Noname manuscript No.
(will be inserted by the editor)

On the Generalized Bin Packing Problem

Mauro Maria Baldi · Maurizio Bruglieri

Received: date / Accepted: date

Abstract The Generalized Bin Packing Problem (GBPP) is a novel packing problem arising in many transportation and logistic settings, characterized by multiple item and bin attributes and by the joint presence of both compulsory and non-compulsory items. We introduce a change in the definition of the problem that does not impact either on its feasible solution set or on its optimal solutions, but guarantees an objective function that is always non-negative, in order to satisfy the non-negativity requirement of the the worst-case ratio definition. In this way we can properly study the approximability of the GBPP. In this paper we study the computational complexity of the GBPP and we prove that the GBPP cannot be approximated by any constant ε, unless $P = NP$. Since the proof of non-approximability exploits the presence of two bin types, a separate study is made with a single bin type. We show that, in this particular case, the GBPP reduces to the Bin Packing with Rejections (BPR), which is approximable. In this particular setting we also study the behavior of standard and widespread heuristics like the FIRST FIT and the BEST FIT, showing that they fail in approximating the GBPP, even with just one bin type. Finally, we prove that the GBPP satisfies the Bellmans optimality principle and, exploiting this result, we develop a dynamic programming solution approach.

Keywords Bin Packing Problem · Variable Cost and Size Bin Packing Problem · approximability · Bellman's optimality principle · dynamic programming

1 Introduction

The Generalized Bin Packing Problem (GBPP) is a novel packing problem recently introduced in the literature by Baldi et al. (2012, 2014). Each instance of the GBPP consists of a set of items (with volume and profit) and of a set of bins (with capacity and cost). Items can be compulsory (i.e., mandatory to load) or non-compulsory, while bins are classified in bin types. Bins belonging to the same bin type have the same capacity and cost. Moreover, a maximum number of bins can be used for each bin type. The aim of the GBPP is to accommodate all the compulsory items and possible non-compulsory items into appropriate bins in order to minimize the overall cost, given by the difference between the costs of the selected bins and the profits of the loaded non-compulsory items.

The importance of the GBPP is twofold. Firstly, it is a problem which naturally arises in many freight transportation and logistic settings. Secondly, it is able to group a number of bin packing and knapsack problems into one distinct problem. This generalization, as the name of the problem suggests, also provides the advantage of exploiting the same resolution techniques to address different problems. It is easy to prove

Mauro Maria Baldi
Department of Control and Computer Engineering, Politecnico di Torino, Torino, Italy
E-mail: mauro.baldi@polito.it

Maurizio Bruglieri
Dipartimento di Design, Politecnico di Milano, Milano, Italy E-mail: maurizio.bruglieri@polimi.it

that the GBPP is rooted in earlier and well-studied bin packing problems which we briefly recall here. The oldest problem on which the GBPP is based is the Bin Packing Problem (BPP), where items are all compulsory and bins are all equal. The BPP was introduced in the seventies by a group of researchers (Ullman, 1971; Garey et al., 1972; Johnson et al., 1974). In these preliminary works, the authors formulated simple algorithms, which are still widely exploited nowadays. These are: the FIRST FIT (FF), the BEST FIT (BF), the FIRST FIT DECREASING (FFD) and the BEST FIT DECREASING (BFD). The Variable Size Bin Packing Problem (VSBPP) was introduced by Friesen and Langston (1986). In this variant of the problem, different bin types are available, but with the cost equal to the capacity. The reason for this choice is that the goal is to minimize the wasted space. In the Variable Cost and Size Bin Packing Problem (VCSBPP) (Correia et al., 2008; Crainic et al., 2011) bins are classified into bin types with different capacity and cost but the items are all compulsory. A less straightforward problem the GBPP is able to encompass is the Bin Packing with Rejections (BPR). The BPR, introduced by Dósa and He (2006), consists in a set of non-compulsory items, to be loaded into a set of identical bins. A penalty is paid if we omit to load an item. The goal is to minimize the sum of the number of bins used and the total penalty for the rejected items. Later works on the BPR are due to Bein et al. (2008), Epstein and Levin (2010) and Epstein (2010). In Section 3 we show how the BPR is also a particular case of the GBPP.

Although relevant work was done on the GBPP, only limited research has been devoted so far to the study of its approximability (Baldi et al., 2013).

In this paper we propose a thorough study on the approximability of the GBPP and show significant results. Firstly, we formally define the GBPP according to the previous works in the literature (Baldi, 2014; Baldi et al., 2012, 2014). According to this definition, given an instance of the problem, the sign objective function of any solution built from the instance can be either negative, zero or positive. Nevertheless the worst-case ratio definition requires the sign of the objective function of any solution of the problem to be non-negative. Therefore, we state an equivalent definition of the GBPP satisfying this requirement. We prove that not only the GBPP is a NP-hard problem, but that even the problem of finding a feasible solution is NP-hard as well. We also claim that the GBPP is not approximable, unless $P = NP$. Finally, we study the particular case of the GBPP with a single bin type and we show how it can be reduced to the BPR (Dósa and He, 2006; Bein et al., 2008; Epstein and Levin, 2010; Epstein, 2010), which is approximable. We also show that, while standard and widespread heuristics like the FF and the BF have a finite worst-case ratio when employed to address the Bin Packing Problem (Johnson et al., 1974), this ratio becomes infinite if they are extended to the GBPP. Finally, we prove a very important result, i.e., the GBPP satisfies the Bellman's optimality principle. Exploiting this principle we develop a dynamic programming solution approach.

