Constant-Ratio Approximation for Robust Bin Packing with Budgeted Uncertainty

Marin Bougeret, György Dósa, Noam Goldberg, Michael Poss

CONSTANT-RATIO APPROXIMATION FOR ROBUST BIN PACKING WITH BUDGETED UNCERTAINTY[*]

MARIN BOUGERET[†], GYÖRGY DÓSA[‡], NOAM GOLDBERG[§], AND MICHAEL POSS[¶]

Abstract. We consider robust variants of the bin packing problem with uncertain item sizes. Specifically we consider two uncertainty sets previously studied in the literature: budgeted uncertainty (the U^Γ model) in which at most Γ items deviate, each reaching its peak value, while other items assume their nominal values. The second uncertainty set, the U^Ω model, bounds the total amount of deviation in each scenario. We show that a variant of the next-fit-decreasing algorithm is a 2 approximation for the U^Ω model, and another variant of this algorithm is a 2Γ approximation for the U^Γ model. Unlike the classical bin packing problem, it is shown that (unless $\mathcal{P} = \mathcal{NP}$) no asymptotic approximation scheme exists for the U^Γ model, already for $\Gamma = 1$. This motivates the question of the existence of a constant approximation factor algorithm for the U^Γ model. Our main result is to answer this question by proving a (polynomial-time) 4.5 approximation algorithm, based on a dynamic-programming approach.

Key words. Bin-packing, robust optimization, approximation algorithms, Next-fit-decreasing, dynamic programming

1. Introduction. This paper studies approximation algorithms for a generalization of the bin packing problem that requires its solutions to be robust with respect to uncertain input data. In general, approximation algorithm bound results may involve asymptotic or absolute approximation ratio guarantees whose definition can be stated as follows: An algorithm has an asymptotic approximation ratio ρ for a minimization problem if and only if there exists a constant c such that it outputs a solution of value that is at most $\rho\mathsf{OPT}(I) + c$, for every instance input I. The definition of an absolute approximation ratio ρ is similar with the same upper bound only that $c = 0$.

Bin packing is the problem of assigning a given set of n items, each item of a specified size, to the smallest number of unit capacity bins. The problem has been the subject of study in an extensive body of research initiated by several publications in the 1970s including the work of Johnson et al. [20]. The problem is $\mathcal{NP}$-hard and in fact a straightforward reduction from the partition decision problem implies that it is $\mathcal{NP}$-hard to determine whether a bin packing instance has a solution using only two bins. This also shows that the problem cannot be approximated within a factor less than $3/2$ unless $\mathcal{P} = \mathcal{NP}$. An absolute approximation factor guarantee of $3/2$ has been proven for the first-fit decreasing algorithm [24]. Several results concern asymptotic setting of classical bin packing. It is known since the problem has been first analyzed that the first-fit decreasing algorithm has an asymptotic approximation ratio of $\frac{11}{9}$ [20]. This result has also been extended to analyze the constant for first-fit decreasing, in particular it turns out that $\frac{11}{9}\mathsf{OPT}(I) + \frac{6}{9}$ is a tight upper bound (so $c = \frac{6}{9}$) [15]. A well known result is the asymptotic fully polynomial-time approximation scheme for bin packing established by [21], following the approximation scheme first proposed by [13].

[*]A preliminary version of this work has been published in [6].

Funding: This research has benefited from the support of the ANR project ROBUST [ANR-16-CE40-0018].

[†]LIRMM, University of Montpellier, CNRS, Montpellier, France (marin.bougeret@lirmm.fr)

[‡]Department of Mathematics, University of Pannonia, Veszprém, Hungary (dosagy@almos.vein.hu)

[§]Department of Management, Bar-Ilan University, Ramat Gan 5290002, Israel (noam.goldberg@biu.ac.il)

[¶]LIRMM, University of Montpellier, CNRS, Montpellier, France (michael.poss@lirmm.fr)

Another distinct line of studies, which may not include the focus of the current paper but has received significant attention in the literature, is that of bin packing online algorithms, a setting in which the instance is not given in advance but items are revealed and packed one at a time as they arrive (note that first-fit decreasing is not an online algorithm). In this context the performance measure, which bounds the ratio of the algorithm's solution objective value to the optimal offline solution objective value, is called the competitive ratio. The best asymptotic and absolute online competitive ratios of 1.578 and 5/3, respectively, have been shown in [3] and [5], respectively, while the best lower bound for the asymptotic competitive ratio of around 1.54278 has been provided in [4]. Approximation results in the offline and online settings have also been developed for different extensions and generalizations of the classical bin packing problem; for example see [12, 18] and references therein.

In many applications, the sizes of the items to be packed are not fully known at the time that the packing is carried out. In cargo shipping, for example, the actual weight of a container may deviate from its declared weight or its measurements may be inaccurate. Bin packing has been used to model the assignment of elective surgeries to operating room in hospitals [14]. Here a bin is a shift of a properly equipped and staffed operating room for performing a certain type of elective surgeries. The room scheduler has to fit in the bins as many cases (patients) as possible. In this setting clearly the length of time of performing each surgery is subject to uncertainty for example in the event of complications. Bin packing variants has also been used to model the scheduling of internet advertising; see for example [1]. When multiple commercials are packed into advertising breaks of online video services and each commercial may be skipped by the viewer then the length of each ad is subject to uncertainty.

One way to model the uncertainty that falls into the framework of robust optimization is to assume that the sizes are uncertain parameters taking any value in a given set $U \subset \mathbb{R}^n$, where each $a \in U$ represents a possible scenario. Modern robust optimization considers such sets U in continuous as well as combinatorial optimization, and most common are discrete, polyhedral, and ellipsoid uncertainty sets; see [7] for a comprehensive reference on continuous robust optimization and [11] as a survey of robust combinatorial optimization. Incorporating U in bin packing leads to the following problem (note that the description of U may not be explicit to avoid an exponential length in n).

Robust bin packing (RBP)
Input: $U \subset \mathbb{R}^n$
Output: A solution that is a partition of $[n]$ into k bins $b_1, \ldots, b_k$ such that $\max_{a \in U} \sum_{i \in b_j} a_i \leq 1$ for each $j \in [k]$
Minimize: k

In this paper we focus on two specific uncertainty sets, defined by simple budget constraints. One of these widely used uncertainty sets [8], U^Γ, supposes that the size of each item is either its given nominal size $\bar{a}_i$, or its peak value $\bar{a}_i + \hat{a}_i$. Furthermore, in any scenario, at most $\Gamma \in \mathbb{N}$ of the items may take their peak value simultaneously.

Formally, U^Γ can be defined as

$$U^\Gamma = \{a | \forall i \in [n], a_i \in \{\bar{a}_i, \bar{a}_i + \hat{a}_i\} \text{ and } \sum_{i \in [n]} (a_i - \bar{a}_i)/\hat{a}_i \leq \Gamma\}.^{1}$$

We also consider a second uncertainty set (used in [19, 22, 26], among others), characterized again by $\bar{a}$ and $\hat{a}$, as well as the number $\Omega \in [0,1]$ stating how much deviation can be spread among all sizes, formally $U^\Omega = \{a \in \times_{i\in[n]}[\bar{a}_i, \bar{a}_i + \hat{a}_i] \mid \sum_{i\in[n]}(a_i - \bar{a}_i) \leq \Omega\}$. The main purpose of this paper is to prove the following two theorems.

Theorem 1.1. *There exists a 2-approximation algorithm for* RBP *with uncertainty set* U^Ω.

Theorem 1.2. *There exists a* $\min(2\Gamma, 4.5)$*-approximation algorithm for* RBP *with uncertainty set* U^Γ.

1.1. Literature review. Uncertainty set U^Γ has been widely used in robust combinatorial optimization with a constant number of constraints because the set essentially preserves the complexity properties of the nominal problem, and to some extent, its approximability properties as well. While the results for this model were first developed for min-max problems and uncertain objective functions [8], they were later extended to uncertain constraints independently in [2, 17]. Our second uncertainty set (used in [19, 22, 26], among others), U^Ω benefits from similar positive results as U^Γ; see [23].

The above positive complexity results (e.g., [17, 23]) imply, for instance, that under mild assumptions there exists a fully-polynomial time approximation scheme (FPTAS) for the robust knapsack problem with uncertain profits and uncertain weights in U^Ω or U^Γ. Interestingly, these positive results may not be applied to most scheduling problems due to the presence of nonlinearities of the objective function in the uncertain data parameters. Also, these general results cannot be applied to the bin packing problem because it involves a non-constant numbers of robust constraints. Further, in the current paper it is proved that approximation results of standard bin packing in the form of asymptotic approximation scheme, cannot carry over (unless $\mathcal{P} = \mathcal{NP}$) to the robust counterpart with U^Γ. While in previous papers [10, 9] (with authors in common) we provided approximability results on robust scheduling, no such results have yet been proposed for the bin packing problem, the only previous work focusing on numerical algorithms [25]. The purpose of this paper is to fill these gaps, as we present constant-factor approximation algorithms the bin packing problem, both for U^Ω and U^Γ.

