POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Generalized Bin Packing Problem for parcel delivery in last-mile logistics

Original
A Generalized Bin Packing Problem for parcel delivery in last-mile logistics / Baldi, M.M., Manerba, D., Perboli, G., Tadei, R.. - In: EUROPEAN JOURNAL OF OPERATIONAL RESEARCH. - ISSN 0377-2217. - 274:3(2019), pp. 990-999. [10.1016/j.ejor.2018.10.056]

Availability:
This version is available at: 11583/2721849 since: 2019-01-11T14:45:33Z

Publisher:
Elsevier B.V.

Published
DOI:10.1016/j.ejor.2018.10.056

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

14 November 2026

European Journal of Operational Research 274 (2019) 990–999

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Production, Manufacturing, Transportation and Logistics

A Generalized Bin Packing Problem for parcel delivery in last-mile logistics

Mauro Maria Baldi[a], Daniele Manerba[a], Guido Perboli[b,c], Roberto Tadei[a,*]

[a] DAUIN, Politecnico di Torino, Turin, Italy
[b] ICT for City Logistics and Enterprises Lab, Politecnico di Torino, Turin, Italy
[c] CIRRELT, Montreal, Canada

ARTICLE INFO

Article history:
Received 13 November 2017
Accepted 30 October 2018
Available online 3 November 2018

Keywords:
Logistics
Generalized Bin Packing Problem
Parcel delivery
Last-mile logistics

ABSTRACT

In this paper, we present a new problem arising at a tactical level of setting a last-mile parcel delivery service in a city by considering different Transportation Companies (TC), which differ in cost and service quality. The courier must decide which TCs to select for the service in order to minimize the total cost and maximize the total service quality. We show that the problem can be modeled as a new packing problem, the Generalized Bin Packing Problem with bin-dependent item profits (GBPPI), where the items are the parcels to deliver and the bins are the TCs. The aim of the GBPPI is to select the appropriate fleet from TCs and determine the optimal assignment of parcels to vehicles such that the overall net cost is minimized. This cost takes into account both transportation costs and service quality. We provide a Mixed Integer Programming formulation of the problem, which is the starting point for the development of efficient heuristics that can address the GBPPI for instances involving up to 1000 items. Extensive computational tests show the accuracy of the proposed methods. Finally, we present a last-mile logistics case study of an international courier which addresses this problem.

© 2018 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

The transportation services market is estimated to be worth approximately 3 trillion euros worldwide with a gross value added (GVA) of 600 billion in the EU-28 at basic prices, corresponding approximately to 5% of the total GVA (European Commission, 2017). In the past decade, new challenges emerged thanks to an increased awareness of stakeholders and companies to a more general vision of the transpor tation sustainability taking into account economic, environmental and social aspects. In particular, last-mile delivery raised as one of the more complex, challenging, and innovating topic. In more detail, the explosion of e-commerce and the need of a global vision of the sustainability of the last-mile brought researchers and practitioners to define new business and operational models. These models must fulfill the increasing demand and the high standards in terms of quality of service required by e-commerce and traditional companies and flexibility asked by end users. In this paper, we consider a crucial aspect in the management of last-mile logistics operations, i.e., the planning of the fleet used in a given urban area to deliver the parcels. More precisely, we address the tactical problem of setting a last-mile parcel delivery service in a city by considering different transportation companies, which differ in cost and service quality. We also show how this problem can be modeled as a packing problem, namely the Generalized Bin Packing Problem with bin-dependent item profits (GBPPI). Packing problems deal with the assignment of items to bins. In the tactical problem presented here, the item model the parcels while the bins represent the vehicles of the TCs available for the deliveries.

The contribution of this paper is twofold. The first contribution refers to the packing problems literature. In details, the introduction of the bin-dependent item profits provides a more flexible packing problem (GBPPI) that enables to tackle the majority of real-world performance indicators and to address problems with mixed objective functions. The mixed objective function consists of two terms with opposite signs: (1) the total cost of the bins used to be minimized; and (2) the total bin-dependent profit of the selected items to be maximized. Although the introduction of bin-dependent item profits might seem an irrelevant development of the original Generalized Bin Packing Problem (GBPP), we show that this modification in the problem setting revolutionizes the methodologies. It is known from the literature that small changes

* Corresponding author.
E-mail addresses: mauro.baldi@polito.it (M.M. Baldi), daniele.manerba@polito.it (D. Manerba), guido.perboli@polito.it (G. Perboli), roberto.tadei@polito.it (R. Tadei).

https://doi.org/10.1016/j.ejor.2018.10.056
0377-2217/© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license. (http://creativecommons.org/licenses/by/4.0/)

in the objective function do not alter problem complexity (Tadei, Perboli, & Perfetti, 2014). However, we show that this is not the case for the GBPPI. This unexpected behavior can be noticed in at least two circumstances. First, as we show in Section 5.3, if a commercial solver is used to address both the GBPP and the GBPPI, the percentage gap significantly increases with the same computational time on instances with the same number and typology of items and bins. Second, all the more so, classical heuristic approaches of packing problems, such as the Best Fit and Next Fit decreasing procedures, fail when the profit becomes bin-dependent. To overcome these issues, after providing an integer-programming we propose efficient methodologies which take bin dependency of profits into account.

The second contribution of the paper is to use the GBPPI to address a real case study arising in last-mile logistics which involves an international courier. Every day, the courier has to serve its customers. Even if the customers can both ask for pick-ups and deliveries, we can relax this scenario. In fact 85% of the operations of a courier are deliveries (Perboli, Rosano, Saint-Guillain, & Rizzo, 2018). Therefore, we can just take into account the effects of deliveries.

In order to perform the service, the courier uses a set of transportation companies (TCs) which fulfill parcel deliveries using their own fleet. The courier must decide each day which TCs to select for the service in order to minimize the total cost and maximize the total service quality. That decision must be taken in a short time (approximately 10 minutes after all parcels to deliver become known) to ensure a prompt assignment of parcels to vehicles and, therefore, an efficient service. Minimizing the total cost and maximizing the total service quality are conflictual objectives. Existing models for last-mile logistics only partially consider conflictual objectives while addressing the selection of different TCs (Perboli, Gobbato, & Perfetti, 2014b; Perboli, Tadei, & Gobbato, 2014c). The introduction of bin-dependent item profits allows the use of the GBPPI in last-mile logistics (Crainic et al., 2014; Tadei, Perboli, & Baldi, 2012).

The remainder of this paper is organized as follows: in Section 2, we introduce the problem and review important literature on packing problems on which the GBPPI is based. In Section 3, the GBPPI model is described, and in Section 4 heuristics for addressing the GBPPI are introduced. Instance sets and computational results are given in Section 5. In more detail, we compare the GBPPI with the classical GBPP, showing how the introduction of the bin dependency of the profits makes our problem much difficult to solve. Then, we show the accuracy and the effectiveness of our metaheuristics and finally we show how using the GBPPI to solve the problem faced by an international courier of choosing the mix of delivery options/sub-contractors can bring to a reduction of its operational costs, with a clear benefit for the company. Conclusions are provided in Section 6.

2. Problem setting and literature review

The operations concerning the very last leg of the supply chain, the so-called *last-mile logistics*, have emerged as one of the most problematic ones to manage, optimize, actuate, and control. These operations, in fact, face significant fulfillment constraints, higher social, environmental and economic costs, and the complexity to maintain their economies of scale and expected service levels. In the recent years, following the enormous increase of the e-commerce and the relative growth of parcel deliveries (and returns) in our cities, last-mile challenges have become more and more complex, and many researchers have focused on finding solutions at various levels.

