Probabilistic Analysis of Online (Class-constrained) Bin Packing and Bin Covering⋆

Carsten Fischer and Heiko Röglin

Department of Computer Science
University of Bonn, Germany
carsten.fischer@uni-bonn.de, roeglin@cs.uni-bonn.de

Abstract. We study online algorithms for bin packing and bin covering in two different probabilistic settings in which the item sizes are drawn randomly or the items are adversarial but arrive in random order. We prove several results on the expected performance of well-known online algorithms in these settings. In particular, we prove that the simple greedy algorithm Dual Next-Fit for bin covering performs in the random-order setting strictly better than in the worst case, proving a conjecture by Christ et al. (Theoretical Computer Science, 556:71-84, 2014).
Additionally we also study class-constrained bin packing and bin covering. In these problems, each item has not only a size but also a color and there are constraints on the number of different colors in each bin. These problems have been studied before in the classical worst-case model and we provide the first probabilistic analysis of these problems. We prove for several simple online algorithms bounds on their expected performance in the two probabilistic models discussed above. We observe that in the case of class constrained bin packing for several algorithms their performance differs with respect to the two probabilistic performance measures.

1 Introduction

Bin packing and bin covering are classical optimization problems, which have been studied extensively both as offline and online problems. In these problems, the input consists of a set of n items with sizes $s_1, \dots, s_n \in [0, 1]$ and one seeks for a partition of the items into bins. In the *bin packing problem* the goal is to partition the items into as few bins as possible such that each bin contains items with a total size of at most 1, whereas in the *bin covering problem* the goal is to partition the items into as many bins as possible such that each bin contains items with a total size of at least 1.

In addition to these pure versions, also several variations of bin packing and bin covering with additional constraints are of interest. One particular line of research is concerned with *class constrained* versions in which an additional

⋆ This research was supported by ERC Starting Grant 306465 (BeyondWorstCase).

parameter k is given and each item i has not only a size $s_i \in [0,1]$ but also a color $c_i \in \mathbb{N}$. In the *class constrained bin packing problem* the goal is to partition the items into as few bins as possible such that each bin contains items with a total size of at most 1 and of at most k different colors. In the *class constrained bin covering problem* the goal is to find a partition into as many bins as possible such that each bin contains items with a total size of at least 1 and of at least k different colors.

The class constrained bin packing problem has been introduced in [13] and studied in a sequence of papers [14, 16, 7]. Its theoretical importance stems from the fact that it generalizes the classical bin packing problem and the *cardinality constrained bin packing problem* (see e.g. [11, 1, 3]). In cardinality constrained bin packing there is a parameter $k \in \mathbb{N}$ given and a bin must contain at most k items. From a practical point of view there are applications in production planning and video-on-demand systems [16]. Given the class constrained bin packing problem, it is natural to also study the class constrained bin covering problem, which has been introduced by Epstein et al. [6] with applications in fault-tolerant communication networks.

In this article, we focus on the online setting, in which the items arrive one after another and an algorithm has to assign each item immediately and irrevocably upon its arrival to one of the bins without knowing the items that come afterwards. We are particularly interested in probabilistic performance measures. We study the setting where the items are drawn independently and identically distributed (i.i.d.) from an adversarial distribution and the random-order model, in which an adversary chooses the set of items, but the items arrive in random order.

Up to now, it is not fully understood when the performance of algorithms coincide or differs in these two probabilistic settings. We prove several new upper and lower bounds on the competitive ratio of online algorithms in these probabilistic models both for the classical and the class constrained versions of bin packing and bin covering. For special cases we observe that even heuristics behave asymptotically optimal on random input. In the case of class constrained bin packing we observe different behaviors of the considered algorithms w.r.t. the two performance measures. Our analysis sheds new light on the nature of these two probabilistic performance measures in the context of bin packing and bin covering and its variants.

1.1 Probabilistic Performance Measures

In all problems considered in this article, an instance I is given by a sequence $(a_1, \dots, a_n)$ of items, and we assume that the items arrive in the order specified by their indices. We denote by $\mathrm{OPT}(I)$ the value of the optimal offline solution, i.e., the minimum number of bins needed to pack the items in the (class constrained) bin packing problem and the maximum number of bins that can be covered by the items in the (class constrained) bin covering problem. Similarly, for an algorithm A we denote by $A(I)$ the value of the solution computed by A on instance I. Furthermore, we denote by $|I|$ the number of items in instance I.

