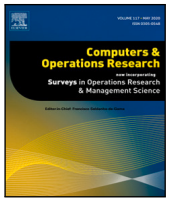
A machine learning optimization approach for last-mile delivery and third-party logistics

Computers & Operations Research 157 (2023) 106262

Contents lists available at ScienceDirect

Computers and Operations Research

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



A machine learning optimization approach for last-mile delivery and third-party logistics

Maria Elena Bruni [a], Edoardo Fadda [b,*], Stanislav Fedorov [c], Guido Perboli [d,e]

[a] DIMEG, University of Calabria, Rende, Italy
[b] DISMA, Politecnico di Torino, Torino, Italy
[c] DAUIN & CARS@Polito, Politecnico di Torino, Turin, Italy
[d] DIGEP & CARS@Polito, Politecnico di Torino, Turin, Italy
[e] Arisk S.p.A., Milan, Italy

ARTICLE INFO

Keywords:
Metaheuristics
Machine learning
Variable cost and size bin packing
Third-party logistics
Last-mile delivery
Capacity planning

ABSTRACT

Third-party logistics is now an essential component of efficient delivery systems, enabling companies to purchase carrier services instead of an expensive fleet of vehicles. However, carrier contracts have to be booked in advance without exact knowledge of what orders will be available for dispatch. The model describing this problem is the variable cost and size bin packing problem with stochastic items. Since it cannot be solved for realistic instances by means of exact solvers, in this paper, we present a new heuristic algorithm able to do so based on machine learning techniques. Several numerical experiments show that the proposed heuristics achieve good performance in a short computational time, thus enabling its real-world usage. Moreover, the comparison against a new and efficient version of progressive hedging proves that the proposed heuristic achieves better results. Finally, we present managerial insights for a case study on parcel delivery in Turin, Italy.

1. Introduction

The growth of the urban population and rising living standards have led to a dramatic increase in the demand for services and goods. These conditions have paved the way to a competitive environment where companies fight for market share by continuously providing flexibility in delivering options while maintaining high resource efficiency. Consequently, the entire last-mile delivery sector is impacted, making it the most challenging in the entire logistic chain (Perboli et al., 2021a; Sergi et al., 2021).

Several management solutions have been developed to achieve the desired efficiency level and hedge against the risks coming from real-world uncertainty. The most important is third-party logistics (3PL), which is an organization's use of third-party businesses to outsource elements on the distribution, warehousing, and/or fulfillment services side (Perboli et al., 2017b). The adoption of 3PL translates into the advantage of reducing fleet investment while maintaining the same quality of service. This work focuses on the usage of 3PL for logistic tasks, i.e., we consider a company booking containers from a third-party logistic company to deliver goods. A practical example of such a decision is when service providers secure long-term distance contracts with the carriers (Crainic et al., 2016). In this context, the primary decision is related to the volume of bins to book for transportation. Such a decision is strongly affected by demand uncertainty and possible supply delays. As a result, researchers have developed several stochastic models to assist the decision-making process (Perboli et al., 2014). Among them, the most relevant is the Variable Cost and Size Bin Packing problem with Stochastic Items (VCSBPPSI) (Crainic et al., 2014a). It considers a two-stage stochastic optimization problem characterized by a tactical and an operational phase. During the tactical phase, the company negotiates a capacity plan with one or more 3PL firms, resulting in medium-term contracts specifying the capacity to be used (quantity and type of bins). The bins included in the capacity plan are chosen in advance without exact knowledge of what items will be dispatched. Extra capacity is purchased at a higher price during the operational phase if the planned capacity is not sufficient. A previous study on the VCSBPPSI has shown that commercial solvers are not able to deal with realistic instances of the problem in a reasonable amount of time, thus justifying the implementation of heuristics (Crainic et al., 2014a). In particular, the progressive hedging (PH) heuristic has proven to be the most effective (Crainic et al., 2016). However, despite its good performance in its classical form, it is not able to solve real-world instances when decisions must be made within a short period of time, as in e-commerce applications. Such a drawback requires new methods, among

* Corresponding author.
E-mail address: edoardo.fadda@polito.it (E. Fadda).

https://doi.org/10.1016/j.cor.2023.106262
Received 17 June 2022; Received in revised form 23 April 2023; Accepted 24 April 2023
Available online 28 April 2023

which the most promising are the ones applying machine learning (ML) techniques. In recent years, the so-called *learning to optimize* (or simply L2O) have quickly surfaced as efficient and effective solutions to many optimization problems (Chen et al., 2021). The key idea of L2O is to train a machine learning algorithm to learn the optimization process over a set of training problem instances and generalize it to new testing problems. Even if these methods require a time-consuming training phase, the inference step requires negligible computational time, thus enabling their usage in a real-time setting. Furthermore, these approaches prove to have good generalization capabilities with respect to realistic instances since they may use practitioners' insights without any architectural modification. Nevertheless, in the literature, there are two major gaps. First, only a few papers consider ML heuristics tailored for the solution of two-stage stochastic optimization problems (Nair et al., 2018; Dumouchelle et al., 2022). Second, ML algorithms are widely used together with other techniques (e.g., with genetic algorithm Achamrah et al., 2021 or with exact solver Baldo et al., 2023) but only a few times are used alone (Dumouchelle et al., 2022). In fact, the major limitation to directly predict the solution is the inability to handle hard constraints (Dumouchelle et al., 2022).

This paper contributes to filling these gaps and improving the results already found. In particular, we introduce a new ML heuristic based on supervised classification techniques and apply it to calculate the first-stage decision variables of the VCSBPPSI. To assess its performance, we compare it against the most recent and effective version of the PH on a set of instances with different characteristics. The computational results show that the proposed methodology is more efficient than the PH, which paves the way to a real-time application of the solution. Moreover, the proposed ML heuristic can be easily generalized and applied to other optimization problems characterized by binary variables. Finally, we show how the ML approach can be integrated into a more complex last-mile process by applying it to a real case study.

To summarize, the contributions of the paper are the following:

- we improve the ML heuristic in Baldo et al. (2023) in order to deal with a two-stage stochastic program and to provide the whole solution of the optimization problem. This new method can be easily applied to other two-stage stochastic problems with binary decision variables;
- we prove that ML algorithms can be used as heuristics to solve two-stage stochastic problems without relying on other solution methods such as genetic algorithm, adaptive large neighborhood search, etc.;
- we provide two new heuristics (the improved PH and the ML heuristic) which enhance the previous results on the VCSBPPSI.
- we apply the approach to the real case study of a medium-sized metropolitan area. The problem setting comes from collaborations of the authors, including work on urban distribution in the metropolitan area of Turin (Italy) as part of the development of the new Logistics and Mobility Plan of the Regional Government of the Piedmont region to be activated in 2025 (Perboli et al., 2021a,b).

The paper is organized as follows. Section 2 covers the review of the existing literature. Section 3 presents the mathematical description of the VCSBPPSI. The ML heuristic and the PH algorithm are described in Section 4, while the experimental setting and numerical results are presented in Section 5. In addition, interesting policy-making and managerial insights are derived from the real case study in Section 6. Finally, conclusions and future work are presented in Section 7.

2. Literature review

In this section, we explore the literature review following two branches. First, we review the most promising and recent heuristics approaches using ML techniques. Then, we present the literature about existing mathematical formulations and heuristics for the VCSBPPSI and related problems.

The application of ML methods for the optimization process is a recent and growing topic in the literature. The most recent surveys on the topics include (Mele et al., 2021), which considers the application of ML to the traveling salesman problem, Talbi (2020), which resumes data-driven ML meta-heuristics, and Ning and You (2019), which analyzes the applications of deep learning to mathematical programming under uncertainty. Based on our literature analysis, the vast majority of applications use ML to decide the optimal values of parameters/hyperparameters of some solution techniques (Lodi and Zarpellon, 2017; Montemanni et al., 2018; da Costa et al., 2021) or to approximate complex terms of the objective function (Bengio et al., 2020). Nevertheless, to our knowledge, few researchers have directly tried to find the optimal solution to an optimization problem by directly using ML techniques (Baldo et al., 2023; Miki and Ebara, 2019). The advantage of these methods is that they can compute accurate solutions in real-time applications. In particular, they can be effective in blockchain-based solutions, where the smart contracts cannot make use of traditional heuristic approaches, due to the consequent decrease of the performance of the overall system (Capocasale et al., 2021; Perboli et al., 2018a). Moreover, for most ML approaches to account for uncertainty, it is not required to provide a proper probability distribution.

This work focuses on applying ML to stochastic combinatorial optimization problems. To the best of the authors' knowledge, it is possible to identify three representative papers in this setting: Mirshekarian and Sormaz (2018), Larsen et al. (2018), and Baldo et al. (2023). The first presents a general agent-based ML adaptation for combinatorial optimization problems, it requires considerable tuning and does not account for uncertainty, which prevents its application to the VCSBPPSI. In the second, the authors apply different ML-based heuristics to approximate the second stage of the two-stage stochastic program. Consequently, it reduces the computational burden related to the scenarios, but it leads to a non-linear optimization problem. The present work concentrates on the first-stage decision variables, providing a robust solution with reduced computational effort. The third paper is an example of effective usage of the L2O stream, it presents a simple-to-implement ML heuristic that uses classifiers to decide the variables' values. In particular, the heuristic exploits the power of a supervised ML algorithm that assigns to each binary variable the probability of being 0 or 1, based on a predefined set of features; from these values, the variables whose result is closest to the bounds are directly set to the predicted value, while for the most uncertain ones, a run of the exact solver is executed, with a significantly reduced search space.

A similar approach dealing with constraints and not with variables has been proposed in Jiménez-Cordero et al. (2022). In particular, the authors first identify offline the invariant constraints set of MILP instances. Then, they train a machine learning method for detecting an invariant constraints set as a function of the problem parameters of each instance. Finally, they predict the invariant constraints set of the new unseen MILP application and use it to initialize the constraint generation method. This approach can be seen as the dual of the approach proposed in Baldo et al. (2023).

The present work has its roots in the heuristic in Baldo et al. (2023) and improves it to the point that the ML procedure can generate the entire solution. This proves that ML approaches can be used in the context of stochastic combinatorial optimization as a heuristic method, whose performance can overpass those of more traditional methods such as the PH.

The first mathematical model for the 3PL problem has been formulated with the variable cost and size bin packing problem (VCSBPP), which is a generalization of the bin packing problem introduced in Crainic et al. (2011). It consists of packing a set of items into a set of heterogeneous bins (i.e., with different sizes and costs), minimizing the cost of all used bins. Two approaches have been considered to

account for demand uncertainty. The first one consists in mitigating uncertainty by using demand forecast (Fadda et al., 2021). Nevertheless, this approach requires accurate tuning of parameters and it does not compute solutions robust to uncertainty. Instead, the second approach is to generalize the problem to its stochastic version by considering the characteristics of items to be dispatched not known in advance (Crainic et al., 2014a). The number of variables in this model increases drastically and prevents the usage of standard off-the-shelf solvers for solving realistic instances. Thus, heuristics to solve VCSBPPSI are required.

