Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (35.th cycle)

Urban Logistics and Last Mile applications: Models and methods to deal with demand uncertainty

Stanislav Fedorov

* * * * *

Supervisor
Prof. Guido Perboli

Doctoral examination committee
Prof. Teodor Gabriel Crainic, Université du Québec à Montréal, Canada
teodorgabriel.crainic@cirrelt.net
Prof. Maria Elena Bruni, Università della Calabria, Italy
mariaelena.bruni@unical.it

Politecnico di Torino
June 8, 2023

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . .

Stanislav Fedorov

Turin, June 8, 2023

Summary

Urbanization and the development of complex transportation infrastructure in the cities present new challenges in urban logistics and parcel delivery problems. Thus, it has been subject to a significant shift in scientific interest over the last decades. Furthermore, with the rapid evolution of delivery services and the e-commerce market, the delivery business model has shifted from offer-driven to demand-driven, with the product frequently becoming available after the order is placed. This shift becomes possible due to the introduction of novel management approaches, including fleet consolidation and third-forth party logistics (3-4PL). However, real-world uncertainty strongly impacts their effectiveness since once the actual demand becomes known, the cost of changing the logistic scheme is never negative. Therefore, optimizing the logistics scheme on a longer time horizon is required, which is also called tactical capacity planning.

This thesis investigates the feasibility of introducing robust tactical planning approaches to assist decision-making under demand uncertainty. Recent approaches to this problem are based on stochastic programming and bin-packing problems as the baseline mathematical framework. However, multiple stochastic variables in the detailed problem description lead to exponential complexity growth and computational limitations related to the real-case problem scenarios. Therefore, we explore the possibilities of introducing (meta-) heuristics and practical frameworks to predict demand, enabling decision-making tools and online platforms on the tactical planning horizon. Data availability and recent advances in machine learning (ML) and deep learning models are exploited to support this process. Furthermore, analysis of the historical demand provides valuable support to the application of ML to demand prediction or even first-stage solutions of physical and temporal capacity allocation (fleet requirements).

As a result, we developed a framework for incorporating ML into existing, well-stated optimization problems. Furthermore, we created a new problem suitable for the logistics managers' actual needs for using 3-4PL. The main scientific impact of this thesis is that we kept involved optimization problems to be deterministic or with low complexity, drastically reducing computational time while threatening demand uncertainty and remaining adaptable to new incoming information. The industrial significance consists in enabling the usage of the so-called intelligence

logistics platforms by providing a fast and accurate solution for realistic problem instances comparable to the actual situation in large urban agglomerations.

Acknowledgements

I would like to express my deepest gratitude to my Ph.D. supervisor, Prof. Guido Perboli, for his invaluable patience, supporting the research ideas, and understanding. His short and intuitive remarks give me the idea of successful academic, business, and industry research.

I want to acknowledge the Polytechnic of Turin and the DAUIN department for their financial and administrative support. It has been an honor to be the inaugural recipient Ph.D. scholarship, and I am indebted for providing me with complete academic freedom in this research.

I am also grateful to Edoardo Fadda for his constructive criticism, editing help, and timely feedback sessions. I also could not have undertaken this journey without his support and interpretation of my initiatives.

Words cannot express my gratitude to Prof. Teodor Gabriel Crainic, who pointed out the excellent idea of concentrating on demand uncertainty treatment. His experience and understanding of the academy and industry interaction deserve worldwide recognition.

I would like to extend my sincere thanks to the collaborators for lending me their expertise and intuition to my scientific problems and sharing the essential data: Ivan Dario Cardenas and Joris Beckers from the University of Antwerp, Prof. Maria Elena Bruni from the University of Calabria.

I want to acknowledge the CARS@PoliTO lab as a great community to share and support the research experience.

Finally, I'm incredibly grateful to my parents and friends for their infinite love, motivation, and support during this journey.

Dedicated to people helped me to face war, love, and research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Urbanization and the development of complex transportation infrastructure in the cities present new challenges in urban logistics. and rising living standards have dramatically increased demand for services and goods in relevantly small areas. These conditions have paved the way for a competitive environment where logistic companies fight for market share by continuously providing flexibility in delivering options while maintaining high resource efficiency.

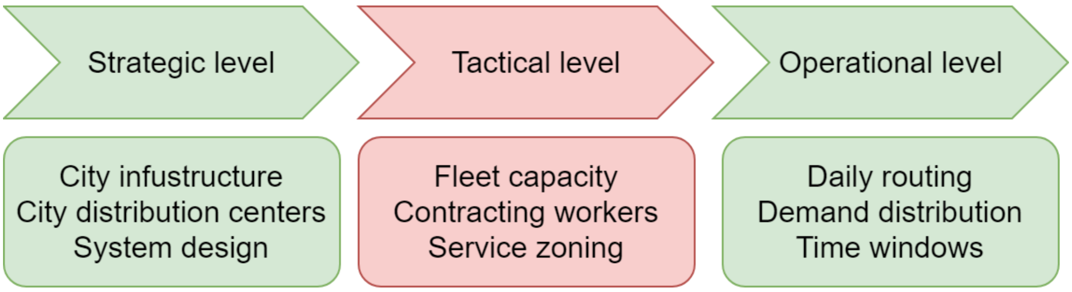


Figure 1.1: Last mile managerial decision levels

Third-party logistics (3PL) becomes an essential part of modern delivery schemes since it allows fleet flexibility and translates into the advantage of reducing fleet investment while maintaining the same quality of service [92]. , which should be optimized under conditions of uncertain demand, i.e., the customers' orders are unknown or only partially known.

In the existing literature, specific models are developed for each decision level.

The crucial In BPP, we want to optimally pack a set of items (goods) into a set of containers (vehicles). Later on, it evolved to account for the different costs and volumes of vehicles as Variable Cost and Size Bin Packing Problem (VCSBPP) [30]. However, the actual demand uncertainty treatment is provided with the involvement of Stochastic Items, namely VCSBPPSI [27]. The bins included in the capacity plan are chosen in advance without exact knowledge of what items will be dispatched. If the planned capacity is insufficient, extra capacity is purchased at a higher price during the operational phase.

A previous study on the VCSBPPSI has shown that commercial solvers are not able to deal with real-scaled instances of the problem with more of the 200 items in a reasonable amount of time, thus justifying the implementation of heuristics or even the introduction of new specific problems [27, 11]. In particular, the progressive hedging (PH) heuristic has proven to be the most effective [26]. However, despite its good performance in its classical form, it cannot solve real-world instances when decisions must be made within a short period, as in e-commerce applications.

Such a drawback requires new methods, among which the most promising are the ones applying machine learning (ML) methods. In recent years, the so-called *learning to optimize* (L2O) have quickly surfaced as efficient and effective solutions to many optimization problems [17]. The key idea of this approach 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 have been proven to have good generalization capabilities with respect to realistic instances since they may use practitioners' insights without any architectural modification. Nevertheless, there are two major gaps in the literature concerned with the L2O concept. First, only a few papers consider ML heuristics tailored to solve two-stage stochastic optimization problems [82, 39]. Second, ML algorithms are widely used together with other techniques (e.g., with genetic algorithm [1] or with exact solver [5]) but only a few times are used alone. In fact, the major limitation to directly predict the solution is the inability to handle variable hard constraints [39]. The present thesis contributes to filling these gaps by introducing a specific ML-based heuristic for VCSBPP, which enables the solution of realistic instances containing up to 1000 orders.

This approach benefits from the generalization abilities of the ML but requires "on the fly" training and a forecasting scheme. Herefore, tuning the forecasters for each delivery scheme application becomes a non-trivial task due to the small data pull available. Moreover, the algorithm's complexity and computing time minimization should be leveraged with the provided accuracy of demand satisfaction. However, there are different methodological aspects of the application of this approach to last-mile delivery, which improve the aforementioned issues. Then, applying the so-called quantile forecast methodologies as Gaussian Process (GP) regression dramatically improves the ability of the whole algorithm to satisfy uncertain delivery demand [100, 99]. Lastly, it is possible to artificially increase the data samples to provide the pre-trained ML algorithms for further use in the "online" manner.

In Chapter 3, we present a detailed literature review of the fleet capacity management models on the tactical decision scheme. It includes the existing optimization models and heuristics to approximate the solution to the problem. In addition, we cover the existing literature on demand forecasts in the tactical range decisions in logistics and existing application scenarios of the outlined problem.

In Chapter 4, we introduce ML-based heuristics for VCSBPPSI. Firstly, we cover the problem formulation and methodological aspects of ML-based heuristics. Then we concentrate on the feature analysis for ML classifiers, which is the most critical issue, and outline stability tests. Then, we compare the introduced ML-based heuristics performance with the existing state-of-the-art methods such as Progressive Hedging heuristics and typical sampling approach for MILP solver. Next, we perform computational experiments on different problem samples with a wide range in sample size. Finally, we conclude this chapter with a real-case study of fleet size management in Turin, Italy, and the scientific and industrial impact of the proposed methodology.

Chapter 5 is entirely devoted to demand forecasting for further fleet capacity management on the tactical decision range. Firstly, we propose a zoning-based demand forecast with an approach to existing deterministic VCSBPP. Next, we cover VCSBPP formulation and methodology of the algorithm, coupling this problem with the city zoning and demand forecast. Then, we outline the application of the proposed method to the real scenario of tactical fleet allocation in Antwerp, Belgium. Having identified the performance of the VCSBPP-based approach, we highlight the strong and weak points of the proposed approach and possible industrial application. We continue the topic of the zone-based demand forecast with the introduction of ZCAP, which aims to assign vehicles to identified zones, minimizing the costs involved. After the mathematical formulation of ZCAP, we provide an updated methodology of the zone-based demand forecast suitable to be coupled with ZCAP. Finally, we outline the experimental results of the proposed approach for the real-case study of fleet size management scenario on the 60 working days of a last-mile delivery company in Antwerp, Belgium.

Chapter 6 outlines the conclusions of the proposed thesis with the future work. We highlight the main scientific and industrial impact of the proposed models and algorithms and discuss future work in the direction of models and methods for last-mile delivery under demand uncertainty.

Chapter 2

Problem description

This chapter recalls the crucial problem tackled in the presented thesis. Stating the framework and highlighting its industrial application emphasizes research and industrial impact and constrains the applied methodology and research field. As mentioned, this thesis deals with the tactical-operational range decision on capacity management under uncertain demand. There are multiple scenarios of such context, which we describe in the following, along with the details and issues facing different stakeholders in this process. Finally, we highlight the primary purpose of the provided research work.

In general logistics applications, having enough capacity to adequately perform crucial activities such as supplying, storing, and distributing goods is paramount for any successful and competitive company. Thus, companies have to plan for sufficient capacity to be available at the desired time and appropriate locations to satisfy the customers' demand [27]. The capacity types could be transportation modes (e.g., space in ship or train, cargo bikes or vans) or storage space within given city distribution center facilities.

Since logistics activities are often subcontracted, capacity planning assumes the contracting 3-4PL firms book the required capacity beforehand. The results of these negotiations often take the form of medium-term contracts, specifying both the capacity to be used and the approximate supply location required. Consequently, changing the planned logistics scheme quickly increases costs for both the shipping company and 3-4PL. Nevertheless, capacity changes can be mitigated with the modern tendency of fleet consolidation, although it should be planned beforehand, as discussed further. The outlined problem setting is relevant in different contexts, where the main ones are last-mile delivery and long-haul transportation.

Capacity planning on a tactical scale in long-haul transportation plays a central role in securing a successful supply. Typically, the shipper negotiates contracts with carriers on multi-type capacities, such as trucks, ships, trains, etc., which are used to perform shipping activities every day, week, or month. Storage capacity is also a subject for planning, enabling the consolidation of goods, and requires specific contracting. Both capacity types are booked for a specific planning horizon, e.g., one semester, season, or year. The result of such an agreement is the contract estimating the quantity, type, and capacity and the expected costs of the booked capacity and possible capacity variations occurring during a typical execution of the operational stage [26, 27]. Indeed, given the time lag 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 affect the contract negotiation. Therefore, uncertainty in demand is commonly presented in a modeling approach for the underlined application.

In the realm of last-mile delivery, tactical capacity planning aims to ensure the efficient execution of consolidation. Private and public stakeholders, such as transit authorities, collaborate to form partnerships for capacity sharing and integrated decision-making. This collaboration aims to consolidate freight and mitigate the impact of urban freight transportation and logistics on the city. In industry, multi-tier city logistics systems are proven to be most effective in regulating the highlighted collaborations [23]. In the capacity planning part of such systems, managers must secure the required numbers of vehicles of various types, which will be available to perform the transportation operations correctly. This decision enables the sharing of the vehicles on the route and city distribution centers' storage space to reduce the economic and environmental operational costs. Nevertheless, the uncertainty in delivery demand impacts greatly such decisions since the shipping of goods in the city area depends on a tremendous amount of parameters and constraints compared to long-haul transportation. Examples of demand-changing factors include the inclusion of e-commerce in a shared logistic scheme, various client-dependent parameters such as discounts, holidays, etc., and goods-related constraints in the food or fragile items supply. Therefore, supporting the last-mile delivery is strictly related to the main goal of the proposed work, capacity planning under uncertainty.

Multi-tier smart urban transportation systems are implementing the provided approaches to the decision-making process and collaborative environment [88, 32]. 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, using multiple transportation tiers enables the system to utilize specifically adapted infrastructure and specialized fleets at each tier to attain the overall goal better.

Consequently, such city logistics systems evolved to provide more integrity and robustness with decision support platforms, namely Intelligent Decision Support Platform (IDSP) [107]. Using IDSP to automate planning and optimization operations results in profits for stakeholders involved in collaborative city logistics systems. The goals of IDSP may slightly differ depending on the exact industrial application, but the core principle is to apply the shared knowledge to balance supply (capacity) and demand. In city logistics, shippers or clients represent the demand side, and carriers of the logistics 3-4PL companies supply the required capacity to move the actual goods. In such a setting, there are different critical requirements for IDSP to meet the expectations of both types of stakeholders.

The timing of the IDSP response plays a central role in actualizing the logistics scheme. This is due to the frequent demand volume changes by shippers since once the new order appears in the system, the whole logistic scheme may be subject to change to reduce operational costs. The system response time is defined by optimization cycles, depending on the underlying problem complexity. Therefore, reducing complexity or providing suitable heuristics to approximate the near-optimal solution to the capacity problem for the supply side is of direct use for every stakeholder in the system.

Another cornerstone of IDSP for planning is accounting for uncertainty on the demand side. In the case of last-mile logistics, it is hardly predictable due to many variations and trends. The uncertainty treatment in such a case requires complex computing solutions, which reciprocally influence the response time of the whole system. Therefore, a certain level of hedging against uncertainty and demand aggregation is required for the forecasting methodology to be accurate and computationally reasonable.

In industrial applications of last-mile logistics, demand is commonly related to the geographical location of the target customers or business. This is due to a strict correlation between the city district and its infrastructure. Therefore, data aggregation into certain city zones is commonly adopted by logistic managers in the demand uncertainty treatment. Moreover, frequently on the tactical decision scale, each city zone is associated with a certain driver and vehicle type provided by the city routes' infrastructure and legal policies. Therefore, delivery data aggregation into city zones is a logical step toward a robust forecast methodology for further capacity management.

In the present research thesis, we approach the mentioned issues enabling the successful applications of IDSP in last-mile logistics. The core mathematical problem is therefore considered VCSBPPSI. Firstly, we propose a new heuristic for the VCSBPPSI suited to tactical capacity management to achieve a reasonable time solution for large-scale problem instances. This achievement applies to intelligent systems in long-haul transportation as well. Secondly, we provide the zone-based forecasting methodology coupled with a deterministic capacity management problem (VCSBPP). Lastly, we provide a new optimization problem coupled with a zone-based demand forecast, enabling fleet capacity optimization approaching city infrastructure but not each order treatment. These findings aim to speed up the re-optimization cycle of IDSP, applied to the last-mile logistics. Consequently, it reduces the operational costs for each stakeholder in the collaborative multi-tier logistic system.

Chapter 3

Literature review

3.1 Optimization problems for logistics planning

Multiple optimization problems concern the tactical decision level last-mile delivery applications. Many of them are based on the concept of packing a set of given items into a set of various containers (bins) [74, 109]. The crucial characteristics enabling the thesis-related classification of the provided problems are the structure of the objective function and the elaboration of the real-world related uncertainty.

One of the fundamental problems in the fleet management application field is the Variable Cost and Size Bin Packing Problem (VCSBPP) [30]. This problem aims to minimize the cost of used containers (bins), which depends on the characteristics of each vehicle in the last-mile application. Moreover, it accounts for the vehicles' different sizes (capacities) and costs, constraining the volume of items to handle. Although this problem received a scientific interest in providing efficient lower bounds and heuristics [30, 52, 58], its applications in the last-mile delivery field and for the tactical range decisions are quite limited [40]. This is because VCSBPP formulation is deterministic in its basic form and, therefore, does not allow the introduction of uncertain variables.

This extension is crucial to represent the uncertain demand related to the tactical decision scheme, where the customer's orders are partially or completely unknown. VCSBPPSI is on a particular scope of this thesis since it was successfully applied in the last-mile tactical decision scheme regarding the usage of 3PL under orders uncertainty in [26]. Furthermore, the limitations related to the involvement of stochastic items in VCSBPP can be solved by applying Machine Learning-based heuristics [5, 11]. Nevertheless, the demand uncertainty in this framework has to be carefully defined through the suitable probability distribution derivation. Therefore, future research is needed for a self-inclusive framework for demand uncertainty elaboration.

Another example of approaching the tactical decision level is presented in [24], Mentioned problems consider the item and bin-specific item-to-bin assignment costs depending on many physical, economic, and temporal parameters. The experimental results of introduced heuristics to this problem in a one-period and multi-period setting showed the high-performance and robustness regarding solution quality and computational efficiency. Nevertheless, this work covers the case of perfect information about the delivery orders (demand), which makes it hardly applicable in the last-mile delivery field due to the aforementioned problems of last-minute orders in e-commerce.

It is worth mentioning that BPP evolved to account for different aspects of the last-mile delivery application, not only uncertainty on item and fleet characteristics. In general, the time dimension recently started to be treated with BPP. The authors in [36] introduced the temporal BPP (TBPP) as a generalization of the standard BPP where each item has to be packed within a given time window. Another problem involving time delivery window elaboration is introduced as VCSBPP-TD in [87]. The authors provided a model formulation and real-case study of the VCSBPP-TD application to manage the satellite hub in the last-mile logistics, reducing the gap between the tactical and operational phases. Finally, in [2], authors account for additional (optional) items to be loaded together with the required items, providing a profit management side to the problem, which brings us to the other optimization strategy for packing problems described below.

In contrast to the cost minimization, in the Multi-Handler Knapsack Problem under Uncertainty (MHKPu), given a set of items characterized by volume and random profit and a set of potential handlers, the authors try to maximize the expected random profit provided by each chosen item [97]. Negligible computing time provided with the deterministic approximation of the problem makes it an excellent predictive tool for considering stochastic handling costs in supply chain problems and tactical last-mile decisions. Finally, the Generalized Bin Packing Problem (GBPP) unifies the objective function and characteristics of the aforementioned problems by selecting the subset of profitable non-compulsory items to be loaded together with the compulsory ones into the appropriate bins [3]. The GBPP provides a generalization approach to the existing state-of-the-art models and can be reduced to various problems, such as VCSBPP or Knapsack Problem. Furthermore, it was extended to account for the random oscillations in the item profit under the name SGBPP, improving the model's applicability for the logistics managers in the last-mile delivery field on the tactical optimization range [96]. Finally, it was successfully applied to a set of realistic problem instances with up to 1000 items in the delivery applications with the suitable heuristics, showing acceptable optimality gaps [4]. Summarising the characteristics of the aforementioned problems that have found application in the last-mile delivery field on the tactical decision range, most of them have been proven to provide robust modeling support for the managerial decisions with the available information but limited uncertainty treatment capabilities due to computing complexity issues [40, 26, 90].

