EURO Journal on Transportation and Logistics 13 (2024) 100142

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

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

# Bundles generation and pricing in crowdshipping

Giusy Macrina [a,*], Claudia Archetti [b], Francesca Guerriero [a]

[a] Department of Mechanical Engineering University of Calabria, Italy
[b] Department of Information Systems, Decision Sciences and Statistics, ESSEC Business School in Paris, France

ARTICLE INFO

*Keywords:*
Crowdshipping
Bundles generation
Pricing
Auctions

ABSTRACT

Crowdshipping is a new delivery paradigm that exploits the capacity of ordinary people who offer their own vehicles and free time to perform deliveries against compensation. In this work, we consider a peer-to-peer logistic platform where a company receives orders from its customers and assigns them to occasional drivers (ODs), or crowdshippers, who perform the delivery operations. We first investigate the problem of deciding how the orders should be partitioned into bundles, where a bundle is a set of orders assigned to the same OD. Then, we focus on the problem of determining the compensation associated with each bundle, with the purpose of minimizing the total delivery costs. The pricing scheme is based on the assumption that each OD is associated with a *willingness-to-serve* function, which is modeled as a random variable that gives the probability that the OD accepts to deliver the bundle given the compensation value. This random variable captures the estimation of the *willingness-to-serve* function that the company has elaborated, for example on the basis of historical data. If the compensation offered by the company is greater than or equal to the *willingness-to-serve* value, the OD performs the delivery, otherwise she/he refuses. In case no OD is available to deliver a bundle, then all packages in the bundle are offered to a third-party delivery company. We simulate two auction systems for the assignment of bundles to ODs: a static and a dynamic auction. In exhaustive simulation tests, we compare different pricing schemes as well as the two auction systems, and outline several managerial insights.

## 1. Introduction

Crowdshipping is an innovative delivery strategy in which occasional drivers (*'crowdshippers'*) offer their service to delivery companies, typically using an online platform (Mckinnon, 2016; Punel and Stathopoulos, 2017; Buldeo Rai et al., 2018; Alnaggar et al., 2021). The platform, also named "peer-to-peer platform", typically operates as follows: the company uploads delivery requests, specifying the area, expected earnings, and task duration. Occasional drivers (ODs) browse these opportunities, selecting those that align with their convenience, and then communicate their availability to perform the service. The platform subsequently assigns the delivery task to an OD, who delivers the parcels and receives payment upon successful completion. Among the most successful crowdshipping platforms we mention *Amazon Flex*, the crowdshipping platform of Amazon, *My Ways* of DHL, *Roadie* of UPS, and *Postmate* of Uber.

This paradigm is gaining success in the last-mile delivery context. The main reason is that hiring an OD is much 'easier' than hiring a regular driver (RD), i.e., a driver with a permanent full contract. Indeed, an OD works temporarily for the company (and he/she is typically engaged just for the time slot associated with the delivery service he/she accepts to perform) and receives compensation for the service provided. RDs instead have a long-term contract and work full-time for the company (and thus receive a full-time salary). Thus, crowdshipping offers a great opportunity to deal with peak of demands by temporarily hiring ODs, avoiding to increase the 'regular delivery capacity' of the company (associated with RDs) which might then become useless when the peak is off. In most cases, third-party companies are engaged in case of missing capacity (neither ODs nor RDs are available to fulfill certain requests).

Implementing a crowdshipping system involves careful consideration of numerous factors. In particular, a critical key point is establishing appropriate compensation for each delivery offered to ODs. As stated in Buldeo Rai et al. (2018), remuneration is the most influential factor influencing willingness to work as a potential crowdshipper (45.36%). Therefore, defining the policy for assigning requests to ODs, both in terms of number of parcels and compensation, is crucial, and this is the focus of our work.

In this paper, we consider a crowdshipping platform that mimics real applications, like Amazon Flex, where customer requests are aggregated into so-called "blocks", which are then offered to the drivers. Blocks include the date, type, expected earnings, start time, and estimated duration. We first address the problem of bundles (blocks)

* Corresponding author.
*E-mail addresses:* giusy.macrina@unical.it (G. Macrina), archetti@essec.edu (C. Archetti), francesca.guerriero@unical.it (F. Guerriero).

https://doi.org/10.1016/j.ejtl.2024.100142
Received 18 April 2023; Received in revised form 24 June 2024; Accepted 5 August 2024
Available online 8 August 2024

generation, proposing a heuristic approach to generate them. Then, we propose a pricing scheme to determine the compensation associated with each bundle. Finally, we develop two auction strategies, a static and a dynamic auction, where bundles are offered to ODs and we simulate the corresponding acceptance.

The main focus of our study is to investigate the impact of pricing and auction strategies in a crowdshipping system and to derive insights that might be useful for applications. Indeed, we focus on decisions related to how bundles should be generated, what should be the corresponding compensation, and how to adjust the compensation in case of a dynamic auction.

The contributions of the paper can be summarized as follows:

- We study a crowdshipping system in which a company receives orders from customers, which are then offered to ODs for delivery. The company has to decide how to construct bundles of orders for the ODs and the compensation associated with each bundle. If no OD accepts to deliver a bundle, all parcels in the bundle are assigned to a third-party delivery company, incurring a higher cost. We assume that ODs accept to serve a bundle according to a *willingness-to-serve* function, which is modeled as a random variable giving the probability of acceptance with respect to the compensation offered.
- We propose a solution approach in which, in the first phase, we determine the bundles to be offered to the ODs. The corresponding compensation for each bundle is then determined in a second phase.
- We develop a bundle generation procedure based on a greedy algorithm, that takes into account both the temporal and spatial proximity of requests. The compensation is then determined according to a bundle pricing scheme, based on the *willingness-to-serve* function.
- We simulate two auction systems for the assignment of bundles to ODs. In the first system, which is called *"static"*, bundles are offered with a compensation that is determined as described above. If a bundle is not accepted, it is offered to the third company at a higher price. In the second auction system, called *"dynamic"*, the company can react to a "rejection" by increasing the compensation associated with a bundle and repeating the auction.
- We perform extensive simulations on synthetic instances with 100 and 1000 requests. We compare the static and the dynamic auctions, as well as two pricing strategies: the one previously mentioned and another where compensation is equal to the pure transport cost of serving the requests belonging to the bundle, estimated based on the distance traveled. We also run simulations on an instance generated on Rio de Janeiro's real road network, considering a case study of a large company that operates in last-mile delivery. We provide managerial insights, by comparing the pricing strategies and the auction systems.

The rest of the paper is organized as follows: in Section 2 we present a brief literature review on crowdshipping. In Sections 3 and 4 we describe the setting of the problem and the solution approach, respectively. In Section 5 we describe the auction systems. Section 6 is devoted to computational results and we finally outline the conclusions and future research perspectives in Section 7.

## 2. Literature review

Crowdshipping gained success not only in practical applications, but also in the scientific community, as academics have started to study the problem from several perspectives. Le et al. (2019) reviewed current practices, academic research, and several case studies ranging from supply and demand management to operations scheduling, while Archetti and Bertazzi (2021) focused on routing problems with crowdshipping. Pourrahmani and Jaller (2021) provided an overview of the operational characteristics of crowdshippping platforms and a comprehensive review of the literature, highlighting the challenges and the research opportunities.

In the following we review some of the recent contributions in crowdshipping, focusing on delivery applications. In particular, we consider the works addressing the three main problems arising when implementing a crowdshipping (or peer-to-peer logistics) platform: matching, routing, and pricing problems.

### 2.1. Matching in crowdshipping

In general, peer-to-peer logistics platforms use two different approaches to match suppliers (i.e., drivers) and requests (i.e., customers): centralized or decentralized (see, Mofidi and Pazour (2019)). In the centralized approach, decisions are taken by the company managing the platform. The platform proposes a request to a supplier and the supplier may accept or reject the proposed request. In the decentralized approach, the suppliers choose which requests they would like to serve.

Mofidi and Pazour (2019) proposed a hierarchical approach for the solution of the matching problem in peer-to-peer logistics platform. The platform first determines which requests should be recommended to each supplier, considering the estimation of the supplier's utility associated with each request. This estimation is based on historical data collected by the platform. Then, the suppliers may select which requests to fulfill. Each request can be offered to more than one supplier, and each supplier can accept or reject the offered request/s. The authors propose a bi-level approach for modeling the problem. Horner et al. (2021) extended the work of Mofidi and Pazour (2019) by considering a stochastic driver behavior and requests size larger than one. Ausseil et al. (2022) studied a dynamic matching, where the problem is modeled as a sequential decision process, with stochastic supply and demand. They proposed a multiple scenario approach, where the solutions of each scenario are combined using a consensus function to derive the final decision. Li et al. (2019) studied a decentralized system and proposed a multi-agent reinforcement learning approach.

### 2.2. Routing in crowdshipping

In this section we review the literature on contributions where the aim is to determine the routes to serve the customers, either performed by ODs or by regular drivers. No decision is taken about the compensation to be given to ODs, which is instead fixed, either as a constant amount or as depending on the distance traveled. The contributions where the compensation is part of the decision process are reviewed in the next section.

The literature related to crowdshipping in delivery operations is quite recent. Following the classification proposed by Boysen et al. (2022), we consider three crowdshipping paradigms: *customer-based*, *driver-based*, and *employee-based*.

The *customer-based* system mimics the Walmart concept, studied by Archetti et al. (2016), who introduced the vehicle routing with ODs (VRPOD). In this system, in-store customers may help the company in delivering parcels to online customers. Several authors extended the problem presented in Archetti et al. (2016), considering different features (see, e.g., Macrina et al. (2017), and Dahle et al. (2019)). Dahle et al. (2017) and Skålnes et al. (2020) considered an uncertain ODs' availability. Dayarian and Savelsbergh (2020) proposed a dynamic and stochastic routing problem in which not only ODs availability, but also customers' orders arrive over time.

The *driver-based* crowdshipping system is based on the Amazon Flex concept, studied by Macrina et al. (2020). In this case, ODs are not in-store customers, hence, they do not start their route from the depot (where parcels are picked up) but from different origins. Behrend et al. (2019) studied a system in which the supply and request of items, as well as crowdshipper availability, are all announced on the same

platform. Hence, they jointly addressed the assignment of supplies to requests and the routing of crowdshippers. Several authors studied different extensions of this problem considering stochastic/dynamic information (see. e.g., Archetti et al. (2021), Yıldız (2021), Torres et al. (2022b,a), and Di Puglia Pugliese et al. (2023)).

The last paradigm, named *employee-based* crowdshipping, was proposed and studied for the first time by Boysen et al. (2022). In this case, the employees drive their private cars after work to docking places for picking up some shipments, then to the customers for delivery.

Focusing on the evaluation of the compensation for the ODs, it can be noted that in all the contributions mentioned above the compensation scheme for the ODs is based on the detour or on a fixed amount per parcel. A few other contributions have, however, studied different compensation schemes. Sampaio et al. (2020) proposed a *driver-based* VRPOD with pickup and delivery, time windows and transfers. Since the time slot in which a driver is available could be shorter than the time required for the deliveries, the authors propose a more flexible service, in which a parcel that must be delivered can be transported by more than one driver from its origin to its destination. The compensation for the ODs is based on the time spent for the deliveries. Dai and Liu (2020) proposed a workforce capacity planning and allocation model for logistics systems where different types of workforces are available: in-house drivers, full-time crowdsourced drivers, and part-time crowdsourced drivers, which differ in their characteristics (e.g., compensation schemes and position). They considered a delivery cost for the full-time crowdsourced drivers composed of two elements: the fixed labor cost for the time periods they work and the number of parcels assigned. The compensation scheme for the part-time crowdsourced drivers depends on the number of orders assigned and not on the detour.

### 2.3. Pricing in crowdshipping

Among the factors that influence the crowdshippers behavior, the delivery compensation is a critical one. Some studies proposed different methods to determine the most attractive pricing strategy (see Le et al. (2019) for a review). Among the studies on routing with crowdshipping, only a few contributions focused on the price definition; indeed, as mentioned above, the majority of the contributions focusing on the routing problems with crowdshipping proposed a compensation based on the distance traveled.

Few studies considered the problem of combining the decisions about matching, pricing, and routing. Recently, Triki (2021) introduced the combinatorial auction technique within the VRPOD. In this framework, the ODs are the bidders that can submit any combination of different bids and they propose the corresponding price. The lower the submitted price, the higher the probability to obtain it. The author proposed a mathematical formulation as well as two heuristic approaches. In particular, a decomposition-based and a cost-comparison approach are proposed, in which the winning bid is selected based on the comparison between the bid's price and the cost of serving the same set of customers by the company's vehicles. The computational study highlighted the benefits, in terms of cost reduction, of using combinatorial auctions to assign the parcels to the ODs.

Mancini and Gansterer (2022) proposed a VRPOD with bundles. In their framework, the company first focuses on the generation of the bundles, and then bundles are offered to the ODs. In particular, bundles are generated following two strategies: a traditional clustering approach and a technique based on the generation of corridors. ODs make a bid for the bundles they consider as attractive. The bid depends on the OD detour, flexibility (maximum acceptable detour), and willingness to work (a parameter that can assume three values according to three levels of willingness: high, medium, and low). Bundles and bids are inputs of a mathematical model that assigns them to the ODs and creates also the routes for the regular drivers.

