committed couriers
are not visible on the bulletin board. The *pooled* class of policies, on the other hand, maintains a single set of orders and utilizes the capacity of the two delivery channels for their fulfillment. As such, all orders are visible on the bulletin board, but can also be assigned to committed couriers. The delivery channel that fulfills an order depends on the channel that processes the order first. That is, pooled policies dynamically allocate orders to the two delivery channels based on the rates at which orders can be served by each of these channels, whereas split policies attempt to make some intelligent split upfront - creating a set orders for the committed couriers that is free of influence from the ad-hoc couriers, and vice versa. Each policy comes with its own advantages and disadvantages. We find that either a split or a pooled policy can outperform the other policy depending on the setting. Specifically, split policies perform better in situations with unexpected low traffic intensity, i.e., a larger than expected ratio of the ad-hoc arrival rate and the order arrival rate, while pooled policies perform better in situations with unexpected high traffic intensity. Due to the split policies’ ability to leverage future information about committed couriers, we find that an intelligently designed split policy tends to outperform a pooled policy.

The remainder of this paper is organized as follows. Section 2 details the relevant literature related to hybrid delivery systems. Section 3 formally defines the planning and operational problems. Section 4 presents fluid models for planning the pure delivery systems. Section 5 expands upon these models to provide a fluid model for the planning of a hybrid delivery system, and leverages this to argue the optimality of a hybrid delivery system. Section 6 defines the coordination policies, presents results illustrating the intuitive trade-offs between the policies, and details the online implementation of such policies. Section 7 provides the details of our simulation study and presents results that validate our hypotheses regarding the performance of the coordination policies. Finally, Section 8 presents the implications of our findings in the form of managerial recommendations and discusses future research directions.
2. Literature Review

This work is at the intersection of many research areas with vast collections of literature (crowdsourced delivery, pricing, vehicle routing, and marketplace design, to name a few). Therefore, we focus specifically on reviewing papers that study crowdsourced delivery platforms that utilize multiple types of couriers. For an overviews of crowdsourced delivery, we direct the reader to Alnaggar et al. (2021) and Savelsbergh and Ulmer (2022). In our work, we distinguish couriers based on how they are scheduled and how orders are assigned to them: centralized versus decentralized. Specifically, we focus on platform-scheduled and platform-assigned couriers, as well as self-scheduling and self-assigning couriers. In the literature, instead, couriers are usually only distinguished based on how they are scheduled; most papers assume orders are assigned by the platform (with only a few exceptions). Thus, while much of the existing research is exclusively concerned with “when” couriers work, we are also concerned with “what” couriers work on, i.e., how orders are allocated to and prioritized based on courier types.

Archetti et al. (2016) proposed a vehicle routing problem with occasional drivers, in which a platform makes deliveries using employee couriers who are assigned a delivery route, and occasional couriers who are assigned a single order with a delivery location that is within an acceptable radius of their destination. While this work is one of the first to introduce a system that use employee and crowdsourced couriers (motivated by a real-world setting at Walmart), it only considers a clairvoyant planner with perfect information (the arrival of crowdsourced couriers is known in advance) and does not consider any form of decentralized order assignment. Arslan et al. (2019) study a dynamic pickup and delivery problem with employee and self-scheduling couriers (which they refer to as ad-hoc couriers, but which are not ad-hoc couriers according to our definition because the platform assigns their orders). They propose a rolling horizon solution method in which a matching problem for order assignment is solved whenever new information becomes available (new orders arrive or self-scheduling couriers arrive). The results of their simulation study suggest that incorporating self-scheduling couriers in addition to employee couriers can decrease the system-wide vehicle miles due to the flexibility in their availability. Again, this paper is concerned with order assignment by the platform, focusing on the uncertainty associated with the availability of self-scheduling couriers. Ulmer and Savelsbergh (2020) also consider a system which relies on couriers that are either centrally scheduled or are self-scheduling. Specifically, they seek to construct shifts for the centrally scheduled couriers given the uncertainty related to the availability of self-scheduling couriers by using a value function approximation approach. Yildiz and Savelsbergh (2019) address many questions surrounding the planning and operations of a meal delivery platform. One of the questions they consider is if employee couriers should be used in conjunction with self-scheduling couriers. Their numerical experiments suggest that a system that relies on both types of couriers
can improve the reliability and profit of the platform. Castillo et al. (2022) uses real world data from a pharmaceutical retailer in New York to investigate the performance of a system utilizing self-scheduling and employee couriers. Their simulation results support the intuition that using per-order compensation for self-scheduling couriers is essential. However, they also find that pricing delivery tasks too low reduces the probability of self-scheduling couriers accepting orders pushed to them and that pricing too high results an inefficient use of couriers (both platform-scheduled and self-scheduled).

Cao et al. (2020) considers a hybrid delivery system (as we have defined it) for last-mile delivery composed of committed employee couriers and ad-hoc couriers and models their operations as a discrete sequential packing problem. The main assumption that limits this work’s usefulness in practice is that the employee couriers are only active at the end of the day (after the ad-hoc couriers finish their operations) to handle demand unserved by ad-hoc couriers. Furthermore, it is assumed that ad-hoc couriers know which orders they want to serve upon arrival to the platform, and their decisions are unaffected by the platforms pricing decision. Santini et al. (2022) consider a similar setting, the so-called Probabilistic Traveling Salesman Problem with Crowdsourcing. A subset of orders is assigned to crowdsourced couriers and are served with a certain probability. The remaining orders and those left unserved by the crowdsourced couriers are served by an employee courier. The goal is to simultaneously determine the (uncertain) route of the employee courier and the set of orders to assign to crowdsourced couriers. The definition is sufficiently general to allow self-assigning ad-hoc couriers. However, the authors do not provide any operational details regarding the crowdsourced couriers. The work of Behrendt et al. (2022) considers a hybrid delivery system that utilizes both committed and ad-hoc couriers (as we have defined them in the previous section), where the goal is to construct shifts for the committed couriers given the uncertainty that results from the self-scheduling and self-assigning of the ad-hoc couriers. This is accomplished with a variant of sample average approximation and simulation optimization, with a prescriptive machine learning method proposed for online speed ups. While this work does consider self-assigning, it does so using a simple random choice model. It does not consider an explicit pricing model for the orders to be served and selected by ad-hoc couriers nor does it explore the real-time coordination of the two types of couriers.

3. Problem Description
We are concerned with the planning and operations of a crowdsourced delivery platform, whose goal is to serve customer demand at minimum cost. Let $\mathcal{T} = [0, T]$ be the operating period (e.g. a day) in which orders arrive according to a Poisson process with rate $\lambda$. Each order has the following characteristics: order placement time, pickup location (origin), delivery location (destination), and
the delivery promise time (deadline). Let the distance \( (d) \) of an order be the travel distance from the pickup location to the delivery location and the relative lead time \( (\ell) \) of an order be the time remaining until the latest possible pickup time that ensures on-time delivery. Specifically, the relative lead time for an order is:

\[
\ell = \max(0, \text{deadline} - (t + d \cdot V)),
\]

where \( t \) is the current time and \( V \) is a constant travel speed. The relative lead time of an order is essentially the lead time when accounting for the travel time required to serve that order. When the relative lead time drops below 0, the order is guaranteed to be delivered late and is considered expired. When an order expires it leaves the system and the crowdsourced delivery platform incurs penalty cost \( \theta \).

In order to serve the dynamically arriving orders, a crowdsourced delivery platform employs two types of couriers: committed and ad-hoc couriers. We assume the platform hires a fleet of \( M \) committed couriers for the entire operating period \( T \). The platform assigns (pushes) orders to the committed couriers and pays them a fixed wage per unit time, \( w \). We further assume that committed couriers do not reject orders assigned to them. This assumptions simplifies the modeling, but the models can easily be extended to accommodate rejecting orders with some probability.

The arrivals of ad-hoc couriers are stochastic and assumed to occur according to a Poisson process with rate \( \mu \). That is, the arrival of ad-hoc couriers is an exogenous process. This assumption ensures tractability of the model proposed in Sections 4 and 5 and is reasonable in many settings. (It is not unreasonable to assume that a platform has a fairly accurate forecast for the ad-hoc arrival processes during an upcoming operating period. The ad-hoc arrival processes are mostly impacted by past interactions between ad-hoc couriers and the platform; not so much by price fluctuations.) Ad-hoc couriers arrive at a random location and examine the orders that are posted on the bulletin board (in their mobile app). This bulletin board displays order characteristics and courier payout, i.e., the compensation received by an ad-hoc courier if the courier decides to serve the order. When an ad-hoc courier arrives in the system, the courier either selects an order to serve and receives the payout, or leaves the system. The random choice behavior of a courier is dictated by a multinomial logit (MNL) choice model described in detail in Section 4.2. The pricing model we consider sets the payout of an order based the order’s relative lead time and origin-destination distance, i.e., \( p_{\ell d} \forall \ell = 1, ..., B; d = 1, ..., D \).

In addition to determining the fleet size of committed couriers and the ad-hoc order pricing policy, the platform needs to coordinate the delivery channels during actual operations. A coordination policy \( \pi \) is a policy that dictates how orders are allocated to each of the delivery channels (i.e.,
which channel an order is allocated to upon its arrival, or the postponement of this decision) and, for the orders allocated to the committed delivery channel, decides the assignment of orders to individual committed couriers.

4. Pure Delivery Systems

For modeling purposes, we discretize both time and origin-destination (O-D) distances. As a consequence, the state of the open orders in the system at any given time can be represented in matrix form, where a row represents orders' relative lead time and a column represent orders' O-D distance. Let $B$ be the maximum possible relative lead time and let $D$ be the maximum possible distance between origin and destination. An entry $X_{\ell d}$ of the matrix, for $\ell = 1, \ldots, B$ and $d = 1, \ldots, D$, represents how many open orders have relative lead time $\ell$ and distance $d$. Let the rate at which new orders arrive in cell $(\ell, d)$ be $\lambda_{\ell d}$. As time passes, the relative lead time (remaining time before an order expires) decreases, and for each unit of time that passes, the matrix “shifts” one column to the left. Let $y_{\ell d}$ be the rate at which orders transition to the adjacent cell to the left. We have that $\sum_{d=1}^{D} y_{\ell d}$ is the rate at which orders expire, each of which incurs a penalty $\theta$. A visual representation of these concepts can be found in Figure 3.

![Figure 3](image.png)

The decision of the platform is to set the rate at which orders are booked by couriers, in which case the order leaves the matrix, as this impacts the rate at which orders transition to the adjacent cell to the left (and, thus, ultimately the penalty for expired orders). The rate at which orders are booked by couriers is determined by the incentives provided to couriers (committed and ad-hoc).
While we can model the pure delivery systems as Markov Decision Processes (MDPs), determining an optimal policy is intractable given the size of the state space. Therefore, we consider a fluid approximation where the state of the system does not vary as orders arrive and are served, allowing us to model the pure systems in steady state as single-stage optimization problems.