3.2 Demand uncertainty treatment

This approach allows us to hedge against real-world uncertainty and recover a feasible solution in the second stage when orders and profits are known [39]. Therefore, the multi-stage approach is frequently considered redundant in the tactical decision scheme since the whole process can be approximated with the two-stage cycles. Despite this widely adopted assumption, multiple models involve the multi-periodicity and so-called non-myopic approaches in the tactical decision scheme [24].

Nevertheless, approximating uncertainty is still a non-trivial task in the typical scenario decomposition approach [11, 28, 26, 96]. It is worth mentioning that even the BPP (the basic component of almost every outlined problem) is known to be NP-hard in the strong sense [60, 47]. In turn, the complexity of the stochastic problem rises exponentially with each scenario involved. For instance, it has been proven that for VCSBPPSI realistic instances with a considerable amount of vehicles and items to deliver (more than 200 items and 50 vans in the big city) become computationally intractable, as in the time spent and the memory consumed [40, 11]. Thus, heuristics are required to solve the two-stage stochastic programs in tactical last-mile applications.

To the author's knowledge, the heuristics that shows the best performance on two-stage stochastic programs so far is the PH [26]. Initially proposed in [101], 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 convergence guarantee if the problem has integer variables (as for the VCSBPPSI). Nevertheless, it has shown good performance as a heuristic for several discrete problems [89, 28]. 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) [42, 66]. 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 or to solve huge realistic instances of big city cases with more than 1000 parcels [61, 11]. These findings the way to the ML-based heuristics and L2O approach advent, to which the current thesis contributes in the further chapters.

3.3 Machine Learning applications in optimization for logistics

The application of ML methods for optimization is a recent and growing topic in the literature. The most recent surveys on the topics include [75], which considers the application of ML to the traveling salesman problem, [48], which includes the ML and optimization applications to the last-mile delivery, [108], which resumes data-driven ML meta-heuristics, and [83], which analyzes the applications of deep learning to mathematical programming under uncertainty. Based on our literature analysis, most applications use ML to decide the optimal values of parameters/hyperparameters of some solution techniques [68, 78, 22] or to approximate complex terms of the objective function [7]. Nevertheless, to our knowledge, few researchers have directly tried to find the optimal solution to an optimization problem by directly using ML techniques [5, 76]. 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 use traditional heuristic approaches due to the consequent decrease in the overall system performance [14, 91].

Concerning the application of ML to stochastic combinatorial optimization problems, the issue is that most works are devoted to being applied to VRP, which models the operational decision level [48]. Nevertheless, to the best of the authors' knowledge, it is possible to identify three representative papers in the thesis main question setting: [77], [63], and [5]. The first presents a general agent-based ML adaptation for combinatorial optimization problems. It requires considerable tuning and does not account for uncertainty. This fact prevents its application to the VCSBPPSI, for instance. 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 leads to a non-linear optimization problem. It is worth mentioning that second-stage approximation with ML was further extended to a richer class of two-stage stochastic problems in [62]. Nevertheless, in the present thesis, we elaborate on the first-stage decision variables approximation, providing a robust solution with reduced computational effort since the second stage remains intact. Therefore, the combination of ML approximation of the first and second stages is a subject for further research. The third paper is an example of the 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. One more similar approach dealing with constraints and not with variables has been proposed in [56]. In particular, the authors first identify the invariant constraint set of MILP instances offline. Then, they train a machine learning method for detecting an invariant constraint set as a function of the problem parameters of each instance. Finally, they predict the invariant constraint 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 [5].

Another approach to account for the demand uncertainty is the straightforward forecast for the stochastic parameters followed by a problem solution with obtained deterministic values. In general, demand prediction is vital for many fields of application, such as the food industry, perishable goods supply, or crowd shipping [41]. Nowadays, e-commerce presents a new challenge for demand forecasting since highly fluctuating data characterize it. Unfortunately, few methods can capture such a complex data structure due to high uncertainty and no evident trend. Thus, more sophisticated and robust approaches have to be applied. For example, in [67], authors successfully used the Long Short Term Memory (LSTM) network to predict the logistics delivery demand in the manually defined sub-region of the city. The result of this work is a two-dimensional LSTM, which enables the company to have a reliable support decision system for future decisions. However, this approach is characterized by many empirical parameters and requires manual city division. Hence, further work is required in this direction.

3.4 Clustering applications in last-mile logistics

In our search for the attempts to improve In other words, it is crucial to account for the variables connected to the spatial allocation of the order to improve the performance of the time series forecast. However, a straightforward introduction of the multivariate regression complicates the problem multiple times [64]. Thereby, the existing approaches adopt clustering to access the spatial information of the demand, as hierarchical clustering approach in [53]. One more example of clustering is introduced in [81], where has been applied the division of the customers into different logical segments following the monthly volume of products delivered for the further applications of the demand forecasting methods to each of them.

In general, many managerial approaches benefit from clustering methods [59]. A large body of research exists on consumer behavior segmentation in an e-commerce context within the marketing field. Part of that research strand focuses on identifying shopping value, or user segmentation [33, 54]. Unsurprisingly, this field of literature received a boost during the pandemic [57, 50]. For instance, authors in [105] propose an approach to zoning the urban area for an effective delivery system.

The finding of this work is based on the commercial software modeling the traffic flow, which confirms the idea of implicit zoning for managerial insides.

Moreover, authors in [38] conduct research on city zoning with managerial insides, proposing a decision-making tool based on clustering methods to identify the best zoning approach for logistic purposes. In that study, the authors mostly involved different spatial and socioeconomic data, proving the importance of infrastructure in urban freight-related decisions. Finally, starting from the socioeconomic characteristics of individuals, in [18], the authors map the overall demand for e-commerce groceries. Unsurprisingly, the strict correlations between the city infrastructure (physical shop locations, etc.) and demand were identified. The remaining gap in converting these online shopping probabilities into an effective value for freight trips is covered in [6], where authors identify the main drivers for the number of household freight trips.

Similarly, in [84], authors analyzed the city division from the placement of collection and delivery points and their accessibility to reduce failed home deliveries for logistics companies. They used a clustering approach to identify the spatial structure of existing stores and evaluate their accessibility. The above efforts demonstrate a shift toward the usability of the mentioned models of city zoning for logistics operators and companies. For years, the logistic managers were separating the urban area and assigning a driver for each of the obtained regions. This city zoning provides time-space correlations but typically approaches traditional information, such as the ZIP codes or historical regions[38]. Moreover, the authors in [39] have proven that the application of unsupervised clustering methods to obtain the city zoning based on the data of the underlying city structure, not necessarily orders history, dramatically improves the effectiveness of the city zoning from the last-mile delivery perspective.

The intuition of city separation is widely adopted in logistics managerial literature concerned with strategic decisions. For instance, the two-tier city logistics model divides the urban area into two regions: external zones with hubs and internal with so-called satellites [31, 25, 103]. The principal component of such a division is the fleet consolidation points, where the goods are moving from one vehicle type to another, e.g., the tram stops in Amsterdam city to transfer the parcels to cargo bikes. Nevertheless, such city separation reflects city infrastructure implicitly due to the rules of satellite placement, as the tram would stop only in locations that are not harmful to the city road and public transport traffic. Almost every two-tier logistic model includes a concept of micro consolidation, representing a fleet consolidation between two tiers (types) of vehicles, which appears to be a driver for the city zoning, as shown in [104]. In this manuscript, authors proved empirically that the clustering (or zoning) of the urban territory basis helps in the implementation of rational logistics solutions in last-mile applications. Notice that clustering the city in the geographical area implies the introduction of the microzones, which has been proven to directly impact the operational level [55], due to the straightforward assignment of each of the vehicles to serve a predefined area. Moreover, it is employed to tweak the Vehicle Routing Problem, which is used on the operational level, with the cluster of the clients, taking which the driver can not leave the area until serving them [110, 45].

Finally, an excellent example of a provided problem combination with city zoning is introduced as Two-Echelon Electric Vehicle Routing Problem with Time Windows and Partial Recharging [12]. The authors used a two-echelon formulation to distinguish the urban zone and restricted traffic zones (low emission zones or LEZ), in which classical transport is not allowed to enter. In fact, with the government restrictions concerning city centers and LEZ, it becomes paramount to combine traditional vehicles with electric cargo bikes and vans [46, 10]. Therefore, zoning or clustering approaches become an essential tool in the logistics models, enabling the inclusion of city structure information.

Recently, a districting problem was introduced and studied, which aims at optimal city separation for multiple tasks, such as the design of commercial sales areas, political districting, waste collection, etc. [37].

Chapter 4

Machine Learning Heuristics for Variable Cost and Size Bin Packing Problem with Stochastic Items

The chapter is organized as follows. Section 4.1 presents the mathematical description of the VCSBPPSI. introduced ML heuristic and the PH algorithm are described in Section 4.2 and Section 4.3 correspondingly. The experimental setting and numerical results are presented in Section 4.4. In addition, interesting policy-making and managerial insights are derived from the real case study on parcel delivery in the metropolitan area of Turin, Italy, in Section 4.5. Finally, the scientific and industrial impact of the introduced heuristics is presented in Section 4.6. The content of this chapter has been published in [11].

4.1 Variable Cost and Size Bin Packing Problem with Stochastic Items

This section recalls the VCSBPPSI problem, possible applications, and the mathematical formulation [27]. The VCSBPPSI concerns the decision problem of a shipper, which needs to secure the capacity of different types from a carrier to meet its uncertain demand.

Let us consider a set of scenarios $\mathcal{S}$ 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

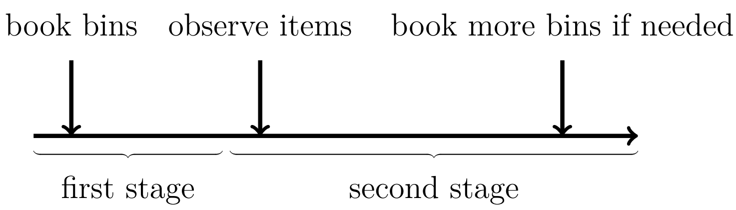


Figure 4.1: Graphical representation of the sequence of decisions and stages.

and volumes are deterministic because they are known in advance and represent a shared knowledge between the decision-maker and the 3-4PL provider. Since the second-stage bins are booked on the spot market and not ahead, we consider that $c_k/V_k > c_j/V_j \; \forall \; j \in \mathcal{J} \; k \in \mathcal{K}^s$. Moreover, the model considers the following binary decision variables:

- $x^s_{ij}, i \in \mathcal{I}, \; j \in \mathcal{J} \cup \mathcal{K}^s, s \in \mathcal{S}$ 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;
- $z^s_k, k \in \mathcal{K}^s$ equal to 1 if bin k is booked in scenario s.

The VCSBPPSI formulation is

$$\min \sum_{j \in \mathcal{J}} c_j y_j + \sum_{s \in \mathcal{S}} p_s \Big(\sum_{k \in \mathcal{K}^s} c_k z^s_k \Big) \tag{4.1}$$

$$\text{s.t.} \sum_{j \in \mathcal{J}} x^s_{ij} + \sum_{k \in \mathcal{K}^s} x^s_{ik} = 1, \qquad \forall i \in \mathcal{I}^s, \; \forall s \in \mathcal{S} \tag{4.2}$$

$$\sum_{i \in \mathcal{I}^s} v^s_i x^s_{ij} \leq V_j y_j, \qquad \forall j \in \mathcal{J}, \; \forall s \in \mathcal{S} \tag{4.3}$$

$$\sum_{i \in \mathcal{I}^s} v^s_i x^s_{ik} \leq V_k z^s_k, \qquad \forall k \in \mathcal{K}^s, \; \forall s \in \mathcal{S} \tag{4.4}$$

$$x^s_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{I}^s, \; \forall j \in \mathcal{J} \cup \mathcal{K}^s \tag{4.5}$$

$$y_j \in \{0,1\}, \qquad \forall j \in \mathcal{J} \tag{4.6}$$

$$z^s_k \in \{0,1\}, \qquad \forall k \in \mathcal{K}^s, \tag{4.7}$$

where p_s is the probability of realization of the scenario $s \in \mathcal{S}$.

The objective function (4.1) 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 (4.2) ensure that each item is packed in a single bin. Constraints (4.3), and (4.4) ensure that each bin's capacity is not exceeded. Finally, Constraints (4.5)–(4.7) specify the domain of the decision variables.

Model (4.1)-(4.7) 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. 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 in a strong sense [47] thus, also the VSBPP is NP-hard [21] as well as the aforementioned stochastic version of it, VCSBPPSI. It is worth mentioning that if we reverse the objective function to account for the profit of each item, VCSBPP boils down to *Multiple Knapsack Problem*, which is known to be an NP-hard in the strong sense [112, 98]. Therefore, we found VCSBPPSI the most useful formulation for 3-4PL usage, but this problem will be discussed in the following thesis chapters.

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 (4.4) 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 (4.1)-(4.7) can be infeasible if the bins' volume is insufficient to contain all the items. Nevertheless, in the following, we do not consider this issue since the number of bins available through 3-4PL providers is usually much greater than the volume of the bins to allocate.

4.2 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 [8]. 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 that 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). We present a pseudo-code in Algorithm 1 to further clarify this procedure.

Algorithm 1 Training set building.

```
D = ∅
for k = 1, 2, ...K do
    Generate a problem instance
    y ← solution of the problem (4.1)-(4.7).
    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 4.1, 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 as a priory, 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 of having 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 fast with the polynomial complexity algorithms. 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, we propose a different way to measure the same quantity, and we expect to select just a few of them. 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.

Table 4.1: 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 (4.1)–(4.7)	-
`Reduced cost`	The normalized value of the reduced cost r_j of the bin in the continuous relaxation of Model (4.1)–(4.7)	$\frac{r_j}{\max_{j \in \mathcal{J}} r_j}$
`Items placed avg`	The average quantity of the items placed into the bin j in the continuous relaxation of Model (4.1)–(4.7)	$\frac{1}{S} \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}} x^s_{ij}$
`Items placed max`	The maximum quantity of the items placed into the bin j in the continuous relaxation of Model (4.1)–(4.7)	$\max_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}} x^s_{ij}$
`Items placed min`	The minimum quantity of the items placed into the bin j in the continuous relaxation of Model (4.1)–(4.7)	$\min_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}} x^s_{ij}$
`Items capacity`	The theoretical number of item that the bin may contain	$\frac{V_j}{\frac{1}{S} \sum_{s \in \mathcal{S}} \frac{1}{I^s} \sum_{i \in \mathcal{I}^s} v^s_i}$
`Items capacity quant`	The theoretical item capacity, where $q(v^s_i, 0.8)$ is the 0.8 quantile of the v^s_i	$\frac{V_j}{\frac{1}{I} \sum_{i \in \mathcal{I}} q(v^s_i, 0.8)}$
`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 ($g_j = c_j/V_j$)	$\frac{g_j}{\frac{1}{K} \sum_k \frac{1}{S} \sum_{s \in S} g^s_k}$
`Unitary cost wrt ss max`		$\frac{g_j}{\max_k \frac{1}{S} \sum_{s \in S} g_k}$
`Unitary cost wrt ss min`		$\frac{g_j}{\min_k \frac{1}{S} \sum_{s \in S} g^s_k}$

The computation of the proposed features can be done in polynomial time since the most expensive operation is to solve the continuous of Model (4.1)–(4.7), which requires negligible computational effort compared to the solution of the exact model. 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 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 [73, 13]). 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. However, the feature selection process is the cornerstone of this approach since it influences the bias/overfitting of the classifier regardless of its own structure. This problem is discussed in detail in Subsection 4.4.4.

Algorithm 2 ML heuristic.