The most important innovations in this area generally share the vision of reducing as much as possible the negative externalities while maintaining the process sustainability in terms of costs and quality. For example, new collaborative business models exploiting crowd-tasking (Wang, Zhang, Liu, Shen, & Lee, 2016) or synchro-modality (Giusti, Manerba, Perboli, Tadei, & Yuan, 2018; de Souza, Goh, Lau, Ng, & Tan, 2014) have appeared, accompanied by a steadily increasing use of self-service technologies, such as parcel lockers, or greener solutions, such as electric vehicles and bicycles (Ranieri, Digiesi, Silvestri, & Roccotelli, 2018; Vakulenko, Hellström, & Hjort, 2018). Moreover, it has also been shown that a more appropriate modeling of the customers' behavior and more adequate pricing schemes for the service time-constraints may help in de-stressing the so-called Attended Home Deliveries, in which the parcel must be delivered within a precise time window in order to find the customer at home (Manerba, Mansini, & Zanotti, 2018; Ülkü & Bookbinder, 2012).

While at an operational level the optimization of last-mile logistics reduces to deal with vehicle routing and scheduling problems (generally complicated by the presence of multiple depots, products, and distribution echelons or by hard loading constraints (Boysen, Schwerdfeger, & Weidinger, 2018; Manerba, Mansini, & Riera-Ledesma, 2017; Zhou, Baldacci, Vigo, & Wang, 2018), the tactical planning is instead fundamental to evaluate the options (e.g., the various transportation tenders) and allocate the right resources (vehicles, operators, products) to the right place (depots, facilities, distribution centers) for the day-by-day operations (Aljohani & Thompson, 2018; Holzapfel, Kuhn, & Sternbeck, 2018). In the last decade, there have been interesting developments in the packing literature so that, in addition to their classical uses at the operational level, packing problems have gradually appeared as tools for modeling strategic and tactical decisions taken in the transportation and supply chain sectors (Crainic et al., 2014; Perboli et al., 2014b). This has led, for instance, to new problems, such as the Variable Size and Cost Bin Packing Problem (Crainic, Perboli, Rei, & Tadei, 2011), the Multi-handler Knapsack Problem under Uncertainty (Perboli et al., 2014c), and the Generalized Bin Packing Problem (GBPP) (Baldi, Crainic, Perboli, & Tadei, 2012). The problem studied in this paper is indeed a generalization of the GBPP and it is used to provide decision support at a tactical level for last-mile logistics.

Packing problems look for an optimal assignment of items to a set of bins able to accommodate them. In the GBPPI, bins are characterized by their capacity and cost of use, and are classified by type, i.e., bins of the same bin type have the same capacity and cost. The items can be compulsory (i.e., mandatory to load) or non-compulsory, and are characterized by weight and profit. The item profits depend on the bin types. The aim of the GBPPI is to load compulsory items and profitable non-compulsory items into appropriate bins in order to minimize net cost. This is given by the difference between the total cost of the bins used and the total profit accruing from the loading of the items. As already discussed in Section 1, the GBPPI naturally arises in last-mile-logistics optimization, where a courier responsible for parcel delivery faces a tactical problem involving the following decisions: (1) select a number of TCs, (2) select a number of vehicles from the fleet of the chosen TCs, and (3) assign parcels to the chosen vehicles. In the GBPPI, the TCs are modeled by the bin types and vehicles by the bins. This implies that cost and capacity for bins of the same type represent the transportation cost and the maximum weight for each vehicle of the fleet of a particular TC. Moreover, each TC has a limited fleet with a maximum number of available vehicles. Parcels are represented by items. Therefore the weight of an item models the weight of a parcel, while its profit takes into account the economic value for the courier due to delivery and the gain due to the service quality of TCs. The courier knows service quality based on feedback from consignees. This feedback takes into account a number of factors including punctuality, integrity

of parcels at the delivery, and courtesy. Compulsory items represent priority parcels which have to be consigned in the current delivery, while non-compulsory items are those parcels which delivery can be procrastinated. Finally, the courier might cope with municipality traffic limitations on the maximum number of circulating vehicles.

The GBPPI is an evolution of bin packing problems which we briefly recall here. The oldest problem is the Bin Packing Problem (BPP), which consists of a set of items to be loaded into bins of equal size such that the number of bins used is minimum (Crainic, Perboli, Pezzuto, & Tadei, 2007a; 2007b; Martello & Toth, 1990). The Variable Sized Bin Packing Problem (VSBPP) is a generalization of the BPP, which was proposed in the 1980s by Friesen and Langston (1986), and involves the introduction of bin types. Monaci (2002) and Haouari and Serairi (2011) studied exact algorithms, whereas heuristic approaches were adopted in Haouari and Serairi (2009), Hemmelmayr, Schmid, and Blum (2012), and Maiza, Labed, and Radjef (2013). A more interesting variant where there is no correlation between the volume and the cost of the bins is discussed in Crainic et al. (2011) for the deterministic form and in Crainic et al. (2014) for the stochastic one.

The Generalized Bin Packing Problem (GBPP) is a generalization of previous bin packing problems, involving the introduction of item profits as well as compulsory and non-compulsory items. Heuristics, and exact and approximate algorithms can be found in Baldi et al. (2012) and Baldi, Crainic, Perboli, and Tadei (2014). Online and stochastic variants were discussed in Baldi, Crainic, Perboli, and Tadei (2013) and Perboli, Tadei, and Baldi (2012). Finally, approximation issues were studied in Baldi and Bruglieri (2016).

The analysis of the literature shows how all problems studied have costs and profits associated to just one of the two sets: usually costs to bins and profits to items. Actually, to the best of our knowledge, no study considers a dependency of the profits from both items and bins. This feature prevents the use of classical heuristics, namely the Best Fit and Next Fit Decreasing heuristics, to address the GBPPI.

3. The GBPPI model

In this section, we propose a model for the GBPPI. We define the following:

- $\mathcal{I}$: set of items
- $n = |\mathcal{I}|$: number of items
- $\mathcal{I}^{C} \subseteq \mathcal{I}$: set of compulsory items
- $\mathcal{I}^{NC} \subseteq \mathcal{I}$: set of non-compulsory items. Clearly, $\mathcal{I}^{C}$ and $\mathcal{I}^{NC}$ are a partition of set I, i.e., $\mathcal{I}^{C} \cup \mathcal{I}^{NC} = \mathcal{I}$ and $\mathcal{I}^{C} \cap \mathcal{I}^{NC} = \emptyset$
- $\mathcal{J}$: set of bins
- $m = |\mathcal{J}|$: number of bins
- $\mathcal{T}$: set of bin types
- $\sigma : \mathcal{J} \longrightarrow \mathcal{T}$: indicator function, where given bin $j \in \mathcal{J}$, reveals its type $t \in \mathcal{T}$, i.e., $\sigma(j) = t$ iff bin $j \in \mathcal{J}$ belongs to type $t \in \mathcal{T}$
- p_{it}: profit generated by item $i \in \mathcal{I}$ when accommodated into a bin of type $t \in \mathcal{T}$
- w_i: volume of item $i \in \mathcal{I}$
- C_t: cost of a bin of type $t \in \mathcal{T}$
- W_t: capacity of a bin of type $t \in \mathcal{T}$
- L_t: minimum number of bins to be used of type $t \in \mathcal{T}$
- U_t: maximum number of bins to be used of type $t \in \mathcal{T}$
- $U \leq \sum_{t \in \mathcal{T}} U_t$: maximum number of bins to be used
- $\mathcal{S} \subseteq \mathcal{J}$: set of bins used in a solution of the GBPPI
- $W_{res}(b) : b \in \mathcal{S}$: residual volume of a bin for a solution of the GBPPI. This is given by the capacity of bin b minus the sum of the volumes of the items loaded in b.