This paper is organized as follows. In Section 2 we introduce the new definition of the GBPP and we study its computational complexity. In Section 3 we prove that the GBPP is not approximable, unless $P = NP$. We also study the particular case with a single bin type. In Section 4 we prove that the GBPP satisfies the Bellman's optimality principle and we propose the dynamic programming algorithm. Finally in Section 5 we draw some conclusions.

2 Definition and complexity of the GBPP

For each instance of the GBPP, a finite set of items I and a set of bins J are given. Each item $i \in I$ is characterized by volume w_i and profit p_i, while each bin $j \in J$ is characterized by capacity W_j and cost C_j. The set of items I is partitioned into two subsets: the subset of compulsory items I^{C} (i.e., those which are mandatory to load into any bin) and the set of non-compulsory items I^{NC}. Bins are also classified in bin types. T denotes the set of bin types. Bins belonging to the same bin type $t \in T$ have the same capacity and cost. For each bin $j \in J$ its type is indicated by $\sigma(j) \in T$. Moreover, a maximum number U_t of bins can be used for each bin type $t \in T$. Note that the case where the number of bins of type t is unlimited corresponds to $U_t = |I|$ (since each bin used must contain at least one item).

The aim of the GBPP is to accommodate all the compulsory items and possibly non-compulsory items into appropriate bins in order to minimize the overall cost, while satisfying capacity and bin-usage constraints. The overall cost is given by the difference between the costs of the chosen bins and the profits of the selected non-compulsory items. The profits of the compulsory items are not taken into account within the objective function because they act as a constant, as compulsory items are always loaded. Capacity constraints require that the sum of the volumes of the items accommodated in each bin must not exceed the capacity of that bin. Bin usage constraints require that, for each bin type $t \in T$ at most U_t bins of type t are used. We formalize the GBPP with the following definition.

Definition 1 *The* GBPP *consists in finding $\tilde{J} \subseteq J$, $\tilde{I}^{NC} \subseteq I^{NC}$ such that:*

$$\text{each item } i \in I^{C} \cup \tilde{I}^{NC} \text{ is assigned to one and only one bin } \beta(i) \in \tilde{J} \tag{1}$$

$$\sum_{i \in I^{C} \cup \tilde{I}^{NC} : \beta(i) = j} w_i \le W_j, \quad \forall j \in \tilde{J} \tag{2}$$

$$|\{j \in \tilde{J} : \sigma(j) = t\}| \le U_t, \quad \forall t \in T, \tag{3}$$

with the objective of minimizing

$$\sum_{j \in \tilde{J}} C_j - \sum_{i \in \tilde{I}^{NC}} p_i \tag{4}$$

Definition 1 is in agreement with the definition of the GBPP introduced in Baldi et al. (2012). However, this definition has the flaw that the sign of the objective function cannot be determined *a priori*. In fact the objective function becomes negative if feasible solutions $\tilde{J} \subseteq J$, $\tilde{I}^{NC} \subseteq I^{NC}$ such that $\sum_{j \in \tilde{J}} C_j < \sum_{i \in \tilde{I}^{NC}} p_i$ exist. This is a problem when dealing with the approximability of the GBPP because the worst-case ratio requires that the optimum is always non-negative (see Section 3). For this reason in this paper we have modified the definition of the GBPP into the following one.

Definition 2 *The* GBPP *consists in finding $\tilde{J} \subseteq J$, $\tilde{I}^{NC} \subseteq I^{NC}$ such that conditions (1), (2), (3) of Definition 1 are satisfied with the objective of minimizing*

$$\sum_{j \in \tilde{J}} C_j - \sum_{i \in \tilde{I}^{NC}} p_i + \sum_{i \in I^{NC}} p_i \tag{5}$$

Since in the new definition we only modify the original objective function (4) by adding the fixed quantity $\sum_{i \in I^{NC}} p_i$, any optimal solution according to Definition 1 is also optimal according to Definition 2. Therefore we are defining an equivalent problem. The advantage of Definition 2 is that the new objective function (5) is clearly always non-negative.

Note that, in some instances of the GBPP, the available bins may be not sufficient to load all the compulsory items. Such instances are clearly infeasible. In Proposition 1 we show that the decisional problem of determining whether a given instance of the GBPP has a feasible solution is $NP-complete$. We refer to this problem as the *existence* version of the GBPP.

