Transportation Research Part E 176 (2023) 103207

Contents lists available at ScienceDirect

Transportation Research Part E

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

A continuum approximation approach to the depot location problem in a crowd-shipping system

Patrick Stokkink, Nikolas Geroliminis *

École Polytechnique Fédérale de Lausanne (EPFL), Urban Transport Systems Laboratory (LUTS), Switzerland

ARTICLE INFO

Keywords:
Crowd-shipping
Depot location problem
Network design
Last-mile delivery

ABSTRACT

Last-mile delivery in the logistics chain contributes to congestion in urban networks due to frequent stops. Crowd-shipping is a sustainable and low-cost alternative to traditional delivery but relies heavily on the availability of occasional couriers. In this work, we propose a crowd-shipping system that uses depots to improve accessibility for potential crowd-shippers to serve a large portion of the demand for small parcels. While small-scale versions of this problem have been recently addressed, scaling to larger instances significantly complexifies the problem. A heuristic approach based on continuum approximation is designed to evaluate the quality of a potential set of depots. By combining an efficient and accurate approximation method with a large neighborhood search heuristic, we can efficiently find a good set of depots, even for large-scale networks. The proposed methodology allows for heterogeneity among crowd-shippers and allows identifying the expected number of delivered parcels in every region, which can be used to enhance lower-level assignment decisions.

A case study on the Washington DC network shows that depots are built at geographically central locations but most importantly at locations around popular origins for crowd-shippers. The optimal number of depots is mainly dependent on the marginal number of parcels that can be served by crowd-shippers from a specific depot, relative to the costs involved in opening that depot. The operational costs approximated by our continuum approximation approach deviate on average 2% from the actual operational costs using dynamic assignment strategies. For small instances, our algorithm finds better solutions than solving a discrete formulation using CPLEX, while being almost 200 times faster. For large instances, the discrete formulation cannot be constructed by CPLEX, whereas the CA-based approach finds good solutions within minutes. Finally, the results show that using CA-based strategies in all three layers of decision-making can improve overall performance by 15% compared to non-predictive strategies.

1. Introduction

The growing demand for e-commerce has led to a substantial increase in the challenges faced in traditional delivery. Nowadays, the *sharing economy* allows for rapidly connecting supply and demand, which can be used to overcome these challenges. Such a system where last-mile delivery is outsourced to a large number of individuals is referred to as *crowd-shipping*. In a crowd-shipping system, individual couriers perform deliveries on their pre-existing route, possibly with a small detour, and thereby contribute to the last-mile delivery of small parcels.

* Corresponding author.
E-mail addresses: patrick.stokkink@epfl.ch (P. Stokkink), nikolas.geroliminis@epfl.ch (N. Geroliminis).

https://doi.org/10.1016/j.tre.2023.103207
Received 3 October 2022; Received in revised form 8 May 2023; Accepted 8 June 2023
Available online 30 June 2023
1366-5545/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Crowd-shipping has numerous advantages for customers, retailers, and society as a whole. Customers, who can be either on the supply or demand side, are offered a fast, flexible, and often low-cost alternative to traditional delivery. Retailers also benefit in terms of cost, as crowd-shippers (in the literature also referred to as occasional couriers) are generally cheaper than standard drivers. Thereby, crowd-shippers are more flexible compared to standard drivers routes which are planned in advance, and can therefore more easily incorporate last-minute changes. Nevertheless, such a system requires a critical mass of crowd-shippers to be available and a large number of parcels, so as to create efficient matches without long detours. Society benefits mostly from the reduced environmental impacts as well as reduced traffic congestion. Traditional delivery is a large source of congestion because of the presence of large delivery vans in urban networks and the roadblocks they cause. According to Generation IM et al. (2020), the vast majority of the emissions in the whole logistics chain are generated by last-mile delivery. The World Economic Forum expects the number of delivery vehicles in cities to increase by 36% until 2030, leading to an increase in delivery emissions of 32% and an increase in congestion of 21%. On top of that, polluting delivery vehicles may not be allowed to enter low-emission or zero-emission zones. All these aspects make it absolutely necessary to look for green, less-polluting, alternatives, such as crowd-shipping.

The success of a crowd-shipping system heavily relies on the availability of crowd-shippers and the potential to match them to demand requests without large detours. The potential pool of crowd-shippers is considered to have a planned personal trip and an associated individual trajectory which is not necessarily near to a parcel trajectory. In this work, we focus on the few-to-many delivery problems of small parcels that are transportable by foot or by bike. If the pickup locations of parcels are poorly accessible by potential crowd-shippers, few parcels can be delivered by crowd-shippers. This can form a major problem for crowd-shipping systems that use in-store pickups. For this reason, we focus on the problem of determining optimal depot locations that function as origins of parcels. Depots are built at central locations in the network such that they are well accessible by crowd-shippers and such that a large portion of the parcel requests can be delivered.

In a depot-based crowd-shipping system, various decisions have to be made to construct a profitable system. These decisions can be divided into strategic, tactical, and operational decisions. A schematic representation of the decision process is illustrated in Fig. 1. In the first stage, the locations of the depots, where parcels are stored for crowd-shippers to pick them up, are determined. This is a strategic decision that has to be made before the system is operational and is therefore made without full knowledge of the trajectories of potential crowd-shippers and parcels. In the second stage, the assignment of parcels to these depots is determined. This is a tactical decision that is made under full knowledge of the set of parcel requests but only expectations of the trajectories of potential crowd-shippers, fed by historical data. Then, crowd-shippers announce themselves, usually in a dynamic fashion, and the parcels are assigned to them in the third (final) stage. These operational decisions are made daily based on full knowledge of parcels and either full or partial knowledge of crowd-shippers' itineraries. One of the complexities of the considered problem is the uncertainty in the requests for parcels and the availability of crowd-shippers. As many companies offer next-day delivery, demand for parcels is generally only known a day in advance. Crowd-shippers may announce their availability only upon departure from their origin.

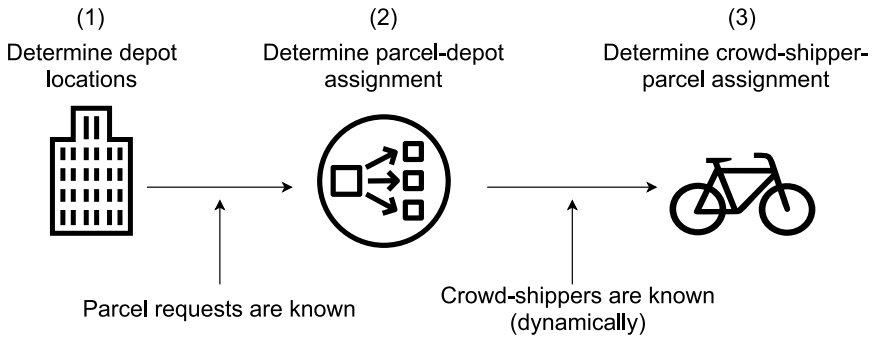


Fig. 1. Schematic representation of decision process.

In this paper, we develop a framework to determine the best depot locations for a crowd-shipping system in a large urban area. This problem is especially difficult because of the dependency on lower-level decisions and costs on upper-level decisions. To track these interactions, we solve the lower-level assignment problem of parcels to potential crowd-shippers through a Continuum Approximation (CA) approach, allowing us to determine the lower-level costs efficiently in a short time. These estimates are based on the physical properties of the matching procedure, as well as expectations of the set of parcels and the set of crowd-shippers, fed by historical data. We develop a large neighborhood search heuristic that exploits the CA estimates to efficiently search a good set of depots that minimizes the operational costs. In addition to this, these estimates are used to design a smart dynamic assignment strategy of parcels to crowd-shippers that outperforms existing strategies. A comparison of our approach to solving a discrete formulation of the problem shows that on small networks the objective obtained by our CA-based approach is slightly better than that of the discrete formulation. In terms of computation time, our CA-based approach is almost 200 times faster. On large networks, the discrete formulation fails to find a feasible solution within a reasonable amount of time, whereas our CA approach can find good depot locations within minutes.

The remainder of this paper is organized as follows. We provide a review of the relevant literature in Section 2, where we highlight the unique features of this problem. The proposed methods to efficiently solve this multi-level problem, including the continuum approximation of the assignment problem, the depot-location algorithm, and a smart dynamic assignment strategy, are

discussed in Section 3. In Section 4 we describe the discrete event simulator and various assignment strategies for the second and third-stage assignment problems. The results are discussed in Section 5, where we evaluate the proposed methodology with comparisons to static and dynamic benchmarks. We compare the performance of our CA-based algorithm against solving a discrete formulation using CPLEX and perform various sensitivity analyses. Performance is evaluated using a discrete event simulator based on a part of the city of Washington DC. The paper is concluded in Section 6.

2. Literature review

The last-mile delivery of parcels is a well-studied topic in the optimization literature. Traditionally, goods are delivered by using delivery vans. In this case, the problem can be formulated as a Pickup-and-Delivery Problem (PDP) (Savelsbergh and Sol, 1995) or a Vehicle Routing Problem (VRP) (Toth and Vigo, 2002). These problems have been extended to include various problem-specific aspects such as time-windows (Dumas et al., 1991; Ropke and Cordeau, 2009) or uncertainty (Fabri and Recht, 2006).

Due to the increase in online shopping, a large number of traditional vans is needed to serve all demand. According to Iwan et al. (2016), delivery vans for last-mile delivery are one of the main causes of congestion in urban areas. As a consequence, many companies are looking for more sustainable options to replace the aforementioned traditional delivery methods. Iwan et al. (2016) analyze the use of parcel lockers where customers can pick up and send small parcels. Results from a pilot survey in Poland indicate that the use of these lockers can potentially reduce the environmental impact of last-mile delivery. Another alternative is drone delivery (Agatz et al., 2018; Karak and Abdelghany, 2019), for which it has been shown that combining truck delivery with a drone can significantly reduce transportation costs. Thereby, drones cause less congestion compared to delivery vans. Akeb et al. (2018) propose a model that relies on the interaction of a network of neighbors to enhance parcel delivery in urban areas.

Another promising alternative to traditional last-mile delivery methods is crowd-shipping. In a crowd-shipping system, the last-mile delivery of small parcels is (partially) outsourced to individual commuters that can deliver the parcel on their pre-existing route. Various empirical studies have investigated the potential and determinants of crowd-shipping (Ermagun and Stathopoulos, 2018; Le and Ukkusuri, 2019; Punel et al., 2019). These studies have shown the potential demand for crowd-shipping and the concerns of potential users. Thereby, they highlight the importance of the availability of occasional couriers. Potential crowd-shippers have a pre-existing itinerary (origin, destination, and approximate departure and arrival times) and trip purpose (for example, a work commute or a leisure trip). Therefore, only parcels that do not create significant inconvenience can be assigned to the crowd-shipper that has to be compensated for the inconvenience through some (monetary) incentive.

Recently, substantial research has been done on the operational problems that arise in crowd-shipping systems. For a review of recent academic research as well as recent practice, the reader is referred to Le et al. (2019). Pourrahmani and Jaller (2021) give an overview of the operational challenges and research opportunities that exist in this field. One of these operational challenges is the matching of parcels to crowd-shippers which has been studied by, among others, Li et al. (2014) and Soto Setzke et al. (2017). We also note the similarities with matching problems in ride-sharing (Masoud and Jayakrishnan, 2017), ride-hailing (Yan et al., 2020), and carpooling (Wang et al., 2018). Another important operational problem is pickup and delivery routing. Clearly, these problems are intertwined and therefore are often tackled jointly. Archetti et al. (2016) model the static routing problem as a Vehicle Routing Problem with Occasional Drivers (VRPOD). It is assumed that an occasional driver is willing to make a delivery if the extra distance traveled to make the delivery is less than a pre-specified portion of the total distance traveled. Li et al. (2014) consider a crowd-shipping scenario where people and parcels share a taxi. They model the problem as a Share-a-Ride Problem (SARP), which is an extension of the Dial-a-Ride Problem (DARP). Dahle et al. (2017) consider a two-stage stochastic program to model the VRP with dynamic occasional drivers. The first-stage decision models the route of the traditional delivery truck. After the occasional drivers make themselves known in the second stage, they are assigned to parcels and the truck route can be changed. Arslan et al. (2019) model the problem as a dynamic PDP. Their heuristic assigns crowd-shipping tasks to occasional (ad-hoc) drivers dynamically. Cohn et al. (2007) integrate matching and routing decisions for carriers of small packages. Yildiz and Savelsbergh (2019) introduce the service and capacity planning problem. With their model, they aim to answer questions that arise in a crowd-shipping system, both on strategic and operational levels.