**Table 1**
Literature review: pricing in crowdshipping.

| Reference | Routing | ODs compensation | Bundles |
|---|---|---|---|
| Triki (2021) | Yes | Bids-based: random bids | Random |
| Mancini and Gansterer (2022) | Yes | Bids-based: ODs destination, flexibility, willingness | Clustering |
| Le et al. (2021) | No | Detour | No |
| This work | No | Tour, willingness | Route-based |

Le et al. (2021) proposed a framework that integrates the matching of requests and offers, together with pricing and compensation schemes. They first generated all the admissible matches couriers-requests, then apply the pricing strategies to determine the compensation. The authors evaluated four pricing and compensation schemes based on *flat* versus *individual* scheme settings. In the *flat* setting, the price and compensation are the same for all requests and delivery trips, while in the *individual* one, they are applied to each request and delivery trip, separately.

In the current paper, we focus on the bundling and the pricing problems. Similarly to Triki (2021), we propose an auction strategy to assign parcels to ODs. However, our framework differs significantly from the one proposed in Triki (2021). Indeed, in contrast to the latter one, where bidders propose and submit bids, in our approach the platform constructs the bundles and determines the compensations. Bundles are then offered to ODs, who decide whether to accept them or not. Our work also shares some similarities with Le et al. (2021), but the approach is different, as they do not consider an auction system. Finally, we differ also with respect to the work of Mancini and Gansterer (2022). First, we consider a platform where deliveries are performed through crowdshipping only. In Mancini and Gansterer (2022) there are both regular and ODs. They first focus on the bundles generation and assignment, and then the remaining customers are served by the company's drivers. Second, the definition of the compensation for the bundles is also different. In Mancini and Gansterer (2022) ODs make bids based on detour, flexibility, and a willingness to work, which is represented as a parameter. In our work, we assume to have no information about ODs destination, hence, the company evaluates a starting compensation based on the route that has to be traveled and on the *willingness-to-serve* function. Our *willingness-to-serve* function, differently from Mancini and Gansterer (2022), is represented as a random variable depending on the compensation, and not as a parameter multiplying the detour. In addition, in Mancini and Gansterer (2022) the authors propose a mathematical model, in which both the bundles and the bids offered by the ODs are input data. Then, the mathematical model finds the best assignment. Instead, in our work we propose a different strategy, where bundles are assigned to ODs through an auction. A similar strategy is proposed in Gansterer and Hartl (2018) and Gansterer et al. (2020), where a bundle generation problem in a collaborative transportation system is studied. However, the problem setting is completely different given that it arises in a collaboration framework where carriers can exchange transportation requests among each other, and decide which requests should be inserted in the auction pool. Then, the auctioneer generates the bundles of requests and offers them to the carriers. Carriers give their bids for the offered bundles, while the auctioneer allocates bundles to carriers based on their bids, hence the profits are distributed among the carriers.

In Table 1 we summarize the main features of the contributions revised in this section and compare them with the features of the system studied in the current paper.

## 3. Problem description and formulation

In this paper, we consider a peer-to-peer logistic platform. Fig. 1 provides a graphical illustration of how the platform works. Specifically, the platform receives the requests from the customers. Requests

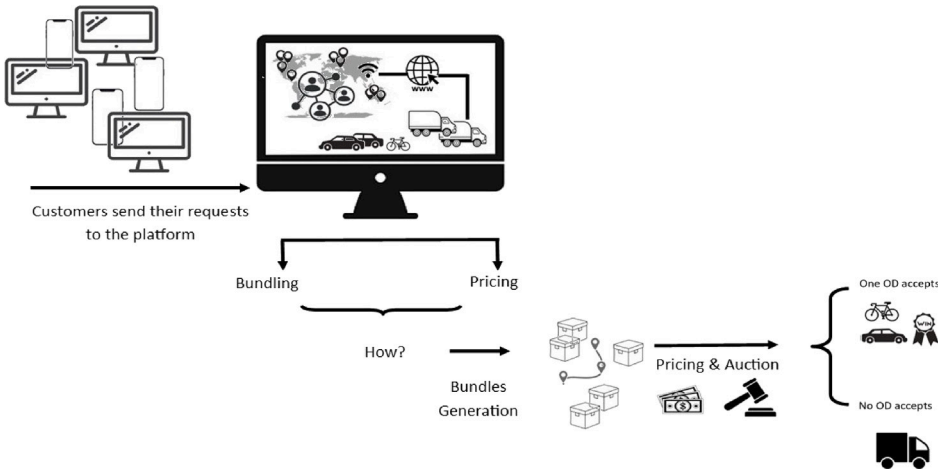



**Fig. 1.** Representation of the peer-to-peer logistic platform.

are then grouped into bundles and a compensation is associated with each bundle. Afterward, bundles are offered to ODs through an auction system. If a bundle is accepted by an OD, the OD receives all information needed to perform the delivery, namely the set of customers with their locations, the route, and the compensation. If no OD accepts to deliver a bundle, the delivery is performed by a third-party delivery company.

In particular, we suppose that customers place their requests the day before the planning. Customers may choose the time window in which they are available to receive their parcel, i.e., morning or afternoon, and must indicate the delivery location. Once the platform collects all customers' orders, the planning phase starts. The company faces two primary decisions: first, it has to determine how the requests should be clustered in bundles. Second, it has to decide what is the compensation associated with each bundle. Once bundles and compensations are set, the corresponding information is posted on the platform. We suppose an infinite population of ODs is available. We are aware that this is a strong assumption, which limits the applicability of the results of our study. However, our results can still serve as a basis for scenarios where this assumption is removed. Each OD accesses the platform and gives the availability for delivering a bundle at the corresponding compensation. Each bundle is assigned to the first OD who accepts to deliver it through an auction system. The compensation is paid for the delivery of all parcels in the bundle. If some bundles are not accepted by any ODs, then the corresponding parcels are delivered by a more expensive third-party delivery company. All delivery routes start from a common depot, where parcels are picked up.

We assume that the probability that an OD accepts to deliver parcels depends on the compensation offered. Each OD accepts to deliver a bundle according to her/his *willingness-to-serve* function, which corresponds to a random variable measuring the probability of acceptance on the basis of the compensation. In addition, we assume that ODs are not willing to make long detours to serve the parcels included in a bundle. Thus we limit the length of the route associated with each bundle to a maximum value $\bar{T}$. The objective of the company is to define the bundles of requests, and the corresponding compensation, in such a way as to minimize the expected total cost, which is given by the sum of expected compensation for ODs plus the expected costs associated with the third-party delivery service.

The problem can be formulated as a non-linear stochastic programming problem as follows. Let $P$ be the set of parcels to be served and $B$ be the set of all bundles that could be offered to ODs (thus, bundles associated with a route whose length does not exceed $\bar{T}$). Each bundle is associated with a binary variable $y_b$ which is equal to 1 in case the bundle is offered to ODs, 0 otherwise. Also, parameter $a_{pb}$ is equal to 1 in case parcel $p$ is included in bundle $b$. Each bundle is also associated with a second decision variable $c_b$ corresponding to the compensation offered. The problem can be modeled as:

$$min \sum_{b \in B} \mathbb{E}(c_b) y_b \tag{1a}$$

$$s.t.$$

$$\sum_{b \in B} a_{pb} y_b = 1, \qquad p \in P \tag{1b}$$

$$c_b \geq 0, \qquad b \in B \tag{1c}$$

$$y_b \in \{0, 1\}, \qquad b \in B \tag{1d}$$

where $\mathbb{E}(c_b)$ in (1a) denotes the expected cost of bundle $b$ when compensation $c_b$ is offered, given by the sum of $c_b$ times the probability of acceptance and the cost of the third-party delivery service times the probability of non-acceptance. Constraints (1b) guarantee that each parcel is included in one bundle only. Constraints (1c) and (1d) define the domain of the variables.

The drawback of formulation (1) is that the set $B$ is typically composed of exponentially many variables, namely, one variable for each feasible bundle. Thus, a branch-and-price approach is needed, where the relaxation of each node of the branch-and-bound tree is solved through column generation.

Given the general definition of the problem described above, we now present the specific assumptions we made in our study, based on which we propose the solution methodology illustrated in Section 4.

1. The probability that an OD accepts to serve a bundle is increasing with respect to the compensation offered.
2. ODs are identical, i.e., they have the same *willingness-to-serve* function. Thus, we are not interested in knowing which OD accepts the bundle, but simply in knowing whether there is at least one OD who accepts.
3. There is an infinite population of ODs, so the acceptance of a bundle has no impact on the probability of accepting the remaining bundles. As a consequence, ODs can be aggregated to determine the distribution function measuring the probability that at least one OD will accept the bundle. Note that, despite seeming simplistic, this assumption is in line with current trends in crowdshipping, where the explosion of crowdsourced drivers availability is experienced (see https://www.cnbc.com/2020/02/09/amazon-flex-drivers-use-bots-to-get-more-work.html). This phenomenon also leads to a homogeneous behavior related to the effort of acquiring as many deliveries services as possible, which corroborates assumption 2 on top, related to the homogeneity of ODs.

As a consequence of the above assumptions, and in particular of assumption 3, each bundle can be considered separately, as the price and the acceptance of a given bundle have no impact on the remaining bundles. This means that objective function (1a) is separable by bundles. Then, the expected cost associated with offering compensation $c_b$ for bundle $b$ is given by:

$$c_b pr_b + C_b(1 - pr_b) \tag{2}$$

where $pr_b$ is the probability that an OD accepts to serve bundle $b$ with compensation $c_b$ (given by the *willingness-to-serve* function) and $C_b$ is the cost (deterministic) of assigning all parcels in the bundle to a third-party delivery company.

In the following we present the solution approach we propose to solve the problem under the assumptions mentioned above. It corresponds to solving sequentially two phases aimed at determining:

1. How should the parcels be clustered into bundles?
2. What should be the compensation offered for each bundle?

## 4. Solution approach

The solution approach is composed of two main phases:

1. *Bundles generation*.
2. *Bundles pricing*.

The two phases are described in Sections 4.1 and 4.2, respectively. Bundles are constructed on the basis of spatio/temporal proximity, to make them attractive for ODs. Then, the compensation associated with each of them is calculated and finally, they are offered to ODs, through an auction system.

### 4.1. Bundles generation

In this section, we present the procedure that constructs the bundles to be offered to the ODs.

Given the assumption mentioned in Section 3, i.e., that ODs are available for a maximum time $\bar{T}$, then bundles are constructed in such a way that the route to serve all parcels in the bundle is no longer than $\bar{T}$.

To construct the routes, we use a greedy algorithm that works as follows. Starting from the depot $s$, it first adds the farthest customer (see, e.g., Caric and Gold (2008), Bräysy and Gendreau (2001) and Bräysy (2003)). Then, during each subsequent iteration, it adds to the route the first feasible (in terms of time windows) and nearest customer, until the maximum routing time $\bar{T}$ is reached. The procedure is iterated until all customers have been processed, i.e., they have been inserted in a bundle. Algorithm 1 provides a sketch of the approach, where $P$ is the set of customers (or parcels). The customers belonging to the same route compose a bundle.

**Algorithm 1:** Greedy $[P, s, \bar{T}]$

```
s ← depot;
while P is not empty do
    initialize a new route r;
    insert s in r;
    find the farthest customer i ∈ P from s;
    insert i in r;
    remove i from P;
    while the stopping criterion (T̄) is not met do
        find the nearest customer i′ ∈ P to i;
        insert i′ in r;
        remove i′ from P;
        i ← i′;
    insert the route r in the bundle set;
```

### 4.2. Bundles pricing

In this section, we describe how we determine the compensation associated with each bundle. A bundle is composed of a set of parcels that should be delivered by an OD.

The price associated with a bundle varies within a range defined by a minimum and a maximum compensation offered to ODs to deliver the parcels included. To fix this range, we consider the cost of the route visiting all customers in the bundle, starting from the depot. Then, we define a compensation factor that multiplies this cost to find the minimum and the maximum value of the range. The compensation offered is then determined within this range by taking into account the OD *willingness-to-serve* function.

We define as $d_b$ the cost of the route associated with bundle $b$, calculated as described above (by the greedy algorithm). Let $c'_b$ be the maximum compensation that is offered to an OD to serve parcels in bundle $b$ and $C_b$ be the price paid to the third-party delivery company to deliver all parcels in $b$. We need to determine, for each bundle $b \in B$, the compensation to offer to the ODs such as to minimize the expected total cost, which includes the compensation paid to ODs in case of acceptance and the price paid to the third-party delivery company in case of non-acceptance.

