EFFICIENT ASSIGNMENT IN PEER-TO-PEER CROWDSHIPPING: AN APPROXIMATE DYNAMIC PROGRAMMING APPROACH

Emma Innocente, Jean-Sébastien Tancrez

LIDAM Discussion Paper CORE
2026 / 04

UCLouvain

LIDAM
CORE | Louvain Institute of Data Analysis and Modeling in economics and statistics

CORE

Voie du Roman Pays 34, L1.03.01

B-1348 Louvain-la-Neuve

Tel (32 10) 47 43 04

Email: immaq-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/core/core-discussion-papers

Efficient Assignment in Peer-to-Peer Crowdshipping: An Approximate Dynamic Programming Approach

Emma Innocente[a,*], Jean-Sébastien Tancrez[a]

[a] *Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, Chaussée de Binche 151, Mons, 7000, Belgium*

Abstract

Crowdshipping is a collaborative delivery system that outsources delivery tasks to ordinary citizens that act as non-professional couriers, with the potential to reduce delivery costs and improve sustainability. The crowdshippers deliver shipments by taking a short detour from their planned trips in return for a compensation fee. The dynamic assignment of crowdshippers to parcels over time is challenging as the arrival of crowdshippers and parcels is stochastic and their availability is dynamically revealed and as present assignments affect future ones. Peer-to-peer crowdshipping, which encompasses all types of crowdshipper journeys and parcel deliveries, with no restrictions on origin or destination to specific locations, is subject to a high level of uncertainty. This work presents an approximate dynamic programming approach based on value function approximation for the dynamic assignment problem of peer-to-peer crowdshipping platforms. The approach is based on the offline adaptive approximation of parcel values and provides non-myopic behavior while only solving a sequence of assignment problems no larger than in a myopic approach. Through numerical results, we demonstrate our methodology's effectiveness as, compared to a myopic benchmark, it increases the cost savings achieved via crowdshipping and reduces crowdshipper detours. Our analysis highlights the parameters affecting the relevance of implementing a non-myopic assignment method.

Keywords:
Dynamic programming, Crowdshipping, Dynamic assignment, Approximate dynamic programming, Crowd logistics

---

Declarations of interest: none.

1. Introduction

E-commerce accounted for 19.4% of global retail sales in 2023, rising from 7.4% in 2015 and expected to reach 22.6% by 2027 (Statista, 2024a). Concurrently, sharing economy services have exploded in popularity over recent years, with many expecting this trend to continue. The total value of the global sharing economy is predicted to increase from 150 billion USD in 2023 to 794 billion USD by 2031 (Statista, 2024b).

*Corresponding author.
Email addresses: `emma.innocente@uclouvain.be` (Emma Innocente), `jean-sebastien.tancrez@uclouvain.be` (Jean-Sébastien Tancrez)

In recent years, crowdshipping has garnered attention as a potentially cost-effective and sustainable alternative to traditional delivery (Mohri et al., 2023). Crowdshipping is a collaborative delivery system that falls within the sharing economy trend in that deliveries are carried out by using existing resources, namely vehicle capacity and drivers from the crowd, i.e., non-professionals. Private drivers execute deliveries using a personal vehicle on an upcoming planned trip (Archetti et al., 2016). To this end, these so-called crowdshippers detour from their direct route, within certain bounds, to pick up a parcel and deliver it to its destination (Behrend & Meisel, 2018). Crowdshippers receive a fee that compensates their detour. By exploiting excess capacity in existing personal transport for the delivery of freight, crowdshipping is inclined to reduce the number of dedicated freight delivery trucks, reducing delivery costs and supporting sustainability (Pourrahmani & Jaller, 2021). However, systematic and comprehensive analyses of this approach remain limited.

Among the various implementations of crowdshipping, the model involving in-store customers is the most extensively examined in the literature. This approach — adopted notably by Walmart (Dayarian & Savelsbergh, 2020) — relies on in-store shoppers to deliver online orders on their way home from the store within few hours. This work focuses on peer-to-peer crowdshipping (Buldeo Rai et al., 2018). It encompasses all types of crowdshipper journeys — whether regular commutes (such as home-to-work trips) or occasional ones (leisure or business travel) — and parcel deliveries, with no restrictions on origin or destination to specific locations like stores or homes. Peer-to-peer crowdshipping is therefore particularly well-suited for the delivery of parcels with sellers, buyers and crowdshippers geographically dispersed, such as in consumer-to-consumer (C2C) platforms like Vinted, Leboncoin or Facebook Marketplace. It is also well-adapted for non-conventional parcels (e.g., bulky items), which are difficult and costly to ship via more traditional delivery. The French crowdshipping company Cocolis (*www.cocolis.fr*) or the American company StowAway (*www.stowawayapp.com*) adopt this model.

The successful implementation of crowdshipping heavily depends on its operational efficiency, with the assignment of crowdshippers to parcels at its core (Le et al., 2019). Crowdshippers need to be assigned to parcels over time. Yet, both crowdshippers and parcels introduce uncertainty. Crowdshippers decide when to be available and declare their availability little to no time in advance, making it difficult to accurately predict their participation, while parcel arrivals further add to the uncertainty. Solving the dynamic assignment problem in a crowdshipping environment becomes challenging as the arrival of crowdshippers and parcels is stochastic and their availability is dynamically revealed (Savelsbergh & Van Woensel, 2016). The future availability of crowdshippers and parcels being uncertain, it implies that assigning a crowdshipper to a parcel immediately may lead to missing a better-suited crowdshipper becoming available later. As a result, current assignment decisions influence future ones. Few researches have made an attempt to address the dynamic and stochastic aspects of crowdshipping in their modeling and operational decision-making (Mousavi et al., 2024). Peer-to-peer crowdshipping is particularly subject to uncertainty, as parcels and crowdshippers originate from different, unknown locations, adding spatial uncertainty to the temporal uncertainty.

This paper focuses on solving the crowdshipper-to-parcel dynamic assignment problem for a peer-to-peer crowdshipping platform, with crowdshippers and parcels arriving stochastically and their availability dynamically revealed. We propose an approximate dynamic programming (ADP) approach based on value function approximation (VFA; see Spivey & Powell, 2004) to assign crowdshippers to parcels over time while considering the uncertainty in the future availability of crowdshippers and parcels. This approach provides non-myopic behavior by considering whether to postpone parcel assignments in anticipation of better-suited crowdshippers becoming available later. Yet, it only requires solving a sequence of assignment problems no larger than would be required with a myopic approach. An *adaptive dynamic programming algorithm* is designed to adaptively approximate parcel values offline, capturing the value of postponing parcel assignments. Using numerical experiments, we demonstrate the effectiveness of our ADP methodology relative to a myopic approach, which makes decisions based solely on available information. Our approach succeeds in increasing the cost savings achieved via crowdshipping and reducing crowdshipper detours. Moreover, those experiments highlight the parameters influencing the relevance of implementing a non-myopic assignment method.

The remainder of this paper is structured as follows. In Section 2, we review the relevant literature on the crowdshipper-to-parcel (dynamic) assignment problem. We present the problem setting in Section 3. Our approximate dynamic programming approach is described in Section 4. In Section 5, we discuss the computational experiments. An extension to our methodology to account for advance information on crowdshipper availability is presented in Section 6. We conclude and discuss potential future research in Section 7.

2. Literature review

Crowdshipping is receiving more and more interest in both practice and academia. Mohri et al. (2023) and Savelsbergh and Ulmer (2024) provide a general overview of the applications of operations research to crowdshipping. Given the scope of our paper, the literature review focuses on optimization approaches addressing the crowdshipper-to-parcel assignment problem. We briefly cover deterministic models and then focus on dynamic and stochastic approaches.

Before that, it is worth noting that same-day, on-demand, and meal delivery problems are an additional source of literature, see Mousavi et al. (2024) for relevant articles. On-demand delivery is a special case of same-day delivery, where orders are delivered within few minutes to hours. Meal delivery is a special case of on-demand delivery, where customers are delivered from a list of restaurants, typically by part-time workers. Most of this literature focuses on either companies' own professional fleet or part-time workers who declare their availability within predefined time blocks, which mostly maintains full control over operations. It differs from crowdshipping where crowdshippers are neither employed by the company, limiting its control over their participation, nor do they commit to specific time blocks, leading to highly uncertain arrivals. Therefore, the methodologies developed in the aforementioned literature are not directly applicable to our work.

The studies cited hereafter address the assignment of crowdshippers to parcels using deterministic models, assuming that all crowdshippers and parcels are known when making decisions. Archetti et al. (2016) are the first to study the vehicle routing problem with crowdshippers. Their work has been further extended in various directions, including the integration of time windows (e.g., Macrina et al., 2020); allowing multiple deliveries per crowdshipper (e.g., Macrina et al., 2017); enabling transshipment between the professional fleet and crowdshippers (e.g., Yu et al., 2022); modeling pickup and delivery operations (e.g., Dahle et al., 2019); and incorporating transshipment between crowdshippers (e.g., Pugliese et al., 2021), among others. To handle the dynamic and stochastic arrival of crowdshippers and parcels within a deterministic framework, online approaches start with re-optimizing the deterministic problem whenever new information becomes available (e.g., Arslan et al., 2019; Dayarian & Pazour, 2022), when a certain amount of time has elapsed (e.g., Tao et al., 2023), or both (e.g., Archetti et al., 2021), adjusting the solution in real-time.

To address the dynamic and stochastic arrival of crowdshippers and parcels, the subsequent approaches incorporate stochastic information on the arrival of *either* crowdshippers *or* parcels. Behrend et al. (2021) study the multi-period item-sharing and crowdshipping problem. They employ a scenario-based planning approach considering stochastic information on the arrival of parcels while crowdshippers are announced sufficiently in advance. Yıldız (2021b) addresses the parcel routing problem with registered crowdshippers. The author applies a Monte Carlo procedure that incorporates anticipated future stochastic information on the arrival of parcels, while crowdshippers register in advance. Dahle et al. (2017) study the stochastic vehicle routing problem with in-store customers acting as crowdshippers. They formulate the problem as a two-stage stochastic problem, considering stochastic information on the arrival of crowdshippers while parcels are known in advance. Skålnes et al. (2020) develop a multi-stage version of the problem.