```
Compute continuous relaxation of the problem (4.1)-(4.7)
for j = 1, 2, ..., |𝒥| do
    f^j ← features for bin j (Table 4.1)
    ŷ_j ← classify(f^j)
end for
Solve problem (4.1)-(4.7) with y_j = ŷ_j for each scenario s
```

4.3 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 [42].

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

which can be rewritten as

$$y_j^s = \bar{y}_j \quad \forall s \in \mathcal{S}, \; j \in \mathcal{J}, \tag{4.8}$$

where $\bar{y}_j = \sum_{s=1}^{|S|} p_s y_j^s$. 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 [42] and successfully applied by [66]. 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:

$$\begin{aligned} \min & \sum_{j \in \mathcal{J}} c_j y_j^s + \sum_{k \in \mathcal{K}^s} c_k z_k + \sum_{j \in \mathcal{J}} \theta_s |\bar{y}_j - y_j^s| \\ \text{s.t.} \quad & (4.2), (4.3), (4.4), (4.5), (4.6) \text{ and } (4.7), \end{aligned} \tag{4.9}$$

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 3, where ρ is the Lagrangian multiplier associated with the non-anticipative constraints, and ε is a stopping parameter for ending the algorithm if $g^{(k)} = p_s \sum_{s=1}^{|S|} ||y^{(k)} - \bar{y_s}^{(k)}||_1$. If the condition (4.8) 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.

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

4.4 Numerical Results

This section reports the numerical experiments for assessing the performance of the proposed heuristics. generation procedure in Subsection 4.4.1. Then, in Subsection 4.4.2, we study the out-of-sample stability of the approach. Moreover,

Algorithm 3 Customized progressive hedging.

$k := 0$
$g^{(k)} := +\infty$
$\bar{y}^{(k)} := 0$
$\theta_s^{(k)} := 0 \quad \forall\ s \in S$
while $g^{(k)} \geq \varepsilon$ **do**
  for s in S **do**
    $y_s^{(k)} \leftarrow$ Solution of the problem (4.9).
    $\theta_s^{(k+1)} := \rho\ |\bar{y}^{(k)} - y_s^{(k)}|$
  end for
  $\bar{y}^{(k+1)} := p_s \sum_{s=1}^{|S|} y_s^{(k)}$
  $g^{(k)} := p_s \sum_{s=1}^{|S|} ||y^{(k)} - \bar{y_s}^{(k)}||_1$
  $k := k + 1$
end while

Subsection 4.4.3 and 4.4.4 present the implementation of the ML heuristic and the feature selection process, respectively. Finally, in Subsection 4.4.5, the optimality gaps of the two heuristics with respect to the exact solver are quantified, and in Subsection 4.4.6, 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 [51].

4.4.1 Instance Generation

The instance generator procedure generalizes the one in [27]. 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 are meant to 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 instance type are shown in Table 4.2.

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`].

Table 4.2: 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

- 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 $[0.7, 1.3]$ [27].
- 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 $[1.4, 1.8]$ [27].

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 3-4PL providers is enough to contain all the items, and the volume of the parcels is smaller than the one of the bins.

4.4.2 Stability Test

Before starting with the experiments, it is necessary to compute the out-of-sample stability of the proposed model [8]. The procedure used is standard: We generate S scenarios and use them to solve Model (4.1)–(4.7). We then generate $S' >> S$ 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 (4.1)–(4.7) and the obtained average for different values of $|S|$ is shown in Figure 4.2. On the x-axis, we outline the number of scenarios S in the problem. On the y-axis, we plot the percentage difference between the objective function of out-of-sample and actual solutions. 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.

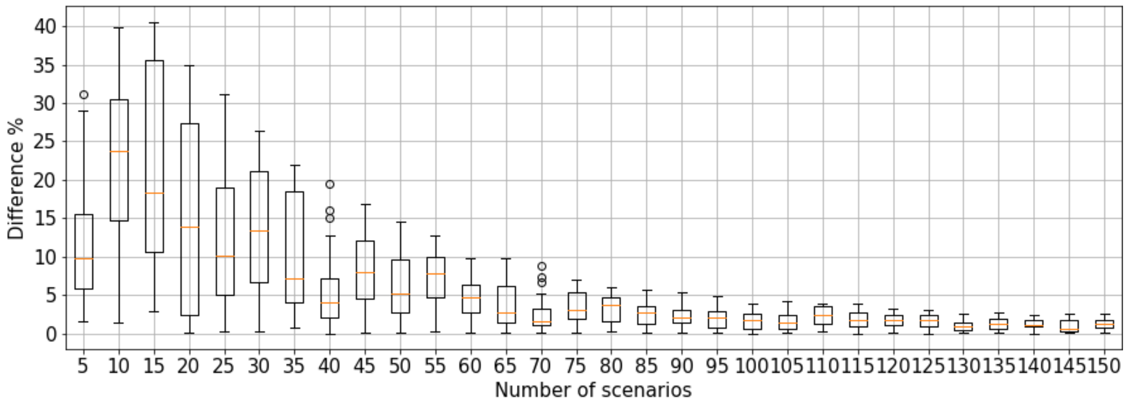


Figure 4.2: Exact approach stability test results.

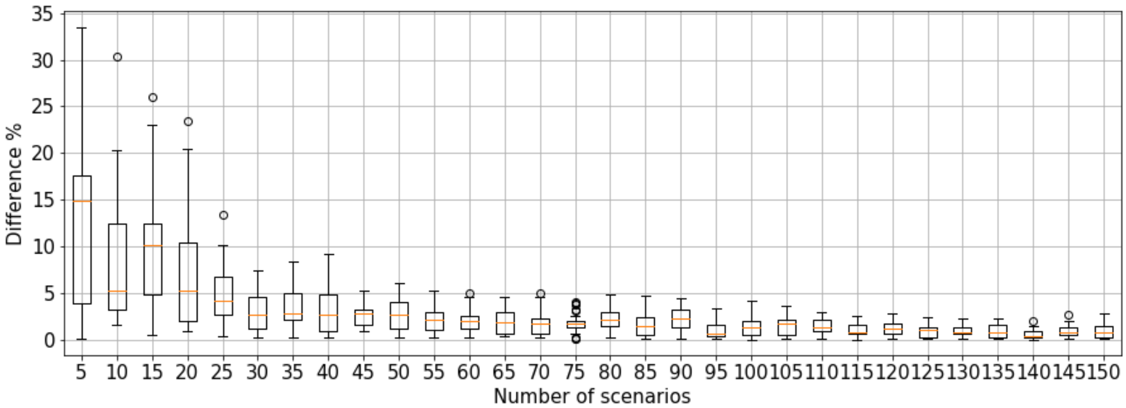


Figure 4.3: PH heuristics stability test results.

The effect of the number of scenarios also has to be quantified in the proposed heuristics. 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 the S scenario 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. Figure 4.3 and 4.4

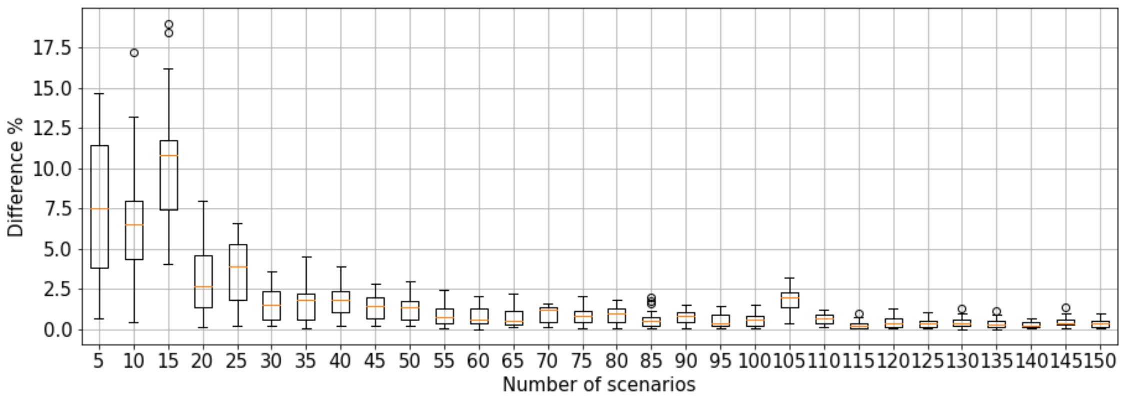


Figure 4.4: ML heuristics stability test results.

show the results, which were averaged over 300 instances (100 `small`, 100 `medium` and 100 `large`).On the x-axis, we outline the number of scenarios S in the main problem (used with PH or ML heuristics). On the y-axis, we plot the percentage difference between the objective function of out-of-sample and actual solutions.

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.

In this section, we analyze the solutions' characteristics and the properties of the mathematical model. In particular, we consider `benchmark` instance and 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.

The obtained results are shown in Table 4.3. All the values are averaged on 10 instances. As expected, the computational time increases as the number of items and bins increases. Since the exact solver generates an out-of-memory exception, we do not solve instances with more than 150 items, 10 first-stage bins, and 10 second-stage bins.

Table 4.3: 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.

	Bins					
Items	**f.s.**	**s.s.**	**Comp. Time [s]**	**First-Stage Cost [%]**	**EVPI [%]**	**VSS [%]**
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)

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 4.2). Despite this, the first-stage cost is never 100%, nor for the instances with 50 items, since, in 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, while $EVPI$ is always lower than 30%, the VSS lowest value is 48.5%, which could generate poor performance in the application. This is also due to the definition of the EVP for a problem in which uncertainty affects the second-stage sets ($\mathcal{I}^s$ and $\mathcal{K}^s$). However, this discussion on the problem structure itself lies beyond the thesis scope.

4.4.3 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 outlined below:

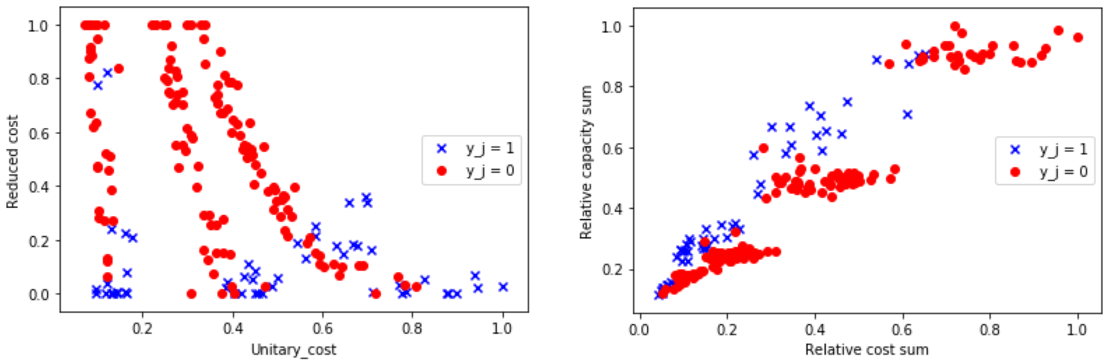
- **KNN**: The K neighbors classifier checks the distance from the test examples to the training sample, also called the centroid. The group of samples that gives the smallest distance is identified as belonging to the same class.
- **L_SVM**: The support vector machine with a linear kernel is a well-known robust classification method, the intuition behind which 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, a more sophisticated version of the SVM, uses the kernel trick to identify the distance through the RBF function, which checks for non-linear correlations in the data.
- **GP**: The Gaussian process classifier with an RBF kernel offers the great potential of GP regression adaptability and a closed-form solution with parameter optimization inside the loop.
- **DT**: The decision tree classifier divides a dataset into smaller subsets based on the chosen criteria. Once the data is divided into one sample, it assigns the class. The current approach identifies the entropy criteria for randomly splitting the data while simultaneously constraining the structure of the tree by a minimal number of samples in the leaf by 30 and 10 samples to split.
- **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 most of the trees, which preserves the method from overfitting.

- **NN_1l**: The multi-layer perceptron (MLP) classifier 1 hidden layer of 100 neurons. The number of features defines the input layer size, and the output neuron provides the class identification with the sigmoid function.
- **NN_ml**: A convolutional neural network (CNN) consisting of 3 hidden layers with a size of $25, 50, 15$ neurons, respectively. The number of features defines the input layer 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, further combining 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. It has performed admirably on highly fluctuating data [113] and demand prediction [67, 40]. A very basic version of LSTM with a 1 hidden layer of 50 neurons is applied.

All the outlined approaches were implemented with the help of the Scikit-learn library [86], 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 labels. 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 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 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 Figure 4.5. 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 $y_j = 0$.



(a) (b)

Figure 4.5: Feature graphs. `Reduced costs` vs. `unitary cost` on the left; `relative capacity sum` vs. `relative cost sum` on the right.

In Figure 4.5a, the considered features are `reduced costs` and `unitary cost`. If the reduced costs are zero or close to zero (the 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 Figure 4.5a). 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 could be due to the fact that we are not considering several other features that can better characterize the bins on the two-dimensional graphs. On the other hand, it is confirmed with the feature analysis in Subsection 4.4.4 that `unitary cost` does not provide the most relevant information concerning the optimal solution.

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 goals.

4.4.4 Features Analysis with SHAP

In this section, we analyze the set of features defined in Section 4.4.1 to select the most significant ones. To do so, we use the SHapley Additive exPlanations (SHAP) values proposed by [69]. 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. More specifically, we adopted the SHAP tree explainer [70], a version of SHAP for tree-based machine learning models. As for the testing model, we chose the random forest classifier introduced above. Figure 4.6 shows each introduced feature's mean SHAP values.

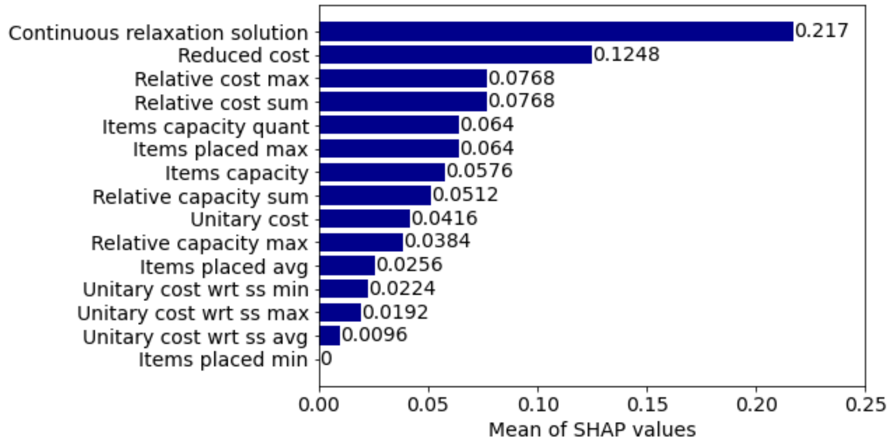


Figure 4.6: Mean of the SHAP values for each of the features.

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 [72]. Nevertheless, the proposed methodology can produce the same results without prior experience, looking only at the data. Thus, it can be extended to other optimization problems characterized by binary variables.

4.4.5 Heuristics Comparison

In this section, we test the performance of the proposed heuristic and the one of the PH against the commercial solver Gurobi [51]. The results are described in Table 4.4. All the values are averaged over 500 `benchmark` instances. The first two columns represent the average computational time and its standard deviation.

Table 4.4: 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

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 4.5.

Table 4.5: 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

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

4.4.6 ML and PH Heuristic Comparison

Since the PH was shown to achieve lower OF gaps in the previous section, we compute the gaps with respect to its performance. Moreover, in order 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 4.6, 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 4.2). 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, making ML-based heuristics 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.

The l_1 distance between the PH and the ML first-stage solutions does not differ by more than 10%. 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.

Table 4.6: Resulting computational time, percentage gap, and the l_1 distance to the PH solution of the real scale problem test.

	Computational time [s]			Gap to PH solution			l_1 distance to PH solution		
Instance	`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

4.5 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 the 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 [87, 95]. Sensitive data for the involved e-commerce and parcel delivery companies have been anonymized and normalized by means of the data-fusion tool provided in [94].

The case study tackles a set of deliveries (characterized by delivery locations and parcel volumes) within the Turin city center area ($2.805 \times 2.447 \text{km}^2$), using a heterogeneous fleet of three types of vehicles: Cargo-Bikes (CBs), Electric Vehicles (EVs), and Light Duty vehicles (LDs). Figure 4.7 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 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.

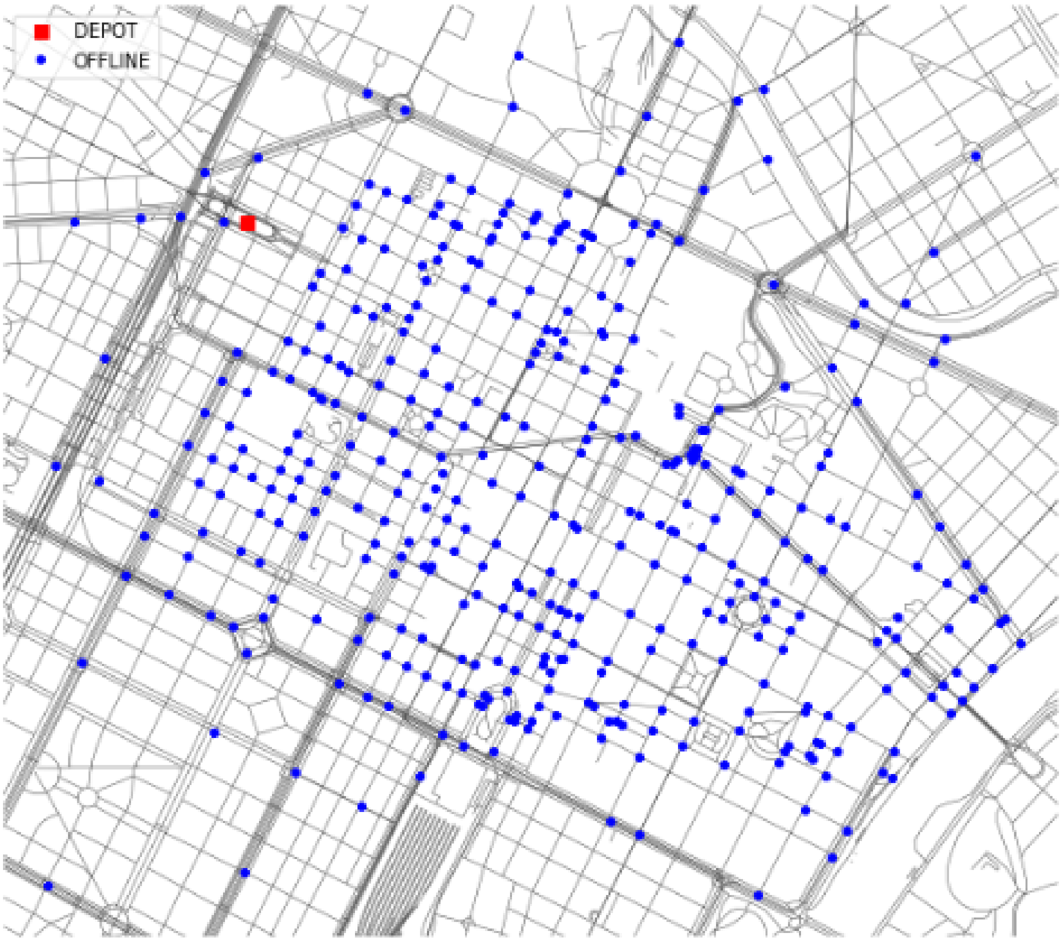


Figure 4.7: Service area in the case study

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 the GP classifier, against a single-echelon approach, which adopts the state-of-the-art stochastic vehicle routing problem by [102], where a fleet of LD vehicles is used. The parameters for the two models are taken from previous works and are summarized in Table 4.7. 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 routing has a limited effect on the final solution due to the small distances [87]. 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 regarding the standard routing strategy often used in practice. Moreover, the results 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.

Table 4.7: Sources of data for the Data-Fusion phase

Data type	**Source**
Satellite localization	[80, 94]
Orders data	[9, 30]
Spatial data of demand	[35]
Time distribution of the orders	[35]
Vehicles characteristics and vehicle booking costs	[94, 93]
Road network	OpenStreet, 5T Road Sensors' data [111]
Time-dependent travel times	[71]
Environmental costs	[9, 49]

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 [111]. The proposed Monte Carlo-based simulation algorithm runs the following steps 10 times (see Figure 4.8 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 one by adjusting the fleet according to the new deliveries and compute the route running:
   - the heuristic in [102], if the VCSBPPSI is considered
   - the stochastic VRP heuristic in [102], 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, given by the fuel production process and the long-haul shipment of the fuel, CO_2 equivalent to including other pollutants (e.g., NOx).

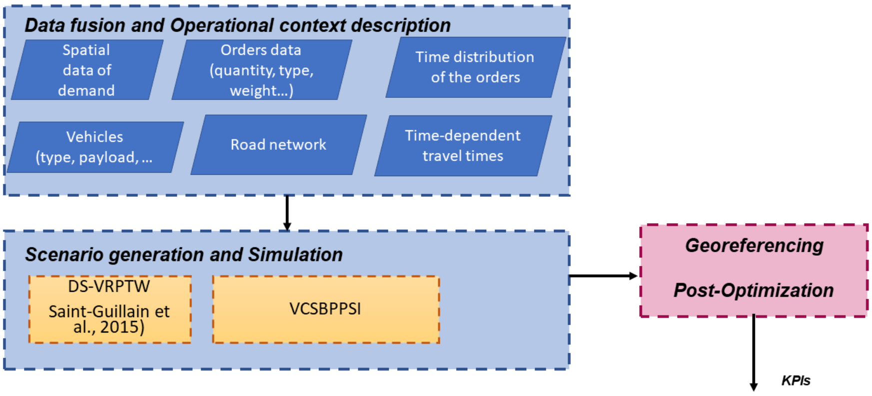


Figure 4.8: Monte Carlo simulation-optimization

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 [102], 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 even more relevant if larger instances are considered.

Table 4.8 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 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 [87].

In Table 4.9, 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 $\alpha = 0$ computed in percentage. Noticing that with $\alpha = 0$ the model has perfect information, thus we call the average percentage gap $EVPI_\alpha$.

Table 4.8: 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

Having perfect knowledge about the deliveries can decrease the delivery costs up to 137% ($\alpha = 100\%$). This justifies investments in methods able to forecast deliveries or business models, reducing uncertainty's impact. 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 toward the integration of big data and prescriptive analytics. As shown in [93], 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 4.9: 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

Table 4.10 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 investing in LDs and EVs attractive for managing 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

Table 4.10: 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

technology adoption curve [79]:

- 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 [93]).

Table 4.11 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. The savings are computed for both indicators 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 of 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 [93], 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.

Table 4.11: 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%

4.6 Scientific and industrial impact

This chapter introduced a new general ML heuristic for solving the VCSBPPSI. By carefully defining a set of features, it can compute reasonable 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 achievement paves the way for the exciting question of generalizing the good results obtained by the precedence rule in scheduling problems for other classes of problems with more complex decision rules. Moreover, the relatively straightforward application of the proposed heuristic enables it to be used in other problems characterized by binary decision variables.

The proposed heuristic achieved a good performance level and increased the maximum size of instances that can be considered, with an impact also in real-world applications. We claim that the proposed technique enables daily managerial decision support for companies engaged in the modern trends of consolidated logistics and 3-4PL usage. Furthermore, it is worth recognizing that the ML heuristic can produce a solution that can be recovered in the second stage. Thus, no particular care has to be taken to ensure the proposed solution's feasibility in normal conditions and that it can be safely applied by logistic managers in the field.