1.2. Notation, problems definitions, and next-fit algorithm. In this paper we consider two special cases of RBP. In the first one, ΓRBP, the input is $\mathcal{I} = (\overline{a}, \hat{a}, \Gamma) \in [0,1]^n \times [0,1]^n \times \mathbb{N}$ where $n \in \mathbb{N}$, and $U = U^\Gamma$. In the second one, ΩRBP, the input is $\mathcal{I} = (\overline{a}, \hat{a}, \Omega) \in [0,1]^n \times [0,1]^n \times [0,1]$ where $n \in \mathbb{N}$ and $U = U^\Omega$.

Let us now define some notation that may be required for formally stating the ΓRBP and ΩRBP problems. Given $n \in \mathbb{N}$, sets $\{0, 1, \dots, n\}$ and $\{1, \dots, n\}$ are respectively denoted $[n]_0$ and $[n]$. Set $\{i, \dots, j\}$ is denoted by $[\![i, j]\!]$. Given a vector $v \in [0,1]^n$ and a subset $X \subseteq [n]$, we define $v(X) = \sum_{i\in X} v_i$. Given two vectors $\overline{a} \in [0,1]^n$, $\hat{a} \in [0,1]^n$ and a subset of items $X \subseteq [n]$, we define $\hat{a}_\Omega(X) = \min\{\hat{a}(X), \Omega\}$,

[1] U^Γ is often defined alternatively in the literature, as the polytope $\{a \in \times_{i\in[n]}[\bar{a}_i, \bar{a}_i + \hat{a}_i] \mid \sum_{i\in[n]}(a_i - \bar{a}_i)/\hat{a}_i \leq \Gamma\}$. For the bin packing problem, one readily verifies using classical arguments that the two definitions lead to the same optimization problem.

$\Gamma(X)$ as the set of Γ items in X with largest $\hat{a}$ values (ties broken by taking smallest indices), or $\Gamma(X) = X$ if $|X| < \Gamma$, and $\hat{a}_\Gamma(X) = \hat{a}(\Gamma(X))$. Accordingly, we define the fill of a bin $b \subseteq [n]$ as $f_\Gamma(b) = \overline{a}(b) + \hat{a}_\Gamma(b)$ for set U^Γ, and $f_\Omega(b) = \overline{a}(b) + \hat{a}_\Omega(b)$ for set U^Ω. The fill of a bin for a general uncertainty set U is denoted as $f_U(b) = \max_{a \in U} a(b)$.

Consider the following example. We are given an ordered set of pairs $(\overline{a}_i, \hat{a}_i)$, $X = \{(0.3, 0.2), (0.4, 0.2), (0.3, 0.1), (0.2, 0.5)\}$ with $\Gamma = 2$ and $\Omega = 0.3$. Thus, $\Gamma(X) = \{(0.3, 0.2), (0.2, 0.5)\}$, $\overline{a}(X) = 1.2$, $\hat{a}_\Gamma(X) = 0.7$, and $f_\Gamma(X) = 1.9$. Similarly, $\hat{a}_\Omega(X) = 0.3$ and $f_\Omega(X) = 1.5$.

Now, observe that $\max_{a \in U} \sum_{i \in b_j} a_i \leq 1$ (the constraint required in RBP) is equivalent to $f_U(b) \leq 1$, and thus to $f_\Gamma(b_j) \leq 1$ for ΓRBP and $f_\Omega(b_j) \leq 1$ for ΩRBP. For example in ΓRBP, $f_\Gamma(b_j) \leq 1$ simply means that the total nominal ($\overline{a}$) size of the items plus the deviating size ($\hat{a}$) of the Γ largest (in $\hat{a}$ values) items must not exceed one. Then, the two optimization problems studied in this paper can be equivalently formulated as follows.

Γ-robust bin packing (ΓRBP)
Input: $\mathcal{I} = (\overline{a}, \hat{a}, \Gamma) \in [0,1]^n \times [0,1]^n \times \mathbb{N}$.
Output: A solution that is a partition of $[n]$ into k bins $b_1, \ldots, b_k$ such that $f_\Gamma(b_j) \leq 1$ for each $j \in [k]$
Minimize: k

Ω-robust bin packing (ΩRBP)
Input: $\mathcal{I} = (n, \overline{a}, \hat{a}, \Omega)$ where $n \in \mathbb{N}$, $\overline{a} \in [0,1]^n$, $\hat{a} \in [0,1]^n$, and $\Omega \in [0,1]$.
Output: A solution that is a partition of $[n]$ into k bins $b_1, \ldots, b_k$ such that $f_\Omega(b_j) \leq 1$ for each $j \in [k]$
Minimize: k

The optimal solution value or cost of either problem is denoted by $\mathsf{OPT}(\mathcal{I}) = k^*$ ($\mathcal{I}$ may be omitted when the instance is clear from the context) and a corresponding optimal solution is denoted by $s^* = \{b_1^*, b_2^*, \ldots, b_{k^*}^*\}$. We introduce in Algorithm 1 a variant of the next-fit-decreasing algorithm.

initialization: $j = 1$
1 Pack items (with smaller index first) in b_j until $f_U(b_j) > 1$ or $n \in b_j$. If $n \notin b_j$ then $j \leftarrow j + 1$ and repeat Step 1. Otherwise, $k' \leftarrow j$ proceed to Step 2.
2 Pack the last item of each bin in a new bin: for any j, let $i = \max(b_j)$, $b_j^1 = b_j \setminus \{i\}$, and $b_j^2 = \{i\}$
return : $\bigcup_{j=1}^{k'} \{b_j^1, b_j^2\}$

Algorithm 1: NEXT-FIT-DECREASING($\mathcal{I}$)

1.3. Structure of the paper. In Section 2 we prove that there is no asymptotic approximation for ΓRBP with a factor less than $3/2$ (unless $\mathcal{P} = \mathcal{NP}$). In Sections 3 and 4, we analyze approximation-factor guarantees for NEXT-FIT-DECREASING in the case of ΩRBP and ΓRBP, respectively. For ΩRBP, using ordering (3.1) (non-increasing ordering on $\frac{\hat{a}_i}{\overline{a}_i}$) the ratio is equal to 2. For ΓRBP, using ordering (4.1) (non-increasing ordering on $\hat{a}_i$), the ratio is bounded by 2Γ. As Theorem 4.5 shows that neither ordering (3.1) or (4.1) leads to a constant ratio using

Next-Fit-Decreasing, this raises the question of the existence of a constant approximation for ΓRBP. We answer the question in Section 5 by providing a dynamic programming algorithm (DP) giving a ratio of 4.5 for ΓRBP and any $\Gamma \in \mathbb{N}$, which is our main result.

2. Innaproximability. First we establish that as opposed to classical bin packing that has asymptotic approximation schemes [13, 21], such approximation schemes are not possible (unless $\mathcal{P} = \mathcal{NP}$) for ΓRBP.

Lemma 2.1. *Suppose $\Gamma = 1$. For any $K \geq 1$, it is $\mathcal{NP}$-complete to decide whether* ΓRBP *has a solution using $2K$ bins or whether at least $3K$ bins are required.*

Proof. Consider an instance I, of the $\mathcal{NP}$-Complete 2-partition decision problem [16], given by a set of distinct positive rational numbers whose sum is 2 (in particular each instance given by a set of N integers can be normalized to get such a set of rational numbers), so $I = \{s_1, s_2, ..., s_N\}$, where $s_1+s_2+...+s_N = 2$ and $0 \leq s_i \leq 1$ for $i \in [N]$. In particular, it is NP-complete to decide the question whether the given set of numbers can be partitioned into two subsets having a sum of one in each subset (that is to decide whether I is a "YES-instance"). Let $s_{\min} = \min_{i\in[N]}\{s_i\}$ and let $0 < \delta < s_{\min}$ (assume that δ is a rational number).

Now choose some (arbitrarily large) integer $K \geq 1$, and create K sets (each of size N) of bin packing items as follows. For $k \in [K]$, let $\varepsilon_k = \delta^k$. Elements of the K copies of the N numbers will be denoted by index set pairs in $[N] \times [K]$. The nominal and deviation sizes of the ΓRBP instance items are defined for $i \in [N]$ and $k \in [K]$ as

$$\overline{a}_{i,k} = \varepsilon_k \cdot s_i \qquad \text{and} \qquad \hat{a}_{i,k} = 1 - \varepsilon_k, \qquad \text{respectively.}$$

Note that for each k, $\sum_{i\in[N]} \overline{a}_{i,k} = 2\varepsilon_k$, and $\overline{a}_{i,k} \leq \varepsilon_k$ for each $i \in [N]$. Also note, that by construction, for all $i \in [N]$, $\overline{a}_{i,k} > \varepsilon_{k+1}$, for $1 \leq k < K$.

First it can be observed that if I is a YES-instance of the 2-partition problem, then the constructed ΓRBP instance $2K$ bins: K pairs of bins, where in each of the two bins of pair $k \in [K]$, the total nominal is ε_k and the deviation is $1 - \epsilon_k$.