The expected cost is calculated with respect to the ODs *willingness-to-serve* function. In practice, the company does not know what compensation an OD will accept for delivering bundle $i$ (i.e., all parcels in the bundle). Thus, this *willingness-to-serve* function is represented as a random variable that gives the probability of acceptance with respect to the compensation offered. This function can be estimated through the analysis of historical data. Note that, because of the assumption above, all ODs are associated with the same *willingness-to-serve* function. If the compensation offered by the company is greater than or equal to the *willingness-to-serve* value, then the OD will accept to serve the bundle, otherwise, she/he will not. We assume that the *willingness-to-serve* function follows a Gaussian distribution $\mathcal{N}(\mu_b, \sigma_b)$ (note that the following model remains valid for any log-concave function), where $\mu_b$ is the expected value of the *willingness-to-serve* and $\sigma_b$ is the standard deviation. In the following, we denote as $f(b)$ the *willingness-to-serve* function associated with bundle $b$ and $F(b)$ the corresponding cumulative function (corresponding to $pr_b$ in (2)). A minimum compensation $\beta_b$ is defined, i.e., the compensation offered to ODs cannot be lower than $\beta_b$. We fix $\beta_b = d_b$. This is reasonable as the compensation should at least cover the cost of the corresponding route. The maximum compensation $c'_b$ is instead fixed to $\beta_b + \Delta_b$, where $\Delta_b$ is the maximum price, in addition to route cost, that the company is willing to pay for ODs' compensation. If no OD accepts the bundle, the price $p_b = (\beta_b + \Delta_b) * \zeta_b$ is paid to the third-party delivery company, with $\zeta_b > 1$.

The following analysis is based on determining an analytical formula for the optimal compensation $\eta_b$ that should be offered for each bundle $b$, i.e., the compensation that minimizes the expected cost. We note that a similar approach has been used in Yildiz and Savelsbergh (2019). However, the setting of the problem in the latter paper is different. Specifically, in Yildiz and Savelsbergh (2019), the authors consider a setting where the compensation is fixed at a flat rate and ODs are associated with a *willingness-to-wait* function, i.e., a function that determines how long they are willing to wait before being offered a service (and thus obtaining the flat rate compensation). The authors then make a theoretical study to determine the best radius for service assignment, taking into account the *willingness-to-wait*, and they derive an analytical formula, as we do in the following for determining the optimal compensation.

Let us call $\eta_b$ the compensation offered to serve bundle $b$. Given the assumptions above, the expected cost for the company associated with bundle $b$ is:

$$\mathbb{E}(\eta_b) = \eta_b F(\eta_b) + (1 - F(\eta_b))((\beta_b + \Delta_b) * \zeta_b). \tag{3}$$

Thus, the company should aim to determine:

$$\min_{\eta_b}\{\mathbb{E}(\eta_b)|\beta_b \le \eta_b \le \beta_b + \Delta_b\}. \tag{4}$$

Note that in (3), the first term, $\eta_b F(\eta_b)$, is monotonically increasing in $\eta_b$ while the second term, $(1 - F(\eta_b))((\beta_b + \Delta_b) * \zeta_b)$, is monotonically decreasing. Thus, we might expect some 'regularity' in the shape of function (3). This is indeed confirmed in the following proposition.

**Proposition 1.** *In the interval $\beta_b \le \eta_b \le \beta_b + \Delta_b$, function (3) is first convex and then concave. Thus, it changes convexity just once.*

**Proof.** The first derivative of (3) is

$$\begin{aligned}\mathbb{E}'(\eta_b) &= \eta_b f(\eta_b) + F(\eta_b) - ((\beta_b + \Delta_b) * \zeta_b) f(\eta_b)\\ &= F(\eta_b) + f(\eta_b)(\eta_b - (\beta_b + \Delta_b) * \zeta_b)\end{aligned}$$

and the second derivative is

$$\begin{aligned}\mathbb{E}''(\eta_b) &= 2f(\eta_b) + f'(\eta_b)(\eta_b - (\beta_b + \Delta_b) * \zeta_b)\\ &= f(\eta_b)(2 - \frac{\eta_b - \mu_b}{\sigma_b}(\eta_b - (\beta_b + \Delta_b) * \zeta_b)).\end{aligned} \tag{5}$$

As $f(\eta_b) > 0$ for each $\eta_b$, the sign of $\mathbb{E}''(\eta_b)$ depends on the second term of the product only. Now, $2 - \frac{\eta_b - \mu_b}{\sigma_b}(\eta_b - (\beta_b + \Delta_b) * \zeta_b)$ is quadratic in $\eta_b$ and its positive for $\frac{-\mu_b - (\beta_b+\Delta_b)*\zeta_b - \sqrt{(\mu_b+(\beta_b+\Delta_b)*\zeta_b)^2+8\sigma}}{2} \le \eta_b \le \frac{-\mu_b-(\beta_b+\Delta_b)*\zeta_b+\sqrt{(\mu_b+(\beta_b+\Delta_b)*\zeta_b)^2+8\sigma}}{2}$. However, $\frac{-\mu_b-(\beta_b+\Delta_b)*\zeta_b-\sqrt{(\mu_b+(\beta_b+\Delta_b)*\zeta_b)^2+8\sigma}}{2} < \beta_b$. Thus, in the interval considered, $\mathbb{E}(\eta_b)$ is convex in $[\beta_b, \min\{\beta_b + \Delta_b, \frac{-\mu_b-(\beta_b+\Delta_b)*\zeta_b+\sqrt{(-\mu_b-(\beta_b+\Delta_b)*\zeta_b)^2+8\sigma}}{2}\}]$ and concave in $[\min\{\beta_b + \Delta_b, \frac{-\mu_b-(\beta_b+\Delta_b)*\zeta_b+\sqrt{(-\mu_b-(\beta_b+\Delta_b)*\zeta_b)^2+8\sigma}}{2}\}, \beta_b + \Delta_b]$. □

This helps in searching for the value of $\eta_b$ providing the minimum expected cost in (4). Indeed, due to Proposition 1, this value is either at interval limits $\beta_b$ and $\beta_b + \Delta_b$ or at its unique local minimum in between.

In order to determine the value $\eta_b$ minimizing the expected cost (4), we proceed through a bisection method working as follows. We start by fixing the value of $\eta_b$ corresponding to $\mu_b$. We calculate the corresponding value of the cumulative distribution function $F(\eta_b)$. Then, we search for the minimum expected value of (4) by moving on the left and on the right on the cumulative function with a step equal to $\omega$. Specifically, we calculate the cost (4) associated with the corresponding value of the cumulative function. We terminate once we determine a local minimum, which, thanks to Proposition 1, approximates the value of the minimum of the function (4). A graphical example is given in Fig. 2 where the cumulative distribution function $F(\eta_b)$ is represented. To find the minimum expected value of (4), we first start by fixing $\eta_b = \mu_b$. We then move on the left and the right on the $F(\eta_b)$ function by using a step equal to $\omega$. We find the associated prices, i.e., $\eta_b^l$ and $\eta_b^r$. Then, we calculate the value of $E(\eta_b^l)$ and $E(\eta_b^r)$, respectively. In the example, we assume that $E(\eta_b^l) > E(\mu_b)$ hence, we stop the search on the left side. On the right side instead, we assume to have $E(\eta_b^r) < E(\mu_b)$ hence, we continue the search on $F(\eta_b)$ with a step equal to $2\omega$. We calculate the corresponding price $\eta_b^{2r}$ and then $E(\eta_b^{2r})$, we observe that $E(\eta_b^{2r}) > E(\eta_b^r)$, hence the search ends and the minimum value is set to $\eta_b^r$.

## 5. Auction systems

Once all bundles are priced according to the procedure described above, they are offered to ODs through the dedicated platform. In practice, the company offers the bundles on the platform, accessed by all ODs, where each bundle is associated with the corresponding compensation. When the bundles are available on the platform, the auctions starts and bundles are offered to ODs. All ODs have access to all bundles, meaning that they can see their composition and compensation. Then, for each bundle, the first OD who accepts it gets the corresponding compensation (once the service is performed). When the auction ends, the company determines which bundles have been accepted by ODs. The remaining will be served by the third-party delivery company. Given that each bundle is independent of the others (due to the assumptions described above), we now describe the auction system for each bundle individually.

We propose two auction mechanisms. The first one is static, i.e., the price is defined initially and the auction is composed of a single round. This means that the system works exactly as described above: the auction starts, the bundles are offered with the corresponding compensation, and when the auction terminates the company determines which ones have been accepted. For the accepted bundles, the corresponding compensation is paid to the ODs who accepted them once the service is completed. For the non-accepted bundles, the corresponding parcels are served by the third-party delivery company.

The second auction is a dynamic auction, specifically, an ascending-price auction. In this case, the auction is composed of multiple runs where, in each run, all bundles that have not been accepted in previous runs (thus, the entire set of bundles in the first run) are offered. Then, the accepted bundles are assigned to the corresponding ODs and removed from the auction. For the remaining bundles, the compensation is increased and they are offered again on the next run. When the auction reaches the last run, all parcels associated with bundles that have not been accepted are assigned to the third-party delivery company.

*Static auction.* The value of $\eta_b$ that minimizes the expected cost is evaluated using the bisection method described in Section 4.2. The static auction is a single round of bidding at this value of compensation.

*Dynamic auction.* Let us assume that the company opens the bundles' auction on day $d$, say from 6 to 8 pm, for the bundles that should be served on day $d + 1$. The company starts with an initial estimation of the ODs *willingness-to-serve* function as described above and determines the initial compensation through the bisection method, as above in the static auction. The idea of the dynamic auction is to have multiple runs of offers so that, in case the bundle is not accepted in a run, the company can react accordingly for the next run, for example by increasing the compensation offered. The auction starts in a first run by offering the same compensation determined in the static auction (which can be seen as a dynamic auction with one run only). The auction repeats at every runs on the bundles that were not assigned in the former runs.

There might be two reasons why the bundle $b$ has not yet been accepted (by any OD) at time $\tau_t$, where $t$ is the $t$th run:

1. the compensation $\eta_b$ is too low;
2. the distribution function $f(b)$ has been badly estimated.

As we assume to have no updated information to correct a bad estimation of the function $f(b)$, we focus on a strategy based on updating the value of $\eta_b$. Specifically, let $\eta_b^t$ be the compensation offered at run $t$. If the bundle $b$ is not assigned at $t < K$, then the corresponding compensation in the following run $\eta_b^{t+1}$ is increased by a certain percentage. In case $\eta_b^{t+1} \ge \beta_b + \Delta_b$, then its value is set to $\eta_b^{t+1} = \beta_b + \Delta_b$. This procedure is repeated until either the bundle is assigned or the auction terminates. If the auction terminates and bundle $b$ is not assigned, the parcels included are assigned to the third-party delivery company at a cost equal to $(\beta_b + \Delta_b) * \zeta_b$.

## 6. Computational results

In this section, we discuss the simulation experiments we performed in order to evaluate the pricing strategy and the auction mechanisms presented above. The main scope of this computational study is to

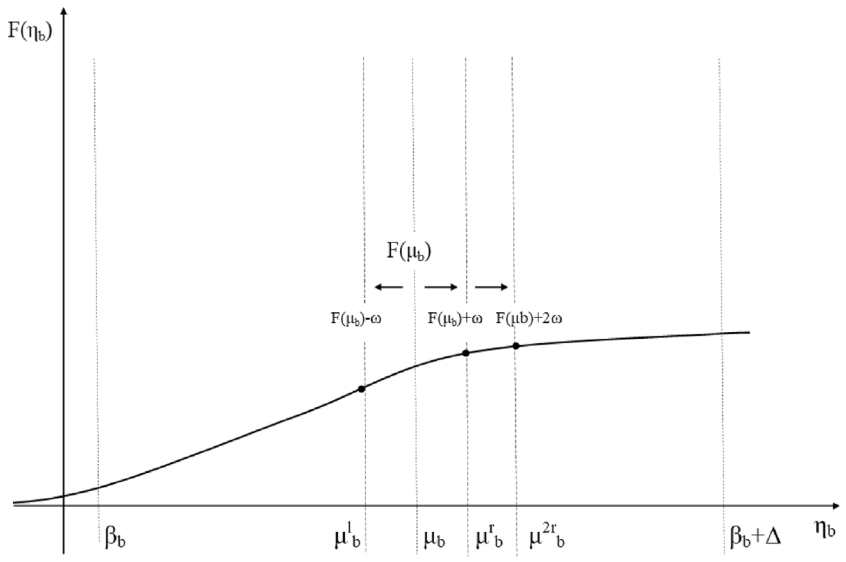



**Fig. 2.** Example of the bisection method used to find the minimum value of Eq. (4).

compare the following four strategies, in which we consider different approaches to calculate $\eta$, i.e., the compensation offered to the OD, and the two auction systems presented in Section 5:

1. *BasePriceStat*: the value of $\eta$ is equal to $\beta$, that is the minimum compensation that can be offered, through a static auction.
2. *OptPriceStat*: the value of $\eta$ is determined according to (4), that is the compensation that minimizes the expected cost, through a static auction.
3. *BasePriceDyn*: the value of $\eta$ is equal to $\beta$ through a dynamic auction.
4. *OptPriceDyn*: the value of $\eta$ is determined according to (4) through a dynamic auction.

The rest of the section is organized as follows. We first describe in Section 6.1 the settings of the simulation environment. Then, we focus on the analysis of the static and dynamic auctions in Sections 6.2 and 6.3, respectively. The two auction systems are compared in Section 6.4. In Section 6.5 we present a case study based on a real network.