4.1. Committed Couriers
As we assume that committed couriers are available for the entire operating period, the platform has to decide the number $M$ of committed couriers to use, where each courier used earns a fixed rate $w$ per unit time. We assume that there is sufficient time between the planning and operational period, so the platform is able to attract the desired number of couriers either by posting shifts for sign-up by crowdsourced couriers and/or by hiring employee couriers. We further assume that committed couriers accept all orders assigned to them. Consequently, to model committed couriers in steady state, we introduce decision variables $r_{\ell d}$ for all $\ell, d$ representing the rate at which committed couriers select orders with relative lead time $\ell$ and O-D distance $d$.

We model the activities of committed couriers in steady state similarly to what has been done by Castillo et al. (2017) and Kleywegt and Shao (2021) in the ride-hailing context by splitting the states of the $M$ couriers into the fraction that is idle, that is making deliveries, and that is en-route to a pickup. We assume that we have a homogeneous region that is sufficiently large to ignore boundary effects (an approximation). Furthermore, we assume that idle couriers are distributed in euclidean space according to a spacial Poisson process with a constant rate. Each time a new order arrives, the idle courier that is the closest to the origin of the order is assigned to it. Using the $L_2$ distance function and the CDF of there being an idle courier within a given distance of an order, the average distance from an order origin to the closest idle courier is $\bar{d} \approx \sqrt{A/4I}$ where $A$ is the area of the region, $I$ is the number of idle couriers, and $I/A$ the density of idle couriers (idle couriers per unit area) (see Chapter 5 of Larson and Odoni (1981) for the derivation of this result). Let $T$ be the average en-route time, i.e., $T = \bar{d}/V$ where $\bar{d}$ is the average distance to a pickup location and $V$ the travel speed. The average distance to a pickup location depends on the density of idle couriers, so a given $T$ implies an $I$ and a given $I$ implies $T$. We consider $T$ and $I$ to be input parameters: $T$ represents a target en-route time and $I$ is the associated number of idle vehicles necessary to ensure that. The activities of the committed couriers in steady state can now be modeled with the Little’s Law constraint

$$\psi \cdot I + \frac{1}{V} \sum_{d=1}^{D} d \sum_{l=1}^{B} r_{\ell d} + T \sum_{d=1}^{D} \sum_{l=1}^{B} r_{\ell d} \leq M,$$

where $\psi \in \{0, 1\}$ and $\psi = 0$ will imply $M = 0$, i.e., that the committed channel is unused. This is necessary to handle situations in which the fleet size is insufficient to achieve the target en-route time, i.e., $M < I$. In our fluid formulation this can be enforced by a constraint $M \leq W \cdot \psi$ for sufficiently large $W$. 
4.1.1. Fluid Model for the Committed System. We have the following fluid formulation for the pure committed courier system:

\[
\begin{align*}
\min_{\{y,r,M\}} & \quad w \cdot M + \theta \sum_{d=1}^{D} y_{ld} \\
\text{s.t.} & \quad y_{\ell+1,d} - y_{\ell,d} = r_{\ell,d} - \lambda_{\ell,d} & \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D \\
& \quad \psi \cdot I + \frac{1}{V} \sum_{d=1}^{D} d \sum_{\ell=1}^{B} r_{\ell,d} + T \sum_{d=1}^{D} \sum_{\ell=1}^{B} r_{\ell,d} \leq M \\
& \quad M - W \cdot \psi \leq 0 \\
& \quad y_{\ell,d} \geq 0, \ r_{\ell,d} \geq 0 & \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D \\
& \quad \psi \in \{0,1\} \\
& \quad M \in \mathbb{Z}_{\geq 0}.
\end{align*}
\]

The objective (3a) seeks to minimize the total cost which is comprised of the wage paid to committed couriers and the penalty incurred for expired orders. Constraint (3b) is a flow balance constraint, which constrains the rates at which committed couriers can be assigned to orders of a given relative lead time and distance based on the orders’ presence in the system. Constraint (3c) is the Little’s Law constraint discussed and (3d) allows the committed channel not to be used. Constraints (3e), and (3g) are for non-negativity and integrality. Now, consider the LP relaxation of Formulation (3), and let the objective be \(c\).

**Theorem 1.** The objective value of the LP relaxation of (3) as a function of \(M\) is piecewise linear convex, with exactly \(D + 1\) breakpoints.

Theorem 1 follows from the fact that the cost of a solution is completely characterized by \(M\) (and the penalty for expiring orders outweighs the cost for increasing the fleet size). The first breakpoint is \(M = I\) when no orders are served (\(r_{\ell,d} = 0\) for \(\ell = 1, \ldots, L\) and \(d = 1, \ldots, D\)). The final breakpoint occurs when all orders are served (\(\sum_{d=1}^{D} y_{1d} = 0\)). The other \(D - 1\) breakpoints are a consequence of Constraint (3c). Specifically, for a given value \(M\), the optimal solution to the LP relaxation first maximizes \(\sum_{d=1}^{D} y_{1d}\), then maximizes \(\sum_{\ell=1}^{B} r_{\ell1}\), and so on, until the constraint is tight.

Theorem 1 reflects that it is more efficient to have available committed couriers serve orders with small origin-destination distance. Because the expiration cost is the same for all orders, serving orders with small origin-destination distance allows committed couriers to serve more orders on average thereby avoiding expiration penalties.

We provide a formal proof for Theorem 1 in the Appendix, and we discuss in Section 5.2 that this is one of the main justifications of a hybrid delivery system.
4.2. Ad-hoc Couriers

Ad-hoc couriers choose orders based on the relative lead time and the O-D distance of an order. When an ad-hoc courier arrives in the system, the courier either selects an order to serve and receives the associated payout, or leaves the system. We model a courier’s choice behavior by means of the following multinomial logit (MNL) choice model, which is similar to that studied by Cao et al. (2022). Given a price (payout) \( p_{\ell d} \), an order with relative lead time \( \ell \) and distance \( d \) has preference weight \( v_{\ell d}(p_{\ell d}) := \exp (\beta p_{\ell d} + \alpha_{\ell d}) \), where \( \beta \) is the price sensitivity parameter and \( \alpha_{\ell d} \) are non-monetary preference parameters. Intuitively, there may be a preference of couriers to serve orders with small distance and high relative lead time, because of the “ease” of the order (quick service, with low risk of being late). This concept is captured by Assumption 1.

**Assumption 1.** The non-monetary preference parameters \( \alpha_{\ell d} \) are non-decreasing in the relative lead time \( \ell \) and non-increasing in the distance \( d \).

The probability that an ad-hoc courier selects an order with relative lead time \( \ell \) and distance \( d \) is

\[
u_{\ell d} = \frac{v_{\ell d}(p_{\ell d})}{\sum_{\ell' = 1}^{B} \sum_{d' = 1}^{D} v_{\ell' d'}(p_{\ell' d'}) + 1},
\]

and the probability that an ad-hoc courier does not select any order is

\[
u_0 = \frac{1}{\sum_{\ell' = 1}^{B} \sum_{d' = 1}^{D} v_{\ell' d'}(p_{\ell' d'}) + 1},
\]

where

\[
\sum_{\ell = 1}^{B} \sum_{d = 1}^{D} u_{\ell d} + u_0 = 1.
\]

The MNL choice model is well studied (see Train 2009) and suitable for this setting as it (1) allows us to model courier “taste variation” between orders with observable characteristics (i.e. distance and relative lead times) and (2) permits itself well to the bulletin board style of order selections, as the independence of irrelevant alternatives property allows for the use of proportional substitution of preference weights when all alternatives may not be available (that is, when there may not be orders present with every possible distance and relative lead time combination). While this specific choice model is static from a modeling perspective (all prices are predetermined for a given distance \( d \) and relative lead time \( \ell \)), it is able to capture the dynamics of our setting as the relative lead time of a given order decreases by one unit for each time period it is not selected, allowing each order to have a dynamic price. While the price of an order is dynamic, the solution matrix \( \mathbf{P} \) represents a static pricing policy.
4.2.1. Fluid Model for the Ad-hoc System. The decision variables in the fluid pricing problem are the rate that orders in cell \((\ell,d)\) are served by ad-hoc couriers, \(q_{\ell d}\). As the ad-hoc couriers can choose which order they fulfill, the rate at which the orders in each cell \(q_{\ell d}\) are served is controlled by the price \(p_{\ell d}\) according to the MNL model previously defined. Here \(p_{\ell d} = (\ln(q_{\ell d}/q_0) - \alpha_{\ell d})/\beta\), which is a consequence of the fact that \(q_{\ell d}/q_0 = u_{\ell d}/u_0 = \exp(\beta p_{\ell d} + \alpha_{\ell d})\). We perform change of variable \(q_{\ell d} = u_{\ell d} \cdot \mu\) and \(q_0 = u_0 \cdot \mu\). The objective is to minimize the sum of the total payout to ad-hoc couriers and the penalty cost of expired orders.

\[
\min_{\{y, q\}} \sum_{\ell=1}^B \sum_{d=1}^D q_{\ell d} \cdot p_{\ell d} + \theta \sum_{d=1}^D y_{\ell d} \tag{4a}
\]

s.t. \(y_{\ell+1,d} - y_{\ell d} = q_{\ell d} - \lambda_{\ell d} \quad \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D\) \hspace{1cm} (4b)

\[
\sum_{d=1}^D \sum_{\ell=1}^B q_{\ell d} + q_0 = \mu \quad \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D \tag{4c}
\]

\(y_{\ell d} \geq 0, \ q_{\ell d} \geq 0 \quad \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D\) \hspace{1cm} (4d)

Constraints (4b) describe balance equations for transitions between cells, where we let \(y_{B+1,d} = 0 \quad \forall d = 1, \ldots, D\). Constraint (4c) ensures that the rate of ad-hoc couriers selecting orders (or not selecting any) is equal to the rate at which they arrive in the system. Finally, Constraints (4d) force non-negativity.

**Theorem 2.** The optimal solution \(q^*\) to the ad-hoc subproblem satisfies

\[
q_{\ell d}^* \geq q_{\ell+1,d}^*, \forall \ell = 1, \ldots, B - 1, \ d = 1, \ldots, D. \tag{5}
\]

Theorem (2) states that any optimal solution to Formulation (4) has a non-decreasing sequence of selection rates for orders with equivalent distances and increasing relative lead times. That is, for constant distance, orders with lower relative lead times should prioritize by ad-hoc couriers. As the pricing component is not the focus of the paper, the proof is shown in detail in the appendix along with a required lemma.