The usual performance measure for an online algorithm A is its (asymptotic) competitive ratio, which essentially measures by which factor the solutions computed by A can be worse than the optimal offline solution. Since competitive analysis is based on the worst-case behavior of algorithms, it often yields too pessimistic results and, in many cases, it is not fine-grained enough to differentiate meaningfully between different algorithms. Worst-case analysis can be viewed as a game between the algorithm designer and an adversary whose goal is to select an input on which the designed algorithm performs as poorly as possible. For the reasons discussed above, we weaken the adversary by studying inputs that are to some extent random.

We first describe the probabilistic measures in terms of (class constrained) bin packing and discuss later how they can be adapted to (class constrained) bin covering. The first probabilistic model we consider is i.i.d. sampling. Let $\mathcal{I}$ denote a (possibly infinite) multiset of items and $p : \mathcal{I} \to [0, 1]$ a probability measure on the set of items. Observe that in the case of class constrained bin packing an item is a tuple (s, c) consisting of a size s and a color c whereas in the normal bin packing problem p is simply a probability measure on item sizes. We will often denote the pair $(\mathcal{I}, p)$ by F. Then, I_n^F denotes a random instance $(A_1, \ldots, A_n)$, where n items are drawn independently according to F. The *asymptotic average performance ratio* of an algorithm A is defined as

$$\mathrm{AAPR}(A) = \sup_F \limsup_{n\to\infty} \mathbb{E}\left[\frac{A(I_n^F)}{\mathrm{OPT}(I_n^F)}\right].$$

We prove that for the distributions and algorithms we consider, the asymptotic average performance ratio can usually also be expressed as

$$\sup_F \limsup_{n\to\infty} \frac{\mathbb{E}\left[A(I_n^F)\right]}{\mathbb{E}\left[\mathrm{OPT}(I_n^F)\right]}.$$

In the case of (class constrained) bin covering we have to replace $\sup \limsup$ by $\inf \liminf$.

Often, we can reduce the analysis of the asymptotic average performance ratio to a restricted class $\mathcal{P}$ of distributions, the so-called *perfect-packing distributions*. We say that F is a perfect-packing distribution if we can represent F in the following way: We assume that there are $m \in \mathbb{N}$ bins that are perfectly packed in the sense that the total size of the items in each bin is exactly 1 and the number of colors in each bin is at most k or at least k for class constrained bin packing or class constrained bin covering, respectively. We denote by ℓ the total number of items and the items are numbered consecutively from 1 to ℓ. The distribution F is obtained by drawing an index i uniformly at random from $\{1, \ldots, \ell\} =: [\ell]$ and choosing the item with the corresponding index. Analogously, a *perfect-packing instance* in the random-order model is an instance in which in the optimal solution all bins are perfectly packed in the above sense.

Now we introduce the second performance measure. For an instance I, we denote by I^σ a random instance, where the items in I are randomly permuted.

Let A be an algorithm for the (class constrained) bin packing problem. Then the *asymptotic random-order ratio* $\mathrm{RR}(A)$ of A is defined as

$$\mathrm{RR}(A) = \limsup_{\mathrm{OPT}(I)\to\infty} \frac{\mathbb{E}\left[A(I^\sigma)\right]}{\mathrm{OPT}(I)}.$$

In the case of (class constrained) bin covering we have to replace lim sup by lim inf.

For all considered problems, the asymptotic average performance ratio cannot be worse than the random-order ratio, i.e., for any algorithm A we have $1 \leq \mathrm{AAPR}(A) \leq \mathrm{RR}(A)$ and $1 \geq \mathrm{AAPR}(A) \geq \mathrm{RR}(A)$ for the (class constrained) bin packing problem and the (class constrained) bin covering covering problem, respectively (see [8] or the full version of the paper).

We also study the special cases of class constrained bin packing and covering, where we have *unit sized items*. In this special case, we are given a parameter $B \in \mathbb{N}$, and all items have size $1/B$. For convenience, we will scale the item sizes to 1 and the bin capacity to B in this case.