### 6.1. Simulation environment and parameters setting

We generated our instances starting from Solomon's benchmark set of VRPTW instances (see Solomon (1987)). We considered two instances, namely R101 and RC101, with 100 customers which are randomly distributed in R101 and partially clustered in RC101. This allows us to verify if and how the results are affected by customers' locations. Then, we generated two additional instances with 1000 customers, by adding 900 random customers to R101 and RC101, respectively, in the same geometric area, obtaining more dense instances, referred to as "R1001" and "RC1001". We randomly generated the time windows for the additional customers in R1001 and RC1001, considering the time required to reach the customer from the depot, thus guaranteeing feasibility. We considered a planning horizon $[0, T_{max}]$, corresponding to a working shift. Then, we divided it into $m$ identical slots of duration equal to $T_{max}/m$. We set $m = 4$ in the experiments. This setting fits with the case where the horizon represents either the morning (i.e., 8:00 am–12:00 am) or the afternoon (i.e., 2:00 pm–6:00 pm) delivery shift, in which each slot is one hour long. In the considered instances, $T_{max}$ is equal to 230 and 240 for R101 and RC101, respectively. The values of $T_{max}$ were kept unchanged for the generated instances R1001 and RC1001. Each OD is available for two consecutive time slots, i.e., $\bar{T} = T_{max}/2$. This is in line with practical applications of crowdshipping where drivers are available for short time slots and thus can deliver a reduced number of parcels.

*Pricing parameters.* We fixed the value of the minimum compensation for each bundle $\beta_b$ equal to the cost of the associated greedy feasible route. The maximum cost $\beta_b + \Delta_b$ is fixed to $3 * \beta_b$. The price $C_b$ paid to the third-party delivery company is set to $(\beta_b + \Delta_b) * \zeta_b$, where $\zeta_b = 2$. We carried out a first set of simulations in which the expected value $\mu_b$ of $f(i)$ is centered in the interval associated with $[\beta_b, \beta_b + \Delta_b]$. Then, we considered three confidence intervals corresponding to the probabilities associated with $\beta_b$ and $\beta_b + \Delta_b$, respectively. In particular, the three considered intervals are: $[30\% - 70\%]$, $[15\% - 85\%]$ and $[5\% - 95\%]$. The idea is to assess the effect of a different variance of the *willingness-to-serve* function $f(b)$: the first interval corresponds to a higher variance while the last one is associated with the smallest variance. In the rest of the paper we will indicate these three confidence intervals as $\sigma_{low}$, $\sigma_{med}$ and $\sigma_{high}$, respectively. Note that in the first set of simulations the value of $\mu_b$ is equal to $((\beta_b + \Delta) + \beta_b)/2$, as it is in the center of the interval $[\beta_b, \beta_b + \Delta_b]$. We named this value of $\mu_b$ as $\mu_{med}$.

Fig. 3 represents the shape of the distribution when varying the variance $\sigma_b$. Looking at the figure we may observe that the higher the variance, the broader and lower the peak. Hence, a higher variance indicates that the points are more spread out from the mean, while when it is lower they are closer to it. Focusing on our setting, having a higher variance corresponds to a higher variability in the ODs' behavior, whereas a lower variance indicates a more homogeneous behavior.

We also conducted two additional sets of simulations where we varied the value of $\mu_b$. Specifically, we set:

1. $\mu_b = \mu_{med} + (20\%\mu_{med})$: the value of $\mu_b$ is shifted on the right. We denoted it as $\mu_{high}$;
2. $\mu_b = \mu_{med} - (20\%\mu_{med})$: the value of $\mu_b$ is shifted on the left. We denoted it as $\mu_{low}$.

As shown in Fig. 4, which represents the Gaussian distribution curve when varying the value of $\mu_b$, shifting $\mu_b$ to the left ($\mu_{low}$) and to the right ($\mu_{high}$) allows to represent different behaviors of the ODs' *willingness-to-serve*. Due to its symmetrical and bell-shaped nature, the probability of a Gaussian distribution has a maximum at its mean. Hence, when shifting $\mu_b$ to $\mu_{low}$, the probability that an OD accepts to deliver a bundle at a lower compensation is higher; on the contrary, when shifting to $\mu_{high}$, the ODs accept a delivery when the compensation offered is higher.

Note that, for these two additional sets of simulations, we kept the same three values of the variance of $f(b)$. As a consequence, the confidence intervals have the same width as above but are associated

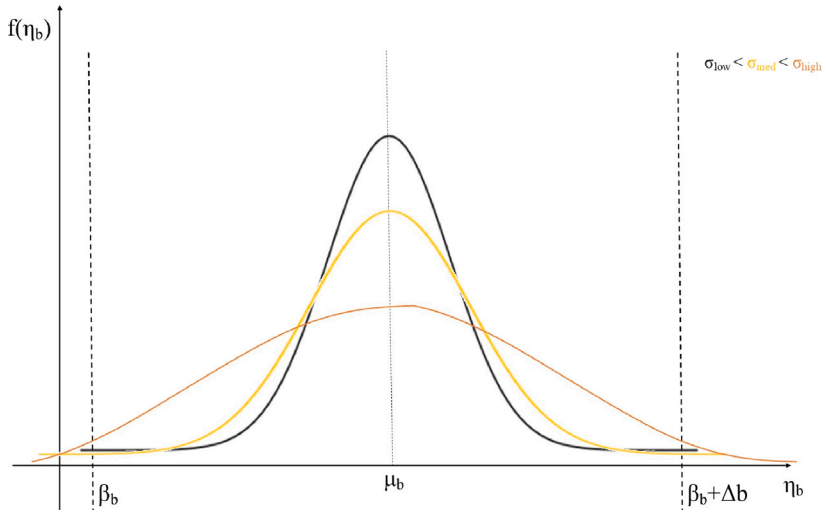



**Fig. 3.** Representation of the curves associated with the ODs *willingness-to-serve* function when varying the variance $\sigma_b$. In particular, $\sigma_{low} < \sigma_{med} < \sigma_{high}$.

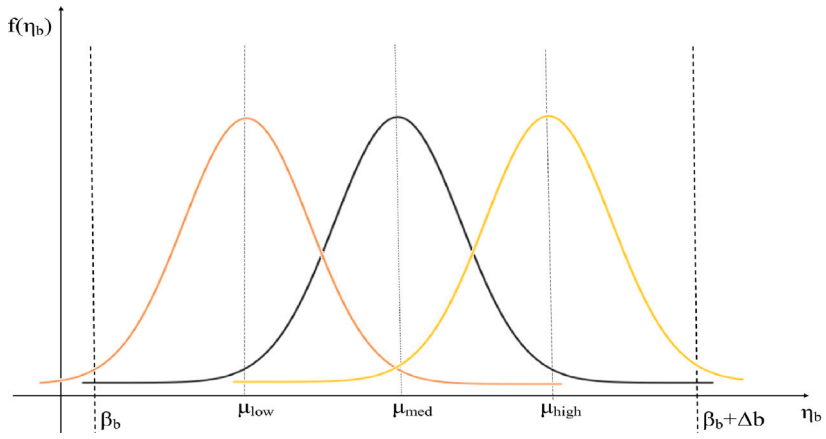



**Fig. 4.** Representation of the curves associated with the ODs *willingness-to-serve* function when shifting $\mu_b$ to the left and right.

with different lower and upper limits. For the sake of completeness, we report a graphical representation of the nine settings we have considered, depicted in Fig. 5, obtained by combining the different values of $\mu_b$ and $\sigma_b$.

As stated at the beginning of this section, we compare two pricing strategies that differ on how the value of the compensation $\eta_b$ is set. In the first one, we set $\eta_b = \beta_b$, while in the second $\eta_b$ corresponds to the value that minimizes the expected cost (4) and is calculated as described in Section 5. Thus, we compare a strategy in which we reduce the compensation value to the minimum ($\eta_b = \beta_b$) but we increase the probability of not assigning the bundle, with the strategy in which we minimize the expected cost through (4). In the latter case, the value of parameter $\omega$ used in the bisection method is fixed at 0.005. Note that we made preliminary tests by using a step equal to 0.05. The results were similar but less accurate. By setting the step equal to 0.005 the simulation is still fast (always runs in milliseconds), so we decided to keep the latter to gain accuracy.

As for the dynamic auction, the starting value of the compensation $\eta_b$ is determined with either of the two pricing strategies mentioned above (thus giving rise to two dynamic pricing auctions).

*Static auction.* We simulated 5000 realizations of the *willingness-to-serve* value for each bundle, according to the Gaussian distribution. If the realization value is equal to or lower than the value of $\eta_b$, the bundle is assigned to the OD, otherwise, the bundle is not assigned and its cost is equal to $(\beta_b + \Delta_b) * \zeta_b$.

*Dynamic auction.* The setting is the same as for the static auction. In addition, in the dynamic auction, up to 8 runs are performed. Thus, in each run, either the bundle is assigned, or the compensation is updated according to the rule: $\eta_b^t = \eta_b^0 * (1 + 0.05t)$, where $\eta_b^0$ is the initial compensation. Thus, the compensation is increased by 5% of the initial compensation in each run. If after 8 runs the bundle is still unassigned, then the cost $(\beta_b + \Delta_b) * \zeta_b$ is paid.

For ease of reading, in the following we remove index $b$ from all notations as we present aggregated results over all bundles.

In the next sections, we analyze the results obtained by the static and dynamic auctions, respectively. In both cases, bundles are generated according to the approach described in Section 4.1. In the following, we concentrate on the KPI for both auctions. Some statistics about the generated bundles can be found in Appendix A.1, whereas Table 2 reports the parameter settings.

### 6.2. Static auction analysis

In this section we present the results associated with static auction system, considering the two pricing strategies, namely, *BasePriceStat* and *OptPriceStat*.

We focus the analysis on the instances with 1000 requests only, as the results for instances with 100 requests are similar. We firstly analyze the percentage of unassigned bundles, on the basis of the values of $\mu$ and $\sigma$, whose trend, for both strategies, is depicted in Fig. 6. Focusing on Fig. 6(a), which depicts the results obtained using

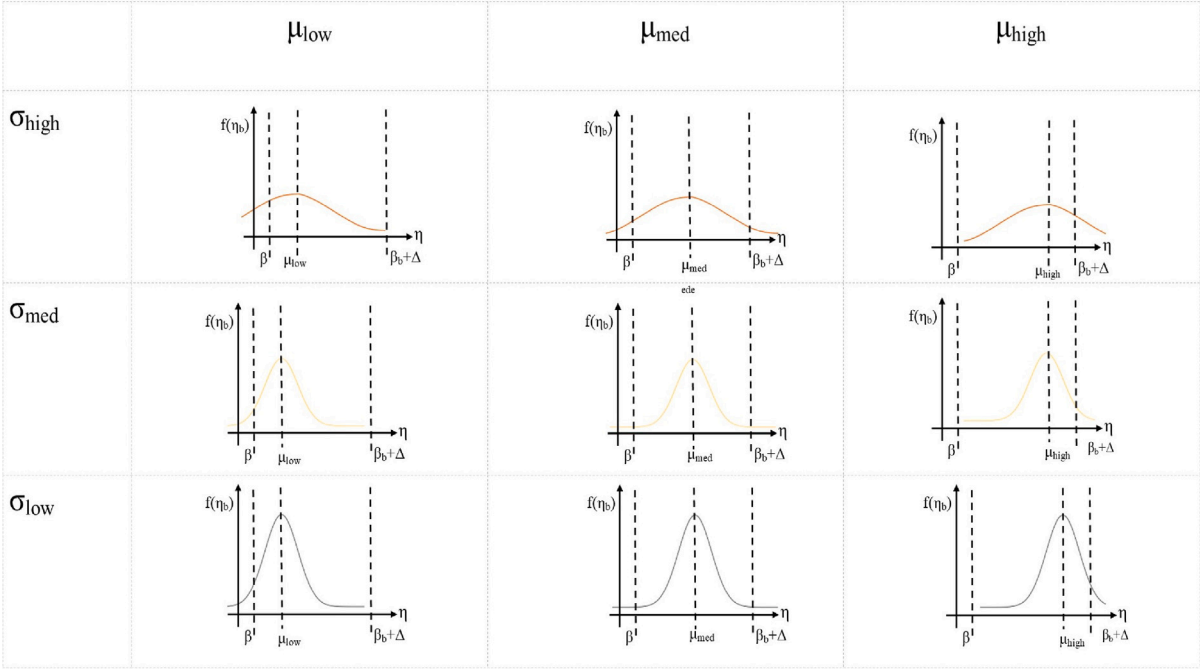



**Fig. 5.** Representation of the shapes of the ODs *willingness-to-serve* function considered in the computational study.