As the availability of potential crowd-shippers is a key determinant of the performance of a crowd-shipping system, parcels may be stored at intermediate depot locations (or transshipment points) such that they are easily reachable by potential suppliers. Wang et al. (2016) consider "pop-stations" distributed around the city where crowd-shippers can perform pickups. For a fixed set of transshipment points, they optimize the utilization of crowd-shippers for last-mile delivery. Raviv and Tenzer (2018) and Macrina et al. (2020) consider a crowd-shipping system where crowd-shippers can pick up parcels either from the depot or from transshipment points. Their results show the economic benefits of such transshipment nodes. Similarly, Yıldız (2021) also considers transshipment points but uses a dynamic programming algorithm to solve their problem. Contrary to the fixed transshipment points in the previous works, Mousavi et al. (2021) consider mobile depots. They do not consider the routing of vehicles, but they determine the optimal location of these mobile depots under uncertainty in supply.

Contrary to the majority of the literature that has studied operational problems, in this paper we consider the strategic planning problem of network design. Specifically, we focus on finding the optimal depot locations. This is closely related to the Facility Location Problem (FLP) (Cornuéjols et al., 1983), where optimal locations of facilities are chosen in a network. This approach has been commonly used to determine the location of depots in freight transportation problems (Fernandes et al., 2014; Gendron et al., 2016). We also note the similarities with hub location problems in passenger transportation. There, hubs function as locations where passengers can switch between mobility modes in a multi-modal shared mobility system (Blad et al., 2022) or between public transport modes (Yatskiv and Budilovich, 2017). A depot for freight transportation is fundamentally different from a hub for

passenger transportation. Whereas freight can be kept at a depot for a long time before a pickup, passengers are sensitive to time and desire rapid transfers. A comprehensive review of various solution algorithms for different variants of the hub location problem is presented by Wandelt et al. (2022).

Our research question is most similar to Mousavi et al. (2021) and Nieto-Isaza et al. (2021) who consider exact formulations for strategic location problems in crowd-shipping systems. However, due to their exact approach, their methods are limited to small-scale instances. Given the large potential of crowd-shipping in large urban areas, we focus on an algorithmic approach that can be applied to large-scale instances. Nieto-Isaza et al. (2021) consider fixed capacities of crowd-shippers on arcs. We allow crowd-shippers to make small detours, thereby enlarging the flexibility and performance, which forms an additional complexity. Thereby, we model the capacity of crowd-shippers, allowing them to carry multiple parcels. In addition to this, Mousavi et al. (2021) and Nieto-Isaza et al. (2021) consider a two-stage model where all crowd-shippers are known at the same instance. We extend this to a more realistic dynamic arrival of crowd-shippers, equivalent to a multi-stage model.

We use a CA approach to approximate with some level of accuracy the cost of the lower-level assignment problems and consecutively use an efficient heuristic to determine the depot locations. The reader is referred to Ansari et al. (2018) for a recent review of the advancements of CA models for logistics and transportation systems. CA approaches have been widely used for the design of large-scale networks. For example, for the design of vehicle routing problems (Daganzo, 2005; Ouyang, 2007), integrated package distribution systems (Smilowitz and Daganzo, 2007) and pickup-and-delivery problems (Lei and Ouyang, 2018). Thereby, CA has been used for various variants of the FLP, such as the reliable FLP (Li and Ouyang, 2010; Cui et al., 2010) and the competitive FLP (Wang and Ouyang, 2013). Although we note the similarities between the FLP and the depot location problem discussed in this paper, we note that the main difference is that our problem introduces an additional layer of complexity due to the underlying assignment problem of parcels to crowd-shippers. This complexity does not allow us to evaluate all the lower-level costs for every potential set of depot locations, as is commonly done a priori in an FLP.

Building on the approximated lower-level costs, we use a large neighborhood search heuristic to solve the depot location problem that minimizes the total cost of the crowd-shipping system. We use performance metrics to efficiently search the neighborhoods of solutions. Due to the fast CA approximation, we can find the optimal depot locations in a reasonable time. In addition to this, the CA estimates are used as input to a smart dynamic assignment strategy, that outperforms existing dynamic strategies by leveraging the expectations of future crowd-shippers. A discrete event simulator is used to evaluate the performance of the continuum approximation, the depot-location algorithm, and the dynamic assignment strategy.

3. Methodology

In this section, we describe the methodological contributions of this paper. In Section 3.1 we give a more detailed description of the problem and formulate the problem as an integer stochastic programming problem. In Section 3.2 we describe the continuum approximation approach used to approximate the cost in the third-stage assignment problem. In Section 3.3 we explain the methods used to identify the depot locations based on the CA results. A notational glossary of the sets, parameters, and variables is provided in Table 2 in Appendix. For the sake of readability, a division has been made for parameters that are introduced in the problem description, continuum approximation approach, and large neighborhood search algorithm.

3.1. Discrete formulation

The problem as described in Fig. 1 can be formulated as an integer stochastic programming problem. In this way, we can incorporate the two types of uncertainty by dividing the problem into three levels. The formulation is similar to that of Mousavi et al. (2021). The main difference is that we consider an additional layer of uncertainty (uncertainty in demand) which makes our problem a three-stage stochastic programming problem, compared to the two-stage stochastic programming problem proposed by Mousavi et al. (2021). In addition to this, crowd-shippers may carry multiple parcels in our formulation, as opposed to them carrying only one parcel in the formulation by Mousavi et al. (2021).

We consider a set of demand requests for small parcels $p \in P$, which is a realization drawn from random variable ξ_P. Similarly, we consider a set of crowd-shippers $c \in C$, which is a realization drawn from random variable ξ_C. Thereby, we consider the set of potential depots D^p and binary decision variables z_d for all $d \in D^p$ equal to 1 if depot d is opened and 0 otherwise. The set of opened depots is hereafter referred to as D and we continue to use D^p for the set of potential depots, that are not necessarily opened. A depot can be opened at a cost ϕ^{depot} and a maximum of $D^{\max}$ depots can be opened. The value of $D^{\max}$ is a modeling choice related to the maximum capital investment a crowd-shipping operator is willing to make. The fixed cost of opening a depot mainly consists of the daily rental costs of a location and the maintenance costs of parcel lockers. Acquiring the parcel lockers is a one-time investment and is therefore neglected. The costs involved with the second and third stages depend on earlier decisions, as well as the realization of parcels and crowd-shippers, and are denoted by $\psi_2(z, P)$ and $\psi_3(z, y, P, C)$ respectively. The first-stage objective is to minimize the sum of the costs of opening depots and the expected costs of the second and third stages. We denote with $\mathbb{E}_\xi[\cdot]$ the expected value function over random variable ξ. The first stage can be formulated as follows:

$$\text{minimize} \quad \phi^{\text{depot}} \sum_{d \in D^p} z_d + \mathbb{E}_{\xi_P}[\psi_2(z, P)], \tag{1}$$

$$s.t. \sum_{d \in D^p} z_d \leq D^{\max}, \tag{2}$$

$$z_d \in \mathbb{B} \qquad \forall d \in D^p. \tag{3}$$

The objective (1) is to minimize the total costs consisting of fixed costs for every opened depot and the operational costs. The operational costs are an expected value of the second- (and indirectly third-) stage costs. Constraint (2) enforces that at most $D^{\max}$ depots can be opened.

In the second stage, we decide which parcel to assign to which depot. For this, we introduce binary decision variables y_{pd} which is equal to 1 if parcel $p \in P$ is assigned to depot $d \in D^p$, and 0 otherwise. We assume there are no costs involved with the second stage (at least, there is no direct cost difference between assigning to different depots) other than the expected costs of the third stage. We consider small and portable parcels, such that depot capacity can be disregarded. This leads to the following formulation of the second stage:

$$\psi_2(z, P) = \mathbb{E}_{\xi_C}[\psi_3(z, y, P, C),] \tag{4}$$

$$s.t. \sum_{d \in D^p} y_{pd} \leq 1 \qquad \forall p \in P, \tag{5}$$

$$y_{pd} \leq z_d \qquad \forall p \in P, d \in D^p, \tag{6}$$

$$y_{pd} \in \mathbb{B} \qquad \forall p \in P, d \in D^p. \tag{7}$$

The second stage costs only consist of the expected costs of the third stage. Every parcel is assigned to at most one depot, which is enforced by Constraints (5). Constraints (6) ensure that a parcel is only assigned to an opened depot.

Finally, in the third stage, the parcels are assigned to crowd-shippers. For this, we introduce binary decision variables x_{pcd}, which is equal to 1 if parcel $p \in P$ is assigned to crowd-shipper $c \in C$ and depot $d \in D^p$, and 0 otherwise. The crowd-shipper fee for every parcel is denoted by ϕ^{cs}_{pd} and depends on the distance between the origin depot d and the destination of parcel p. Specifically, $\phi^{\text{cs}}_{pd} = \phi^{\text{cs},1}_p + \phi^{\text{cs},2}_p \cdot t_{d,dest(p)}$ where the first term is a fixed compensation per delivery and the second term is a variable compensation depending on the distance between the origin and destination of the parcel. The cost of regular delivery of parcel $p \in P$ is defined as ϕ^{reg}_p and comprises all costs associated with last-mile delivery such as fuel cost, driver salary, and cost of maintenance and repair. It is clear that for a parcel p, crowd-shipping is only favored over regular delivery if $\phi^{\text{cs}}_{pd} < \phi^{\text{reg}}_p$. A parcel can be assigned to only one crowd-shipper and a crowd-shipper can be assigned at most q_c parcels. Although a crowd-shipper can carry multiple parcels, we assume these parcels need to have identical origins and destinations, to limit the inconvenience of pickup and delivery that a crowd-shipper encounters. Thereby, a parcel can only be assigned to a crowd-shipper if they can feasibly pick up and deliver this parcel, given the depot the parcel was assigned to in the second stage. We use a binary parameter f_{pcd} which is equal to 1 if crowd-shipper $c \in C$ can feasibly pickup parcel $p \in P$ from depot $d \in D^p$ and deliver it to the final destination of the parcel. A potential crowd-shipper $c \in C$ is assumed to have a maximum detour τ_c he/she is willing to make to pick up and deliver a parcel, which defines this feasibility parameter. The third stage can then be formulated as:

$$\psi_3(z, y, P, C) = \sum_{p \in P} \phi^{\text{reg}}_p + \sum_{p \in P} \sum_{c \in C} \sum_{d \in D^p} x_{pcd}(\phi^{\text{cs}}_{pd} - \phi^{\text{reg}}_p), \tag{8}$$

$$\sum_{p \in P} \sum_{d \in D^p} x_{pcd} \leq q_c \qquad \forall c \in C, \tag{9}$$

$$\sum_{c \in C} \sum_{d \in D^p} x_{pcd} \leq 1 \qquad \forall p \in P, \tag{10}$$

$$x_{pcd} \leq y_{pd} f_{pcd} \qquad \forall c \in C, p \in P, d \in D^p, \tag{11}$$

$$x_{p_1 c d_1} + x_{p_2 c d_2} \leq 1 \qquad \forall c \in C, p_1, p_2 \in P, d_1, d_2 \in D^p : (d_1 \neq d_2 \parallel dest(p_1) \neq dest(p_2)), \tag{12}$$

$$x_{pcd} \in \mathbb{B} \qquad \forall c \in C, p \in P, d \in D^p. \tag{13}$$

The operational costs of the third stage in Eq. (8) are made up of penalties for regular delivery and compensations awarded to crowd-shippers. Without crowd-shipping, all parcels would be delivered by regular delivery vehicles, therefore incurring penalties $\sum_{p \in P} \phi^{\text{reg}}_p$. Every parcel that is delivered by crowd-shippers then costs ϕ^{cs}_{pd} for compensation, but reduces the penalties by ϕ^{reg}_p. Every shipper can be assigned at most q_c parcels and every parcel can be assigned to at most one crowd-shipper, which is enforced by Constraints (9) and (10) respectively. Thereby, through Constraints (11), we ensure that a parcel should be picked up from the depot to which it was assigned in the second stage and that the match is feasible. Constraints (12) enforce that only parcels with identical origins and destinations are allowed to be carried by the same crowd-shipper.