**Corollary 1.** The optimal price satisfies

\[
p_{\ell d}^* \geq p_{\ell+1,d}^*, \forall \ell = 1, \ldots, B - 1, \ d = 1, \ldots, D. \tag{6}
\]

Corollary 1 implies that for constant distance, orders with lower relative lead times will have prices that are at least as high as those with larger relative lead times, in order for the ad-hoc couriers to prioritize more urgent orders.

Even though the formulation is nonlinear, it can be solved in polynomial time. The objective function is convex and the problem can be reformulated as optimizing in an exponential cone. For
\[ q_{ld} > 0, \text{ we let } -t_{ld} \geq q_{ld} \ln(q_{ld}/q_0) \text{ which holds if and only if } q_0 \geq q_{ld} \exp(t_{ld}/q_{ld}). \]

The exponential cone is a convex subset of \( \mathbb{R}^3 \) defined as
\[ K_{\exp} = \text{cl}\{ (x_1, x_2, x_3) : x_1 \geq x_2 \exp(x_3/x_2), x_2 > 0 \}. \]

Thus the previous optimization problem is equivalent to the conic program:

\[
\begin{align*}
\text{min} & \quad \left\{ y, q, r, T, I \right\} \\
\text{s.t.} & \quad y_{\ell+1,d} - y_{\ell,d} = q_{\ell,d} - \lambda_{\ell,d} \quad \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D \\
& \quad \sum_{d=1}^{D} \sum_{\ell=1}^{B} q_{\ell,d} + q_0 = \mu \\
& \quad (q_0, q_{\ell,d}, T_{\ell,d}) \in K_{\exp} \quad \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D \\
& \quad y_{\ell,d} \geq 0, \ q_{\ell,d} \geq 0 \quad \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D
\end{align*}
\]

Thus, given \( \Lambda = (\lambda_{\ell,d}) \in \mathbb{R}^{B \times D}, \mu \text{ and } \theta \) for a time period, we can solve formulation (6) to generate an optimal set of prices for the bulletin board.

5. Hybrid Delivery System

We next present a fluid formulation for a hybrid delivery system with both committed and ad-hoc couriers. The platform has to decide the number of committed couriers and the price matrix \( P \).

The formulation accommodates solutions in which only one delivery channel is used. This feature allows us to demonstrate that a hybrid system is better in most realistic settings.

5.1. Fluid Model for the Hybrid System

In a hybrid system the ad-hoc and committed courier channels share their delivery capacity. Therefore, our formulation models the hybrid system with a single matrix from which orders are selected by the channels. We adjust Formulation (4) by adding the fixed wage component to the objective, by adding Constraints (3c), (3d), (3f) and (3g), and by adjusting Constraint 4b to include the rate at which orders are selected by committed couriers, \( r \). The given the following formulation

\[
\begin{align*}
\text{min} & \quad \left\{ y, q, r, T, I \right\} \\
\text{s.t.} & \quad y_{\ell+1,d} - y_{\ell,d} = q_{\ell,d} + r_{\ell,d} - \lambda_{\ell,d} \quad \forall \ell = 1, \ldots, B, \ d = 1, \ldots, D \\
& \quad \sum_{d=1}^{D} \sum_{\ell=1}^{B} q_{\ell,d} + q_0 = \mu \\
& \quad \psi \cdot I + \frac{1}{V} \cdot \sum_{d=1}^{D} \sum_{\ell=1}^{B} r_{\ell,d} + T \sum_{d=1}^{D} \sum_{\ell=1}^{B} r_{\ell,d} \leq M \\
& \quad M - W \cdot \psi \leq 0
\end{align*}
\]
The objective is to minimize the cost of expired orders, the expected payment to the ad-hoc couriers, and the total wage paid to committed couriers. We can still model the ad-hoc system as a conic program and solve the resulting formulation efficiently. We do not present that formulation as Formulation (7) is easier to understand.

5.2. Optimality of a Hybrid System

We now argue that a hybrid delivery system is optimal under the assumption that the ad-hoc price preferences are comparable to the committed courier wage. This ensures that it is not trivially true that the optimal solution to either of the pure formulations is also an optimal solution to the hybrid formulation. We will expand upon comparable incentives in section 5.2.1. A hybrid solution to Formulation (7) is one where $M^* > 0$ and $\sum_{d=1}^{D} \sum_{\ell=1}^{B} q^*_\ell d > 0$. Hybrid solutions can (1) use committed couriers to avoid the asymptotic behavior of the ad-hoc pricing channel as $\sum_{d=1}^{D} \sum_{\ell=1}^{B} q_\ell d \rightarrow \mu$, (2) use ad-hoc couriers to increase the effectiveness of the committed couriers, and (3) mitigate the effect of price increases when relative lead times decrease.

To our first point, we note that objective (7a) is jointly convex in both $q$ and $\sum_{d=1}^{D} \sum_{\ell=1}^{B} q_\ell d$. This is a well established result so we do not provide proof for this statement, however it is easily established with Lemma 2 from Li and Huh (2011) and the preservation of convexity described by Boyd et al. (2004). Furthermore, we recognize that the prices will reveal asymptotic behavior as $\sum_{d=1}^{D} \sum_{\ell=1}^{B} q_\ell d \rightarrow \mu$. Of course, we can imagine a case where $\mu \gg \lambda$ and all demand can be served by the ad-hoc channel without incurring the price increases that occur when the ad-hoc channel reaches its capacity. In reality, such a situation is unlikely as a large mismatch of supply of ad-hoc couriers and demand 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 $\mu > \lambda$ but with both values of similar magnitude, which makes it unlikely that a pure ad-hoc system is able to avoid the asymptotic price behavior. In a hybrid system, using committed couriers will take the burden of the ad-hoc system, so that it does not have to operate near or at its capacity.

To our second point, recall the piecewise linear convexity in $M$ of the objective value of the linear relaxation. Using a similar argument as above, a pure committed system can be improved by including ad-hoc couriers to avoid the marginally increasing cost with $M$. In realistic terms (because this convexity is caused by the variation in O-D distance of orders) this translates to
using the committed couriers time more efficiently to reduce the number of expired orders. This is highlighted in Figure 4a, where for a small instance we report the relative rates that orders from each cell are selected by each type of courier. As we can see, short orders are allocated to the committed system (which heavily depends on $\alpha$ and $\beta$). The behavior shown in Column 4 of the matrix leads us into our third point: hybrid systems can mitigate the price increase associated with decreasing lead time (Corollary 1). The platform can control the rates at which committed couriers select orders from each cell and is indifferent between assigning them orders with different lead times with the same distance (unlike ad-hoc couriers). We highlight this further in Figure 4b where we have increased $\alpha$ to exacerbate the preference to serve less urgent orders. Points two and three hinge on the fact that the sharing of delivery capacity of a hybrid system allows the platform to be more flexible with how it allocates orders, and makes it possible to exploit the advantages of both delivery channels.

5.2.1. An Illustrative Example. The relationship between the committed wage $w$ and the ad-hoc prices (whose scale is determined by $\alpha$ and $\beta$) is the key consideration given realistic ad-hoc courier and demand arrival processes. When the committed wage is small or zero it is always optimal to serve the full demand with committed couriers. For each setting (relative lead times, MNL pricing parameters, etc.), there is a threshold wage under which the a solution to the fluid model results in a constant $M^*$ and $\sum_{d=1}^{D} \sum_{\ell=1}^{B} q_{\ell,d}^* \approx 0$. As $w$ increases, we see increased involvement of ad-hoc couriers. Clearly, there is some upper threshold for $w$ after which $M^* = 0$ and the optimal solution is a pure ad-hoc system (which requires $\mu > \lambda$ for sufficiently high $\theta$). We will illustrate
Figure 5 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.

this relationship and supports our hypothesis that a hybrid system (with comparable wage and ad-hoc prices) is optimal in a realistic setting.

In this setting, we consider orders (e.g. placed at retail stores by online customers) with a delivery location in a circular service region of area $A = 350\text{km}^2$ (e.g. a medium sized city). We consider a service window of 60 minutes. That is, an order must be delivered within 60 minutes of its placement time. As such, our maximum relative leave time $B$ is 60. We assume that the maximum travel time from a pickup location to a delivery location is 20 minutes and that these travel times are uniformly distributed on $(0, 20]$. Vehicles travel at a constant speed $V = 45\text{km/hr}$, and, therefore, we can consider the O-D travel time of order $i$ ($d_i/V$) as opposed to O-D distance for our discretization procedure, consequently $D = 20$. We let the target en-route time be $T = 15$ minutes, which implies that orders will be picked up on average 15 minutes after their placement. We let $\beta = 2$ and, consistent with Assumption 1, $\alpha_d = -30 - .1d + .05\ell \forall d = 1,...D, \ell = 1,...,B$. The expired penalty cost is $\theta = 100$. Finally, we let the arrival rate of orders be $\lambda = 2$ and the arrival rate of ad-hoc couriers be $\mu = 3$. Since $\lambda < \mu$, a pure ad-hoc system is feasible. Figure 5 reports the results for the various fluid models for varying committed courier wage $w$. We do this to understand how the solution structure and the expected improvement over a pure system changes for a hybrid system with different ratios of committed and ad-hoc incentives.

The first thing to note about Figures 5a and 5b is that they confirm our hypothesis that there are lower and upper thresholds of wage compatibility below and above which pure systems are optimal. In this example, a hybrid delivery system is strictly better than either pure system for $2.40 < w < 32.40$. Below and above this range pure committed and ad-hoc systems are optimal, respectively. In Figure 5a, we see that even when the committed courier wage becomes large
Figure 6  Average savings of a hybrid delivery system over the best performing pure delivery system for varying ad-hoc capacity to demand ratios ($\mu/\lambda$).

(>$25/hr) it is still optimal to use a small fleet of committed couriers. We see too that even with a small committed courier wage (resulting a large fleet of committed couriers), employing ad-hoc couriers still provides benefit. A platform can price orders lower than this wage to incur savings in the (low probability) case that an ad-hoc courier selects the order. This may occur when there is an ad-hoc courier for which the order results in very little deviation of their pre-planned route. In Figure 5b we see that the sweet spot for savings (calculated as $100 \cdot \frac{\min(z_C^*, z_A^*) - z_H^*}{z_H^*}$) is for relatively small committed courier wage. In practice, committed couriers typically make $15-$20 dollars per hour (not including tips). In this range of wages, we still see an average savings of $3 - 5\%$.