An optimal solution of an instance of GBPPI must satisfy the following requirements:

- The overall cost given by the difference between the cost of the bins used and the profits incurred by the loaded items is minimized.
- All compulsory items must be accommodated into some bins.
- The sum of the volumes of the items loaded into a bin cannot exceed the capacity of that bin.

In order to provide a model for the GBPPI, we need to introduce the following binary variables:

$$x_{ij} = \begin{cases} 1 & \text{if item } i \in \mathcal{I} \text{ is accommodated into bin } j \in \mathcal{J} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$y_j = \begin{cases} 1 & \text{if bin } j \in \mathcal{J} \text{ is used} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

A model for the GBPPI can then be formulated as follows:

$$\min \quad \sum_{j \in \mathcal{J}} C_{\sigma(j)} y_j - \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}} p_{i\sigma(j)} x_{ij} \tag{3}$$

$$\text{s. t.} \quad \sum_{i \in \mathcal{I}} w_i x_{ij} \leq W_{\sigma(j)} y_j \qquad j \in \mathcal{J} \tag{4}$$

$$\sum_{j \in \mathcal{J}} x_{ij} = 1 \qquad i \in \mathcal{I}^{C} \tag{5}$$

$$\sum_{j \in \mathcal{J}} x_{ij} \leq 1 \qquad i \in \mathcal{I}^{NC} \tag{6}$$

$$\sum_{j \in \mathcal{J}: \sigma(j) = t} y_j \leq U_t \qquad t \in \mathcal{T} \tag{7}$$

$$\sum_{j \in \mathcal{J}: \sigma(j) = t} y_j \geq L_t \qquad t \in \mathcal{T} \tag{8}$$

$$\sum_{j \in \mathcal{J}} y_j \leq U \tag{9}$$

$$y_j \in \{0, 1\} \qquad j \in \mathcal{J} \tag{10}$$

$$x_{ij} \in \{0, 1\} \qquad i \in \mathcal{I}, \quad j \in \mathcal{J} \tag{11}$$

The objective function (3) ensures that the solution minimizes the total net cost, given by the cost due to the bin used minus the profit obtained from the loading of the items into the bins. Constraints (4) are the so-called capacity constraints that ensure that the sum of the volumes of the items loaded into a bin does not exceed the bin capacity. Constraints (5) ensure that all compulsory items are loaded, whereas constraints (6) state that non-compulsory items may or may not be accommodated. Constraints (7)–(9) are bin usage constraints. Finally, constraints (10)–(11) force the variables involved to be binary.

We point out that the presence of bin usage constraints (7)–(9) and a limited number m of bins might lead to infeasible solutions. As shown in Baldi et al. (2012), in order to ensure that the problem is feasible, a dummy bin with a large capacity and high cost is added to the problem.

Model (3)–(11) inherits all the variables and constraints from the Emodel for the GBPP described in Baldi et al. (2012) except for the objective function. Although the change in the objective function due to the introduction of bin-dependent item profits might seem trivial at first glance, in Section 5.3 we present a detailed comparison between the GBPP and the GBPPI and show that the latter is harder to solve. Moreover, as we show in Section 4, the introduction of bin-dependent item profits implies a relevant generalization of the constructive heuristics used to tackle both problems.

4. Heuristics

In this section, we present efficient heuristics for addressing the GBPPI. As mentioned in the introduction and will be shown in Section 5.3, the use of commercial solvers is not sufficient to efficiently address the problem, in particular if fast solutions with a computational time of at most 10 minutes are required (i.e., the average time for reaching decisions at the operational level). For this reason, we developed a series of heuristics useful for addressing the GBPPI. The common principle of our heuristics is to address problems where the sign of the objective function can be either positive, null, or negative. In previous bin-packing problems the goal was to minimize a single objective: the number of bins used, the wasted space, etc. This implied an objective function which is always non-negative. Vice versa, in generalized bin packing problems like the the GBPP or the GBPPI, we deal with an objective function which terms can have different signs. In fact, minimizing the net cost implies the optimization of two contributes: the minimization of the costs (which signs are non negative) and the maximization of the profits (which signs are non positive). As it will be shown in this section, our heuristics take this broadening of the objective function into account. Thus, they are also suitable to address those problems with a mixed target in the objective function. The proposed heuristics are:

- one constructive heuristic named Best profitable (BP)
- one constructive heuristic named Best Assignment (BA),
- one metaheuristic named Greedy Adaptive Search Procedure (GASP) (Festa & Resende, 2011),
- a parallel matheuristic named Model-Based Matheuristic (MBM).

These heuristics provide a flexible trade-off between quality of solution and computational time, in light of the imposed maximum computational time of 10 minutes.

4.1. The constructive heuristics

The proposed constructive heuristics are a variant of Best Fit Decreasing (BFD) introduced by Johnson, Demeters, Hullman, Garey, and Graham (1974) to address BPP. As already discussed, this generalization is necessary to address GBPPI and problems with a mixed objective function. Our constructive heuristics are called Best profitable (BP) and Best Assignment (BA), and operate with a list of available bins *SBL* and one of sorted items *SIL*. The major variant implemented in order to address the GBPPI is the broadening of the definition of the *best* bin. Let $\mathcal{S} \subseteq \mathcal{J}$ be the set of bins used in a solution of a bin packing problem, and let $W_{res}(b)$ be the residual volume of bin $b \in \mathcal{S}$. In previous versions of the bin-packing problem, the best bin for an item was defined as the one that can accommodate the item such that the residual space is minimized. In GBPPI, instead of considering the minimum residual space, we compute a figure of merit consisting of a weighted sum that takes into account both item profit and bin volume. Again, this choice is motivated by the fact that in the GBPPI, we need to consider two factors. In classical bin packing problems the aim is to minimize residual space in order to reduce the number of bins used or the cost of bin usage. The adoption of this approach in the GBPPI does not always guarantee effective outcomes because item profits in such cases rely heavily on bins. Our weighted figure of merit, defined as $\alpha \cdot p_{i\sigma(j)} - (1-\alpha) \cdot W_{res}(j)$, simultaneously addresses these two loading policies. The term $p_{i\sigma(j)}$ maximizes item profit, whereas the term $W_{res}(j)$ minimizes residual space. $\alpha \in [0,1]$ is a coefficient that is varied during the execution of GASP (cf. Section 4.2), and allows both loading policies to be spanned.