The approaches discussed below incorporate stochastic information on the arrival of *both* crowdshippers *and* parcels. Yıldız (2021a) solves the express parcel routing problem with crowdshippers using a dynamic programming approach under simplifying assumptions, as well as Monte-Carlo simulations to address these assumptions. Their approach is constrained by the inherent limitations in the applicability of dynamic programming. Silva et al. (2023) implement deep reinforcement learning for crowdshipping last-mile delivery. In their problem, the authors assume that all parcels originate from a single depot and must be delivered either by the company's own fleet through routing or by assigning them to crowdshippers as they become available. Dayarian and Savelsbergh (2020) introduce a dynamic and stochastic routing problem, with in-store customers as crowdshippers. The problem is solved using sample scenario planning. De Maio et al. (2024) solve their dynamic and stochastic pickup-and-delivery problem with in-store customers as crowdshippers by adapting a real-time insertion method enhanced with a cost function approximation. Mousavi et al. (2024) formulate a dynamic pickup-and-delivery problem with in-store customers as crowdshippers. The solution methodology they develop results in up to 25.2% savings in operational cost and 9.8% increase in served orders compared to a myopic approach. Their work relates to ours as it is the only one proposing an approximate dynamic programming approach using value function approximation for the assignment of crowdshippers to parcels. However, they focus on crowdshipping with in-store customers (single origin shared by crowdshippers and parcels). Furthermore, they adopt a forward-pass ADP algorithm initially developed for large-scale fleet management problems (e.g., Simão et al., 2009; Al-Kanj et al., 2020), while we develop a backward *adaptive dynamic programming algorithm* tailored to the particular characteristics of our problem (see Section 4 for further detail on the algorithm). Dayarian and Savelsbergh (2020), De Maio et al. (2024) and Mousavi et al. (2024) all consider a single location for parcel pickup and crowdshipper origin (e.g., a store). In their setting, solely the destination of crowdshippers and parcels is uncertain as the origin, the store, is known and identical for both. In addition, the risk of missed assignments is reduced due to the proximity of the store to parcel and crowdshipper destinations. In contrast, our research focuses on peer-to-peer crowdshipping, which is subject to high uncertainty due to parcels and crowdshippers originating from different, unknown, locations. This results in stochastically appearing origins and destinations for both parcels and crowdshippers.

The main contributions of our paper are as follows:

- We are the first to study the dynamic assignment problem in the context of a peer-to-peer crowdshipping platform with multiple possible parcel and crowdshipper origins, considering the uncertainty in the future availability of crowdshippers and parcels.
- We develop an approximate dynamic programming (ADP) approach using value function approximation (VFA) to overcome the challenge of decision-making over time and under uncertainty. The characteristics of the dynamic assignment problem under study require adapting the ADP algorithm to address its specific characteristics.
- By running a set of computational experiments, we demonstrate the effectiveness of our ADP approach and reveal the different parameters affecting the relevance of implementing a non-myopic assignment method.

3. Problem formulation

This section describes the crowdshipper-to-parcel dynamic assignment problem under study. We start by presenting the problem setting. Then, we formally describe the dynamic assignment problem through five components (as done by e.g., Powell, 2019; Mousavi et al., 2024): the exogenous information, the state and decision variables as well as the transition and objective functions. Table 1 summarizes the notations of the problem.

3.1. Problem setting

The peer-to-peer crowdshipping platform determines the assignment of crowdshippers to parcels over time. In this dynamic assignment setting, decisions are made sequentially in a system where crowdshipper and parcel arrivals are stochastic and their availability is dynamically revealed to the platform. The time horizon is modeled as a discrete set of decision times (henceforth also referred to as times), $t \in \mathcal{T}$. The platform operates over a geographic area discretized into a set of zones, $\mathcal{D}$. Parcels and crowdshippers may originate from and be destined to any zone, with no restrictions to specific locations such as stores or homes.

Parcels, referred to as $l \in \mathcal{L}$, are characterized by three attributes: their origin, $o_l \in \mathcal{D}$, their destination, $d_l \in \mathcal{D}$, and the first time they are available for assignment, $\theta_l \in \mathcal{T}$. Parcels have an assignment window spanning several decision times: parcels can be assigned at time θ_l or at any of the following δ times, with the latest possible assignment at time $\theta_l + \delta$. As such, assignment is possible over a window of $\delta + 1$ decision times. Distinct parcels that share the same attributes, meaning the same origin o_l, destination d_l and first time they are available for

Notation	Description
$\mathcal{L}$	Set of parcels, $l \in \mathcal{L}$
$\mathcal{R}$	Set of crowdshippers, $r \in \mathcal{R}$
$\mathcal{T}$	Set of decision times, $t \in \mathcal{T}$
$\mathcal{D}$	Set of zones, $o_r, o_l, d_r, d_l \in \mathcal{D}$
θ_l	First decision time when parcel l is available for assignment, $\theta_l \in \mathcal{T}$
θ_r	Decision time when crowdshipper r is available for assignment, $\theta_r \in \mathcal{T}$
$\delta + 1$	Length of the assignment window for parcels (in decision times)
$c^{\theta_r - \theta_l}_{o_r d_r, o_l d_l}$	Cost savings achieved when assigning a crowdshipper with origin o_r and destination d_r to a parcel with origin o_l and destination d_l, while the parcel has already been available for assignment for $\theta_r - \theta_l \in \{0, 1, ..., \delta\}$ decision times
L^0_t	Set of parcels first available for assignment at decision time t
R^0_t	Set of crowdshippers first available for assignment at decision time t
L^X_t	Set of parcels assigned at decision time t
L^A_t	Set of parcels available for assignment at decision time t, $L^A_t = L^A_{t-1} - L^X_{t-1} - L^0_{t-\delta-1} + L^0_t$
R^A_t	Set of crowdshippers available for assignment at decision time t, $R^A_t = R^0_t$

Table 1: Notations for the problem formulation.

assignment θ_l, are treated as a single parcel as they would be grouped together and delivered by the same crowdshipper. Not all parcels need to be assigned if no crowdshipper is found for whom crowdshipping is cost-effective. We assume that a parcel not assigned by the end of its assignment window is transferred to the alternative delivery system.

Crowdshippers, referred to as $r \in \mathcal{R}$, are characterized by three attributes: their origin, $o_r \in \mathcal{D}$, their destination, $d_r \in \mathcal{D}$, and the time they are available for assignment, $\theta_r \in \mathcal{T}$. Crowdshippers are, by nature, available for assignment at only one decision time. A new set of crowdshippers is therefore considered at each time. By default, we assume that crowdshippers disclose their availability no time in advance, i.e., between times $\theta_r - 1$ and θ_r. However, we also investigate the possibility of them announcing it further in advance (see Section 6). Distinct crowdshippers may share the same origin o_r, destination d_r, and be available for assignment at the same time θ_r. In our problem setting, crowdshippers cannot deliver multiple parcels with different origins and destinations. This assumption is reasonable since crowdshippers make a detour on a planned trip and may not be interested in making several stops at different pickup and delivery locations. A crowdshipper may remain unassigned and not be utilized for crowdshipping if no suitable parcel is available.

The objective of our dynamic assignment problem is to assign crowdshippers to parcels over time, maximizing the cost savings achieved via crowdshipping relative to an alternative delivery. This objective is chosen as it is well-suited for guiding assignment decisions: crowdshipping is chosen only when it offers a cost advantage over the alternative delivery. While we adopt this particular objective function, others could be considered without altering the overall methodology. Only the nature of the objective would change, reflecting different operational goals such as maximizing the number of parcels delivered or minimizing crowdshipper detours.

3.2. Exogenous information

Crowdshipper and parcel arrivals in the system are stochastic (e.g., following a Poisson process) and their availability is dynamically revealed. The crowdshipper and parcel arrival processes are independent of each other. New crowdshippers arriving in the system between times $t-1$ and t are described by the set of crowdshippers first available for assignment at time t, denoted R_t^0. New parcels arriving in the system between times $t-1$ and t are described by the set of parcels first available for assignment at time t, noted L_t^0.

3.3. State variables

The state of the system at time t is characterized by the two sets, R_t^A and L_t^A, respectively the set of crowdshippers and parcels available for assignment at time t. Crowdshippers are available for assignment at only one decision time (at time θ_r). Parcels are available for assignment from the first time they are available for assignment (θ_l) until the end of their assignment window (at time $\theta_l + \delta$), and as long as they remain unassigned.

3.4. Decision variables

At each time t, decisions involve determining the assignment of crowdshippers available for assignment (R_t^A) to parcels available for assignment (L_t^A). The decisions are represented by binary variables as follows.

$$x_{r,l}^t = \begin{cases} 1 & \text{if crowdshipper } r \text{ is assigned to parcel } l \text{ at decision time } t \\ 0 & \text{otherwise} \end{cases}$$

Variables $x_{r,l}^t$ must satisfy the following set of constraints, that define the feasible region of the problem, denoted $\mathcal{X}(R, L)$. The constraints are written using generic sets R and L, later specified at different stages of the article.

$$\sum_{r \in R} x_{r,l}^t \leq 1 \quad \forall\, l \in L \tag{1}$$

$$\sum_{l \in L} x_{r,l}^t \leq 1 \quad \forall\, r \in R \tag{2}$$

$$\theta_l \cdot x_{r,l}^t \leq \theta_r \quad \forall\, r \in R,\ l \in L \tag{3}$$

$$\theta_r \cdot x_{r,l}^t \leq \theta_l + \delta \quad \forall\, r \in R,\ l \in L \tag{4}$$

$$x_{r,l}^t \in \{0, 1\} \quad \forall\, r \in R,\ l \in L \tag{5}$$

Constraints (1) stipulate that a parcel must be assigned to at most one crowdshipper and constraints (2) that a crowdshipper cannot be assigned to more than one parcel. Constraints (3) and (4) guarantee that a crowdshipper is only assigned to a parcel if the crowdshipper is available for assignment at some time in the parcel's assignment window. Constraints (5) ensure the binary nature of the variables.