The one that shows the best performance so far is the PH (Crainic et al., 2016). Originally proposed in Rockafellar and Wets (1991), it is based on the augmented Lagrangian relaxation of the non-anticipative constraints of the stochastic problem. The PH converges to the optimal solution if the problem is convex, while there is no guarantee of convergence if the problem has integer variables (as for the VCSBPPSI). Nevertheless, it has shown good performance as a heuristic for several discrete problems (Perboli et al., 2017a; Crainic et al., 2014b). Recent advances in PH-based heuristics have proposed using the sum of the absolute rather than the sum of the squared (as done in the augmented Lagrangian relaxation) (Fadda et al., 2019; Li et al., 2020). This improvement leads to a linear programming problem instead of a quadratic one, thus reducing the computational burden and increasing the solver's efficiency and effectiveness. Despite its excellent performance, the PH algorithm requires solving several integer sub-problems. As a result, its computational time is not negligible, and it cannot be used in real-time applications such as logistic platform support (Kwak et al., 2020). This discussion confirms the necessity of developing the ML-based heuristics for the VCSBPPSI to account for the uncertainty and to be used efficiently in real-time applications.

3. Variable cost and size bin packing problem with stochastic items

In this section, we recall VCSBPPSI problem, its possible applications and the mathematical formulation (Crainic et al., 2014a). The VCSBPPSI concerns the decision problem of a shipper which needs to secure capacity of different types from a carrier, to meet uncertain demand. The capacity types could be transportation modes (e.g., ship or train slots, containers, space in cargo bikes or vans), specific carriers, or storage space within given facilities. Each type has particular characteristics in terms of unit cost and size. This problem setting is relevant in different contexts, where the main ones are last-mile delivery and long-haul transportation.

In last-mile delivery, tactical capacity planning aims to ensure that consolidation can be efficiently performed. Specifically, managers must secure the required numbers of vehicles of various types, which will be available to correctly perform the transportation operations. Private and public (e.g., transit authorities) carriers and service providers make coalitions for capacity sharing and integrated decision-making to consolidate freight and reduce the impact of freight transportation and logistics on the city. Also multi-tier smart urban transportation systems are implementing these approaches (Perboli and Rosano, 2020; Crainic et al., 2021). The goal of such systems is to reduce the negative impacts (i.e., costs, congestion, noise, etc.) associated with the vehicles transporting freight in urban areas by more efficiently using their capacity (i.e., increasing the average vehicle fill rate and reducing the number of empty trips that are performed). These multi-tier systems are based on the application of two general principles: *(i)* the consolidation of loads originating from different shippers within the same vehicles and *(ii)* the coordination of the distribution operations within the city. In this case, the use of multiple transportation tiers enables the system to utilize specifically adapted infrastructure and specialized fleets at each tier to better attain the overall goal. In this context, the main source of uncertainty is due to short-term customers' deliveries due to same-day service provided by companies such as Amazon (Perboli et al., 2021a), and VCSBPPSI aims to ensure that consolidation can be efficiently performed by forecasting the proper mix of vehicles serving the different tiers.

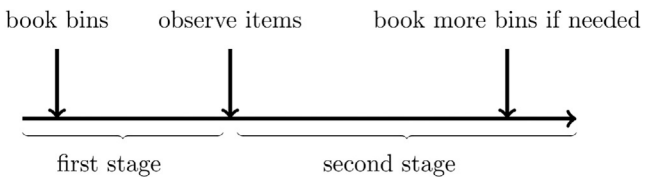


Fig. 1. Graphical representation of the sequence of decisions and stages.

Instead, in long-haul transportation, the VCSBPPSI is more related to tactical capacity planning. The shipper negotiates multi-type capacity in advance and uses it to perform its shipping or storage activities repeatedly, e.g., every day, week, or month, over a certain planning horizon, e.g., one semester, season, or year. The output of this negotiation is a medium-term contract, which includes the quantity, i.e., the number of units, the capacity of each type, the costs to use the contracted capacity (Crainic et al., 2014a). Indeed, given the time lag that usually exists between the signing of the contract and the logistics operations, as well as the hazards and risks associated with predicting future supply and demand levels, several sources of uncertainty are affecting the contract negotiation.

The source of uncertainty we consider in the VCSBPPSI is the demand, i.e., the orders that the shipper transports or stores. Indeed, even in the most 'regular" context of operations, the demand fluctuates in time and what one observes at any given occurrence of activity is generally different from a single-value (also called point forecast) prediction of the number of units to transport or store and the size of each of those units. This may result in insufficient booked capacity available on the shipping day, compromising the fulfillment of the contract and generating additional costs to handle the situation.

The VCSBPPSI problem is described as a two-stage stochastic program. In the first stage the decision maker, which we assume to be a company, has to rent some bins from a 3PL provider. Then, in the second stage, all the information about the items is available, and the decision maker has to decide to rent more bins. A graphical representation of the sequence of decisions and stages is depicted in Fig. 1.

Let us consider a set of scenarios S with cardinality S. Since the information related to the items is unknown in the first stage, we define the set of items to be packed in scenario s as $\mathcal{I}^s$. Each item $i \in \mathcal{I}^s$ has a random volume v_i^s. Bins are divided into two sets: the set of bins available during the first stage $\mathcal{J}$ and the set of bins available in scenario s, $\mathcal{K}^s$. Each bin $j \in \mathcal{J}$ and $k \in \mathcal{K}^s$ is characterized by a cost c_j and c_k and by a volume V_j and V_k, respectively. Notice that both bin costs and volumes are deterministic because they are known in advance and represent a shared knowledge between the decision-maker and the 3PL provider. Since the second-stage bins are booked on the spot market and not ahead, we consider that $c_k/V_k > c_j/V_j \ \forall\ j \in \mathcal{J}\ k \in \mathcal{K}^s$.

The model considers the following binary decision variables:

- $x_{ij}^s, i \in \mathcal{I},\ j \in \mathcal{J} \cup \mathcal{K}^s, s \in S$ equal to 1 if the item i is packed in the bin j in scenario s;
- $y_j,\ j \in \mathcal{J}$ equal to 1 if bin j is booked;
- $z_k^s, k \in \mathcal{K}^s$ equal to 1 if bin k is booked in scenario s.

The VCSBPPSI formulation is

$$\min \sum_{j\in\mathcal{J}} c_j y_j + \sum_{s\in S} p_s \Big(\sum_{k\in\mathcal{K}^s} c_k z_k^s\Big) \tag{1}$$

$$\text{s.t.} \sum_{j\in\mathcal{J}} x_{ij}^s + \sum_{k\in\mathcal{K}^s} x_{ik}^s = 1, \qquad \forall i \in \mathcal{I}^s,\ \forall s \in S \tag{2}$$

$$\sum_{i\in\mathcal{I}^s} v_i^s x_{ij}^s \le V_j y_j, \qquad \forall j \in \mathcal{J},\ \forall s \in S \tag{3}$$

$$\sum_{i \in \mathcal{I}^s} v_i^s x_{ik}^s \leq V_k z_k^s, \qquad \forall k \in \mathcal{K}^s,\ \forall s \in \mathcal{S} \tag{4}$$

$$x_{ij}^s \in \{0,1\}, \qquad \forall i \in \mathcal{I}^s,\ \forall j \in \mathcal{J} \tag{5}$$

$$x_{ik}^s \in \{0,1\}, \qquad \forall i \in \mathcal{I}^s,\ \forall k \in \mathcal{K}^s \tag{6}$$

$$y_j \in \{0,1\}, \qquad \forall j \in \mathcal{J} \tag{7}$$

$$z_k^s \in \{0,1\} \qquad \forall k \in \mathcal{K}^s, \tag{8}$$

where p_s is the probability of realization of the scenario $s \in \mathcal{S}$. The objective function (1) minimizes the sum of the cost of booking the first-stage bins and the expected cost associated with the extra capacity bought on the spot market. Constraints (2) ensure that each item is packed in a single bin. Constraints (3), and (4) make sure that the capacity of each bin is not exceeded. Finally, Constraints (5)–(8) specify the domain of the decision variables.

Model (1)–(8) has $J + J\sum_s I_s + \sum_s K_s + \sum_s K_s I_s$ variables, where the terms account for y_j, x_{ij}^s, z_k^s, and x_{ik}^s, respectively. If just one or no scenarios are considered, the problem boils down to the variable sized bin packing problem. This problem contains the classical one-dimensional bin packing problem as a particular case. There, all the bins have the same capacity and cost. Since the bin packing problem is an NP-hard problem (Garey, 1979) thus, also the VSBPP is NP-hard (Correia et al., 2008) as well as the VCSBPPSI.

It is worth noting that considering deterministic bin volumes in the second stage (i.e., V_k instead of V_k^s) does not entail a modeling restriction. In fact, V_k affects Constraints (4) only and, by dividing both terms of the equation for V_k, we obtain

$$\sum_{i \in \mathcal{I}^s} \psi_i^s x_{ik}^s \leq z_k^s,$$

where $\psi_i^s = v_i^s / V_k$ is a random coefficient even if V_k is not. Thus, by considering a suitable distribution for v_i^s, it is possible to incorporate the effect of stochasticity in the second-stage volumes. Moreover, it is possible to model an item whose order is known in the first stage by setting $v_i^s = \bar{v}_i,\ \forall s \in \mathcal{S}$.

Finally, please notice that model (1)–(8) can be infeasible if the volume of the bins is not sufficient to contain all the items. Nevertheless, in the following, we do not consider this issue since usually the number of bins available through 3PL providers is much greater than the volume of the bins to allocate.

4. Solution heuristics

In this section, we present the proposed ML heuristic in 4.1. Since our goal is not only to prove its effectiveness but also to compare its performances with another state-of-the-art heuristic, we consider an improved version of PH whose traditional version has been used in Crainic et al. (2014a). We present the heuristic in Section 4.2 for the sake of completeness.

4.1. Machine learning heuristic

This section introduces the ML heuristic based on a supervised classifier applied to the first-stage variables. We are not interested in the second stage variables, since the real decisions that have to be implemented are the ones of the first stage (Birge and Louveaux, 1997). The main idea is to classify each bin according to whether it belongs to the set that must be booked or not by using its characteristics. Intuitively, given a set of bins and items to deliver, it is reasonable to assume that the bins characterized by the lowest cost per unit of volume will be in the solution. However, this may lead to a simplistic and sub-optimal decision rule, i.e., to book all the bins with a cost per unit lower than a threshold. The ML heuristic generalizes this guess by considering more features (e.g., the exact value of the continuous relaxation of the associated decision variable, reduced costs, etc.) and a more sophisticated decision rule, provided by the non-linearity of the ML classifiers.

In particular, for each first-stage bin $j \in \mathcal{J}$, a vector of features f^j is computed. This vector is fed into a classifier which has been trained to compute $\hat{y}_j$, an estimation of y_j. In other words, the classifier defines two classes of bins: those used in the solution ($\hat{y}_j = 1$) and those that are not ($\hat{y}_j = 0$).

The training dataset is built by collecting the features and the respective optimal value of y_j for all the items j of several instances. Thus, the training dataset is made by a set of observations (f^j, y_j). To further clarify this procedure, we present a pseudo-code in Algorithm 1.