A further generalization is that this definition of the best bin is applied to a subset of $N \ll |SIL|$ items, rather than a single item. When we consider item i in list *SIL*, we take into account the sublist $SIL' = \{i, i+1, \ldots, i+N-1\}$. For each item, we compute the best bin; at the end of this process, we select the best item $i^* \in SIL'$ and the best bin b that maximizes the aforementioned figure of merit. Our computational experience confirmed that this "medium-term" memory improves the performance of the heuristics. The behavioral difference between BP and BA heuristics can be observed when we cannot load item $i \in SIL$ into any of the already bins used in $\mathcal{S}$. In this case, we try to select a new bin where to accommodate item i. The BP heuristic considers item i with the remaining succeeding items in *SIL*, and selects the bin that minimizes the difference between bin cost and the sum of profits from items that can be loaded into that bin. If this difference is positive and item i is non-compulsory, then item i is discarded because it is not convenient to open a new bin. Similarly, the BA heuristic selects the bin that maximizes profit for item i. Both heuristics perform a post-optimization procedure consisting of two parts. In the first part, for each bin $b \in \mathcal{S}$ contributing to the solution, we try to perform, if possible, the best swap with a bin $b' \in \mathcal{J} \setminus \mathcal{S}$ that has not been used. Clearly, the swap is possible if the items loaded into bin b can also be accommodated into bin b', and the difference between the cost of bin b and the sum of the profits of the items loaded in it is greater than the difference between the cost of bin b' and the sum of the profits of the same items loaded into bin b'. Furthermore, in the second part of the post-optimization procedure, we remove bins from the solution that are not profitable and do not contain compulsory items. It is clear that the solutions provided by the BP and BA heuristics depend on the two parameters α and N. As we show in Section 4.2, the values of these parameters are varied using the GASP procedure, which employs BP and BA as sub-heuristics. In contrast to classical bin packing heuristics such as the Best Fit Decreasing, we wish to point out that the ordering of items for the BP and BA is something outside the algorithmic framework. In fact, in Section 4.2 we show that GASP will be responsible to sort the items before using any constructive heuristics. Otherwise, if the BP and BA are used as stand-alone heuristics, many sortings of items are possible due to the presence of multiple and bin-dependent attributes. This behavior can also be observed in GBPP, where Baldi et al. (2012) studied different sortings for constructive heuristics. According to their study and to the knapsack-problem heuristics (Martello & Toth, 1990), the best performance on average can be obtained sorting the items by non-increasing profit over weight ratios and then by non-increasing weight.

4.2. The GASP

Greedy Adaptive Search Procedure (GASP) algorithms consist of a multi-start procedure to find a good initial solution and of a loop where, at each iteration, a new solution is generated by means of a simpler heuristic. In our GASP for GBPPI, we employ the Best profitable and Best Assignment heuristics, which were already described in Section 4.1.

Moreover, we try to improve the solution performing a steepest descent local search where the neighborhood consists of "1 to 1 swaps". We perform the swaps each time an improving solution is found. A swap consists in unloading one item, say i_1, to create sufficient room to accommodate an unloaded item i_2. The swap is only performed if it is possible and profitable.

At each iteration of the main loop, we try to generate a different and improved solution by varying the order of the items in the list *SIL*. This is performed by associating a *score* with each item. A score update procedure randomly assigns a different score value to each item.

Finally, a long-term reinitialization procedure is executed each time a solution does not improve in consecutive iterations. Its pur-

pose is to explore a different area of the feasible set by changing parameters α and N of the constructive heuristics. The proposed method incorporates some ideas from Perboli, Crainic, and Tadei (2011). The pseudo-code of our GASP is proposed in Algorithm 1.

Algorithm 1 The GASP.

```
IS: initial solution provided by the MULTI-START INITIALIZATION procedure
BS: best solution
BS := IS
numConsecutive: number of consecutive non-improving solutions
numConsecutive := 0
while time limit has not been reached do
    sort the items
    perform either the BP or the BA constructive heuristic
    store the resulting solution as CS
    if CS < BS then
        BS := CS
        perform "1 to 1" swaps
        numConsecutive := 0
    else
        numConsecutive := numConsecutive + 1
    end if
    SCORE UPDATE procedure
    if numConsecutive = MAXCONSECUTIVE then
        LONG-TERM REINITIALIZATION procedure
        numConsecutive := 0
    end if
end while
```

The algorithm presented in this section satisfies the terminology of the term GASP, namely Greedy Adaptive Search Procedure. It is a greedy algorithm because it is based on the BP and BA procedures that can be classified as greedy algorithms. It is also an adaptive-search algorithm because the long-term reinitialization procedure is helpful to explore new regions of the solution space.

The GASP metaheuristic can be easily parallelized. Let $\mathcal{P}$ be the set of available threads in a parallel computation. It is enough to execute one GASP metaheuristic for each thread $p \in \mathcal{P}$ and with a different seed for the random-number generator. Let $BP(p)$ be the final best solution provided by the GASP executed by thread $p \in \mathcal{P}$, then the overall best solution BS will be

$$BS = \min_{p \in \mathcal{P}} BS(p).$$

4.2.1. Multi-start initialization

The purpose of our multi-start initialization procedure is twofold: to feed the main loop of GASP with a good initial solution, and automatically calibrate the parameters used by the constructive heuristics. As already discussed when introducing the constructive heuristics, in contrast to the classical bin packing problem the concept of *best bin* does not merely depend on the residual volume. The introduction of bin-dependent item profits makes it impossible to define the best bin just in terms of residual volume. In fact, the following factors should be taken into account: (1) the residual volume itself, (2) the item profit, and (3) the bin type. The multi-start initialization is an auto-calibrating procedure where the BEST PROFITABLE heuristcs is executed a number of times, each time varying the parameter N within a given range. The value of N providing the best solution is used in the following part of the heuristic.

4.2.2. Score update

Item scores are randomly extracted from a discrete uniform distribution. The motivation for this choice is that working with integer values rather than real values implies a faster resolution of the sorting procedure. We prefer to distribute the item scores in a range proportional to the number of items, and then use integer scores rather than concentrating the scores with real values within a smaller range.

4.3. The Model-Based Matheuristic

We present here a parallel matheuristic for the GBPPI, where a set of computer threads $\mathcal{P}$ run simultaneously. It consists of a loop, where at each iteration each thread solves a subproblem using model (3)–(11). The resolution of subproblems with model (3)–(11) allow us to take into account two targets at the same time, namely the minimization of the cost and the maximization of the profits. Thus, this matheuristic is suitable to address problems with a similar structure.

In each subproblem, a small set of bins is randomly selected from the incumbent solution and the set of available bins. Then, we solve model (3)–(11) using these bins, the items loaded into the bins in the incumbent solution, and the items that have not been loaded in the incumbent solution. In order to further improve the given solution, we merge couples of partial solutions that do not have any bins and items in common. The best solution is updated if the new solution is an improvement over the incumbent one. This process continues until an overall time limit is reached.

According to the taxonomy of parallel methods proposed by Crainic and Toulouse (2010), this approach can be classified as 1C/RS/MPSS, where the following hold:

- 1C: One Control, i.e., one master thread controls all the remaining threads.
- RS: Rigid Synchronization, i.e., at each iteration, we wait for all threads to complete their computations.
- MPSS: Multiple points same strategy, i.e., each thread solves a different subproblem, but using the same strategy.

We chose to use this kind of parallelism because it can easily be implemented in our matheuristic. As we show in the next section, although the matheuristic can be exploited without parallel computation, this approach is strongly recommended when GBPPI is employed as a subproblem of a larger problem, and significantly improves the performance of the matheuristic. The main steps of the MBM matheuristic are shown in Algorithm 2.

Algorithm 2 The MBM matheuristic.

```
P: set of threads
IS: initial solution provided to the MBM metaheuristic
BS: best solution
BS := IS
while time limit has not been reached do
    for all p ∈ P do
        randomly select a subset b(p) of bins from the set of bins S making up solution BS.
        solve a GBPPI subproblem with a solver with a time limit of 1s, the bins b(p) plus selected available bins, the items loaded into bins b(p), and the items not loaded in solution BS.
    end for
    merge partial solutions provided by each thread and store the new current solution in CS.
    if CS < BS then
        BS := CS
    end if
end while
```

The initial solution can be any feasible solution. However, for better results it is important to start with a good solution. In our

Table 1
Instance features.