In future studies, the performance of the outlined approach can be improved by introducing real data features for each vehicle, e.g., CO_2 emissions and the delivery time. Moreover, we will consider further improving the proposed methodology in future studies. Other exciting 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 by bilevel models.

Chapter 5

Zone-based demand forecast for fleet optimization

Provided the total demand volumes for each city zone, it is possible to apply VCSBPP to optimize the required fleet capacity and answer the question of the number of vehicles to rent/buy. Such application of VCSBPP with the zoning-based demand forecast will be discussed in Section 5.2.

Another critical question arising in the city division application is the structure of the city zoning. For years, most logistic companies separated the city following the historical urban regions or the postal codes of each zone [104]. This approach can be significantly improved with modern unsupervised learning or clustering tools even without detailed data on the delivery orders [38]. Nevertheless, in this chapter, we provide the city's zoning regarding the homogeneity of the related delivery demand variance, grounding this knowledge on the usage of the real data provided by the last-mile delivery company. Furthermore, we handle the city zoning from the fleet management perspective, relying on the reduced demand variance and the possible outcomes of capacity and fleet distribution.

To better state the intuition behind the city zoning supporting the demand forecast, let us look at the last-mile delivery orders locations for different days. Figure 5.1 outlines the locations of delivery orders for two different days marked with different colors (red and blue dots), plotted on the map of the city of Antwerp, Belgium. The reader may notice that the overall pattern of the logistics request remains the same in the geographical sense. In other words, crowded city zones provide higher delivery demand than the zones with sparse and smaller buildings and sparse demand. Moreover, the regions with rivers, parks, or industrial buildings are logically characterized by the almost complete absence of orders. That confirms our guess on the necessity of the location information incorporation to the demand forecast and the importance of proper zoning of the urban areas for the logistic companies.

Further, we employ ML-based demand volume predictions as the scenarios for

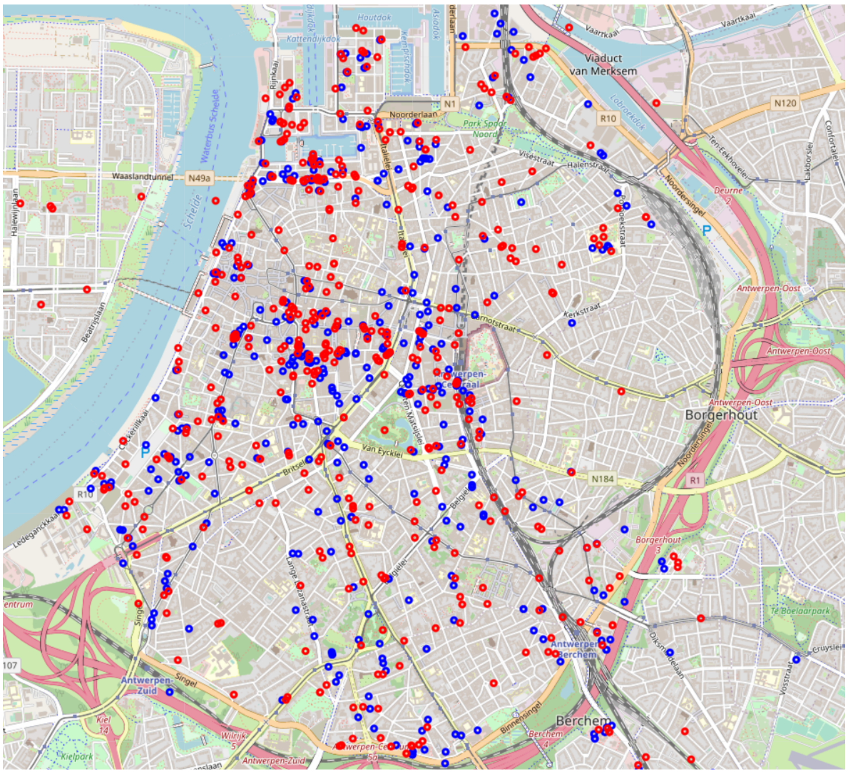


Figure 5.1: Order locations in the city of Antwerp for two days highlighted with red and blue colors for the first and the second day correspondingly. The overall pattern remains the same.

the ZCAP, providing a practical and data-driven approach to tactical decisions related to fleet capacity optimization.

of the logistics company in Subsection 5.2.3. The content of this chapter has been partially published in [40].

5.1 Data acquisition and elaboration

Data collection is a crucial step in any real-world application in operations research. For instance, generating realistic instances for optimization problems requires knowledge about the probability distribution that can approximate real-world phenomena.

During this process, we communicated with the logistic managers working on the logistic hub in one of the biggest last-mile delivery companies, which is desired to remain anonymous, in Antwerp, Belgium. information about the order location in terms of latitude and longitude, the method of delivery (vehicle type), the cycle of delivery, the timestamp of actual delivery handled along with the date, the number and weight of the parcels for each of the stop and the zip-code of the region of delivery. Additionally, it presents a set of internal data codes as production groups of orders, etc. These details can be useful for a general clustering approach but are considered irrelevant to this work application since we are particularly interested in geographical location correlations and the generalization of the introduced model‘s applicability for any logistics company. It is worthwhile to mention that the data was collected in 2019. Hence, the COVID-19 pandemic has not affected the provided distributions.

The data elaboration starts with the homogeneous division of orders with multiple parcels to provide a straight correspondence of each data row for each parcel. This assumption does not violate the data presentability since, in total, the fraction of orders with multiple packages is less than 10%, each representing a relatively small overall weight. Next, we decoded the vehicle numbers to provide a vehicle type and corresponding capacity. Lastly, the latitude and longitude of each order are mapped to the [0,1] range to exploit it as the features for the unsupervised clustering algorithms and secure the anonymity of the data.

The last step of the data elaboration is to provide the ability to construct a time series. Therefore, we used the provided timestamps and routing cycles to separate orders with different consequential working days, ranging from day 1 to day 60. Table 5.1 provides the obtained data sample. The reader can notice that we reformulate parcel weights as the theoretical volume each vehicle can handle up to a specific limit, depending on the vehicle type. Given the recommended weight load constraints for each vehicle type provided by managers, this assumption allows us to represent the vehicle capacities in the same terms.

Table 5.1: Elaborated data sample structure.

Day	Vehicle	Volume	Latitude	Longitude
1	Cargo bike	0.5	0.242795	0.421381
1	Cargo bike	2.02	0.235659	0.421326
1	Cargo bike	0.2	0.2358	0.424618
1	E-van	0.1	0.236413	0.424969
1	Cargo bike	0.53	0.235914	0.426217
1	Van	1.3	0.233161	0.424218
1	Cargo bike	0.96	0.231968	0.425738
1	E-van	0.08	0.231473	0.422909
1	Cargo bike	0.42	0.230358	0.416805
1	Cargo bike	0.54	0.230225	0.411416
1	Van	2.1	0.230205	0.411439

5.2 Zone-based demand forecast with VCSBPP

This section outlines the first approach of zone-based demand forecast for fleet management on the tactical logistic scheme. It involves the usage of the deterministic VCSBPP, outlined in Subsection 5.2.1. Afterward, we provide a forecast algorithm outlined in Subsection 5.2.2. This algorithm unifies the unsupervised K-means clustering with the regression utilized to obtain predicted values. Next, we check the performance of 4 different forecasters based on the ARIMA model, GP regression, NN with a simple one-layer structure, and the recent LSTM network. Finally, in Subsection 5.2.3 are outlined the computational results of the application of the proposed algorithm for the 50 days "on-the-fly" working scenario.

5.2.1 Variable Cost and Size Bin Packing Problem

approach to capacity optimization in the literature [44, 52, 15]. Firstly, VCSBPP was introduced along with the lower bounds and the suitable heuristic in [30]. Here, we briefly recall the VCSBPP formulation to approach it in the following sections.

Let $\mathcal{I}(|\mathcal{I}| \leq \infty)$ be the set of items to be loaded. Each item $i \in \mathcal{I}$ has a volume v_i. Let $\mathcal{J}(|\mathcal{J}| \leq \infty)$ be the set of available bins and let V_j and c_j be the volume and the cost of the bin j, respectively. It is worth mentioning that VCSBPP is a one-stage problem, in contrast to its stochastic version provided in Section 4.1. Therefore, we observe only one finite set of available bins $\mathcal{J}$. Notice that both bin costs and volumes are deterministic but not necessarily integers. Compared to the original problem, this improvement becomes available due to the application of a modern MILP solver and relatively small deterministic problem size, which makes the application of the heuristic redundant. However, it does not impact the overall performance, except that we can use the continuous volumes provided by the forecasters. Finally, the model considers the following binary decision variables:

- $y_j,\ j \in \mathcal{J}$ equal to 1 if bin j is booked and 0 otherwise;
- $x_{ij}, i \in \mathcal{I},\ j \in \mathcal{J}$ equal to 1 if the item i is packed in the bin j and 0 otherwise.

The VCSBPP formulation is the following:

$$\min \sum_{j \in \mathcal{J}} c_j y_j \tag{5.1}$$

$$\text{s.t.} \sum_{j \in \mathcal{J}} x_{ij} = 1, \qquad \forall i \in \mathcal{I}, \tag{5.2}$$

$$\sum_{i \in \mathcal{I}} v_i x_{ij} \leq V_j y_j, \qquad \forall j \in \mathcal{J}, \tag{5.3}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{5.4}$$

$$y_j \in \{0,1\}, \qquad \forall j \in \mathcal{J}. \tag{5.5}$$

The objective function (5.1) minimizes the total fixed cost of the selected bins. Constraints (5.2) ensure that each item $i \in \mathcal{I}$ is packed in a single bin. Constraints (5.3) ensure that each bin's capacity is not exceeded. Finally, Constraints (5.4) and (5.5) specify the decision variables' domain.

5.2.2 Zone-based demand forecast algorithm with VCSBPP

Thus, our starting goal is to group the orders in different clusters according to their geographical location to define zones. Therefore, we deal with the case of unsupervised learning, where the only guess on parameters required is the number of microzones. Among all the possible clustering algorithms in this field, we initially selected the K-means because it is easy to tune state-of-the-art algorithm [20, 19]. After the data is clustered with K-means, we can forecast the next-day demand for each defined microzone by approaching the time series of each cluster demand. In this experiment, we explore four different forecasting approaches, comparing the performances by computing the real-case scenarios of the data provided.

Firstly, we fit the Gaussian Processes (GP) regression model [100]. To apply it, we adopt the theoretical assumption of the joint Gaussian distribution of the random variables, which allows us to use the closed-form solution, which is quickly adaptable to changes in the data trends. The fitting process in the context of the GP regression is the optimization of the parameters of the mean and kernel functions, which are chosen to be linear mean and Radial basis functions, respectively. The output from the fitted model is the mean $m(t+1)$ and the variance $\sigma^2(t+1)$ on the next day (time $t+1$), which provides the safety of choosing the predicted value. More precisely, we recall that the 2σ bounds around the predicted mean guarantee the 99% probability of finding the next sample in the considered boundaries, which is proven in [106]. We use this fact to introduce safety in the next-day demand by choosing the prediction for the next time slot to be $v(t+1) = m(t+1) + \frac{1}{2}\sigma(t+1)$.

Another model to compare the performance is the ARIMA model, which consists of three basic components: differencing order, auto-regressive (AR), and moving-average (MA) terms. Each term is characterized by the associated parameters, which are d, q, p correspondingly [16, 34]. To set the parameters, we used the python package *pmdarima* model[1], able to fit the ARIMA model automatically defining the d, q, p coefficients.

The Neural Network-based forecast recently showed high accuracy and robustness to highly chaotic data, which is the case of demand delivery history. Hence, we choose to use two models based on it. As the primary case, we feed the 1 hidden layer NN with the data with an input lag of one day. In such a way, we move the window over the delivery history data for each cluster to train NN, and for the last step, we predict the expected volume. In addition, we use the Long-Short Term Memory (LSTM) network, which is the type of basic Recurrent Neural Network with a particular cell structure. The details of the LSTM structure can be found in [67, 113, 65]. For our purpose, we train the simple LSTM network without the hidden layers by feeding it with the data with the input lag.

Having the predicted total volume of orders for each of the defined clusters, we can move to the next step. Defined problem setting requires not just the conclusion on the overall volume of goods to move but the number and types of vehicles to use. To compute these quantities, we use the VCSBPP. In the problem formulation, the bin volumes are assigned in correspondence to the vehicles presented in the given data. In particular, the vehicles used are three types of vans with a big capacity (150-500 kg) and one type of cargo bicycle with a small capacity of 20 kg. In this experiment, the bin costs are defined, with the help of the company, to be $P_b = C_b + 0.2 * \max(C_b)$, where P_b is the cost of the selected type b of the bin, and C_b is the capacity of associated bins. This choice reflects the real case scenario of increased cost in terms of rent, fuel, and emissions for bigger vehicles. The addition of the 20% of the capacity of the largest bin represents the fixed costs as, for example, a driver's salary.

Concerning the definition of the items in the VCSBPP formulation, the predicted total parcel volume for the whole cluster is considered one item, which is the essential feature of our approach. This way, we assign each vehicle to a specific geographical microzone since each item corresponds to the cluster. However, the case of a small number of clusters leads to obtaining the items with a volume higher than any capacity available since they represent the sum of the volumes of the parcels for the whole cluster. In such a case, the VCSBPP is infeasible by definition.

To deal with this issue, we increment the number of clusters in the K-means algorithm. This approach has its profits and drawbacks. First, we have to notice that the K-means clustering algorithm introduces the new centers by separating the largest existing cluster. To prove it, we plot an example of obtained clusters on the same day data but with a different number of clusters in Figure 5.2. The reader can

[1]http://alkaline-ml.com/pmdarima/

notice that the newly introduced cluster lies in the zone with the highest number of orders of the previous setting, marked with the red circle. Thus, we obtain a process of division of the largest cluster. Generalizing this idea, by increasing the number of clusters, we lower the demand for each of them, thus balancing the demand and available vehicle volumes. This process allows us to tune the number of clusters by the simple increment of it whenever we receive the infeasibility of the VCSBPP. We can also observe that increasing the number of clusters leads to better accuracy of the predictions but increases the computational burden as a drawback. It is related to the fact that the forecast must be made once for each identified cluster. Moreover, it increases the size of the instance we have to solve with VCSBPP, but this impact is negligible due to deterministic problem formulation. Finally, obtaining an accurate prediction in the limit case of the same number of clusters and customers is impossible. However, in practice, it is never reachable until one of the customer's demands occupies the whole largest vehicle. The outlined approach is a practical way to balance the algorithm's performance in terms of accuracy and computational complexity.

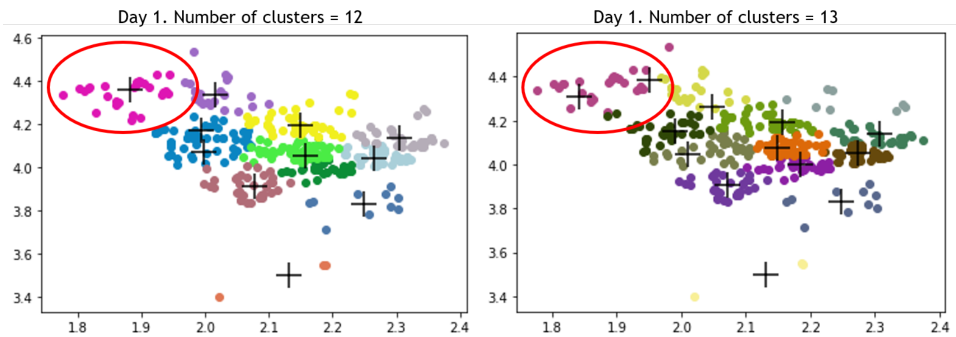


Figure 5.2: First-day demand clustered with K-means with 12 clusters (left) and 13 clusters (right). Highlighted the zone of new introduced cluster. The x and y axis represent the latitude and longitude values correspondingly.

The proposed overall prediction-optimization framework follows the approach in [114]. We run the algorithm "on the fly," starting the simulation with the assumption of the knowledge of the orders made in the first 5 days. After receiving this information, we perform a clustering step with a few starting clusters. Then, we perform the forecast of the aggregated volume of the orders for each of the clusters. Each predicted microzone demand volume is considered one item we want to pack into the available vehicles (bins). To do so, we optimize the VCSBPP with the Gurobi solver package [51]. If the problem is infeasible, we increase the number of clusters and repeat the procedure. The pseudo-code of the overall Capacity Forecast algorithm is outlined below as the Algorithm 4, where the n is the starting number of clusters, V_b is the set vehicle capacities given, and b is the indicator set of bin type for each vehicle available. The *Data* consists of the historical data of the orders done in the form of the geographical position of the order and the volume of the corresponding parcels, i.e., the last three columns of Table 5.1. Further, the $Data(c)$ contains the total volume of orders separated by each cluster c.

Algorithm 4 Capacity Forecast algorithm