Algorithm 1 Training set building.

```
D = ∅
for k = 1, 2, ...K do
    Generate a problem instance
    y ← solution of the problem (1)–(8).
    for j = 1, 2, ...J do
        Compute the features f^j
        Add to D the features f^j and the label y_j
    end for
end for
```

Since the classifier acts as a general function that for each bin computes the associated y_j variables, the features selected must collect information on the quality of the single bin as well as the interaction between the other bins. In Table 1, we summarize the considered features with a short description and the corresponding equations to compute them. In particular, we consider a first set of features describing the characteristics of the single bin from an economic point of view (`Relative cost sum`, `Relative cost max` `Unitary cost`) as well as from a physical point of view (`Relative capacity sum`, `Relative capacity max`). Despite being related to every single bin, these features are normalized with respect to the characteristics of the other bins. Since the best normalization procedure is not known a priori, several features consider different strategies. Of course, we expect that when applying feature selection just a small subset of them would be selected. Then, we consider features collecting the information about the likelihood to have a particular bin in the final solution, namely `Continuous relaxation`, `Reduced cost`, `Items placed avg`, `Items placed max` and `Items placed min`. All these quantities are obtained by solving the continuous relaxation of the model, thus they can be computed really quickly. While `Continuous relaxation` and `Reduced cost` measure the likelihood of a given bin to be booked, `Items placed avg`, `Items placed max` and `Items placed min` measure their utility in the second stage. Since x_{ij}^s is equal to one if the item i is assigned to container j in scenario s, the greater is the $\sum_j x_{ij}^s$, the more the usefulness of bin j. Here, again we propose a different way to measure the same quantity and we expect that the feature will select just a few of them. Moreover, `Items capacity` and `Items capacity quant` measure the number of items each bin can contain. Bins with high values of these coefficients can be successfully used to hedge against the unexpected volume in the second stage of the problem. Finally, `Unitary cost wrt ss avg`, `Unitary cost wrt ss max`, `Unitary cost wrt ss min` measure the economic convenience of using a first stage bin against a second stage one. This coefficient is particularly useful to detect possible bins whose unitary costs are not too convenient with respect to the ones in the second stage.

The computation of the proposed features can be done in polynomial time since the most expensive operation is to solve the continuous realization of Model (1)–(8). It is interesting to point out that from a general point of view, the ML heuristic can be seen as a complex decision rule similar to the precedence rule used in scheduling problems. In fact, as in scheduling applications we use a given feature (earliest due date, weighted shortest processing time first, etc.) to decide which

Table 1
Feature description and computation.

Feature name	Description	Expression
`Relative cost sum`	Relative cost of bin j, normalized through summation	$\frac{c_j}{\sum_{j\in\mathcal{J}} c_j}$
`Relative cost max`	Relative cost of bin j, normalized through maximum	$\frac{c_j}{\max_{j\in\mathcal{J}} c_j}$
`Relative capacity sum`	Relative capacity of bin j, normalized through summation	$\frac{V_j}{\sum_{j\in\mathcal{J}} V_j}$
`Relative capacity max`	Relative capacity of bin j, normalized through maximum	$\frac{V_j}{\max_{j\in\mathcal{J}} V_j}$
`Unitary cost`	Unitary cost of bin j , defined as its "gain"	$\frac{c_j/V_j}{\max_{j\in\mathcal{J}}(c_j/V_j)}$
`Continuous relaxation solution`	The value of y_j in the continuous relaxation of Model (1)–(8)	–
`Reduced cost`	The normalized value of the reduced cost r_j of the bin in the continuous relaxation of Model (1)–(8)	$\frac{r_j}{\max_{j\in\mathcal{J}} r_j}$
`Items placed avg`	The average quantity of the items placed into the bin j in the continuous relaxation of Model (1)–(8)	$\frac{1}{S}\sum_{s\in S}\sum_{i\in I} x_{ij}^s$
`Items placed max`	The maximum quantity of the items placed into the bin j in the continuous relaxation of Model (1)–(8)	$\max_{s\in S}\sum_{i\in I} x_{ij}^s$
`Items placed min`	The minimum quantity of the items placed into the bin j in the continuous relaxation of Model (1)–(8)	$\min_{s\in S}\sum_{i\in I} x_{ij}^s$
`Items capacity`	The theoretical number of item that the bin may contain	$\frac{V_j}{\frac{1}{S}\sum_{s\in S}\frac{1}{I^s}\sum_{i\in I^s} v_i^s}$
`Items capacity quant`	The theoretical item capacity, where $q(v_i^s, 0.8)$ is the 0.8 quantile of the v_i^s	$\frac{V_j}{\frac{1}{I}\sum_{i\in I} q(v_i^s,0.8)}$
`Unitary cost wrt ss avg`	The unitary cost of the bin with respect to estimations of the second-stage unitary cost, defined through averaging, maximum and minimum functions ($g_j = c_j/V_j$)	$\frac{g_j}{\frac{1}{K}\sum_k \frac{1}{S}\sum_{s\in S} g_k^s}$
`Unitary cost wrt ss max`		$\frac{g_j}{\max_k \frac{1}{S}\sum_{s\in S} g_k}$
`Unitary cost wrt ss min`		$\frac{g_j}{\min_k \frac{1}{S}\sum_{s\in S} g_k^s}$

operation must be processed first, the ML heuristic uses a complex decision function based on a set of features to decide which first stage bins to book.

Finally, it is worth noting that the decision-maker can introduce other features to add particular insights (e.g., the risk or financial exposition of the provider of bins j, or integrating with IoT data Malagnino et al., 2021; Cagliero et al., 2018). This fact is indeed one of the main strengths of the presented heuristics. Another one is its generality and the possibility of applying it to every two-stage stochastic problem characterized by binary variables.

4.2. Customized progressive hedging

The PH consists of an augmented Lagrangian relaxation of the non-anticipative constraints. Its convergence to the optimal solution is guaranteed if the problem is convex, and it has proved to be an effective heuristic in several cases when convexity is not guaranteed (Fadda et al., 2019).

Since Model (1)–(8) separate the decision variables of the first and second stages, it does not have non-anticipative constraints. Hence, to apply the PH framework, we replace each first-stage variable (y_j) with new decision variables, one for each scenario (y_j^s), and we add the following non-anticipative constraints:

$$y_j^s = y_j^{s'} \quad \forall s \neq s' \text{ with } \; s, s' \in S \text{ and } \; j \in \mathcal{J},$$

which can be rewritten as

$$y_j^s = \bar{y}_j \quad \forall s \in S, \; j \in \mathcal{J}, \tag{9}$$

where $\bar{y}_j = \sum_{s=1}^{|S|} p_s y_j^s$. The PH takes advantage of Lagrangian relaxation of Constraints (9) by splitting the original problem in $|S|$ independent sub-problems. As mentioned above, the PH is the augmented Lagrangian relaxation of the non-anticipativity constraints. The term "augmented" refers to the fact that, in order to increase the speed of convergence, the typical PH scheme adds the squared l_2 norm of the error term (i.e., given $u, v \in \mathbb{R}^n$, $\|u - v\|_2^2 = \sum_{i=1}^n (u_i - v_i)^2$) to the cost function. To maintain the problem's linearity, we used a customized version of the PH, proposed by Fadda et al. (2019) and successfully applied by Li et al. (2020). In this customized version, the l_2 norm is replaced with the l_1 norm (i.e. given $u, v \in \mathbb{R}^n$, $\|u-v\|_1 = \sum_{i=1}^n |u_i - v_i|$). As a result, for each scenario s, the following problem is obtained:

$$\begin{aligned} \min \; & \sum_{j\in\mathcal{J}} c_j y_j^s + \sum_{k\in\mathcal{K}^s} c_k z_k + \sum_{j\in\mathcal{J}} \theta_s |\bar{y}_j - y_j^s| \\ \text{s.t.} \; & \text{(2), (3), (4), (5), (7) and (8).} \end{aligned} \tag{10}$$

where θ_s is the penalty term that weights the difference between the different solutions of the scenario problems.

The overall heuristic is outlined in Algorithm 2, where ρ is the Lagrangian multiplier associated with the non-anticipative constraints, and ε is a stopping parameter for ending the algorithm if $g^{(k)} = p_s \sum_{s=1}^{|S|} \|y^{(k)} - \bar{y}_s{}^{(k)}\|_1$ is small enough. If the condition (9) holds, we say that the solutions have reached a consensus and the algorithm converges. As the reach of the consensus in a mixed-integer problem is not guaranteed, a variable fixing procedure is used if needed.

Algorithm 2 Customized progressive hedging.

```
k := 0
g^(k) := + ∞
ȳ^(k) := 0
θ_s^(k) := 0   ∀ s ∈ S
while g^(k) ≥ ε do
    for s in S do
        y_s^(k) ← Solution of the problem (10).
        θ_s^(k+1) := ρ |ȳ^(k) − y_s^(k)|
    end for
    ȳ^(k+1) := p_s Σ_{s=1}^{|S|} y_s^(k)
    g^(k) := p_s Σ_{s=1}^{|S|} ||y^(k) − ȳ_s^(k)||_1
    k := k + 1
end while
```

In the following, with a slight abuse of notation, PH refers to the described technique.

5. Numerical results

This section reports the numerical experiments for assessing the performance of the proposed heuristics. In particular, we start by presenting the instance generation procedure in Section 5.1. In Section 5.2,

Table 2
Parameters used for instance generation.

Instance type	benchmark	small	medium	large
Max items	100	1000	2000	3000
Known items	50	500	1000	1000
Min volume	3	10	3	3
Max volume	20	15	15	5
Bins available	10	120	60	30
Max bins second stage	10	120	60	30
Min bin capacity	10	10	50	100
Max bin capacity	500	50	100	500

we study the out-of-sample stability of the approaches. Moreover, in Sections 5.4 and 5.5 we present the implementation of the ML heuristic and the feature selection process, respectively. Finally, in Section 5.6, the optimality gaps of the two heuristics with respect to the exact solver are quantified, and in Section 5.7 the comparison between the PH and the ML heuristic on large instances is performed. All the computation experiments are performed on an Intel Core i7-9750H CPU @2.60 GHz. The exact solver used for comparison is Gurobi 9.1.2 (Gurobi Optimization, LLC, 2021) (the source codes are available upon request).

5.1. Instance generation

The instance generator procedure generalizes the one in Crainic et al. (2014a). In particular, for each instance, we consider the following parameters:

- `Max items`: the maximum number of items that must be shipped, i.e., the maximum cardinality of $\mathcal{I}^s$ over all the scenarios;
- `Known items`: the number of items with known volumes (i.e., the one for which $v_i^s = \bar{v}_i,\ \forall s \in \mathcal{S}$);
- `Min volume` and `Max volume`: the minimum and maximum volume of each item;
- `Bins available`: the number of bins available during the first stage;
- `Max bins second stage`: the maximum number of bins in the second stage, i.e., the maximum cardinality of $\mathcal{K}^s$ over all the scenarios;
- `Min bin capacity` and `Max bin capacity`: the maximum and minimum volume of the bin capacity.

In the experiments, four types of instances are considered, namely, `small`, `medium`, `large`, and `benchmark`. The `small` instances represent situations in which each bin may contain at most a few items; this is common in applications such as food delivery, in which the freight is delivered by bikes. The `large` instances model bins that may contain several items; this is a setting when large vans and containers are considered for transportation. The `medium` instances have intermediate characteristics. The `benchmark` instances have the highest variability but the lowest dimension, so they are used to compare the two heuristics with respect to the exact solver and to train the ML heuristics.

The parameters for each type of instance are shown in Table 2.