3.5. Objective function

The overall objective of our problem is to find the optimal assignment of crowdshippers to parcels over the time horizon $\mathcal{T}$ so as to maximize the total cost savings achieved via crowdshipping relative to an alternative delivery. This amounts to maximizing the sum of the cost savings over all decision times $t \in \mathcal{T}$. The cost savings over one decision time t are given in the following objective function.

$$\max \left[\sum_{r \in R_t^A} \sum_{l \in L_t^A} c_{o_r d_r, o_l d_l}^{\theta_r - \theta_l} \cdot x_{r,l}^t \right] \tag{6}$$

where $c_{o_r d_r, o_l d_l}^{\theta_r - \theta_l}$ denotes the cost savings achieved when assigning a crowdshipper with origin o_r and destination d_r to a parcel with origin o_l and destination d_l, while the parcel has already been available for assignment for $\theta_r - \theta_l$ decision times. The number of decision times for which the parcel has already been available for assignment (i.e., the parcel's delivery time) is computed as the difference between the time the crowdshipper is available for assignment (θ_r) and the first time the parcel is available for assignment (θ_l). The cost savings achieved via crowdshipping are defined as the difference between the cost of the alternative delivery and that of crowdshipped delivery (see Subsection 5.1 for a detailed formula). Crowdshipping is chosen only when it offers a cost advantage over the alternative delivery.

3.6. Transition functions

The state of the system is updated immediately before decision times according to the following transition functions. The set of parcels available for assignment at time t (L_t^A) is updated as follows.

$$L_t^A = L_{t-1}^A - L_{t-1}^X - L_{t-\delta-1}^0 + L_t^0 \tag{7}$$

At time t, the set of parcels available for assignment consists of those that remain unassigned at time $t-1$ ($L_{t-1}^A - L_{t-1}^X$), excluding parcels whose assignment window closes after time $t-1$. Those parcels arrived in the system between times $t-\delta-2$ and $t-\delta-1$ ($L_{t-\delta-1}^0$). The new parcels that arrive in the system between times $t-1$ and t are included (L_t^0).

The decisions made at time t are represented in the set L_t^X, defined as the set of parcels assigned at time t, as follows.

$$L_t^X = \{l \mid x_{r,l}^t = 1,\ l \in L_t^A,\ r \in R_t^A\} \tag{8}$$

where the variables $x_{r,l}^t$ lie within the feasible region $\mathcal{X}(R_t^A, L_t^A)$ and are chosen to solve objective function (6).

The set of crowdshippers available for assignment at time t (R_t^A) is updated as follows.

$$R_t^A = R_t^0 \tag{9}$$

A new set of crowdshippers is considered at every decision time. All crowdshippers available for assignment at time $t-1$ leave the system by time t, being assigned to a parcel or not. Figure 1 summarizes the sequential evolution of the system between decision times $t-1$ and t: new crowdshippers and parcels (R_t^0 and L_t^0) arrive in the system between times $t-1$ and t, the state of the system (L_t^A and R_t^A) is updated immediately before time t, and assignment decisions (L_t^X) are made at time t.

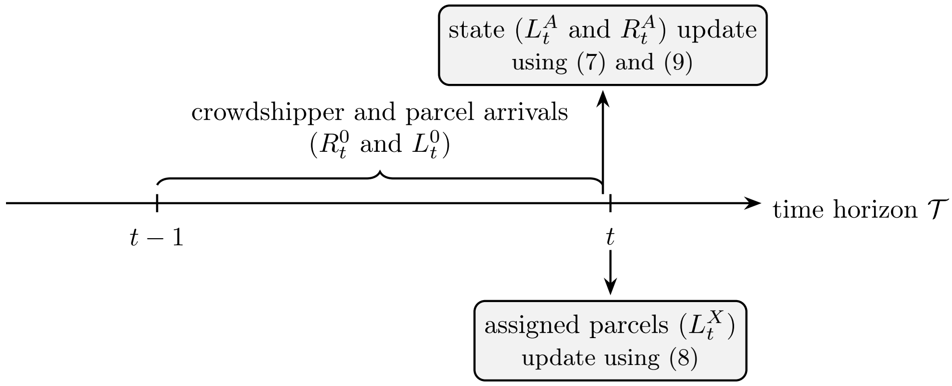

Figure 1: Sequential evolution of the system between decision times $t-1$ and t.

Notation	**Description**
$\beta_{o_l d_l}$	Number of occurrences of parcels with origin o_l and destination d_l through the simulation of parcel arrivals
α	Smoothing factor
γ	Number of decision times the algorithm steps back in time to compute parcel value estimates
$\hat{v}^n_{o_l d_l, t}$	Parcel value estimate of a parcel with origin o_l and destination d_l, that has already been available for assignment for n decision times, obtained at decision time t (with n expressed according to context)
$\overline{v}^n_{o_l d_l}$	Smoothed parcel value estimate of a parcel with origin o_l and destination d_l, that has already been available for assignment for n decision times (with n expressed according to context)
L_t^V	Set of parcels first available for assignment between decision times $t-\delta$ and t
L_t^S	Set of parcels first available for assignment between decision times $t-\delta$ and $t+\gamma$, that are not assigned before decision time t
R_t^S	Set of crowdshippers first available for assignment between decision times t and $t+\gamma$

Table 2: Notations for the solution methodology.

4. Solution methodology

We solve the dynamic assignment problem defined in Section 3 with an approximate dynamic programming (ADP) approach based on value function approximation (VFA), inspired by Spivey and Powell (2004). This approach is well-suited for problems where sequential decisions must be made over time in an uncertain system, and where decisions may affect future options and rewards. It allows efficient handling of large and uncertain state spaces (Powell, 2007). The notations used to describe the solution methodology are summarized in Table 2.

4.1. ADP approach

In this dynamic assignment setting, decisions are made sequentially while crowdshipper and parcel arrivals are stochastic and their availability is dynamically revealed. If objective function (6) were solved at each decision time t, decision-making would be myopic as it would neglect the uncertainty in the future availability of crowdshippers and parcels. In particular, assigning a crowdshipper to a parcel immediately may lead to missing a better-suited crowdshipper becoming available later. We address this by subtracting an estimate of the parcel's expected value at the next decision time, denoted $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$, from the immediate cost savings achieved when assigning the parcel at the current time $c_{o_r d_r, o_l d_l}^{\theta_r-\theta_l}$, as follows.

$$\max \left[\sum_{r \in R_t^A} \sum_{l \in L_t^A} \left(c_{o_r d_r, o_l d_l}^{\theta_r-\theta_l} - \overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1} \right) \cdot x_{r,l}^t \right] \tag{10}$$

The smoothed parcel value estimate $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$ approximates the cost savings achieved when assigning a parcel with origin o_l and destination d_l after one additional decision time, i.e., when it has already been available for assignment for $(\theta_r - \theta_l) + 1$ times, before the end of its assignment window. It represents the value of keeping the parcel available for assignment at the next time, capturing the likelihood that a more efficient crowdshipper may become available. At the last decision time of the parcel's assignment window, $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1} = 0$ since the parcel exits the system. The assignment of a parcel l will be postponed if all potential crowdshippers $r \in R_t^A$ yield lower immediate cost savings $c_{o_r d_r, o_l d_l}^{\theta_r-\theta_l}$ than the expected future ones $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$, that is, if $(c_{o_r d_r, o_l d_l}^{\theta_r-\theta_l} - \overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}) \leq 0$, or if the potential crowdshippers for whom $(c_{o_r d_r, o_l d_l}^{\theta_r-\theta_l} - \overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}) > 0$ are more suitable for other parcels. In such cases, $x_{r,l}^t = 0 \; \forall \; r \in R_t^A$, and the parcel assignment is postponed in anticipation of a better-suited crowdshipper becoming available later.

The dynamic assignment problem defined in Section 3 is therefore solved sequentially at each time t with ADP objective function (10) replacing the myopic one (6). It is solved in real time using a commercial solver, which ensures very fast computation. In contrast, computing the smoothed parcel value estimates $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$ required for objective function (10) is more time-consuming, but this computation is carried out only once in advance, offline. These estimates guide the assignment decisions and constitute the key element that differentiates our approach from a myopic one, as they allow to take the uncertainty in the future availability of crowdshippers and parcels into account. The *adaptive dynamic programming algorithm* used to compute the smoothed parcel value estimates $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$ is presented in the next subsection.

4.2. Adaptive dynamic programming algorithm

Because parcel and crowdshipper arrivals are stochastic, parcel value estimates $\hat{v}_{o_l d_l, t}^n$ must be computed under various state conditions and then smoothed to yield more robust estimates. To explore a wide range of state conditions, the *adaptive dynamic programming algorithm* simulates the system forward in time, generating parcel and crowdshipper arrivals and solving the corresponding assignment problems.

At each decision time, the *adaptive dynamic programming algorithm* performs a backward step to compute the parcel value estimates $\hat{v}_{o_l d_l, s}^n$ at retrospective time s. The computation of the parcel value estimates $\hat{v}_{o_l d_l, s}^n$ leverages not only the information available at retrospective time s, but also the future information revealed thereafter. The parcel value estimates $\hat{v}_{o_l d_l, s}^n$ therefore reflect the future availability of parcels and crowdshippers and the downstream impact of assignment decisions.

An assignment problem is solved once with the parcel included and once with it excluded; the difference between the two optimal objective values provides the parcel value estimate $\hat{v}_{o_l d_l, s}^n$.