The difficulties of solving this problem in (1)–(13) are three-fold. First, we are dealing with two separate layers of uncertainty. The first layer of uncertainty is in the parcels; with next-day delivery being extremely common, the number of parcel requests in every region is uncertain up to a day before delivery. The second layer of uncertainty is in the crowd-shippers. The number of crowd-shippers and their itineraries are generally uncertain up to shortly before the departure of the crowd-shipper. Whereas some crowd-shippers may know their schedule well in advance, others may only make themselves available a few minutes before departure. Thereby, exact schedules may be prone to last-minute changes. The second difficulty is that decisions in each of the three stages are heavily intertwined. Decisions in the first and second stages are made based on expected costs and actions in the third stage, whereas the optimal third-stage decisions and corresponding costs depend on the decisions that were made in the first

and second stages. This illustrates the importance of solving the problem as a whole and the inability to decompose it. The third difficulty is that the large size of the problem in urban areas causes a computational burden.

These difficulties combined make it impossible for the problem to be solved exactly for a realistic case study. Therefore, in this paper, we approximate the second and third-stage operational costs using a continuum approximation approach. Based on the approximated operational costs, we optimize the first-stage strategic decisions. By using an approximation of the second and third stages, we can evaluate many potential depot combinations within a reasonable amount of time and therefore explore a large search space.

3.2. Continuum approximation

To approximate the third-stage costs, as well as estimate the served parcels in every region of the network, we use a CA approach. Rather than using a formulation based on individual crowd-shippers and parcels like in Section 3.1, we reformulate the second and third stages to a region-based formulation. We consider a network split into R regions. Daily demand for small parcels in every region $r \in R$ is equal to μ_r. To allow for heterogeneity among crowd-shippers, potential crowd-shippers are divided into classes. We denote $\tilde{C}$ as the set of crowd-shipper classes, not to be confused with the discrete set of crowd-shippers C. Every class $c \in \tilde{C}$ corresponds to a homogeneous group of crowd-shippers with origin $or(c)$, destination $dest(c)$, and a maximum detour τ_c. We note that additional heterogeneity may be added, such as maximum distance traveled or value of time, but this is omitted in this work. The number of crowd-shippers in class c is denoted by λ_c. For the sake of approximation, a crowd-shipper with capacity q_c is counted as q_c separate crowd-shippers in λ_c.

We define parameter e_{rcd} which is equal to 1 if a crowd-shipper of class c can pick up a parcel at depot d and deliver it to the final destination in region r, and 0 otherwise. When $e_{rcd} = 1$, this is referred to as a feasible assignment. For all feasible assignments for which crowd-shipping is more expensive than regular delivery ($\phi^{\text{cs}}_{rd} > \phi^{\text{reg}}_{r}$) we set $e_{rcd} = 0$. We remark the relation between the parameter e_{rcd} and the parameter f_{pcd} in the discrete formulation. One of the main advantages of this reformulation is that the computation of the variables e_{rcd} relies only on the size of the network (that is, the number of regions) and the level of heterogeneity and not on the number of crowd-shippers nor the number of parcels. Thereby, it is independent of the realizations of parcels and crowd-shippers, whereas f_{pcd} depends on the realizations of the uncertain sets P and C.

Geometrically, the matching problem and the definition of e_{rcd} can be interpreted through two ellipses, as depicted in Fig. 2. The original route from the origin to the destination of a crowd-shipper is depicted by the black line. The crowd-shipper has to make a detour by performing a pickup at d and delivery at the final destination of the parcel r. For this to be feasible within the maximum detour τ_c, the following should hold. The depot location d should lie within the ellipse with focus points $or(c)$ and $dest(c)$, where the distance between the focus and the closest vertex is equal to τ_1. Thereby, the final destination r should lie within a second ellipse with focus points d and $dest(c)$, where the distance between the focus and the closest vertex is equal to τ_2. If $\tau_1 + \tau_2 \leq \tau_c$, the pickup and delivery can be made within the maximum detour. The difficulty of this problem is marked by the variability in the values of τ_1 and τ_2 under the constraint $\tau_1 + \tau_2 \leq \tau_c$. In addition to this, there is a dependency between the size of the two ellipses. As the point d has to lie within the black ellipse, it restricts the size of the blue ellipse. We also note that we cannot omit the second ellipse by simply considering the line segment between d and $dest(c)$ inside the black ellipse instead of just point d, because of the influence of the direction of this segment on the total detour. Due to these difficulties, we resort to numerical approximations.

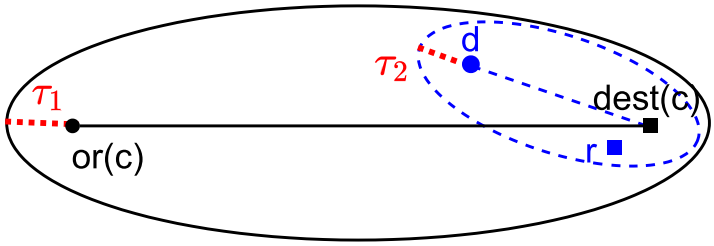


Fig. 2. Geometric interpretation of matching problem.

3.2.1. LP approximation

We use a region-based approximation of the discrete formulation in Section 3.1. We consider region-based cost parameters ϕ^{reg}_{r} for regular delivery to region $r \in R$ and ϕ^{cs}_{rd} for crowd-shipped delivery to region $r \in R$ from depot $d \in D$. We define decision variables x_{rcd} as the number of crowd-shippers of class $c \in \tilde{C}$ that perform a delivery from depot $d \in D$ to region $r \in R$. The approximation can then be formulated using the following Linear Programming (LP) problem:

$$\text{maximize} \sum_{c \in \tilde{C}} \sum_{r \in R} \sum_{d \in D} (\phi^{\text{reg}}_{r} - \phi^{\text{cs}}_{rd}) x_{rcd} \tag{14}$$

$$\sum_{c \in \tilde{C}} \sum_{d \in D} x_{rcd} \leq \hat{\mu}_r \qquad \forall r \in R \tag{15}$$

$$\sum_{r \in R} \sum_{d \in D} x_{rcd} \leq \hat{\lambda}_c \qquad \forall c \in \tilde{C} \tag{16}$$

$$x_{rcd} \in \mathbb{R} \cap [0, e_{rcd} \min(\hat{\mu}_r, \hat{\lambda}_c)] \qquad \forall c \in \tilde{C}, r \in R, d \in D \tag{17}$$

The objective is to minimize the total operational costs. Here, we simplified the objective function in (14) to only account for the delivered parcels. All parcels that are not delivered by crowd-shippers incur a cost for regular delivery. Constraints (15) and (16) ensure that not more deliveries are made than there are parcels requested and not more crowd-shippers are used than there are available. Constraints (17) define the range of the decision variables and ensure that only feasible deliveries are made.

Although this drastically simplifies the second and third layer of the discrete problem in Section 3.1 computation times and memory consumption for constructing and solving the LP problem can still be problematic for large networks. Thereby, this formulation ignores the level of uncertainty between hub assignment (second-stage decisions) and crowd-shipper assignment (third-stage decisions). This generally leads to an underestimation of the costs in a more realistic dynamic setting with uncertainty. Therefore, we next propose an algorithmic approximation that encompasses these two aspects.

3.2.2. Algorithmic approximation

For the sake of approximation, we treat the second and third-stage decisions in reverse order. First, we merge all depots and approximate the third-stage assignment. Second, based on the third-stage approximations we approximate the second-stage parcel-depot assignments. The second-stage approximations are needed because of their influence on the costs ϕ^{cs}_{rd}.

When multiple depots are opened, the main difficulty is that both parcels and crowd-shippers have to be split over the various depots. Only when the expected number of parcels delivered by a specific depot is independent of the other depots, this problem can be separated into subproblems. However, especially when two potential depots are close together, it is clear that this independency is not true. When multiple depots are opened, each parcel can originate from multiple depots and each crowd-shipper can pick up parcels from multiple depots as long as they are within their maximum detour τ_c. We first relax the second stage assignment of parcels to depots. We define $\tilde{e}_{rc}$ which is equal to 1 if a crowd-shipper with origin at $or(c)$ and destination at $dest(c)$ can pick up a parcel *from at least one open depot* and deliver it to the final destination in region r, and 0 otherwise. Specifically:

$$\tilde{e}_{rc} = \min(1, \sum_{d \in D} e_{rcd}) \qquad \forall c \in \tilde{C}, r \in R. \tag{18}$$

We want to approximate the number of parcels delivered from a given set of depots to all regions separately. To approximate this, we let $\hat{\mu}_r$ be the expected number of parcels with a destination in region r and let $\hat{\lambda}_c$ the expected number of crowd-shippers in class $c \in \tilde{C}$. Then, we define $\bar{\mu}_c = \sum_{r \in R} \tilde{e}_{rc} \hat{\mu}_r$ as the total number of parcels that can potentially be served by crowd-shippers of class c. For the sake of the approximation, we assume that a crowd-shipper is equally likely to choose any of the parcels they can feasibly deliver. Following from this, the probability that he picks a parcel with destination region r is equal to $\frac{\hat{\mu}_r}{\bar{\mu}_c}$ if $\tilde{e}_{rc} = 1$ and 0 otherwise. We can then consider all potential crowd-shippers to obtain the following estimation of the number of served parcels:

$$u_r = \sum_{c \in \tilde{C}} \tilde{e}_{rc} \hat{\lambda}_c \frac{\hat{\mu}_r}{\bar{\mu}_c} \qquad \forall r \in R. \tag{19}$$

Here, we sum over all classes of potential crowd-shippers. A class is only considered if the assignment is feasible. Every crowd-shipper in $\hat{\lambda}_c$ is considered to make the delivery with the same probability $\frac{\hat{\mu}_r}{\bar{\mu}_c}$. If no crowd-shippers can be assigned to a region r, that is either $\hat{\lambda}_c = 0$ or $\tilde{e}_{rc} = 0$ for every origin–destination pair, then the number of delivered parcels will always be zero. Similarly, if the expected number of parcels $\hat{\mu}_r$ is zero, u_r will also be zero. We also note that it is possible for $\bar{\mu}_c = 0$, in which case Eq. (19) is undefined. In this case, $\frac{\hat{\mu}_r}{\bar{\mu}_c}$ is naturally set to 0.

If $\hat{\mu}_r$ is non-zero, crowd-shippers with different origin–destination pairs may be assigned to the same parcel-destination region r. As this could lead to an overestimation of the number of served parcels in that region ($u_r > \hat{\mu}_r$), we take into account that at most $\hat{\mu}_r$ parcels can be delivered to a region r. Therefore, we define v_r which is the minimum of those two.

$$v_r = \min(\hat{\mu}_r, u_r) \qquad \forall r \in R. \tag{20}$$

Especially if the number of crowd-shippers is high, by overestimating u_r in region r (i.e., $u_r > \hat{\mu}_r$), it is likely that $u_{r'}$ for another region $r' \neq r$ will be underestimated. Therefore, we use an iterative process to ensure that this unavoidable overestimation is accounted for in the other regions. We consider the leftover number of parcels $l_r = \max(0, u_r - \hat{\mu}_r)$ and split it evenly over the potential crowd-shippers. Similar to the assignment of parcels to crowd-shippers, we assume that every crowd-shipper that can be feasibly assigned to a region r (that is, every crowd-shipper of class c for which $\tilde{e}_{rc} = 1$), is equally likely to be assigned to one of the leftover parcels in l_r. Therefore, the l_r leftover parcels are split over the origin–destination pairs proportional to the number of crowd-shippers that could be feasibly assigned to region r. We define the leftover crowd-shippers as follows:

$$\hat{\lambda}'_c = \sum_{r \in R} l_r \frac{\tilde{e}_{rc} \hat{\lambda}_c}{\sum_{c \in \tilde{C}} \tilde{e}_{rc} \hat{\lambda}_c} \qquad \forall c \in \tilde{C}. \tag{21}$$

Thereby, we define the undelivered parcels $\hat{\mu}'_r = \hat{\mu}_r - v_r$. All parcels that are already expected to be served by previously assigned crowd-shippers no longer need to be considered and are therefore disregarded. We then compute u_r according to Eq. (19), but now using $\hat{\lambda}'_c$ and $\hat{\mu}'_r$ as inputs in stead of $\hat{\lambda}_c$ and $\hat{\mu}_r$. Using these we find an additional portion of parcels that can be delivered and we update the estimated number of delivered parcels and the leftover parcels as follows:

$$v_r = \min(\hat{\mu}_r, v_r + u_r) \qquad \forall r \in R. \tag{22}$$

$$l_r = \max(0, u_r - \hat{\mu}'_r) \qquad \forall r \in R. \tag{23}$$

This iterative process can be repeated until the number of leftover parcels l_r is zero for all regions $r \in R$. Intuitively, if $l_r = 0$, all previously overestimated delivered parcels have been compensated for. However, we emphasize that the simplifying assumptions of proportional assignment in Eqs. (19) and (21) can lead to suboptimal assignments.