Given the parameters, the following operations for the instance's generation are performed:

- The number of items for each scenario is sampled from a uniform in [`Known items`, `Max items`].
- For each item, its volume is uniformly sampled in [`Min volume`, `Max volume`] (if the item is deterministic, only one realization is considered for all the scenarios).
- The number of bins available during the second stage is sampled from a uniform in [0, `Max bins second stage`].
- For each bin, its capacity is uniformly sampled in [`Min bin capacity`, `Max bin capacity`].
- The costs for the first-stage bin $j \in \mathcal{J}$ is $C_j = V_j^{2\eta}$, where η is uniformly sampled from $[0.7, 1.3]$ (Crainic et al., 2014a).
- The costs for the second-stage bin $k \in \mathcal{K}^s$ is $C_j = V_j^{2\rho}$, where ρ is sampled uniformly from $[1.4, 1.8]$ (Crainic et al., 2014a).

As discussed in Section 3, we consider model (1)–(8) to be feasible. From a practical point of view, we achieve that by enforcing that the sum of the volumes of the items is smaller than the sum of the volumes of the bins and that the volume of each bin is much greater than the size of each item. These two assumptions are reasonable since in several applications the number of bins available through 3PL providers is enough to contain all the items and the volume of the parcels is smaller than the one of the bins.

5.2. Stability test

Before starting with the experiments, it is necessary to compute the out-of-sample stability of the proposed model (Birge and Louveaux, 1997). The procedure used is standard: we generate S scenarios and use them to solve Model (1)–(8). We then generate $S' \gg S$ new scenarios (called "out-of-sample" because they are not used for the solution computation), and we compute the average cost over the S' scenarios by fixing the first-stage solution obtained. The percentage difference between the optimal value of Model (1)–(8) and the obtained average for different values of $|S|$ is shown in Fig. 2. In particular, for each value of S, we consider $S' = 10000$, and we average the results over a 100 instance of the `benchmark` type. As the reader can notice, 150 scenarios are needed to have a percentage gap lower than 3%. Thus, we set $S = 150$ for the exact method in the following.

The effect of the number of scenarios has to be quantified in the proposed heuristics as well. In general, we expect that the greater the number of scenarios S, the better the solution performance in the out-of-sample test. Therefore, we consider the average gap between the solution obtained using S scenarios and the one obtained by using 5% more scenarios (i.e., $(1+0.05)S$). In other words, we are considering the marginal gain from adding 5% more scenarios to the solution. Figs. 3 and 4 show the results, which were averaged over 300 instances (100 `small`, 100 `medium` and 100 `large`).

For the ML heuristics, 110 scenarios are enough to have a marginal gain from adding more scenarios close to zero. Instead, the PH needs 150 scenarios. In the following, we set $S = 150$ for the PH and $S = 110$ for the ML heuristics.

5.3. Exact method performance analysis

In this section, we analyze the solutions' characteristics and the properties of the mathematical model. In particular, we consider the `benchmark` instance, and we set different numbers of items and bins available in the two stages. For each combination, we compute the following:

- the computational time of the exact solver;
- the fraction of the total cost due to the renting of first-stage bins (first-stage cost);
- the expected value of perfect information ($EVPI$) computed as $EVPI = \frac{RP-WS}{RP}$, where RP is the optimal value of the recourse problem, and WS is the optimal value of the wait-and-see problem (i.e., the problem that assumes perfect knowledge of the future). For the RP computation, we consider 150 scenarios, while we compute the $EVPI$ by using 1000 out-of-sample scenarios;
- the value of the stochastic solution (VSS) computed as $VSS = \frac{EEV-RP}{EEV}$, where EEV is the expected value of using the solution of the expected value problem. Because of the stochasticity influence, the second stage sets $\mathcal{I}^s$ and $\mathcal{K}^s$ to compute the expected value problem, we consider the sets $\bar{\mathcal{I}}$ and $\bar{\mathcal{K}}$ obtained by selecting all the items and bins that are present in more than 50% of the scenarios. As for the $EVPI$, the RP is computed by considering 150 scenarios, and the VSS is computed using 1000 out-of-sample scenarios.

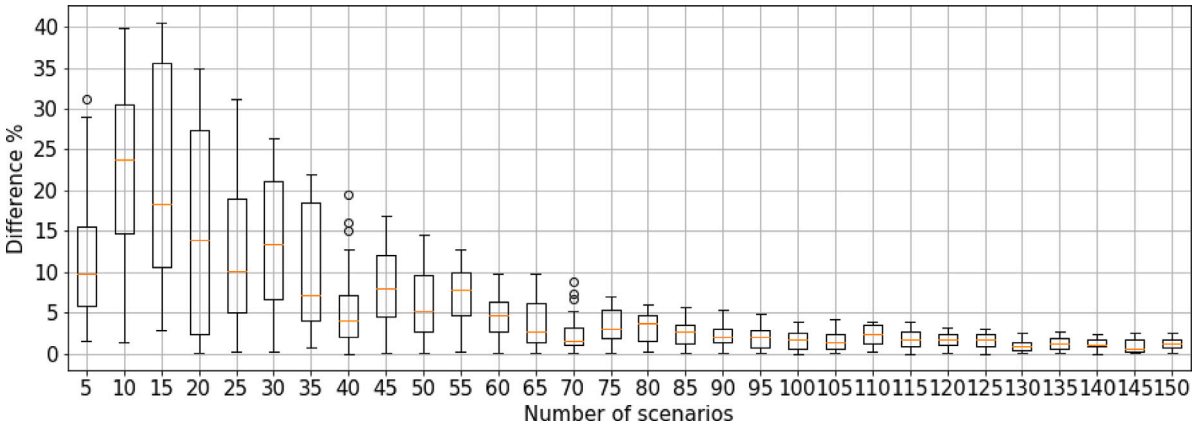


Fig. 2. Exact approach stability test results.

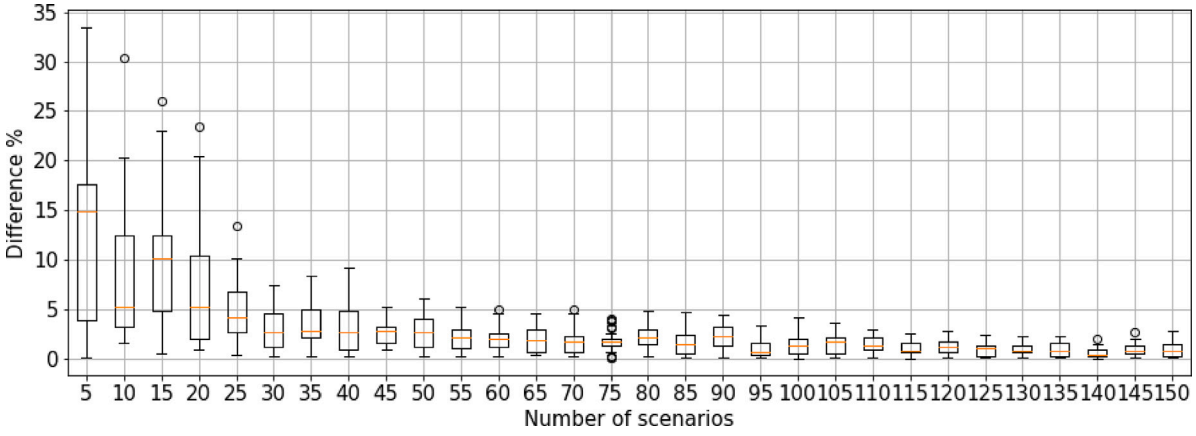


Fig. 3. PH heuristics stability test results.

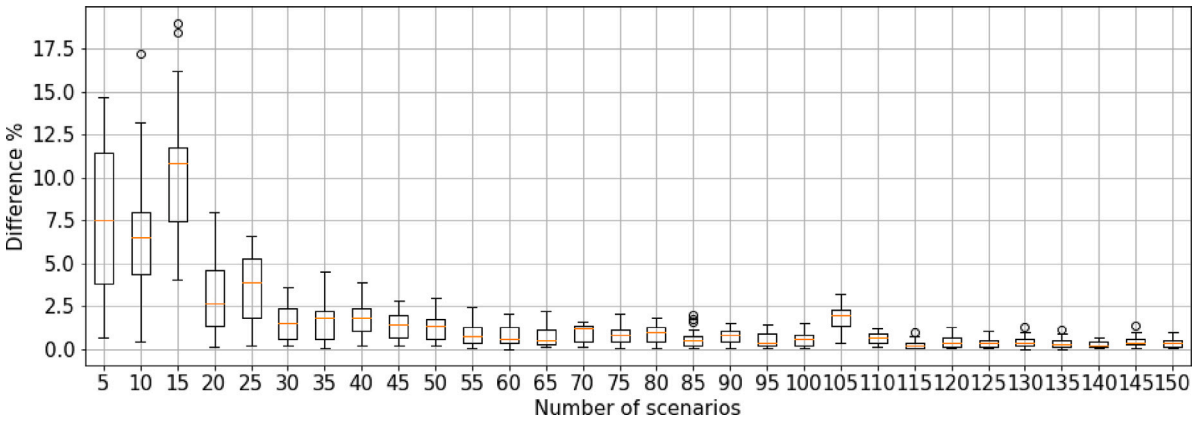


Fig. 4. ML heuristics stability test results.

The obtained results are shown in Table 3. All the values are averaged on 10 instances. As expected, the computational time increases as the number of items and bins increases. We do not solve instances with more than 150 items, 10 first-stage bins, and 10 second-stage bins since the exact solver generates an out-of-memory exception.

It is interesting to note that the more items and second-stage bins that are considered, the more the percentage of the first-stage cost decreases. The reason for this is that waiting to allocate items with a wider set of alternatives in the second stage becomes more convenient. Moreover, having more bins in the second stage increases the probability of a convenient bin appearing, along with the fact that the volume and quantity of the items become fully known in the second stage, making the second-stage bins more attractive. In every instance, 50 items are deterministically generated (as stated in Table 2). Despite this, the first-stage cost is never 100% since, with the exception of a few instances, the first-stage capacity is not enough to pack all the items. As can be seen, $EVPI$ decreases as the number of second-stage bins increases. The same behavior can be observed for the VSS. This effect is due to the lower importance of hedging against risk in the first stage when there are more bins in the second stage. Nevertheless,

Table 3
For a different instance dimension, the computational time of the exact solver, the fraction of the cost given by the first-stage bins, the $EVPI$, and the VSS are computed.