Proposition 1 *The existence version of the* GBPP *is $NP-complete$, even with one bin type.*

Proof Consider the particular case of the GBPP where all the items are compulsory and all the available bins have the same capacity and same cost. Hence, $|T| = 1$ and U_1 identical bins are available. In this case, the *existence* version of the GBPP corresponds to the *existence* version of the BPP with a fixed number of bins (U_1) that Jansen et al. Jansen et al. (2010) proved to be $NP-complete$ (already for $U_1 = 2$). Since the *existence* version of the GBPP is $NP-complete$ in a particular case, it is so also in general. □

Note that in the proof of Proposition 1 we exploit the presence of compulsory items, while the particular case of the GBPP where all the items are non-compulsory is always feasible since $\tilde{I}^{NC} = \emptyset$ is a trivial feasible solution.

Despite Proposition 1, the GBPP is feasible under suitable hypothesis as established by the following proposition.

Proposition 2 *Every instance of the* GBPP *satisfying the following two conditions is feasible:*

$$w_i \le \max_{j \in J} W_j \ \forall i \in I^C \tag{6}$$

$$\exists j' \in J : W_{j'} = \max_{j \in J} W_j \text{ and } U_{\sigma(j')} \ge |I^C| \tag{7}$$

Proof Note that condition (6) is necessary to the feasibility since if it was violated for an item $\bar{\imath} \in I^C$ then no bin would have sufficient capacity to contain it, although item $\bar{\imath}$ is compulsory. Thanks to condition (7), a feasible solution can be always obtained assigning each compulsory item to a bin of type $\sigma(j')$ since there are $|I^C|$ bins of such a type and each of them can contain any item. □

It is also worth noting that we can transform each instance π of the GBPP, which in principle might be infeasible, into a new instance π' of the GBPP, always feasible, such that the possible infeasibility of π can be detected *a posteriori* solving the GBPP for π'. In fact, each compulsory item can be treated as a non-compulsory one by setting its profit equal to $\sum_{i \in I^{\mathrm{NC}}} p_i + \max_{j \in J} C_j + 1$. In this way, it is always convenient to load all the original compulsory items because their profits are bigger than the total profit of the original non-compulsory items and than the cost of every bin. Thus, the only reason that impedes loading an original compulsory item is an insufficient number of bins, that corresponds to the infeasibility of π. We can also prove that if π is feasible, then π' is equivalent to π i.e. any optimal solution of π is also optimal for π' and viceversa; moreover the optimal values of the two instances coincide. This is stated in the following proposition.

Proposition 3 *If π is feasible, π' is equivalent to π.*

Proof Let us indicate all the non-compulsory items of π' corresponding to the compulsory items of π with $\hat{I}^{NC}_{\pi'}$ and let us indicate the latter with I^C_π. While let I^{NC}_π and $I^{NC}_{\pi'}$ denote all the non-compulsory items of π and π', respectively. Since π is feasible, all the elements of $\hat{I}^{NC}_{\pi'}$ are always chosen in every optimal solution of π'. Therefore according to Definition 2, the optimal value is given by

$$\sum_{j \in \tilde{J}} C_j - \sum_{i \in \tilde{I}^{NC}_{\pi'}} p_i + \sum_{i \in I^{NC}_{\pi'}} p_i =$$

$$= \sum_{j \in \tilde{J}} C_j - \sum_{i \in \hat{I}^{NC}_{\pi'}} p_i - \sum_{i \in \tilde{I}^{NC}_{\pi'} \setminus \hat{I}^{NC}_{\pi'}} p_i + \sum_{i \in \hat{I}^{NC}_{\pi'}} p_i + \sum_{i \in I^{NC}_{\pi'} \setminus \hat{I}^{NC}_{\pi'}} p_i =$$

$$= \sum_{j \in \tilde{J}} C_j - \sum_{i \in \tilde{I}^{NC}_{\pi}} p_i + \sum_{i \in I^{NC}_{\pi}} p_i$$

Since π' was obtained by π without changing neither the capacity or the cost of the bins, nor the weights of the items or the profits of the non-compulsory items, it follows that the solution $\tilde{J}$, $\tilde{I}^{NC}_\pi$ is also feasible for π and therefore optimal. □

Note that even if conditions (6) and (7) hold, the GBPP remains $NP-hard$ as established in the following proposition.

Proposition 4 *The* GBPP *is strongly $NP-hard$ even if conditions (6) and (7) hold.*

Proof Consider the particular case of the GBPP where $|I^C|$ bins are available having the same capacity W and cost 1 (thus satisfying condition (7)) and all the items are compulsory (i.e., $I^{NC} = \emptyset$) with volume less than or equal W (thus condition (6) is also satisfied). In this case the objective function (5) reduces straightforward to the number of bins used. Then this particular case turns out to be the BPP which is *strongly $NP-hard$* (see Jansen et al. (2010)). □

3 On the approximability of the GBPP

Given a minimization problem Π, an instance $\pi \in \Pi$ of the problem, an algorithm ALG, the optimum $\mathrm{OPT}(\pi) \geq 0$ and the value $\mathrm{ALG}(\pi)$ of the solution computed by the algorithm, the worst-case ratio of algorithm ALG is the smallest $\alpha \geq 0$ such that, for each instance of the problem, the following inequality holds:

$$\mathrm{ALG}(\pi) \leq \alpha \cdot \mathrm{OPT}(\pi), \qquad \forall\, \pi \in \Pi \tag{8}$$

We wish to point out how inequality (8) requires that the optimum is always non-negative. Thanks to Definition 2 this is now guaranteed. Hereafter Δ denotes the non-negative offset given by the sum of the profits of all the non-compulsory items introduced in Definition 2.