Due to the asymptotic nature of prices in the ad-hoc system, the savings we can expect to see vary based on the capacity of the ad-hoc channel (roughly speaking $\mu$) in relation to the demand ($\lambda$). In Figure 6, we show the cost savings of a hybrid delivery system for varying ad-hoc capacity to demand ratios ($\mu/\lambda$) for a constant committed courier wage. We set $\lambda = 2$ and $w = $18/hr. For systems with capacity to demand ratios close to 1 (i.e., the ad-hoc channel is operating near its capacity when it has to serve almost all of the demand), we can see average savings of over 20%. The savings quickly decrease when the ratio increases. When the ad-hoc channel capacity is 1.05 times the demand, we still see savings of more than 8%, and even the ad-hoc channel capacity is 2 times the demand, we still see savings of around 3%. The curve reveals that even when an ad-hoc channel has plenty of capacity, employing committed couriers can still produce significant savings.
6. Coordination Policy Design

Given capacities of both delivery channels determined from the last section, we now study how to coordinate order assignment to each of the subsystems. In particular, we consider two classes of policies for order assignment: split and pooled policies, depicted in Figure 2 in the introduction section. Split policies are policies that construct two mutually exclusive sets of orders, one for each delivery channel, where each order is assigned to a channel based on a set of rules. In a split policy, the two channels use their delivery capacity independently of one another. That is, orders on the bulletin board are not serviceable by committed couriers and orders assigned to committed couriers are not visible on the bulletin board. Pooled policies, on the other hand, maintain a single set of orders, and the delivery capacity of the two channels is shared to serve the orders. The delivery channel that serves the order is the one that seeks to process the order first. In a pooled policy, every order that has not yet been selected by an ad-hoc courier or has not yet been assigned to a committed courier is both visible on the bulletin board and considered for assignment to committed couriers. That is, the platform does not restrict the set of orders ad-hoc couriers can see nor the set of orders that can be assigned to committed couriers. On the other hand, split policies restrict these sets. At time $t$, an order is either visible to ad-hoc couriers on the bulletin board or held for assignment to committed couriers. However, the choice is not permanent. An order may be held for assignment to committed couriers at time $t$, but may be transferred to the ad-hoc channel and be visible on the bulletin board at time $t+1$, and vice versa.

6.1. Policy Tradeoffs

We next discuss the main tradeoffs between the two classes of policies. In Section 6.1.1, we present a Markov Decision Process that models a simplified version of our setting, in order to gain intuition on how varying supply to demand ratios affect the optimal policy. After that, in Section 6.1.2, we compare the fluid approximations for both classes of policies to describe the tradeoff between reducing empty miles and average ad-hoc payout. Finally, in Section 6.1.3, we describe the benefit of split policies in terms of planning future decisions for committed couriers.

6.1.1. Traffic Intensity. We start by presenting a simplified version of our setting as a variant of a two sided queue, illustrated in Figure 7. We have an order queue where orders arrive at a rate $\lambda$ and with length $q_t$ at time $t$. The maximum queue length is $N$ and any orders that arrive when the queue is full results in a penalty cost $\theta$ and are rejected. Additionally, we have a committed courier queue with length $k_t$ at time $t$. The total number of committed couriers is $M$ (a parameter) and the expected service rate is $\bar{s}$. Consequently, the maximum queue length is $M$. The arrival rate of committed couriers is then $\bar{s} \cdot k_t$, as the rate at which couriers arrive back in the system is larger when more couriers are serving orders. The action space at time $t$ represents the possible
Figure 7  A simplified setting as a two-sided queue MDP with the presence of ad-hoc couriers.

assignments of orders to committed couriers, \( x_t \in [0, \ldots, \min(q_t, k_t)] \), which incurs a cost of \( x_t \cdot c \). Lastly, we have the ad-hoc system which has action space \( y_t \in \{0, 1\} \) where 0 represents the ad-hoc delivery channel being turned off and 1 represents the ad-hoc delivery channel to be turned on. When the ad-hoc delivery channel is turned on, the order at the front of the order queue is removed at rate \( \mu \) incurring a fixed cost \( p \).

We can simplify the state space by leveraging our assumption that \( c < p \) (in fact, \( c = 0 \) in our case as we assume that the \( M \) committed couriers are pre-paid) which implies that any order and any committed courier would be matched with each other immediately upon arrival. Therefore, either \( q_t = 0 \) or \( k_t = 0 \) for any \( t \). Let \( s_t = q_t - k_t \in [-M, \ldots, N] \) be the new state space and \( A_s = \{0, 1\} \) for states \( s > 0 \) be the action space. (We do not have actions for states \( s \leq 0 \) because we have assumed that orders are immediately matched to waiting committed couriers upon their arrival.) In this setting, a pooled policy is a policy that turns on the ad-hoc delivery channel in every state, i.e., \( a_s = 1 \) for all \( s > 0 \). A split policy, on the other hand, is a policy that turns off the ad-hoc delivery channel for at least one state, i.e., \( \exists s > 0 \) with \( a_s = 0 \).

**Theorem 3.** There exists a monotone optimal stationary policy that has a control limit structure. That is, there exists some threshold state \( R \) such that the optimal actions are

\[
a^*(s) = \begin{cases} 
0 & s < R \\
1 & s \geq R.
\end{cases}
\]

Theorem 3 implies that splitting orders based on the number of orders in the system can be used as a mechanism to control the number of orders entering each delivery channel while prioritizing the committed couriers. This is in line with the intuition that we should maximize the utilization of the pre-paid committed couriers before making orders available to ad-hoc couriers. We establish Theorem 3 by verifying the conditions for the existence of an optimal threshold policy for unichain MDPs presented in Puterman (2014) and provide the proof in the appendix.
Given Theorem 3, we can apply policy iteration algorithms to find the optimal threshold state for small instances. We generated instances of up to $N = 50$ and $M = 5$ and found the optimal threshold state with the discounted policy iteration algorithm. Proposition 1 summarizes our findings.

**Proposition 1.** The threshold state $R$ is (a) decreasing in $\lambda$, (b) increasing in $\bar{s}$, (c) increasing in $\mu$, and (d) increasing in $p$.

Thus, the threshold decreases as the relative traffic intensity increases for the ad-hoc delivery channel ($\lambda/\mu$) and as the per-order cost of an ad-hoc courier increases. Furthermore, there are situations where pooled policies are optimal (heavy traffic, cheap ad-hoc prices) and where splitting policies are optimal (low traffic, expensive ad-hoc prices). A larger threshold implies a larger set of orders held for committed couriers. As such, we can interpret the threshold as the value of prioritizing committed couriers over ad-hoc couriers. We note that if the platforms uses a larger number of committed couriers than needed in order to hedge against heavy traffic, splitting policies are expected to outperform pooled policies for the majority of normal daily operations. However, when faced with situations where platforms are operating near or above their capacity (due to driver shortages, extreme events etc.) pooling policies can cope with excess demand more efficiently. In our computational study, we validate these hypotheses.

**6.1.2. Empty Miles and Ad-hoc Payout.** Lacking from the previous model, however, are the time/bulletin board dependencies of the ad-hoc prices/choice model and the effect of empty miles on the efficiency of the committed couriers. We next present a fluid model for the split policy to highlight the relationship between these characteristics. Firstly, observe that the fluid formulation for pooled policies is equivalent to Formulation (7), because both ad-hoc and committed couriers select from the same set of orders. In split policies, however, the delivery channels act independently. We model this by considering two separate matrices (Figure 3) to store the state of orders for each channel. The removal of orders from each cell is controlled by decision variables $q$ for the ad-hoc courier channel and $r$ for the committed courier channel. We consider a probabilistic split policy, as a policy can be characterized by the rates at which orders enter each of the delivery channels. Specifically, for each cell $(\ell, d)$ pair, we decide the rate that an order will go to the bulletin board or to committed channel upon arrival. Let matrix $\Lambda = (\lambda_{\ell d}) \in \mathbb{R}^{B \times D}$ represent the rates that orders with relative lead time $\ell$ and distance $d$ enter the system. Next, let $\Lambda = \Lambda_c + \Lambda_a$ where $\Lambda_c$ and $\Lambda_a$ represent the rates that orders enter the committed courier channel and the ad-hoc delivery channel, respectively. These rates follow from the split policy the platform adopts. Thus, the probability that an arriving order with relative lead time $\ell$ and distance $d$ is assigned to the committed delivery channel is $\lambda_{\ell d}/\lambda_{\ell d}$ and the probability the order is assigned to the ad-hoc delivery channel is the $(\lambda_{\ell d} - \lambda^c_{\ell d})/\lambda_{\ell d}$. We now present Formulation (8) for the probabilistic split
policy where the objective is to find an optimal split of \( \Lambda \) into \( \Lambda = \Lambda_c + \Lambda_a \) which minimizes the cost of expired orders, the expected payment to the ad-hoc couriers, and the total wage paid to the committed couriers:

\[
\min \{ y_a, y_c, q, r, \lambda_a, \lambda_c \}
\]

\[
\sum_{\ell=1}^{B} \sum_{d=1}^{D} q_{\ell d} \cdot p_{\ell d} + w \cdot M + \theta \left( \sum_{d=1}^{D} y_{a d}^c + y_{a d}^c \right)
\]

\[
\text{s.t. } \begin{align*}
y_{a \ell+1, d}^a - y_{a \ell, d}^a &= q_{\ell d} - \lambda_{a \ell d}^a \\
y_{c \ell+1, d}^c - y_{c \ell, d}^c &= r_{\ell d} - \lambda_{c \ell d}^c \\
\sum_{d=1}^{D} \sum_{\ell=1}^{B} q_{\ell d} + q_0 &= \mu \\
\psi \cdot I + \frac{1}{V} \sum_{d=1}^{D} \sum_{\ell=1}^{B} r_{\ell d} + T \sum_{d=1}^{D} \sum_{\ell=1}^{B} r_{\ell d} &\leq M \\
M - W \cdot \psi &\leq 0 \\
\lambda_{a \ell d}^a + \lambda_{c \ell d}^c &= \lambda_{\ell d} \\
y_{a \ell d}^a \geq 0, q_{\ell d} \geq 0, \lambda_{a \ell d}^a \geq 0 \quad &\forall \ell = 1, \ldots, B, d = 1, \ldots, D \\
y_{c \ell d}^c \geq 0, r_{\ell d} \geq 0, \lambda_{c \ell d}^c \geq 0 \quad &\forall \ell = 1, \ldots, B, d = 1, \ldots, D \\
\psi &\in \{0, 1\} \\
M &\in 0 \cup \mathbb{Z}_+ .
\]

Naturally, this formulation is similar to Formulation (4), however there are duplicate flow balance constraints for the cells (one for each delivery channel). Additionally, we add Constraint (8g) to construct our probabilistic split policy.

**Theorem 4.** For equivalent parameter input the pooled (7) and split (8) fluid formulations are equivalent.

Theorem 4 states that given the same input parameters, an optimal solution to one formulation can be transformed into an optimal solution to the other formulation and that these solutions have the same objective value (the proof is given in the appendix).