Features	Values
# of items	25
	50
	100
	200
	500
Item volume	I1: [1, 100]
	I2: [20, 100]
	I3: [50, 100]
Bin types	A: 100, 120, 150
	B: 60, 80, 100, 120, 150

computational tests we used the solution found by the GASP as initial solution of the MBM.

5. Computational results

In this section we analyze different computational aspects of the GBPPI. First, we compare the GBPPI with the classical GBPP, showing how the introduction of the bin dependency of the profits makes our problem much difficult to solve. Then, we show the accuracy and the effectiveness of our metaheuristics. Finally, we show how using the GBPPI to solve the problem faced by an international courier of choosing the mix of delivery options/subcontractors can bring to a reduction of the operational costs, with a clear benefit for the company.

5.1. Test environment

We extended the original instances for the GBPP (Baldi et al., 2012) by introducing bin dependency in the item profits. Table 1 lists the features of these instances in terms of items and bin types.

Ten instances were generated for each combination of features in Table 1, for a total of 300 instances. For each instance, the minimum and the maximum number of available bins for bin type $t \in \mathcal{T}$ is respectively set to

$$L_t = 0, \qquad \forall t \in \mathcal{T}$$
$$U_t = \left\lceil \frac{\sum_{i \in \mathcal{I}} w_i}{W_t} \right\rceil, \qquad \forall t \in \mathcal{T}.$$

The aforementioned 300 basic instances are then used to generate four classes of instances, numbered from 0 to 3, differing for the presence of compulsory items and the profit of the not compulsory ones. Class 0 considers the special case in which all the items as compulsory, and thus the items have no profit associated to them. In classes 1 and 2 all the items are non compulsory and an item profit p_i of item $i \in \mathcal{I}$ is given by

$$p_i \in \lceil \mathcal{U}(0.5, 3) w_i \rceil \qquad \text{class 1}$$
$$p_i \in \lceil \mathcal{U}(0.5, 4) w_i \rceil \qquad \text{class 2}$$

respectively, where $\mathcal{U}$ denotes uniform distribution.

Finally, Class 3 is a 500-item class, with 60 instances selected from classes 0–2 with 0, 25, 50, 75, and 100% of compulsory items.

Given the item profit (for the instances of classes from 1 to 3), the bin-dependant profit p_{it} is generated as

$$p_{it} = p_i + \theta_t$$

with

$$\theta_t = \max_{l \in L} \left\{ 0.4 \frac{C_{\min} C_{\max}}{W_t} \vartheta \right\}. \tag{12}$$

Like in Perboli et al. (2014c), ϑ can either be extracted from a uniform distribution $\mathcal{U}(0, 1)$ or from a Gumbel distribution $\mathcal{G}(0, 1)$. Values in (12) ensure that profit values approximate realistic settings. $C_{\min}$ and $C_{\max}$ are the extremes of the uniform distribution and of the truncated Gumbel distribution. This is done to avoid unrealistic values. For the sake of brevity, we refer to instances of GBPPI with item profits extracted from a Gumbel distribution according to (12) as Gumbel instances, and instances of GBPPI with item profits extracted from a uniform distribution according to (12) as uniform instances. GBPPI instances are also available on line (ORO group, 0000).

Computational resources were provided by HPC@POLITO (HPC@POLITO, 0000), on a system with Opteron 2.3 gigahertz processors and 124 gigabytes of RAM. The matheuristic was implemented in C++, with eight threads. We used the commercial solver CPLEX 12.6 (Cplex 12.6., 0000) to solve each subproblem and for the comparison between the GBPP and the GBPPI (see Section 5.3). Notice that, to avoid simmetry issues, the following constraints can be added to model (3)–(11):

$$y_j \leq y_{j+1} \qquad \forall j \in \mathcal{J} : \sigma(j) = \sigma(j+1). \tag{13}$$

5.2. Calibration

We select the 20% of the instances and run the algorithm within a range of parameters. The values yielding the best performance are then selected.

At each iteration of the GASP scores are randomly extracted in the range $[0, 4n]$ from a uniform and discrete distribution, where n is the number of items.

We set the maximum number of iterations of GASP to 2000 and the maximum number of consecutive non-improving solutions to 200.

The matheuristic randomly selects from a uniform discrete distribution $\min(4, k-1)$ bins from the incumbent solution, where k is the number of bins in the incumbent solution. This number is relatively small because greater values would cause the solver to take more time for the resolution of each subproblem. Moreover, small values of the available bins allow an easier merging of the subproblems. For the same reasons we added 3 extra available bins for each bin type in addition to the selected bins. Furthermore, we used the following time limits: 1 second for each subproblem solved with the solver, and 120 seconds for the matheuristic in total.

5.3. Comparison between the GBPP and the GBPPI

We compare here the classical version of the GBPP versus the new GBPPI. It is important to design a heuristic that is accurate and fast at the same time, in order to be able to exploit the benefits of the GBPPI in a city logistics setting, as discussed in Section 1.

To make the comparison, we selected 30 instances of the GBPP along with the corresponding Gumbel and uniform instances of the GBPPI, as explained in the text. Thus, the number of instances in this instance set is 90. The number of items range from 25 to 500. We solved each instance with the solver, eight threads and a time limit of six hours. Furthermore, we monitored the percentage gap provided by the solver after one minute, 10 minutes, one hour, and six hours of computation. The results of these initial tests are summarized in Table 2, where we show the gaps in mean percentage between test instances, grouped by the number of items and type of instances. In particular, C denotes the original instances of GBPP, G stands for the Gumbel instances of GBPPI, and U represents the uniform instances of GBPPI.

From Table 2, we can see that the gaps provided by the solver for the GBPPI instances (both Gumbel and uniform) are clearly higher than those of the original GBPP. For example, the gap in mean percentage of the 500-item-GBPP instances is 0.43%, whereas

Table 2
Comparisons between the classical GBPP and the new GBPPI.

Items	Problem	1 Minute	10 Minutes	1 Hour	6 Hours
	C	0.34	0.00	0.00	0.00
25	G	0.00	0.00	0.00	0.00
	U	2.29	0.43	0.43	0.00
	C	0.55	0.38	0.32	0.29
50	G	3.53	2.12	1.41	0.87
	U	2.92	1.60	1.17	0.86
	C	1.25	0.83	0.67	0.52
100	G	4.37	3.31	2.71	2.13
	U	4.86	3.75	3.18	2.43
	C	3.49	1.02	0.71	0.62
200	G	11.35	8.40	4.72	3.75
	U	9.31	5.02	3.71	3.50
	C	***	1.43	1.40	0.43
500	G	***	38.19	5.54	2.50
	U	***	20.57	5.52	2.72

Table 3
Percentage gaps of the constructive algorithms with the bin dependent profits generated according to the Gumbel and Uniform distributions.

Items	Gumbel	Uniform
25	13.07	14.28
50	15.05	14.06
100	14.76	12.56
200	13.77	11.86
500	9.98	8.24
Overall	**13.32**	**12.20**

the gap in mean percentage of the corresponding GBPPI instances is 2.50% for the Gumbel instances and 2.72% for the uniform instances. This suggests that the introduction of the dependence of item profits on bins involves greater computational effort.