Theorem 1 *The* GBPP *cannot be approximated for any constant* $\varepsilon > 1$*, unless* $P = NP$*, even if conditions (6) and (7) hold.*

Proof Note that in absence of conditions (6) and (7), the result is trivial. Indeed if condition (6) does not hold, the GBPP cannot be approximated since it is infeasible. If condition (7) does not hold we do not know *a priori* if the problem is feasible and the feasibility cannot be detected in polynomial-time according to Proposition 1. Hence, the problem can neither be approximated for any constant $\varepsilon > 1$, unless $P = NP$. While if conditions (6) and (7) hold, the proof is based on a reduction from the BPP.

Consider an instance π of the BPP in its decisional version i.e. n items, each one with volume w_i for $i = 1,\ldots,n$, and an unlimited number of bins with capacity W where $W \geq w_i$ forall $i = 1,\ldots,n$. The decisional version of the BPP asks for an assignment of all the items to the bins in such a way that at most k bins are used. The instance π can be transformed into an instance $\tilde{\pi}$ of the GBPP considering n compulsory items, each one with volume w_i for $i = 1,\ldots,n$, k bins with capacity W and cost 1, and n bins with capacity W and cost $k\varepsilon$. We notice that $\tilde{\pi}$ satisfies conditions (6) and (7). We observe that if π is a YES-answer instance of the BPP then the optimal value of the GBPP will be $\leq k$, otherwise if it is a NO-answer instance the GBPP optimal value will be $> k\varepsilon$ (since in the latter case at least one bin with cost $k\varepsilon$ must be used).

If, by contradiction, a polynomial time algorithm approximating the GBPP with a constant $\varepsilon > 1$ existed, then through such an algorithm we could discriminate, also in polynomial time, the YES-answer instances of the BPP from the NO-answer instances of the BPP. In fact the algorithm would return value $\leq k\varepsilon$ for the instances of the GBPP corresponding to the YES-answer instances of the BPP and value $> k\varepsilon$ for those ones corresponding to the NO-answer instances of the BPP. Unless $P = NP$, this is impossible since the decisional version of the BPP is $NP-complete$. □

Note that in the proof of Theorem 1 we do not use non-compulsory items (which are a peculiar feature of the GBPP), but we exploit the fact that the bins have different costs and just a limited number of them is available for at least one bin type. Therefore we also state the following corollary.

Corollary 1 *The Variable Cost and Size Bin Packing Problem with a Limited Number of Bins (*VCSBPPLNB*) cannot be approximated for any constant* $\varepsilon > 1$*, unless* $P = NP$*.*

Note that this result is not in contradiction with Epstein and Levin (2008), where an approximating algorithm is detected for the (classical) VCSBPP, since the VCSBPPLNB represents a generalization of it. In fact the classical VCSBPP (where the number of available bins is unlimited) is a particular case of the VCSBPPLNB where the availability U_t of each bin type $t \in T$ is equal to the number of items $|I|$.

The proof of Theorem 1 exploits the presence of two bin types. Therefore it may not hold for the particular case of the GBPP with one bin type. In this case, in the following proposition we prove that, if in addition all the items are non-compulsory, the GBPP becomes approximable.

Lemma 1 *The* GBPP *with one bin type and all non-compulsory items can be approximated with absolute worst-case ratio equal to 1.5.*

Proof The proof is based on the fact that in this case the GBPP can be transformed into a Bin Packing Problem with Rejection (BPR) for which Epstein and Levin (2010) stated an approximation algorithm with absolute worst-case ratio equal to 1.5. To show this, note that (5) can be rewritten as follows:

$$\sum_{j\in\tilde{J}} C - \sum_{i\in\tilde{I}^{NC}} p_i + \sum_{i\in I^{\mathrm{NC}}} p_i = C|\tilde{J}| + \sum_{i\in I^{\mathrm{NC}}\setminus\tilde{I}^{NC}} p_i. \tag{9}$$

where C denotes the cost of the only one bin type. But $I^{\mathrm{NC}} \setminus \tilde{I}^{NC}$ is the set of the unloaded items, therefore the minimization of (9) consists in minimizing the number of bins used and the penalties of the rejected items. This problem is clearly a BPR. □

Theorem 2 *Every feasible instance of the* GBPP *with one bin type can be approximated with absolute worst-case ratio equal to 1.5.*

Proof Thanks to Proposition 3, the instance can be transformed into an equivalent instance π' of the GBPP with all the items non-compulsory, preserving the objective function value of every feasible solution. Thanks to Lemma 1 the approximation result holds for π', hence it must hold also for the original instance. □

Note that the feasibility requested by Theorem 2 can be guaranteed for example through conditions (6) and (7).