However, as a consequence of the differences between the two policies, some parameters may actually be different. For example, the time to pickup \( T \) that is implied for a given \( I \) may be different for the two policies. A feature of a split policy is that specific orders can be allocated to the committed delivery channel, and this can be exploited to reduce the time to pickup \( T \), i.e., the empty miles, given the same number committed couriers. Let \( T_P \) and \( T_S \) be the time to pickup for the pooled and split policies, respectively. It is easy to see that if \( T_S < T_P \), then the split policy outperforms the pooled policy in steady state. Essentially, the committed couriers are utilized more efficiently leading to fewer expired orders, less ad-hoc payout and/or potentially fewer used couriers.
- each of which translates to cost savings. In practice, this result depends on a real-time split policy that reduces the empty miles without affecting the arrival process of orders to the ad-hoc delivery channel.

The main mechanism by which real-time split policies reduce the empty miles is *batching*, which we define as postponing the decision to assign an order to a delivery channel. Thus, the platform is able to “cherry-pick” the orders to give to the committed couriers, before they are affected by the inherent uncertainty associated with the bulletin board. While beneficial for the efficient use of committed couriers, batching also spawns downstream effects by altering the arrival process of orders to the bulletin board. Orders arriving to the bulletin board will have a relative lead time that is smaller than when there is no batching. As we have shown in Corollary 1, this translates to larger ad-hoc prices. We conjecture that the empty miles savings do not always outweigh the cost increase in ad-hoc courier payout. However, in same-day delivery settings with high order volume and relatively low ad-hoc courier payout per order, a small batching window may result in significant empty miles savings, but a negligible cost increase - we validate this notion in our simulation experiments.

### 6.1.3. Deterministic Planning.

Once an order is allocated to the committed courier delivery channel in a split policy, the platform has full control over the order and its assignment to a committed courier. This implies that the platform can determine whether or not it is possible to deliver an order on time if it is allocated to the committed courier delivery channel. As a consequence, the platform will never allocate orders to the committed courier delivery channel if they cannot be delivered on time, but instead would allocate them to the ad-hoc courier delivery channel, i.e., leave them on the bulletin board. Furthermore, if an order can be served by a committed courier in a future period, it may be beneficial to postpone allocating the order to a delivery channel. From a fluid model perspective, this implies that the committed courier delivery channel should never overflow or be underutilized (i.e., allocating orders to the committed courier delivery channel at a rate larger/smaller than it is capable of serving). This intuition is supported by Theorem 5 (with proof in the appendix), which states that there is an optimal solution to the split policy fluid formulation that allocate orders to the committed courier delivery channel at the exact rate that they can be served.

**Theorem 5.** There exists an optimal solution to the split policy fluid formulations where \( X_{td} = r_{td} \) for all \( t = 1, ..., B \) and \( d = 1, ..., D \).

A *holdout set* is a set of orders that is held for committed couriers to serve after they have completed the delivery of their currently assigned order (i.e., orders that are not present on the bulletin board). We are familiar with the concept of a simple holdout set in the context of batching,
as described in the previous section. Such a set is especially useful in scenarios of low traffic, as orders are held for committed couriers in future periods to reduce their idle time and to avoid paying ad-hoc couriers for deliveries that can be made by committed couriers. Additionally, however, holdout sets can be viewed as a tentative plan for committed couriers. As new orders arrive, we may find that they would further reduce the empty miles of committed couriers when compared to the orders currently in the holdout set. As such, a holdout set can be dynamically managed intelligently by using projected locations for the committed couriers which may outperform managing the set naively through simple order batching based on the placement times.

6.2. Online Policies

We next describe how the concepts of pooling, batching, and holdout sets can be incorporated when coordinating the two courier delivery channels in practice. Firstly, however, we observe that greedily assigning orders to the nearest idle committed courier, while a necessary assumption in our fluid models, is not necessarily an effective strategy in practice. We consider the more realistic case where the platform assign orders to committed couriers by repeatedly solving a matching problem (e.g., in a rolling horizon framework). Specifically, let $\tau = 1, 2, \ldots$ represent decision points separated by time intervals of equal length $\varepsilon$ (depicted in Figure 8). We refer to the time periods between decision points as arrival horizons. In an arrival horizon orders and ad-hoc couriers arrive, but the platform does not make any assignments of orders to committed couriers. The handling of orders that arrive in an arrival horizon is predetermined and dictated by the coordination policy, $\pi$. When an ad-hoc courier arrives, the courier observes the bulletin board and selects (or does not select) an order and leaves the system, regardless of the coordination policy. That is, even though the platform is not making any active decisions, the state of the bulletin board is constantly changing during

![Figure 8](image-url)
arrival horizons. At decision points $\tau$ the platform makes order assignment decisions for committed couriers and potentially other order coordination decisions, depending on the policy. Regardless of the policy, irrevocable order assignment decisions only involve idle committed couriers. As a consequence, committed couriers are assigned one order at a time. The coordinating policies differ in the set of orders considered for this assignment. Thus, we can fully characterize a coordinating policy $\pi$ by defining: (1) the management of order arrivals during arrival horizons; (2) the matching of orders and idle committed couriers at decision points; and (3) the management of unmatched orders at decision points. We consider three policies; pooled, split, and look ahead.

**Pooled.** Let $\tau$ be the current decision point. Orders arriving in the arrival horizon $[\tau - 1, \tau)$ are immediately added to the bulletin board and are able to be selected by ad-hoc couriers as they arrive to the platform in real time. At decision point $\tau$, we solve a bipartite matching between all orders on the bulletin board and all idle committed couriers. An edge between an order $i$ and idle committed courier $m$ is present if order $i$ can be delivered on time by courier $m$ and the cost of using the edge $c_{im}$ is the distance from current location of the courier (the last delivery location) to the pickup location of the order. Each courier can be matched to at most one order ($\sum_{i \in N} x_{im} \leq 1 \ \forall m \in M$) and each order can be matched to at most one courier ($\sum_{m \in M} x_{im} \leq 1 \ \forall i \in N$). We solve a min-cost maximum matching problem between the nodes in $N$ and $M$. This is accomplished by first solving a maximum matching (maximizing the number matches),

$$n_{\text{max}} := \max \left\{ \sum_{i \in N} \sum_{m \in M} x_{im} \mid \sum_{i \in N} x_{im} \leq 1 \ \forall m \in M, \ \sum_{m \in M} x_{im} \leq 1 \ \forall i \in N \right\}, \tag{9}$$

and then solving a min-cost matching with the added constraint $\sum_{i \in N} \sum_{m \in M} x_{im} = n_{\text{max}}$. The optimal solution to this min-cost maximum matching is then executed immediately, and the associated orders are removed from the bulletin board and assigned to the identified idle committed couriers.

**Batched.** The batched policy is an order splitting policy. Again, let $\tau$ be the current decision point. Orders arriving in the arrival horizon $[\tau - 1, \tau)$ are immediately added to the holdout set $\hat{N}$. If the number of orders in $\hat{N}$ is greater than $N_{\text{max}}$ (the maximum cardinality of the holdout set) then the oldest members of the set are removed and placed on the bulletin board (first in first out). At decision point $\tau$, we solve a min-cost maximum matching problem (the same as the pooled policy) between the orders in the holdout set and the idle committed couriers. The solution to this matching problem is then executed by the selected idle couriers. Orders not matched with a committed courier remain in the holdout set.

**Look ahead.** The look ahead policy belongs to the class of split policies and similar to the batched policy, orders arriving in arrival horizon $[\tau - 1, \tau)$ are added to the holdout set $\hat{N}$. At decision
point $\tau$, we again solve a min-cost maximum matching problem between orders in the holdout set and idle committed couriers. The solution to this matching problem is immediately executed by the selected idle committed couriers. However, rather than keeping all remaining orders in the holdout set, only a subset of these orders is kept in the holdout set. Let the “time until idle” for committed courier $m$ be 0 if courier $m$ is idle and be the delivery time of their current order minus the current time if they have an order assigned. Let $\hat{M}$ be the set of $\min(N_{\text{max}}, M)$ committed couriers with the smallest “time until idle”. We consider a bipartite matching between the set of orders that were not selected in the aforementioned matching and the committed couriers in $\hat{M}$ where edges are present if order $i$ can feasibly be served by courier $m$ after courier $m$ completes its currently assigned order. The cost of an arc is the same as previously defined. Finally, a min-cost maximum matching is solved and all orders selected in the solution become the new holdout set. All orders that are unmatched are immediately added to the bulletin board.

In this manner, the construction of the holdout set looks one decision ahead for the committed couriers. We note that for each of these three policies, once an order is added to the bulletin board it is never removed unless served by a courier. In the case of pooled, this means selected by an ad-hoc courier or assigned to a committed courier. In batched and look ahead policies, this only occurs when an ad-hoc courier selects the order from the bulletin board. The flow of orders is one directional; an order may be held for committed couriers and then placed on the bulletin board, but never removed for the bulletin board to be held for committed couriers. Once it is determined that the order is not a fit for the committed couriers, it is placed on the bulletin board and is not reconsidered for committed matching. The maintenance of a full routing plan for the committed couriers is beyond the scope of this paper, and we believe these three policies are sufficient to highlight the key considerations for coordinating between multiple courier types, which we illustrate in the following section.

7. Simulation Experiments

In this section, we investigate the performance of the three policies defined in the previous section (pooled, batched, and look ahead) via simulation experiments. We have a constant 1 unit per minute vehicle speed and order O-D distances uniformly distributed on $(0, 30]$. Specifically, order pickup locations are drawn uniformly at random in a circular region with radius 20 distance units. For each pickup location, a delivery location is drawn uniformly at random for a circular region with the pickup location as the center with radius 30 distance units. We assume $\lambda = 3$ orders per minute and $\mu = 2$ ad-hoc couriers per minute. Consistent with Assumption 1, we let $\alpha_{d\ell} = -11.5 - .1d + .05\ell \forall d = 1,...,D, \ell = 1,...,B$. Furthermore, we set $\beta = 2, w = \$30/hr, T = 15$ minutes, and $\theta = \$50$. The manner in which ad-hoc couriers select orders in the simulation is as follows.
Ad-hoc couriers arrive on the center region of radius 20 (the same as the order pickup locations) at random and select orders that have the smallest empty miles to their randomly generated location. This is because the pricing model prices orders depending on their lead time and distance and as such we assume the couriers would simply select the order that is closest to them (and receive the associated payout). Of course, we still have a null probability (probability they select no order) upon their arrival. Finally, we have a warm-up period of 200 simulation minutes and collect metrics for a time period of 2000 minutes. The total cost of a simulation run for any policy is the sum of the wages paid to committed couriers, the payout to ad-hoc couriers, and the cost of expired orders.