Items	Bins		Comp. time [s]	First-stage cost [%]	EVPI [%]	VSS [%]
	f.s.	s.s.				
50	5	5	4.5 (2.4)	98.8 (0.8)	21.7 (16.8)	129.1 (34.4)
75	5	5	12.4 (3.8)	82.4 (2.4)	16.7 (17.2)	137.4 (54.7)
100	5	5	45.1 (4.2)	75.8 (3.5)	18.3 (13.6)	72.3 (31.1)
125	5	5	129.3 (10.7)	62.1 (3.7)	5.4 (8.8)	81.3 (37.4)
150	5	5	510.2 (21.4)	65.9 (3.4)	9.6 (9.1)	84.0 (32.9)
50	5	10	75.4 (1.8)	95.3 (0.5)	12.4 (11.4)	48.5 (31.2)
75	5	10	198.7 (10.5)	66.9 (5.4)	7.3 (8.1)	53.0 (19.0)
100	5	10	1074.4 (64.6)	67.8 (6.3)	10.7 (10.6)	60.0 (43.9)
125	5	10	1235.7 (91.3)	62.9 (6.5)	11.3 (12.1)	78.6 (43.8)
150	5	10	1384.9 (111.9)	60.9 (5.2)	3.1 (5.8)	53.2 (18.9)
50	10	5	5.0 (3.2)	97.1 (1.3)	26.3 (17.3)	126.3 (74.3)
75	10	5	10.2 (4.2)	84.3 (6.4)	18.9 (15.9)	129.5 (60.8)
100	10	5	14.3 (5.6)	81.3 (5.8)	15.9 (14.0)	116.5 (55.9)
125	10	5	86.5 (11.2)	80.3 (4.3)	16.8 (12.4)	76.5 (27.5)
150	10	5	446.8 (23.7)	77.8 (7.2)	12.6 (11.5)	72.7 (32.0)
50	10	10	50.5 (9.2)	91.8 (2.2)	10.2 (10.2)	53.6 (39.1)
75	10	10	274.7 (15.6)	80.0 (5.1)	10.5 (11.3)	85.8 (56.4)
100	10	10	2028.1 (129.9)	77.5 (5.9)	11.4 (11.0)	60.9 (31.5)
125	10	10	2828.2 (213.5)	67.4 (6.8)	8.6 (7.9)	55.0 (19.8)
150	10	10	3755.0 (401.3)	66.2 (6.6)	9.0 (9.1)	54.2 (20.1)

while $EVPI$ is always lower than 30%, the VSS lowest value is 48.5%, which generates poor performance in the application. This poor performance is also due to the definition of the $EVPI$ for a problem in which uncertainty affects the second-stage sets ($\mathcal{I}^s$ and $\mathcal{K}^s$). Since an ad-hoc discussion of this topic is beyond the scope of the present paper, we will address it in future work.

5.4. Machine learning heuristic

In this section, we explore different configurations for the ML heuristic. We considered 12 different classification approaches for the ML heuristic and compared their performances during the experimental phase. The considered classifiers are as follows:

- **KNN**: The K neighbors classifier checks the distance of the new data from the so called centroid. Then the new data is associated to the closest centroids.
- **L_SVM**: The support vector machine with a linear kernel is a well-known robust classification method, the intuition behind it is to draw a hyper-plane in the data space to maximize the distance between points belonging to different classes identified by this hyper-plane. The standard version of the distance measure is the linear kernel.
- **RBF_SVM**: The support vector machine with a radial basis function (RBF) kernel ia a more sophisticated version of the SVM that uses the kernel trick to identify the distance through the RBF function.
- **GP**: The Gaussian process classifier with an RBF kernel offers the great potential of GP regression adaptability and a closed-form solution.
- **DT**: The decision tree classifier divides a dataset into smaller subsets based on the chosen criteria. The current approach uses the entropy criteria for randomly splitting the data.
- **RF**: The random forest classifier consists of multiple random DT classifiers linked together, which means that RF is an ensemble learning method. Hence, the final class is assigned by the majority of the trees, which preserves the method from overfitting.
- **NN_1l**: The multi-layer perceptron (MLP) classifier with 1 hidden layer of 100 neurons. The number of features defines the input layer size, and the output neuron provides the class identification.
- **NN_ml**: A convolutional neural network (CNN) consisting of 3 hidden layers with a size of $25, 50, 15$ neurons, respectively. The input layer is defined by the number of features for all the feed-forward CNN. This structure of CNN is classified as a deep neural network and can construct more complex functions.
- **AB** The AdaBoost classifier is a meta-estimator, obtained by fitting a chosen classifier (DT in this case) and then making additional copies of one on the same dataset, but with the additional adjustment of the weights for incorrectly classified instances.
- **LR**: Logistic regression is a classifier built on statistical regression, where the output is categorized with a 0 or 1 label, depending on the probability of belonging to a specific class.
- **LDA**: Linear discriminant analysis is represented by the projection of all data points into a line, with the further combination of them into classes based on their distance from a chosen point or centroid. This classifier has the advantage of dimensionality reduction but is best fitted for the linear relations in the data.
- **LSTM**: The long short-term memory network-based classifier is based on recent advancements in the recurrent NN structure, which provides us with short-term memory. A very basic version of LSTM with 1 hidden layer of 50 neurons is applied.

All the outlined approaches were implemented with the help of the Scikit-learn library (Pedregosa et al., 2011) and we set all the methods to return a 0–1 label. In fact, several ML techniques may produce results between 0 and 1 instead of pure 0–1 label. Such algorithms' results can be associated with the probability of a given item being selected in the optimal solution. In such a case, it is possible to obtain a solution to the problem by fixing a threshold on that probability. Nevertheless, this threshold is a hyper-parameter of the algorithm requiring ad-hoc tuning. Thus, for the sake of simplicity, we just consider ML algorithms that return a label postponing the investigation of more general ML techniques to future study.

The training set for the classifiers is generated by solving multiple benchmark problems with the exact solver and using the optimal solutions as labels. The training set used for the experiments includes 400 solved instances, for a total of 4000 records (each benchmark instance has 10 first-stage bins).

Let us consider some visual examples to straighten the classifier application logic and better explain the heuristic. To do so, we pick 200 random samples of training data collected, and we show two feature diagrams in Fig. 5. Each point of the two graphs represents two features related to a first-stage variable y_j, $j \in \mathcal{J}$, and it is colored in blue if, in the optimal solution, $y_j = 1$ and in red if $y_j = 0$.

In Fig. 5(a), the considered features are `reduced costs` and `unitary cost`. If the reduced costs are zero or close to zero (the

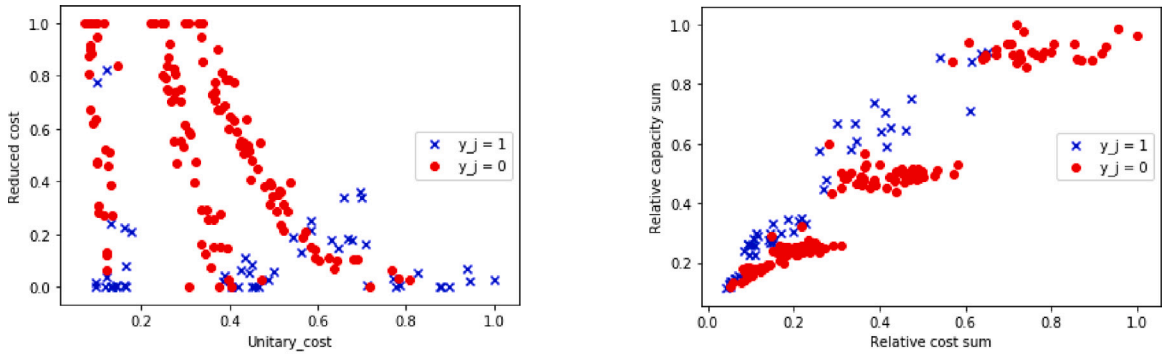


Fig. 5. Feature graphs. Reduced costs vs. unitary cost on the left; relative capacity sum vs. relative cost sum on the right.

bottom part of the graph), there are more points with $y_j = 1$. By moving from the bottom of the graph up to the top (i.e., considering the greater reduced cost), the density of $y_j = 1$ decreases while the density of $y_j = 0$ increases. There are a few points with high reduced costs and $y_j = 1$ (near the point (0.2, 0.8) in Fig. 5(a)). This is due to a particular lower unitary cost for those bins. The behavior with respect to the unitary costs is peculiar: there are several points with low unitary costs with $y_j = 0$ and a consistent number of points with high unitary costs with $y_j = 1$. This is due to the fact that we are not considering several other features that can better characterize the bins on the two-dimensional graphs.

Fig. 5(b) shows the 2D scatter plot of the relative capacity sum and relative cost sum features. There are two regions of the plane almost linearly separable in which the lower part (high cost and low capacity) is characterized by variables with an optimal value of 0 and an upper part (low cost and high capacity) in which the optimal value is 1.

In conclusion, it is important to stress that these graphs are just graphical examples to better explain the intuition behind the ML heuristic and have no other goal.

5.5. Features analysis with SHAP

In this section, we analyze the set of features defined in Section 5.1 to select the most significant ones. To do so, we use the SHapley Additive exPlanations (SHAP) values proposed by Lundberg and Lee (2017). SHAP is a widely used approach for explaining machine learning models based on cooperative game theory. The feature values act as players in a coalition, and Shapley values represent the weights of a fair distribution of the "payout" between them. In other words, it is a measure of the contribution of each of the features to receiving a targeted class. Hence, the mean of SHAP values is essential for this work, as it represents the total contribution to the bin selection, regardless of whether it is positive or negative. Fig. 6 shows the ordered mean SHAP values for each of the introduced features.

This result shows that the decision value of continuous relaxation is an essential feature in our approach, but not the only one contributing to the final outcome. Furthermore, the reduced costs and bin characteristic features provide important information. In contrast, the second stage estimates are the least important in this model, and the Items placed min has no impact on the decision. Therefore, in the following experiments, we consider the 5 features with the greatest SHAP value: Continuous relaxation, Reduced cost, Relative cost max, Relative cost sum, and Items capacity quant. It is worth noting that some of the most important features are the same used for other heuristics (Maggioni and Wallace, 2010). Nevertheless, using the proposed methodology can produce the same results without any prior experience, looking only at the data. Thus, it can be extended to other optimization problems characterized by binary variables.

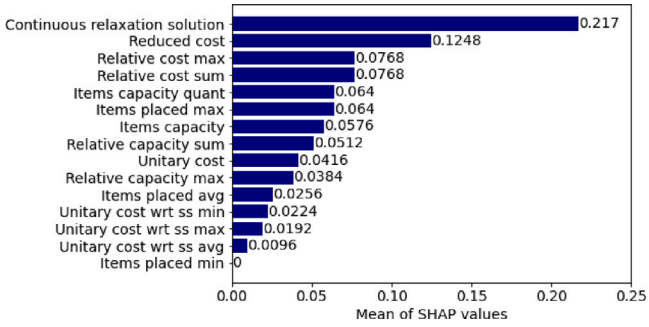


Fig. 6. Mean of the SHAP values for each of the features.

5.6. Heuristics comparison

In this section, we test the performance of the proposed heuristic and the one of the PH, against the commercial solver Gurobi (Gurobi Optimization, LLC, 2021). The results are described in Table 4. All the values are averaged over 500 benchmark instances. The first two columns represent the average computational time and its standard deviation. The second and third columns report the average gap with respect to the optimal solution computed by using 1000 out-of-sample scenarios and their standard deviations. Finally, the last two columns represent the average and standard deviation of the gap with respect to the optimal solution in the first-stage costs.

Both the ML heuristic and the PH significantly decrease the computational time needed in exchange for a reasonable gap. In particular, the ML heuristic is the fastest heuristic method, with an average time of around 4.5 s. The RBF_SVM classifier provides the best gap, with an average of 3.87% and a standard deviation of 3.15%. The L_SVM classifier performs poorly with optimality gaps greater than 20%. This is because the investments in the first stage are too small (23% less than the optimal solution), which leads to a strategy that books the majority of bins in the spot market, leading to the worst average performance and very high variance. This high variance is due to both the lucky case in which a book in the spot market is cheap and the unlucky case in which booking beforehand at a lower price is the optimal strategy. The PH has an average gap lower than the ML, but it took double the time. Other methods that have good performance are the DT and the RF. Both methods achieve reasonable results. Moreover, they are human-readable, making them attractive in the application. To better understand the solution structure, the averaged l_1 distance between the exact first-stage decision variables and the one computed by all heuristic approaches is highlighted in Table 5.