Suppose now that I is a NO-instance of 2-partition. Let $b_1, \ldots, b_L$ be a ΓRBP solution for some $L > 1$. To show that $L \geq 3K$, first define for each $k \in [K]$, $I_k = \{(i,k), i \in [N]\}$ to be referred to as the set of items of type k, and observe that no bin can contain items of two different types. Indeed, suppose by contradiction that for some $k, k' \in [K]$ satisfying $k \neq k'$ (say $k > k'$), and $i, i' \in [N]$, item (i,k) is packed with item (i',k'). This implies $\overline{a}_{i,k} + \overline{a}_{i',k'} + \max\{\hat{a}_{i,k}, \hat{a}_{i',k'}\} \geq \overline{a}_{i,k} + \hat{a}_{i',k'} > \varepsilon_{k'} + (1 - \varepsilon_{k'})$, a contradiction. Thus, none of the bins $b_1, \ldots, b_l$ contain items of two different types, that is for each $l \in [L]$, there exists some $k \in [K]$ such that $b_l \subseteq I_k$. Since I is a NO-instance, each I_k is packed in at least 3 distinct bins out of $b_1, \ldots, b_L$, implying $L \geq 3K$. ∎

The following theorem is immediate from Lemma 2.1.

Theorem 2.2. *Unless $\mathcal{P} = \mathcal{NP}$ there is no asymptotic approximation for* ΓRBP *of ratio $\rho < 3/2$.*

Note that this result is in contrast with the approximation schemes that exist for classical bin packing [21, 13]. Furthermore, recall that in contrast to Theorem 2.2, for the classical bin packing, the first-fit decreasing algorithm packs the items of any input I into at most $\lfloor \frac{11}{9} OPT(I) + 6/9 \rfloor$ bins [15].

3. Next-fit-decreasing for ΩRBP. Unlike the classical bin packing problem, executing Next-Fit-Decreasing on arbitrarily ordered items can lead to arbitrarily bad solutions. For example, given ϵ with $0 < \epsilon \leq \frac{1}{2n}$, consider an instance with $\Omega = 1 - \epsilon$, and items $((2\epsilon, 0), (0, 1-\epsilon), \ldots, (2\epsilon, 0), (0, 1-\epsilon))$, where item $i \in [n]$ is denoted by the pair $(\bar{a}_i, \hat{a}_i)$. Using this ordering, Next-Fit-Decreasing will create $n/2$ bins b_j with $f_\Omega(b_j) > 1$ for any $j \in [n]$ (which will be turned into n bins $\{b_j^1, b_j^2\}$), whereas the optimal solution uses 2 bins. This example also illustrates that, unlike in the standard bin packing, the total size argument no longer apply to the robust counterpart as having $f_\Omega(b_j) > 1$ for any $j \in [n]$ does not imply a large (depending on n) lower bound on the optimum.

Next, we consider an ordering of the items such that

$$\hat{a}_1/\bar{a}_1 \geq \cdots \geq \hat{a}_n/\bar{a}_n, \tag{3.1}$$

and recall that k' is the number of bins opened in Step 1 of Next-Fit-Decreasing.

Lemma 3.1. *Suppose that the items are ordered according to* (3.1)*. Then* $k' \leq k^*$*.*

Proof. Consider an optimal solution $b_1^*, \ldots, b_{k^*}^*$ and the subset of optimal bins given by $G^* = \{j \in [k^*] \mid \hat{a}(b_j^*) > \Omega\}$. Let

$$A = \sum_{i \in [n]} (\bar{a}_i + \hat{a}_i) = \sum_{j \in [k']} (\bar{a}(b_j) + \hat{a}(b_j)) = \sum_{j \in [k^*]} \left(\bar{a}(b_j^*) + \hat{a}(b_j^*)\right).$$

Let G denote the first $|G^*|$ bins opened in Step 1 of Next-Fit-Decreasing. If $k' \in G$ then clearly $k' \leq k^*$. Otherwise, it can be observed that for each $l \in G$, $\bar{a}(b_l) > 1 - \Omega$ (as $\bar{a}(b_\ell) + \hat{a}_\Omega(b_\ell) > 1$ and $\hat{a}_\Omega(b_\ell) \leq \Omega$) and $1 - \Omega \geq \max_{j \in G^*} \bar{a}(b_j^*)$ (as $f_\Omega(b_j^*) \leq 1$). Thus, $\sum_{j \in G} \bar{a}(b_j) > \sum_{j \in G^*} \bar{a}(b_j^*)$ and so by the assumed ordering (3.1) of the items, following a standard knapsack argument, $\sum_{j \in G} \hat{a}(b_j) > \sum_{j \in G^*} \hat{a}(b_j^*)$. Letting $\bar{G} = [k'] \setminus G$ and $\bar{G}^* = [k^*] \setminus G^*$, it follows that

$$\sum_{j \in \bar{G}} (\bar{a}(b_j) + \hat{a}(b_j)) = A - \sum_{j \in G} (\bar{a}(b_j) + \hat{a}(b_j)) \leq$$
$$A - \sum_{j \in G^*} \left(a(b_j^*) + \hat{a}(b_j^*)\right) = \sum_{j \in \bar{G}^*} \left(\bar{a}(b_j^*) + \hat{a}(b_j^*)\right)$$

(equality may hold throughout if $G^* = \emptyset$). Further, for each $j \in \bar{G} \setminus \{k'\}$, $\bar{a}(b_j) + \hat{a}(b_j) \geq f_\Omega(b_j) > 1$ and for each $j \in \bar{G}^*$, $\bar{a}(b_j^*) + \hat{a}(b_j^*) \leq 1$. Therefore, $|\bar{G}| \leq \left\lceil \sum_{j \in \bar{G}} (\bar{a}(b_j) + \hat{a}(b_j)) \right\rceil \leq \left\lceil \sum_{j \in \bar{G}^*} \left(\bar{a}(b_j^*) + \hat{a}(b_j^*)\right) \right\rceil \leq |\bar{G}^*|$ and $k' \leq k^*$ as claimed. □

The lemma combined with Step 2 of Next-Fit-Decreasing immediately imply the following theorem.

Theorem 3.2. *If the items are ordered according to* (3.1) *then* Next-Fit-Decreasing *is a 2-approximation algorithm for* ΩRBP*.*

4. Next-fit-decreasing for ΓRBP. From now on, we focus on problem ΓRBP. Remark first that using an arbitrary ordering leads to arbitrarily bad solutions, considering $\Gamma = 1$ and the same items $((2\epsilon, 0), (0, 1-\epsilon), \ldots, (2\epsilon, 0), (0, 1-\epsilon))$ as in the previous section. Thus, we consider here an ordering of the items such that

$$\hat{a}_1 \geq \cdots \geq \hat{a}_n. \tag{4.1}$$

The main result of this section is the following. We also note that this result has been improved compared with the result that appears in the preliminary extended abstract version of this paper [6].

THEOREM 4.1. *Suppose that the items are ordered according to* (4.1)*. Then* NEXT-FIT-DECREASING *is a* 2Γ*-approximation algorithm for* ΓRBP*.*

Recall that k' is the number of bins used in Step 1 and let $s' = (b_1, \dots, b_{k'})$ be the bins output at the end of Step 1. Let $s^* = \{b_1^*, \dots, b_{k^*}^*\}$ be an optimal solution. Define $i_j^* = \max(\Gamma(b_j^*))$, and also let $i_j = \max(\Gamma(b_j))$ for each $j \in [k']$.

The key element in proving Theorem 4.1 is the following counterpart of Lemma 3.1, a result which immediately implies an approximation-factor guarantee of 2Γ.

LEMMA 4.2. *Suppose that the items are ordered according to* (4.1)*. Then* $k' \leq \Gamma k^*$*.*

Let $\mathcal{M}' \subseteq s'$ be the set of bins that contain only items that either do not deviate or are the smallest deviations, i_j^* for some $j \in [k^*]$. So for each $b \in \mathcal{M}'$, $b \subseteq [n] \setminus \bigcup_{j=1}^{k^*} \Gamma(b_j^*) \cup \{i_1^*, \dots, i_{k^*}^*\}$. In what follows, we bound $|\mathcal{M}'|$ and $|s' \setminus \mathcal{M}'|$ by multiples of k^*.

LEMMA 4.3. $|s' \setminus \mathcal{M}'| \leq (\Gamma - 1)k^*$

Proof. As each bin $b \in s' \setminus \mathcal{M}'$ contains at least one item from $\bigcup_{j=1}^{k^*} \Gamma(b_j^*) \setminus \{i_1^*, \dots, i_{k^*}^*\}$, and as $|\bigcup_{j=1}^{k^*} \Gamma(b_j^*) \setminus \{i_1^*, \dots, i_{k^*}^*\}| \leq (\Gamma - 1)k^*$, the result is immediate. $\square$

The case of $\mathcal{M}'$, which is slightly more involved, is addressed in the following lemma.