Algorithm 1 Adaptive dynamic programming algorithm

1: \\Initialization
2: Initializing $L_t^0, R_t^0, L_t^X, L_t^A \quad \forall\, t = 0, 1, \ldots, \gamma + \delta - 1$
3: $t = \gamma + \delta$
4: \\Stopping criterion
5: **while** $|o_l, d_l \in \mathcal{D} \times \mathcal{D} : \beta_{o_l, d_l} \geq b| < \lceil a \cdot |\mathcal{D} \times \mathcal{D}| \rceil$ **do**
6: \\Simulating crowdshipper and parcel arrivals
7: $R_t^0 \leftarrow$ new crowdshippers r, with $o_r, d_r, \theta_r = t$
8: $L_t^0 \leftarrow$ new parcels l, with $o_l, d_l, \theta_l = t$
9: $\beta_{o_l d_l} \leftarrow \beta_{o_l d_l} + 1 \quad \forall\, l \in L_t^0$
10: \\Updating system state
11: $L_t^A \leftarrow L_{t-1}^A - L_{t-1}^X - L_{t-\delta-1}^0 + L_t^0$
12: $R_t^A \leftarrow R_t^0$
13: \\Solving the assignment problem
14: $x_{r,l}^t \leftarrow$ Solve (10) over feasible region $\mathcal{X}(R_t^A, L_t^A)$
15: $L_t^X \leftarrow \{l \mid x_{r,l}^t = 1,\ l \in L_t^A,\ r \in R_t^A\}$
16: \\Backward step
17: $s = t - \gamma$
18: \\State condition formulation
19: $R_s^S \leftarrow \biguplus_{s \leq i \leq t} R_i^0$
20: $L_s^S \leftarrow \biguplus_{s-\delta \leq i \leq t} L_i^0 - \biguplus_{s-\delta \leq i < s} L_i^X$
21: \\Computing parcel value estimates
22: $L_s^V \leftarrow \biguplus_{s-\delta \leq i \leq s} L_i^0$
23: $C(R_s^S, L_s^S) \leftarrow \max \sum_{r \in R_s^S} \sum_{l \in L_s^S} c_{o_r d_r, o_l d_l}^{\theta_r - \theta_l} \cdot x_{r,l}^{\theta_r}$ over feasible region $\mathcal{X}(R_s^S, L_s^S)$
24: **for** $l \in L_s^V$ **do**
25: **if** $l \in L_s^S$ **then**
26: $C(R_s^S, L_s^S - \{l\}) \leftarrow \max \sum_{r \in R_s^S} \sum_{l \in L_s^S - \{l\}} c_{o_r d_r, o_l d_l}^{\theta_r - \theta_l} \cdot x_{r,l}^{\theta_r}$ over feasible region $\mathcal{X}(R_s^S, L_s^S - \{l\})$
27: $\hat{v}_{o_l d_l, s}^{s-\theta_l} = C(R_s^S, L_s^S) - C(R_s^S, L_s^S - \{l\})$
28: **else if** $l \notin L_s^S$ **then**
29: $C(R_s^S, L_s^S + \{l\}) \leftarrow \max \sum_{r \in R_s^S} \sum_{l \in L_s^S + \{l\}} c_{o_r d_r, o_l d_l}^{\theta_r - \theta_l} \cdot x_{r,l}^{\theta_r}$ over feasible region $\mathcal{X}(R_s^S, L_s^S + \{l\})$
30: $\hat{v}_{o_l d_l, s}^{s-\theta_l} = C(R_s^S, L_s^S + \{l\}) - C(R_s^S, L_s^S)$
31: **end if**
32: \\Smoothing parcel value estimates
33: $\overline{v}_{o_l d_l}^{s-\theta_l} \leftarrow \alpha \cdot \hat{v}_{o_l d_l, s}^{s-\theta_l} + (1 - \alpha) \cdot \overline{v}_{o_l d_l}^{s-\theta_l}$
34: **end for**
35: $t \leftarrow t + 1$
36: **end while**

The value of a parcel thus corresponds to its marginal contribution to the assignment problem. Computing parcel value estimates $\hat{v}_{o_l d_l, s}^n$ in this manner ensures that they reflect not only the possibility that a better-suited crowdshipper (e.g., one with a shorter detour) may appear in the future, but also whether such a crowdshipper is actually available for the parcel and not more suitable for another parcel. This approach explicitly captures the interdependencies between parcels.

Algorithm 1 details the *adaptive dynamic programming algorithm*, whose steps are outlined below.

`Initialization (lines 1-3)` The algorithm starts by initializing the sets L_t^0, R_t^0, L_t^X and L_t^A to the empty set for times 0 to $\gamma + \delta - 1$. The computation of parcel value estimates begins at time $\gamma + \delta$, since the algorithm requires information extending up to $\gamma + \delta$ decision times into the past.

`Stopping criterion (lines 4-5)` The algorithm computes the parcel value estimates until at least b parcels with origin o_l and destination d_l have been observed, for a proportion a of all possible origin o_l destination d_l pairs.

`Simulating crowdshipper and parcel arrivals (lines 6-9)` The algorithm simulates the arrival of crowdshippers (R_t^0) and parcels (L_t^0) and updates the counter of occurrences of parcels with origin o_l and destination d_l ($\beta_{o_l d_l}$).

`Updating system state (lines 10-12)` The algorithm updates the state of the system (L_t^A and R_t^A) according to transition functions (7) and (9).

`Solving the assignment problem (lines 13-15)` Objective function (10) is solved over the feasible region $\mathcal{X}(R_t^A, L_t^A)$ using the latest smoothed parcel value estimates $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$ available. The set of parcels assigned at time t (L_t^X) is updated according to equation (8).

`Backward step (lines 16-17)` The algorithm performs a backward step of γ decision times, returning to retrospective time $s = t - \gamma$. Parcel value estimates $\hat{v}_{o_l d_l, s}^n$ are derived from the information available at retrospective time s, together with the future information revealed over the subsequent γ times up to time t. This allows parcel value estimates $\hat{v}_{o_l d_l, s}^n$ to reflect the future availability of crowdshippers and parcels as well as the downstream impact of assignment decisions. Since parcels can only be assigned over a fixed window and leave the system once that window closes, their influence on decision-making is mostly confined to those decision times. Therefore, looking further back and considering more future information than the length of the assignment window yields little added benefit. By default, we thus suppose $\gamma = \delta + 1$, though other values for γ could be used.

`State condition formulation (lines 18-20)` To compute parcel value estimates $\hat{v}_{o_l d_l, s}^n$, the sets L_s^S and R_s^S are considered. They include the parcels and crowdshippers available for assignment at retrospective time s, together with those that will enter the system up to time t. L_s^S is characterized by the parcels first available for assignment at times $s-\delta \leq i \leq t$ and not assigned before time s (`line 20`).

`Computing parcel value estimates (lines 21-31)` Parcel value estimates $\hat{v}_{o_l d_l, s}^{s-\theta_l}$ are computed for the origin o_l destination d_l pairs of parcels in L_s^V — meaning parcels whose assignment window allows assignment at retrospective time s (`line 22`) — and, for each parcel, for the number of decision times for which it has already been available for assignment at retrospective time s, namely $s - \theta_l$. Parcel value estimates $\hat{v}_{o_l d_l, s}^{s-\theta_l}$ correspond to the difference between the optimal objective values of two assignment problems, one with the parcel included and the other with it excluded. Concretely, a first assignment problem is solved using the information available at retrospective time s, together with the future information revealed up to time t (L_s^S and R_s^S, `line 23`). $C(R_s^S, L_s^S)$ represents the optimal objective value of this first assignment problem. Then, two cases may arise.

`Case 1 (line 25):` If parcel l is available for assignment at retrospective time s ($l \in L_s^S$), a second assignment problem, identical to the first but excluding parcel l, is solved (`line 26`). The corresponding optimal objective value is denoted $C(R_s^S, L_s^S - \{l\})$. The parcel value estimate $\hat{v}^{s-\theta_l}_{o_l d_l, s}$ is obtained as the difference between the optimal objective values of the assignment problem solved with and without the parcel (`line 27`).

`Case 2 (line 28):` If parcel l is not available for assignment at retrospective time s ($l \notin L^S_s$), a second assignment problem, identical to the first but including parcel l, is solved (`line 29`). The corresponding optimal objective value is denoted $C(R^S_s, L^S_s + \{l\})$. The parcel value estimate $\hat{v}^{s-\theta_l}_{o_l d_l, s}$ is obtained as the difference between the optimal objective values of the assignment problem solved with and without the parcel (`line 30`).

`Smoothing parcel value estimates (lines 32-36)` The parcel value estimate $\hat{v}^{s-\theta_l}_{o_l d_l, s}$ is smoothed with the ones obtained previously, with the smoothing factor α. The algorithm then moves forward in time.

Inspired by Spivey and Powell (2004), our methodology tailors the *adaptive dynamic programming algorithm* (Algorithm 1) to the specific characteristics of the crowdshipper-to-parcel dynamic assignment problem under study. The differences introduced by this adaptation constitute our main methodological contributions. Firstly, a key feature of our method is the explicit integration of parcel assignment windows, which restrict parcel availability to specific decision times. As a result, parcel value estimates are computed only over the span of the assignment window rather than across the remainder of the operational horizon as in most ADP approaches (e.g., Yang & Strauss, 2017; Van Heeswijk et al., 2019; Rivera & Mes, 2022). Secondly, the value of parcels is assumed to be independent of the time within the operational horizon; the decisive factors of their value are only their origin o_l, destination d_l, and the number of decision times for which they have already been available for assignment $\theta_r - \theta_l$. Parcel value estimates are thus estimated through a single simulation of parcel and crowdshipper arrivals through time, rather than requiring repeated simulations of the entire operational horizon as in most ADP methods (e.g., Powell, 2007; Powell & Topaloglu, 2014; Mes & Rivera, 2017), whose value estimations are time-sensitive.

5. Computational experiments

In this section, we present the results of our computational experiments to assess the effectiveness of the proposed ADP approach and how various parameters affect it.

5.1. Experimental setting

Before presenting the results, the experimental setting is described. Fictitious dynamic assignment problems are generated based on real-world data from two Belgian organizations. Belgium is chosen as the study area as its size (approximately 280 km $\times$ 230 km) provides a reasonable scale for peer-to-peer crowdshipping. The data is structured at the NUTS-3 level, with Belgium comprising 44 NUTS-3 regions ($|\mathcal{D}| = 44$) that serve as origin (o_l and o_r) and destination (d_l and d_r) zones for parcels and crowdshippers. Euclidean distances computed between zone centroids are employed. Assignment decisions are made at the beginning of each day (hereafter decision times are referred to as days for readability), since the available data are reported on a daily basis.

The spatial distribution of daily parcel arrivals is informed by a dataset provided by the postal service company bpost (*www.bpost.be*), giving the average number of parcels they deliver per day for all 44 $\times$ 44 pairs between two NUTS-3 regions. Among those, we only keep the pairs with a daily volume higher than 0.1% of the total daily volume over all pairs, leading to 136 effective origin o_l destination d_l pairs. The daily parcel arrivals are modeled for each pair as a Poisson process with a rate proportional to the daily volume of that pair observed in the bpost dataset, and so that the total daily volume across all pairs equals Λ. The parameter Λ represents the expected total number of parcels arriving per day over all origin o_l destination d_l pairs. It will take the following values in our experiments: $\Lambda = 5,\ 10,\ 20$ and 40 parcels/day. Three assignment window lengths for parcels are analyzed, namely 3, 5 and 7 days, meaning that parcels can be assigned on the first day or on the next $\delta = 2,\ 4$ and 6 days.