```
procedure CAPACITY FORECAST( n, Data, V_b, b )
    while True do
        Clusters ← Clustering(n, Data)
        for c in Clusters do
            v_n ← Forecast(Data(c))
        end for
        Vehicles ← VCSBPP (items = v_n, bins = V_b(b))
        if Infeasible then
            n ← n + 1
        else
            return (Vehicles, v_n)
        end if
    end while
end procedure
```

The algorithm's output is the number and types of bins to book, the predicted volume for each cluster v_n, and the total cost incurred, computed by the value of the objective function of the VCSBPP. To test the performance of the proposed method, we compare the cost obtained using the outlined capacity forecast algorithm and the cost received from the solution of the VCSBPP using the actual order volumes observation of the next day. By doing so, we can estimate the relative loss of resources with the application of our approach compared to the theoretical case of perfect information. It is worthwhile noting that for the usage of the VSCBPP, it is crucial to get an estimation of the cost of the fleet assignment. In fact, since the VSCBPP is a discrete problem, the relation between the parameter of the instance and the final cost is nonlinear. Thus, evaluating our method just by using the error in the forecast demand may lead to an unfair or incomplete evaluation. This characteristic follows from the choice to consider discrete bins.

5.2.3 Experimental results of zone-based demand forecast with VCSBPP

Meanwhile, the first 5 days are considered passed, and the demand is known. The data acquisition for this experiment is described in the section 5.1. Then, to forecast the demand volume, we construct a multiple time series of the daily volumes of the parcels for each identified cluster, assuming the knowledge of the previous 5 days delivery history and enriching it with new days while the algorithm runs daily.

As mentioned above, the main step of the algorithm validation is the comparison of the objective function cost values of the VCSBPP solution obtained by passing into it the predicted order volumes and the real volumes. For simplicity, the obtained solutions are called the predicted and real (actual) demand VCSBPP solutions accordingly. Since the cost and capacity of the bins in our setting are linearly dependent, it is possible to consider the difference in the cost as the difference in the capacity required to rent predicted by our algorithm and the actual required capacity of the next day. We expect to have equal or higher values of the capacity of the predicted solution mainly due to considering the total volume for each cluster as one item. Thus, our solution provides a slight approximation error by construction. Nevertheless, it is possible to fix this issue only by introducing a suitable optimization problem, which will be discussed in the following sections.

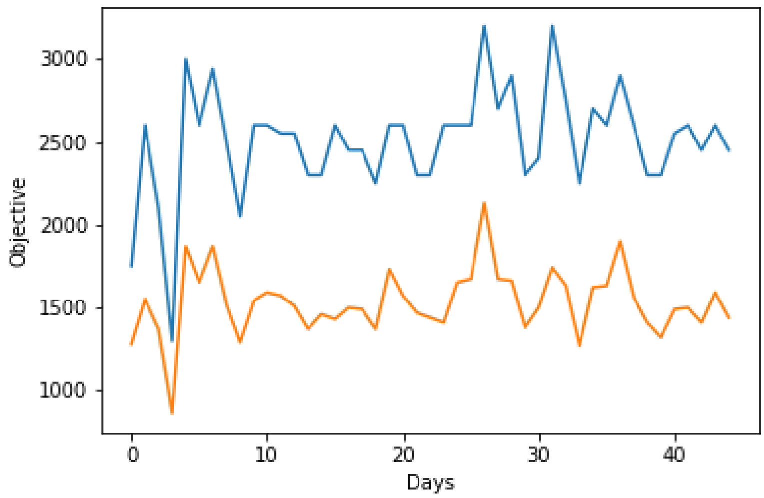


Figure 5.3: The daily comparison of the values of the objective function for the solutions of the VCSBPP for the predicted with GP regression (blue) and actual (brown) demand.

Particular attention in results elaboration should be devoted to the case of the GP regression. It provides a safety treatment in the forecast by its construction since the predicted volume is computed as $m(t+1) + \frac{1}{2}\sigma(t+1)$. Thus, a higher amount of predicted volumes follows with the higher capacity requirements. Figure 5.3 plots the daily comparison of the VCSBPP objective function values for the demand predicted with GP regression and actual data. On the y-axis, we plot the obtained objective function values of the VCSBPP solution with predicted and actual demand volumes. On the x-axis, we outline the day number in the simulation. The reader may notice that the predicted solution is way higher than the actual. On the other hand, from the tactical decision point of view, there must be no points with the predicted capacity being lower than the real VCSBPP solution, i.e., there are no scenarios in which the company can not satisfy the demand. Hence, this forecasting method assumes maximal safety in the decision about the required capacity. However, it is required to carefully choose the σ bound for the predicted values to be safe in satisfying the demand and not overestimate the demand. In this case, we observe that a capacity that can be roughly compared to one van distinguishes the predicted solution with the highest capacity from the actual one. That is acceptable from the tactical decisions point of view since the solution perfectly satisfies the completely uncertain demand, but further improvement in the forecasting mechanics could decrease this gap. However, the reader can notice that the predicted capacity with the GP regression perfectly follows the shape of the actual demand. Thus, with a manual tune and update of safety bounds (σ multiplier), it is possible to obtain an accurate empirical solution for the provided task.

The required capacities obtained by ARMIMA, NN, and LSTM-based forecasts coupled with VCSBPP are outlined in Figure 5.4. On the y-axis, we plot the obtained objective function values of the VCSBPP solution with predicted and real demand volumes. On the x-axis, we outline the day number in the simulation. The reader can notice that their output accurately follows the pattern of the actual demand solution objective function values. However, there are multiple points (days) where it is lower than the real solution, meaning the demand would only be partially satisfied. Thus, the introduction of some safety to the decision is required. The fixed threshold for each cluster's predicted volumes could be one option. However, introducing such a threshold is a highly empirical task, and it should be treated from the financial perspective of the problem, which is out of the scope of this application.

To better compare the forecasters, we address the RMSE metrics. The averaged among clusters demand prediction RMSE for all the forecasting models is outlined in Figure 5.5. On the y-axis, we plot the obtained RMSE values of the forecasters. On the x-axis, we outline the day number in the simulation. This comparison shows forecasters' performance regardless of the optimization problem usage. Still, in such complex correlations and constantly changing trends of demand forecasting data, it is impossible to expect a convergence of any forecast to a stable error level. Nevertheless, we can observe that GP regression is the most stable method; the LSTM network and ARIMA model show the trend to improve the performance with the data enrichment; the one-layer NN-based forecast appears to be very accurate in the beginning and hence fast to train, but loses the performance with time, that could be due to the overfitting issues.

The VCSBPP solution structure for all four forecasting methods is outlined

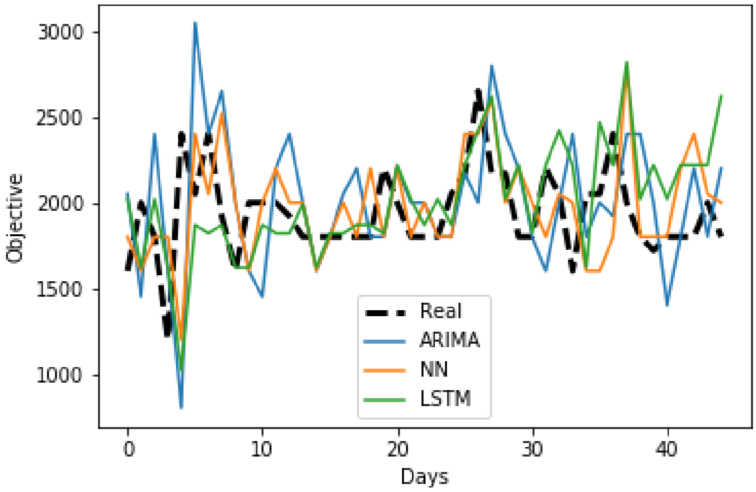


Figure 5.4: The daily comparison of the values of the objective function of solutions of the VCSBPP with perfect information and the three forecasting methods: ARIMA, NN, and LSTM.

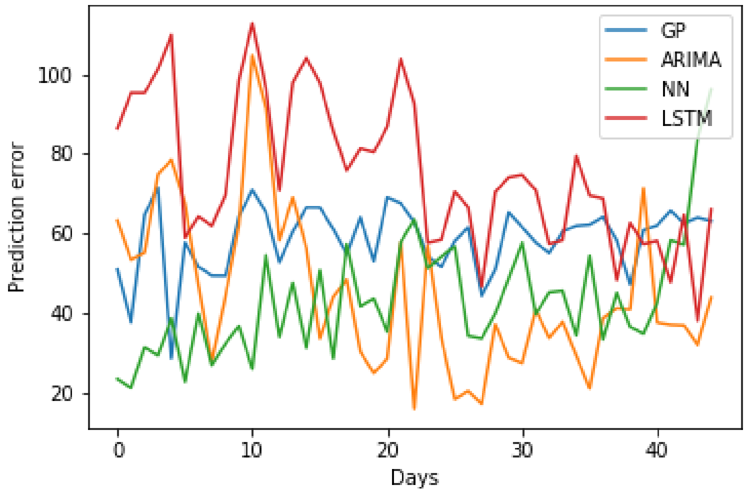


Figure 5.5: The averaged among clusters demand prediction Root Mean Square Error for the four different forecasting methods.

in Table 5.2. It shows the percentage of the bins chosen by different algorithms from all available capacities. The vehicle type is outlined with the corresponding capacity. This percentage is averaged over all the solutions of the 45 days. Hence, the sum of it for each of the methods is not supposed to be 100%. From the table, we can check which capacity is preferred by the logic of the overall framework and the influence of the particular forecasting approach. The results can be compared to the actual VCSBPP solution structure with the full knowledge of the real demand. The bin costs and the mean of Root Mean Squared Errors (RMSE) for the total demand volume prediction in the 45 days are also outlined.

Table 5.2: The VCSBPP solution structure comparison.

Capacities, kg	150	500	300	20	RMSE
Costs, eu	250	600	400	120	-
Real demand solution	52.86%	23.21%	37.05%	68.75%	-
GP prediction	9.29%	79.46%	35.98%	2.68%	58.5
ARIMA prediction	2.75%	61.76%	15.69%	0.98%	44.8
NN prediction	1.18%	61.27%	16.18%	0.49%	43.4
LSTM prediction	3.53%	52.49%	23.24%	0.49%	75.4

From the solution structure table, it is clear that the proposed approach is prone to choose the biggest vehicle available regardless of the forecasting tool. This is because we consider the collected volumes of the orders inside each cluster as one item. However, this is not a problem for the case when the set of bin types does not differ much in capacities, as in the typical supply chain problem [26] and in last-mile and e-commerce applications, where the parcel sizes are small [9]. Moreover, this issue can be fixed by introducing the appropriate optimization problem, as discussed below.

- The introduction of a suitable optimization problem of assigning vehicles (capacities) to the zone is required to improve the approximating issues of the whole framework and to cope with the zone-based demand forecast.
- Introduction of the more training data available is essential to provide a reasonable background of the forecasters with NN-based regressions.
- The optimal city zoning from the fleet managing perspective has to be elaborated.

Outlined issues are elaborated on in Section 5.3.

5.3 Zone-based demand forecast with ZCAP

The outlined problem refers to the different demand volume scenarios, which can be obtained with the forecasting algorithms outlined in Subsection 5.3.2. In addition, we discuss the methodological aspects of the whole framework improvements, compared to the existing approach outlined in Section 5.2. In particular, we discuss two strategies for city zoning in Subsection 5.3.3 and the training data enrichment process provided in Subsection 5.3.4. Finally, we discuss the computational results of the provided approach in Subsection 5.3.7 and the scientific and industrial impact of the provided results in Subsection 5.3.8.

5.3.1 Zone Capacity Assignment Problem

This problem aims to assign a fleet of capacities (vehicles) to provided city zones with minimal cost. This assignment considers the constraints on uncertain demand volume satisfaction for each zone with the provided vehicle capacities. In contrast to existing approaches, such as VCSBPPSI, we assign capacities to the urban areas' zones. This change improves the accuracy of the overall demand forecast and fleet capacity management framework, as proven in [40].

To introduce demand uncertainty, let us consider a set of scenarios $\mathcal{S}$ with a finite cardinality S. Let $\mathcal{J}$ be the set of identified city zones where delivery demand must be satisfied with the available vehicles. We consider each sample of prediction data as the different scenario $s \in S$ of problem realization with the probability π^s. Therefore, for each zone $j \in \mathcal{J}$, we assign the demand $V_j^s,\ j \in \mathcal{J},\ s \in \mathcal{S}$ for each scenario.

Let $\mathcal{I}$ be the finite set of vehicles available in the first stage of the problem. Each vehicle available in the first stage is characterized by the non-negative cost $c_i,\ i \in \mathcal{I}$ and capacity $v_i,\ i \in \mathcal{I}$. Analogously, we define the set of vehicles available on the second stage of the problem $\mathcal{K}$ with corresponding costs $c_k, k \in \mathcal{K}$ and capacities $v_k, k \in \mathcal{K}$. Let us consider the following decision variables:

- $y_i = 1$ if vehicle $i \in \mathcal{I}$ is used, and 0 otherwise
- $z_k^s = 1$ if vehicle $k \in \mathcal{K}$ is used in scenario $s \in \mathcal{S}$, and 0 otherwise
- $x_{ij} \in [0,1]$ is the fraction of capacity v_i of vehicle $i \in \mathcal{I}$ assigned to the zone $j \in \mathcal{J}$
- $\theta_{kj}^s \in [0,1]$ is the fraction of capacity v_k of vehicle $k \in \mathcal{K}$ assigned to the zone $j \in \mathcal{J}$ in realization $s \in \mathcal{S}$

Using the above notation, the zone capacity assignment problem can be stated as follows:

$$\min \sum_{i\in\mathcal{I}} c_i y_i + \sum_{s\in\mathcal{S}} \pi^s \sum_{k\in\mathcal{K}} c_k z_k^s, \tag{5.6}$$

$$\text{s.t.} \sum_{j\in\mathcal{J}} x_{ij} \leq y_i,\ \forall i \in \mathcal{I} \tag{5.7}$$

$$\sum_{j\in\mathcal{J}} \theta_{kj}^s \leq z_k^s,\ \forall k \in \mathcal{K},\ \forall s \in \mathcal{S} \tag{5.8}$$

$$\sum_{i\in\mathcal{I}} v_i x_{ij} + \sum_{k\in\mathcal{K}} v_k \theta_{kj}^s \geq V_j^s,\ \forall\, j \in \mathcal{J},\ \forall s \in \mathcal{S} \tag{5.9}$$

$$y_i \in \{0,1\},\ \forall i \in \mathcal{I} \tag{5.10}$$

$$z_k^s \in \{0,1\},\ \forall k \in \mathcal{K},\ \forall s \in \mathcal{S} \tag{5.11}$$

$$x_{ij} \in [0,1],\ \forall i \in \mathcal{I},\ \forall j \in \mathcal{J} \tag{5.12}$$

$$\theta_{kj}^s \in [0,1],\ \forall k \in \mathcal{K},\ \forall j \in \mathcal{J},\ \forall s \in \mathcal{S} \tag{5.13}$$

The objective function (5.6) aims to minimize the total cost of vehicles used in the first stage and the expected cost of vehicles used in the second stage. Constraints (5.7) ensure that each assigned first-stage vehicle would impact the final cost. Constraints (5.8) ensure that each assigned second-stage vehicle would impact the final cost. Constraint (5.9) represents the absolute predicted demand satisfaction requirement for each scenario. Finally, Constraints (5.10-5.11) are the integrality constraint on the variables y_i and θ_{kj}^s. Meanwhile, Constraints ((5.12) - (5.13)) regulate that the capacity assignment variables remain continuous, enabling travels to multiple zones.

5.3.2 Demand forecast algorithm with ZCAP

This subsection is devoted to the Zone-based demand forecast algorithm, which employs ZCAP. The general idea of the approach follows the "divide and conqueror" strategy, presented in [40]. In detail, we follow the main steps of the outlined methodology: divide the urban area into N_{zones} zones, forecast the demand volume for each zone, and use the prediction results to provide a fleet requirement. However, we adapt this strategy to the ZCAP, improving the robustness and accuracy of the entire framework.

The algorithm's first step is the city grid separation into different zones. With the grid city separation, we collect the delivery demand history for each identified zone by assigning each customer in the data to one zone. Summarizing the demand volume (parcel weight in case of data available) for each zone and each day, we obtain the time series, where each time leg (day) corresponds to the summed demand volume. These time series are available for each zone, thus formulating a two-dimensional data array with dimensions N_{zones} x *Days*, where *Days* is the number of working days in the data available up to a current moment. It is worth mentioning that the identified zones without assigned orders in the entire dataset are not considered in the following procedure since due to the city infrastructure (parks, highways, etc.).

Further, this data is fed to a set of forecasters trained for each identified zone. Each time series of specified demand volumes of the corresponding zone forms a training data sample for the forecaster. After training, we receive a predicted value for each zone with the chosen forecaster based on the given time series. This value is considered the "next-day" demand for each zone j in one possible realization of uncertainty, i.e., one scenario $s \in S$ in ZCAP outlined in Subsection 5.3.1. the cardinality of the set of scenarios S for ZCAP. Finally, the last step of the proposed approach is the optimization of ZCAP, identifying the fleet requirements for the next time slot.

Each forecasting approach used in the outlined algorithm and its short description with the parameters used is provided in the following:

- **ARIMA**. Forecast based on the well-known ARIMA model [43]. To set the parameters, we used the python package *pmdarima* model[2], able to fit the ARIMA model automatically defining the coefficients.
- **GP**. A Gaussian Process-based regression can be applied with the assumption of joint Gaussian distribution of random variables [100]. We follow the methodology in [40], but reduce the predicted volume to be $mean(t+1) + \frac{3}{8}\sigma(t+1)$, given that it satisfies the next day demand.
- **RF** A forecaster based on a Random Forest (RF) regression. The time series is fed to RF with a sliding window of 10 sample size. The following approaches are fed with the same sliding window methodology of 10 samples. To construct RF, 100 estimators are applied.
- **XGBoost** Forecast based on XGBoost classifier. We identified the 30 estimators parameter as the best fit for the provided forecast scenario.
- **NN_1L** NN-based forecast. We follow the approach provided in [40]. In this setting, we used the 1 hidden layer of 100 neurons with "Relu" activation function.
- **NN_ML** NN-based forecast with augmented network structure of 2 hidden layers with 32 and 8 neurons.
- **DNN** NN-based forecast with 3 hidden layers, which can be classified as the Deep NN. We set 64, 32, and 16 neurons for these DNN hidden layers.

[2]http://alkaline-ml.com/pmdarima/

- **LSTM** Long-Short Term Memory network-based forecast. We train the simple LSTM network with one hidden layer of 100 neurons by feeding it with the data with the input lag [40, 67, 113].
- **LSTM_ML** Long-Short Term Memory network-based forecast with a multilayer structure consisting of 2 layers with 64 and 32 neurons.