All the heuristics return solutions close to the optimal ones in terms of first-stage solutions, which is an important outcome that validates

Table 4
Comparison of exact and heuristic solutions.

Model	Computing time [s]	Computing time σ [s]	Gap [%]	Gap σ[%]	Gap f.s. [%]	Gap f.s. σ [%]
Exact	1829	2560	–	–	–	–
PH	9.86	2.15	1.75	2.94	4.36	23.67
KNN	4.47	0.81	5.91	7.05	7.49	23.62
L_SVM	4.43	0.76	28.68	56.38	**−23.97**	38.54
RBF_SVM	4.44	**0.74**	**3.87**	3.18	1.48	**15.71**
GP	5.59	1.14	4.88	4.90	7.71	21.39
DT	4.46	0.76	4.15	3.46	5.00	20.86
RF	4.45	0.80	5.29	5.99	2.93	18.20
NN_1l	4.49	0.88	6.13	10.36	2.69	23.44
NN_ml	4.45	0.80	6.44	12.41	7.96	24.25
AB	4.46	0.79	7.80	13.73	−1.19	23.51
LR	4.44	0.81	6.31	7.08	−11.80	21.56
LDA	4.45	0.82	6.31	7.08	−11.80	21.56
LSTM	4.74	0.82	4.03	7.96	−10.93	21.18

Table 5
Solution structure.

Solution	PH	KNN	L_SVM	RBF_SVM	GP	DT	RF	NN_1l	NN_ml	AB	LR	LDA	LSTM
Distance to exact solution	0.646	**0.375**	**1.167**	0.438	0.479	0.458	0.396	0.500	0.417	0.417	0.667	0.667	0.646

these approaches. Nonetheless, a difference in the first-stage solution equal to 1.16 causes the L_SVM method to have 28.68% higher costs in the out-of-sample simulation. Thus, even a small error in the first-stage solution may lead to bad out-of-sample performance.

5.7. ML and PH heuristic comparison

Since the exact solver is not able to solve realistic instances, in this section, we compare the performance of the PH and the ML heuristic. Since the PH was shown to achieve lower gaps in the previous section, we compute the gaps with respect to its performance. Moreover, to better quantify the algorithm performance in the various application domains, we present the results for the three types of instances (`small`, `medium`, and `large`). The results are reported in Table 6, with each value being averaged over 100 runs. The first three columns represent the average computational time of the two methods. The second three columns show the out-of-sample gap calculated on 1000 scenarios, and the last three columns show the average l_1 norm of the distance between the first-stage solutions.

It is important to notice that the computational time increases from `small` to `large` instances. This is due to the fact that `large` instances consider a greater number of items and bins than the other types (Table 2). This leads both the PH and the ML heuristic to take more time (the PH must solve bigger sub-problems, while the ML heuristic must solve larger, continuous problems).

The ML approach generally takes from 5 to 75 times less computational time than the PH. In particular, the ML heuristic takes 5 times less for `small` instances, 15 times less for `medium` instances, and 75 times less for `large` instances. Thus, the proposed methodology provides more advantages if the bins are large with respect to the item size. This means that this heuristic is handy if large vans and containers are considered.

The gaps also depend on the instance type. For `small` instances, the best results are achieved by the GP; for `medium` instances, the LDA classifier is best; for `large` instances, the best classifier is the LSTM. By considering their definition, it is clear that classifiers with more expressive power are required if the complexity of the problem increases. By considering all types of instances, the GP classifiers achieve the best performance in terms of gap and computational time. However, the difference in accuracy for almost all classification approaches lies in the boundaries of 2%. Therefore, in practice, when choosing a classifier, other parameters, such as the interpretability of the decisions, the transparency of the algorithm (as for the decision trees), or the robustness of the application with respect to the industrial setting, may be considered.

FInally, the l_1 distance between the PH and the ML first-stage solutions does not differ by more than 10%. In particular, among the best classifiers, this difference is less than 4%, meaning that the two heuristics compute highly similar solutions. Nevertheless, as stated above, these small differences in solutions may lead to strikingly different performances in the second stage.

In summary, the proposed ML approach showed to be very promising, not only in terms of computational time but also in terms of the optimality gap.

6. Case study and managerial insights

In this section, we analyze the potential impact of the usage of the VCSBPPSI solved via the ML heuristic and present relevant managerial insights on a case study coming from recent industrial and institutional collaborations. We opted for a last-mile application due to its increasing importance in terms of economic value and number of orders. In fact, the rapid increase of e-commerce is making last-mile delivery one of the most challenging fields of development and research in transportation. This research is part of the new Logistics and Mobility Plan to be activated in 2025 and considers the effect of same-day deliveries (Perboli et al., 2021a,b). Sensitive data for the involved e-commence and parcel delivery companies have been anonymized and normalized by means of the data-fusion tool provided in Perboli et al. (2018b).

The case study tackles a set of deliveries (characterized by delivery locations and parcel volumes) within the Turin city center area (2.805×2.447 km^2), using a heterogeneous fleet of three types of vehicles: Cargo-Bikes (CBs), Electric Vehicles (EVs), and Light Duty vehicles (LDs). Fig. 7 displays the service area (red square) and an example of the location of customers (blue circles). We consider 40 instances with 500 deliveries each. Since these deliveries have no information related to their delivery time, we generate the second-stage orders by randomly picking a percentage of deliveries that we interpret to be originated from same-day orders. More in detail, we call this percentage α and we consider $\alpha \in \{10\%, 30\%, 50\%, 70\%, 100\%\}$. Scenarios are generated by applying the same process used for instance generation. For each one of the 40 instances and each value of α, 10, instances are generated for a total of $40 \times 5 \times 10 = 2000$ instances.

We compare the operational costs (i.e., the total cost of satisfying all the deliveries) of the consolidation strategy considered in the VCSBPPSI, solved by the ML heuristic using GP classifier, against a single-echelon approach, which adopts the state-of-the-art stochastic vehicle routing problem by Saint-Guillain et al. (2015), where a fleet of LD vehicles is used. The parameters for the two models are taken from

Table 6
Resulting computational time, percentage gap, and the l_1 distance to the PH solution of the real scale problem test.

Instance	Computational time [s]			Gap to PH solution			l_1 distance to PH solution		
	small	medium	large	small	medium	large	small	medium	large
PH	177.1	2276	26076	–	–	–	–	–	–
KNN	34.00	147.06	344.33	1.22%	1.99%	4.73%	1.227	2.063	3.800
L_SVM	34.03	146.89	348.11	2.16%	2.50%	6.87%	1.600	2.125	3.400
RBF_SVM	33.91	146.33	348.06	3.09%	1.84%	5.35%	3.727	3.813	8.100
GP	35.07	147.01	344.53	**0.03%**	2.20%	4.73%	1.200	2.125	3.800
DT	33.93	146.25	340.57	1.57%	2.01%	4.73%	1.273	2.063	3.800
RF	33.84	146.10	341.53	0.42%	2.01%	4.73%	1.545	2.063	3.800
NN_1l	33.85	146.35	353.53	1.22%	2.20%	6.12%	1.227	2.125	3.700
NN_ml	33.96	146.62	345.46	1.22%	1.99%	4.73%	1.227	2.063	3.800
AB	33.89	146.08	347.38	4.06%	2.68%	6.45%	4.391	4.325	9.700
LR	33.88	143.75	340.09	0.91%	2.07%	5.00%	1.045	2.375	**3.000**
LDA	33.88	142.89	341.93	0.91%	**1.69%**	5.53%	1.045	2.313	3.100
LSTM	34.34	147.92	347.68	1.02%	2.53%	**4.61%**	**0.818**	2.188	3.400

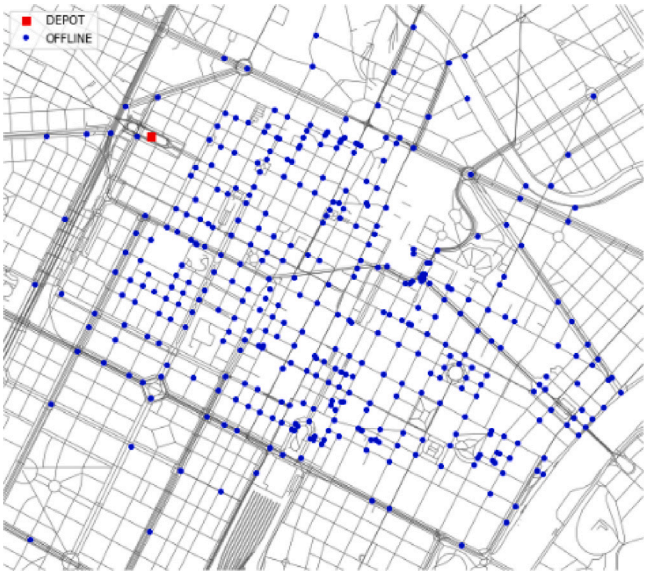


Fig. 7. Service area in the case study.

previous works and are summarized in Table 7. It is worth noting that we compare two different models: the VCSBPPSI, which considers only consolidation and a single-echelon routing problem. This comparison makes sense in an urban environment since, due to the small distances, routing has a limited effect on the final solution (Perboli et al., 2021a). In other words, in city logistic capacitated routing problems, capacity constraints are the binding ones. Therefore, the comparison of the solutions of these two models enables us to quantify the gain of consolidation against standard routing strategies often used in practice. Moreover, the results that we provide are lower bounds on the real efficiency of consolidation since the VCSBPPSI does not address routing and better solutions can be obtained by merging the two problems.

The comparison of the solution provided by the two problems is performed by a Monte Carlo-based simulation framework which is made by a module for simulating the instance, one for georeferencing the data and one for simulating the given routes with real traffic data gathered by the network of sensors of the Municipality of Turin provided by the company 5T (5T Web Site, 2019). The proposed Monte Carlo-based simulation algorithm runs the following steps 10 times (see Fig. 8 for a depiction of the overall system):

1. Generate the instance by setting α and all the aforementioned parameters.
2. Solve the first stage problem:
   - run it and compute the mix of vehicles for the first stage fleet, if the VCSBPPSI is considered
   - consider a fixed fleet of LD vehicles if the single-echelon problem is considered
3. Given the first stage solution, solve the related second stage by adjusting the fleet according to the new deliveries and compute the route running:
   - the heuristic in Saint-Guillain et al. (2015) applied to several TSP problems, if the VCSBPPSI is considered
   - the stochastic VRP heuristic in Saint-Guillain et al. (2015), if the single-echelon problem is consider
4. Compute the Key Performance Indicators related to the quality of service (in terms of the number of parcels per hour, nD/h) and the environmental cost (in terms of CO_2 emissions of the overall last-mile chain) for each route. In particular, according to the latest regulation, the ISO/TS 14067:2013 "Greenhouse gases - Carbon footprint of product - Requirements and guidelines for quantification and communication", we consider the sum of direct emissions from the fuel combustion process, indirect emissions, emitted by the fuel production process and the long-haul shipment of the fuel, CO_2 equivalent to including other pollutants (e.g., NOx).