As the costs ϕ_p^{cs} depend on the distance between the depot and the final destination of the parcel p, we make an approximation of how many parcels are at which depot through the previously approximated total deliveries. That is, we approximate the second-stage decisions that we previously relaxed to approximate the operational costs. Let $v_r(D)$ be the number of parcels delivered to region r if depots D are opened. We recall ϕ_{rd}^{cs} as the compensation of crowd-shippers making a delivery between depot d and destination region r. Thereby, we define a_{rd} the number of parcels with final destination $r \in R$ that are stored at depot $d \in D$ as follows:

$$a_{rd} = \hat{\mu}_r \frac{v_r(\{d\})}{\sum_{d \in D} v_r(\{d\})}. \tag{24}$$

This approximation assumes that a parcel is not necessarily assigned to the closest depot, but may be assigned to a further depot. In Section 5.6 we show that such an assignment is substantially better than a closest-depot assignment. The reason for this is that the flow of crowd-shippers is often not homogeneous across the network. In that case, storing a parcel at a depot further away may increase the likelihood of it being delivered by crowd-shippers if the depot and the destination of the parcel (in that order) are on a route that is a more common itinerary for potential crowd-shippers.

The total cost can be obtained directly from the results of the approximation. The total approximated cost, similar to that defined in Section 3.1, is as follows. The first term comprises the fixed costs of constructing and maintaining depots. The second term approximates the crowd-shipper compensations by using the total number of delivered parcels derived by the iterative procedure and using the parcel-depot assignment from Eq. (24). The third term approximates the costs of regular delivery. As the costs of regular delivery are assumed to be equal for all parcels we do not distinguish between depots.

$$C(D) = \phi^{\text{depot}}|D| + \sum_{r \in R}\left[\sum_{d \in D} \phi_{dr}^{\text{cs}} \frac{a_{rd}}{\sum_{d \in D} a_{rd}}\right] v_r(D) + \phi^{\text{reg}} \sum_{r \in R} (\hat{\mu}_r - v_r(D)). \tag{25}$$

This approximation can be easily extended for a probability distribution rather than a single expected value. For example, the distributions ξ_P and ξ_C in Section 3.1 can be used to generate multiple scenarios for which the approximation is repeated. The average value can then be used as an approximation of the costs.

3.3. Determining depot locations

The cost for a given set of depot locations can be approximated efficiently using the methods discussed in the previous section. Despite this, in large urban networks, the number of options for depot locations to consider can still be extremely large. Specifically, the number of possible combinations grows exponentially with the number of possible depot locations and therefore with the size of the network. Enumerating all options is impossible for large networks and therefore we design an efficient heuristic to determine the best depot locations.

We propose a Large Neighborhood Search (LNS) heuristic to solve this problem. LNS heuristics explore a complex neighborhood to find better candidate solutions (Pisinger and Ropke, 2010). We efficiently explore the neighborhood by using metrics for the quality of solutions, that allow us to select candidates to *destroy* and *repair* in a smart way. The advantage of this algorithm is that we do not need to evaluate the full neighborhood at every iteration but we can use a quality and similarity metric to select a single candidate which is evaluated. Thereby, the high level of randomness allows for diversification of the search and therefore finding robust solutions. The general structure of the algorithm is described in Algorithm 1. The remainder of this subsection describes the initialization of the algorithm and the details of the destroy and repair operators, that exploit the specific features of our problem.

Initialization

To initialize the heuristic, we compute several components that are input to the algorithm. First, we compute the 3-dimensional matrix E with elements e_{rcd} for every potential depot. Thereby, we compute the single-depot objectives for each depot $d \in D^p$ which will be used as a quality metric for the depots. This metric will be referred to as m_d, which is defined as $m_d = C(\emptyset) + \phi^{\text{depot}} - C(\{d\})$. The intuition behind this is that a depot that performs well on its own is more likely to perform well in combination with other depots. Nevertheless, depots that are serving parcels with similar destinations and attract crowd-shippers with similar itineraries might perform poorly if they operate together, as one depot has little added value over the other. For this reason, we construct a similarity measure s_{d_1,d_2} for how similar two depots are in terms of the service area of crowd-shippers using that depot. Specifically, s_{d_1,d_2} is determined as follows:

$$s_{d_1,d_2} = \frac{[\sum_{r \in R} \sum_{c \in \tilde{C}} \min(e_{rcd_1}, e_{rcd_2})\lambda_c]^2}{[\sum_{r \in R} \sum_{c \in \tilde{C}} e_{rcd_1}\lambda_c][\sum_{r \in R} \sum_{c \in \tilde{C}} e_{rcd_2}\lambda_c]}. \tag{26}$$

Clearly, two depots that are very similar in terms of the service area are likely to have a lower gain in performance when they are combined. For this reason, this similarity measure will be used to select dissimilar depots to be combined.

We use a multi-start heuristic so that we randomly determine η initial solutions. Every initial solution is generated according to a simple construction heuristic. Every depot is randomly chosen with a probability proportional to the quality of the depot in the single-depot solution. By using a multi-start heuristic we aim to increase the search space and therefore decrease the likelihood of ending up at a local optimum.

Body of algorithm

We terminate the LNS algorithm after a fixed number of κ iterations. In every iteration, we consider the following operations on the current solution Ω and obtain the corresponding objective value. A newly generated solution is always accepted if it is an improvement over the previous solution and a worse solution is never accepted.

1. O_1 - *Repair operator:* For every depot d that is not in the current solution Ω, we compute the following metric: $\frac{m_d^\alpha}{[\sum_{\omega\in\Omega} s_{d,\omega}]^\beta}$ where α and β are tuning parameters that determine the relative importance of single-depot performance and inter-depot similarity. This metric determines the best depot to be added to the current solution, taking into account the quality of the depot in the single-depot solution as well as the similarity to the other depots in the current solution. We use the sum of the similarity with all depots in the current solution Ω, such that a depot that is similar to two depots rather than only one should have a lower metric value. Alternatively, the maximum similarity across all depots in the current solution Ω can be used in the denominator to replace the sum. A new depot is added randomly with a probability proportional to the value of this metric.
2. O_2 - *Destroy operator:* For every depot d that is in the current solution Ω, we determine the same metric as in O_1 and randomly drop a depot with a probability proportional to the inverse of the metric in O_1. By taking the inverse, depots that are very similar to other depots and depots with relatively low single-depot performance are most likely to be removed.
3. O_3 - *Swapping operator:* A sequential combination of O_1 and O_2 where a depot in the current solution Ω is replaced by another depot that is not in the current solution. We first destroy a depot $\omega \in \Omega$ using destroy operator O_2 and then add a depot using repair operator O_1, based on the similarity with the remaining depots $\Omega \setminus \{\omega\}$.

Algorithm 1: Large Neighborhood Search Algorithm

Input: For every depot d the quality metric m_d and for every pair of depots (d_1, d_2) the similarity metric s_{d_1,d_2}
for $n \in [1, \eta]$ **do**
  Generate an initial solution Ω_n^0
  for $k \in [1, \kappa]$ **do**
    $\Omega_n^k \leftarrow \Omega_n^{k-1}$
    Destroy a depot in Ω_n^k according to O_2
    Repair a depot in Ω_n^k according to O_1
    if $C(\Omega_n^k) > C(\Omega_n^{k-1})$ **then**
      $\Omega_n^k \leftarrow \Omega_n^{k-1}$
return $\arg\min_{\Omega\in\{\Omega_1^\kappa,\ldots,\Omega_\eta^\kappa\}} C(\Omega)$

4. Discrete event simulator

We develop a discrete event simulator to simulate the dynamic operational process. Considering the decision process in Fig. 1, the depot locations are determined using the methods described in Section 3, and the second and third-stage decisions are simulated. We consider various assignment strategies both for the second and third stage decisions, of which we evaluate the performance in Section 5.6.

The simulator is initialized by generating a set of parcel requests consisting of only a destination region (the origin is at one of the depots and will be determined in the second stage) and a set of potential crowd-shippers consisting of an origin and destination region and a starting time of the trip. For the sake of this simulation, crowd-shippers are assumed to make themselves available at the start of their trip. All generated parcel requests are assigned to a depot based on one of the strategies described in Section 4.1. All potential crowd-shippers are sorted in ascending order of their start times. The generation of parcels and crowd-shippers is performed using a pseudo-random number generator, such that simulations using various policies can be directly compared.

Upon the arrival of a crowd-shipper, a parcel or a set of parcels is assigned to this crowd-shipper based on one of the strategies described in Section 4.2. After the assignment, the assigned parcel(s) is/are reserved for the crowd-shipper for pickup and the crowd-shipper departs from his origin to the origin of the parcel(s). A new pickup event is scheduled, taking into account the travel time between the origin of the crowd-shipper and the origin of the parcel. As soon as a parcel is assigned to a crowd-shipper, it is no longer available to be assigned to other crowd-shippers, even when it is not yet picked up.

For every pickup event, a delivery event is scheduled taking into account the travel time between the origin and destination of a parcel. As only parcels with identical origins and destinations are assigned to the same crowd-shipper, pickup events that correspond to different parcels but the same crowd-shipper occur simultaneously. The same is true for delivery events. For every delivery event, the number of served parcels and total costs are updated and the detour made by the crowd-shipper is stored. The simulation ends when all parcels have been delivered or when all crowd-shippers have either completed a delivery or have failed to be assigned to a parcel. We emphasize that the simulator allows us to consider time synchronization constraints. Crowd-shippers are considered in order of their arrival time and the system is constantly updated such that only the parcels that are available at the arrival time of the crowd-shipper are considered for pickup.

4.1. Stage 2: Parcel-depot assignment

After the depots are determined in stage 1, parcels have to be assigned to depots on a day-to-day basis. At this stage, parcels are assumed to be known exactly (all orders of parcels have been collected), but crowd-shippers can announce their availability last minute and are therefore unknown. By simply assigning parcels to the closest depot in terms of distance, the importance of the flow of potential crowd-shippers is neglected. For the sake of comparison, we consider a distance-based metric that assigns all the parcels of a region to the opened depot that is closest to that region. Consider the set of opened depots D and consider μ_r parcels with destinations in region R. We recall that the travel time between depot d and region r is defined as t_{dr}. We define a_{rd} the number of parcels with final destination $r \in R$ that are stored at depot $d \in D$ as follows:

$$a_{rd} = \mu_r \mathbb{1}\left[t_{dr} = \min_{d' \in D} t_{d'r}\right]. \tag{27}$$

Although a depot can be close in terms of distance, if very few crowd-shippers can feasibly deliver a parcel from that depot to the final destination, such a parcel-depot assignment can perform poorly. Therefore, we develop an assignment strategy based on the CA-estimates obtained using the algorithm as described in Section 3.2. We solve the single-depot-approximation for every depot $d \in D$ to obtain the expected number of parcels delivered to region $r \in R$, $v_r(\{d\})$. To obtain the single-depot approximations, we ignore the capacity of crowd-shippers to be slightly more conservative, which has shown to perform better. We then assign the parcels proportional to the expected number of parcels delivered from a specific depot location. In this case, we define a_{rd} as follows:

$$a_{rd} = \mu_r \frac{v_r(\{d\})^\gamma}{\sum_{d \in D} v_r(\{d\})^\gamma}. \tag{28}$$

The tuning parameter γ can be used to give extra weight to larger depots (depots with high expected deliveries) and less weight to smaller depots (depots with lower expected deliveries). We emphasize the similarity of this assignment and parcel-depot assignment in the CA estimation in Eq. (24).