Vice versa, some classical and widespread heuristics for the BPP (namely the FF and the BF) fail in approximating the GBPP, even with just one bin type. Further details are provided in the Appendix.

4 Solving the GBPP via dynamic programming

Theorem 3 *The* GBPP *satisfies Bellman's principle of optimality.*

Proof We recall that a problem Π satisfies the Bellman's principle of optimality if, given an instance $\pi \in \Pi$, any subset $S_1^*(\pi)$ of an optimal solution $S^*(\pi)$ is also optimal for the instance obtained removing from π the elements already used by $S^*(\pi) \setminus S_1^*(\pi)$. Therefore in the case of the GBPP, given an optimal solution $\tilde{J} \subseteq J$, $\tilde{I}^{NC} \subseteq I^{NC}$ and given any subset $\tilde{J}_1 \subseteq \tilde{J}$, we want to prove that for the instance made up by the compulsory items belonging to $\tilde{J}_1$ (i.e., $I^C(\tilde{J}_1) = \{i \in I^C : \beta(i) \in \tilde{J}_1\}$), by the non-compulsory items not contained in $\tilde{J} \setminus \tilde{J}_1$ (i.e., $I^{NC} \setminus I(\tilde{J} \setminus \tilde{J}_1)$) and by the bins not in $\tilde{J} \setminus \tilde{J}_1$, a solution better than $\tilde{J}_1, I^{NC}(\tilde{J}_1)$ cannot be built (where $I^{NC}(\tilde{J}_1) = \{i \in I^{NC} : \beta(i) \in \tilde{J}_1\}$). Let us suppose, by contradiction, that such a better solution exists i.e., $\hat{J} \subseteq (J \setminus \tilde{J}) \cup \tilde{J}_1$ and $\hat{I} \subseteq I^C(\tilde{J}_1) \cup I^{NC} \setminus I(\tilde{J} \setminus \tilde{J}_1)$ such that

$$\sum_{j \in \hat{J}} C_j - \sum_{i \in \hat{I} \cap I^{NC}} p_i + \sum_{i \in I^{NC} \setminus I(\tilde{J} \setminus \tilde{J}_1)} p_i < \sum_{j \in \tilde{J}_1} C_j - \sum_{i \in I^{NC}(\tilde{J}_1)} p_i + \sum_{i \in I^{NC} \setminus I(\tilde{J} \setminus \tilde{J}_1)} p_i \tag{10}$$

that is equivalent to

$$\sum_{j \in \hat{J}} C_j - \sum_{i \in \hat{I} \cap I^{NC}} p_i < \sum_{j \in \tilde{J}_1} C_j - \sum_{i \in I^{NC}(\tilde{J}_1)} p_i \tag{11}$$

In this case the bin set $\hat{J} \cup (\tilde{J} \setminus \tilde{J}_1)$ with the item set $(\hat{I} \cup I(\tilde{J} \setminus \tilde{J}_1)) \cap I^{NC}$ would be a feasible solution for the whole original instance with value less than the optimal one. Indeed its objective value would be

$$\begin{gathered}
\sum_{j \in \hat{J} \cup (\tilde{J} \setminus \tilde{J}_1)} C_j - \sum_{i \in (\hat{I} \cup I(\tilde{J} \setminus \tilde{J}_1)) \cap I^{NC}} p_i + \sum_{i \in I^{NC}} p_i = \\
\sum_{j \in \hat{J}} C_j - \sum_{i \in \hat{I} \cap I^{NC}} p_i + \sum_{j \in \tilde{J} \setminus \tilde{J}_1} C_j - \sum_{i \in I(\tilde{J} \setminus \tilde{J}_1) \cap I^{NC}} p_i + \sum_{i \in I^{NC}} p_i < \\
\sum_{j \in \tilde{J}_1} C_j - \sum_{i \in I^{NC}(\tilde{J}_1)} p_i + \sum_{j \in \tilde{J} \setminus \tilde{J}_1} C_j - \sum_{i \in I(\tilde{J} \setminus \tilde{J}_1) \cap I^{NC}} p_i + \sum_{i \in I^{NC}} p_i = \\
\sum_{j \in \tilde{J}} C_j - \sum_{i \in I(\tilde{J}) \cap I^{NC}} p_i + \sum_{i \in I^{NC}} p_i
\end{gathered}$$

where the inequality is due to (11).