The sliding window size indicated for each forecasting method is tuned manually to obtain the best forecast accuracy in terms of the Rounded Mean Squared Error (RMSE). Most forecasting methods are implemented with the help of the Scikit-learn library [86]. Meanwhile, the NN-based forecasters are tuned with PyTorch library [85].

The main parameter to be tuned in the proposed setting is the number of zones N_{zones}, corresponding to the data aggregation level. The low number of zones leads to underestimating the underlying city infrastructure complexity and poor forecasting accuracy. On the contrary, increased N_{zones} leads to a low zoning data aggregation level, which provides forecasters with poor generalization ability. In other words, we expect a high error level on the test dataset, similar to overfitting issues in ML nomenclature.

Therefore, we must identify optimal N_{zones} for each time slot since the data available is enriching daily with the new orders becoming known. Experimental results in Subsection 5.3.5 show that the variance dependency function on the zone number is not convex. Nevertheless, we can apply a global optimization algorithm to identify optimal N_{zones} and underlying city separation in the daily scheme. For this purpose, we involve GP regression as the global optimization scheme for parameter N_{zones}. This approach allows us to approximate the optimal N_{zones} regarding the obtained forecasters' variance with a minimal function sampling, which is the most computationally heavy part [99]. Moreover, the methodology of GP regression implies the closed-form solution with the hyper-parameters auto-tuning, which provides a more robust environment for final users as logistic managers.

The flowchart of the introduced algorithm is outlined in Figure 5.6. The reader can notice that the algorithm's first step is dividing the past delivery D days data according to a grid on the map with N_{zones} cells. If the obtained N_{bar} with GP regression coincides with the current N_{zones} parameter, we continue to ZCAP. If not, we repeat the outlined N_{zones} optimization procedure with setting $N_{zones} = N_{bar}$.

The entire daily zone-based demand forecast with the ZCAP is outlined as Algorithm 5. It is worth mentioning that this algorithm takes a set of all outlined classifiers, the entire dataset available for the current day, capacities of vehicles available (C_b), as well as its available number b. The outlined algorithm returns a set of vehicles required to serve the next-day demand and the predicted demand volumes. In this algorithm, with a slight abuse of notation, we set each demand forecast as one scenario $s \in S$ for ZCAP. In addition, we denote $GPregression(var(V_c))$ as the procedure of fitting the GP regression to the function, which takes N_{zones}

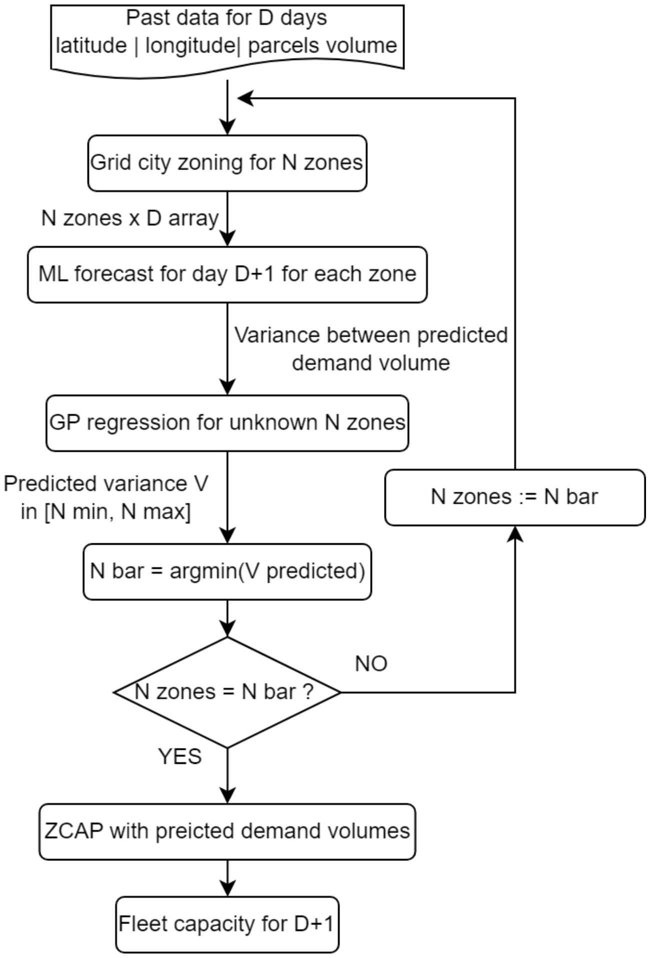


Figure 5.6: Zone-based forecast with ZCAP N_{zones} optimization flowchart.

as input and provides the resulting total standard deviation of predicted values for each forecaster.

5.3.3 City zoning: clustering vs. grid

Division of the urban service area into different regions is a widely adopted practice by logistic managers. For years division of the city by historical areas or with postal codes was the basic knowledge to approach the tactical decisions in the last mile. Furthermore, given this division, the fleet is usually assigned to each fixed zone. This intuition lies behind the formulation of ZCAP, outlined in Subsection 5.3.1. However, recent studies showed that the zoning of the logistics service area with modern clustering tools dramatically improves the quality of modeling approaches, reducing daily demand variance for each zone and thus implicitly providing stronger spatial correlations [38]. Therefore, the optimal city zoning for

Algorithm 5 Fleet optimization algorithm with ZCAP

1: **procedure** DAILY FLEET OPTIMIZATION($Data$, $Forecasters$, C_b, b)
2:     **while** True **do**
3:         $Zones \leftarrow Grid\_separation(N_{zones}, Data)$
4:         **for** s in $Forecasters$ **do**
5:             **for** c in $Clusters$ **do**
6:                 $V_c^s \leftarrow Forecast(Data(c), method = s)$
7:             **end for**
8:         **end for**
9:         $\hat{N_{bar}} \leftarrow \arg\min(GPregression(var(V_c)))$
10:         **if** $\hat{N_{bar}} == N_{zones}$ **then**
11:             $Vehicles \leftarrow$ ZCAP $(Demand = V_n^s, bins = C_b(b))$
12:             **return** $(Vehicles, V_n^s)$
13:         **else**
14:             $N_{zones} = \hat{N_{bar}}$
15:             Go to Step 3
16:         **end if**
17:     **end while**
18: **end procedure**

uncertain logistics models must be elaborated.

Introduced ZCAP in Subsection 5.3.1 provides more variability for the zoning approaches. Therefore, we decided to compare the clustering of the previous orders by introducing a grid on the city map with the K-means clustering approach outlined in [40]. The grid is chosen to symmetrically separate the entire provided region with equivalent segments on latitude and longitude. Unfortunately, the more sophisticated grid sells require more parameters and are thus less adaptable to further optimization. Nevertheless, this question requires further research.

We compare the performance of the two outlined approaches on the aforementioned dataset of the last-mile delivery demand history in Antwerp, Belgium. First, we randomly select 8 different days in the data and apply grid and K-means clustering. Then, for the following experiment, we fixed the number of zones to be $N_{zones} = 120$ to check both methods' performance.

Figure 5.7 outlines the demand volume distribution for each zone obtained with a grid separation on 8 different working days. The equivalent demand distribution on clusters obtained with the K-means for the same days is outlined in Figure 5.8. We plot each zone's associated demand volume on the y-axis and the zone number on the x-axis. The zones are reordered in demand volume decreasing order for each outlined day in both figures. In addition, the demand volume for each day is marked with a different color. Therefore, we obtain a demand volume distribution for multiple days with different zoning approaches.

The reader can notice that the peak volume for the grid distribution is higher than with the K-means clustering. This peak is related to the group of particularly loaded regions in the city, such as commercial districts. Meanwhile, the demand distribution obtained with K-means clustering is smoother and characterized by a so-called "fat tail", meaning there are more clusters with low but not zero demand. However, in both cases, we receive a similar form of distribution that confirms an equivalence between both methods.

Therefore, we plot the received demand volume variance with both methods for the entire dataset (60 working days) in Figure 5.9. We plot each zone's associated demand volume variance on the y-axis and the zone number on the x-axis. In this plot, the zone number remains fixed for each day but is reordered in a variance-decreasing order. At this point, we receive completely different distributions from the demand forecast perspective since we aim to reduce the demand variance. From the integral of both distributions (surface under function), we receive 26.7% overall variance reduction with grid separation. That reduction is related to the rigid coupling of the spatial information with the grid clustering compared to the K-means. Moreover, identified zero-demand zones are excluded from further elaboration with the forecasters, significantly reducing the computing time of the following prediction step. Therefore, we provide the experimental results based on the grid separation of the city in the following work.

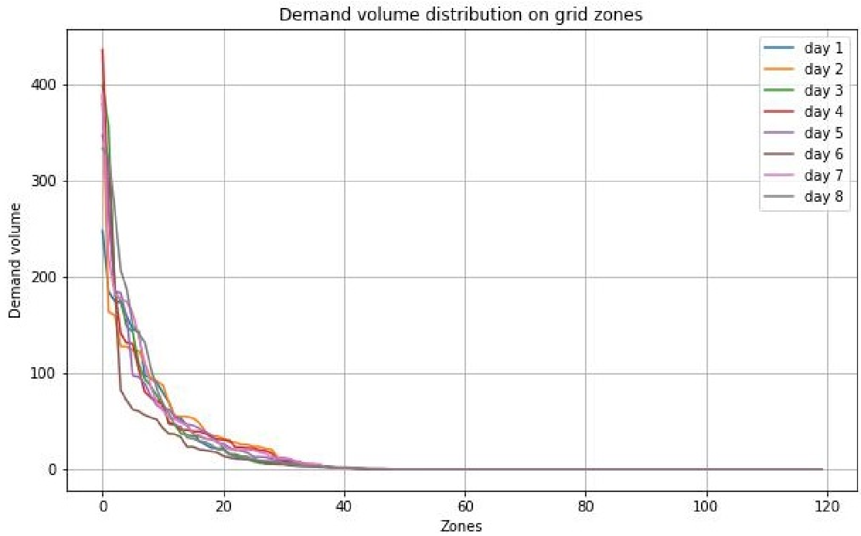


Figure 5.7: Distribution of demand volume for 120 zones obtained with a grid city separation for 8 different days.

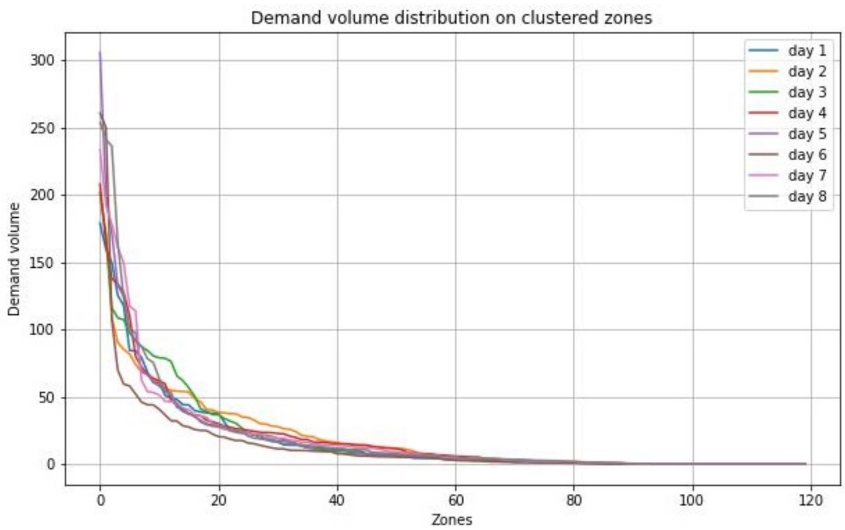


Figure 5.8: Distribution of demand volume for 120 zones obtained with a K-means city separation for 8 different days.

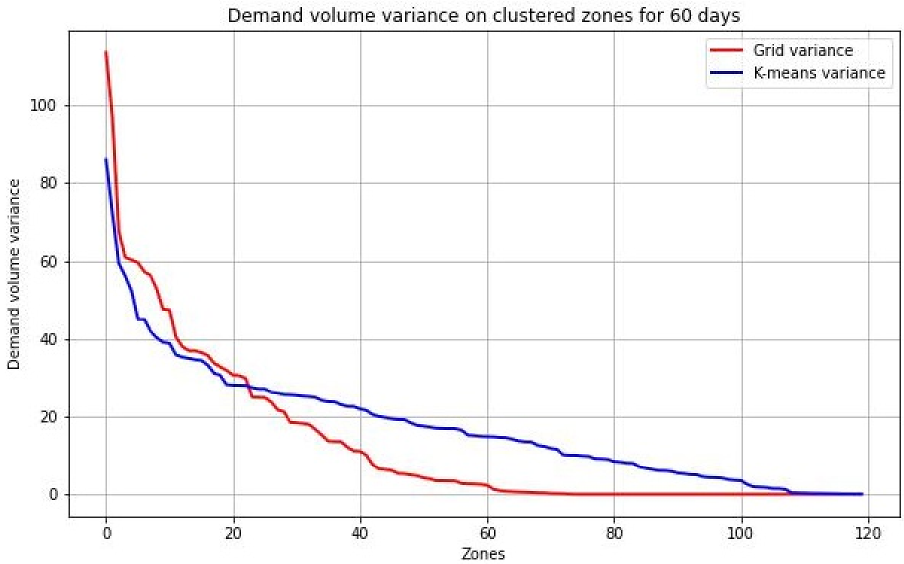


Figure 5.9: Distribution of demand volume variance for 120 zones obtained with a K-means clustering (blue) and grid city separation (red) for 60 working days.

5.3.4 Demand data enrichment for Zone-based demand forecast

Most modern forecasting methods rely on the enormous sets of data available (Big Data phenomena), providing a complex forecasting mechanism enabling approximation of the real-world data-generating process. This requirement is particularly relevant for forecasters based on Deep NN (DNN) models, which assume a complex multi-layer structure of the NN employed. Therefore, this section discusses the methodology of demand data generation for further training the forecasters.

Delivery demand data generation is a complex task, which requires derivation of the probability distribution for parcels' weight and location of order. The basic idea of the proposed approach follows the city division into quadrants of different demand densities, outlined in [29]. However, in contrast to the previous approach, we introduce the probability of order appearing for each region. Therefore, we apply the grid of the urban area with N cells. To each cell of the grid $n \in [1, N]$, we assign a probability p_n, which is computed as follows:

$$p_n = \frac{q_n}{\max(Q_n)}, \tag{5.14}$$

where $q_n \in Q_n$ is the number of orders in the available data, and Q_n is the vector containing the order numbers for each identified cell. In such a way, we provide a two-dimensional probability distribution from which we can sample the training data, which improves the approximating ability with the increase of the parameter

N. Moreover, we obtain a continuous probability distribution in the limit case of $N \longrightarrow \infty$.

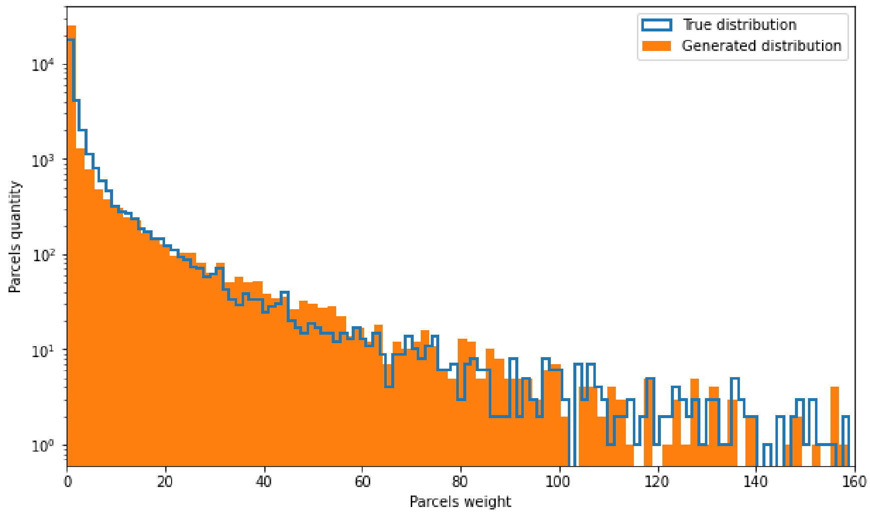


Figure 5.10: Histogram comparing the true parcel weight distribution with the generated chi-squared distribution with 0.15 degrees of freedom.

Concerning the actual parcel weight, it requires manually deriving the distribution. Figure 5.10 shows a histogram comparing the true parcel weight distribution with the generated Chi-squared distribution with 0.15 degrees of freedom and the same sample size of the whole dataset available. The parcels' weight is outlined on the x-axis with the bars, and the parcels' quantity is outlined on the y-axis. To match the values of the parcels‘ weight, we multiply every generated value by a re-scaling constant coefficient of 20. As the reader can notice, the generated sample follows the true data, and the obtained $X \sim 20\ \chi^2(0.15)$ is suitable for this particular problem case. Nevertheless, the Chi-squared distribution can be suitable to approximate every case of parcel weight distribution. Meanwhile, the daily total orders number parameter follows the normal distribution with the mean and variance of the actual data available.

Lastly, we study the dependence of the zone number parameter N on demand volume variance for each zone in the outlined data generation process to improve its accuracy. For this purpose, we generate a sample of demand data with the outlined procedure of the same size as the whole data available and N parameter ranging from 10 to 200 clusters. Then, we check the standard deviation of demand for each zone in the unified data sample (generated and available). Figure 5.11 outlines each zone's resulting demand volume standard deviation with the zone number parameter N change in the data generation process. On the x-axis, we plot the changing zone number parameter N; on the y-axis, we outline the demand volume standard deviation between zones. The reader may notice that the demand variance decreases with a more detailed city grid (higher zone number) up to a certain parameter $N = 150$. After this point, the more detailed city grid provides more demand variations for each zone, resulting in an "overfitting" of the proposed data generation scheme. Therefore, we adopt the parameter $N = 150$ suitable for the provided city area.

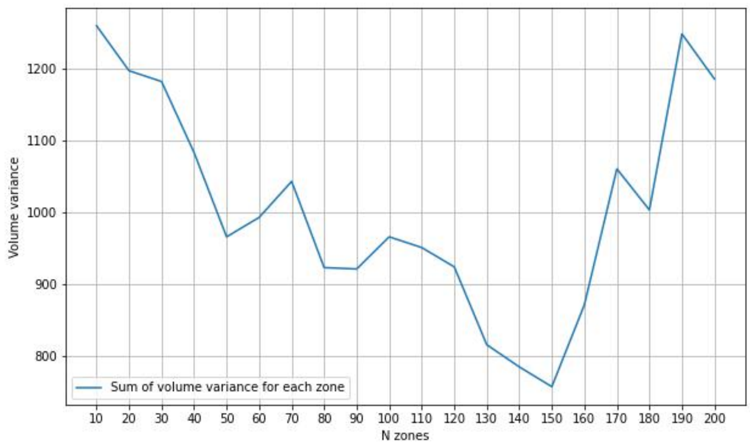


Figure 5.11: Total demand volume standard deviation for each zone with the change of the zone number parameter N zones in the data generation process.

5.3.5 Zone number influence to demand forecast

To check this correlation, we fixed the day $D = 25$ and changed the number of zones parameter N_{zones}. The mean of prediction error percentage between each forecaster, introduced in Subsection 5.3.2, with the change of N_{zones} is outlined in Figure 5.12. On the y-axis is plotted the error between the actual demand volume and predicted value in percentage, and on the x-axis are outlined zone number parameters. The reader can notice an improvement in the accuracy of all forecasting methods around the 64 and 99 identified zones, corresponding to the best city zoning settings.

Meanwhile, Figure 5.13 plots the overall standard deviation of predicted demand volumes with all aforementioned forecasters for different zone number N_{zones} parameters for the same day. On the x-axis, we provide the number of zones (N_{zones}); on the y-axis, we plot the standard deviation between the predicted volumes of each forecaster.

The reader can notice that the obtained function is not convex. However, the point of 64 zones provides the least variance between forecasters. This, in fact, corresponds to the parameters of the least forecasting error in Figure 5.12, which

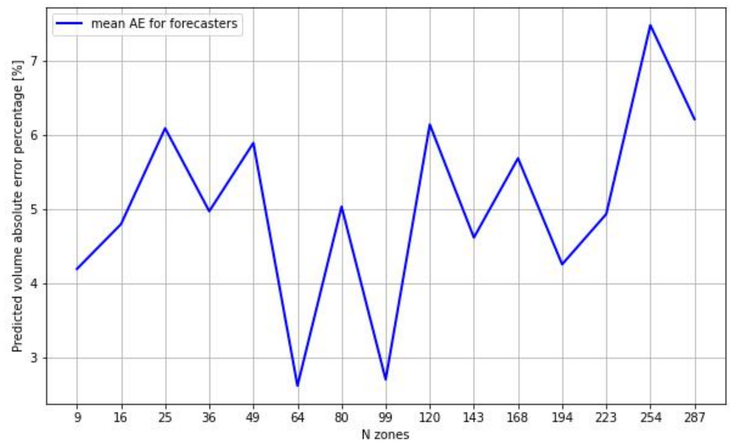


Figure 5.12: Total absolute error percentage of predicted demand volumes sum with different forecasters for different N_{zones} parameter.

confirms our guess that the least variance between forecasters provides the best accuracy for each forecaster. Moreover, the reader can notice that the accuracy of forecasters, in general, improves with the decrease of variance between predicted values. Therefore, we obtain the suitable parameter to identify the optimal city zoning for the zone-based demand forecast since improvements in forecast accuracy provide more accurate fleet management with the ZCAP as well.

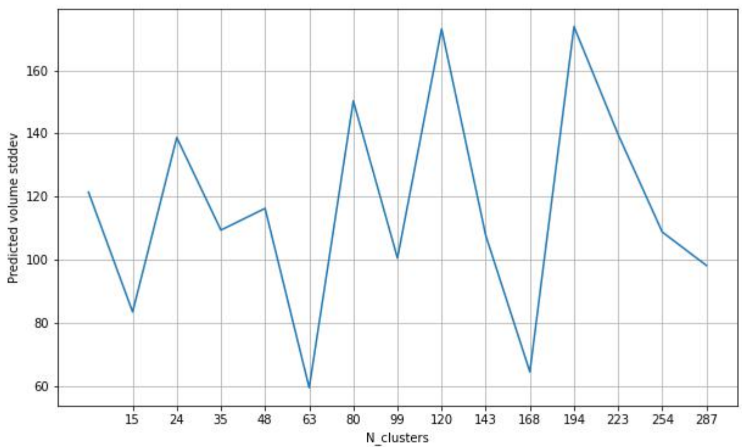


Figure 5.13: Total standard deviation of predicted demand volumes sum with different forecasters (y-axis) for different N_{zones} parameter (x-axis).

Although the provided experiments propose a good intuition on the optimal city zoning, the forecasting accuracy can be significantly improved by defining the optimal zone number N_{zones} for each working day. To show it empirically, we provide the resulting forecasting accuracy for 3 different days with the zone number change. For this result, we outline the forecasting scenario for 3 non-consecutive days with the step of 5 days with changing zone number parameters in the city grid. In Figure 5.14 are outlined the resulting forecasting accuracy mean (left picture, y-axis) and standard deviation of the predicted zone volumes (right picture, y-axis) with the changing zone number parameter N_{zones} on the x-axis for 3 different days. This setting allows us to catch the forecasters' performance with the data enrichment and zone number changes. For the day 25, we can see that the setting of 64 zones shows the lowest prediction error and variance. In addition, we receive around 5% accuracy improvement for all the classifiers compared to the worst scenario of the 49 zones, which is a considerable improvement. The division to 64 zones corresponds to the optimal zone number for the day 25. However, with the following forecast simulation and data enrichment, the optimal zone number in terms of forecasters' accuracy is increasing. For the day 30, we can observe a minimal absolute prediction error at the point of the 99 zone number parameter. This setting follows with the minimal standard deviation of the predicted values in the same point of 99 zones for this day. Moreover, the same result is observable for the day 35 in the point of 120 zones. Herefore, with the data enrichment, the city zoning should become more detailed to provide the best accuracy of the underlying city structure. Moreover, these results confirm the previous guess that it is necessary to identify the best zoning parameter for each forecasting scenario, which is possible to do by relying on the standard deviation of the predicted values for all the forecasters available.

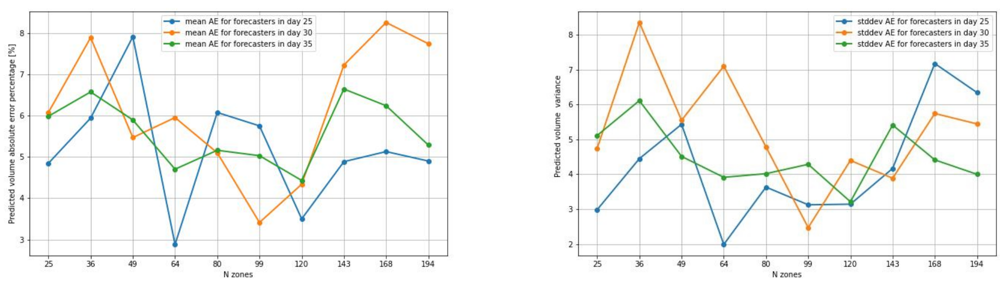


Figure 5.14: Absolute prediction error mean (left) and standard deviation of the predicted zone volume (right) of forecasters with different zone number parameters for three different days.

5.3.6 Experimental setup for ZCAP and VCSBPPSI

This subsection briefly recalls the experimental setup details used in this work. To validate the performance of the proposed zone-based demand forecast with the ZCAP approach, we compare it with the existing baseline method of the solution of the VCSBPP with the stochastic items (VCSBPPSI) [27, 26]. The experiment was performed on the real data of 60 days delivery orders in Antwerp, Belgium, with the assumption of knowledge of the first 10 days. During the simulation, the data sample increased accordingly to the new daily data available, and all the previous days were used as the training data for forecasters. Moreover, we enrich the "known" days for 200 days with the data generation process outlined in Subsection 5.3.4. To compute the solution of the *ZCAP* and VCSBPPSI employed off-the-shelf solver Gurobi 9.1.2 [51]. All the computation was performed on the machine with the Intel Core i7-9750H CPU @ 2.60 GHz. Due to the high computational complexity of VCSBPPSI, we apply the Progressive Hedging heuristics introduced in [42], and refer to the approximated solution as the VCSBPPSI result.