4.2. Stage 3: Parcel-crowd-shipper matching

In the third stage of our problem, when the depots are known and parcels are distributed over these depots, parcels have to be assigned to crowd-shippers. Generally, parcels are matched to crowd-shippers dynamically, upon arrival of the crowd-shippers. A parcel can only be matched to a crowd-shipper if they can pick up and deliver the parcel within their maximum detour, this is referred to as a *feasible match*. If a crowd-shipper can be feasibly matched to multiple parcels, the operator has to decide which parcel to assign to the crowd-shipper to maximize the total number of delivered parcels over the entire planning horizon. We consider the following three alternative matching approaches. The static matching approach relies on solving an integer linear programming problem, whereas the minimal-detour and CA-based matching use a simple decision rule. Furthermore, the static matching approach uses information about future crowd-shippers, whereas the other two approaches, more realistically, only consider one crowd-shipper at a time.

Static matching

The static matching approach assumes complete knowledge of all future crowd-shippers that will arrive. The static matching can be obtained by solving an ILP problem, that can be taken from the third stage of the stochastic programming formulation (8)–(13). Here, the depots are fixed and the parcel-depot assignment has been made. Therefore, the variable y_{pd} can be fixed to 0 or 1 according to the previously made assignments. This then simplifies the right-hand side of Constraints (10), while the rest of the formulation remains unchanged.

As the static matching is made without uncertainty about the future crowd-shippers, this forms a lower bound to the other matching approaches. In reality, this could be very unrealistic if crowd-shippers make themselves known only shortly before departing. Despite this, it forms a useful benchmark to compare the dynamic assignment strategies as well as the continuum approximation.

Minimal-detour matching

The minimal detour matching matches the crowd-shipper to the parcel for which the crowd-shipper minimizes the detour. This matching strategy, therefore, does not use any knowledge of future crowd-shippers. This strategy minimizes crowd-shipper inconvenience but is likely to be suboptimal as it does not use any information on future crowd-shippers. We consider that the set P only contains parcels that are not yet assigned to other crowd-shippers, and we consider $c \in C$ the current crowd-shipper. We let $or(\cdot)$ and $dest(\cdot)$ be the origin and destination location, respectively, for a crowd-shipper or parcel. Then the minimal detour matching chooses the parcel that minimizes the detour a crowd-shipper makes to pick up and deliver a parcel. This is computed as follows:

$$p_{min} = \arg\min_{p \in P}\left[t_{or(c)or(p)} + t_{or(p)dest(p)} + t_{dest(p)dest(c)} - t_{or(c)dest(c)}\right]. \tag{29}$$

If multiple parcels with the same origin and destination have the lowest detour, up to q_c parcels are assigned to the current crowd-shipper c.

CA-based matching

The minimal-detour matching only uses information on the crowd-shipper that is currently available. In this way, part of the information about potential crowd-shippers arriving in the future remains unused, although this can be useful. Generally, historic data is available that provides insights into the expected number of crowd-shippers. In practice, expectations of demand based on historic data are used to make strategic decisions. Actual demand is usually known at least one day in advance and is therefore used for operational decisions. The CA-based matching approach exploits the information on the approximated number of delivered parcels to improve the matching quality. An arriving crowd-shipper is assigned the parcel with the destination region that has the lowest expected number of delivered parcels relative to the total demand in that region ($\frac{\upsilon_r}{\hat{\mu}_r}$). The intuition behind this assignment strategy is that we favor parcels that are less likely to be delivered in the future. By doing so, we increase the total expected number of parcels delivered over the entire planning horizon. If the chosen destination region has multiple parcels available, up to q_c parcels are assigned to the current crowd-shipper c. To obtain the estimates υ_r, we disregard the capacity of crowd-shippers to be slightly more conservative. This shows an improvement in the performance of the CA-based matching. Compared to static matching, CA-based matching only uses a simple metric to determine the assignment, rather than solving an ILP problem. Therefore, the match can be determined extremely fast, similar to minimal-detour matching, which forms a major advantage for our real-time application. Here, we do not take into account fluctuations in the arrival rates of crowd-shippers during the day. This is marked as an interesting direction for future research.

5. Results

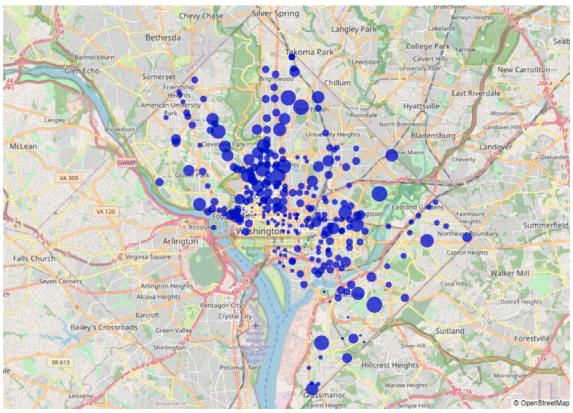
In this section, we evaluate the performance of the developed CA approach to find the optimal depots, as well as the potential of depot-based crowd-shipping. Our results are obtained through a case study based on the city of Washington DC, of which the details are described in Section 5.1. In Section 5.2 we evaluate the accuracy of our continuum approximation by comparing it to two exact benchmarks. In Section 5.3 we further evaluate the performance of our CA approach by comparing our solution to the solution obtained by solving the discrete formulation using a CPLEX solver. In Section 5.4 we evaluate the results on the network and in Section 5.5 we perform a sensitivity analysis on the optimal number of depots. Finally, in Section 5.6 we evaluate the effect of incorporating historic information about crowd-shippers and parcels in the three levels of decision-making on the results.

5.1. Case study

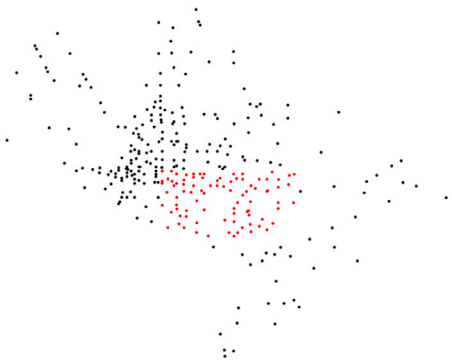
We use the city of Washington DC and the surrounding metropolitan area as a case study. The city of Washington DC has around 700,000 inhabitants and the entire agglomeration has around 7 million inhabitants. Washington DC hosts one of the biggest Bike-sharing platforms in the USA: Capital Bikeshare. Capital Bikeshare has over 500 stations and 4500 bikes (Capital Bikeshare, 2020). Such a bike-sharing platform with a large number of users forms a good base for a crowd-shipping service. We consider the locations of the bike-sharing stations as potential depot locations and use them as approximations for regions making up the entire service area.

Based on the station names, approximate coordinates of the locations are extracted from Google Maps (2020). An approximation of the surrounding population has been made using Census Reporter (2021) data, which in turn has been used as a proxy for demand for small parcels. We note that the actual coordinates of the stations may slightly deviate from the approximated coordinates due to misinterpreted station names. Nevertheless, the obtained network is used as a representation of an actual network. A bike-sharing station is used as a potential demand region with the expected number of parcels proportional to the population around that station. Historic system data from the Capital Bikeshare (2020) database has been used to identify origin–destination pairs for the crowd-shippers. The stations that are used as regions are displayed in Fig. 3(a). This figure displays a bubble chart of those regions, where the size of the bubble is determined relative to the population around the corresponding station. Whereas the most used stations in terms of origins and destination of crowd-shippers are around Union Station, the Mall, and the center of Washington DC, demand is higher in the suburbs. This shows the large asymmetry in crowd-shippers' origin and destination locations on the one hand and parcel destinations on the other that is usually present in crowd-shipping systems in real urban areas, making this case study especially realistic and interesting. We consider 330 stations in the center of Washington DC and the nearest suburbs, as displayed in Fig. 3(a). In Sections 5.2 and 5.3 we consider a smaller network of only 90 nodes in the center, as the benchmark approaches used in those sections are computationally intractable for the larger instance. The network is displayed in Fig. 3(b) with the smaller subnetwork in red. For the sake of computation time, we only use crowd-shippers that have their origin and destination within the selected area. In our experiments, the number of parcels has been fixed at the expected value such that crowd-shippers are the only uncertain variable in the problem.

We use the following parameter values to get a realistic interpretation of the results. We assume that the cost of opening and operating a depot ϕ^{depot} is equal to 1000$ per day. This is based on the average rental price in Washington DC to place the storage lockers and the maintenance cost involved in operating the depots. The cost of regular delivery is set to be 15$, similar to Le et al. (2019). A part of these costs is still made to serve the depots, therefore, we consider a slightly lower value for ϕ_p^{reg} equal to 10$. Our baseline scenario assumes a homogeneous set of crowd-shippers that are willing to make a detour (τ) of at most 500 m and wish to receive (ϕ_p^{cs}) 5$ to make a delivery plus 1$ for every kilometer traveled with a parcel. The daily demand for parcels is assumed to be roughly 20,000 per day, proportional to the population in a region. The number of potential crowd-shippers is set to approximately 25,000. Of these crowd-shippers, 60% can carry only 1 parcel, 30% can carry 2 and 10% can carry up to 3 parcels. The demand scenario is fixed to the expected value, whereas 10 different supply scenarios are generated based on a Poisson process.



(a) Bubble chart of bike-sharing stations, where the size of the bubble is determined by the population in the area.



(b) Abstract network used as input to the optimization and simulation, with smaller subnetwork in red

Fig. 3. Network of Washington DC used for the case-study. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The LNS parameters are tuned in order to find a good objective value within a reasonable time. The algorithm uses 5 initial solutions (η) and 500 iterations (κ). The repair and destroy operators take parameters α equal to 4.5 and β equal to 8. These values were chosen based on a grid search over a large range of parameter values. The parameters were chosen such that they maximize the number of multi-start initial solutions for which the best objective was found and at the same time minimize the number of iterations needed to find this objective. The tuning parameter γ, used for the parcel-depot assignment, is chosen equal to 1. CPLEX version 12.6.3.0 is used in Java to solve all MILPs.

5.2. Comparison of continuum approximation to static and dynamic assignment strategies

To evaluate the quality of the continuum approximation, we compare the estimated total costs to two benchmarks. The first benchmark assumes full knowledge of crowd-shippers *before* parcels are distributed to the depots. This means that we use the formulation of the second and third-stage as given by Eqs. (4)–(13), but for a single realization of parcels and crowd-shippers which is known with certainty. This eliminates the expected value in Eqs. (4) and (8) and allows us to solve the problem to optimality using a standard Mixed Integer Programming (MIP) solver, such as CPLEX. The objective is to minimize the operational costs which are given in (8). The second benchmark is a realistic dynamic assignment procedure. We use the CA-based parcel-depot assignment and the CA-based matching of parcels and crowd-shippers, which outperforms other assignment strategies (see Section 5.6).

For the sake of computation time, we consider a subset of the full network of only 90 regions and we set the capacity of every crowd-shipper to 1. In this way, we eliminate Constraints (12), which significantly reduces the computation time of the static assignment method. A large sensitivity analysis is performed for the expected number of potential crowd-shippers ($\hat{\lambda}$), the maximum detour (τ), and the number of depots ($|D|$). We use daily demand for parcels $\hat{\mu}$ equal to approximately 4000 and varying $\frac{\hat{\lambda}}{\hat{\mu}}$ from 0.5 to 2. We vary τ from 250 m to 1000 m and we consider 1, 3, 5, and 7 depots, which is appropriate given the small network. Given the lower number of parcels, we also change the depot cost ϕ^{depot} to 100$. We compare the objective obtained through the CA approach as well as the LP approximation. The results are visualized in Figs. 4 to 7, that display the percentage difference between the predicted objective and the actual (simulated) objective. We compare the LP-approximation and the algorithmic approximation to a static and dynamic assignment strategy. A set of 10 demand and supply instances is used for all simulations.

It is clear that the LP approximation performs well compared to the static benchmark, with near 0 differences between the approximation and the actual objective. The reason for this is that both the benchmark and the approximation disregard uncertainty and are therefore able to reach a lower bound on the total costs. However, compared to the more realistic dynamic benchmark the LP approximation underestimates the actual total costs by on average 5% and can go up to 13%. Clearly, with increasing uncertainty, the performance of the LP approximation will decrease. Thereby, the performance also deteriorates with the number of hubs and slightly deteriorates with the number of crowd-shippers and the maximum detour. We emphasize that the objective generally decreases when $\frac{\hat{\lambda}}{\hat{\mu}}$, τ, and $|D|$ increase and therefore the percentual difference between the actual and approximated objective is amplified.