But this implies that $\tilde{J} \subseteq J$, $\tilde{I}^{NC} \subseteq I^{NC}$ is not an optimal solution, which is clearly a contradiction. □

Since we proved that the GBPP satisfies Bellman's optimality principle, it follows that it can be addressed through dynamic programming, like for the BPP (see Jansen et al. (2010)). Let us consider the particular case where k types of items and $|T|$ types of bins are present. Without loss of generality we can assume that all the items are non-compulsory (Proposition 3). We can represent any instance of this type through the vector $(i_1, \ldots, i_k, b_1, \ldots, b_{|T|})$ meaning that there are i_h items of the type h for $h = 1, \ldots, k$, and b_t bins of the type t for $t = 1, \ldots, |T|$. Moreover p_h and w_h indicate the profit and the weight of each item of type h, respectively. In analogous way, let C_t and W_t denote the cost and the capacity of each bin of type t. Let us suppose that we want to solve the instance of the GBPP given by the vector

$(n_1,\dots,n_k,m_1,\dots,m_{|T|})$. Initially for each bin type we consider every combination of items that can be profitably packed into one bin of that type, i.e., every combination of items with total profit greater than the bin cost C_t and total weight not greater than W_t. Let us indicate this set of feasible solutions as $Q(t)$. Note that the set $Q(t)$ can be computed by exhaustive search with a complexity $O(n^k)$. Let us denote the optimal value for the instance $(q_1,\dots,q_k,\mathbf{e_t})$ with $OPT(q_1,\dots,q_k,\mathbf{e_t})$, for each $t \in T$ and $(q_1,\dots,q_k) \in Q(t)$, where $\mathbf{e_t}$ indicates the vector belonging to $\{0,1\}^{|T|}$ with value 1 in position t and 0 elsewhere. Clearly, $OPT(q_1,\dots,q_k,\mathbf{e_t}) = \min\{0, C_t - \sum_{i=1}^{k} q_i p_i\}$ where the minimum with 0 allows rejection of the items if their overall profit does not warrant the use of the bin.[1]

The recursive step of the dynamic programming algorithm is

$$OPT(i_1,\dots,i_k,b_1,\dots,b_{|T|}) = \min_{t\in T, q\in Q(t)} \left(OPT(i_1-q_1,\dots,i_k-q_k,\mathbf{b}-\mathbf{e_t}) + C_t - \sum_{i=1}^{k} q_i p_i\right)$$

In the end this gives us the value $OPT(n_1,\dots,n_k,m_1,\dots,m_{|T|})$ which is the optimal value of the GBPP. The detailed pseudocode of the dynamic programming algorithm for the GBPP is given by Algorithm 1. It is easy to see that the computational complexity of the algorithm is $O(|I|^k|J|^{|T|})$, therefore pseudo-polynomial.

Algorithm 1 Pseudocode for the dynamic programming algorithm for the GBPP

$OPT(i_1,\dots,i_k,b_1,\dots,b_{|T|}) = +\infty\ \ \forall i_h = 0,\dots,n_h,\ \forall h = 1,\dots,k,\ \ \forall b_t = 0,\dots,m_t,\ \forall t = 1,\dots,|T|$; ▷ Initialization
$OPT(i_1,\dots,i_k,0,\dots,0) = 0\ \ \forall i_h = 0,\dots,n_h,\ \forall h = 1,\dots,k$; ▷ Case of no bin available
$OPT(0,\dots,0,b_1,\dots,b_{|T|}) = 0\ \ \forall b_t = 0,\dots,m_t,\ \forall t = 1,\dots,|T|$; ▷ Case of no item available
$OPT(q_1,\dots,q_k,\mathbf{e_t}) = \min\{0, C_t - \sum_{i=1}^{k} q_i p_i\}\ \ \forall t \in T$ and $(q_1,\dots,q_k) \in Q(t)$; ▷ Case of one bin available
for $b_1 = 0,\dots,m_1$ **do**
  for $b_2 = 0,\dots,m_2$ **do**
    ...
    for $b_T = 0,\dots,m_{|T|}$ **do**
      for $i_1 = 0,\dots,n_1$ **do**
        for $i_2 = 0,\dots,n_2$ **do**
          ...
          for $i_k = 0,\dots,n_k$ **do**
            $M = +\infty$;
            for each $t \in T, (q_1,\dots,q_k) \in Q(t)$ such that $i_h \geq q_h\ \forall h = 1,\dots,k$ and $(i_1,\dots,i_k) \neq (q_1,\dots,q_k)$ and $b_t \geq 1$ **do**
              if $M > OPT(i_1-q_1,\dots,i_k-q_k,\mathbf{b}-\mathbf{e_t}) + OPT(q_1,\dots,q_k,\mathbf{e_t})$ **then**
                $M = OPT(i_1-q_1,\dots,i_k-q_k,\mathbf{b}-\mathbf{e_t}) + OPT(q_1,\dots,q_k,\mathbf{e_t})$;
              end if
            end for each
            $OPT(i_1,\dots,i_k,b_1,\dots,b_{|T|}) = \min\{M, OPT(i_1,\dots,i_k,b_1,\dots,b_{|T|})\}$;
          end for
        end for
      end for
    end for
  end for
end for

5 Conclusions

In this paper we studied the computational complexity and the approximability of the Generalized Bin Packing Problem (GBPP), a novel packing problem arising in many transportation and logistic settings. The definition of the worst-case ratio requires the objective function of the studied problem to be non-negative. This is not the case of the GBPP, in which the difference between the costs of the bins used and the profits of the selected non-compulsory items is minimized. This difference, in fact, can lead to negative objective function values. In order to overcome this issue, we proposed an equivalent definition of the GBPP which always provides non-negative objective function values.