Our simulation is implemented in python and is event based. The events are: order arrival, ad-hoc courier arrival, and decision point reached. As such, we have directly implemented the policies outlined in the previous section. We let the arrival horizon length \( \varepsilon \) be 2 simulation minutes and omit our search results to find the best performing \( \varepsilon \), as that is not the focus of this paper. However, we find that 2 simulation minutes performs well because it is small enough to have little effect on the relative lead times of orders and idle times of committed couriers, while allowing for a significant reduction in empty miles. In terms of the maximum holdout size for the batched and look ahead policies, we consider sizes of \( N_{\text{max}} = \lfloor M/4 \rfloor, \lfloor M/2 \rfloor, \lfloor 3M/4 \rfloor, M \).

### 7.1. Results

We first consider a case with 60 minute lead times. We realize that different policies may perform better with different fleet and holdout set sizes. As such, when comparing the policy performance in this section, we first perform a grid search on \( M \) and \( N_{\text{max}} \) for each of the three policies and report the results for the best configuration. We solve the fluid model with the fixed value of \( M \) to obtain the price matrix \( P \) to be implemented in the simulation. The average ad-hoc price is \( \approx \$10/\text{order} \) for all \( M \) between 20 and 60. The results are reported in Table 1 and the columns “% Ad-hoc”, “% Committed”, and “% Expired” refer to the proportion of orders that were successfully delivered by ad-hoc couriers, committed couriers or orders that expired.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Total cost</th>
<th>( M )</th>
<th>( N_{\text{max}} )</th>
<th>% Ad-hoc</th>
<th>% Committed</th>
<th>% Expired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look ahead</td>
<td>69,357.29</td>
<td>36</td>
<td>( \lfloor M/4 \rfloor )</td>
<td>57.1</td>
<td>41.9</td>
<td>1.0</td>
</tr>
<tr>
<td>Pooled</td>
<td>71,014.46 (+2.4%)</td>
<td>34</td>
<td>-</td>
<td>62.9</td>
<td>35.7</td>
<td>1.4</td>
</tr>
<tr>
<td>Batched</td>
<td>71,011.93 (+2.4%)</td>
<td>38</td>
<td>( \lfloor M/2 \rfloor )</td>
<td>56.8</td>
<td>42.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

The look ahead policy performs the best among the three, with the pooled and batched policies performing on average 2.4% worse. We note that when the compensation of ad-hoc and committed couriers is similar, batched and pooled policies have a similar total cost even with different
committed fleet sizes. Interestingly, we see a key difference between the pooled and the batched and look ahead policies. We see that ad-hoc couriers serve almost $2/3$ of the demand (which is the upper limit, as $\mu/\lambda = 2/3$) in the pooled policy. The use of ad-hoc couriers for a majority of the orders induces uncertainty, and we see a higher number of expired orders. When using the batched and look ahead policies on the other hand, more orders are allocated to the committed courier delivery channel. The look ahead policy is able to utilize committed couriers more efficiently than the batched policy by planning future committed courier decisions. With two fewer committed couriers, the look ahead policy allocates roughly the same fraction of orders to the committed courier delivery channel and achieves roughly the same fraction of expired orders as the batched policy.

Next, we investigate how the performance of these policies changes when increasing the lead time; Table 2 shows results for 120 minute lead times. With the higher relative lead times, the average ad-hoc price is $\approx \$8.20$ per order for the new $P$. With the increased lead time the average price has decreased due to the preference of ad-hoc couriers to serve orders with longer lead times (and they accept lower prices in that case).

<table>
<thead>
<tr>
<th>Policy</th>
<th>Total cost</th>
<th>$M$</th>
<th>$N_{max}$</th>
<th>% Ad-hoc</th>
<th>% Committed</th>
<th>% Expired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Look ahead</td>
<td>62,075.46</td>
<td>36</td>
<td>27 ($\lfloor \frac{M}{4} \rfloor$)</td>
<td>61.4</td>
<td>38.1</td>
<td>0.5</td>
</tr>
<tr>
<td>Pooled</td>
<td>63,489.20 (+2.3%)</td>
<td>36</td>
<td>-</td>
<td>63.9</td>
<td>35.6</td>
<td>0.5</td>
</tr>
<tr>
<td>Batched</td>
<td>62,652.47 (+1.0%)</td>
<td>34</td>
<td>34 ($M$)</td>
<td>63.7</td>
<td>35.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

The results are similar to the 60 minute case, but the gap between the batched and look ahead policies has shrunk (2.4% to 1.0%), while it remains essentially the same between the pooled and look ahead policies (2.4% to 2.3%). We notice that with the lower ad-hoc cost (on average) due to the larger lead times, the ad-hoc couriers are utilized more in the batched and look ahead policies than in the 60-minute lead time case. We also observe that the best holdout set size is larger for the 120-minute case than for the 60-minute case, likely due to the smaller chance of order expiration. It is also larger for the batched policy than the look ahead policy. Intuitively, we conjecture this is because the look ahead policy creates significantly more empty miles savings with a smaller holdout set than the batched policy, and will show this in the next subsection of the results.

Figure 9 shows the results of the search for the best committed courier fleet size for the three policies for the 60 minute case. Each data point is using the best performing holdout set size for the batched and look ahead policies. As expected, pooled outperforms batched when the fleet size is low, and vice versa. Interestingly, the look ahead policy is the best regardless of fleet size.
7.1.1. Empty Miles and Relative Lead Time. As we discussed in Section 6.1.2, the use and size of the holdout set affect the empty miles of committed couriers and the relative lead time of orders on the bulletin board. This notion is validated by our simulation. Figure 10 shows the distribution of the relative lead time of orders on the bulletin board at various times in the simulation for the different policies and their optimal configurations of $M$ and $N_{\text{max}}$ from Table 1. Clearly, the pooled policy has the largest relative lead times as orders are immediately added to the
bulletin board upon their arrival. Both the batched and look ahead policies see significantly smaller relative lead times. Their means are similar, but the look ahead policy tends to have more outlying replications with lower relative lead times relative to the batched policy. This is due to the batched policy following a first in first out principle for the holdout set, whereas the look ahead policy has more uncertainty when it comes to how long an order remains in the holdout set. Next, we explore in more detail how the size of the holdout set affects the empty miles of the committed couriers and the relative lead times of orders posted on the bulletin board. The results can be found in Table 3 (using the number of committed couriers, $M$, shown in Table 1). For reference, the empty miles for the committed couriers in the pooled policy (configuration from Table 1) is 16.5. We see that for each unit of increase of the holdout set size the batch policy’s empty miles see a greater reduction than the look ahead policy’s empty miles. This is due to the fact that the batching policy only relies on the size of the holdout set to drive the reduction of the empty miles (more choices for the matching leads to better solution), while the look ahead policy also uses future information regarding committed couriers. As consequence, the look ahead policy can recognize the majority of empty mile savings with a small holdout set. For increasing holdout set size the relative lead times on the bulletin board decrease for both policies. The reduction is slower per unit of holdout set size for the look ahead policy and we see that when the holdout set is large the look ahead policy actually has lower empty miles and higher relative lead times than the batching policy. The look ahead policy only replaces an order in the holdout set if an order with fewer empty miles for a committed courier arrives. The batched policy, on the other hand, has less control due to first in first out policy of the holdout set and the reduction in relative lead time on the bulletin board is less certain. As a consequence, the look ahead policy is able to reduce the empty miles for the committed couriers with a smaller holdout set than the batched policy which also implies a smaller impact on the relative lead times of orders on the bulletin board. Finally, we observe that reduced empty miles for the committed couriers do not automatically result in lower total cost, because of the increased cost associated with orders posted on the bulletin board (as their relatively lead time is smaller).

Table 3  Total costs, average empty miles (EM), and average relative lead times (RL) on the bulletin board for different holdout set sizes.

<table>
<thead>
<tr>
<th>Policy</th>
<th>$M/4$</th>
<th>$M/2$</th>
<th>$3M/4$</th>
<th>$M$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EM (d)</td>
<td>RL (t)</td>
<td>EM (d)</td>
<td>RL (t)</td>
</tr>
<tr>
<td>Look Ahead</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total cost</td>
<td>14.9</td>
<td>32.4</td>
<td>14.4</td>
<td>29.6</td>
</tr>
<tr>
<td>Batched</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total cost</td>
<td>16.2</td>
<td>35.0</td>
<td>15.3</td>
<td>30.4</td>
</tr>
<tr>
<td></td>
<td>72,162.61</td>
<td>71,011.93</td>
<td>72,013.89</td>
<td>75,321.93</td>
</tr>
</tbody>
</table>
7.1.2. Robustness. In the previous section we found that the look ahead policy performs the best when we have an accurate forecast of the demand and ad-hoc supply arrival processes. Now, we will investigate how the performance of these policies is affected by unexpected deviations of the demand and ad-hoc supply arrival processes in Figure 11.

Figure 11a shows the effect of scaling the demand arrival rate. The same $M$, $P$ and $\mu$ is used for all of the policies, and we consider 60 minute lead times. The x-axis is the amount that the base order arrival rate ($\lambda = 3$) is scaled by for the the simulation. For example, 1.0 implies that orders arrive at the rate expected, 1.2 implies a 20% increased rate of arrivals, and so on. As such, x-axis values $< 1$ are unexpected low traffic scenarios and $> 1$ are unexpected heavy traffic scenarios. We see that batched performs better than pooled in low traffic due to prioritizing the prepaid committed couriers and avoiding ad-hoc couriers when they are unnecessary. On the other hand, pooled outperforms batched in heavy traffic. Surprisingly, the look ahead policy is the best of both worlds, as it remains robust in both low and heavy traffic scenarios. We have a similar result in Figure 11b, where x-axis values $< 1$ are unexpected heavy traffic scenarios and $> 1$ are unexpected low traffic scenarios, as we are scaling the base ad-hoc arrival rate ($\mu = 2$) instead and keeping the order arrival rate constant. As such, $< 1$ indicates less ad-hoc couriers arrive than expected, and $> 1$ more than expected. Again, the pooled policy performs well in the heavy traffic scenario because it better utilizes ad-hoc couriers. The solution quality degrades in increasing $\mu$ however as it over utilizes ad-hoc couriers when compared to committed couriers. By prioritizing committed couriers when $\mu$ grows large, the batched and look ahead policies avoid unnecessarily utilizing additional ad-hoc couriers.
8. Discussion and Recommendations

Using a hybrid delivery system introduces a new challenge in the planning and operation of crowdsourced delivery platforms: how to simultaneously and effectively manage two delivery channels? An analysis of the fluid models we developed for the planning of a hybrid delivery system shows that it can achieve significant cost savings, especially by platforms which operate a pure bulletin board ad-hoc courier system at the margins of its delivery capacity. By introducing a small number of committed couriers, e.g., through crowdsourced shift sign-ups, the platform can take the pressure of the ad-hoc courier delivery channel. Platforms that currently operate a pure committed courier system (in the extreme case employee couriers) must first build a healthy market of ad-hoc couriers, which will take time. For large companies with name recognition and a good reputation, this is less of a hurdle and such an initiative can pay off relatively quickly. Of course, acquisitions of existing bulletin board platforms offer the opportunity to immediately capitalize on an existing (and hopefully, loyal) base of ad-hoc couriers. Ultimately, the use of committed and ad-hoc couriers has the potential to significantly improve the profit of crowdsourced delivery platforms.