A similar reasoning is employed to generate the daily crowdshipper arrivals. The spatial distribution of daily crowdshipper arrivals is informed by a dataset provided by the public agency Federal Planning Bureau (*www.plan.be*), giving the average number of person trips per day for all 44 $\times$ 44 pairs between two NUTS-3 regions. Among those, we only keep the pairs with a daily volume higher than 0.1% of the total daily volume over all pairs, leading to 191 effective origin o_l destination d_l pairs. The daily crowdshipper arrivals are modeled for each pair as a Poisson process with a rate proportional to the daily volume of that pair observed in the Federal Planning Bureau dataset, and so that the total daily volume across all pairs equals $\rho * \Lambda$. The parameter ρ represents the ratio of the expected total number of crowdshippers relative to the expected total number of parcels arriving per day. It will take the following values in our experiments: $\rho = 0.5,\ 1.0$ and 2.0, corresponding respectively to twice fewer, the same number, and twice more crowdshippers than parcels arriving per day.

The objective of our dynamic assignment problem is to maximize the cost savings achieved via crowdshipping relative to an alternative delivery (see Subsection 3.5). The cost savings achieved when assigning a crowdshipper with origin o_r and destination d_r to a parcel with origin o_l and destination d_l, while the parcel has already been available for assignment for $\theta_r - \theta_l$ decision times, are denoted $c^{\theta_r-\theta_l}_{o_rd_r,o_ld_l}$ and are given as follows.

$$c^{\theta_r-\theta_l}_{o_rd_r,o_ld_l} = \Big(fc_A + vc_A \cdot dist_{o_ld_l}\Big) \cdot \Big(1 + \mu * \big(\delta - (\theta_r - \theta_l)\big)\Big) - \Big(fc_C + vc_C \cdot det_{o_rd_r,o_ld_l}\Big) \quad (11)$$

The term $(fc_A + vc_A \cdot dist_{o_ld_l})$ is the cost of the alternative delivery when performed on the last day of the assignment window, when the parcel has already been available for assignment for δ days. It includes a fixed component, fc_A, and a variable component, $vc_A \cdot dist_{o_ld_l}$, proportional to the delivery distance $dist_{o_ld_l}$. The term $(1 + \mu * (\delta - (\theta_r - \theta_l)))$ reflects the cost increase required by the alternative delivery to deliver the parcel earlier. $\theta_r - \theta_l$ denotes the parcel's delivery time, i.e., the number of days between the parcel's arrival in the system (on day θ_l) and its delivery on day θ_r. The cost is increased by μ for each day saved relative to the last possible delivery day, $\delta - (\theta_r - \theta_l)$. In our computational experiments, we set $\mu = 5\%$, so that the term $(1 + \mu * (\delta - (\theta_r - \theta_l)))$ is 1 on the last day of the assignment window (when $\theta_r - \theta_l = \delta$), 1.05 one day earlier, 1.10 two days earlier, etc.

The term $(fc_C + vc_C \cdot det_{o_rd_r,o_ld_l})$ is the cost of crowdshipping as paid to the crowdshipper, consisting of a fixed compensation per trip, fc_C, and a variable compensation, $vc_C \cdot det_{o_rd_r,o_ld_l}$, proportional to the detour distance $det_{o_r d_r, o_l d_l}$, as done by Mousavi et al. (2024).

To fix the values of the cost parameters in our experiments, we define $\Delta vc = \frac{vc_C}{vc_A}$ and $\Delta fc = fc_A - fc_C$. Three values are considered for Δvc, namely $\Delta vc = 2,\ 3,$ and 4, which implies that one kilometer of detour, as paid to the crowdshipper, is between two and four times more costly than one kilometer of the alternative delivery. Accordingly, we set $\Delta fc = \{0,\ 20,\ 40\} \cdot vc_A$, meaning that the crowdshipping fixed compensation is lower by an amount equivalent to the cost of 0, 20, and 40 kilometers of the alternative delivery.

We employ a full factorial experimental design, in which all combinations of the parameter values are considered, resulting in $4 \cdot 3 \cdot 3 \cdot 3 \cdot 3 = 324$ distinct parameter configurations. The algorithm is implemented in Python, the optimization models are solved using Gurobi, and both run on a 1.7 GHz computer with 32 GB of RAM. The algorithm parameters are configured as follows. The smoothing factor α is defined as $\frac{1}{\beta_{o_l d_l}}$, where $\beta_{o_l d_l}$ is the current number of occurrences of parcels with origin o_l and destination d_l in the simulation of parcel arrivals (see `line 9` of Algorithm 1). Following George and Powell (2006) and our preliminary tests, this factor ensures rapid and stable convergence by reducing the weight of new observations over time. Parcel value estimates are computed until at least $b = 10$ parcels with origin o_l and destination d_l are observed for $a = 99\%$ of all origin o_l destination d_l pairs (see `line 5` of Algorithm 1). Our experiments show that beyond this threshold, the optimal objective value over the time horizon no longer improves significantly.

For each of the 324 parameter configurations, the *adaptive dynamic programming algorithm* (Algorithm 1) is applied to compute the smoothed parcel value estimates $\overline{v}_{o_l d_l}^{(\theta_r - \theta_l)+1}$. Then, for each configuration, 100 dynamic assignment problems over a 100-day time horizon ($|\mathcal{T}| = 100$) are generated and solved using our ADP approach with the precomputed smoothed parcel value estimates. To evaluate the effectiveness of our ADP method, the same 100 dynamic assignment problems (for each parameter configuration) are also solved with a myopic approach, which serves as a benchmark. This comparison quantifies the benefit of our ADP approach, i.e., of explicitly accounting for the uncertainty in the future availability of crowdshippers and parcels. In the myopic approach, decisions rely solely on the information available at the time of assignment, namely the crowdshippers and parcels available for assignment (R_t^A and L_t^A). Each day, objective function (6), or equivalently objective function (10) with $\overline{v}_{o_l d_l}^{(\theta_r - \theta_l)+1} = 0$, is solved over the feasible region $\mathcal{X}(R_t^A, L_t^A)$. The chosen objective function, expressed as a cost comparison with an alternative delivery, prevents the myopic approach from being overly simplistic. Only assignments for which crowdshipping is less costly than the alternative delivery are considered, thereby eliminating poor decisions.

The following metrics are considered to assess and discuss the effectiveness of our ADP approach. The metrics are computed for each of the 324 parameter configurations.

`ADP cost savings increase` measures the effectiveness of our ADP method compared to the myopic approach. For each of the 100 dynamic assignment problems, we compute the relative difference between the optimal objective value over the time horizon obtained with ADP and that obtained with the myopic approach, normalized by the latter. `ADP cost savings increase` is then obtained as the average of the normalized relative differences across the 100 dynamic assignment problems.

`ADP detour reduction`, comparing our ADP approach to the myopic one, denotes the reduction in the average detour. It is computed analogously to `ADP cost savings increase`, but using the average detour per parcel assigned.

`ADP postponement increase`, comparing our ADP approach to the myopic one, denotes the increase in parcel assignment postponement. It is computed analogously to `ADP cost savings increase`, but using the average number of days a parcel assignment is postponed behind its first day of availability for assignment.

`Nb. pot. crowdshippers` represents the average number of potential crowdshippers to whom a parcel can be assigned over its assignment window, i.e., for whom crowdshipping yields positive cost savings ($c^{\theta_r-\theta_l}_{o_r d_r, o_l d_l} > 0$). This average is computed across all parcels within each dynamic assignment problem, and then averaged across the 100 dynamic assignment problems.

`avg`($\overline{v}^{(\theta_r-\theta_l)+1}_{o_l d_l}$) denotes the average smoothed parcel value estimate $\overline{v}^{(\theta_r-\theta_l)+1}_{o_l d_l}$, taken over all parcels assigned across the 100 dynamic assignment problems.

5.2. ADP approach performance

In this section, we analyze the results of our computational experiments and discuss the performance of our ADP method. Our ADP approach is able to extract value from accounting for the uncertainty in the future availability of crowdshippers and parcels in all cases, and more significantly in specific configurations. Our results show that our ADP approach significantly outperforms the myopic approach, yielding an `ADP cost savings increase` of 4.1% on average over the 324 parameter configurations, and up to 20.5%. Out of the 324 parameter configurations, 29.3% (resp. 15.4% and 7.7%) yield an `ADP cost savings increase` above 5% (resp. 7.5% and 10%). `ADP detour reduction` reaches 26.4% on average over the 324 parameter configurations, and up to 60.8%. We recall here that the computational cost of solving the dynamic assignment problem with our ADP approach is equivalent as with the myopic approach. The only added computational cost comes from the *adaptive dynamic programming algorithm* that is run beforehand and once.

Parcel assignments are, on average, postponed by 1.42 days with the ADP approach, compared to 0.98 days with the myopic one. This confirms that the ADP method, through explicitly accounting for the uncertainty in the future availability of crowdshippers and parcels, effectively delays parcel assignments in anticipation of better-suited crowdshippers becoming available later. Such postponement, however, leads to a slight decrease in the number of assigned parcels: on average, 2.1% fewer parcels are assigned under ADP compared to the myopic approach.

Table 3 highlights the main results, showing the impact of each parameter by reporting the averages of the metrics across all parameter configurations in which that parameter is fixed at a given value. The analysis of Table 3 shows that the following parameter variations lead to higher `ADP cost savings increase`: a longer assignment window (δ), a higher expected total number of parcels arriving per day (Λ), a higher crowdshipper-to-parcel arrival ratio (ρ), a larger fixed cost of the alternative delivery ($\Delta fc = fc_A - fc_C$), and a lower crowdshipper detour compensation ($\Delta vc = \frac{vc_C}{vc_A}$). The increase in `ADP cost savings increase` is, in most cases, associated with an increase in `ADP detour reduction` (except for Δvc).