The CA approximation accounts for uncertainty between the two lower levels of decision-making whereas the static benchmark ignores this uncertainty. Therefore, the CA approximation overestimates the objective of the static benchmark and performance deteriorates especially when the number of hubs increase as the static benchmark can optimally make the parcel-hub assignment, whereas the CA approximation cannot. Compared to the more realistic dynamic benchmark, the approximated objective differs on average 2% from the actual objective with ranges between 0 and 5%. A comparison of Figs. 5 and 7, shows that incorporating the uncertainty between the two lower-level decisions significantly improves the performance of the approximation. With respect to

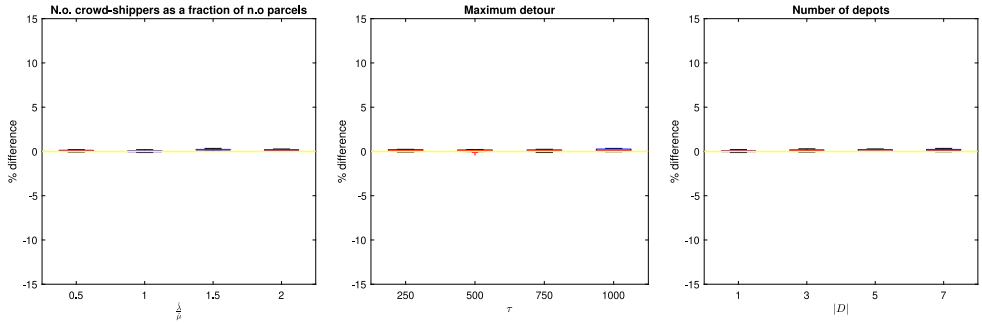


Fig. 4. Percentage difference between the predicted objective by the LP approximation and the actual objective simulated by the static assignment strategy.

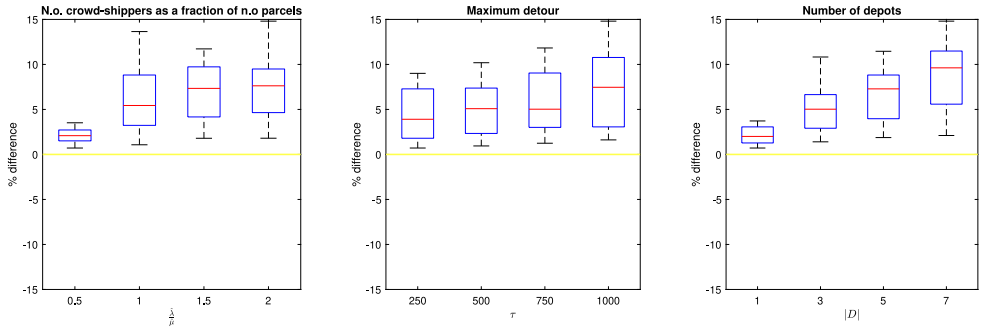


Fig. 5. Percentage difference between the predicted objective by the LP approximation and the actual objective simulated by the dynamic assignment strategy.

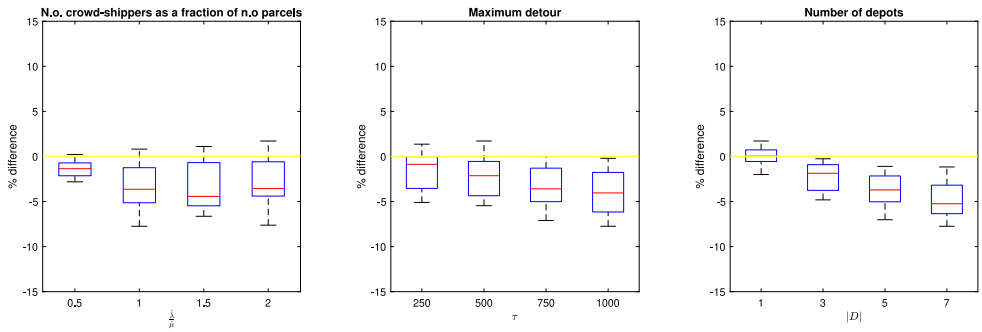


Fig. 6. Percentage difference between the predicted objective by the CA approximation and the actual objective simulated by the static assignment strategy.

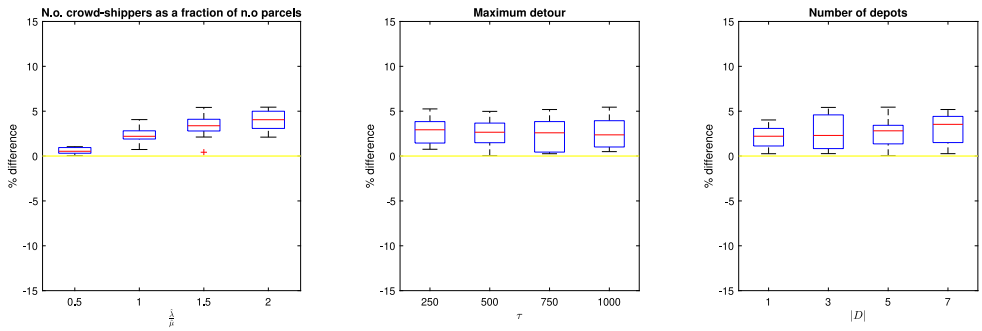


Fig. 7. Percentage difference between the predicted objective by the CA approximation and the actual objective simulated by the dynamic assignment strategy.

computational time, the CA approximation is on average 100 times faster on the small network. Thereby, the number of variables and constraints will increase for the full-size network, making it even more computationally demanding for the problem to be constructed and solved by CPLEX.

5.3. Comparison of CA approach to discrete formulation

In this section, we further evaluate the performance of our algorithm by comparing the CA-based solution algorithm proposed in this work to solving the discrete formulation in Section 3.1 using CPLEX. For the discrete formulation to be solvable within a reasonable amount of time, we consider a subset of the network with 90 nodes, limit the number of scenarios in ξ_P and ξ_C to 1 and the capacity of the crowd-shippers to 1. In this way, the expected values in the objectives in Eqs. (1) and (4) are eliminated and due to the full knowledge of supply and demand, the variables y_{pd} can be omitted as well. Thereby, the capacity of 1 allows us to eliminate Constraints (12). We emphasize that this significantly simplifies the discrete formulation, as for larger networks with more realistic settings (such as those considered in Sections 5.4–5.6), CPLEX fails to find an optimal or feasible solution or even fails to construct the model due to the size of the problem. We also note the discrete formulation requires integer inputs. In the 90×90 network, this may lead to a relatively high difference between the two estimates which could potentially influence the solution.

The results are displayed in Fig. 8. The left-hand panel displays the objective of the CA method relative to the objective of the discrete formulation. The objective values were computed as an average of 10 simulations using a CA-based dynamic assignment strategy for the parcel-depot and parcel-crowd-shipper assignments. Rather than directly comparing the objective values, we subtract a baseline of €5 for every parcel to properly quantify the percentual difference in the objective value. The right-hand panel displays the computation times of both methods for various settings, where we note the log scale of the y-axis. A time limit of 1 h has been implemented and CPLEX solves the problem up to a 5% optimality gap (without this, the solver may continue looking for negligible improvements, creating a biased comparison of CPU times). For this experiment, demand for parcels $\hat{\mu}$ and the number of crowd-shippers $\hat{\lambda}$ vary between 1000 and 2000 and the maximum detour is either 250 m or 500 m. Compared to the previous experiment, smaller values have been chosen such that the discrete formulation is solvable within a reasonable amount of time.

For the comparison of the objective values, we emphasize that the hub locations for both the discrete formulation and the CA-based algorithm were determined using only a single scenario, which can lead to suboptimality in the dynamic simulation setting. We observe that the CA-based method generally obtains better solutions, which can be partially explained by the chosen 5% optimality gap for the discrete formulation. Clearly, the results obtained by the CA-based algorithm are more robust and find good solutions even when using only a single scenario. The discrete formulation, on the other hand, does not find robust solutions and requires a higher number of scenarios to adapt the depot locations. Thereby, the current simplification ignores the uncertainty which is captured by the CA-based algorithm, which may therefore also lead to suboptimality. We also observe that for settings for which the discrete formulation cannot be solved within the 1-h limit, the improvement of the CA-based algorithm over the discrete formulation is significantly higher.

Computation times are significantly higher for solving the discrete formulation than for the CA-based method. Computation time for both methods increases with the number of hubs. However, the CA-based approach is only marginally influenced by the number of crowd-shippers, parcels, and maximum detour. Averaged over all tested settings, the CA-based algorithm is almost 150 times faster than solving the discrete formulation. For the high-demand case with 2000 parcels, this even goes over 400 times faster, despite computation time being limited to one hour. Thereby, we emphasize again that for larger networks with more realistic settings such as multiple scenarios, the discrete formulation cannot be used at all whereas the CA-based method only requires a couple of minutes to find high-quality solutions.

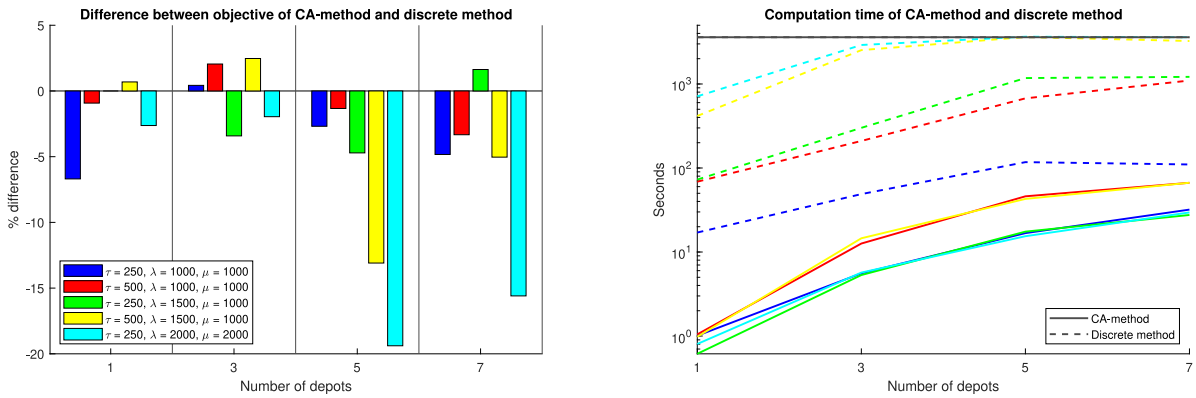


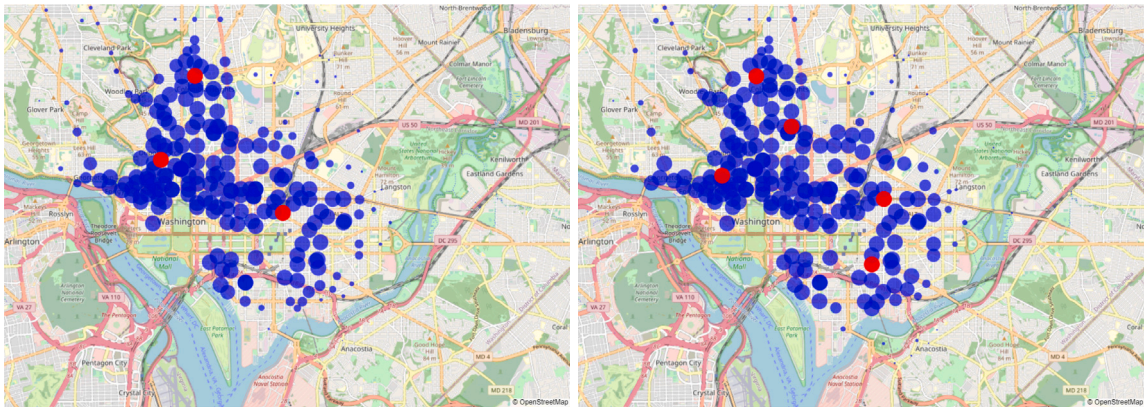
Fig. 8. Comparison of the CA-based method and solving the discrete formulation.