[1] For sake of simplicity we are now using Definition 1 rather than Definition 2.

We proved that the GBPP is *NP-hard*, and the problem of finding a feasible solution (if it exists) is *NP-complete*. In particular this problem can be reduced to the solution of a GBPP with all non-compulsory items.

We also showed a very significant result: the GBPP cannot be approximated, unless $P = NP$. We also extended the proof to the Variable Cost and Size Bin Packing Problem with a Limited Number of Bins. Since our proof exploited the presence of multiple bin types, we studied the particular case of a single bin type, separately. We showed how this case reduces to the Bin Packing with Rejections (BPR), which is approximable with an absolute worst-case ratio equal to 1.5. We also proved that the classical FF and BF heuristics cannot approximate the problem, unlike the BPP, because their worst-case ratio can be arbitrarily large, even for the variant which rejects non-profitable bins.

Finally, we proved that the GBPP satisfies the Bellman's optimality principle. Exploiting this result, we developed a dynamic programming solution approach with pseudo-polynomial computational complexity $O(|I|^k|J|^{|T|})$, being k and $|T|$ the number of item types and bin types, respectively.

Appendix

In the following propositions we prove that some classical and widespread heuristics for the BPP (namely the FF and the BF) fail in approximating the GBPP, even with just one bin type.

The FF and the BF algorithms were proposed by Johnson et al. (1974) and are defined as follows. Given a list of items IL and a list of available bins $BL \subseteq J$, the FF tries to place an item of the list into the *first* bin in BL able to contain it. That is, the residual space of that bin must be at least the volume of the item. The process continues until all items have been analyzed. Afterwards, all empty bins are removed from BL.

The BF works like the FF with the only difference that it tries to place an item into the *best* bin. The *best* bin is defined as follows: it must be able to contain the item and the residual space after loading the item must be minimum.

A pseudo-code for the FF and BF algorithms is proposed in Algorithm 2.

In Proposition 5 we prove that FF and BF cannot approximate the GBPP.

Proposition 5 *Algorithms* FF *and* BF *cannot approximate the* GBPP*, even if all the bins are of one type.*

Proof Consider an instance π of the GBPP with one bin type, with capacity W and cost C, and $n > 0$ non-compulsory items, with volume W and positive profit $\varepsilon < C$. The offset Δ is given by the sum of all non-compulsory items, i. e., $\Delta = n\varepsilon$. Clearly, $\text{OPT}(\pi) = \Delta + 0 = n\varepsilon$, while algorithm XF loads all the non-compulsory items. Therefore, $\text{XF}(\pi) = n\varepsilon + n \cdot (C - \varepsilon) = nC$. Substituting these values into (8), we have $nC \leq \alpha \cdot n\varepsilon$, which implies $\alpha \geq \frac{C}{\varepsilon}$. This means, however, that α can be arbitrarily large, and the proposition holds. □

In contrast to the classical Bin Packing Problem, the FF and BF heuristics can lead to unprofitable bins when applied to the GBPP. Consider, for example, an instance π of the GBPP with one bin type with capacity W and cost C and two non-compulsory items with volume W and profits $p_1 = C + \varepsilon$ and $p_2 = C - \varepsilon$, with $\varepsilon < C$. The optimum solution consists in loading item 1 and discarding item 2. Nevertheless, both the FF and the BF load both the items into two bins. The bin containing item 2 is not profitable because $p_2 < C$. Therefore, a spontaneous extension of the FF and BF consists in rejecting all the non-profitable bins (together with its items) and without compulsory items, at the end of the process. In fact, we cannot unload compulsory items. We name these variants of the FF and BF respectively as FIRST FIT WITH REJECTION (FFR) and BEST FIT WITH REJECTION (BFR). A pseudo-code for the FFR and BFR algorithms is proposed in Algorithm 2. Note that, if heuristics FFR and BFR are applied to instance π, the bin containing item 2 is rejected and the resulting solution is optimal. Unfortunately this improvement is not enough to approximate the GBPP. This is proved in Proposition 6.

Proposition 6 *Algorithms* FFR *and* BFR *cannot approximate the* GBPP*, even if all the bins are of one type.*

Proof Consider an instance π of the GBPP with one bin type, with capacity W and cost C, and the set of items I is split into two subsets, A and B, with $|A| = |B| = k$. An item which belongs to subset $X \in \{A, B\}$ is called a type X item. Let type A items be non-compulsory with $w_A = \frac{W}{k}$ and $p_A = C - \varepsilon$, type B items be non-compulsory with $w_B = \frac{k-1}{k}W$, $p_B = \frac{\varepsilon}{2}$, with $\varepsilon > 0$ small enough. Moreover, let XFR be either the FFR or the BFR algorithm. The offset Δ is given by the sum of the profits of all the non-compulsory items: $\Delta = kp_A + kp_B = k \cdot (p_A + p_B) = k \cdot \left(C - \frac{\varepsilon}{2}\right)$.