Parameter	*Value*	`ADP cost savings increase`	`ADP detour reduction`	`ADP postponement increase`	`Nb. pot. crowd-shippers`	`avg`($\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$)
δ	2	3.2%	21.1%	58.3%	1.3	9.1
	4	4.1%	26.5%	60.3%	2.3	12.9
	6	5.0%	31.6%	63.6%	3.3	15.6
Λ	5	2.7%	17.8%	25.1%	0.7	7.3
	10	3.8%	23.3%	41.7%	1.3	10.2
	20	4.7%	29.1%	67.8%	2.5	13.8
	40	5.2%	35.3%	108.4%	4.7	18.9
ρ	0.5	1.9%	16.1%	34.8%	1.0	5.8
	1	4.0%	25.7%	55.8%	2.0	11.5
	2	6.5%	37.4%	91.6%	3.9	20.2
Δfc	0	1.8%	19.7%	37.9%	1.3	6.4
	20	4.1%	27.4%	57.1%	2.1	12.3
	40	6.5%	32.1%	87.3%	3.4	18.8
Δvc	2	5.7%	25.4%	83.7%	3.2	12.7
	3	3.8%	30.0%	54.5%	2.1	12.5
	4	2.8%	23.7%	44.0%	1.5	12.4

Table 3: `ADP cost savings increase`, `ADP detour reduction`, `ADP postponement increase`, `Nb. pot. crowdshippers` and `avg`($\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$) with respect to the length of the assignment window (δ), the expected total number of parcels arriving per day (Λ), the crowdshipper-to-parcel arrival ratio (ρ), the difference in fixed (Δfc) and variable (Δvc) costs between the alternative delivery and crowdshipping.

To justify the effect of those parameter variations, a common underlying explanation emerges. For those five variations, the number of potential crowdshippers to whom a parcel can be assigned increases (`Nb. pot. crowdshippers`, see Table 3). `Nb. pot. crowdshippers` increases with a longer delivery window (δ), since parcels are available for assignment for more decision times. It also increases with a higher expected total number of parcels arriving per day (Λ), as this raises network density and, consequently, the number of crowdshipper-to-parcel pairs for which $c_{o_r d_r, o_l d_l}^{\theta_r-\theta_l} > 0$. A larger crowdshipper-to-parcel arrival ratio (ρ) increases the number of crowdshippers relative to parcels arriving per day, thereby increasing `Nb. pot. crowdshippers`. Finally, `Nb. pot. crowdshippers` also increases with a higher fixed cost of the alternative delivery (Δfc) and a lower crowdshipper detour compensation (Δvc) as both make longer detours cost-effective (i.e., still yielding $c_{o_r d_r, o_l d_l}^{\theta_r-\theta_l} > 0$). Having more potential crowdshippers per parcel (`Nb. pot. crowdshippers`) means that there are higher chances of finding better-suited crowdshippers in the future, and thus more potential benefit and less risk of postponing parcel assignments (higher `ADP cost savings increase`). As smoothed parcel value estimates $\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$ capture the expected cost savings when postponing parcel assign-

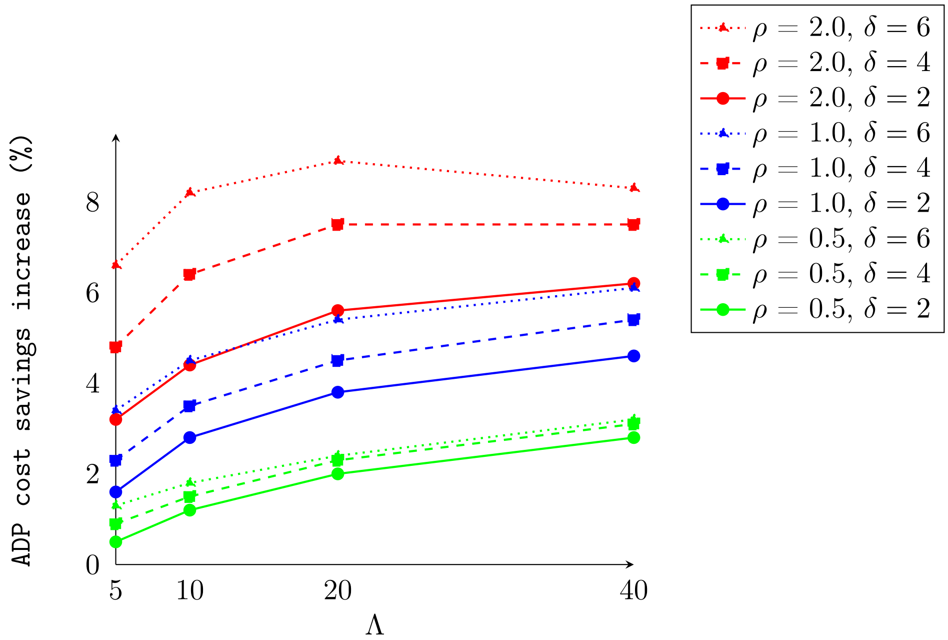

Figure 2: Impact of the expected total number of parcels arriving per day (Λ), the crowdshipper-to-parcel arrival ratio (ρ) and the length of the assignment window (δ) on `ADP cost savings increase`.

ments, the higher availability of crowdshippers per parcel is reflected in higher smoothed parcel value estimates ($\texttt{avg}(\overline{v}_{o_l d_l}^{(\theta_r-\theta_l)+1}$), see Table 3). Higher smoothed parcel value estimates increase the frequency of postponement of parcel assignments (`ADP postponement increase`, see Table 3), discouraging premature or inefficient parcel assignments and providing the opportunity to achieve better crowdshipper-to-parcel assignments. Compared to the myopic approach — that does not seek to postpone parcel assignments — the difference in performance intensifies, and `ADP cost savings increase` grows.

Figure 2 is provided to complement Table 3 by jointly analyzing the impact of the expected total number of parcels arriving per day (Λ), the crowdshipper-to-parcel arrival ratio (ρ), and the assignment window length (δ). Figure 2 also shows an upward trend in `ADP cost savings increase` as the expected total number of parcels arriving per day (Λ), the crowdshipper-to-parcel arrival ratio (ρ), and the assignment window length (δ) increase. Furthermore, we observe that `ADP cost savings increase` rises more steeply with less parcels arriving per day (Λ). Even more strikingly, for the larger problems ($\rho = 2$, $\delta = 6$), `ADP cost savings increase` declines when the expected total number of parcels arriving per day (Λ) grows from 20 to 40. When possible assignment options are numerous — `Nb. pot. crowdshippers` reaches around 10 or more, e.g., 11.6 on average when $\Lambda = 40$, $\rho = 2$, $\delta = 6$ — the myopic approach manages to find good crowdshipper-to-parcel assignments right away, thereby reducing the advantage of the ADP approach.

These results help identify the parameters that justify the use of a non-myopic assignment method for the dynamic assignment problem in peer-to-peer crowdshipping. The application of our ADP approach is particularly warranted when there is sufficient, but not excessive, availability of potential crowdshippers (`Nb. pot. crowdshippers`) — i.e., when the length of the assignment window (δ), the crowdshipper-to-parcel arrival ratio (ρ), and the fixed cost of the alternative delivery (Δfc) are high, when the crowdshipper detour compensation (Δvc) is low, and when the expected total number of parcels arriving per day (Λ) is moderate. Averaged over the more favorable parameter configurations (with $\delta \geq 4,\ \Lambda \geq 20,\ \rho \geq 1,\ \Delta fc \geq 20,\ \Delta vc \leq 3$), `ADP cost savings increase` and `ADP detour reduction` reach 9.8% and 46.4% respectively.

6. Impact of advance information

In the problem setting considered in the core of this work, it is supposed that crowdshippers declare their availability between times $\theta_r - 1$ and θ_r, and are available for assignment at time θ_r, i.e., they declare their availability no time in advance. However, in other cases, platforms could allow crowdshippers to declare their availability further in advance. In this section, we explore these cases. The new parameter ϵ represents the number of decision times by which crowdshipper availability is provided ahead of time. Crowdshippers still declare their availability between times $\theta_r - 1$ and θ_r, but are available for assignment at time $\theta_r + \epsilon$. Consequently, at each time t, information on crowdshippers is known for the next ϵ times. Crowdshippers whose availability is known but who are not available for assignment yet are included in the assignment problems solved at each time t; however, assignments involving them remain provisional and are reoptimized at subsequent times until the time they are available for assignment (at time $\theta_r + \epsilon$).

Subsection 6.1 reviews the changes that advance information on the availability of crowdshippers brings to the dynamic assignment problem, to the ADP approach, and to the *adaptive dynamic programming algorithm*. Subsection 6.2 discusses the impact on the effectiveness of our ADP methodology.

6.1. Methodological changes

In this section, we detail the methodological changes in the case where crowdshippers declare their availability ϵ decision times in advance. For simplicity of exposition, we suppose that $\epsilon \leq \delta$ (even if it is not a necessary assumption). We start by focusing on the problem formulation. In the following, we only give the components that are modified as compared to Section 3, in their new form, and keeping the same order as in Section 3.

New crowdshippers arriving in the system between times $t-1$ and t are described by the set R_t^0, the set of crowdshippers declaring their availability between times $t-1$ and t (instead of crowdshippers first available for assignment, see Subsection 3.2).

The set R_t^A is the set of crowdshippers considered in the assignment problem at time t (instead of crowdshippers available for assignment, see Subsection 3.3). Crowdshippers are included in the assignment problem from their arrival in the system until the time they become available for assignment (at time $\theta_r + \epsilon$).