LEMMA 4.4. $|\mathcal{M}'| \leq k^*$*.*

Proof. For convenience in the following without loss of generality let $\mathcal{M}' = \{b_1, \dots, b_{k''}\}$ where $k'' = |\mathcal{M}'| \leq k'$, and let $\{b_1^*, \dots, b_{k^{**}}^*\} \subseteq s^*$ be the set of optimal solution bins each containing Γ deviations, with bins ordered in non-increasing smallest deviation size, so $\hat{a}_{i_k^*} \geq \hat{a}_{i_{k+1}^*}$ for each $k \in [k^{**} - 1]$, and where $k^{**} \leq k^*$. Assume for the sake of deriving a contradiction that $k^{**} < k''$. Then, since $\bigcup_{j=1}^{k''} b_j \subseteq \bigcup_{j=1}^{k^{**}} b_j^*$,

$$\sum_{j \in [k'']} \bar{a}(b_j) \leq \sum_{j \in [k^{**}]} \bar{a}(b_j^*) \leq k^{**} - \sum_{j \in [k^{**}]} \hat{a}_\Gamma(b_j^*).$$

We now show that $\sum_{j \in [k^{**}]} \hat{a}_\Gamma(b_j) \leq \sum_{j \in [k^{**}]} \hat{a}_\Gamma(b_j^*)$. To do so we show by induction on $k = 1, \dots, k^{**} - 1$ for a fixed instance and corresponding algorithm bins and optimal solution bins, $b_1, \dots, b_k''$ and $b_1^*, \dots, b_{k^{**}}$, respectively. For $k = 1$, and all $i \in b_1$ by the fact that these items either do not deviate in s^* or are in the set of smallest deviations, $\{i_1^*, \dots, i_{k^{**}}^*\}$, it follows that $\hat{a}_i \leq \hat{a}_{i_1^*}$, so $\hat{a}_\Gamma(b_1) \leq \hat{a}_\Gamma(b_1^*)$. Now assume $\sum_{j \in [k]} \hat{a}_\Gamma(b_j) \leq \sum_{j \in [k]} \hat{a}_\Gamma(b_j^*)$ in order to prove that $\sum_{j \in [k+1]} \hat{a}_\Gamma(b_j) \leq$

$\sum_{j\in[k+1]} \hat{a}_\Gamma(b_j^*)$. By the induction hypothesis

$$\begin{aligned}\sum_{j\in[k]} \bar{a}(b_j) &> 1 - \sum_{j\in[k]} \hat{a}_\Gamma(b_j)\\ &\geq 1 - \sum_{j\in[k]} \hat{a}_\Gamma(b_j^*)\\ &\geq \sum_{j\in[k]} \bar{a}(b_j^*)\\ &\geq \sum_{j\in[k]} \bar{a}(b_j^* \setminus \Gamma(b_j^*) \cup \{i_j^*\}).\end{aligned}$$

The above inequality and the ordering of the items imply that

$$\hat{a}_{\max} \equiv \max_{i\in\bigcup_{j=k+1}^{k^{**}}(b_j^*\setminus\Gamma(b_j^*)\cup\{i_j^*\})} \hat{a}_i \geq \max_{i\in b_{k+1}} \hat{a}_i.$$

Since $\hat{a}_{i_{k+1}^*}=\hat{a}_{\max}$ it follows that $\hat{a}_\Gamma(b_{k+1}^*) \geq \hat{a}_\Gamma(b_{k+1})$. Together with the induction hypothesis it implies that $\sum_{j\in[k+1]} \hat{a}_\Gamma(b_j) \leq \sum_{j\in[k+1]} \hat{a}_\Gamma(b_j^*)$.

Note that $\hat{a}_\Gamma(b_j) \leq 1$ for $j = 1, \dots, k''$; otherwise if $\hat{a}_\Gamma(b_j) > 1$ for some $j \in [k'']$, then there is some $i \in b_j$ that has $\hat{a}_i > 1/\Gamma$. But then $i \in b_l^*$ for some $l \in [k^{**}]$ and since i is either a nondeviating item and $\hat{a}_\Gamma(b_l^*) > 1$, or $i = i_l^*$ and it implies that b_l^* may contain only $\Gamma - 1$ deviating items, thereby establishing a contradiction. Now, by definition of the algorithm,

$$\sum_{j\in[k'']} \bar{a}(b_j) > \sum_{j\in[k''-1]} \bar{a}(b_j) > k'' - 1 - \sum_{j\in[k''-1]} \hat{a}_\Gamma(b_j) \geq k^{**} - \sum_{j\in[k^{**}]} \hat{a}_\Gamma(b_j^*).$$

The last inequality followed from $k'' - 1 \geq k^{**}$ and $\hat{a}_\Gamma(b_j) \leq 1$ for $j \in [k'']$, implying that $\sum_{j=k^{**}+1}^{k''-1} \hat{a}_\Gamma(b_j) \leq k'' - k^{**}$. ∎

Proof of Lemma 4.2. Lemmas 4.3 and 4.4 immediately imply that $k' \leq \Gamma k^*$, thus proving the claim of Lemma 4.2.

Proof of Theorem 4.1. After step 2 of the algorithm the total number of bins is at most $2\Gamma k^*$, concluding the proof of Theorem 4.1.

To complete the analysis, we establish the following lower bound on the approximation ratio of Next-Fit-Decreasing.

Theorem 4.5. *If the items are ordered according to* (4.1) *or* (3.1)*, then the approximation ratio of* Next-Fit-Decreasing *for* ΓRBP *is at least* $\frac{2\Gamma}{3}$.

Proof. Let us define an instance where the ordering (4.1) can lead to Step 1 of Next-Fit-Decreasing using $k' = \Gamma$ bins while $\mathsf{OPT} = 3$. Every row of the $\Gamma \times \Gamma$ matrix below corresponds to the set of items in a bin (after the Step 1) of Next-Fit-Decreasing algorithm

$$\begin{array}{llll}(\epsilon, 1/\Gamma - \delta_1) & (0, 1/\Gamma - \delta_1) & \dots & (0, 1/\Gamma - \delta_1)\\ \vdots & \vdots & \ddots & \vdots\\ (\epsilon, 1/\Gamma - \delta_\Gamma) & (0, 1/\Gamma - \delta_\Gamma) & \dots & (0, 1/\Gamma - \delta_\Gamma)\end{array} \tag{4.2}$$

where $\epsilon \leq 1/\Gamma$ and $\delta_1 \leq \cdots \leq \delta_\Gamma < \epsilon/\Gamma$. On the one hand, $\epsilon + \Gamma \cdot (1/\Gamma - \delta_l) > 1$ for each $l \in [\Gamma]$, so step 1 of Next-Fit-Decreasing outputs Γ bins. On the other hand, an optimal solution can pack all the items above except the ones in the first column into a single bin because $\Gamma \cdot 1/\Gamma - \delta_1 \leq 1$. Further, the total weight of the first $\Gamma/2$ items of the first column sums up to $\Gamma/2 \cdot (1/\Gamma + \epsilon) - \sum_{l=1}^{\Gamma/2} \delta_l \leq 1 - \sum_{l=1}^{\Gamma/2} \delta_l \leq 1$, and similarly for the last $\Gamma/2$ items, so an optimal solution may pack the first column using two bins.

This instance can be adapted to establish a lower bound for the approximation ratio of Next-Fit-Decreasing when items are ordered according to (3.1). We consider an example that yields a lower bound on the approximation ratio of Next-Fit-Decreasing in solving ΓRBP when the items are ordered according to (3.1). For some $c \geq \Gamma^2$, $\epsilon' = \frac{2}{\Gamma(1+c)}$, $\epsilon = \frac{1}{\Gamma(\Gamma^2+\Gamma-1)}$ consider the instance given by the following $\Gamma \times \Gamma$ matrix:

$$\begin{matrix} (\epsilon', c\epsilon') & (\epsilon, c\epsilon) & \dots & (\epsilon, c\epsilon) \\ \vdots & \vdots & \ddots & \vdots \\ (\epsilon', c\epsilon') & (\epsilon, c\epsilon) & \dots & (\epsilon, c\epsilon) \end{matrix}$$

It can be verified that Next-Fit-Decreasing opens a bin for each row, since $(1+c)\epsilon' + (\Gamma-1)(1+c)\epsilon > 1$. The optimal solution opens 3 bins, 2 bins to store the items of the first column and another bin to store the rightmost $\Gamma - 1$ columns. Although in this example all items $i \in [n]$ are set to have ratios $\hat{a}_i/\overline{a}_i = c$, the example can be extended in a straightforward manner with slight perturbations of the item sizes so that the ratios will be strictly decreasing for the items ordered from left-to-right and top-to-bottom in this matrix. $\square$

We conclude the section by emphasizing our results for the special cases of $\Gamma = 1$ and $\Gamma = 2$, where the straight-forward Next-Fit-Decreasing algorithm with ordering (4.1) obtains the best approximation guarantees using the analysis of the current paper.

Corollary 4.6. *If the items are ordered according to* (4.1) *and $\Gamma = 1$ or $\Gamma = 2$, then* Next-Fit-Decreasing *is a 2-factor or 4-factor approximation algorithm, respectively, for* ΓRBP*.*

In the section that follows we consider constant-factor approximation factors that do not depend on the parameter Γ.

5. A constant-factor approximation algorithm for ΓRBP. The algorithm presented in this section relies on three main ideas. First, we show in Section 5.1 that we can restrict ourselves to instances of ΓRBP with small items; that is, $\overline{a}_i \leq \frac{1}{\Gamma}$ and $\hat{a}_i \leq \frac{1}{\Gamma}$ for each $i \in [n]$. Specifically, we show how to convert any ρ-approximation algorithm for the latter special case of the problem with small items into a general $(\rho + \rho_{\text{bp}})$-approximation for ΓRBP, where ρ_{bp} is an approximation factor guarantee of some algorithm for classical bin packing. First note that having only small items, any set of $\lfloor \Gamma/2 \rfloor$ items can always be packed together into a bin. This fact is stated as the following observation.