Our research has also generated insight into how to best manage the operations of a hybrid delivery system. An intelligently designed and carefully tuned order splitting policy performs best among the policies considered. However, a natural and simple pooled policy performs reasonably good too. A pooled policy has the advantage that it is simple and does not require maintaining a set of orders for each courier type. Surprisingly, the pooled policy is naturally robust and can handle demand surges even though it is myopic. Platforms operating in environments of high demand and frequent surges - which is typical in this domain - and that are considering using both committed and ad-hoc couriers may want to adopt the following strategy. Start with a pooled policy while testing and tuning an intelligent splitting policy; this allows time to optimize the splitting policy (and constructs a baseline for comparison), while hedging against demand surges and lack of ad-hoc courier engagement - common challenges faced by new platforms.

As we’ve mentioned, the related literature on this topic 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. A bulletin board 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 this choice may be influenced by the three previous alternatives,
may simply be an avoidance of rejection penalties or have been made for any number of reasons. A bulletin board system, on the other hand, can learn individual courier preferences by presenting all available options. After which, this information can be used to curate personalized menus of orders with custom prices which can benefit both the couriers and platform. As such, a bulletin board system can still have influence over self-assigning couriers, much 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 construction of a holdout set, given perfect demand information. We discussed online policies for holdout set management, but finding a partition of a known set of orders into routes for committed couriers and a set left to the bulletin board 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.

References


Appendix

Proof of Theorem 1

Recall Formulation (3) and consider the linear relaxation. That is, \( M \geq 0 \) and \( \psi \geq 0 \). In the IP, we use \( \psi \), \( W \) and constraint (3d) to handle the case when there are 0 couriers. Clearly, for sufficiently large \( \theta \) we have \( M \geq I \) in the IP. As such, when analyzing the LP, we let \( \psi = 1 \) and remove constraint (3d). We have:

\[
\min_{y, r, M} w \cdot M + \theta \sum_{d=1}^{D} y_{1d}
\]

s.t.
\[
y_{\ell+1, d} - y_{\ell d} = r_{\ell d} - \lambda_{\ell d} \quad \forall \ \ell = 1, ..., B, \ d = 1, ..., D
\]
\[
I + \frac{1}{V} \sum_{d=1}^{D} d \sum_{\ell=1}^{B} r_{\ell d} + T \sum_{d=1}^{D} \sum_{\ell=1}^{B} r_{\ell d} \leq M
\]
\[
y_{\ell d} \geq 0, \ r_{\ell d} \geq 0 \quad \forall \ \ell = 1, ..., B, \ d = 1, ..., D
\]
\[
M \geq 0.
\]

We can represent objective (10a) as a function of the single variable \( M \), which we denote at \( c(M) \). Consider formulation (10) where \( M \) is a parameter. At optimality, constraint (10c) is tight making the feasible region just system of linear equations and the objective is to simply minimize the term \( \theta \sum_{d=1}^{D} y_{1d} \). As a consequence of constraint (10b) we have:

\[
y_{1d} = \sum_{l=1}^{B} \lambda_{ld} - \sum_{l=1}^{B} r_{ld} \quad \forall d = 1, ..., D.
\]

Furthermore, from constraint (10c) we have:

\[
M = I + \sum_{d=1}^{D} \left( \frac{d}{V} + T \right) \sum_{l=1}^{B} r_{ld}.
\]

Clearly, maximizing \( \sum_{d=1}^{D} \sum_{l=1}^{B} r_{ld} \) gives an optimal solution to the problem. This is done by considering each \( y_{1d} \) from equation (11) separately and first maximizing \( \sum_{l=1}^{B} r_{\ell 1} \), then \( \sum_{l=1}^{B} r_{\ell 2} \) and so on, until equation (12) is satisfied. As such, \( c(M) \) can be completely written in terms of the parameters of the problem.

Finally, we will show that \( c(M) \) is piecewise linear convex with \( D + 1 \) breakpoints. We say the first breakpoint is when \( M = I \) as the derivative is undefined at this point. Next, consider the \( D \) points

\[
M_d = I + \sum_{j=1}^{d} \left( \frac{j}{V} + T \right) \sum_{l=1}^{B} \lambda_{ld} \quad \forall d = 1, ..., D,
\]

that correspond to the solutions where all \( \sum_{l=1}^{B} r_{lj} = \sum_{l=1}^{B} \lambda_{lj} \) for \( j = 1, ..., d \) and \( \sum_{j=d+1}^{D} \sum_{l=1}^{B} r_{lj} = 0 \). Then,

\[
c(M_d) = w \cdot \left[ I + \sum_{j=1}^{d} \left( \frac{j}{V} + T \right) \sum_{l=1}^{B} \lambda_{ld} \right] + \theta \cdot \left[ \sum_{j=d+1}^{D} \sum_{l=1}^{B} \lambda_{lj} \right]
\]

is the value of the objective at each of these points. The slope leaving point \( M_d \) is

\[
m_d = \frac{c(M_{d+1}) - c(M_d)}{M_{d+1} - M_d},
\]

which is \( w - \frac{\theta}{(d/V) + T} \) for \( d = 1, ... D - 1 \) and is \( w \) for \( d = D \). As such, the slope between each point is linear and is increasing for increasing \( d \), making \( c(M) \) piecewise linear convex.

\[\square\]
Proof of Lemma 1.

For conciseness we remove the subscript $s$, and consider a single contiguous subsequence

\[ q_\ell = q_{\ell+1} = \ldots = q_{\ell+(n-1)} < q_{\ell+n}. \]

Let

\[ \Delta = \frac{\sum_{k=\ell+1}^{n} (q_k - (n-1)q_\ell)}{n}. \]

then $\sum_{i=\ell}^{\ell+n} \Delta_i = 0$ and $q_i + \Delta_i = q_i$, $\forall \ell \leq i \leq \ell+n$. Clearly, this replacement still satisfies constraint (4c) as the sum does not change. We now show that the $n$ equality constraints from (4a) are still satisfied. That is,

\[ y'_i - y'_{i+1} + q = \lambda_i, \forall i = [\ell, \ell+1, \ldots, \ell+n]. \]

Let $y'_{\ell+n} = y_{\ell+n} - \Delta_{\ell+n}$ and $y'_1 = y_1 + \sum_{k=1}^{n-1} \Delta_k$, then it is simple to verify the constraints are satisfied. The $n$th constraint is satisfied because $q_{\ell+n}$ was increased by $\Delta_{\ell+n}$ while $y_{\ell+n}$ was decreased by the same. The $n-1$ constraint holds by the fact that $\sum_{i=\ell}^{\ell+n} \Delta_i = 0$, as $q_{\ell+n}$ is increased by $\Delta_{\ell+n}$, $y_{\ell+n}$ is increased by $\sum_{k=\ell}^{\ell+(n-2)} \Delta_k$ and $y_{\ell+n}$ is decreased by $\Delta_{\ell+n}$. The other $n-2$ constraints hold because the $y_{i+1}$ term is increased by $\sum_{k=\ell}^{n-1} \Delta_i$, $y_i$ is increased by $\sum_{k=\ell}^{n-1} \Delta_i$, and $q_i$ is increased by $\Delta_i$.

Next, we show that this replacement does not worsen the objective which is given by

\[ F(q) = \frac{1}{\beta} \sum_{i=1}^{B} (q_i \ln(\frac{q_i}{q_0}) - q_i \alpha_\ell) + C y_1. \]

The term $C y_1$ is independent of the replacement, as $y'_1 = y_1 + \sum_{k=1}^{n} \Delta_k$, which is just $y_1$. Let $f(q) = q \ln(q/q_0)$ where $q_0$ is a constant. As $f''(q) = 1/q$ is strictly positive for all $q > 0$, $f(q)$ is then strictly convex on the same domain. We wish to show

\[ n f(\bar{q}) < \sum_{k=\ell}^{\ell+n} f(q_k), \]

which is a direct result of Jensen’s inequality

\[ f\left( \frac{\sum_{k=\ell}^{\ell+n} q_k}{n} \right) < \frac{\sum_{k=\ell}^{\ell+n} f(q_k)}{n}. \]

Lastly, let $g(q, \alpha) = -q \alpha$. We will show

\[ \sum_{k=\ell}^{\ell+n} g(\bar{q}, \alpha_k) \leq \sum_{k=\ell}^{\ell+n} g(q_k, \alpha_k). \]

\[ \sum_{k=\ell}^{\ell+n} g(\bar{q}, \alpha_k) = \sum_{k=\ell}^{\ell+n} g(q_k + \Delta_k, \alpha_k) \]

\[ = \sum_{k=\ell}^{\ell+n} -(q_k + \Delta_k)\alpha_k \]

\[ = \sum_{k=\ell}^{\ell+n} g(q_k, \alpha_k) - \Delta_{\ell+n} \alpha_{\ell+n} - \left( \sum_{k=\ell}^{\ell+(n-1)} \Delta_k \right). \]
The desired results then follows directly from the fact that \(-\Delta_{\ell+n} \alpha_{\ell+n} - (\sum_{k=\ell}^{\ell+(n-1)} \Delta_k) \geq 0\), which a consequence of
\[
\alpha_{\ell+n} \geq \bar{\alpha} = \frac{\sum_{k=\ell}^{\ell+(n-1)} \alpha_k \Delta_k}{-\Delta_{\ell+n}},
\]
where \(\bar{\alpha}\) is the weighted mean of \(\alpha_{\ell}, \alpha_{\ell+1}, ..., \alpha_{\ell+(n-1)}\), since \(\sum_{k=\ell}^{\ell+(n-1)} \Delta_k = -\Delta_{\ell+n}\). Due to the fact that \(\alpha_{\ell+n} \geq \alpha_i \forall i = [\ell, \ell+1, ..., \ell+(n-1)]\) from our assumption that \(\alpha_{\ell+i}\) is non-decreasing in \(\ell\), it must be at least as large as \(\bar{\alpha}\).