It is easy to see that type B items do not appear in the optimal solution because they are not profitable. In fact, they are very large but their profit is very small. Therefore, the optimal solution consists of one bin accommodating all the type A items: $\text{OPT}(\pi) = \Delta + C - kp_A = C + k\frac{\varepsilon}{2}$.

Let us denote by i_X an item which belongs to the subset X and consider the following sequence of items.

$$\overbrace{i_A \quad i_B \quad \dots \quad i_A \quad i_B}^{k \text{ times}}$$

If we apply algorithm XFR to this sequence of items, we get k bins, each containing one type A and one type B item. These items will be discarded because $p_A + p_B = C - \varepsilon + \frac{\varepsilon}{2} = C - \frac{\varepsilon}{2} < C$. Therefore, $\text{XFR}(\pi) = \Delta + 0 = k \cdot \left(C - \frac{\varepsilon}{2}\right)$. Applying definition 8 of the worst-case ratio, we have $k \cdot \left(C - \frac{\varepsilon}{2}\right) \leq \alpha \cdot (C + k)$, which implies (as the right side of the inequality is positive) $\alpha \geq \frac{(2C-\varepsilon)k}{2C+k\varepsilon}$. Note that $\lim_{\varepsilon \to 0} \alpha = \frac{2Ck}{2C} = k$, which can be arbitrarily large and the proposition holds. □

Algorithm 2 Pseudo-code of the proposed heuristics

```
for items i ∈ IL do
    Select from BL the bin b ∈ S where to load item i:
  – FF, FFR: the first bin with enough residual space to accommodate item i
  – BF, BFR: the bin with the minimum free volume which is able to accommodate item i
    if b exists then
        load item i into bin b
    else
        skip item i
    end if
end for
Remove all empty bins from BL
FFR, BFR: Remove all non-profitable bins without compulsory items from BL
```

References

M. M. Baldi. Generalized bin packing problems. *4OR*, 12(3):293–294, 2014. doi: 10.1007/s10288-013-0252-1.

M. M. Baldi, T. G. Crainic, G. Perboli, and R. Tadei. The generalized bin packing problem. *Transportation Research Part E*, 48(6):1205–1220, 2012. doi: 10.1016/j.tre.2012.06.005.

M. M. Baldi, T. G. Crainic, G. Perboli, and R. Tadei. Asymptotic results for the generalized bin packing problem. *Procedia - Social and Behavioral Sciences*, 111:663–671, 2013. doi: 10.1016/j.sbspro.2014.01.100.

M. M. Baldi, T. G. Crainic, G. Perboli, and R. 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(1): 125–141, 2014. doi: 10.1007/s10479-012-1283-2.

W. Bein, J. R. Correa, and X. Han. A fast asymptotic approximation scheme for bin packing with rejection. *Theoretical Computer Science*, 393:14–22, 2008.

I. Correia, L. Gouveia, and F. Saldanha-da-Gama. Solving the variable size bin packing problem with discretized formulations. *Computers & Operations Research*, 35:2103–2113, 2008.

T. G. Crainic, G. Perboli, W. Rei, and R. Tadei. Efficient lower bounds and heuristics for the variable cost and size bin packing problem. *Computers & Operations Research*, 38:1474–1482, 2011.

G. Dósa and Y. He. Bin packing problems with rejection penalties and their dual problems. *Information and Computation*, 204(5):795–815, 2006.

L. Epstein. Bin packing with rejection revisited. *Algorithmica*, 56(4):505–528, 2010.

L. Epstein and A. Levin. An aptas for generalized cost variable-sized bin packing. *SIAM Journal on Computing*, 38(1):411–428, 2008.

L. Epstein and A. Levin. Afptas results for common variants of bin packing: A new method for handling the small items. *SIAM Journal on Optimization*, 20(6):3121–3145, 2010.

D. K. Friesen and M. A. Langston. Variable sized bin packing. *SIAM Journal on Computing*, 15:222–230, 1986.

M. R. Garey, R. L. Graham, and J. D. Ullman. Worst-case analysis of memory allocation algorithms. In *Proceedings of the fourth annual ACM symposium on Theory of computing*, STOC '72, pages 143–150, New York, NY, USA, 1972.

K. Jansen, S. Kratsch, D. Marx, and I. Schlotter. Bin packing with fixed number of bins revisited. In *Algorithm Theory - SWAT 2010*, volume 6139, pages 260–272. Springer Berlin Heidelberg, 2010.

D. S. Johnson, A. Demeters, J. D. Hullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3:299–325, 1974.

J. D. Ullman. The performance of a memory allocation algorithm. Technical Report 100, Princeton University, 1971.