It is worth remarking on the trend in gaps with respect to the range of the minutes. This interval is crucial in city-logistics settings, where decisions need to be made quickly (Perboli, De Marco, Perfetti, & Marone, 2014a). It was not possible to report the percentage gaps for the 500-item instances after one minute of computation because the solver was unable to determine an initial solution (i.e., the percentage gap was arbitrarily large). The percentage gaps provided by the solver after 10 minutes for the GBPPI instances were acceptable for the 25-item instances only. As the number of items increased to 50, the gaps increased to 2.12% for Gumbel instances and to 1.60% for uniform instances. The percentage gaps tended to increase as the number of items increased. This trend became considerably more evident for the 500-item instances, with the GBPP instances having a mean percentage gap of 1.43 and the GBPPI having gaps of 38.19% and 20.57%, respectively, for the Gumbel and uniform instances. These gaps were clearly prohibitive. Moreover, for some instances the solver was unable to find an initial solution after 10 minutes of computation. All these considerations indicate that the solver alone is not appropriate to address GBPPI in a city-logistics environment where decisions are made quickly and the number of items is greater than 500.

In Table 3, we list the percentage gaps between the algorithms proposed in this paper and the percentage gap of the best (minimum) between the Best profitable (BP) and Best Assignment (BA) constructive heuristic with respect to the solver the best solution found by the solver with a time limit of 1 hour. The results are reported according to the type of instance (Gumbel and uniform) and the number of items (from 25 to 500). This means that both algorithms are used and the best solution is kept. The reason for this choice is twofold: (1) these constructive heuristics are simple and to execute each of them does not impact on the overall computational time, and (2) the BP and BA algorithms do not dominate each other. Over the 600 instances, in 225 instances (about one third) the BP was better than the BA. For the remaining instances, the BA provided a better result.

Table 4
Comparison between the BFD and the proposed constructive algorithms.

Items	Gumbel	Uniform
25	40.55	42.01
50	44.76	42.56
100	41.74	41.31
200	37.87	35.88
500	33.38	32.33
Overall	**39.66**	**38.82**

The best constructive heuristic was compared to the solver with one thread only because the execution of BP and the BA heuristic does not require parallelism. The computational time of the constructive heuristic was practically zero, but this immediate execution was paid for in terms of the highest percentage gaps. In fact, the overall is quite high, with a mean gap greater than 12%. This remarks again the difficulty given by the introduction of the bin dependency of the items' profits. In fact both the BP and BA extends quite known concepts in the packing literature that normally gives gaps less than 1% (Baldi et al., 2012; Crainic et al., 2011). Actually, as already noticed in multi-dimensional packing problems (Crainic, Perboli, & Tadei, 2008; Perboli et al., 2014b), the presence of multiple ordering options deteriorates the performances of traditional concepts. This, in conjunction with a profit scheme linking the sets of bins and items, make the best and next fit concepts to have bad performances.

Results in Table 3 become more significant if we compare the proposed constructive heuristics with those constructive heuristics designed for previous bin-packing problems. The most effective constructive heuristic is the Best Fit Decreasing (BFD) (Baldi et al., 2014). We used the instances presented in this section to compare the BP and BA with the BFD. As already discussed in Section 4.1, the main innovations for the new constructive methods are the introduction of a memory for the candidate items to be accommodated and a mixed figure of merit for selecting the best bin. In Table 4, we report the percentage gap of the best constructive function with the classical BFD, namely

$$100 \cdot \frac{BFD - \min(BP, BA)}{|BFD|}.$$

These results show that while the BFD is still useful for the GBPP, it has to be replaces with better constructive heuristics when addressing the GBPPI because, as already discussed, the introduction of bin-dependent item problems does not change the solution set but strongly modifies the nature of the problem.

5.4. Computational results of the heuristic methods

In Table 5, we list the percentage gaps between the algorithms proposed in this paper and the best solution found by the solver in one hour. The results are reported according to the type of instance (Gumbel and uniform), the number of items (from 25 to 200), and the number of threads (one and eight). We also wanted to estimate the benefits of a parallel computation in comparison with a single-thread approach. The columns in Table 5 describe the following: col. (1) is the type of distribution for item profits; col. (2) shows the number of items; col. (3) shows the percentage gap in the best constructive heuristic with respect to the solver with one thread; cols. (4) and (5) are the percentage gaps of GASP and the MBM with one thread with respect to the solver with one thread; and cols. (6) and (7) are the percentage gaps of GASP and the MBM with eight threads with respect to the solver with eight threads.

Table 5
Percentage gaps in the proposed algorithms reported according to item profit distribution (column 1), number of items (column 2), and parallelism (columns 4–7).

Distribution	Items	Constr.	1 Thread		8 Threads	
			GASP	MBM	GASP	MBM
Gumbel	25	13.07	0.97	0.10	0.86	0.02
	50	15.05	3.94	0.16	2.69	0.14
	100	14.76	5.69	0.39	4.66	0.31
	200	13.77	6.89	0.12	4.56	−0.02
	500	9.98	2.99	−1.95	1.03	−2.94
	Overall	**13.32**	**4.10**	**−0.24**	**2.76**	**−0.50**
Uniform	25	14.28	1.03	0.05	0.97	0.04
	50	14.06	3.71	0.24	2.89	0.17
	100	12.56	5.93	0.41	4.52	0.39
	200	11.86	7.24	0.19	5.01	0.16
	500	8.24	2.44	−2.19	1.04	−2.97
	Overall	**12.20**	**4.07**	**−0.26**	**2.89**	**−0.44**
Overall		**12.76**	**4.08**	**−0.25**	**2.82**	**−0.47**

Table 6
Best computational times for the MBM reported according to item profit distribution (column 1), number of items (column 2), and parallelism (columns 3–4).

Distribution	Items	1 Thread	8 Threads
Gumbel	25	2.94	2.48
	50	31.16	13.56
	100	64.97	47.30
	200	92.88	70.87
	500	99.03	104.50
	Overall	**58.20**	**47.74**
Uniform	25	5.74	1.84
	50	29.00	14.31
	100	57.25	45.05
	200	93.70	74.67
	500	99.03	109.43
	Overall	**56.94**	**49.06**
Overall		**57.57**	**48.40**

The best constructive heuristic was given by the minimum of the Best profitable and Best Assignment heuristics.

In column 3 we report the results of the constructive heuristics. Although this value was high, the Best profitable and Best Assignment heuristics found their utility in the execution of GASP and MBM, which yielded much lower percentage gaps. Moreover, from Table 5 we can also observe the benefits of introducing parallelism. The percentage gap of GASP was almost halved when switching from one to eight threads. This gap reduced from 4.08 to 2.82%, which is a considerably improved result if we consider that the computational time of GASP was approximately 1 second. The percentage gaps of the MBM were satisfactory. Again, parallelism improved quality of solution. In fact, gaps reduced from −0.25 to −0.47% when the number of threads increased from 1 to 8.

Table 6 presents the average computational times of the last best solution found by the MBM. The columns in Table 6 indicate the following: col. (1) represents the type of distribution for the item profits, col. (2) shows the number of items, cols. (3) and (4) list the average computational times needed to find the best solution for the MBM with one and eight threads, respectively.

The analysis of Table 6 reveals that the average time tended to increase in line with instance size. Moreover, the overall average time was approximately one minute.