The feasible region $\mathcal{X}(R, L)$ accounts for the fact that crowdshippers are available for assignment at time $\theta_r + \epsilon$ (instead of time θ_r, see Subsection 3.4). Constraints (1), (2), and (5) remain unchanged, while constraints (3) and (4) are given as follows.

$$\theta_l \cdot x_{r,l}^t \leq (\theta_r + \epsilon) \quad \forall\ r \in R,\ l \in L \tag{3'}$$

$$(\theta_r + \epsilon) \cdot x_{r,l}^t \leq \theta_l + \delta \quad \forall\ r \in R,\ l \in L \tag{4'}$$

The number of decision times a parcel has already been available for assignment (i.e., the parcel's delivery time) is computed as the difference between the time the crowdshipper is available for assignment ($\theta_r + \epsilon$ instead of θ_r, see Subsection 3.5) and the first time the parcel is available for assignment (θ_l). The objective function over one decision time (6) is therefore given in the following equation (the modification also applies to equation (11) in Subsection 5.1).

$$\max \left[\sum_{r \in R_t^A} \sum_{l \in L_t^A} c_{o_r d_r, o_l d_l}^{(\theta_r+\epsilon)-\theta_l} \cdot x_{r,l}^t \right] \tag{6'}$$

Only the assignments involving crowdshippers who are available for assignment at time t — that declared their availability between times $t-\epsilon-1$ and $t-\epsilon$ ($R_{t-\epsilon}^0$ instead of R_t^A, see equation (8) in Subsection 3.6) — are confirmed. The set of parcels assigned at time t, L_t^X, thus writes as follows.

$$L_t^X = \{l \mid x_{r,l}^t = 1,\ l \in L_t^A,\ r \in R_{t-\epsilon}^0\} \tag{8'}$$

In addition to the new crowdshippers that declare their availability between times $t-1$ and t (R_t^0, see equation (9) in Subsection 3.6), the assignment problem at time t also includes the crowdshippers who were already present at time $t-1$ (R_{t-1}^A), except for those who became available for assignment at that time, i.e., those who declared their availability between times $t-\epsilon-2$ and $t-\epsilon-1$ ($R_{t-\epsilon-1}^0$). The set of crowdshippers considered in the assignment problem at time t, R_t^A, thus writes as follows.

$$R_t^A = R_{t-1}^A - R_{t-\epsilon-1}^0 + R_t^0 \tag{9'}$$

Now that the modifications to the problem formulation are established, we turn our attention to the ADP approach (Subsection 4.1). The modified ADP objective function aims to decide whether to assign currently known crowdshippers to parcels or to postpone parcel assignments in anticipation of better-suited crowdshippers whose availability is still unknown, and will be declared at time $(t+\epsilon)+1$ or later. It writes as follows.

$$\max \left[\sum_{r \in R_t^A} \sum_{l \in L_t^A} \left(c_{o_r d_r, o_l d_l}^{(\theta_r+\epsilon)-\theta_l} - \overline{v}_{o_l d_l}^{(t+\epsilon)-\theta_l+1} \right) \cdot x_{r,l}^t \right] \tag{10'}$$

Subsequently, we look at the components that were changed in the *adaptive dynamic programming algorithm* (Algorithm 1, Subsection 4.2), specifying the lines to be adjusted.

`line 12` The set of crowdshippers considered in the assignment problem at time t (R_t^A) is computed as in equation (9').

`line 14` Objective function (10') is solved over feasible region $\mathcal{X}(R_t^A, L_t^A)$ with constraints (3') and (4').

`line 15` The set of parcels assigned at time t (L_t^X) is computed as in equation (8').

`line 19` The set R^S_s is defined as the crowdshippers whose availability is known at retrospective time s, i.e., those that declared their availability between times $s-\epsilon$ and s, and those that will arrive in the system until time t, meaning those that declare their availability between times s and t. The set R^S_s is computed as follows.

$$R^S_s \leftarrow \biguplus_{s-\epsilon \leq i \leq t} R^0_i$$

`lines 23, 26, 29` The optimal objective values $C(R, L)$ are computed as follows, where the sets R and L are replaced by the corresponding ones in each line.

$$C(R, L) \leftarrow \max \sum_{r\in R} \sum_{l\in L} c^{(\theta_r+\epsilon)-\theta_l}_{o_r d_r,\, o_l d_l}\, x^{\theta_r+\epsilon}_{r,l}$$

$$\text{over feasible region } \mathcal{X}(R, L) \text{ with constraints (3') and (4')}$$

6.2. ADP approach performance

Additional experiments are conducted to assess the impact of advance information regarding the availability of crowdshippers on the effectiveness of our ADP methodology in comparison to the myopic approach. The following baseline configuration is defined : $\delta = 4,\ \Lambda = 20,\ \rho = 1,\ \Delta fc = 20$ and $\Delta vc = 3$. From this baseline, each parameter is varied individually to isolate its effect (no factorial experimental design). Specifically, we vary the assignment window length ($\delta = 2,\ 4,\ 6$), the expected total number of parcels arriving per day ($\Lambda = 5,\ 10,\ 20,\ 40$), the crowdshipper-to-parcel arrival ratio ($\rho = 0.5,\ 1,\ 2$), and the difference in fixed ($\Delta fc = 0,\ 20,\ 40$) and variable ($\Delta vc = 2,\ 3,\ 4$) costs between the alternative delivery and crowdshipping. Parameter ϵ is varied from 0 to δ in all parameter configurations. We do not consider values of ϵ beyond δ, since in that case full knowledge of all the crowdshippers over the parcel's assignment window is available. This experimental design results in 60 different parameter configurations.

Figures 3 and 4 show how the advance information on the availability of crowdshippers (ϵ) affect `ADP cost savings increase`. The earlier the crowdshippers declare their availability in advance, the more `ADP cost savings increase` decreases, until reaching 0% when crowdshipper availability is known for the entire assignment window ($\epsilon = \delta$). Declaring availability one day in advance already leads to a substantial reduction in `ADP cost savings increase`, with an average decrease of 51% across all parameter configurations. ADP's advantage stems from accounting for the uncertainty in the future availability of crowdshippers and parcels. When information on future crowdshipper availability becomes available, it reduces the uncertainty and the ADP method progressively loses its advantage over the myopic approach until both methods make identical decisions and perform alike.

7. Conclusion

By promising to exploit excess capacity in existing personal transport for the delivery of freight, crowdshipping has garnered attention in recent years as a potentially cost-effective and sustainable alternative to traditional delivery (Pourrahmani & Jaller, 2021). However, the successful implementation of crowdshipping heavily depends on its operational efficiency, with the

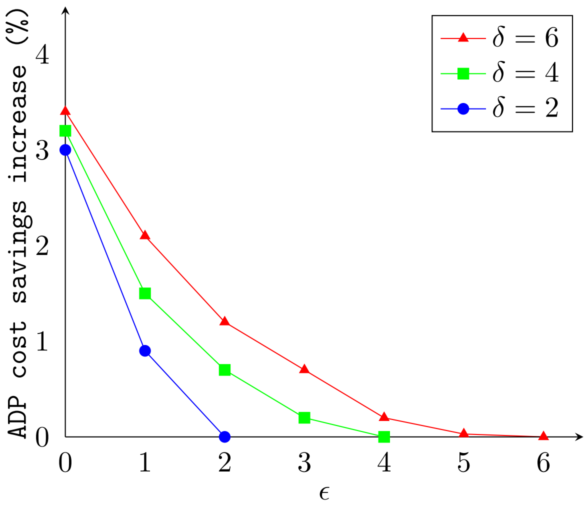

Figure 3: Impact of the advance information on the availability of crowdshippers (ϵ) and the assignment window length (δ) on `ADP cost savings increase`, with baseline configuration ($\delta = 4,\ \Lambda = 20,\ \rho = 1,\ \Delta fc = 20,\ \Delta vc = 3$).

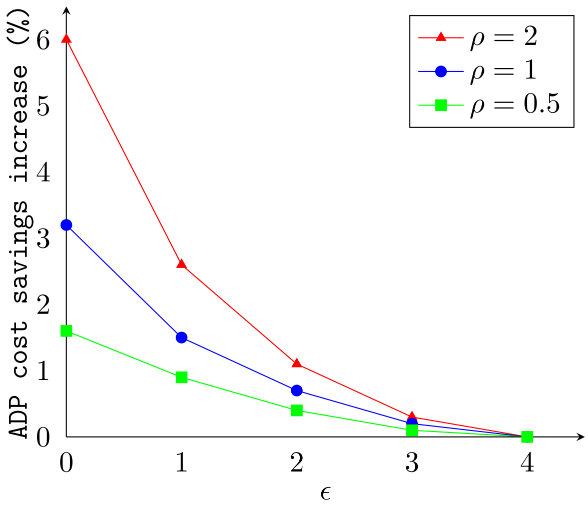

Figure 4: Impact of the advance information on the availability of crowdshippers (ϵ) and the crowdshipper-to-parcel arrival ratio (ρ) on `ADP cost savings increase`, with baseline configuration ($\delta = 4,\ \Lambda = 20,\ \rho = 1,\ \Delta fc = 20,\ \Delta vc = 3$).

assignment of crowdshippers to parcels at its core (Le et al., 2019). Crowdshippers need to be assigned to parcels over time, yet crowdshipping operations face uncertainty. This paper focuses on solving the crowdshipper-to-parcel dynamic assignment problem for a peer-to-peer crowdshipping platform, with crowdshippers and parcels arriving stochastically and their availability dynamically revealed. We propose an approximate dynamic programming (ADP) approach based on value function approximation (VFA) to assign crowdshippers to parcels over time while considering the uncertainty in the future availability of crowdshippers and parcels. Our method estimates whether parcels will be more valuable in the future, in which case it may be advantageous to postpone their assignment rather than assigning them immediately.

Our computational experiments show that our ADP methodology outperforms the myopic approach, increasing the cost savings achieved via crowdshipping by 4.1% on average and up to 20.5% and reducing the average detour per parcel by 26.4% on average and up to 60.8% compared to the myopic approach. These experiments highlight the parameters that justify the use of a non-myopic assignment method for the dynamic assignment problem in peer-to-peer crowdshipping: when the length of the assignment window, the crowdshipper-to-parcel arrival ratio and the fixed cost of the alternative delivery are high, the crowdshipper detour compensation is low, the expected total number of parcels arriving per day is moderate and crowdshippers declare their availability no time in advance. The application of our ADP approach is particularly warranted when there is sufficient, but not excessive, availability of potential crowdshippers. Averaged over the more favorable parameter configurations (with $\delta \geq 4,\ \Lambda \geq 20,\ \rho \geq 1,\ \Delta fc \geq 20,\ \Delta vc \leq 3$), the increase in cost savings compared to the myopic approach reaches 9.8%. We also adapt our ADP approach to investigate the possibility of crowdshippers announcing their availability in advance. The earlier the crowdshippers declare their availability in advance, the more the increase in cost savings compared to the myopic approach decreases, with a decrease of 51% on average when declaring their availability one decision time in advance.