Another alternative method is to use a simulation–optimization approach in combination with the described large neighborhoods search algorithm. Here, a simulation is used to evaluate the objective rather than the CA-based approximation. The performance of this method is highly dependent on the details of the simulator and the efficiency of the implementation. Thereby, to obtain a good estimate, the simulation has to be repeated many times to obtain an average. For large-scale systems, such as the one considered in

the remainder of this paper, evaluating the objective by simulation is computationally too time-consuming to obtain results within a reasonable amount of time or will also run into memory issues like the discrete formulation. Note that even the initialization of the LNS algorithm requires strong computational effort for large systems.

5.4. Results on the network

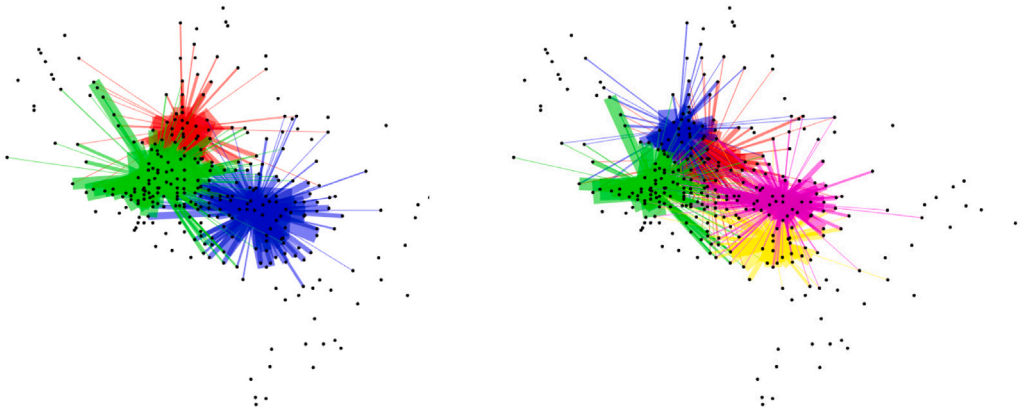
To obtain managerial insights regarding the exact location of depots, we evaluate the depot locations in the network. We emphasize that from this section onwards we use the 330 node network with the baseline parameters outlined in Section 5.1. The bubble chart in Fig. 9 displays the considered network where a blue bubble represents a regular demand region and a red bubble represents a demand region that was chosen to have a depot. The size of the bubble represents the fraction of demand that was served by crowd-shippers. That is, a full bubble implies that all parcels in the region are expected to be delivered by crowd-shippers, according to the continuum approximation, whereas a smaller bubble implies that only a fraction of the parcels is expected to be delivered by crowd-shippers. Fig. 9(a) displays the approximation for three depots and Fig. 9(b) displays the approximation for five depots.



(a) Three depots (b) Five depots

Fig. 9. Bubble chart of parcels served by crowd-shippers in the network. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We observe that depots are located in the city center of Washington DC where the number of potential crowd-shippers is the highest. As we are using bike-sharing users to approximate the commuting patterns of crowd-shippers, these are mainly in the city center. We notice that most depots are in the northwest of the city center, where demand is the highest as depicted in Fig. 3(a). In addition to this, depots are spread sufficiently to attract more crowd-shippers and to have a broader service area. Many of the chosen depots are either at popular origins of potential crowd-shippers, such as a train station, or at popular intersections where many crowd-shippers pass by either directly or within their maximum detour.



(a) Spider chart with 3 depots (b) Spider chart with 5 depots

Fig. 10. Spider chart linking the origins and destinations of delivered parcels.

The strong inter-dependency between the depots is shown in Fig. 10, which displays from which depots the parcels delivered to each region originate. Fig. 10(a) shows the spider chart for three depots and Fig. 10(b) shows the spider chart for five depots. The

majority of the destination regions are served by multiple depots. Only the regions in the outskirts of the network are served by a single depot. To quantify the results in this figure, we calculate how many regions are served by more than one depot. Specifically, we count the number of regions for which at most 90% of the delivered parcels originate from one depot. For three depots, 37% of the regions where at least one parcel is delivered are served by more than 1 depot. For five depots, this is as high as 70%. Typically we observe that the higher the number of depots, the lower the number of regions served by a single depot and thus the higher the inter-dependency. Intuitively, this corresponds to the fact that when more depots are constructed, the distance between depots is lower, and therefore their similarity (see Section 3.3) increases.

5.5. Optimal number of depots

In this section, we evaluate the optimal number of depots for τ varying between 500 and 1000 m. Fig. 11 displays the costs for a varying number of depots, as approximated by the CA approach, for a set of depots that is obtained through the CA-based heuristic. The total costs are split into the three components that make up the objective function in Eq. (25) and consist of costs of maintaining depots, crowd-shipper compensations, and penalties for undelivered parcels.

Increasing the value of τ has two opposing effects on the total costs. On the one hand, potential depot locations that previously were not able to serve sufficient demand to be profitable may now be able to serve more demand because the maximum detour is higher, thereby increasing the optimal number of depot locations. On the other hand, a depot may be able to reach more demand regions due to the increase of τ, making other depots obsolete, thereby reducing the optimal number of depots. A similar effect is true for an increase or decrease in the total expected number of crowd-shippers, which may either lead to an increase or decrease in the optimal number of depots. We emphasize that this is specific to the chosen scenario and costs.

The marginal percentage of demand served due to the addition of one more depot is diminishing. The first depots are the most profitable and can serve a relatively large portion of the demand. As we showed in Section 5.4, these depots are built at central locations where the number of potential crowd-shippers is high. Afterward, additional depots can be opened at less busy locations to further increase served demand, but the effect is substantially lower.

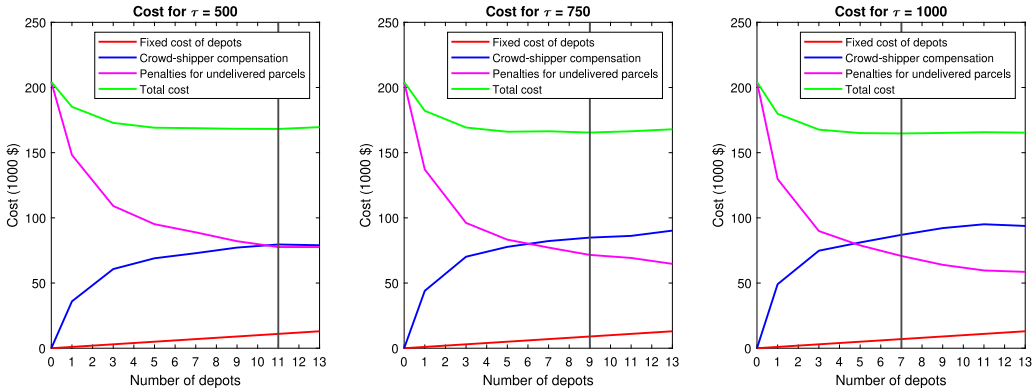


Fig. 11. Decomposition of costs for varying number of depots and maximum detour τ.

5.6. Comparison to non-predictive strategies

As described in Section 3, the proposed solution approaches to all three stages contain a predictive component to incorporate the influence of parcel and crowd-shipper patterns and their interaction in decision-making. We use historic information on parcels and crowd-shippers to enhance the decision-making on all three layers. In this section, we compare the predictive CA-based strategies to those that do not use such a predictive component and only base the decisions on geographical distances. The first stage decisions are compared to an FLP minimizing the total distance of every region to the closest depot. The second stage parcel-depot assignment is compared to an assignment strategy where each parcel is assigned to the closest opened depot. The third stage matching is compared to a minimal-detour strategy for crowd-shippers. These non-predictive approaches ignore the connection between the three stages and solve every stage separately, basing their decisions solely on distance rather than estimates of the pattern of parcels, crowd-shippers, and the interaction of the two. The service levels are displayed in Table 1. For every scenario, the percentage of parcels served by crowd-shippers is an average of 10 simulation runs. We consider a base case with 5 depots, roughly 20,000 parcel requests, and 25,000 crowd-shippers (with varying capacities that are 1.5 on average). The maximum detour for crowd-shippers is 500 m.

Using predictive methods that incorporate expected patterns and interactions of parcels and crowd-shippers significantly improves the performance of the depot-based system. Especially the use of our algorithm to determine the optimal depot locations compared to a simple distance-based algorithm that chooses depots at central locations increases the service level by more than 10%. Additionally, using predictive strategies to make second and third-stage decisions can improve the objective by up to 5%. Overall, the results indicate that for this specific set of configurations (i.e., number of depots, maximum detour, etc.) using predictive components in all three levels of decision-making can improve the service level by 15%. We emphasize that results may vary for

Table 1
Comparison of predictive CA-based strategies to non-predictive strategies.

First-stage	Second-stage	Third-stage	
		Non-predictive	Predictive
Non-predictive	Non-predictive	35.4	38.4
	Predictive	36.5	38.7
Predictive	Non-predictive	46.1	48.1
	Predictive	48.1	50.9

The reported values are the number of parcels delivered by crowd-shippers as a percentage of the total demand for parcels.

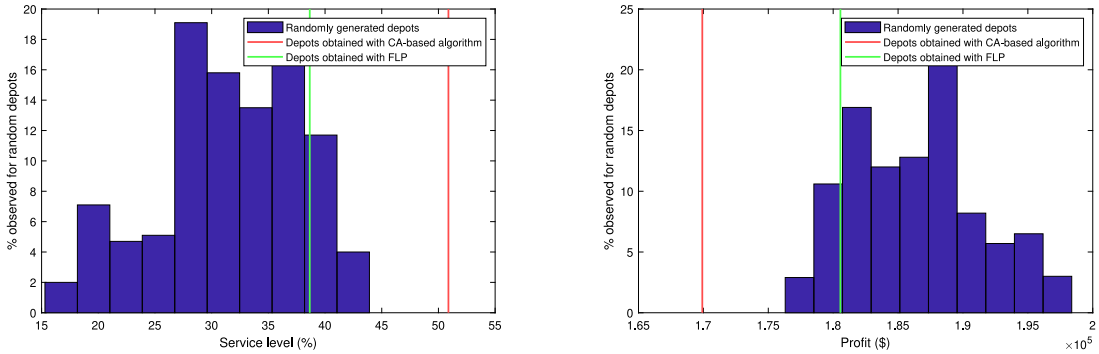


Fig. 12. Histogram of service level and profit for random sets of 5 depots.

other configurations. For example, if we increase the number of depots to 10, the fully non-predictive service level is 44%, whereas the fully predictive service level is 56.1%. Typically, we observe that higher service levels are more difficult to improve.

For the same settings as before, we compare the service level and profit of the predictive and non-predictive approach to randomly generated sets of 5 depots. This provides additional insights into the performance of our methods as well as into the influence of the exact depot locations on service level and profit. We randomly generate 100 sets and evaluate them all on the same 10 instances. This then yields 1000 observations of profit and service level that are displayed in Fig. 12. The spread of the histograms shows that the exact locations of depots significantly influence the service level and profit of the crowd-shipping system. Clearly, the non-predictive FLP method is outperformed by some randomly generated sets. The predictive CA-based approach, on the other hand, outperforms all randomly generated sets.

6. Conclusion

Crowd-shipping is a promising alternative to traditional last-mile delivery methods that can help to reduce congestion, and pollution and improve the overall performance of the delivery system. One of the main drivers is the availability of sufficient suppliers. In this paper, we proposed a depot-based crowd-shipping system where crowd-shippers pick up parcels from depot locations and deliver them to the final destination. By constructing depots at strategic locations, more crowd-shippers can be attracted to serve more demand.

We approximate the lower-level decisions of assigning parcels to depots and later to crowd-shippers using a Continuum Approximation (CA) approach. The approximation shows to provide an accurate estimation of the actual objective using dynamic assignment strategies, on average being within 2% of the optimal dynamic assignment problem, with deviations being at most 5% on the tested instances. We developed a heuristic approach to efficiently determine the optimal depot locations. The heuristic uses CA to approximate the lower-level operational decisions and a large neighborhood search heuristic to make the strategic upper-level decisions (i.e., find the best set of depots). Using quality and similarity metrics, the search space is explored efficiently and a solution is found within a reasonable amount of time. We compare the performance of our algorithm to solving a discrete formulation. This comparison indicates that, on a set of smaller instances of 90 regions, our algorithm is on average almost 200 times faster on the tested, going up to 1000 times for the largest tested instance. In terms of solution quality, our algorithm generally finds slightly better solutions for instances where the CPLEX solver converges with the one-hour time limit, but significantly better solutions for instances where the solver does not terminate. For realistic instances of 330 regions, our algorithm can find good depot locations within a reasonable amount of time whereas the cannot even be constructed, let alone find feasible solutions.