Finally, we compared the MBM with the branch-and-price by Baldi et al. (2014) on the same instances as the classical GBPP. We computed the percentage gaps of the MBM with respect to the branch-and-price results. The results are presented in Table 7 according to the number of items. In particular, columns 2, 4, 6, and 8 indicate the percentage gaps of the branch-and-price with respect to its best lower bound computed at the root node (cf. Baldi et al., 2014 for further details). Columns 3, 5, 7, and 9 present the percentage gaps of the MBM with respect to the best objective function provided by the branch-and-price. The instances in Class 3 were defined for 500 items only. The overall percentage gap of the MBM compared to the branch-and-price was approximately 0.22%. Nevertheless, we observed that for 34 instances, better results were found than those provided by the branch-and-price in Baldi et al. (2014), and within a considerably smaller computational time. The time limit of the branch-and-price was one hour, while that of the MBM was two minutes. Moreover, for a third of the instances of GBPP, the Model-Based Matheuristic could find the same results generated by the branch-and-price.

5.5. Smart city case study

As described in Section 1, the case study refers to the planning of parcel deliveries of an international courier. In this case study, we compare our MBM solutions with those of the courier based on its business policy. We analyzed 30 instances (i.e., 30 competitive tenders) with 1000 daily parcel deliveries (i.e., the items), 10 TCs (i.e., the bin types), and up to 100 trucks (i.e., the bins) per TC, available in an urban distribution area.

In Table 8, we present the percentage gaps of our MBM compared to the business policy and the solver (this time with a time limit of two hours) over the 30 competitive tenders. Our MBM always finds better results than those provided by the business policy and the solver.

A more interesting outcome results from the analysis of the instances from an economic and managerial point of view. The size of each instance is representative of the daily parcels of a parcel delivery company in a medium-sized city. If we compare the solution with one based on expert opinion (tactical decisions based on expert opinion and day-to-day decisions optimized by means

Table 7
Comparisons between the branch and price and the MBM. Columns 2, 4, 6, and 8 show the percentage gap of the branch and price with respect to the best lower bound. Columns 3, 5, 7, and 9 show the percentage gap of the MBM with respect to the best objective function provided by the branch and price.[1] The instances in Class 3 were defined for 500 items only.

Items	**Class 0**		Class 1		Class 2		Class 3 [1]	
	BEP	MHEUR	BEP	MHEUR	BEP	MHEUR	BEP	MHEUR
25	0.00	0.01	0.00	0.02	0.00	0.00	N/A	N/A
50	0.00	0.19	0.00	0.14	0.01	0.12	N/A	N/A
100	0.02	0.32	0.03	0.26	0.01	0.15	N/A	N/A
200	0.06	0.33	0.02	0.37	0.03	0.28	N/A	N/A
500	0.12	0.38	0.12	0.34	0.11	0.26	0.57	0.59
Overall	**0.04**	**0.25**	**0.03**	**0.23**	**0.03**	**0.16**	**0.57**	**0.59**

Table 8
Percentage gaps of the MBM compared with the business policy and the solver. Columns 1 and 4 present the instance number of the case study, columns 2 and 5 the percentage gaps of the business policy compared to the MBM, and columns 3 and 6 the percentage gaps of the solver compared to the MBM.

Instance	Courier gap	Solver gap	Instance	Courier gap	Solver gap
Instance 1	−4.10	−1.22	Instance 16	−3.19	−1.47
Instance 2	−3.54	−0.98	Instance 17	−3.14	−1.04
Instance 3	−3.89	−1.21	Instance 18	−3.17	−1.15
Instance 4	−3.85	−1.86	Instance 19	−3.18	−0.69
Instance 5	−3.80	−1.95	Instance 20	−3.70	−1.28
Instance 6	−3.79	−1.12	Instance 21	−3.05	−2.15
Instance 7	−3.82	−1.35	Instance 22	−3.05	−0.91
Instance 8	−3.56	−1.18	Instance 23	−2.80	−2.25
Instance 9	−3.84	−1.05	Instance 24	−3.14	−2.57
Instance 10	−4.24	−0.92	Instance 25	−3.12	−2.57
Instance 11	−3.45	−1.66	Instance 26	−2.88	−2.29
Instance 12	−2.90	−0.52	Instance 27	−2.86	−2.62
Instance 13	−3.24	−1.49	Instance 28	−3.10	−2.04
Instance 14	−3.22	−1.27	Instance 29	−3.07	−1.83
Instance 15	−3.33	−0.26	Instance 30	−2.89	−0.79
			Overall	**−3.36**	**−1.46**

of specific optimization tools), the assignments given by GBPPI achieve a constant reduction of the overall cost by between 3 and 4%. In terms of the economic impact of costs, this amounts to 120,000–180,000 euros for a medium-sized city (the interval depends on different scenarios of annual numbers of parcels). Moreover, this economic impact will increase in the near future, due to the increase in B2C flows resulting from e-commerce, mass customization, and decentralized production (Inc, 2016; 2017).

6. Conclusions

In this paper, we introduced a new packing problem named Generalized Bin Packing Problem with bin-dependent item profits (GBPPI). We have shown that GBPPI can be applied at both tactical and operational levels. At the tactical level, GBPPI models cross-country and multi-modal transportation settings. At an operational level, GBPPI describes the problem of a courier selecting the appropriate number and type of vehicles from a set of available transportation companies. We have also demonstrated that the introduction of bin-dependent item profits is not trivial in terms of problem resolution. We presented a number of heuristics to efficiently address the problem within limited computational time. We have also presented extensive computational results and a case study of a well-known international courier operating in northern Italy.

GBPPI can also be a starting point for future research. In fact, GBPPI can be exploited as a subproblem in the resolution of the stochastic variant of GBPP, namely Stochastic Generalized Bin Packing Problem. Stochastic problems are affected by uncertainty. The Progressive Hedging Algorithm is an iterative technique for addressing these problems (Crainic, Gobbato, Perboli, & Rei, 2016; Manerba & Perboli, 2019). At each iteration, random variables describing uncertain attributes are fixed to particular values. In this way, each iteration of the Progressive Hedging Algorithm applied to Stochastic Generalized Bin Packing Problem implies the solution of a deterministic GBPPI subproblem. This solution may successfully be performed through the heuristics proposed in this article.

Acknowledgments

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 636354 and the *PRIN 2015 project, Italian University and Research Ministry*.