**Table 2**
Parameters setting.

| Pricing parameters | |
|---|---|
| Parameter | Value |
| $\beta_b$ | Greedy cost (Appendix A.1, Table A.1) |
| $\Delta_b$ | $2^*\beta_b$ |
| $\zeta_b$ | 2 |
| **Bisection method** | |
| Parameter | Value |
| $\omega$ | 0.005 |
| **Static auction** | |
| Parameter | Value |
| $\#realizations$ | 5000 |
| **Dynamic auction** | |
| Parameter | Value |
| $\#realizations$ | 5000 |
| $\#runs$ | 8 |
| $\eta_b^t$ | $\eta_b^0 * (1 + 0.05t)$ |

**Table 3**
Comparison between *BasePriceStat* and *OptPriceStat*.

| $\mu$ | $\sigma$ | $GapUnassigned$ (%) | $GapAuctionCost$ (%) |
|---|---|---|---|
| $\mu_{low}$ | $\sigma_{high}$ | 44.8% | 5.0% |
| | $\sigma_{med}$ | 121.7% | 27.6% |
| | $\sigma_{low}$ | 78.9% | 34.8% |
| $\mu_{med}$ | $\sigma_{high}$ | 68.0% | 15.6% |
| | $\sigma_{med}$ | 180.1% | 47.6% |
| | $\sigma_{low}$ | 123.3% | 52.7% |
| $\mu_{high}$ | $\sigma_{high}$ | 113.9% | 35.1% |
| | $\sigma_{med}$ | 226.7% | 61.1% |
| | $\sigma_{low}$ | 135.2% | 51.1% |

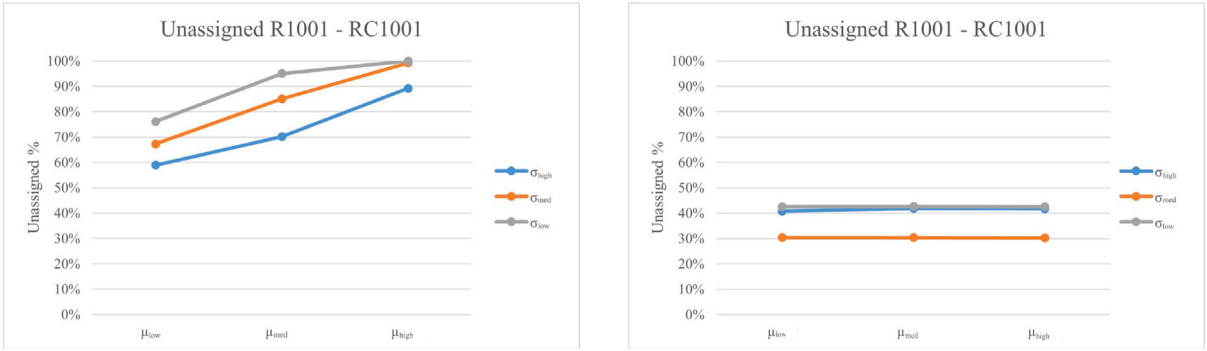
*BasePriceStat*, we can observe that moving $\mu$ from left to right increases the percentage of unassigned bundles, and the same happens when decreasing the variance. This is easily explained by the fact that, in both cases, the cumulative probability associated with the value of $\beta$ (which corresponds to the compensation offered in *BasePriceStat*) decreases. Also, we notice that the percentage of unassigned bundles is rather high. This is even more evident when comparing Figs. 6(a) to 6(b), which depicts the results obtained using *OptPriceStat*. Focusing on Fig. 6(b), we can observe that the percentage remains stable over the different values of $\mu$ and changes according to $\sigma$, with an average of 41%, 30% and 42% for $\sigma_{high}$, $\sigma_{med}$, and $\sigma_{low}$, respectively. This is consistent with what is reported in Tables A.2 and A.3 of Appendix A.2 concerning the cumulative probability associated with different values of $\eta$. Thus, we can conclude that *OptPriceStat* is more stable than *BasePriceStat* with respect to ODs' behavior in terms of unassigned bundles.

Then, we focus on the auction cost. As done for the unassigned bundles, Fig. 7 depicts the trend by varying $\mu$ and $\sigma$. For $BasePriceStat$, focusing on Fig. 7(a), we see that the observed trend is similar to that observed for the unassigned bundles, i.e., the cost increases with respect to $\mu$ and decreases with respect to $\sigma$. This is due to the fact that the number of unassigned bundles increases and, thus, the cost associated with paying the third-party delivery company increases as well. $OptPriceStat$ (Fig. 7(b)) is instead more stable and, most importantly, the cost is always lower than $BasePriceStat$.

We now compare the two strategies in more detail by measuring the gap between the solutions obtained, in terms of both unassigned bundles and total auction cost. Results are summarized in Table 3, which reports the following two statistics, in percentage:

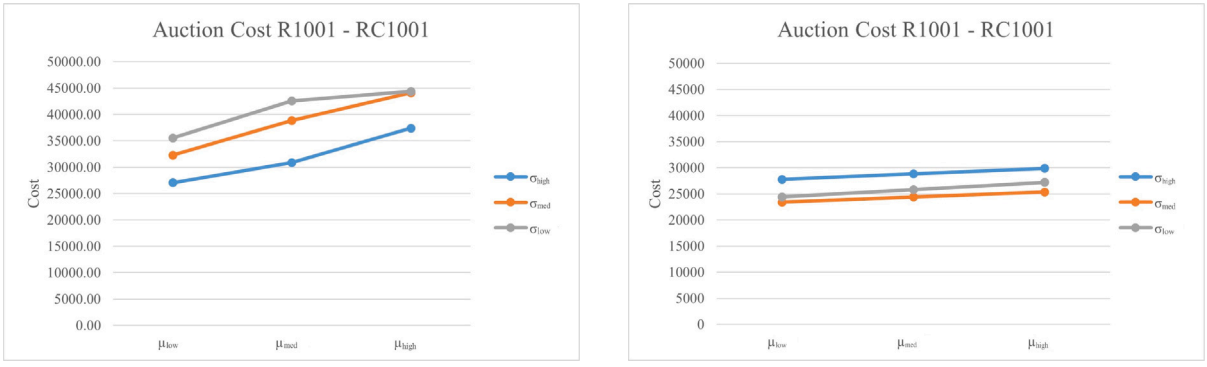
- $GapUnassigned$ measured as $\frac{U(BasePriceStat)-U(OptPriceStat)}{U(OptPriceStat)}$, where $U(\cdot)$ is the number of unassigned bundles for strategy $\cdot$,
- $GapAuctionCost$ measured as $\frac{AC(BasePriceStat)-AC(OptPriceStat)}{AC(OptPriceStat)}$, where $AC(\cdot)$ is the auction cost for strategy $\cdot$.

We notice that the percentage of unassigned bundles is much larger in *BasePriceStat*, being sometimes even more than 3 times larger than the one of *OptPriceStat* (when $\sigma = \sigma_{med}$ and $\mu = \mu_{high}$), and on average it is two times larger. Focusing on cost, we observe that the auction cost is smaller for *OptPriceStat* and the gap increases with increasing values of $\sigma$ and $\mu$. Overall, *OptPriceStat* allows to assign a higher number of bundles at a lower cost.




(a) *BasePriceStat* (b) *OptPriceStat*

**Fig. 6.** Percentage of unassigned bundles for R1001 and RC1001 instances by varying $\sigma$ and $\mu$ in the static auction system.




(a) *BasePriceStat* (b) *OptPriceStat*

**Fig. 7.** Auction cost for R1001 and RC1001 instances by varying $\sigma$ and $\mu$ in the static auction system.

Thus, the following conclusions hold concerning the comparison of the two pricing strategies when using a static auction:

- *OptPriceStat* is less sensitive than *BasePriceStat* to the mean and the variance of the *willingness-to-serve* function, in terms of both unassigned bundles and auction cost.
- *BasePriceStat* might be too risky in terms of the number of unassigned bundles. Indeed, this percentage reaches almost 100% in some cases. This might clearly have a negative impact on the customer service. Also, the risk of missed service increases, as the third-party delivery company might not be able to serve all unassigned bundles.
- In all scenarios, *OptPriceStat* outperforms *BasePriceStat* in terms of both the number of assigned requests and cost.

### 6.3. Dynamic auction analysis

We now move to the analysis of the dynamic auction, hence focusing on the *BasePriceDyn* and *OptPriceDyn* strategies. As done above, we consider instances with 1000 requests as the results on the instances with 100 requests are similar. As previously mentioned, in the dynamic auction, up to 8 runs are performed where, each time a run is unsuccessful (i.e., the bundle is not assigned), the compensation is updated. Fig. 8, depicts the trend of unassigned bundles by varying $\sigma$ and $\mu$ using *BasePriceDyn*, considering the 8 runs. Looking at Fig. 8 it is clear that the trend observed for the static auction is maintained. When we consider a low variance, we expect that the majority of the ODs accept to deliver a bundle if the compensation offered is close to the mean value. In this case, using *BasePriceDyn* is not convenient, since the compensation offered is too low, resulting in a high number of unassigned bundles. We do not report the statistic related to unassigned bundles for *OptPriceDyn* as this strategy nearly always assigns all bundles, with very few exceptions. This leads to an average of 0% unassigned bundles across all cases, calculated to the second decimal place.

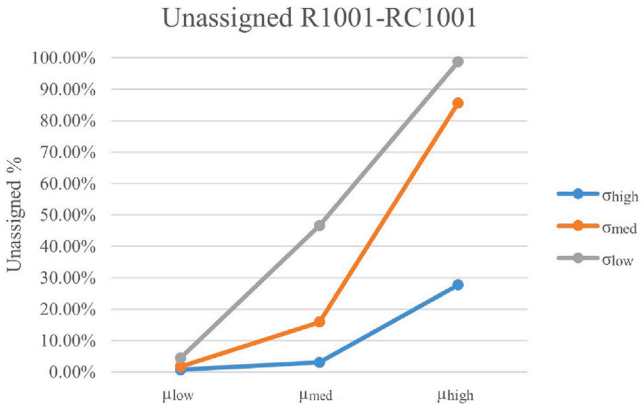
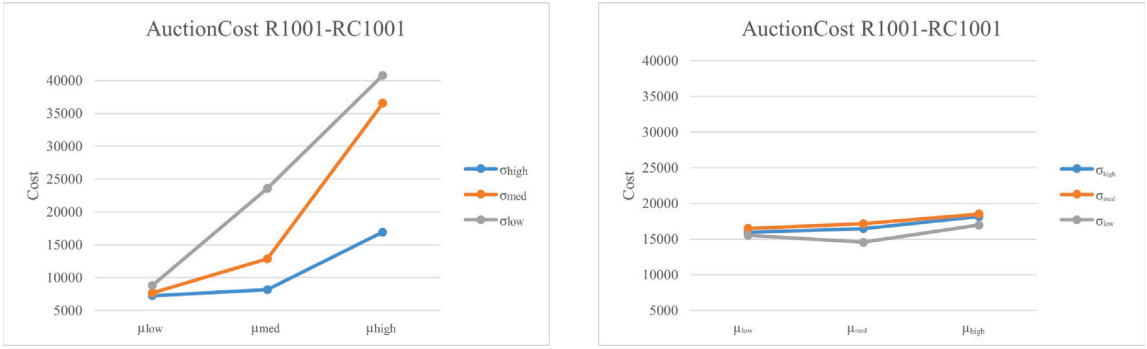



**Fig. 8.** Average percentage of unassigned bundles for R1001 and RC1001 instances by varying $\sigma$ and $\mu$ and using *BasePriceDyn*.

In terms of auction costs, results are illustrated in Fig. 9. The trends for $BasePriceDyn$ are shown in Fig. 9(a) and those for $OptPriceDyn$ are depicted in Fig. 9(b). Looking at Fig. 9(a) we may observe a similar trend to that obtained for $BasePriceStat$. Indeed, moving $\mu$ from left to right increases the auction cost. Furthermore, the variance has a significant impact, with the auction cost increasing as the variance decreases. This is consistent with the trend of unassigned bundles, in fact, the higher the number of unassigned bundles, the higher the auction costs. Focusing on $OptPriceDyn$ in Fig. 9(b), we observe that the cost variation is not significant. We highlight a slight increase in




(a) *BasePriceDyn* (b) *OptPriceDyn*

**Fig. 9.** Auction cost for R1001 and RC1001 instances by varying $\sigma$ and $\mu$ in the dynamic auction system.

**Table 4**
Comparison between *BasePriceDyn* and *OptPriceDyn* in terms of auction cost.

| $\mu$ | $\sigma$ | *GapAuctionCost* (%) |
|---|---|---|
| $\mu_{low}$ | $\sigma_{high}$ | −54.6% |
| | $\sigma_{med}$ | −53.6% |
| | $\sigma_{low}$ | −33.6% |
| $\mu_{med}$ | $\sigma_{high}$ | −50.5% |
| | $\sigma_{med}$ | −24.9% |
| | $\sigma_{low}$ | 62.0% |
| $\mu_{high}$ | $\sigma_{high}$ | −6.6% |
| | $\sigma_{med}$ | 97.1% |
| | $\sigma_{low}$ | 139.7% |

cost when using $\sigma_{med}$. This is an expected result since we have shown that when using $\sigma_{med}$ the final value of $\eta$ is higher with respect to the ones obtained using $\sigma_{low}$ and $\sigma_{high}$ (see Appendix A.2).