Observation 1. *Given an instance $\mathcal{I}$ to the* ΓRBP *satisfying $\hat{a}_i \leq 1/\Gamma$ and $\overline{a}_i \leq 1/\Gamma$ for each $i \in [n]$, any subset $X \subseteq [n]$ can be packed in at most $\lceil \frac{2|X|}{\Gamma} \rceil$ bins.*

Next, we introduce in Section 5.2 variants of ΓRBP where items are packed into bins as in standard bin packing but some of the items are placed in two designated special bins. The deviations will be ignored so that they are irrelevant in these two bins, while in each of the other regular bins only a single item can deviate. One of the special bins that will be referred to as the trash bin, cannot contain more than $k(\Gamma-1)$ items, where k is the number of regular bins used in the solution. The trash contains and "mimics" the $\Gamma-1$ deviating items of each bin in an optimal solution. Following Observation 1, items in the trash can be packed into at most $2k$ additional bins. The problem with trash remains hard because of the capacity of the regular bins, so we focus on almost feasible solutions, which are allowed to exceed each regular bin by one item. We show in that section how finding almost feasible solutions no worse than the optimal solution for the problem with trash leads to an approximation algorithm for ΓRBP with small items.

Finally, we present in Section 5.3 a dynamic programming (DP) algorithm that outputs an almost feasible solution using a number of bins that is at most that of the optimal solution of the original ΓRBP problem. Essentially, for each bin the DP algorithm guesses a particular item that deviates. Then it greedily packs the remaining items with largest nominal values into the trash.

We maintain throughout the section the assumption that the items are ordered according to (4.1).

5.1. Robust bin packing with small items. We define ΓRBP *with small values* as the ΓRBP problem restricted to inputs where for any $i \in [n]$, $\overline{a}_i \leq \frac{1}{\Gamma}$ and $\hat{a}_i \leq \frac{1}{\Gamma}$. The following proposition motivates our focus on ΓRBP with small values.

PROPOSITION 5.1. *Any polynomial ρ-approximation algorithm for* ΓRBP *with input satisfying $\hat{a}_i \leq 1/\Gamma$ and $\overline{a}_i \leq 1/\Gamma$ for each $i \in [n]$, can be turned into a polynomial-time $(\rho+\rho_{bp})$-approximation algorithm for* ΓRBP*.* [2]

Proof. Given an instance $\mathcal{I}$ of ΓRBP, we define the small items $\mathcal{S} = \{i \in [n] : \overline{a}_i \leq 1/\Gamma \text{ and } \hat{a}_i \leq 1/\Gamma\}$ and the large item as $\mathcal{B} = [n] \setminus \mathcal{S}$. We use the given ρ-approximation algorithm to pack $\mathcal{S}$ into $k_{\mathcal{S}}$ bins, so that $k_{\mathcal{S}} \leq \rho \mathsf{OPT}(\mathcal{I})$. Then, we observe that in any packing of $\mathcal{B}$, each bin contains at most Γ items, so that all items deviate in these bins. Hence, the least number of bins needed to pack items in $\mathcal{B}$ is is given by a solution that is optimal to the standard bin packing problem for this same set of items $\mathcal{B}$ where the size of each item $i \in \mathcal{B}$ is $a'_i = \overline{a}_i + \hat{a}_i$. Let $\mathcal{I}'$ denote this instance of standard bin packing So, $\mathsf{OPT}_{\mathrm{bp}}(\mathcal{I}') \leq \mathsf{OPT}(\mathcal{I})$ (where $\mathsf{OPT}_{\mathrm{bp}}$ denotes the optimal objective value of standard bin packing), and observe that any optimal solution of standard bin packing for items $\mathcal{B}$ is a solution that is feasible for the ΓRBP instance $\mathcal{I}$ using the same number of bins. Using a ρ_{bp}-approximation algorithm for standard bin packing to pack $\mathcal{B}$ in $k_{\mathcal{B}}$ bins, it follows that $k_{\mathcal{B}} \leq \rho_{\mathrm{bp}} \mathsf{OPT}_{\mathrm{bp}}(\mathcal{I}') = \rho_{\mathrm{bp}} \mathsf{OPT}(\mathcal{I}) \leq \rho_{\mathrm{bp}} \mathsf{OPT}(\mathcal{I})$. Thus, the packing of $\mathcal{B}$ and $\mathcal{S}$ is a packing of $\mathcal{I}$ satisfying $k_{\mathcal{S}} + k_{\mathcal{B}} \leq (\rho + \rho_{\mathrm{bp}})\mathsf{OPT}(\mathcal{I})$. ∎

Notice that instances with small items are not easier to approximate by NEXT-FIT-DECREASING as illustrated by the instance defined by (4.2), in Section 4, whose items satisfy $a_i, \hat{a}_i \leq 1/\Gamma$ for all $i \in [n]$.

5.2. Bin-packing with trash. For any $X \subseteq [n]$, we define $\tilde{a}_\Gamma(X) = \Gamma \hat{a}_1(X)$ ($\tilde{a}_\Gamma(X)$ is Γ times the largest deviating value of an item in X) and $\tilde{f}(X) = \overline{a}(X) + \tilde{a}_\Gamma(X)$. We introduce next a decision problem ΓRBP-T that is related to ΓRBP.

[2] In general, if we have a polynomial time additive approximation algorithm using $OPT + f(OPT)$ bins and polynomial time ρ-approximation algorithm for ΓRBP with small values then our algorithm uses $OPT(\rho+1) + f(OPT)$ bins for ΓRBP in polynomial time.

	ΓRBP-T (Robust bin packing with trash)
Input:	$(\mathcal{I}, k, t)$ where $\mathcal{I}$ is an instance of ΓRBP and $k, t \in \mathbb{N}$.
Output:	'Yes' if a solution exists, which is a partition of the set of items into $k+1$ sets $b_1, \dots, b_k$ and T (called the trash) such that: • $\tilde{f}(b_j) \le 1$ for each $j = 1, \dots, k$ • $\lvert T\rvert \le t$ and 'No' otherwise.

Notice that although the input of ΓRBP-T is assumed to include only small items, it is possible to have an item $i \in [n]$ such that $\tilde{f}(\{i\}) > 1$, implying that $i \in T$. The following two lemmas suggest how the decision problem ΓRBP-T may be used to determine an approximate solution of ΓRBP.

LEMMA 5.2. *For any input $\mathcal{I}$ of ΓRBP where $k^* = OPT(\mathcal{I})$, $(\mathcal{I}, k^*, (\Gamma - 1)k^*)$ is a yes instance of ΓRBP-T.*

Proof. Given an optimal solution with objective value k^* of ΓRBP we create a solution to ΓRBP-T problem as follows. For some (arbitrary) $j \in [k^*]$, let b_j^* be a bin of the considered optimum. Let $N_j = b_j^* \setminus \Gamma(b_j^*)$ (the non-deviating items of b_j^*). For $j = 1, \dots, k^*$, let

$$X_j = \begin{cases} \max(\Gamma(b_j^*)) & |\Gamma(b_j^*)| = \Gamma \\ \emptyset & \text{otherwise.} \end{cases}$$

We define $b'_j = N_j \cup X_j$, and adjoin items of $Y_j = b_j^* \setminus b'_j$ to the trash. Note that Y_j is either the set of $\Gamma - 1$ largest deviating items of b_j^*, or otherwise $b_j^* = \Gamma(b_j^*)$ and $|\Gamma(b_j^*)| < \Gamma$. So, $(b'_1, \dots, b'_{k^*}, T)$ is a yes instance for the ΓRBP-T problem since evidently $\tilde{f}(b'_j) = \overline{a}(b'_j) + \tilde{a}_\Gamma(b'_j) \le \overline{a}(b_j^*) + \hat{a}_\Gamma(b_j^*) \le 1$ and also $|T| \le (\Gamma - 1)k^*$. ∎

The next lemma establishes that yes-instances of ΓRBP-T can be used to construct solutions of ΓRBP using at most a number bins that is a constant factor of the number of bins used by ΓRBP-T.

LEMMA 5.3. *For any instance $\mathcal{I}$ of ΓRBP satisfying $\hat{a}_i \le 1/\Gamma$ and $\overline{a}_i \le 1/\Gamma$ for each $i \in [n]$, and integer k, given a yes-instance of $(\mathcal{I}, k, \Gamma k)$ of ΓRBP-T, we can compute in polynomial time a solution of $3k$ bins for $\mathcal{I}$.*

Proof. Given a solution $b_1, \dots, b_k$, T for $(\mathcal{I}, k, \Gamma k)$ of ΓRBP-T the bins remain feasible in ΓRBP as for each $j \in [k]$, $f_\Gamma(b_j) = \overline{a}(b_j) + \hat{a}_\Gamma(b_j) \le \overline{a}(b_j) + \tilde{a}(b_j) = \tilde{f}(b_j)$. Then, Observation 1 implies that the trash T can be packed into $\lceil k\Gamma/(\Gamma/2) \rceil \le 2k$ additional bins. ∎

Notice that the trash T in the instance constructed in this lemma has $|T| = \Gamma k$, which exceeds the $|T| = (\Gamma - 1)k$ in Lemma 5.2. The additional k "slots" are necessary for deciding ΓRBP-T as will be illustrated in the analysis that follows.

In order to develop an algorithm and in particular a dynamic program (DP) for deciding ΓRBP-T it is convenient to consider an optimization variant of ΓRBP-T, to be called G-ΓRBP-T for "generalized robust bin packing with trash". This variant is defined in the following for a fixed instance $\mathcal{I}$ of ΓRBP and a given integer k (so that they are not considered a part of the input).