The VCSBPPSI model assumes that the probability distribution for the uncertain demand volume is given. However, in the case of data available, it is possible to manually derive a suitable probability distribution to adapt it to the given data. We refer to the data generation process description in Subsection 5.3.4 for the details on probability distribution derivation. We make the same hypothesis of Chi-squared distribution ($\mathcal{X} \sim 20\ \chi^2(0.15)$), which is confirmed to be the most suitable for the data available (see Figure 5.10).

Next, we must set up the fleet parameters, such as vehicle types, capacity, and costs. In the provided data, we identified the presence of 3 vehicle types: cargo bikes, e-vans, and light-duty vans. The vehicles' capacities correspondingly were 100, 300, and 600 kilograms. To model the most realistic scenario, we follow the number of available vehicles in the data to have 12 bikes, 12 e-vans, and 6 regular vans in the first and second stages of both problems. Lastly, the cost of the vehicles is subjected to uncertainty and generated following the procedure outlined in [27, 11] and practitioners' suggestions. The first-stage cost c_i is generated as vehicle capacity powered to q, where q equals two multiplied by uniformly random value with a different range, depending on vehicle type. The second-stage costs c_k are computed similarly, but q equals three multiplied by the same distribution. This setting refers to the typical case of increased vehicle rent costs in the second stage of the problem (operational phase). The summary of the fleet parameters can be found in Table 5.3.

Table 5.3: Experimental setting summary.

Vehicle type	Cargo bike	E-van	Van
v_i	100	300	600
c_i	$v_i^{**}2\mathcal{U}(0.2, 1.2)$	$v_i^{**}2\mathcal{U}(0.6, 1.6)$	$v_i^{**}2\mathcal{U}(0.9, 1.9)$
c_k	$v_i^{**}3\mathcal{U}(0.2, 1.2)$	$v_i^{**}3\mathcal{U}(0.6, 1.6)$	$v_i^{**}3\mathcal{U}(0.9, 1.9)$

5.3.7 Experimental results of zone-based demand forecast with ZCAP

This subsection discusses the experimental results of zone-based demand forecast with ZCAP compared to the traditional VCSBPPSI model. The experimental setting for the 50 working days simulation is outlined in Subsection 5.3.6. We start with the graphical comparison of both approaches' performance on a daily basis. Then, we compare both approaches' obtained solution structure and averaged characteristics.

Firstly, we check the demand satisfaction criteria, as the most important in the tactical decision scheme. We compute the parameter Δ as the difference between the actual demand $D_z, z \in [1, N_{zones}]$ and the capacity of the chosen vehicles:

$$\Delta = \sum_{i \in \hat{\mathcal{I}}} c_i - \sum_{1}^{N_{zones}} D_z, \tag{5.15}$$

where $\hat{\mathcal{I}}$ is the subset of the vehicles chosen on both stages of the optimization problem. In addition, we compute the same parameter for the actually used vehicles in the provided dataset.

Figure 5.15 outlines the resulting Δ for VCSBPPSI, ZCAP, and actually used vehicles on a daily basis. in which we want to simulate the fleet management scenario, and on the y axis, we plot the Δ parameter in units corresponding to the capacity defined in Subsection 5.3.6. The reader may notice that the lowest Δ parameter values correspond to the ZCAP solutions, except in the first time step. Meanwhile, with the VCSBPPSI, we obtain a better logistics scheme in terms of capacity used compared to the vehicles involved in the given dataset. This is true except at one point in the day 12, where the optimization point suggests more capacity than used in the real data. It is related to the drastic drop in the demand in the day 12, which is not estimated with the introduced probability distribution of demand. Finally, it is worth mentioning that both approaches satisfy the next-day demand since there are no points with the negative Δ parameter. However, the ZCAP shows a considerable improvement in used capacity reduction, especially close to the end of the simulation (days 54 - 60). We relate this to the fact that the proposed zone-based forecasting algorithm with ZCAP has been adapted (trained) to real data, thus showing better accuracy with time.

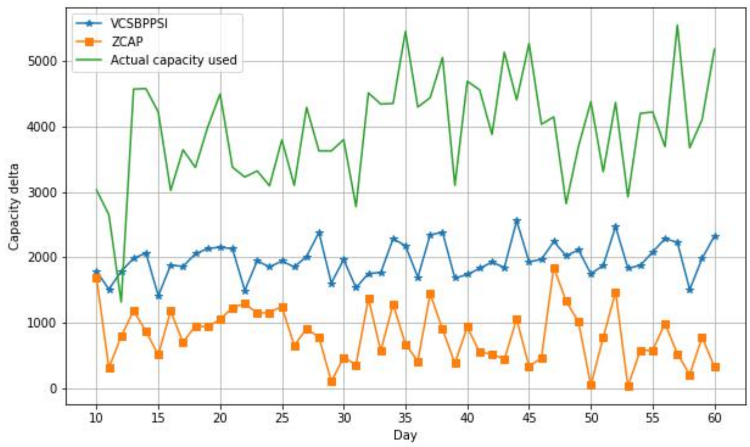


Figure 5.15: Resulting Δ for VCSBPPSI, ZCAP, and actually used vehicles on a daily basis

Figure 5.16 plots the daily cost reduction (in percentage, x-axis) of application VCSBPPSI and ZCAP problems, compared to the cost of the actual solution in the given data. On the y-axis is outlined the day number in the simulation. in almost every case compared with the VCSBPPSI application. We can observe a cost reduction of up to 64% compared to the used fleet capacity and, on average, around 20% cost improvements compared to the state-of-the-art approach. This improvement confirms the initial guess on the benefits of a data-driven approach compared to the traditional methods. Moreover, with the daily data enrichment, we expect more stable results on demand forecast and fleet capacity management in the following.

Concerning the obtained solution structure, in Table 5.4 we outline the decision values for each vehicle type: cargo bikes (CB), e-vans (EV), and light duty vans (LD). We separated each problem's first and second-stage decisions and averaged them over the 50 days working scenario. The reader may notice that in both approaches, the first-stage vehicles are more attractive in terms of cost and, thus, are frequently chosen to serve the demand regardless of the vehicle type. Additionally, we can observe a significant overall reduction in the total number of vehicles chosen with ZCAP. In fact, ZCAP provides a more flexible fleet management strategy due to the increased fraction of second-stage vehicles chosen. This fact corresponds to the lucky cases when the second-stage capacity better fits the demand regardless of the increased cost. Finally, we can observe that ZCAP is slightly better at utilizing the smaller vehicles overall, which provides a better cost reduction. Nevertheless, without loss of generality, ZCAP allows the inclusion of CO_2 related costs and constraints to each vehicle type to provide an even better balanced capacity

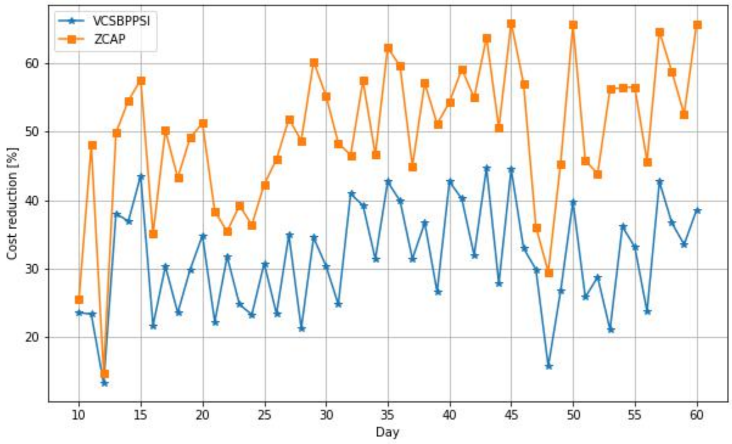


Figure 5.16: Total absolute error percentage of predicted demand volumes sum with different forecasters for different N_{zones} parameter.

distribution.

Table 5.4: Solution structure of VCSBPPSI and ZCAP.

Vehicle type	CB	EV	LD
VCSBPPSI F.S.	0.977124	0.970588	0.993464
VCSBPPSI S.S.	0.01634	0.003268	0.019608
ZCAP F.S.	0.856209	0.811765	0.760784
ZCAP S.S.	0.054902	0.004575	0.001307

Finally, in Table 5.5, we outline the main performance metrics for each approach, averaged over the entire test period. They include the total capacity volume used in the solution, the averaged capacity Δ outlined above, the overall solution cost, the percentage of cost reduction compared to the really used fleet of vehicles, and a computing time of the optimization problem. The reader may notice the overall better performance of the ZCAP-based approach: we managed to reduce the used capacity by around 27.4%, provided a fleet with almost 60% less empty space without knowledge about demand, significantly reduced the involved costs and computing time of the problem. Nevertheless, to provide a fair comparison between algorithms, we outline the computing time of the optimization problem only, which is almost negligible for ZCAP (around 0.2 seconds on Intel i7-9750H CPU @ 2.60GHz), regardless of the data sampling, zoning, or prediction. Evidently, the entire zone-based forecasting algorithm outlined in Subsection 5.3.2 requires much more computational effort. However, we expect this issue will be mitigated with the logistics company working scenario due to the constantly increasing availability of more data. This, in turn, stabilizes involved parameters (as N_{zones}) and enables the usage of pre-trained forecasters, especially in the case of NN-based regression.

Table 5.5: Performance comparison between VCSBPPSI and ZCAP.

Approach	VCSBPPSI	ZCAP
Capacity used	4172.549	3027.843
Capacity Δ	1958.805	814.0987
Costs involved	515026.3	226461
Cost reduction [%]	31.54897	49.72781
Computing time [s]	288.275	0.202957

5.3.8 Industrial and methodological impact of zone-based demand forecast with ZCAP

All the forecasting methods performed considerably better in terms of RMSE with additional training data generated with the process described in this chapter. Finally, the results of the introduced pipeline application to the 50 working days of the logistics company simulation with preliminary knowledge of the past 10 days were compared to the VCSBPPSI application with the derived Chi-squared probability distribution for the uncertain items. The results showed a significant reduction in operational costs, used capacity, and empty space in the vehicle load compared to the traditional approach based on VCSBPPSI.

Furthermore, the city's division into different service zones mimics the widely adopted capacity optimization strategies in-field on the tactical level, reducing the gap between industry and operational research.   the potential reduction in half-empty carriers around the city. Furthermore, achieved safety in demand coverage raises the possibilities of ML applications in the logistic service providers‘ management software. Likewise, recently appearing online logistics platforms unifying capacity management and consolidation can adapt the proposed framework freely due to low computational time and a quick response time from the IDSP. Finally, the proposed approach relies on historical delivery data only, facilitating its industrial application to every business scheme.

such as e-van charging limits, bike travel distance from hub limits, etc. These changes could reduce the CO_2 around the city and lower the noise in the city centers. Furthermore, following the practitioners' experience, the time load of the drivers remains an emerging problem for the study of last-mile delivery problems, which has to be included in the derived framework in the future. Lastly, we are concerned about optimal city zoning with the ML classification to support logistics managers on the tactical and strategic decision levels and further research toward reducing the gap between the academy and industry.

Chapter 6

Conclusions

Involvement of the uncertain items in such problems enables its application to tactical capacity management scenarios. In particular, Variable Cost and Size Bin Packing Problem with Stochastic Items (VCSBPPSI) provides a fleet with various capacities while minimizing the cost in uncertain demand conditions. However, VCSBPPSI is characterized by a high computational complexity related to uncertainty treatment. Therefore, in this thesis, we introduced new Machine Learning (ML) based heuristics to obtain its solution for realistic problem instances in a reasonable time with a high number of orders and vehicles (up to 2000 orders).

We compare the proposed heuristic performance with the exact (sampling) approach and widely adopted Progressive Hedging approach. In addition, we provide a case study and of ML heuristics application to fleet management scenarios in the urban area of Turin, Italy. where the solution should be requested in an online manner. Moreover, the relatively straightforward application of the proposed heuristic enables its usage in other problems characterized by binary decision variables. engaged in the modern trends of consolidated logistics.

Firstly, we apply the deterministic VCSBPP to the obtained forecast setting to provide the required fleet capacity for the ”next-day” delivery. Then, we replace it with the introduced Zone Capacity Assignment Problem (ZCAP), which enables fleet capacity management under uncertain demand with existing city zoning and forecast approaches. Finally, we provide extensive computational experiments for both methods on the 60 working days scenario of a logistics company in Antwerp, Belgium. The performance comparison of the introduced methods with VCSBPPSI shows an improvement in daily demand volume coverage, computing time reduction, a significant reduction in costs, used capacity, and empty space in the vehicle load.

Bibliography

[1] Fatima Ezzahra Achamrah, Fouad Riane, and Sabine Limbourg. “Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning”. *International Journal of Production Research* 60.20 (Oct. 2021). DOI: 10.1080/00207543.2021.1987549.

[2] Mauro Maria Baldi, Teodor Gabriel Crainic, Guido Perboli, and Roberto Tadei. “Branch-and-price and beam search algorithms for the variable cost and size bin packing problem with optional items”. *Annals of Operations Research* 222 (2014).

[3] Mauro Maria Baldi, Teodor Gabriel Crainic, Guido Perboli, and Roberto Tadei. “The generalized bin packing problem”. *Transportation Research Part E: Logistics and Transportation Review* 48.6 (2012).

[4] Mauro Maria Baldi, Daniele Manerba, Guido Perboli, and Roberto Tadei. “A generalized bin packing problem for parcel delivery in last-mile logistics”. *European Journal of Operational Research* 274.3 (2019).

[5] Alessandro Baldo, Matteo Boffa, Lorenzo Cascioli, Edoardo Fadda, Chiara Lanza, and Arianna Ravera. “The Polynomial Robust Knapsack Problem”. *European Journal of Operational Research* (2022).

[6] Joris Beckers, Ivan Cardenas, and Ivan Sanchez-Diaz. “Managing household freight: The impact of online shopping on residential freight trips”. *Transport Policy* 125 (2022).

[7] Yoshua Bengio, Emma Frejinger, Andrea Lodi, Rahul Patel, and Sriram Sankaranarayanan. “A learning-based algorithm to quickly compute good primal solutions for Stochastic Integer Programs” (2020).

[8] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[9] Luce Brotcorne, Guido Perboli, Mariangela Rosano, and Qu Wei. “A Managerial Analysis of Urban Parcel Delivery: A Lean Business Approach”. *Sustainability* 11.12 (2019). DOI: 10.3390/su11123439.

[10] Michael Browne, Julian Allen, and Stephen Anderson. "Low emission zones: the likely effects on the freight transport sector". *International Journal of Logistics: Research and Applications* 8.4 (2005).

[11] Maria Elena Bruni, Edoardo Fadda, Stanislav Fedorov, and Guido Perboli. "A machine learning optimization approach for last-mile delivery and third-party logistics". *Computers & Operations Research* (2023).

[12] Leonardo Caggiani, Aleksandra Colovic, Luigi Pio Prencipe, and Michele Ottomanelli. "A green logistics solution for last-mile deliveries considering e-vans and e-cargo bikes". *Transportation Research Procedia* 52 (2021).

[13] Luca Cagliero, Luigi De Russis, Laura Farinetti, and Teodoro Montanaro. "Improving the Effectiveness of SQL Learning Practice: A Data-Driven Approach". 1 (2018). DOI: 10.1109/compsac.2018.00174.

[14] Vittorio Capocasale, Danilo Gotta, Stefano Musso, and Guido Perboli. "A Blockchain, 5G and IoT-based transaction management system for Smart Logistics: an Hyperledger framework". *2021 IEEE 45th Annual Computers, Software, and Applications* (2021). DOI: 10.1109/COMPSAC51774.2021.00179.

[15] Marco Casazza. "New formulations for variable cost and size bin packing problems with item fragmentation". *Optimization Letters* 13.2 (2019).

[16] P. Castrogiovanni, E. Fadda, G. Perboli, and A. Rizzo. "Smartphone Data Classification Technique for Detecting the Usage of Public or Private Transportation Modes". *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.2982218.

[17] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. *Learning to Optimize: A Primer and A Benchmark*. 2021. DOI: 10.48550/ARXIV.2103.12828.

[18] Graham Clarke, Christopher Thompson, and Mark Birkin. "The emerging geography of e-commerce in British retailing". *Regional Studies, Regional Science* 2.1 (2015).

[19] Fulvio Corno, Luigi De Russis, Andrea Marcelli, and Teodoro Montanaro. "An Unsupervised and Non-Invasive Model for Predicting Network Resource Demands". *IEEE Internet of Things Journal* PP (July 2018). DOI: 10.1109/JIOT.2018.2860681.

[20] Fulvio Corno, Luigi De Russis, and Teodoro Montanaro. "Estimate user meaningful places through low-energy mobile sensing" (2016).

[21] Isabel Correia, Luiés Gouveia, and Francisco Saldanha-da-Gama. "Solving the variable size bin packing problem with discretized formulations". *Computers & Operations Research* 35.6 (June 2008). DOI: 10.1016/j.cor.2006.10.014.

[22] Paulo da Costa, Jason Rhuggenaath, Yingqian Zhang, Alp Akcay, and Uzay Kaymak. "Learning 2-Opt Heuristics for Routing Problems via Deep Reinforcement Learning". *SN Computer Science* 2.5 (July 2021). DOI: 10.1007/s42979-021-00779-2.

[23] Teodor Gabriel Crainic, Jesus Gonzalez Feliu, Nicoletta Ricciardi, Frédéric Semet, and Tom Van Woensel. *10. Operations research for planning and managing city logistics systems*. Edward Elgar Publishing, 2023, p. 190.

[24] Teodor Gabriel Crainic, Franklin Djeumou Fomeni, and Walter Rei. "Multi-period bin packing model and effective constructive heuristics for corridor-based logistics capacity planning". *Computers & Operations Research* 132.105308 (2021).

[25] Teodor Gabriel Crainic, Bernard Gendron, and Mohammad Rahim Akhavan Kazemzadeh. "A taxonomy of multilayer network design and a survey of transportation and telecommunication applications". *European Journal of Operational Research* 303.1 (2022). ISSN: 0377-2217. DOI: 10.1016/j.ejor.2021.12.028.