This research can be extended in several directions to circumvent some limitations and explore other research challenges. First, the crowdshipper-to-parcel assignment method developed in this paper can be used to evaluate the impacts of crowdshipping, especially economic and environmental. Detours need to be minimized in order to reduce the delivery costs and the greenhouse gas emissions (Paloheimo et al., 2016), which our methodology helps to do. Second, in this work, a parcel is delivered to its final destination by a single crowdshipper. Future research could relax this assumption by enabling transshipment, which allows parcels to be handed off between crowdshippers.

References

Al-Kanj, L., Nascimento, J., & Powell, W. B. (2020). Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. *European Journal of Operational Research*, *284*(3), 1088–1106. https://doi.org/10.1016/j.ejor.2020.01.033

Archetti, C., Guerriero, F., & Macrina, G. (2021). The online vehicle routing problem with occasional drivers. *Computers & Operations Research*, *127*, 105144. https://doi.org/10.1016/j.cor.2020.105144

Archetti, C., Savelsbergh, M., & Speranza, M. G. (2016). The Vehicle Routing Problem with Occasional Drivers. *European Journal of Operational Research*, *254*(2), 472–480. https://doi.org/10.1016/j.ejor.2016.03.049

Arslan, A. M., Agatz, N., Kroon, L., & Zuidwijk, R. (2019). Crowdsourced Delivery – A Dynamic Pickup and Delivery Problem with Ad Hoc Drivers. *Transportation Science*, *53*(1), 222–235. https://doi.org/10.1287/trsc.2017.0803

Behrend, M., & Meisel, F. (2018). The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? *Transportation Research Part B: Methodological*, *111*, 227–243. https://doi.org/10.1016/j.trb.2018.02.017

Behrend, M., Meisel, F., Fagerholt, K., & Andersson, H. (2021). A multi-period analysis of the integrated item-sharing and crowdshipping problem. *European Journal of Operational Research*, *292*(2), 483–499. https://doi.org/10.1016/j.ejor.2020.10.034

Buldeo Rai, H., Verlinde, S., & Macharis, C. (2018). Shipping outside the box. Environmental impact and stakeholder analysis of a crowd logistics platform in Belgium. *Journal of Cleaner Production*, *202*, 806–816. https://doi.org/10.1016/j.jclepro.2018.08.210

Dahle, L., Andersson, H., & Christiansen, M. (2017). The Vehicle Routing Problem with Dynamic Occasional Drivers. In T. Bektaş, S. Coniglio, A. Martinez-Sykora, & S. Voß (Eds.), *Computational Logistics* (pp. 49–63). Springer. https://doi.org/10.1007/978-3-319-68496-3_4

Dahle, L., Andersson, H., Christiansen, M., & Speranza, M. G. (2019). The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research*, *109*, 122–133. https://doi.org/10.1016/j.cor.2019.04.023

Dayarian, I., & Pazour, J. (2022). Crowdsourced order-fulfillment policies using in-store customers. *Production and Operations Management*, *31*(11), 4075–4094. https://doi.org/10.1111/poms.13805

Dayarian, I., & Savelsbergh, M. (2020). Crowdshipping and Same-day Delivery: Employing In-store Customers to Deliver Online Orders. *Production and Operations Management*, *29*(9), 2153–2174. https://doi.org/10.1111/poms.13219

De Maio, A., Ohlmann, J. W., Stoia, S., & Vocaturo, F. (2024). Analysis of in-store crowdshipping in a stochastic dynamic pickup-and-delivery system. *Central European Journal of Operations Research*, *33*(3), 1149–1170. https://doi.org/10.1007/s10100-024-00939-8

George, A. P., & Powell, W. B. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, *65*(1), 167–198. https://doi.org/10.1007/s10994-006-8365-9

Le, T. V., Stathopoulos, A., Van Woensel, T., & Ukkusuri, S. V. (2019). Supply, demand, operations, and management of crowd-shipping services: A review and empirical evidence. *Transportation Research Part C: Emerging Technologies*, *103*, 83–103. https://doi.org/10.1016/j.trc.2019.03.023

Macrina, G., Pugliese, L., Guerriero, F., & Laganà, D. (2017). The Vehicle Routing Problem with Occasional Drivers and Time Windows. In A. Sforza & C. Sterle (Eds.), *Optimization and Decision Science: Methodologies and Applications* (pp. 577–587). Springer. https://doi.org/10.1007/978-3-319-67308-0_58

Macrina, G., Pugliese, L., Guerriero, F., & Laporte, G. (2020). Crowd-shipping with time windows and transshipment nodes. *Computers & Operations Research*, *113*, 104806. https://doi.org/10.1016/j.cor.2019.104806

Mes, M. R. K., & Rivera, A. E. P. (2017). Approximate Dynamic Programming by Practical Examples. In R. Boucherie & N. van Dijk (Eds.), *Markov Decision Processes in Practice* (pp. 63–101). Springer. https://doi.org/10.1007/978-3-319-47766-4_3

Mohri, S. S., Ghaderi, H., Nassir, N., & Thompson, R. G. (2023). Crowdshipping for sustainable urban logistics: A systematic review of the literature. *Transportation Research Part E: Logistics and Transportation Review*, *178*, 103289. https://doi.org/10.1016/j.tre.2023.103289

Mousavi, K., Bodur, M., Cevik, M., & Roorda, M. J. (2024). Approximate dynamic programming for pickup and delivery problem with crowd-shipping. *Transportation Research Part B: Methodological*, *187*, 103027. https://doi.org/10.1016/j.trb.2024.103027

Paloheimo, H., Lettenmeier, M., & Waris, H. (2016). Transport reduction by crowdsourced deliveries – a library case in Finland. *Journal of Cleaner Production*, *132*, 240–251. https://doi.org/10.1016/j.jclepro.2015.04.103

Pourrahmani, E., & Jaller, M. (2021). Crowdshipping in last mile deliveries: Operational challenges and research opportunities. *Socio-Economic Planning Sciences*, *78*, 101063. https://doi.org/10.1016/j.seps.2021.101063

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons. https://doi.org/10.1002/9780470182963

Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of Operational Research*, *275*(3), 795–821. https://doi.org/10.1016/j.ejor.2018.07.014

Powell, W. B., & Topaloglu, H. (2014). Approximate dynamic programming for large-scale resource allocation problems. In *Informs tutorials in operations research* (pp. 123–147). INFORMS. https://doi.org/10.1287/educ.1063.0027

Pugliese, L., Guerriero, F., Macrina, G., & Scalzo, E. (2021). Crowd-Shipping and Occasional Depots in the Last Mile Delivery. In R. Cerulli, M. Dell'Amico, F. Guerriero, D. Pacciarelli, & A. Sforza (Eds.), *Optimization and Decision Science: ODS, Virtual Conference, November 19, 2020* (pp. 213–225). Springer. https://doi.org/10.1007/978-3-030-86841-3_18

Rivera, A. E. P., & Mes, M. R. K. (2022). Anticipatory scheduling of synchromodal transport using approximate dynamic programming. *Annals of Operations Research*, *350*(1), 95–129. https://doi.org/10.1007/s10479-022-04668-6

Savelsbergh, M., & Ulmer, M. W. (2024). Challenges and opportunities in crowdsourced delivery planning and operations – an update. *Annals of Operations Research*, *343*(2), 639–661. https://doi.org/10.1007/s10479-024-06249-1

Savelsbergh, M., & Van Woensel, T. (2016). 50th Anniversary Invited Article – City Logistics: Challenges and Opportunities. *Transportation Science*, *50*(2), 579–590. https://doi.org/10.1287/trsc.2016.0675

Silva, M., Pedroso, J. P., & Viana, A. (2023). Deep reinforcement learning for stochastic last-mile delivery with crowdshipping. *EURO Journal on Transportation and Logistics*, *12*, 100105. https://doi.org/10.1016/j.ejtl.2023.100105

Simão, H. P., Day, J., George, A. P., Gifford, T., Nienow, J., & Powell, W. B. (2009). An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application. *Transportation Science*, *43*(2), 178–197. https://doi.org/10.1287/trsc.1080.0238

Skålnes, J., Dahle, L., Andersson, H., Christiansen, M., & Hvattum, L. M. (2020). The Multistage Stochastic Vehicle Routing Problem with Dynamic Occasional Drivers. In E. Lalla-Ruiz, M. Mes, & S. Voß (Eds.), *Computational Logistics* (pp. 261–276). Springer. https://doi.org/10.1007/978-3-030-59747-4_17

Spivey, M. Z., & Powell, W. B. (2004). The Dynamic Assignment Problem. *Transportation Science*, *38*(4), 399–419. https://doi.org/10.1287/trsc.1030.0073

Statista. (2024a). Global e-commerce share of retail sales 2027. Retrieved August 14, 2024, from https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/

Statista. (2024b). Value of the global sharing economy 2023. Retrieved August 14, 2024, from https://www.statista.com/statistics/830986/value-of-the-global-sharing-economy/

Tao, Y., Zhuo, H., & Lai, X. (2023). The pickup and delivery problem with multiple depots and dynamic occasional drivers in crowdshipping delivery. *Computers & Industrial Engineering*, *182*, 109440. https://doi.org/10.1016/j.cie.2023.109440

Van Heeswijk, W. J. A., Mes, M. R. K., & Schutten, J. M. J. (2019). The Delivery Dispatching Problem with Time Windows for Urban Consolidation Centers. *Transportation Science*, *53*(1), 203–221. https://doi.org/10.1287/trsc.2017.0773

Yang, X., & Strauss, A. K. (2017). An approximate dynamic programming approach to attended home delivery management. *European Journal of Operational Research*, *263*(3), 935–945. https://doi.org/10.1016/j.ejor.2017.06.034

Yıldız, B. (2021a). Express package routing problem with occasional couriers. *Transportation Research Part C: Emerging Technologies*, *123*, 102994. https://doi.org/10.1016/j.trc.2021.102994

Yıldız, B. (2021b). Package routing problem with registered couriers and stochastic demand. *Transportation Research Part E: Logistics and Transportation Review*, *147*, 102248. https://doi.org/10.1016/j.tre.2021.102248

Yu, V. F., Jodiawan, P., & Redi, A. A. N. P. (2022). Crowd-shipping problem with time windows, transshipment nodes, and delivery options. *Transportation Research Part E: Logistics and Transportation Review*, *157*, 102545. https://doi.org/10.1016/j.tre.2021.102545