**Proof of Theorem 2**

For conciseness we remove the subscript \(s\) and we prove the above theorem by contradiction. Without loss of generality, consider the first violation of statement (5) and the associated contiguous subsequence defined by
\[
i = \min\{\ell : q^*_{\ell-1} < q^*_{\ell} \forall \ell = [1, 2, ..., B]\},
\]
and
\[
k = \min\{\ell : q^*_{\ell} = q^*_{\ell-1}, \forall \ell < i\}.
\]
We then have
\[
q^*_{k-1} > q^*_{k} = ... = q^*_{i-1} < q^*_{i}.
\]
Recall the objective of formulation 6 is equivalent to
\[
F(q) = \frac{1}{\beta} \sum_{\ell=1}^{B} (q_{\ell} \ln\left(\frac{q_{\ell}}{q_{0}}\right) - q_{\ell} \alpha_{\ell}) + Cy_1.
\]
Then, applying Lemma 1 results in a new solution
\[
q_{\text{new}} = \{q_{\ell} : q_{\ell} = q^*_{\ell} \forall \ell \notin \{k, ..., i\}, q_{\ell} = \bar{q} \forall \ell \in \{k, ..., i\}\},
\]
where \(\bar{q}\) is from the statement of Lemma 1 and \(F(q_{\text{new}}) < F(q^*)\). Note that we do not yet have the desired contradiction, as this \(q_{\text{new}}\) may induce the “\(>\)” between the \(k-1\)th entry and the \(k\)th entry to become “\(<\)”, which is problematic. However, there exists some \(\varepsilon > 0\) where \(\bar{q} = \varepsilon q^*_k + (1-\varepsilon)\bar{q}\), such that \(q^*_{k-1} < \bar{q}\). Let
\[
\bar{q} = \{q_{\ell} : q_{\ell} = q^*_{\ell} \forall \ell \notin \{k, ..., i\}, q_{\ell} = \bar{q} \forall \ell \in \{k, ..., i\}\}.
\]
By the strict convexity of \(F\) (shown in the proof of Lemma 1) \(F(\bar{q}) < F(q^*)\), which is the desired contradiction.

**Proof of Theorem 3**

First, we will view our problem is a minimization problem. Next, for the sake of this proof we ignore states \(-M, ..., -1\) as there are no decisions. As such, for the remainder of this proof let \(S = 0, 1, ..., N\). In order to prove the theorem is suffices to verify the following conditions (from Puterman (2014)):
\begin{enumerate}
  \item \(r(s, a)\) is non-decreasing in \(s\) for all \(a \in A'\) where \(A_s = A'\) for all \(s \in S\),
  \item \(q(k|s, a) \equiv \sum_{j=k}^{\infty} p(j|s, a)\) is non-decreasing in \(s\) for all \(k \in S\) and \(a \in A'\),
  \item \(r(s, a)\) is a subadditive function on \(S \times A'\), and
  \item \(q(k|s, a)\) is a superadditive function on \(S \times A'\) for all \(k \in S\).
\end{enumerate}
We will verify these conditions for the case referenced in the main text where \( c < p \). We now start with condition (1). As \( A^* = \{0, 1\} \), we first consider \( a = 0 \)

\[
r(s, 0) = \begin{cases} 
\bar{\sigma} c & s = 1, \ldots, N - 1 \\
\bar{\sigma} c + \lambda (1 - \bar{\sigma}) \theta & s = N
\end{cases}
\]

which is clearly non-decreasing. For \( a = 1 \) we have

\[
r(s, 1) = \begin{cases} 
\bar{\sigma} c + (1 - \bar{\sigma}) \mu p & s = 1, \ldots, N - 1 \\
\bar{\sigma} c + (1 - \bar{\sigma}) \mu p + \lambda (1 - \mu)(1 - \bar{\sigma}) \theta & s = N
\end{cases}
\]

which is also non-decreasing.

Condition (2) also follows almost immediately from the following. Let \( a = 0 \), and let \( p_{s, s'} \) denote the probability of going from \( s \) to \( s' \). For any \( s = 1, \ldots, N - 1, s' \) can only take values \( s - 1, s \) and \( s + 1 \). As such, \( p_{s, s - 1} + p_{s, s} + p_{s, s + 1} = 1 \) and these probabilities are the same for all \( s = 0, \ldots, N \), with the exception that for state \( N \) we let \( p_{s, s} = p_{s, s + 1} + p_{s, s + 1} \), because the chain has a finite state space. For \( k = 0 \), \( q(k|s = 0, 0) < 1 \) and \( q(k|s = 1, \ldots, N, 0) = 1 \) so the condition is true. For \( k = 1, \ldots, N - 1 \), \( q(k|s = k - 1, 0) = 1 \) and \( q(k|s = k, 0) = 0 \) is non-decreasing in \( s \). Lastly, for \( k = N \) \( q(k|s = k - 1, 0) = p_{s, s + 1} \), \( q(k|s = k, 0) = p_{s, s} + p_{s, s + 1} \). As such, condition (2) is true for \( a = 0 \). We will not show this for \( a = 1 \), as the only difference is the exact probabilities of moving up or down, which does not affect the result.

For condition (3), we utilize a following useful Lemma from Puterman (2014) which states that any real-valued function \( g(s, a) \) on \( S \times A \), with \( A = \{0, 1\} \) and \( S = 0, 1, \ldots \), that satisfies

\[
\begin{align*}
g(s + 1, 1) - g(s + 1, 0) &\leq g(s, 1) - g(s, 0)
\end{align*}
\]

for all \( s \), then \( g(s, a) \) is subadditive. This is easily verifiable for \( s = 1, \ldots, N - 2 \) as the equation is exactly 0. As the for \( s = N - 1 \) case, the result is \(-\mu \lambda \theta (1 - \bar{\sigma}) < 0 \), which verifies the condition.

In order to show condition (4) we can use the same Lemma from condition (3), with the sign reversed. We have

\[
p(s, s'|a = 0) = \begin{cases} 
\lambda (1 - \bar{\sigma}) & s' = s + 1 \\
(1 - \lambda) \bar{\sigma} & s' = s - 1
\end{cases}
\]

for \( s = 1, \ldots, N - 1 \). For \( s = N \) there is no larger state, and as such we have \( p(s = N, s = N)|a = 0) = p(s, s + 1|a = 0) + p(s, s|a = 0) \). Similarly, we have

\[
p(s, s'|a = 1) = \begin{cases} 
\lambda (1 - \bar{\sigma})(1 - \mu) & s' = s + 1 \\
(1 - \lambda)[\bar{\sigma} - (1 - \bar{\sigma})]\mu & s' = s - 1
\end{cases}
\]

for \( s = 1, \ldots, N - 1 \). For \( s = N \) there is no larger state, and as such we have \( p(s = N, s = N)|a = 1) = p(s, s + 1|a = 1) + p(s, s|a = 1) \). Condition (4) follows. \( \square \)

**Proof of Theorem 4**

We begin by showing any feasible solution to (8) can be used to construct a feasible solution to (7) that has the same objective value, and the reverse. First we will show the (8) to (7) direction, as it is the more simple case. A solution to (8) is \( y^c, y^d, q, r, \lambda^c, \lambda^d \) and \( t \), and when transforming it to a solution to (7), we let
\[ q, r \text{ and } t \text{ remain the same and } y_{\ell d} = y^c_{\ell d} + y^d_{\ell d}. \] Note, \( \lambda_{\ell d} \) is a parameter in (7) so we do not need to provide a construction. Clearly the objective has the same value and most constraints are satisfied immediately. We now show explicitly that constraint (7b) is satisfied.

\[
y_{\ell+1,s} - y_{\ell d} = q_{\ell d} + r_{\ell d} - \lambda_{\ell d} \\
= y^c_{\ell+1,s} - y^c_{\ell d} + \lambda^c_{\ell d} + y^d_{\ell+1,s} - y^d_{\ell d} + \lambda^d_{\ell d} - \lambda_{\ell d} \quad \text{From (8b) and (8c)} \\
= (y^c_{\ell+1,s} + y^d_{\ell+1,s}) - (y^c_{\ell d} + y^d_{\ell d})
\]

Next we show the (7) to (8) direction. A solution to (7) is \( y, q, r, t \). Again, we let \( q, r \) and \( t \) remain the same. We use constraint (7b) to build

\[
\delta_{\ell d} = \frac{q_{\ell d}}{\lambda_{\ell d} + y^c_{\ell+1,s} + y^d_{\ell+1,s}} \quad \forall \ell d,
\]

where \( \delta_{\ell d} \in [0, 1] \) and

\[
1 - \delta_{\ell d} = \frac{r_{\ell d}}{\lambda_{\ell d} + y^c_{\ell+1,s} + y^d_{\ell+1,s}} \quad \forall \ell d.
\]

Let \( y^e_{\ell d} = \delta_{\ell d} y_{\ell d}, \ y^d_{\ell d} = (1 - \delta_{\ell d}) y_{\ell d}, \lambda^e_{\ell d} = \delta_{\ell d} \lambda_{\ell d} \) and \( \lambda^d_{\ell d} = (1 - \delta_{\ell d}) \lambda_{\ell d} \), which satisfy constraints (8b) and (8c).

\[ \Box \]

**Proof of Theorem 5**

Let \( y^a, y^c, \lambda^a, \lambda^c \) be the values of an optimal solution to the split formulation where \( \exists \lambda^c_{\ell d} \neq r_{\ell d} \). As the rows of the matrix are independent, we consider a fixed \( d \) and leave out the subscript for clarity of presentation. Let \( \Delta_{\ell} = r_{\ell} - \lambda^c_{\ell}, \lambda^e_{\ell} = \lambda^c_{\ell} - \Delta_{\ell} \) and \( y^e_{\ell} = y^c_{\ell} - \sum_{k \leq \ell} \Delta_k \). Replacing with these new values still satisfies constraint (8c) as

\[
y^e_{\ell+1} - y^e_{\ell} = y^c_{\ell+1} - \sum_{k=\ell+1}^{B} \Delta_k - (y^c_{\ell} - \sum_{k=\ell}^{B} \Delta_k) \\
= y^c_{\ell+1,s} + y^d_{\ell+1,s} - (y^c_{\ell d} - \sum_{k=\ell}^{B} \Delta_k) \\
= r_{\ell} - \lambda^c_{\ell} + \Delta_{\ell} \\
= r_{\ell} - \lambda^e_{\ell}.
\]

We then let \( \lambda^e_{\ell} = \lambda_{\ell} - \Delta_{\ell} \) and \( y^e_{\ell} = y^c_{\ell} - \sum_{k=\ell}^{B} \Delta_k \) and by a similar algebraic substitution constraint (8b) is satisfied. Finally, as \( y^e_{0} = (y^c_{0} - \sum_{k=0}^{B} \Delta_k) \) and \( y^e_{0} = (y^c_{0} + \sum_{k=0}^{B} \Delta_k) \) we see that the objective function remains unchanged. Therefore, given an optimal solution to the split formulation, we can construct a new optimal solution such that \( \lambda^e_{\ell d} = r_{\ell d} \ \forall \ell = 1, ..., B, d = 1, ..., D. \)

\[ \Box \]