Considering the computational effort, each run of the Monte Carlo requires about 40 minutes when we solve the optimization problem by the dynamic and stochastic vehicle routing algorithm by Saint-Guillain et al. (2015), while about 1 minute in the case of the ML approach. Notice that the computational time also accounts for data fusion and post-optimization modules. Thus, our ML approach has a significant advantage over the classic dynamic and stochastic vehicle routing problem. This issue becomes more relevant if larger instances are considered.

Table 8 reports the average percentage gap between the total cost of the single-echelon policy versus the consolidation for different values of α. The gap is defined by $(SE-SAT)/SE$, where SE and SAT are the expected costs computed by the Monte Carlo simulation of the single-echelon policy and of the consolidation, respectively. Thus, a positive value means a gain in the percentage of the consolidation policy with respect to the single-echelon one. It is worth noting that a consolidation policy always has better performance than a single-echelon one (up to 67% of cost saving). This is further evidence that at the urban level, routing is less important than consolidation (Perboli et al., 2021a).

In Table 9 we illustrate the effect of same-day deliveries on the total expected cost computed with the Monte Carlo simulation. With this aim, we simulate the behavior of the VCSBPPSI model, solved considering all deterministic deliveries ($\alpha = 0$), and in each column, we report the average increase cost with respect to the case with $\alpha = 0$

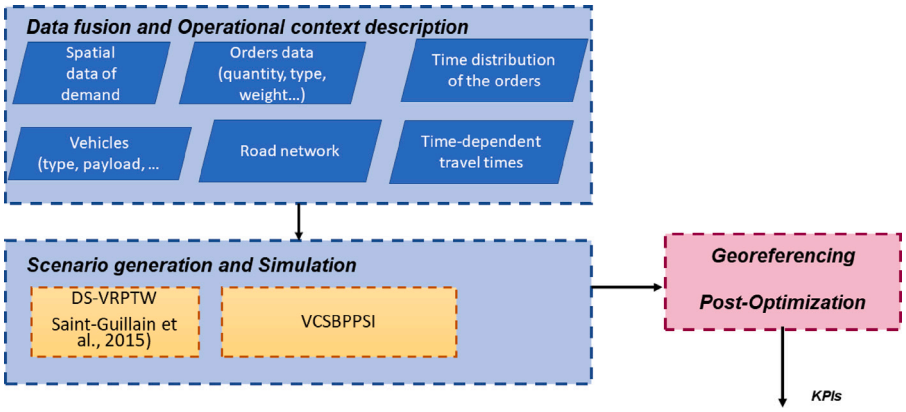


Fig. 8. Monte Carlo simulation–optimization.

Table 7
Monte Carlo simulation-optimization architecture.

Data type	Source
Satellite localization	Municipality of Turin (2018), Perboli et al. (2018b)
Orders data	Brotcorne et al. (2019), Crainic et al. (2011)
Spatial data of demand	De Marco et al. (2017)
Time distribution of the orders	De Marco et al. (2017)
Vehicles characteristics and vehicle booking costs	Perboli et al. (2018b), Perboli and Rosano (2019)
Road network	OpenStreet, 5T Road Sensors' data (5T Web Site, 2019)
Time-dependent travel times	Maggioni et al. (2014)
Environmental costs	Brotcorne et al. (2019), Giusti et al. (2019)

Table 8
Percentage gain of the consolidation policy with respect to the single-echelon one.

α:	10%	30%	50%	70%	100%
$\frac{SE-SAT}{SE}$:	3.5	23.2	31.7	50.2	67.6

Table 9
Cost increase due to a lack of customers' preferences analysis.

α:	10%	30%	50%	70%	100%
$EVPI_\alpha$:	13.4	47.2	61.4	87.4	137.9

computed in percentage. Noticing that with $\alpha = 0$ the model has perfect information, thus we call the average percentage gap $EVPI_\alpha$. Having perfect knowledge about the deliveries can decrease the delivery costs up to 137% ($\alpha = 100\%$). This justifies investments in methods able to forecasts deliveries or in business models, which reduces the impact of uncertainty. Nevertheless, even with a rather limited impact of same-day delivery ($\alpha = 30\%$), the cost increment can be sufficiently high to require a specific redesign of the business model towards the integration of big data and prescriptive analytics. As shown in Perboli and Rosano (2019), allowing full freedom to the customers without any prevision on the customers' preferences may cost the e-commerce company between 0.7 and 1.2 million euros per year in the case of a medium-sized city such as Turin. While such inefficiency can still be accepted in the present situation, where the e-commerce market is growing by two digits per year, this becomes unacceptable in a more saturated market situation, where the innovation curve moves towards the full competition phase and efficiency becomes a key factor for a company to be in the business.

Table 10 shows the usage of the vehicle for different values of α. Each row reports the average number of vehicles of each type for the first stage, second stage, and the total. Notice that, being the averages of the values, they can be fractional.

If same-day delivery is limited ($\alpha \leq 50\%$), a larger portion of CBs is used in the second stage. This is not due to the unitary cost per volume (which is larger for the greater impact of the freight dispersion and the consequent under-usage of the volume), but to the flexibility of CBs, able to better absorb the effects of the first-stage decisions. This explains the high investments of Venture Capital in alternative and small-sized delivery options, such as drones, small robots, cargo bikes, and other similar options. Instead, when $\alpha > 50\%$, the number of first-stage LDs and EVs booked increases, since booking big vehicles in the first stage is cheaper, and with a large number of second-stage deliveries, the probability of filling them is high. This characteristic of hedging against uncertainty makes the investment in LDs and EVs attractive for the management of the increasing impact of customers' preferences.

Finally, we analyze the two paradigms in terms of sustainability. The sustainability of the service is computed as a mix of environmental, social, and operational impacts. We consider the case of three different scenarios, according to the Moore technology adoption curve (Moore, 2014):

- Early phase of the penetration of the same-day delivery service. It corresponds to the middle of the "scale-up" phase of the innovation sigmoid, which corresponds to $\alpha = 10\%$;
- Market penetration of the same-day delivery service. It corresponds to the beginning of the "compete" phase of the innovation sigmoid, which corresponds to $\alpha = 50\%$;
- Maturity of the same-day delivery service. It corresponds to the end of the "compete" phase of the innovation sigmoid, which corresponds to $\alpha = 70\%$.

For each scenario we consider three market situations:

- Current situation: the instances described above;
- Downturn: the market is contracting up to 30% (only 70% of the whole set of delivery is considered);
- Growth: the market is increasing up to 30% (the new delivery are randomly generated as in Perboli and Rosano, 2019).

Table 10
Effect of the customers' choices over vehicle usage.

Stage	10%			30%			50%			70%			100%		
	CB	EV	LD	CB	EV	LD	CB	EV	LD	CB	EV	LD	CB	EV	LD
1 stage	0.01	1.29	7.17	0.01	1.26	7.21	0.04	1.68	7.83	0.04	2.94	13.09	0.11	4.27	15.86
2 stage	26.40	0.00	0.00	26.43	0.00	0.00	26.50	0.00	0.00	14.93	0.00	0.00	7.95	0.00	0.00
Total	26.41	1.29	7.17	26.44	1.26	7.21	26.54	1.68	7.83	14.96	2.94	13.09	8.06	4.27	15.86

Table 11
Sustainability analysis.

$\alpha = 10\%$		
Market condition	CO_2 savings [ton (%)]	nD/h [%]
Current situation	18 (−21%)	11%
Downturn	13 (−18%)	5%
Growth	19 (−36%)	19%
$\alpha = 50\%$		
Market condition	CO_2 savings [ton (%)]	nD/h [%]
Current situation	22 (−34%)	12%
Downturn	16 (−24%)	9%
Growth	24 (−38%)	14%
$\alpha = 70\%$		
Market condition	CO_2 savings [ton (%)]	nD/h [%]
Current situation	26 (−30%)	15%
Downturn	24 (−21%)	11%
Growth	36 (−40%)	18%

Table 11 reports the results of the sustainability analysis in terms of CO_2 savings (Column 2) and nD/h (Column 3). All the savings are reported in terms of percentage gap with respect to the single-echelon scenario. For both indicators, the savings are computed by considering 360 working days. For CO_2 we also report the tons gained by the mix of technology.

Generally speaking, the adoption of the VCSBPPSI leads to a consistent decrease of CO_2 compared to the traditional delivery, with a gain up to 40% in the case of the largest diffusion of the service. Notice that, being this phase associated with the "compete" phase of technological penetration, this reduction becomes more crucial since it gives the companies adopting such a scheme in the early phase of their life, a competitive advantage in terms of value proposition, with a more environmental-friendly impact. The nD/h increases, in line with the results by Perboli and Rosano (2019), with an efficiency gain that is quite constant. Finally, we can notice how the need for more flexible solutions is in line with the increasing adoption of more eco-friendly solutions, such as cargo bikes and electric vans in the present, drones, small automatic vehicles, and automated mobile lockers in the near future.

7. Conclusions and future work

This paper introduces a new general ML heuristic for solving the VCSBPPSI. By carefully defining a set of features, it can compute good solutions in a significantly shorter time than the off-the-shelf solvers and other state-of-the-art heuristics such as the PH. This result shows that it is possible to mimic the behavior of the VCSBPPSI with an ML algorithm that behaves similarly to a complex decision rule. This paves the way to the interesting research question of generalizing the good results obtained by the precedence rule in scheduling problems for other classes of problems with more complex decision tools.

Moreover, the proposed heuristic can achieve good performance and increase the maximum size of instances that can be considered, with an impact also in real-world applications.

Finally, the easy application of the proposed heuristic enables it to be used in other problems characterized by binary decision variables.

As a final contribution, we can claim that the proposed technique enables daily managerial decision support for companies engaged in the modern trends of consolidated logistics. In future studies, the performance of the outlined approach can be improved by introducing real data features for each of the vehicles, e.g., CO_2 emissions and the time of delivery. Moreover, in future studies we will consider further improving the proposed methodology. Other interesting lines of research can be to add into the model other realistic issues such as 3D constraints, weight restrictions, as well as discounts and tariffs definition.

CRediT authorship contribution statement

Maria Elena Bruni: Use case, Validation. **Edoardo Fadda:** Conceptualization, Methodology, Validation, Writing, Supervision, Project administration. **Stanislav Fedorov:** Sofware. **Guido Perboli:** Conceptualization, Methodology, Validation, Formal analysis, Writing, Use case.

Data availability

Data will be made available on request.

Acknowledgments

While working on this paper, prof. Perboli was the head of the Urban Mobility and Logistics Systems (UMLS) initiative of the interdepartmental Center for Automotive Research and Sustainable mobility (CARS) at Politecnico di Torino, Turin, Italy, the representative for Politecnico di Torino in the Logistic Table in the Regional Council of Piedmont, and Chief Scientific Officer of Arisk S.p.A., Deeptech company operating Risk Management performed by means of Artificial Intelligence.

Prof. Maria Elena Bruni acknowledges financial support from: PNRR MUR project PE0000013-FAIR.

References

5T Web Site, 2019. 5T srl. URL http://www.cars.polito.it/. (Last Accessed, 03 August 2020).

Achamrah, F.E., Riane, F., Limbourg, S., 2021. Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. Int. J. Prod. Res. 60 (20), 6187–6204. http://dx.doi.org/10.1080/00207543.2021.1987549.