The estimates of the CA approach can also be used to develop dynamic assignment strategies for the parcel-depot and the parcel-crowd-shipper assignment. These assignment strategies then incorporate the interaction of potential crowd-shippers and

parcels, whereas classic strategies are mostly distance based. A comparison of our suggested approach where all three levels of decision-making incorporate historic information of crowd-shippers and parcels outperforms distance-based methods by 15%.

A natural extension of this work is to incorporate vehicle routing costs directly into the depot location problem. The framework designed in this paper allows to efficiently incorporate routing decisions as long as the problem is solved or approximated fast. Routing decisions can then be included as second-stage decisions. Furthermore, a generalization of the problem to include (soft) time windows would enable customers to indicate their preferred delivery time.

CRediT authorship contribution statement

Patrick Stokkink: Conceptualization, Methodology, Investigation, Validation, Writing – original draft, Writing – review & editing, Software. **Nikolas Geroliminis:** Conceptualization, Methodology, Investigation, Validation, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

Appendix

See Table 2.

Table 2
Notational glossary.

Problem description	
Sets	
C	Set of potential crowd-shippers (indexed c)
D	Set of opened depots (indexed d)
D^p	Set of potential depots (indexed d)
P	Set of parcels requested (indexed p)
Parameters	
q_c	Capacity of crowd-shipper c
$D^{\max}$	Maximum number of depots that can be built
f_{pcd}	Parameter indicating if crowd-shipper c can feasibly pick up and deliver parcel p, with an origin at depot d
t_{od}	Travel time between node o and node d
ϕ^{depot}	Fixed costs of opening a depot
ϕ^{cs}_{pd}	Reward given to crowd-shippers to pick up and deliver parcel p, with an origin at depot d
ϕ^{reg}_{p}	Cost of regular delivery of parcel p
$\psi_k(\cdot)$	Function describing the costs of the kth stage for a given set of inputs
ξ_P (ξ_C)	Random variable of which the realizations are vectors of parcel requests (potential crowd-shippers)
Decision variables	
x_{pc}	Decision variable indicating whether parcel p is matched to crowd-shipper c
y_{pd}	Decision variable indicating whether parcel p is assigned to depot d
z_d	Decision variable indicating whether depot d is opened
Continuum approximation algorithm	
Sets	
$\bar{C}$	Set of crowd-shipper classes
R	Set of regions
Parameters	
e_{rcd}	Binary parameter indicating if crowd-shippers of class c can feasibly pick up a parcel at depot d and deliver it to region r
$\tilde{e}_{rc}$	Binary parameter indicating if crowd-shippers of class c can feasibly pick up a parcel at *at least one* depot and deliver it to region r
$or(c)$	Origin of crowd-shippers in class c
$dest(c)$	Destination of crowd-shippers in class c
μ_r	Number of parcels with destination in region r
$\hat{\mu}_r$	Expected number of parcels with destination in region r
λ_c	Actual number of potential crowd-shippers of class c
$\hat{\lambda}_c$	Expected number of potential crowd-shippers of class c
τ_c	Maximum detour crowd-shippers of class c are willing to make to pick up and deliver a parcel
ϕ^{cs}_{rd}	Reward given to crowd-shippers to perform a pickup at depot d and delivery to region r
ϕ^{reg}_{r}	Cost of regular delivery of a parcel to region r
Variables	
a_{rd}	Number of parcels with final destination r stored at depot d
$C(D)$	Total expected cost of opening the set of depots D
l_r	Leftover parcels with a destination in region r in iterative CA procedure
u_r	Intermediate estimate for the expected number of parcels delivered by crowd-shippers to region r
v_r	Expected number of parcels delivered by crowd-shippers to region r
$v_r(D)$	Total expected number of parcels delivered by crowd-shippers to region r for a set of depots D
x_{rcd}	Decision variable indicating how many parcels with a destination in region r are assigned to depot d and crowd-shipper class c
$\tilde{\mu}_r$	Remaining expected number of parcels with destination in region r in iterative CA procedure
$\bar{\mu}_c$	Total number of parcels that can potentially be served by crowd-shippers of class c
$\tilde{\lambda}_c$	Remaining expected crowd-shippers of class c in iterative CA procedure

(*continued on next page*)

Table 2 (*continued*).

Large neighborhood search algorithm	
$s_{d_1 d_2}$	Similarity of depots d_1 and d_2
m_d	Quality metric for opening single depot d
α, β, γ	Tuning parameters
η	Number of initial solutions in depot-location algorithm
κ	Iteration limit of depot-location algorithm
Ω	Current solution (i.e., set of depots) in the depot-location algorithm

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transp. Sci. 52 (4), 965–981.

Akeb, H., Moncef, B., Durand, B., 2018. Building a collaborative solution in dense urban city settings to enhance parcel delivery: An effective crowd model in Paris. Transp. Res. E 119, 223–233.

Ansari, S., Başdere, M., Li, X., Ouyang, Y., Smilowitz, K., 2018. Advancements in continuous approximation models for logistics and transportation systems: 1996–2016. Transp. Res. B 107, 229–252.

Archetti, C., Savelsbergh, M., Speranza, M.G., 2016. The vehicle routing problem with occasional drivers. European J. Oper. Res. 254 (2), 472–480.

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

Blad, K., de Almeida Correia, G.H., van Nes, R., Annema, J.A., 2022. A methodology to determine suitable locations for regional shared mobility hubs. Case Stud. Transp. Policy.

Capital Bikeshare, 2020. Capital bikeshare. https://www.capitalbikeshare.com/system-data.

Census Reporter, 2021. Census reporter. https://censusreporter.org/. Accessed: 10-03-2021.

Cohn, A., Root, S., Wang, A., Mohr, D., 2007. Integration of the load-matching and routing problem with equipment balancing for small package carriers. Transp. Sci. 41 (2), 238–252.

Cornuéjols, G., Nemhauser, G., Wolsey, L., 1983. The uncapicitated facility location problem. Technical report, Cornell University Operations Research and Industrial Engineering.

Cui, T., Ouyang, Y., Shen, Z.-J.M., 2010. Reliable facility location design under the risk of disruptions. Oper. Res. 58 (4-part-1), 998–1011.

Daganzo, C.F., 2005. Logistics Systems Analysis. Springer Science & Business Media.

Dahle, L., Andersson, H., Christiansen, M., 2017. The vehicle routing problem with dynamic occasional drivers. In: International Conference on Computational Logistics. Springer, pp. 49–63.

Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. European J. Oper. Res. 54 (1), 7–22.

Ermagun, A., Stathopoulos, A., 2018. To bid or not to bid: An empirical study of the supply determinants of crowd-shipping. Transp. Res. A 116, 468–483.

Fabri, A., Recht, P., 2006. On dynamic pickup and delivery vehicle routing with several time windows and waiting times. Transp. Res. B 40 (4), 335–350.

Fernandes, D.R., Rocha, C., Aloise, D., Ribeiro, G.M., Santos, E.M., Silva, A., 2014. A simple and effective genetic algorithm for the two-stage capacitated facility location problem. Comput. Ind. Eng. 75, 200–208.

Gendron, B., Khuong, P.-V., Semet, F., 2016. A Lagrangian-based branch-and-bound algorithm for the two-level uncapacitated facility location problem with single-assignment constraints. Transp. Sci. 50 (4), 1286–1299.

Generation IM, Preston, F., Kukrika, N., Matthews, H.S., Martelo, M.A.J., 2020. The carbon footprint of retail: Ecommerce vs bricks & mortar.

Google Maps, 2020. Google maps. Accessed: 13-03-2020.

Iwan, S., Kijewska, K., Lemke, J., 2016. Analysis of parcel lockers' efficiency as the last mile delivery solution–the results of the research in Poland. Transp. Res. Procedia 12, 644–655.

Karak, A., Abdelghany, K., 2019. The hybrid vehicle-drone routing problem for pick-up and delivery services. Transp. Res. C 102, 427–449.

Le, T.V., Stathopoulos, A., Van Woensel, T., Ukkusuri, S.V., 2019. Supply, demand, operations, and management of crowd-shipping services: a review and empirical evidence. Transp. Res. C 103, 83–103.

Le, T.V., Ukkusuri, S.V., 2019. Influencing factors that determine the usage of the crowd-shipping services. Transp. Res. Rec. 2673 (7), 550–566.

Lei, C., Ouyang, Y., 2018. Continuous approximation for demand balancing in solving large-scale one-commodity pickup and delivery problems. Transp. Res. B 109, 90–109.

Li, B., Krushinsky, D., Reijers, H.A., Van Woensel, T., 2014. The share-a-ride problem: People and parcels sharing taxis. European J. Oper. Res. 238 (1), 31–40.

Li, X., Ouyang, Y., 2010. A continuum approximation approach to reliable facility location design under correlated probabilistic disruptions. Transp. Res. B 44 (4), 535–548.

Macrina, G., Pugliese, L.D.P., Guerriero, F., Laporte, G., 2020. Crowd-shipping with time windows and transshipment nodes. Comput. Oper. Res. 113, 104806.

Masoud, N., Jayakrishnan, R., 2017. A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system. Transp. Res. B 106, 218–236.

Mousavi, K., Bodur, M., Roorda, M.J., 2021. Stochastic last-mile delivery with crowd-shipping and mobile depots. Transp. Sci..

Nieto-Isaza, S., Fontaine, P., Minner, S., 2021. The value of stochastic crowd resources and strategic location of mini-depots for last-mile delivery: A benders decomposition approach.

Ouyang, Y., 2007. Design of vehicle routing zones for large-scale distribution systems. Transp. Res. B 41 (10), 1079–1093.

Pisinger, D., Ropke, S., 2010. Large neighborhood search. In: Handbook of Metaheuristics. Springer, pp. 399–419.

Pourrahmani, E., Jaller, M., 2021. Crowdshipping in last mile deliveries: Operational challenges and research opportunities. Socio-Econ. Plan. Sci. 101063.

Punel, A., Ermagun, A., Stathopoulos, A., 2019. Push and pull factors in adopting a crowdsourced delivery system. Transp. Res. Rec. 2673 (7), 529–540.

Raviv, T., Tenzer, E.Z., 2018. Crowd-shipping of small parcels in a physical internet.

Ropke, S., Cordeau, J.-F., 2009. Branch and cut and price for the pickup and delivery problem with time windows. Transp. Sci. 43 (3), 267–286.

Savelsbergh, M.W., Sol, M., 1995. The general pickup and delivery problem. Transp. Sci. 29 (1), 17–29.

Smilowitz, K.R., Daganzo, C.F., 2007. Continuum approximation techniques for the design of integrated package distribution systems. Netw.: Int. J. 50 (3), 183–196.

Soto Setzke, D., Pflügler, C., Schreieck, M., Fröhlich, S., Wiesche, M., Krcmar, H., 2017. Matching drivers and transportation requests in crowdsourced delivery systems. In: Twenty-Third Americas Conference on Information Systems, Boston.

Toth, P., Vigo, D., 2002. The Vehicle Routing Problem. SIAM.

Wandelt, S., Dai, W., Zhang, J., Sun, X., 2022. Toward a reference experimental benchmark for solving hub location problems. Transp. Sci. 56 (2), 543–564.

Wang, X., Agatz, N., Erera, A., 2018. Stable matching for dynamic ride-sharing systems. Transp. Sci. 52 (4), 850–867.

Wang, X., Ouyang, Y., 2013. A continuum approximation approach to competitive facility location design under facility disruption risks. Transp. Res. B 50, 90–103.
Wang, Y., Zhang, D., Liu, Q., Shen, F., Lee, L.H., 2016. Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. Transp. Res. E 93, 279–293.
Yan, C., Zhu, H., Korolko, N., Woodard, D., 2020. Dynamic pricing and matching in ride-hailing platforms. Nav. Res. Logist. 67 (8), 705–724.
Yatskiv, I., Budilovich, E., 2017. A comprehensive analysis of the planned multimodal public transportation HUB. Transp. Res. Procedia 24, 50–57.
Yıldız, B., 2021. Express package routing problem with occasional couriers. Transp. Res. C 123, 102994.
Yildiz, B., Savelsbergh, M., 2019. Service and capacity planning in crowd-sourced delivery. Transp. Res. C 100, 177–199.