Generalized robust bin packing with trash (G-ΓRBP-T)

Input:	$\mathcal{I}' = (q, t, \ell)$, where $q \in [n]$, $t \in [(\Gamma-1)k]_0$, and $\ell \in [k+1]$.
Output:	A feasible solution s is a partition of $[\![q, n]\!]$ into $k-\ell+3$ sets, given as a triple $(L, \{b_j : j \in [\![\ell, k]\!]\}, T)$, such that • for any $j \in [\![\ell, k]\!]$, $\tilde{f}(b_j) \leq 1$ (the $k-\ell+1$ regular bins must satisfy the fill constraints of ΓRBP-T) • $\lvert T\rvert \leq t$ (we only allow t items in the trash) • $\min(b_\ell) = q$ (meaning that the deviating item of b_ℓ is q)
Minimize:	$c(s) = \overline{a}(L)$

The objective of G-ΓRBP-T is to pack a part (defined by $[\![q, k]\!]$) of the instance $\mathcal{I}$ for ΓRBP-T given a fixed budget of resources (the number of bins and the size of the trash) while minimizing the sum only of nominal sizes of items in the leftover itemset L. The last constraint (the deviating item of b_ℓ is q), which may appear somewhat artificial, will allow determining optimal solutions of ΓRBP-T from optimal solutions of G-ΓRBP-T, by carefully enumerating possibilities of largest deviating item to be packed each bin in an intelligent way; considering only those possibilities corresponding to solutions that are close to being feasible (a notion to be defined more precisely in the following) and whose objective value is at most that of an optimal solution. This enumeration scheme will be shown to be efficiently solvable by a DP. For convenience, the objective value of infeasible solutions s, including for example $s = (\emptyset, \emptyset, \emptyset)$, is defined as $c(s) = \infty$.

The capacity constraints of the bins make G-ΓRBP-T hard to solve in general. Hence, following the spirit of NEXT-FIT-DECREASING introduced previously, we introduce below almost feasible solutions, which can exceed the capacity of each bin by one item.

DEFINITION 5.4 (almost feasible solution). *We say that a bin b* exceeds by at most one item *iff $\tilde{f}(b) > 1$ and $\tilde{f}(b \setminus \{i\}) \leq 1$ where $i = \max(b)$. Given an input $\mathcal{I}' = (q, t, \ell)$ of* G-ΓRBP-T*, we say that a solution is* almost feasible *iff all the* G-ΓRBP-T *constraints are satisfied, except that for any $j \in [\![\ell, k]\!]$, we allow that b_j exceeds by at most one item instead of $\tilde{f}(b_j) \leq 1$.*

DEFINITION 5.5 (an optimal almost-feasible solution). *Given an input $\mathcal{I}' = (q, t, \ell)$ of* G-ΓRBP-T*, we say that a solution $s = (L, \{b_1, \dots, b_k\}, T)$ is an* optimal almost-feasible solution *iff s is almost feasible with $c(s) \leq \mathsf{OPT}(\mathcal{I}')$.*

The relation between G-ΓRBP-T and ΓRBP-T is characterized in the following two lemmas. Let $\mathsf{OPT}(\mathcal{I}')$ be the optimal solution cost of G-ΓRBP-T instance $\mathcal{I}'$ (notice that an optimal solution must also be feasible for the given problem).

LEMMA 5.6. *For any input $\mathcal{I}$ of* ΓRBP *and k such that $(\mathcal{I}, k, (\Gamma-1)k)$ is a yes input of* ΓRBP-T*, there exist a positive integer $q \in [(\Gamma-1)k]$ such that $\mathsf{OPT}(q, (\Gamma-1)k - (q-1), 1) = 0$.*

Proof. Consider a ΓRBP-T yes-instance $(\mathcal{I}, k, (\Gamma-1)k)$ and corresponding partition solution $\{b_1, \dots, b_k\}, T$. Let $q = \min\left(\bigcup_{j=1}^{k} b_j\right)$ (the item with smallest index that is packed in a bin). Let $t = (\Gamma-1)k - (q-1)$. By definition of q and (4.1), items in $[q-1]$ must be in T, and thus it means that the triple $(\emptyset, \{b_1, \dots, b_k\}, T \setminus [q-1])$ is a feasible solution of G-ΓRBP-T $(q, t, 1)$ with cost 0. ∎

Lemma 5.7. *Let us fix $\mathcal{I}$ an input of* ΓRBP *and k an integer. For any $q \in [(\Gamma-1)k], t = (\Gamma-1)k-(q-1)$, given an almost feasible* 0 *cost solution for* G-ΓRBP-T *instance $\mathcal{I}' = (q,t,1)$, a solution for* ΓRBP-T *instance $(\mathcal{I}, k, \Gamma k)$ can be determined in polynomial time.*

Proof. Let $(\emptyset, \{b_1, \ldots, b_k\}, T)$ be an optimal solution of G-ΓRBP-T instance $\mathcal{I}' = (q,t,1)$ with cost 0. For $j \in [k]$, let $b'_j = b_j \setminus \max(b_j)$. Let $T' = T \cup [q-1] \cup \bigcup_{j=1}^{k} \max(b_j)$. We now have $\tilde{f}(b'_j) \leq 1$ for any $j \in [k]$ (as b_j exceeds by at most one item), and $|T'| \leq \Gamma k$, so $\{b'_1, \ldots, b'_k, T\}$ is a yes-instance for ΓRBP-T $(\mathcal{I}, k, \Gamma k)$ thus concluding the proof. ∎

The above lemmas imply that any algorithm outputting an (approximately) optimal almost-feasible solution for G-ΓRBP-T can be used to devise approximation algorithm for ΓRBP with small values, which is illustrated by Algorithm 2.

> $s^* \leftarrow ([n], \emptyset, \emptyset), lb \leftarrow 1, ub \leftarrow n$
> **while** $lb \neq ub$ **do**
> $k \leftarrow \lceil \frac{lb+ub}{2} \rceil$
> $\mathcal{I}' \leftarrow (1, \Gamma(k-1), 1)$
>   Let $s = (L, \{b_1, \ldots, b_k\}, T)$, be an optimal almost-feasible solution returned by the G-ΓRBP-T algorithm given input $\mathcal{I}'$
>   **if** $c(s) = 0$ **then**
> $ub \leftarrow k$
> $s^* \leftarrow s$ (accordingly $T^* \leftarrow T$)
>   **else**
> $lb \leftarrow k+1$
> Pack items in T^* into bins $b^*_{k+1}, \ldots, b^*_{\bar{k}}$ according to Observation 1.
> **return:** $b^*_1, \ldots, b^*_{\bar{k}}$

Algorithm 2: Algorithm for ΓRBP with small values.

Proposition 5.8. *Algorithm 2 is a* 3*-approximation for* ΓRBP *with small values.*

Proof. Let A be a G-ΓRBP-T algorithm that is guaranteed to output an optimal almost-feasible solution given instance (q,t,ℓ), and whose output will be denotes by $c(A(q,t,\ell))$. Let $k^* = \mathsf{OPT}(\mathcal{I})$ be the optimal value of ΓRBP. Lemma 5.2 implies that $(\mathcal{I}, k^*, (\Gamma-1)k^*)$ is a *yes*-instance of ΓRBP-T, so Lemma 5.6 implies that there exists $q \in [(\Gamma-1)k^*]$ and $t = (\Gamma-1)k^* - (q-1)$ for which $\mathsf{OPT}(q,t,1) = 0$. Thus, the assumption implies that A outputs an almost feasible solution s that satisfies $c(s) \leq 0$, so that $c(s^*) \leq k^*$. Applying Lemma 5.7, we can compute a solution of $(\mathcal{I}, k^*, \Gamma k^*)$ for ΓRBP-T, which corresponds to a solution using $3k^*$ bins for ΓRBP following Lemma 5.3. ∎

The next section describes a dynamic programming algorithm that outputs an optimal almost-feasible solution for G-ΓRBP-T.

5.3. A DP algorithm for G-ΓRBP-T. We now define a DP scheme that given instance $\mathcal{I}' = (q,t,\ell)$ provides an *almost feasible* solution s with $c(s) \leq \mathsf{OPT}(\mathcal{I}')$. We start by providing a gentle description before formally defining the algorithm and proving its correctness. Let s^* be an optimal solution for $\mathcal{I}'$, with bins ordered in non-increasing deviation size. The DP algorithm starts by enumerating $g = (q', t') \in [\![q+1, n]\!] \times [t]_0$. One of the enumerated g must necessarily correspond to the optimal solution s^* in the following sense:

- $q' = \min(b^*_{\ell+1})$

- t' is the number of items trashed from X' in s^*, where $X' = [\![q, q'-1]\!]$

Because of the ordering of the bins in the optimal solution s^*, the items of X' must be packed in bins b^*_ℓ, L^* or T^*. As the DP algorithm mimics the optimal solution, it packs X' in b_ℓ, L and T. Specifically, the DP algorithm:

- Packs q to b_ℓ (as required by the corresponding constraint of G-ΓRBP-T).
- Packs the remaining t' largest nominal value items of X' in the trash.
- Packs the remaining items of X' into b_ℓ until $\tilde{f}(b_\ell) > 1$ or $X' = \emptyset$.
- Packs the remaining items of X' into L until $X' = \emptyset$.