Baldo, A., Boffa, M., Cascioli, L., Fadda, E., Lanza, C., Ravera, A., 2023. The polynomial robust knapsack problem. European J. Oper. Res. 305 (3), 1424–1434. http://dx.doi.org/10.1016/j.ejor.2022.06.029.

Bengio, Y., Frejinger, E., Lodi, A., Patel, R., Sankaranarayanan, S., 2020. A learning-based algorithm to quickly compute good primal solutions for stochastic integer programs. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer, pp. 99–111.

Birge, J.R., Louveaux, F., 1997. Introduction to stochastic programming. In: Springer Series in Operations Research.

Brotcorne, L., Perboli, G., Rosano, M., Wei, Q., 2019. A managerial analysis of urban parcel delivery: A lean business approach. Sustainability 11, 3439.

Cagliero, L., Russis, L.D., Farinetti, L., Montanaro, T., 2018. Improving the effectiveness of SQL learning practice: A data-driven approach. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference, Vol. 1. COMPSAC, IEEE, pp. 980–989. http://dx.doi.org/10.1109/compsac.2018.00174.

Capocasale, V., Gotta, D., Musso, S., Perboli, G., 2021. A Blockchain, 5G and IoT-based transaction management system for Smart Logistics: An Hyperledger framework. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference. COMPSAC, IEEE, pp. 1285–1290. http://dx.doi.org/10.1109/COMPSAC51774.2021.00179.

Chen, T., Chen, X., Chen, W., Heaton, H., Liu, J., Wang, Z., Yin, W., 2021. Learning to optimize: A primer and a benchmark. http://dx.doi.org/10.48550/ARXIV.2103.12828, URL https://arxiv.org/abs/2103.12828.

Correia, I., Gouveia, L., da Gama, F.S., 2008. Solving the variable size bin packing problem with discretized formulations. Comput. Oper. Res. 35 (6), 2103–2113. http://dx.doi.org/10.1016/j.cor.2006.10.014.

Crainic, T.G., Gobbato, L., Perboli, G., Rei, W., 2016. Logistics capacity planning: A stochastic bin packing formulation and a progressive hedging meta-heuristic. European J. Oper. Res. 253 (2), 404–417.

Crainic, T.G., Gobbato, L., Perboli, G., Rei, W., Watson, J.-P., Woodruff, D.L., 2014a. Bin packing problems with uncertainty on item characteristics: An application to capacity planning in logistics. Procedia-Soc. Behav. Sci. 111, 654–662.

Crainic, T.G., Hewitt, M., Rei, W., 2014b. Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. Comput. Oper. Res. 43, 90–99.

Crainic, T.G., Perboli, G., Rei, W., Tadei, R., 2011. Efficient lower bounds and heuristics for the variable cost and size bin packing problem. Comput. Oper. Res. 38 (11), 1474–1482.

Crainic, t., Perboli, G., Ricciardi, N., 2021. City logistics. In: Crainic, t., Gendreau, M., Gendron, B. (Eds.), Network Design with Applications in Transportation and Logistics. Springer, Boston, pp. 507–537.

da Costa, P., Rhuggenaath, J., Zhang, Y., Akcay, A., Kaymak, U., 2021. Learning 2-opt heuristics for routing problems via deep reinforcement learning. SN Comput. Sci. 2 (5), http://dx.doi.org/10.1007/s42979-021-00779-2.

De Marco, A., Mangano, G., Zenezini, G., Cagliano, A.C., Perboli, G., Rosano, M., Musso, S., 2017. Business Modeling of a City Logistics ICT Platform. In: Proceedings - International Computer Software and Applications Conference, Vol. 2. pp. 783–789.

Dumouchelle, J., Patel, R., Khalil, E.B., Bodur, M., 2022. Neur2SP: Neural two-stage stochastic programming. http://dx.doi.org/10.48550/ARXIV.2205.12006.

Fadda, E., Fedorov, S., Perboli, G., Barbosa, I.D.C., 2021. Mixing machine learning and optimization for the tactical capacity planning in last-mile delivery. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference. COMPSAC, IEEE, pp. 1291–1296.

Fadda, E., Perboli, G., Tadei, R., 2019. A progressive hedging method for the optimization of social engagement and opportunistic IoT problems. European J. Oper. Res. 277 (2), 643–652. http://dx.doi.org/10.1016/j.ejor.2019.02.052.

Garey, M.R., 1979. A guide to the theory of NP-completeness. Comput. Intractability.

Giusti, R., Iorfida, C., Li, Y., Manerba, D., Musso, S., Perboli, G., Tadei, R., Yuan, S., 2019. Sustainable and de-stressed international supply-chains through the SYNCHRO-NET approach. Sustainability 11 (4), 1083. http://dx.doi.org/10.3390/su11041083.

Gurobi Optimization, LLC, 2021. Gurobi Optimizer Reference Manual. URL https://www.gurobi.com.

Jiménez-Cordero, A., Morales, J.M., Pineda, S., 2022. Warm-starting constraint generation for mixed-integer optimization: A machine learning approach. Knowl.-Based Syst. 253, 109570. http://dx.doi.org/10.1016/j.knosys.2022.109570.

Kwak, S.Y., Cho, W.S., Seok, G.A., Yoo, S.-G., 2020. Intention to use sustainable green logistics platforms. Sustainability 12 (8), 3502.

Larsen, E., Lachapelle, S., Bengio, Y., Frejinger, E., Lacoste-Julien, S., Lodi, A., 2018. Predicting tactical solutions to operational planning problems under imperfect information. arXiv preprint arXiv:1807.11876.

Li, Y., Zhang, J., Yu, G., 2020. A scenario-based hybrid robust and stochastic approach for joint planning of relief logistics and casualty distribution considering secondary disasters. Transp. Res. Part E: Logist. Transp. Rev. 141, 102029. http://dx.doi.org/10.1016/j.tre.2020.102029.

Lodi, A., Zarpellon, G., 2017. On learning and branching: A survey. TOP 25 (2), 207–236. http://dx.doi.org/10.1007/s11750-017-0451-6.

Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 4768–4777.

Maggioni, F., Perboli, G., Tadei, R., 2014. The multi-path traveling salesman problem with stochastic travel costs: Building realistic instances for city logistics applications. Transp. Res. Procedia 3, 528–536.

Maggioni, F., Wallace, S.W., 2010. Analyzing the quality of the expected value solution in stochastic programming. Ann. Oper. Res. 200 (1), 37–54. http://dx.doi.org/10.1007/s10479-010-0807-x.

Malagnino, A., Montanaro, T., Lazoi, M., Sergi, I., Corallo, A., Patrono, L., 2021. Building information modeling and internet of things integration for smart and sustainable environments: A review. J. Clean. Prod. 312, 127716. http://dx.doi.org/10.1016/j.jclepro.2021.127716.

Mele, U.J., Gambardella, L.M., Montemanni, R., 2021. Machine learning approaches for the traveling salesman problem: A survey. In: 2021 the 8th International Conference on Industrial Engineering and Applications(Europe). ACM, pp. 182–186. http://dx.doi.org/10.1145/3463858.3463869.

Miki, S., Ebara, H., 2019. Solving traveling salesman problem with image-based classification. In: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence. ICTAI, IEEE, pp. 1118–1123. http://dx.doi.org/10.1109/ictai.2019.00156.

Mirshekarian, S., Sormaz, D., 2018. Machine learning approaches to learning heuristics for combinatorial optimization problems. Procedia Manuf. 17 (2), 102–109. http://dx.doi.org/10.1016/j.promfg.2018.10.019.

Montemanni, R., D'ignazio, F., Chou, X., Gambardella, L.M., 2018. Machine learning and Monte Carlo sampling for the probabilistic orienteering problem. In: 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems. ISIS, IEEE, pp. 14–18. http://dx.doi.org/10.1109/scis-isis.2018.00014.

Moore, G.A., 2014. Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers, third ed. Collins Business Essentials.

Municipality of Turin, 2018. Torino living lab. URL http://torinolivinglab.it/en/. (Last Accessed 11 March 2023).

Nair, V., Dvijotham, D., Dunning, I., Vinyals, O., 2018. Learning fast optimizers for contextual stochastic integer programs. In: Globerson, A., Silva, R. (Eds.), Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence. UAI 2018, Monterey, California, USA, August 6-10, 2018, AUAI Press, pp. 591–600, URL http://auai.org/uai2018/proceedings/papers/217.pdf.

Ning, C., You, F., 2019. Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. Comput. Chem. Eng. 125, 434–448.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Perboli, G., Brotcorne, L., Bruni, M.E., Rosano, M., 2021a. A new model for last-mile delivery and satellite depots management: The impact of the on-demand economy. Transp. Res. Part E: Logist. Transp. Rev. 145, 102184.

Perboli, G., Gobbato, L., Maggioni, F., 2017a. A progressive hedging method for the multi-path travelling salesman problem with stochastic travel times. IMA J. Manag. Math. 28 (1), 65–86.

Perboli, G., Gobbato, L., Perfetti, F., 2014. Packing problems in transportation and supply chain: New problems and trends. Procedia-Soc. Behav. Sci. 111, 672–681.

Perboli, G., Musso, S., Rosano, M., 2018a. Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. IEEE Access 6, 62018–62028. http://dx.doi.org/10.1109/ACCESS.2018.2875782.

Perboli, G., Musso, S., Rosano, M., Tadei, R., Godel, M., 2017b. Synchro-modality and slow steaming: New business perspectives in freight transportation. Sustainability 9 (10), 1843.

Perboli, G., Rosano, M., 2019. Parcel delivery in urban areas: Opportunities and threats for the mix of traditional and green business models. Transp. Res. C 99, 19–36. http://dx.doi.org/10.1016/j.trc.2019.01.006.

Perboli, G., Rosano, M., 2020. A taxonomic analysis of smart city projects in North America and Europe. Sustainability 12, 7813. http://dx.doi.org/10.3390/su12187813.

Perboli, G., Rosano, M., Saint-Guillain, M., Rizzo, P., 2018b. Simulation–optimisation framework for city logistics: An application on multimodal last-mile delivery. IET Intell. Transp. Syst. 12 (4), 262–269. http://dx.doi.org/10.1049/iet-its.2017.0357.

Perboli, G., Rosano, M., Wei, Q., 2021b. A simulation-optimization approach for the management of the on-demand parcel delivery in sharing economy. IEEE Trans. Intell. Transp. Syst..

Rockafellar, R.T., Wets, R.J.B., 1991. Scenarios and policy aggregation in optimization under uncertainty. Math. Oper. Res. 16 (1), 119–147.

Saint-Guillain, M., Deville, Y., Solnon, C., 2015. A multistage stochastic programming approach to the dynamic and stochastic VRPTW. In: 12th Int. Conf. Integration of AI and OR Techniques in Constraint Programming. CPAIOR 2015, pp. 357–374.

Sergi, I., Montanaro, T., Benvenuto, F.L., Patrono, L., 2021. A smart and secure logistics system based on IoT and cloud technologies. Sensors 21 (6), 2231. http://dx.doi.org/10.3390/s21062231.

Talbi, E.-G., 2020. Machine learning into metaheuristics: A survey and taxonomy of data-driven metaheuristics. working paper or preprint. URL https://hal.inria.fr/hal-02745295.