[26] Teodor Gabriel Crainic, Luca Gobbato, Guido Perboli, and Walter Rei. "Logistics capacity planning: A stochastic bin packing formulation and a progressive hedging meta-heuristic". *European Journal of Operational Research* 253.2 (2016).

[27] Teodor Gabriel Crainic, Luca Gobbato, Guido Perboli, Walter Rei, Jean-Paul Watson, and David L Woodruff. "Bin packing problems with uncertainty on item characteristics: An application to capacity planning in logistics". *Procedia-Social and Behavioral Sciences* 111 (2014).

[28] Teodor Gabriel Crainic, Mike Hewitt, and Walter Rei. "Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design". *Computers & Operations Research* 43 (2014).

[29] Teodor Gabriel Crainic, Guido Perboli, Simona Mancini, and Roberto Tadei. "Two-echelon vehicle routing problem: a satellite location analysis". *Procedia-Social and Behavioral Sciences* 2.3 (2010).

[30] Teodor Gabriel Crainic, Guido Perboli, Walter Rei, and Roberto Tadei. "Efficient lower bounds and heuristics for the variable cost and size bin packing problem". *Computers & Operations Research* 38.11 (2011).

[31] Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. "Models for evaluating and planning city logistics systems". *Transportation science* 43.4 (2009).

[32] Crainic, Teodor Gabriel and Gendreau, Michel and Gendron, Bernard. "City Logistics". In: *Network Design with Applications in Transportation and Logistics*. Springer, Boston, 2021. Chap. 16, pp. 507–537.

[33] Shannon Cummins, James W Peltier, John A Schibrowsky, and Alexander Nill. “Consumer behavior in the online context”. *Journal of Research in Interactive Marketing* (2014).

[34] Alfredo Cuzzocrea, Carson K. Leung, Deyu Deng, Jiaxing Jason Mai, Fan Jiang, and Edoardo Fadda. “A Combined Deep-Learning and Transfer-Learning Approach for Supporting Social Influence Prediction”. *Procedia Computer Science* 177 (2020). DOI: 10.1016/j.procs.2020.10.025.

[35] Alberto De Marco, Giulio Mangano, Giovanni Zenezini, Anna Corinna Cagliano, Guido Perboli, Mariangela Rosano, and Stefano Musso. “Business Modeling of a City Logistics ICT Platform”. 2 (2017).

[36] Mauro Dell’Amico, Fabio Furini, and Manuel Iori. “A branch-and-price algorithm for the temporal bin packing problem”. *Computers & operations research* 114 (2020).

[37] Antonio Diglio, Juanjo Peiró, Carmela Piccolo, and Francisco Saldanha-da-Gama. “Approximation schemes for districting problems with probabilistic constraints”. *European Journal of Operational Research* 307.1 (2023).

[38] Raphaëlle Ducret, Bernard Lemarié, and Alain Roset. “Cluster analysis and spatial modeling for urban freight. Identifying homogeneous urban zones based on urban form and logistics characteristics”. *Transportation Research Procedia* 12 (2016).

[39] Justin Dumouchelle, Rahul Patel, Elias B. Khalil, and Merve Bodur. *Neur2SP: Neural Two-Stage Stochastic Programming*. 2022. DOI: 10.48550/ARXIV.2205.12006.

[40] Edoardo Fadda, Stanislav Fedorov, Guido Perboli, and Ivan Dario Cardenas Barbosa. “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. 2021, pp. 1291–1296.

[41] Edoardo Fadda, Dario Mana, Guido Perboli, and Roberto Tadei. “Multi Period Assignment Problem for Social Engagement and Opportunistic IoT”. In: *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 2. 2017, pp. 760–765. DOI: 10.1109/COMPSAC.2017.173.

[42] Edoardo Fadda, Guido Perboli, and Roberto Tadei. “A progressive hedging method for the optimization of social engagement and opportunistic IoT problems”. *European Journal of Operational Research* 277.2 (2019).

[43] Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj El Moussami, and Abdeslam Lachhab. “Forecasting of demand using ARIMA model”. *International Journal of Engineering Business Management* 10 (2018).

[44] Héctor J Fraire-Huacuja, Alejandro Estrada Padilla, Laura Cruz-Reyes, Claudia Gómez-Santillán, Nelson Rangel-Valdez, Marıéa Lucila Morales-Rodrıéguez, and Juan Frausto. "Variable neighborhood search algorithm for the variable cost and size bin packing problem". In: *Handbook of Research on Metaheuristics for Order Picking Optimization in Warehouses to Smart Cities*. IGI Global, 2019, pp. 1–17.

[45] Matheus Freitas, João Marcos Pereira Silva, and Eduardo Uchoa. "A unified exact approach for Clustered and Generalized Vehicle Routing Problems". *Computers & Operations Research* 149 (2023). DOI: 10.1016/j.cor.2022.106040.

[46] Andrii Galkin, Tibor Schlosser, Ivan Cardenas, Dominika Hodakova, and Silvia Capayova. "Freight Demand and Supply Assessment for Implementation of Crowdsourcing Technology: A Case Study in Bratislava, Slovakia". *Sustainability* 13.7 (2021).

[47] Michael R Garey and David S Johnson. *Computers and intractability*. Vol. 174. freeman San Francisco, 1979.

[48] Nadia Giuffrida, Jenny Fajardo-Calderin, Antonio D Masegosa, Frank Werner, Margarete Steudter, and Francesco Pilla. "Optimization and machine learning applied to last-mile logistics: A review". *Sustainability* 14.9 (2022).

[49] Riccardo Giusti, Chiara Iorfida, Yuanyuan Li, Daniele Manerba, Stefano Musso, Guido Perboli, Roberto Tadei, and Shuai Yuan. "Sustainable and De-Stressed International Supply-Chains Through the SYNCHRO-NET Approach". *Sustainability* 11.4 (2019). DOI: 10.3390/su11041083.

[50] Shengyu Gu, Beata Ślusarczyk, Sevda Hajizada, Irina Kovalyova, and Amina Sakhbieva. "Impact of the covid-19 pandemic on online consumer purchasing behavior". *Journal of Theoretical and Applied Electronic Commerce Research* 16.6 (2021).

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

[52] Jorge Herrera-Franklin, Alejandro Rosete, and Milton Garcıéa-Borroto. "A fuzzy approach for the variable cost and size bin packing problem allowing incomplete packing". *Inteligencia Artificial* 24.67 (2021).

[53] Jakob Huber, Alexander Gossmann, and Heiner Stuckenschmidt. "Cluster-based hierarchical demand forecasting for perishable goods". *Expert systems with applications* 76 (2017).

[54] Elodie Huré, Karine Picot-Coupey, and Claire-Lise Ackermann. "Understanding omni-channel shopping value: A mixed-method study". *Journal of retailing and consumer services* 39 (2017).

[55] Jochen Janssens, Joos Van den Bergh, Kenneth Sörensen, and Dirk Cattrysse. "Multi-objective microzone-based vehicle routing for courier companies: From tactical to operational planning". *European Journal of Operational Research* 242.1 (2015).

[56] Asunción Jiménez-Cordero, Juan Miguel Morales, and Salvador Pineda. "Warm-starting constraint generation for mixed-integer optimization: A Machine Learning approach". *Knowledge-Based Systems* 253 (Oct. 2022). DOI: 10.1016/j.knosys.2022.109570.

[57] Petra Jıélková and Petra Králová. "Digital consumer behaviour and ecommerce trends during the COVID-19 crisis". *International Advances in Economic Research* 27.1 (2021).

[58] Jangha Kang and Sungsoo Park. "Algorithms for the variable sized bin packing problem". *European Journal of Operational Research* 147.2 (2003).

[59] David J. Ketchen and Christopher L. Shook. "The application of cluster analysis in strategic management research: an analysis and critique". *Strategic management journal* 17.6 (1996).

[60] Bernhard Korte and Jens Vygen. *Combinatorial Optimization*. Springer, 2005.

[61] Su-Young Kwak, Woo-Sung Cho, Gil-Am Seok, and Seung-Gyun Yoo. "Intention to Use Sustainable Green Logistics Platforms". *Sustainability* 12.8 (2020).

[62] Eric Larsen, Emma Frejinger, Bernard Gendron, and Andrea Lodi. "Fast Continuous and Integer L-shaped Heuristics Through Supervised Learning". *arXiv preprint arXiv:2205.00897* (2022).

[63] Eric Larsen, Sébastien Lachapelle, Yoshua Bengio, Emma Frejinger, Simon Lacoste-Julien, and Andrea Lodi. "Predicting tactical solutions to operational planning problems under imperfect information". *arXiv preprint arXiv:1807.11876* (2018).

[64] Maobin Li, Shouwen Ji, and Gang Liu. "Forecasting of Chinese E-commerce sales: an empirical comparison of ARIMA, nonlinear autoregressive neural network, and a combined ARIMA-NARNN model". *Mathematical Problems in Engineering* 2018 (2018).

[65] Yuanyuan Li, Stefano Carabelli, Edoardo Fadda, Daniele Manerba, Roberto Tadei, and Olivier Terzo. "Machine learning and optimization for production rescheduling in Industry 4.0". *The International Journal of Advanced Manufacturing Technology* (2020). DOI: 10.1007/s00170-020-05850-5.

[66] Yuchen Li, Jianghua Zhang, and Guodong Yu. "A scenario-based hybrid robust and stochastic approach for joint planning of relief logistics and casualty distribution considering secondary disasters". *Transportation Research Part E: Logistics and Transportation Review* 141 (2020). DOI: 10.1016/j.tre.2020.102029.

[67] Yao-San Lin, Yaofeng Zhang, I-Ching Lin, and Che-Jung Chang. "Predicting logistics delivery demand with deep neural networks". In: *2018 7th International Conference on Industrial Technology and Management (ICITM)*. IEEE. 2018, pp. 294–297.

[68] Andrea Lodi and Giulia Zarpellon. "On learning and branching: a survey". *TOP* 25.2 (June 2017). DOI: 10.1007/s11750-017-0451-6.

[69] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions" (2017).

[70] Scott M Lundberg et al. "Explainable AI for trees: From local explanations to global understanding". *arXiv preprint arXiv:1905.04610* (2019).

[71] Francesca Maggioni, Guido Perboli, and Roberto Tadei. "The multi-path traveling salesman problem with stochastic travel costs: Building realistic instances for city logistics applications". *Transportation Research Procedia* 3 (2014).

[72] Francesca Maggioni and Stein W. Wallace. "Analyzing the quality of the expected value solution in stochastic programming". *Annals of Operations Research* 200.1 (Nov. 2010). DOI: 10.1007/s10479-010-0807-x.

[73] Ada Malagnino, Teodoro Montanaro, Mariangela Lazoi, Ilaria Sergi, Angelo Corallo, and Luigi Patrono. "Building Information Modeling and Internet of Things integration for smart and sustainable environments: A review". *Journal of Cleaner Production* 312 (2021). DOI: 10.1016/j.jclepro.2021.127716.

[74] Silvano Martello and Paolo Toth. "Bin-packing problem". *Knapsack problems: Algorithms and computer implementations* (1990).

[75] Umberto Junior Mele, Luca Maria Gambardella, and Roberto Montemanni. "Machine Learning Approaches for the Traveling Salesman Problem: A Survey" (Jan. 2021). DOI: 10.1145/3463858.3463869.

[76] Shoma Miki and Hiroyuki Ebara. "Solving Traveling Salesman Problem with Image-Based Classification" (Nov. 2019). DOI: 10.1109/ictai.2019.00156.

[77] Sadegh Mirshekarian and Dusan Sormaz. "Machine learning approaches to learning heuristics for combinatorial optimization problems". *Procedia Manufacturing* 17 (2018).

[78] Roberto Montemanni, Federico D'ignazio, Xiaochen Chou, and Luca Maria Gambardella. "Machine Learning and Monte Carlo Sampling for the Probabilistic Orienteering Problem" (Dec. 2018). DOI: 10.1109/scis-isis.2018.00014.

[79] Geoffrey A. Moore. *Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers, 3rd edition*. Collins Business Essentials, 2014.

[80] Municipality of Turin. *Torino Living Lab*. Last accessed, 20/05/2023. 2018. URL: http://torinolivinglab.it/en/.

[81] Paul W. Murray, Bruno Agard, and Marco A. Barajas. "Forecasting supply chain demand by clustering customers". *IFAC-PapersOnLine* 48.3 (2015).

[82] Vinod Nair, Dj Dvijotham, Iain Dunning, and Oriol Vinyals. "Learning Fast Optimizers for Contextual Stochastic Integer Programs" (2018). Ed. by Amir Globerson and Ricardo Silva.

[83] Chao Ning and Fengqi You. "Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming". *Computers & Chemical Engineering* 125 (2019).

[84] Leise Kelli de Oliveira, Renata Lúcia Magalhães de Oliveira, Luisa Tavares Muzzi de Sousa, Ian de Paula Caliari, and Carla de Oliveira Leite Nascimento. "Analysis of accessibility from collection and delivery points: Towards the sustainability of the e-commerce delivery". *urbe. Revista Brasileira de Gestão Urbana* 11 (2019).

[85] A. Paszke et al. "PyTorch: An imperative style, high-performance deep learning library". *Advances in Neural Information Processing Systems* 32 (2019).

[86] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research* 12 (2011).

[87] Guido Perboli, Luce Brotcorne, Maria Elena Bruni, and Mariangela Rosano. "A new model for Last-Mile Delivery and Satellite Depots management: The impact of the on-demand economy". *Transportation Research Part E: Logistics and Transportation Review* 145 (2021).

[88] Guido Perboli, Stanislav Fedorov, and Mariangela Rosano. "The European Concept of Smart City: A Taxonomic Analysis". In: *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. 2020, pp. 1725–1730. DOI: 10.1109/COMPSAC48688.2020.000-6.

[89] Guido Perboli, Luca Gobbato, and Francesca Maggioni. "A progressive hedging method for the multi-path travelling salesman problem with stochastic travel times". *IMA Journal of Management Mathematics* 28.1 (2017).

[90] Guido Perboli, Luca Gobbato, and Francesca Perfetti. "Packing problems in transportation and supply chain: new problems and trends". *Procedia-Social and Behavioral Sciences* 111 (2014).

[91] Guido Perboli, Stefano Musso, and Mariangela Rosano. "Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases". *IEEE Access* 6 (2018). ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2875782.

[92] Guido Perboli, Stefano Musso, Mariangela Rosano, Roberto Tadei, and Moritz Godel. "Synchro-modality and slow steaming: New business perspectives in freight transportation". *Sustainability* 9.10 (2017).

[93] Guido Perboli and Mariangela Rosano. "Parcel delivery in urban areas: Opportunities and threats for the mix of traditional and green business models". *Transportation Research Part C: Emerging Technologies* 99 (2019). DOI: 10.1016/j.trc.2019.01.006.

[94] Guido Perboli, Mariangela Rosano, Michael SaintGuillain, and Pietro Rizzo. "Simulation–optimisation framework for City Logistics: an application on multimodal last-mile delivery". *IET Intelligent Transport Systems* 12.4 (2018).

[95] Guido Perboli, Mariangela Rosano, and Qu Wei. "A Simulation-Optimization Approach for the Management of the On-Demand Parcel Delivery in Sharing Economy". *IEEE Transactions on Intelligent Transportation Systems* (2021).

[96] Guido Perboli, Roberto Tadei, and Mauro M Baldi. "The stochastic generalized bin packing problem". *Discrete Applied Mathematics* 160.7-8 (2012).

[97] Guido Perboli, Roberto Tadei, and Luca Gobbato. "The multi-handler knapsack problem under uncertainty". *European journal of operational research* 236.3 (2014).

[98] David Pisinger. "An exact algorithm for large multiple knapsack problems". *European Journal of Operational Research* 114.3 (May 1999). DOI: 10.1016/s0377-2217(98)00120-9.

[99] Carl Edward Rasmussen and Hannes Nickisch. "The gpml toolbox version 4.0". *Technical Documentation* (2016).

[100] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN: 026218253X.

[101] R Tyrrell Rockafellar and Roger J-B Wets. "Scenarios and policy aggregation in optimization under uncertainty". *Mathematics of operations research* 16.1 (1991).

[102] Michael Saint-Guillain, Yves Deville, and Christine Solnon. "A multistage stochastic programming approach to the dynamic and stochastic VRPTW". In: *Integration of AI and OR Techniques in Constraint Programming: 12th International Conference, CPAIOR 2015, Barcelona, Spain, May 18-22, 2015, Proceedings 12*. Springer. 2015, pp. 357–374.

[103] Fatemeh Sarayloo, Teodor Gabriel Crainic, and Walter Rei. *An Integrated Learning and Progressive Hedging Matheuristic for Stochastic Network Design Problem*. CIRRELT, 2020.

[104] LV Savchenko and VV Davydenko. "Models of zoning of urban territory for rational delivery in the microconsolidation system". *Intellectualization of logistics and Supply Chain Management* 3 (2020).

[105] Serhii Frantsovych Smerichevskyi and LV Savchenko. *Clusterization of urban territory for building an effective delivery sustem*. 2020.

[106] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. "Information-theoretic regret bounds for gaussian process optimization in the bandit setting". *IEEE Transactions on Information Theory* 58.5 (2012).

[107] Gita Taherkhani, Ioana C Bilegan, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. "Tactical capacity planning in an integrated multi-stakeholder freight transportation system". *Omega* 110 (2022).

[108] El-Ghazali Talbi. "Machine learning into metaheuristics: A survey and taxonomy of data-driven metaheuristics". working paper or preprint. June 2020. URL: https://hal.inria.fr/hal-02745295.

[109] Jeffrey D. Ullman. "The performance of a memory allocation algorithm., Princeton University, Department of Electrical Engineering". *Computer Science Laboratory* 47 (1971).

[110] Thibaut Vidal, Maria Battarra, Anand Subramanian, and Güneş Erdog˘an. "Hybrid metaheuristics for the Clustered Vehicle Routing Problem". *Computers & Operations Research* 58 (2015). ISSN: 0305-0548. DOI: 10.1016/j.cor.2014.10.019.

[111] Web Site of 5T. *5T srl*. Last accessed, 20/05/2023. 2019. URL: http://www.cars.polito.it/.

[112] Li'ang Zhang and Suyun Geng. "The complexity of the 0/1 multi-knapsack problem". *Journal of Computer Science and Technology* 1.1 (Mar. 1986). DOI: 10.1007/bf02943300.

[113] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. "LSTM network: a deep learning approach for short-term traffic forecast". *IET Intelligent Transport Systems* 11.2 (2017).

[114] Xuesong Zhou and Hani S Mahmassani. "A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework". *Transportation Research Part B: Methodological* 41.8 (2007).

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.