We discuss next where the others items (of $[\![q', n]\!]$) are packed. Notice that in s^*, bin b^*_ℓ may contain items of $[\![q', n]\!]$, and thus the DP algorithm may also have to pack items of $[\![q', n]\!]$ into b_ℓ. To allow for that possibility, we postpone the decision of which items of $[\![q', n]\!]$ to pack into b_ℓ. Specifically, let Δ_ℓ be the size of the empty space in b_ℓ after packing X' as described above, and let $L^{X'} = L \cap X'$. After the previous steps, the DP algorithm makes a recursive call to get a solution $\tilde{s}$ that packs $[\![q', n]\!]$ into regular bins, a trash, and a leftover itemset $\tilde{b}_0$. So far solution $\tilde{s}$ has not benefited from the empty space Δ_ℓ. However, we can unpack items from $\tilde{b}_0$ to b_ℓ while ensuring that these items do not deviate in b_ℓ (as all these items have index greater than q).

Our DP scheme is defined by Algorithm 3. An iteration of this algorithm is further illustrated in Figure 1.

$s \leftarrow (\emptyset, \emptyset, \emptyset)$ `// Where` $c((\emptyset,\emptyset,\emptyset)) = \infty$
if $\ell = k$ **then**
    $X' \leftarrow [\![q, n]\!]$
    $T \leftarrow \{\min(t, n-q)$ largest items in terms of nominal value in $X'\}$
    Pack $X' \setminus T$ in b_k until $\tilde{f}(b_k) > 1$ or $X' \setminus T = \emptyset$
    Pack the remaining items in L
    return: $s = (L, \{b_k\}, T)$
for $g = (q', t') \in [\![q+1, n]\!] \times [\min(t, q'-q)]_0$ **do**
    Pack q in b_ℓ
    $X' \leftarrow [\![q+1, q'-1]\!]$
    $T' \leftarrow \{t'$ largest items in terms of nominal value in $X'\}$
    Pack $X' \setminus T'$ in b_ℓ until $\tilde{f}(b_\ell) > 1$ or $X' \setminus T' = \emptyset$
    Pack the remaining items in $X' \setminus T'$ in L
    $\tilde{s} = (\tilde{b}_0, \{\tilde{b}_{\ell+1}, \dots, \tilde{b}_k\}, \tilde{T}) \leftarrow \mathrm{DP}(q', t-t', \ell+1)$.
    Unpack $\tilde{b}_0$ into b_ℓ until $\tilde{f}(b_\ell) > 1$ or all items of $\tilde{b}_0$ are unpacked, then unpack the potentially remaining items of $\tilde{b}_0$ into L.
    $s_g \leftarrow (L, \{b_\ell, \tilde{b}_{\ell+1} \dots, \tilde{b}_k\}, \tilde{T} \cup T')$
    if $c(s_g) < c(s)$ **then** $s \leftarrow s_g$
return: s

Algorithm 3: $\mathrm{DP}(q, t, \ell)$

Let us introduce notations to describe the packing of the DP. Let $b^{X'}_\ell = b_\ell \cap X'$ be the items of X' packed in b_ℓ by the DP, and let $\Delta_\ell = 1 - \tilde{f}(b^{X'}_\ell)$ (Δ_ℓ could be negative).

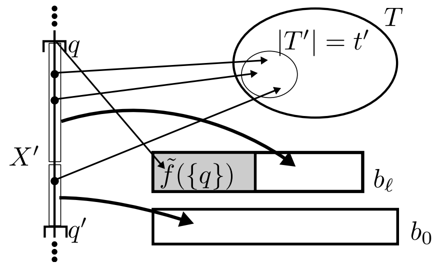

FIG. 1. *DP algorithm handling guess (q', t'), starting from item q.*

We define similarly $L^{X'} = L \cap X'$. Let $c_0 = \bar{a}(L^{X'})$. Notice that c_0 corresponds to the total non deviating size of items packed in L after Step 12 of Algorithm 3. The following lemma establishes the correctness of the algorithm.

LEMMA 5.9. *For any $\mathcal{I}' = (q, t, \ell)$ input of* G-ΓRBP-T*,* DP($\mathcal{I}'$) *determines an optimal almost feasible solution.*

Proof. The proof is by induction on ℓ.

$\ell = k$.. Consider a solution $s = (L, \{b_k\}, T)$ output by the algorithm and optimal solution $s^* = (b_0^*, \{b_k^*\}, T^*)$. In step 5, either $b_\ell = \{q\} \cup X' \setminus T$, in which case $L = \emptyset$ and $\bar{a}(L) \le \bar{a}(b_0^*)$, or $\tilde{f}(b_k) > 1 \ge \tilde{f}(b_k^*)$. Then, as $q = \min(b_k) = \min(b_k^*)$ (since both solutions must satisfy this constraint) it implies that $\Gamma\hat{a}_1(b_k^*) = \Gamma\hat{a}_1(b_k)$, and so $\bar{a}(b_k) > \bar{a}(b_k^*)$. Then, by the definition of T (in step 4 of the algorithm), also $\bar{a}(T) \ge \bar{a}(T^*)$, so it must be that $\bar{a}(L) < \bar{a}(b_0^*)$.

Inductive step.. Suppose now that, for some $\hat{\ell} \in [k-1]$, the lemma holds for $\ell = \hat{\ell} + 1$ and that $s^* = (b_0^*, \{b_{\hat{\ell}}^*, \dots, b_k^*\}, T^*)$ is optimal for G-ΓRBP-T $(q, t, \hat{\ell})$. Let $q^* = \min(b_{\hat{\ell}+1})$, let $X^* = [\![q, q^* - 1]\!]$ and observe that $X^* \neq \emptyset$. Next, observe that $(b_0^* \setminus X^*, \{b_{\hat{\ell}+1}^*, \dots, b_k^*\}, T^* \setminus X^*)$ is optimal for G-ΓRBP-T $(q^*, \hat{t}, \hat{\ell}+1)$ where $\hat{t} = |T^* \setminus X^*| \le t$. Otherwise, there would exist some feasible $(\check{L}, \{\check{b}_{\hat{\ell}+1}, \dots, \check{b}_k\}, \check{T})$ that has $\bar{a}(\check{L}) < \bar{a}(b_0^* \setminus X^*)$ and then $(\check{L} \cup (b_0^* \cap X^*), \{\{q\} \cup (b_{\hat{\ell}} \cap X^*), \check{b}_{\hat{\ell}+1}, \dots, \check{b}_k\}, \check{T} \cup (T^* \cap X^*))$ would be feasible for $(q, t, \hat{\ell})$ with $\bar{a}(\bar{b} \cup (b_0^* \cap X^*)) < \bar{a}(b_0^*)$, thereby contradicting the optimality of s^*.

By the inductive hypothesis $\bar{a}(\tilde{L}) \le \bar{a}(b_0^* \setminus X^*)$ for some almost feasible $(\tilde{L}, \{\tilde{b}_{\hat{\ell}+1}, \dots, \tilde{b}_k\}, \tilde{T})$ that is output by the algorithm for $(q^*, \hat{t}, \hat{\ell}+1)$, where $q^* > q$ and $\hat{t} = |T^* \setminus X^*| \le t$. Therefore,

$$\bar{a}(b_0^*) = \bar{a}(b_0^* \setminus X^*) + \bar{a}(b_0^* \cap X^*) \ge \bar{a}(\tilde{L}) + \bar{a}(b_0^* \cap X^*). \tag{5.1}$$

Let us now consider the iteration $g = (q^*, t - \hat{t})$ of the main loop (starting in step 7) and L and $b_{\hat{\ell}}$ the corresponding bins of s_g. By steps 8, 11 and 14 of the algorithm, $L \setminus \tilde{L} = (X^* \setminus T') \setminus b_{\hat{\ell}}$. In line 11 of the algorithm if $\tilde{f}(b_{\hat{\ell}}) \le 1$, then $L \subseteq \tilde{L}$ and $L \setminus \tilde{L} = \emptyset$. Otherwise $\tilde{f}(b_{\hat{\ell}}) > 1 \ge \tilde{f}(b_{\hat{\ell}}^*)$. Recalling that both solutions must satisfy the given constraint, $q = \min(b_{\hat{\ell}}) = \min(b_{\hat{\ell}}^*)$, it implies that $\Gamma\hat{a}_1(b_{\hat{\ell}}) = \Gamma\hat{a}_1(b_{\hat{\ell}}^*)$. Thus, $\bar{a}(b_{\hat{\ell}}) > \bar{a}(b_{\hat{\ell}}^*)$. Also, by definition of T', $\bar{a}(T') \ge \bar{a}(T^* \cap X^*)$. Thus, $\bar{a}(L \setminus \tilde{L}) \le \bar{a}(b_0^* \cap X^*)$. Combined with (5.1) it finally implies that $\bar{a}(b_0^*) \ge \bar{a}(\tilde{L}) + \bar{a}(b_0^* \cap X^*) \ge \bar{a}(L \setminus \tilde{L}) + \bar{a}(\tilde{L}) = \bar{a}(L)$. ∎

LEMMA 5.10. *Suppose that $a_i, \hat{a}_i \le 1/\Gamma$ for all $i \in [n]$. Then, Algorithm 2, with Algorithm 3 as a subroutine for determining almost-feasible optimal solutions for* G-ΓRBP-T*, is a 3-factor approximation algorithm for* ΓRBP*. Further, the running* *time complexity bound of this algorithm is in $\mathcal{O}(n^6 log(n))$.*

Proof. Proposition 5.8 and Lemma 5.9 applied to Algorithm 3 imply that there exists a 3-approximation for ΓRBP with small values. The different number of inputs of the DP is $\mathcal{O}(n^3)$, and the running time for a fixed input is $\mathcal{O}(n^3)$. Then, since the binary search of Algorithm 2 takes $O(\log n)$ iterations it follows that the total running time is in $\mathcal{O}(n^6 \log n)$.