Comparing Figs. 9(a) and 9(b), we highlight that the auction cost for $BasePriceDyn$ strongly depends on the ODs behavior. However, when considering $\sigma_{high}, \sigma_{med}$ and $\mu_{med}, \mu_{low}$, it outperforms $OptPriceDyn$ in terms of cost. This is confirmed also by the results summarized in Table 4, in which we report the *GapAuctionCost*, calculated as in the former section (see Table 3). We do not report $GapUnassigned$ as $OptPriceDyn$ assigns always almost all bundles, as mentioned above. The results highlight that *OptPriceDyn* performs better when the mean $\mu$ is shifted on the right, and, in general, when the variance $\sigma$ is low; in other words, when the ODs require in general a relatively high compensation and their willingness-to-serve is not sparse.

In conclusion, the results of the comparison between the two pricing strategies when using a dynamic auction can be summarized as follows:

- *OptPriceDyn* assigns almost all bundles in any scenario related to ODs behavior, while the number of unassigned bundles is largely impacted by ODs behavior in *BasePriceDyn*.
- As for the static auction, *BasePriceDyn* might be too risky in terms of number of unassigned bundles, particularly in specific scenarios where this number reaches 90%.
- In terms of costs, *BasePriceDyn* outperforms *OptPriceDyn* when the number of unassigned bundles for *BasePriceDyn* is lower. The overall cost of the auction is reduced due to the lower compensation offered to the ODs. This occurs when the mean of the *willingness-to-serve* is shifted to the left and when the variance is high, resulting in the expected value of the *willingness-to-serve* being positioned close to the minimum compensation that can be offered to the ODs.
- Overall, as observed in the static auction, *OptPriceDyn* is more stable and less sensitive to ODs behavior.

**Table 5**
Percentage of unassigned bundles when varying the number of runs, $\sigma$, and $\mu$ for both $BasePrice$ and $OptPrice$.

| | $\sigma$ | *Unassigned (%) BasePrice* | | | *Unassigned (%) OptPrice* | | |
|---|---|---|---|---|---|---|---|
| | | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ |
| 1 run | $\sigma_{high}$ | 58.9% | 70.2% | 89.2% | 40.7% | 41.8% | 41.7% |
| | $\sigma_{med}$ | 62.7% | 85.1% | 99.2% | 28.3% | 28.3% | 28.3% |
| | $\sigma_{low}$ | 76.1% | 95.0% | 100.0% | 39.7% | 39.7% | 39.7% |
| | $\sigma$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ |
| 2 runs | $\sigma_{high}$ | 20.3% | 34.3% | 70.4% | 7.2% | 7.6% | 7.6% |
| | $\sigma_{med}$ | 29.8% | 60.9% | 97.6% | 2.9% | 2.9% | 2.8% |
| | $\sigma_{low}$ | 43.1% | 85.2% | 100.0% | 7.9% | 7.8% | 7.4% |
| | $\sigma$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ |
| 4 runs | $\sigma_{high}$ | 6.8% | 16.4% | 54.7% | 1.1% | 1.2% | 1.2% |
| | $\sigma_{med}$ | 12.8% | 42.5% | 95.6% | 0.3% | 0.2% | 0.2% |
| | $\sigma_{low}$ | 23.4% | 75.2% | 100.0% | 1.2% | 1.2% | 1.0% |
| | $\sigma$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ |
| 8 runs | $\sigma_{high}$ | 0.7% | 3.0% | 27.7% | 0.0% | 0.0% | 0.0% |
| | $\sigma_{med}$ | 1.7% | 15.9% | 85.7% | 0.0% | 0.0% | 0.0% |
| | $\sigma_{low}$ | 4.5% | 46.5% | 98.8% | 0.0% | 0.0% | 0.0% |

### 6.4. Static vs. dynamic auction analysis

While in the former two sections, we focused on the two auctions separately and compared the behavior of the two pricing strategies in each, we now compare the two auction systems. We point out that a dynamic auction with a single run reduces to the static auction.

Table 5 shows the results of the comparison between the static and the dynamic auctions for both $BasePrice$ and $OptPrice$, in terms of the percentage of unassigned bundles. Static auction corresponds to having one run. Focusing on $BasePrice$ we can see that the dynamic auction allows a higher number of bundles to be assigned than the static auction. The dynamic auction mechanism is extremely efficient in improving the system performance: the overall percentage of unassigned bundles decreases from 82% to 24% with respect to the static auction. More specifically, we observe that:

- Increasing the number of runs has a large impact on the number of unassigned bundles in all scenarios except the worst case, which is the one where $\mu = \mu_{high}$ and $\sigma = \sigma_{low}$.
- Still, the percentage of unassigned bundles remains rather high even after 8 runs for $\mu = \mu_{high}, \mu_{high}$ and $\sigma = \sigma_{med}, \sigma_{low}$.

Focusing on the $OptPrice$, the number of unassigned bundles decreases with the increase in the number of runs. As already mentioned, $OptPriceDyn$ is able to assign almost all bundles.

Fig. 10 shows the average trend of the auction costs when varying the number of runs, for both the $BasePriceDyn$ and the $OptPriceDyn$.

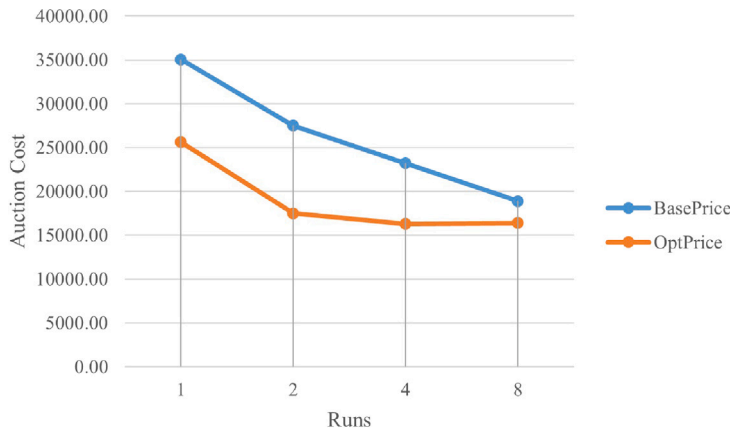



**Fig. 10.** Average auction cost: comparison between static (1 run) and dynamic auctions with 2, 4, and 8 runs, for *BasePrice* and *OptPrice*.

Looking at Fig. 10 we observe a decrease of the gap in the auction cost between the two strategies, with the increase in the number of runs. On average $OptPriceDyn$ outperforms $BasePriceDyn$, however, the gap tends to decrease as the number of runs increases.

The comparison between the static and the dynamic auction results, averaged on $\mu$ and $\sigma$, clearly shows that the dynamic auction outperforms the static one. In particular:

- The number of unassigned bundles decreases significantly from approximately 80% to 60% when comparing one run to two runs using $BasePriceDyn$, and from about 40% to about 6% when using $OptPriceDyn$. Consequently, the dynamic auction for both pricing strategies results in a higher number of bundles being assigned in a few runs.
- Overall, the costs are reduced of about 45% and 36% after 8 runs, for $BasePriceDyn$ and $OptPriceDyn$, respectively.
- If it is not possible to perform more than one run, $OptPriceStat$ outperforms $BaseOptStat$ in terms of both the number of unassigned bundles and cost.

### 6.5. Real network case study

We tested our approach on a real network, using one of the Capacitated VRP (CVRP) urban deliveries dataset provided in Loggi (2021). This dataset simulates the case of a large delivery company that operates in last-mile delivery in Rio de Janeiro. Customers are unevenly distributed over the area. Distances among customer locations are calculated considering the real road network distances. In particular, we focused on Loggi-n601-k42, an instance composed of 600 customers and one depot, depicted in Fig. 11. To adapt the CVRP instance to our problem, we generated the work shift and the time windows for the customers. Starting from the optimal solution provided by the DIMACS repository at http://dimacs.rutgers.edu, we considered the longest route and set $T_{max}$ to the value of its duration; i.e., $T_{max} = 19102$. Then, we divided the planning horizon into $m = 4$ intervals of identical duration, as in the Solomon's-based benchmark instances, and we generated the time windows for the customers according to these four intervals. We calculated the distance from the depot, then we randomly assigned a compatible and feasible period to each customer. Each OD is available for two consecutive slots, i.e., $\bar{T} = 9551$.

We then constructed the bundles according to the greedy algorithm presented in Section 4.1 (results are summarized in Appendix B.1) and performed several computational tests, considering the strategies presented in Section 6, i.e., *BasePriceStat*, *OptPriceStat*, *BasePriceDyn* and *OptPriceDyn*.

**Table 6**
Comparison between $BasePriceStat$ and $OptPriceStat$ for the static auction on the Loggy-n601-k42 instance.

| $\mu$ | $\sigma$ | $GapUnassigned$ (%) | $GapAuctionCost$ (%) |
|---|---|---|---|
| $\mu_{low}$ | $\sigma_{high}$ | 44.6% | 4.9% |
| | $\sigma_{med}$ | 122.0% | 27.7% |
| | $\sigma_{low}$ | 79.2% | 34.9% |
| $\mu_{med}$ | $\sigma_{high}$ | 67.8% | 15.6% |
| | $\sigma_{med}$ | 181.3% | 47.8% |
| | $\sigma_{low}$ | 123.5% | 52.7% |
| $\mu_{high}$ | $\sigma_{high}$ | 113.8% | 35.1% |
| | $\sigma_{med}$ | 228.8% | 61.4% |
| | $\sigma_{low}$ | 134.8% | 51.0% |

*Static auction analysis.* Fig. 12 depicts the trend of the percentage of unassigned bundles for $BasePriceStat$ and $OptPriceStat$. The observed trends in Section 6.2 are confirmed. For $BasePriceStat$, see Fig. 12(a), moving $\mu$ from left to right increases the percentage of unassigned bundles. The value of $\sigma$ also affects the number of unassigned bundles. In fact, the lower the value of $\sigma$, the higher the number of unassigned bundles. Fig. 12(b) depicts the trend for $OptPriceStat$ and we observe that the percentage remains stable over the different values of $\mu$. Overall, $OptPriceStat$ assigns more bundles regardless of the scenario.

Focusing on the auction cost, whose trend is depicted in Fig. 13, it can be observed that the trend is consistent with the findings presented in Section 6.2. Fig. 7(a) illustrates that the auction cost follows a similar trend to the one observed for the unassigned bundles for $BasePriceStat$. Specifically, the auction cost increases with increasing values of $\mu$ and decreasing $\sigma$, whereas for $OptPriceStat$ the auction cost is more stable.

To conclude, we report in Table 6 the percentage values of $GapUnassigned$ and $GapAuctionCost$ for the static auction. Looking at Table 6, it is clear that in the case of a static auction, the best strategy is $OptPriceStat$.

*Dynamic auction analysis.* Fig. 14 shows the trend of unassigned bundles for *BasePriceDyn*. We do not report the trend for *OptPriceDyn* since it assigns almost all bundles. The trend depicted in Fig. 14 is similar to the one observed for the static auction. However, the increase is sharpened when moving $\mu$ from the left to the right. When considering $\mu_{low}$, the value of the compensation offered to the ODs is closest to the peak of the bell of their *willingness-to-serve* function, therefore carrying out several runs will allow to increase the compensation and thus the probability of acceptance. On the contrary, when considering $\mu_{high}$ the peak of the bell is significantly distant from the initial compensation. Consequently, even with multiple runs, the increase in compensation is insufficient to fit the behavior of the ODs. This results in a large number of unassigned bundles, as observed for the static auction.