This project reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

References

Aljohani, K., & Thompson, R. G. (2018). The case of Melbourne's fruit & vegetable wholesale market. *Case Studies on Transport Policy, 6*(2), 279–288.
Amazon Inc (2016). Annual report. http://www.annualreports.com/HostedData/AnnualReportArchive/a/NASDAQ_AMZN_2016.pdf
Amazon Inc (2017). Annual report. https://ir.aboutamazon.com/static-files/917130c5-e6bf-4790-a7bc-cc43ac7fb30a
Baldi, M. M., & Bruglieri, M. (2016). On the generalized bin packing problem. *ITOR*. doi:10.1111/itor.12258.
Baldi, M. M., Crainic, T. G., Perboli, G., & Tadei, R. (2012). The generalized bin packing problem. *Transportation Research Part E, 48*(6), 1205–1220. doi:10.1016/j.tre.2012.06.005.
Baldi, M. M., Crainic, T. G., Perboli, G., & Tadei, R. (2013). Asymptotic results for the generalized bin packing problem. *Procedia Social and Behavioral Sciences, 111*, 663–671. doi:10.1016/j.sbspro.2014.01.100.
Baldi, M. M., Crainic, T. G., Perboli, G., & Tadei, R. (2014). Branch-and-price and beam search algorithms for the variable cost and size bin packing problem with optional items. *Annals of Operations Research, 222*(1), 125–141. doi:10.1007/s10479-012-1283-2.
Boysen, N., Schwerdfeger, S., & Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research, 271*(3), 1085–1099.
Cplex, 12.6. http://www.ibm.com/support/knowledgecenter/SSSA5P_12.6.2/ilog.odms.cplex.help/CPLEX/homepages/CPLEX.html.
Crainic, T. G., Gobbato, L., Perboli, G., & Rei, W. (2016). Logistics capacity planning: A stochastic bin packing formulation and a progressive hedging meta-heuristic. *European Journal of Operational Research, 253*(2), 404–417.
Crainic, T. G., Gobbato, L., Perboli, G., Rei, W., Watson, J. P., & Woodruff, D. L. (2014). Bin packing problems with uncertainty on item characteristics: An application to capacity planning in logistics. *Procedia Social and Behavioral Sciences, 111*, 654–662.
Crainic, T. G., Perboli, G., Pezzuto, M., & Tadei, R. (2007a). Computing the asymptotic worst-case of bin packing lower bounds. *European Journal of Operational Research, 183*, 1295–1303.
Crainic, T. G., Perboli, G., Pezzuto, M., & Tadei, R. (2007b). New bin packing fast lower bounds. *Computers & Operations Research, 34*, 3439–3457. doi:10.1016/j.cor.2006.02.007.
Crainic, T. G., Perboli, G., Rei, W., & Tadei, R. (2011). Efficient lower bounds and heuristics for the variable cost and size bin packing problem. *Computers & Operations Research, 38*, 1474–1482.
Crainic, T. G., Perboli, G., & Tadei, R. (2008). Extreme point-based heuristics for three-dimensional bin packing. *INFORMS Journal on Computing, 20*, 368–384.
Crainic, T. G., & Toulouse, M. (2010). Parallel meta-heuristics. In M. Gendreau, & J. Y. Potvin (Eds.), *Handbook in metaheuristics* (pp. 497–542). Norwell, MA: Kluwer Academic Publishers.
de Souza, R., Goh, M., Lau, H. C., Ng, W. S., & Tan, P. S. (2014). Collaborative urban logistics - synchronizing the last mile. A Singapore research perspective. *Procedia - Social and Behavioral Sciences, 125*, 422–431. Eighth International Conference on City Logistics 17–19 June 2013, Bali, Indonesia.
European Commission (2017). Energy and transport. transport in figure 2017. Technical report, Publications Office of the European Union.

Festa, P., & Resende, M. G. C. (2011). Grasp: basic components and enhancements. *Telecommunication Systems, 46*(3), 253–271.
Friesen, D. K., & Langston, M. A. (1986). Variable sized bin packing. *SIAM Journal on Computing, 15*, 222–230.
Giusti, R., Manerba, D., Perboli, G., Tadei, R., & Yuan, S. (2018). A new open-source system for strategic freight logistics planning: The SYNCHRO-NET optimization tools. In *Transportation research procedia: 30* (pp. 245–254).
Haouari, M., & Serairi, M. (2009). Heuristics for the variable sized bin-packing problem. *Computers & Operations Research, 36*, 2877–2884.
Haouari, M., & Serairi, M. (2011). Relaxations and exact solution of the variable sized bin packing problem. *Computational Optimization and Applications, 48*, 345–368.
Hemmelmayr, V., Schmid, V., & Blum, C. (2012). Variable neighbourhood search for the variable sized bin packing problem. *Computers & Operations Research, 39*, 1097–1108.
Holzapfel, A., Kuhn, H., & Sternbeck, M. G. (2018). Product allocation to different types of distribution center in retail logistics networks. *European Journal of Operational Research, 264*(3), 948–966.
HPC@POLITO. A project of academic computing within the department of control and computer engineering at the politecnico di torino. http://www.hpc.polito.it.
Johnson, D. S., Demeters, A., Hullman, J. D., Garey, M. R., & Graham, R. L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing, 3*, 299–325.
Maiza, M., Labed, A., & Radjef, M. S. (2013). Efficient algorithms for the offline variable sized bin-packing problem. *Journal of Global Optimization, 57*(3), 1025–1038.
Manerba, D., Mansini, R., & Riera-Ledesma, J. (2017). The traveling purchaser problem and its variants. *European Journal of Operational Research, 259*(1), 1–18.
Manerba, D., Mansini, R., & Zanotti, R. (2018). Attended home delivery: reducing last-mile environmental impact by changing customer habits. *IFAC-PapersOnLine, 51*(5), 55–60.
Manerba, D., & Perboli, G. (2019). New solution approaches for the capacitated supplier selection problem with total quantity discount and activation costs under demand uncertainty. *Computers and Operations Research, 101*, 29–42.
Martello, S., & Toth, P. (1990). *Knapsack Problems – Algorithms and computer implementations*. Chichester, UK: John Wiley & Sons.
Monaci, M. (2002). Algorithms for packing and scheduling problems. (Ph.D. thesis),Italy: Università di Bologna, Bologna.
ORO Group. Gbppi instances. https://bitbucket.org/ORGroup/gbppi_instances.
Perboli, G., Crainic, T. G., & Tadei, R. (2011). An efficient metaheuristic for multi-dimensional multi-container packing. In *Proceedings of the IEEE Conference on Automation Science and Engineering (CASE)* (pp. 563–568). doi:10.1109/CASE.2011.6042476.
Perboli, G., De Marco, A., Perfetti, F., & Marone, M. (2014a). A new taxonomy of smart city projects. *Transportation Research Procedia, 3*, 470–478.
Perboli, G., Gobbato, L., & Perfetti, F. (2014b). Packing problems in transportation and supply chain: New problems and trends. *PROCEDIA - Social and Behavioral Sciences, 111*, 672–768.
Perboli, G., Rosano, M., Saint-Guillain, M., & Rizzo, P. (2018). Simulation optimisation framework for city logistics: an application on multimodal last-mile delivery. *IET Intelligent Transport Systems*. doi:10.1049/iet-its.2017.0357.
Perboli, G., Tadei, R., & Baldi, M. M. (2012). The stochastic generalized bin packing problem. *Discrete Applied Mathematics, 160*, 1291–1297.
Perboli, G., Tadei, R., & Gobbato, L. (2014c). The multi-handler knapsack problem under uncertainty. *European Journal of Operational Research, 236*(3), 1000–1007.
Ranieri, L., Digiesi, S., Silvestri, B., & Roccotelli, M. (2018). A review of last mile logistics innovations in an externalities cost reduction vision. *Sustainability, 10*(3), 782.
Tadei, R., Perboli, G., & Baldi, M. M. (2012). The capacitated transshipment location problem with stochastic handling costs at the facilities. *International Transactions in Operational Research, 19*(6), 789–807.
Tadei, R., Perboli, G., & Perfetti, F. (2014). The multi-path traveling salesman problem with stochastic travel costs. *EURO Journal on Transportation and Logistic, 6*, 2–23. doi:10.1007/s13676-014-0056-2.
Ülkü, M. A., & Bookbinder, J. H. (2012). Optimal quoting of delivery time by a third party logistics provider: The impact of shipment consolidation and temporal pricing schemes. *European Journal of Operational Research, 221*(1), 110–117.
Vakulenko, Y., Hellström, D., & Hjort, K. (2018). What's in the parcel locker? Exploring customer value in e-commerce last mile delivery. *Journal of Business Research, 88*, 421–427.
Wang, Y., Zhang, D., Liu, Q., Shen, F., & Lee, L. H. (2016). Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. *Transportation Research Part E: Logistics and Transportation Review, 93*, 279–293.
Zhou, L., Baldacci, R., Vigo, D., & Wang, X. (2018). A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. *European Journal of Operational Research, 265*(2), 765–778.