Now, Lemma 5.10 together with Proposition 5.1 imply the following theorem using the $\rho_{\text{bp}} = \frac{3}{2}$-approximation first-fit decreasing (FFD) algorithm, whose running time complexity bound is clearly dominated by that of the algorithm for ΓRBP with small values, for standard bin packing; see for example [24].

THEOREM 5.11. *There exists a* 4.5*-approximation algorithm for* ΓRBP*with an* $\mathcal{O}(n^6 log(n))$ *runtime complexity bound.*

In particular, combining FFD to to pack large items $i \in [n]$, for which with $\bar{a}_i > 1/\Gamma$ or $\hat{a}_i > 1/\Gamma$ and Algorithm 2 using Algorithm 3 as a subroutine to pack the other (small) items, is an algorithm that satisfies the claim of Theorem 5.11. Also note that while not a focus of the current paper, in the asymptotic setting (as n tends to be large) using instead an asymptotic FPTAS for packing the large items, for $\epsilon > 0$, an asymptotic approximation of $4 + \epsilon$ could be guaranteed in running time that is polynomial in n and $1/\epsilon$. Finally, although this result establishes a constant factor approximation for our problem when Γ is a part of the input, it can be observed that in the special case that $\Gamma \leq 2$, the next-fit-decreasing approximation established by Theorem 4.6 may be preferred as a practical and fast $O(n \log n)$ algorithm with a $2\Gamma \leq 4$ approximation guarantee.

6. Conclusion. This paper considered the bin-packing with item size uncertainty, following the robust optimization approach with two widely used variants of budget uncertainty sets, U^Γ and U^Ω. We have shown that problem ΓRBP belongs to a different complexity class than its deterministic counterpart, in particular that no asymptotic approximation scheme exists for that problem, unless $\mathcal{P} = \mathcal{NP}$. We have further provided constant factor approximation algorithms for problems ΩRBP and ΓRBP. While the algorithm devised for ΩRBP is based on a rather natural extension of next-fit-decreasing, the latter provided only non-constant ratios for ΓRBP. Instead we are able to devise a constant-ratio for ΓRBP through a more involved algorithm, separately handling "small" and "large" items, and applying a dynamic programming-based algorithm to the small items, or rather to a relaxed version of the problem where not all of the items need to be packed in bins.

Future work may address the remaining gap between approximations ratio lower and upper bounds for ΩRBP and ΓRBP with different values of Γ (in particular the cases considered in this paper of $\Gamma = 1, 2$ and general $\Gamma > 2$). While is is shown that ΓRBP may not have a polynomial asymptotic approximation scheme, it remains open whether the same may be true for ΩRBP. A different line of research could seek to extend our results to natural extensions of the uncertainty sets considered herein. For instance, an important variant of these sets would consider smoothing constraints, correlating the amount of deviation of subsets of dependent items. In particular, results obtained for the U^Ω model seem to naturally extend to this smoothed variant. Applications may include, for example, medical procedures that last longer when the staff performing them is not well rested. For such medical-procedure scheduling applications, also of interest are two stage models in which some of the items can be repacked at a cost, once the uncertainty is revealed. Clearly, if the cost of repacking is higher than unity, then the single stage approximation results of this paper would apply as a special case of this general setting.

REFERENCES

[1] M. Adler, P. Gibbons, and Y. Matias, *Scheduling space-sharing for internet advertising*, Journal of Scheduling, 5 (2002), pp. 103–119.

[2] E. Álvarez-Miranda, I. Ljubic, and P. Toth, *A note on the Bertsimas & Sim algorithm for robust combinatorial optimization problems*, 4OR, 11 (2013), pp. 349–360, https://doi.org/10.1007/s10288-013-0231-6.

[3] J. Balogh, J. Békési, G. Dósa, L. Epstein, and A. Levin, *A New and Improved Algorithm for Online Bin Packing*, in ESA, Y. Azar, H. Bast, and G. Herman, eds., vol. 112 of LIPIcs, Dagstuhl, Germany, 2018, pp. 5:1–5:14.

[4] J. Balogh, J. Békési, G. Dósa, L. Epstein, and A. Levin, *A new lower bound for classic online bin packing*, in International Workshop on Approximation and Online Algorithms, Springer, 2019, pp. 18–28.

[5] J. Balogh, J. Békési, G. Dósa, J. Sgall, and R. van Stee, *The optimal absolute ratio for online bin packing*, in Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, 2015.

[6] A. Basu Roy, M. Bougeret, N. Goldberg, and M. Poss, *Approximating robust bin-packing with budgeted uncertainty*, 2019. Algorithms and Data Structures Symposium (WADS).

[7] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton University Press, 2009, https://doi.org/10.1515/9781400831050.

[8] D. Bertsimas and M. Sim, *Robust discrete optimization and network flows*, Math. Program., 98 (2003), pp. 49–71.

[9] M. Bougeret, K. Jansen, M. Poss, and L. Rohwedder, *Approximation results for makespan minimization with budgeted uncertainty*, in International Workshop on Approximation and Online Algorithms, Springer, 2019, pp. 60–71.

[10] M. Bougeret, A. A. Pessoa, and M. Poss, *Robust scheduling with budgeted uncertainty*, Discrete Applied Mathematics, 261 (2019), pp. 93–107.

[11] C. Buchheim and J. Kurtz, *Robust combinatorial optimization under convex and discrete cost uncertainty*, EURO J. Computational Optimization, 6 (2018), pp. 211–238, https://doi.org/10.1007/s13675-018-0103-0, https://doi.org/10.1007/s13675-018-0103-0.

[12] H. I. Christensen, A. Khan, S. Pokutta, and P. Tetali, *Approximation and online algorithms for multidimensional bin packing: A survey*, Computer Science Review, 24 (2017), pp. 63–79.

[13] W. F. de la Vega and G. S. Lueker, *Bin packing can be solved within 1+epsilon in linear time*, Comb., 1 (1981), pp. 349–355, https://doi.org/10.1007/BF02579456.

[14] F. Dexter, A. Macario, and R. D. Traub, *Which algorithm for scheduling add-on elective cases maximizes operating room utilization? use of bin packing algorithms and fuzzy constraints in operating room management.*, Anesthesiology, 91 (1999), pp. 1491–1500.

[15] G. Dósa, *The tight bound of first fit decreasing bin-packing algorithm is FFD(I) <= 11/9 OPT(I) + 6/9*, in Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, First International Symposium, ESCAPE 2007, Hangzhou, China, April 7-9, 2007, Revised Selected Papers, B. Chen, M. Paterson, and G. Zhang, eds., vol. 4614 of Lecture Notes in Computer Science, Springer, 2007, pp. 1–11.

[16] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[17] K. Goetzmann, S. Stiller, and C. Telha, *Optimization over integers with robustness in cost and few constraints*, in WAOA, 2011, pp. 89–101.

[18] N. Goldberg and S. Karhi, *Online packing of arbitrary sized items into designated and multipurpose bins*, European Journal of Operational Research, 279 (2019), pp. 54–67.

[19] C. E. Gounaris, W. Wiesemann, and C. A. Floudas, *The robust capacitated vehicle routing problem under demand uncertainty*, Operations Research, 61 (2013), pp. 677–693.

[20] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham, *Worst-case performance bounds for simple one-dimensional packing algorithms*, SIAM Journal on Computing, 3 (1974), pp. 299–325.

[21] N. Karmarkar and R. M. Karp, *An efficient approximation scheme for the one-dimensional bin-packing problem*, in Proc. 23rd Annual Symp. Foundations of Computer Science (sfcs 1982), Nov. 1982, pp. 312–320.

[22] A. A. Pessoa, M. Poss, R. Sadykov, and F. Vanderbeck, *Branch-and-cut-and-price for the*

robust capacitated vehicle routing problem with knapsack uncertainty, Operations Research, (2020), https://hal.inria.fr/hal-01958184/. In press.

[23] M. Poss, *Robust combinatorial optimization with knapsack uncertainty*, Discrete Optimization, 27 (2018), pp. 88–102, https://doi.org/10.1016/j.disopt.2017.09.004.

[24] D. Simchi-Levi, *New worst-case results for the bin-packing problem*, Naval Research Logistics, 41 (1994), pp. 579–585.

[25] G. Song, D. Kowalczyk, and R. Leus, *The robust machine availability problem–bin packing under uncertainty*, IISE Trans., (2018), pp. 1–35.

[26] B. Tadayon and J. C. Smith, *Algorithms and complexity analysis for robust single-machine scheduling problems*, J. Scheduling, 18 (2015), pp. 575–592.