Now we move on to the analysis of the auction cost, whose trend is depicted in Fig. 15. If we focus on *BasePriceDyn*, reported in Fig. 15(a),

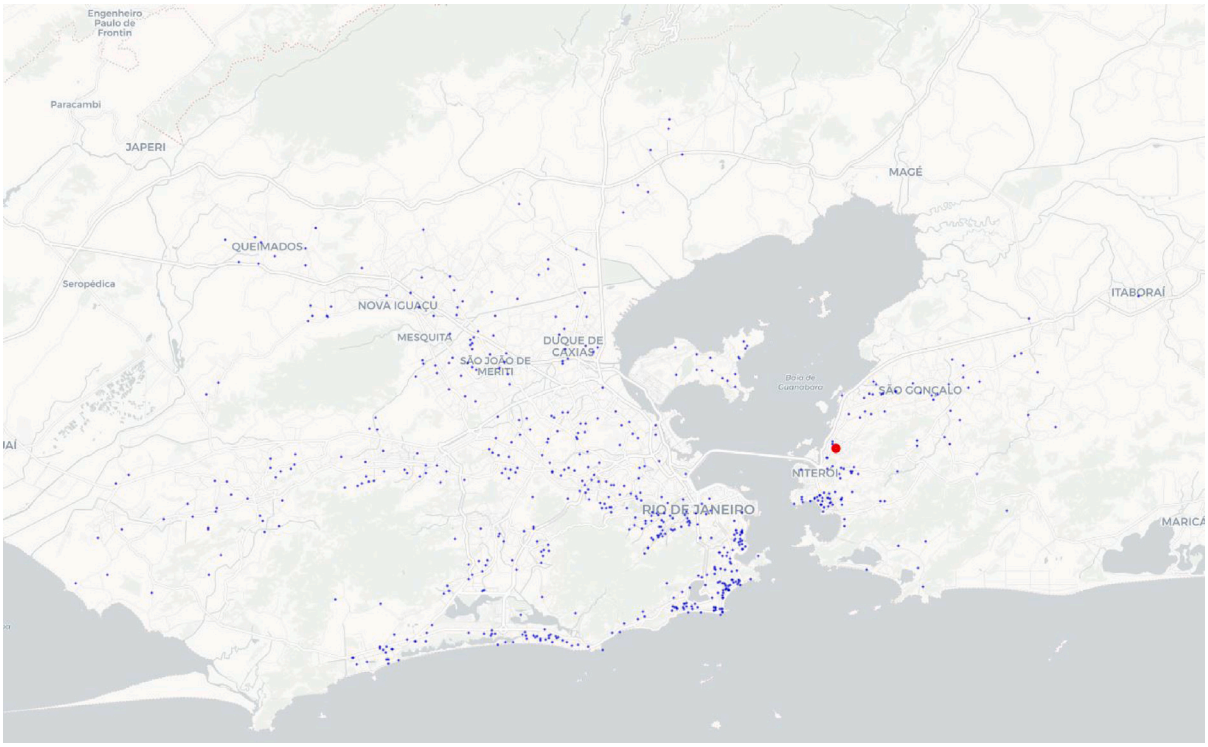
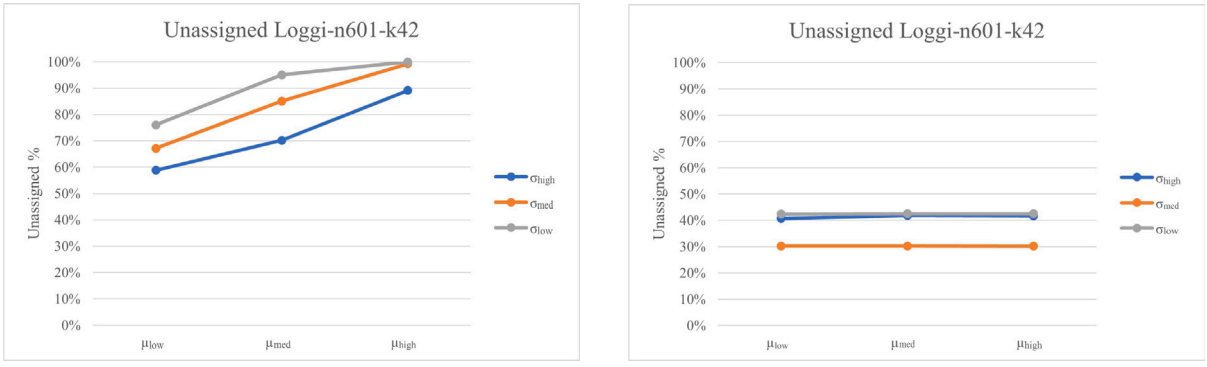
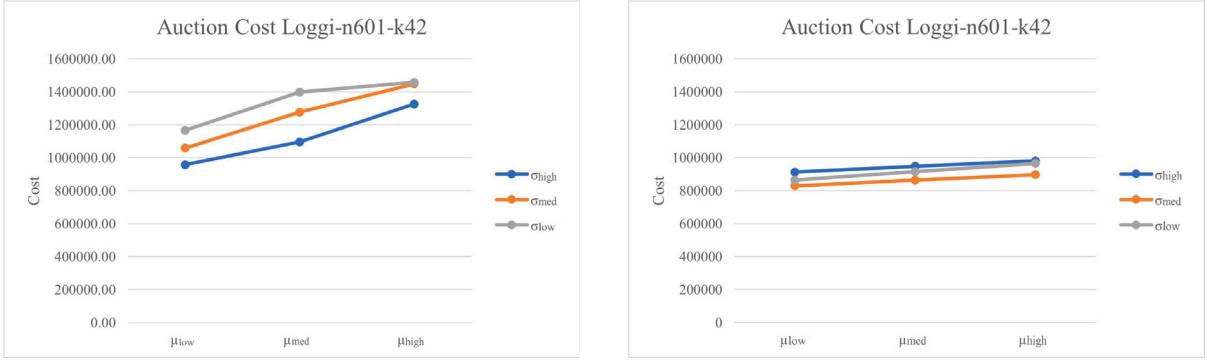



**Fig. 11.** Loggi-n601-k42 representation provided by Loggi (2021). Blue circles represent customer locations, the red circle is the depot. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)




(a) *BasePriceStat* (b) *OptPriceStat*

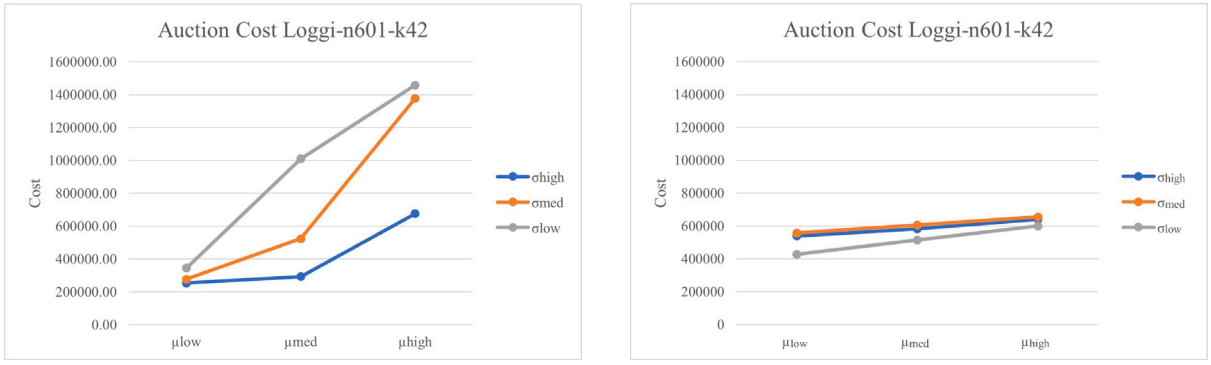
**Fig. 12.** Percentage of unassigned bundles for Loggi-n601-k42 instance by varying $\sigma$ and $\mu$ in the static auction system.




(a) *BasePriceStat* (b) *OptPriceStat*

**Fig. 13.** Auction cost for Loggi-n601-k42 instance by varying $\sigma$ and $\mu$ in the static auction system.


**Fig. 14.** Average percentage of unassigned bundles for the Loggy-n601-k42 instance. by varying $\sigma$ and $\mu$ and using *BasePriceDyn*.




(a) *BasePriceDyn* (b) *OptPriceDyn*

**Fig. 15.** Auction cost for Loggi-n601-k42 instance by varying $\sigma$ and $\mu$ in the dynamic auction system.

and compare this trend with the one shown in Fig. 14, we can observe the similarities. The auction cost increases as the number of unassigned bundles increases. Instead, the trend depicted in Fig. 15, which represents the auction cost for *OptPriceDyn*, is more stable, as the impact of $\mu$ and $\sigma$ is minimal.

In Table 7 we summarize the results obtained using $BasePriceDyn$ and $OptPriceDyn$ in terms of the percentage $GapAuctionCost$ (%). Since when using $OptPriceDyn$ all bundles are always assigned, we do not show the gap value on unassigned bundles. The results confirm the trend observed in Section 6.3. The cost of the solutions obtained using $BasePriceDyn$ is on average lower than that obtained using $OptPriceDyn$ when the number of unassigned requests is low. In particular, $BasePriceDyn$ performs better than $OptPriceDyn$ in terms of costs, when $\mu_{low}$ and $\mu_{med}$ and high and medium values of variance. In other words, if the OD behavior is sparse and the expected OD compensation is low, the strategy of starting with a lower price and increasing it through a dynamic auction is more profitable. However, the number of unassigned bundles is larger for $BasePriceDyn$, especially with low values of variance.

We may conclude that using $OptPriceDyn$ allows to assign all the bundles to the ODs, even if the parameters related to the *willingness-to-serve* function vary. This result confirms that:

- using $OptPriceDyn$ to define the price of the bundles is the most reliable strategy, even if there is scarce information about the ODs behavior;
- if there is detailed information related to the ODs' behavior and their *willingness-to-serve* function, the company may choose to use $BasePriceDyn$, minimizing the costs.

**Table 7**
Comparison between $BasePriceDyn$ and $OptPriceDyn$ for the dynamic auction with 8 runs on the Loggy-n601-k42 instance.

| | | $GapAuctionCost$ (%) |
|---|---|---|
| $\mu_{low}$ | $\sigma_{high}$ | −53.0% |
| | $\sigma_{med}$ | −50.3% |
| | $\sigma_{low}$ | −19.5% |
| $\mu_{med}$ | $\sigma_{high}$ | −49.6% |
| | $\sigma_{med}$ | −13.5% |
| | $\sigma_{low}$ | 96.3% |
| $\mu_{high}$ | $\sigma_{high}$ | 5.4% |
| | $\sigma_{med}$ | 110.0% |
| | $\sigma_{low}$ | 142.7% |

*Static auction vs. dynamic auction.* It is clear that $BasePriceDyn$ assigns a higher number of bundles compared to the $BasePriceStat$. This trend is confirmed when comparing the unassigned bundles for $OptPriceDyn$ and $OptPriceStat$.

Focusing on auction cost, Table 8 summarizes the results in terms of $GapAuctionCost$ (%), evaluated as $\frac{AC(BasePriceStat)-AC(BasePriceDyn)}{AC(BasePriceDyn)}$ and $\frac{AC(OptPriceStat)-AC(OptPriceDyn)}{AC(OptPriceDyn)}$, where $AC(\cdot)$ is the auction cost for strategy $\cdot$, for the $BasePrice$ and $OptPrice$ strategies, respectively. Looking at the first column, i.e., focusing on the $BasePriceStat$ vs. $BasePriceDyn$ comparison, the trend confirms the assessment outlined in Section 6.4. The dynamic auction allows to minimize the auction costs, the only exception is when almost all bundles are not assigned, i.e., when $\mu_{high}$ and $\sigma_{low}$. Moving to the

**Table 8**
Comparison between static and dynamic auctions for the Loggy-n601-k42 instance.

| $\mu$ | $\sigma$ | *GapAuctionCost* (%) | |
|---|---|---|---|
| | | *BasePriceStat* vs. *BasePriceDyn* | *OptPriceStat* vs. *OptPriceDyn* |
| $\mu_{low}$ | $\sigma_{high}$ | 277.7% | 69.2% |
| | $\sigma_{med}$ | 282.4% | 48.9% |
| | $\sigma_{low}$ | 238.5% | 102.1% |
| $\mu_{med}$ | $\sigma_{high}$ | 273.6% | 62.9% |
| | $\sigma_{med}$ | 143.4% | 42.5% |
| | $\sigma_{low}$ | 38.5% | 78.0% |
| $\mu_{high}$ | $\sigma_{high}$ | 96.1% | 53.1% |
| | $\sigma_{med}$ | 5.3% | 37.0% |
| | $\sigma_{low}$ | 0.0% | 60.8% |

comparison between $OptPriceStat$ and $OptPriceDyn$, whose gap values are reported in the second column, we may observe that the dynamic auction outperforms the static one in terms of total cost. The average cost gap is on average equal to 62%.

To conclude, comparing the static and the dynamic auctions, it is clear that the dynamic one outperforms the static one. The dynamic auction allows to better exploit the ODs service, assigning a larger number of bundles while minimizing the overall costs.

## 7. Conclusions

We presented a framework to handle the problem of managing customer requests in a crowdshipping platform. In particular, we addressed the pricing problem, to maximize the number of assigned parcels and minimize the costs. After generating the bundles to be assigned to occasional drivers, using a greedy algorithm, we developed two pricing strategies and two types of auctions: static and dynamic. In the static auction the price is fixed and there is a single run. The dynamic auction is instead an ascending-auction with multiple runs. As for pricing, we compared two strategies: in the first one the price is calculated by considering a base cost, that is an estimation of the cost of the route associated with the bundle, while in the second one, the price is evaluated considering the *willingness-to-serve* function associated with occasional drivers. We carried out a computational study on instances with 100 and 1000 customers as well as on an instance derived from real data. The results suggested that the dynamic auction is more efficient than the static one in reducing the auction cost and the number of unassigned bundles. Focusing on the pricing strategies, they behave differently in static and dynamic auctions. In the static auction, the second strategy outperforms the first one in terms of both reducing costs and unassigned requests. In the dynamic one, instead, even if the cost using the first pricing strategy is in several cases lower than the second one, the percentage of unassigned bundles is much higher. The second strategy assigns all the requests in almost all cases. Hence, the decision maker has to handle the trade-off between service quality and cost, by selecting the most appropriate pricing mechanism. As a final observation, we remark that the pricing strategy is robust with respect to the instance size as the tests provided similar results on instances with 100 and 1000 customers as well as on the real case instance with 600 customers.

As a future extension, we aim at considering a combined *matching*, *pricing* and *routing* problem, including also more complex real-life features such as dynamic arrival of requests, ODs' behavior, traffic congestion, delay in delivery. Another interesting aspect is the correlation between bundles generation and expected cost. Therefore, a more sophisticated approach integrating bundles' generation and pricing could be considered as future work. In addition, more sophisticated pricing techniques are worth being investigated, in particular dynamic pricing strategies to effectively calculate and adjust prices in real-time.

## CRediT authorship contribution statement

**Giusy Macrina:** Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Claudia Archetti:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Francesca Guerriero:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

Thanks are due to the two anonymous reviewers whose useful comments helped improving a previous version of the paper.

## Appendix A. Solomon's based instances computational study

### A.1. Bundles analysis

Bundles are constructed using the greedy algorithm described in Section 4.1 (i.e., Algorithm 1). The results for instances R101, RC101, R1001, and RC1001 are given in Table A.1. In particular, for each instance, we report in the column *#bundles* the number of bundles generated, in *#mean* the average number of customers (or parcels) in each bundle, while the columns *#max* and *#min* give the size of the larger and smaller bundles, respectively. Finally, in the last column, we report the average cost of the bundles, i.e., the average routing cost. We generated 16 bundles for R101, 17 for R101, 71 for R1001 and 82 for RC1001. Focusing on the number of customers in each bundle, for the instance R101 (RC101) the bundles contain 6 (5) customers on average, the smallest bundle contains 1 (2) customers, while the largest one 11 (10) customers. Considering instances with 1000 customers, the number of parcels in each bundle in R1001 (RC1001) varies on average from a minimum of 2 (1) to a maximum of 27 (36), while the average number of parcels in each bundle is 14 (12). We point out that we are not considering an upper limit on the number of customers in each bundle, but there is a limit on the duration of the working shift. As in common practice, bundles with a large number of customers might be related to cases where all customers belong to the same neighborhood or building.

### A.2. Correlation between $\eta$, $\sigma$ and $\mu$ in static auction

Focusing on the results obtained using the Static Auction on Solomon's based instances, we observe that the value of $\eta$ when using *the OptPriceStat* depends on the *willingness-to-serve* function $f$, while this is not the case with *the BasePriceStat*. Thus, we first show the effect of the expected value ($\mu$) and the variance ($\sigma$) of the *willingness-to-serve* function $f$ on the value of $\eta$ for the *OptPriceStat*. Tables A.2 and A.3 report, for each value of $\mu$ and $\sigma$, the value of the cumulative probability associated with $\eta$ determined according to the *OptPriceStat*, and the average value of $\eta$ over all solutions, for instances R and RC, respectively.

When analyzing the values in Tables A.2 and A.3, we first notice that the cumulative probability, i.e., the probability that a bundle will be accepted at the offered price $\eta$, does not change when comparing instances with 100 and 1000 customers. Moving the value of $\mu$ to $\mu_{low}$ or $\mu_{high}$ has a slight effect on the cumulative probability associated with $\eta$. Instead, the variance $\sigma$ has a strong impact and the highest value of the cumulative probability is associated with the mid value

**Table A.1**
Bundles generated for instances R101, RC101, R1001 and RC1001.

| R101 | | | | | RC101 | | | | |
|---|---|---|---|---|---|---|---|---|---|
| *# bundles* | *# mean* | *# max* | *# min* | *avg cost* | *# bundles* | *# mean* | *# max* | *# min* | *avg cost* |
| 16 | 6 | 11 | 1 | 89.53 | 17 | 5 | 10 | 2 | 97.15 |
| R1001 | | | | | RC1001 | | | | |
| *# bundles* | *# mean* | *# max* | *# min* | *avg cost* | *# bundles* | *# mean* | *# max* | *# min* | *avg cost* |
| 71 | 14 | 27 | 2 | 88.86 | 82 | 12 | 36 | 1 | 90.21 |

**Table A.2**
Cumulative probability and average value of $\eta$ for R instances using *OptPriceStat*.

| $\sigma$ | Cumulative probability | | | Avg $\eta$ for *OptPriceStat* | | |
|---|---|---|---|---|---|---|
| | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ |
| 100 R customer instances | | | | | | |
| $\sigma_{high}$ | 0.55 | 0.58 | 0.63 | 198.82 | 214.48 | 236.13 |
| $\sigma_{med}$ | 0.62 | 0.70 | 0.77 | 205.38 | 223.44 | 241.51 |
| $\sigma_{low}$ | 0.35 | 0.58 | 0.78 | 157.62 | 189.33 | 221.04 |
| 1000 R customer instances | | | | | | |
| $\sigma_{high}$ | 0.55 | 0.58 | 0.63 | 197.32 | 212.87 | 234.35 |
| $\sigma_{med}$ | 0.62 | 0.70 | 0.77 | 203.83 | 221.76 | 239.69 |
| $\sigma_{low}$ | 0.35 | 0.57 | 0.78 | 156.43 | 187.90 | 219.37 |

**Table A.3**
Cumulative probability and average value of $\eta$ for RC instances using *OptPriceStat*.

| $\sigma$ | Cumulative probability | | | Avg $\eta$ for *OptPriceStat* | | |
|---|---|---|---|---|---|---|
| | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ | $\mu_{low}$ | $\mu_{med}$ | $\mu_{high}$ |
| 100 RC customer instances | | | | | | |
| $\sigma_{high}$ | 0.55 | 0.58 | 0.63 | 215.73 | 232.73 | 256.22 |
| $\sigma_{med}$ | 0.62 | 0.70 | 0.77 | 222.85 | 242.45 | 262.05 |
| $\sigma_{low}$ | 0.35 | 0.58 | 0.78 | 171.03 | 205.43 | 239.84 |
| 1000 RC customer instances | | | | | | |
| $\sigma_{high}$ | 0.55 | 0.58 | 0.63 | 200.33 | 216.11 | 237.92 |
| $\sigma_{med}$ | 0.62 | 0.70 | 0.77 | 206.93 | 225.13 | 243.33 |
| $\sigma_{low}$ | 0.35 | 0.57 | 0.78 | 158.81 | 190.76 | 222.71 |

of the variance. In order to explain this behavior, we have to consider the link between the cumulative probability and the value of $\eta$. When moving from a large to a medium value of the variance, with a slight increase in the value of the offered compensation $\eta$ (3%), the cumulative probability increases by around 18%. Thus, the increase in the value of $\eta$ is compensated by a more than proportional increase of bundles that are assigned (and, thus, for which the company avoids paying the high cost $(\beta + \Delta) * \zeta$). When moving instead from $\sigma_{med}$ to $\sigma_{low}$, the cumulative probability decreases by around 14% reaching a similar value to the one related to the high variance scenario. However, the value of $\eta$ is on average 19% smaller than the one associated with the medium variance scenario. Thus, the smaller number of assigned bundles is compensated by a lower compensation offered to ODs.

## Appendix B. Real network case study computational study

### B.1. Bundles analysis

The results related to the analysis of the bundles are reported in Table B.4. In particular, column #*bundles* reports the number of bundles generated, in #*mean* we have the average number of customers in the bundles, while columns #*max* and #*min* report the size of the larger and smaller bundles, respectively, and finally, in the last column, we report the average cost of the bundles, i.e., the average routing cost. The total number of generated bundles is 69. The smallest bundle contains one customer, while the largest one 36. On average, each bundle contains 8 customers. As in the Solomon's based instances, we did not consider an upper limit to the number of customers in each bundle, but a limit on the temporal route length.

**Table B.4**
Bundles generated for the Loggi-n601-k42 instance.

| Loggi-n601-k42 | | | | |
|---|---|---|---|---|
| *# bundles* | *# mean* | *# max* | *# min* | *avg cost* |
| 69 | 8 | 36 | 1 | 3523.82 |

## References


Alnaggar, A., Gzara, F., Bookbinder, J.H., 2021. Crowdsourced delivery: A review of platforms and academic literature. Omega 98, 102139.

Archetti, C., Bertazzi, L., 2021. Recent challenges in routing and inventory routing: E-commerce and last-mile delivery. Networks 77, 255–268.

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

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

Ausseil, R., Pazour, J.A., Ulmer, M.W., 2022. Supplier menus for dynamic matching in peer-to- peer transportation platforms. Transp. Sci. 56, 1304–1326.

Behrend, M., Meisel, F., Fagerholt, K., Andersson, H., 2019. An exact solution method for the capacitated item-sharing and crowdshipping problem. European J. Oper. Res. 279 (2), 589–604.

Boysen, N., Emde, S., Schwerdfeger, S., 2022. Crowdshipping by employees of distribution centers: Optimization approaches for matching supply and demand. European J. Oper. Res. 296 (2), 539–556.

Bräysy, O., 2003. A reactive variable neighborhood search for the vehicle-routing problem with time windows. INFORMS J. Comput. 15 (4), 347–368.

Bräysy, O., Gendreau, M., 2001. Route Construction and Local Search Algorithms for the Vehicle Routing Problem with Time Windows. Technical Report, Sintef Report, STF42 A 1024.

Buldeo Rai, H., Verlinde, S., Merckx, J., Macharis, C., 2018. Can the crowd deliver? Analysis of crowd logistics' types and stakeholder support. In: City Logistics 3: Towards Sustainable and Liveable Cities - the 10th International Conference on City Logistics - JW Marriott Phuket Resort and Spa Hotel, Phuket Island, Thailand. Vol. 5, John Wiley & Sons, Ltd, pp. 89–108.

Caric, T., Gold, H., 2008. Vehicle Routing Problem. IntechOpen, Rijeka.

Dahle, L., Andersson, H., Christiansen, M., 2017. The vehicle routing problem with dynamic occasional drivers. In: Bektaş, T., Coniglio, S., Martinez-Sykora, A., Voß, S. (Eds.), Computational Logistics. Springer International Publishing, Cham, pp. 49–63.

Dahle, L., Andersson, H., Christiansen, M., Speranza, M.G., 2019. The pickup and delivery problem with time windows and occasional drivers. Comput. Oper. Res. 109, 122–133.

Dai, H., Liu, P., 2020. Workforce planning for O2O delivery systems with crowd-sourced drivers. Ann. Oper. Res. 291, 219–245.

Dayarian, I., Savelsbergh, M., 2020. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. Prod. Oper. Manage. 29 (9), 2153–2174.

Di Puglia Pugliese, L., Ferone, D., Macrina, G., Festa, P., Guerriero, F., 2023. The crowd-shipping with penalty cost function and uncertain travel times. Omega 115, 102776.

Gansterer, M., Hartl, R., 2018. Centralized bundle generation in auction based collaborative transportation. OR Spectrum 40 (3), 613–635.

Gansterer, M., Hartl, R.F., Sörensen, K., 2020. Pushing frontiers in auction-based transport collaborations. Omega 94, 102042.

Horner, H., Pazour, J., Mitchell, J.E., 2021. Optimizing driver menus under stochastic selection. Transp. Res. 153, 102419.

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

Le, T., Ukkusuri, S., Xue, J., Van Woensel, T., 2021. Designing pricing and compensation schemes by integrating matching and routing models for crowd-shipping systems. Transp. Res. 149, 102209.

Li, M., Qin, Z., Jiao, Y., Yang, Y., Wang, J., Wang, C., Wu, G., Ye, J., 2019. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In: The World Wide Web Conference. pp. 983–994.

Loggi, 2021. LoggiBUD: Loggi Benchmark for Urban Deliveries. Technical Report.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., Laganà, D., 2017. The vehicle routing problem with occasional drivers and time windows. In: Sforza, A., Sterle, C. (Eds.), Optimization and Decision Science: Methodologies and Applications. In: Springer Proceedings in Mathematics & Statistics, vol. 217, Springer, Cham, Switzerland, pp. 577–587.

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

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

Mckinnon, A., 2016. Crowdshipping: A Communal Approach to Reducing Urban Traffic Levels. Logistics White Paper - Kühne Logistics University: Hamburg, Germany.

Mofidi, S.S., Pazour, J.A., 2019. When is it beneficial to provide freelance suppliers with choice? A hierarchical approach for peer-to-peer logistics platforms. Transp. Res. 126, 1–23.

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

Punel, A., Stathopoulos, A., 2017. Modeling the acceptability of crowdsourced goods deliveries: role of context and experience effects. Transp. Res. 105, 18–38.

Sampaio, A., Savelsbergh, M., Veelenturf, L., Van Woensel, T., 2020. Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. Networks 76, 232–255.

Skålnes, J., Dahle, L., Andersson, H., Christiansen, M., Hvattum, L.M., 2020. The multistage stochastic vehicle routing problem with dynamic occasional drivers. In: Lalla-Ruiz, E., Mes, M., Voß, S. (Eds.), Computational Logistics. Springer International Publishing, Cham, pp. 261–276.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35, 254–265.

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

Torres, F., Gendreau, M., Rei, W., 2022b. Vehicle routing with stochastic supply of crowd vehicles and time windows. Transp. Sci. 56 (3), 631–653.

Triki, C., 2021. Using combinatorial auctions for the procurement of occasional drivers in the freight transportation: A case-study. J. Clean. Prod. 304, 127057.

Yıldız, B., 2021. Package routing problem with registered couriers and stochastic demand. Transp. Res. 147, 102248.

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