1.2 Related Work

There is vast body of literature on the classical versions of bin packing and bin covering. We discuss only the results that are most relevant for our article. Kenyon [10] introduced the notion of asymptotic random-order ratio for bin packing and proved that the asymptotic random-order ratio of the best-fit algorithm (BF) lies between 1.08 and 1.5, while its (worst-case) competitive ratio is well-known to be 1.7 [15, 5]. In contrast to this, Coffman et al. [9] showed that the random-order ratio of the next-fit algorithm (NF) equals its (worst-case) competitive ratio 2. Christ et al. [4] adapted the asymptotic random-oder ratio to bin covering and proved that the random-order ratio of the dual next-fit algorithm (DNF) is at most 0.8, which was later improved to 2/3 [8]. In [8], we proved that the asymptotic average performance ratio of DNF is $0.5 + \varepsilon$ for a small constant $\varepsilon > 0$ for every discrete distribution F. However, this lower bound does not carry over to the random-order ratio of DNF and no lower bound except for the trivial bound of 0.5 is known for this.

The class constrained bin packing problem has been introduced in [13] and studied in a sequence of papers [14, 16, 7]. All results so far concern the competitive ratio in the classical worst-case model. In the case of unit sized items there is a lower bound of 2 for the asymptotic competitive ratio that can be achieved and this bound is achieved by the first-fit algorithm (FF) and the algorithm CS [NF] (also called ColorSets) introduced in [14]. In the general case, for all values of k, there exists an online algorithm for class constrained bin packing with a competitive ratio of at most 2.63492 [7]. Also the competitive ratios of several other online algorithms have been analyzed [7, 16] and approximation schemes for the offline problem have been obtained [7].

The class constrained bin covering problem has been introduced by Epstein et al. [6]. Also this problem has not been studied in a probabilistic setting before. Epstein et al. consider only the case of unit sized items and they prove several results. They obtain a polynomial-time algorithm for the offline problem and an upper bound of $\left(\frac{(B-1)(B-k+1)}{B(B-k)+B-1} \cdot (\frac{B-k}{B-1} + H_{k-1})\right)^{-1}$ for the competitive ratio of any online algorithm. Furthermore, they prove that DNF is not competitive for class constrained bin covering and they introduce the algorithm COLOR&SIZE and prove that it is $\Omega(1/k)$-competitive. They also introduce the algorithm FF2 and prove that its competitive ratio is exactly $1/B$.

1.3 Our Contributions

All mentioned algorithms are described in detail in Section 1.4.

Classical Bin Packing and Covering We prove that for bin covering the simple greedy algorithm DNF achieves a random-order ratio of at least 0.501. While this is only a small improvement over the trivial bound of 0.5, it is the first bound that shows that DNF performs better in the random-order setting than in the worst case. This has already been conjectured in [4] and posed as an open problem. The conclusions in [4] also discuss the challenges in proving such a result. While the different bins covered by DNF are not independent in the random-order model, one main observation in our proof is that they are identically distributed. Given this observation, the proof relies on analyzing the expected overshoot of the first filled bin, where the overshoot is defined as the total size of the items assigned to that bin minus its capacity 1. We show that the expected overshoot is strictly less than 1. This proof strategy is analogous to our analysis of the asymptotic average performance ratio [8] but the technical details are quite different because instead of sampling with replacement the harder setting of sampling without replacement has to be analyzed.

Since the random-order ratio of the dual harmonic algorithm DH_k is 0.5 [4], this result separates DNF from DH_k in the random-order model, while their performance cannot be distinguished in a worst-case analysis. This is interesting because DH_k was designed to guard against pathological worst-case inputs and it is already discussed in [4] that one would expect DNF to perform better than DH_k on more realistic inputs. As an additional minor result, we prove that DNF and DH_k are also separated in terms of their asymptotic average performance ratio by showing that this ratio is 0.5 for DH_k while our lower bound of 0.501 for DNF also carries over to this setting.

In contrast to this, we show for bin packing that next-fit, worst-fit, and smart-next-fit do not perform better in the random-order setting and not even in the i.i.d. setting than in the worst case. (For the random-order ratio of next-fit this result was already known [9].)

Class Constrained Bin Packing and Covering We mention only the most interesting results. For class constrained bin packing we can show that there exists a sequence of algorithms whose asymptotic average performance ratios tend to $h_\infty \approx 1.691$. h_∞ is known in classical bin packing as the lower bound for bounded-space online-algorithms shown by C.C. Lee and D.T. Lee [12]. This is far better than the competitive ratio of the best known algorithm 2.635 for class constrained bin packing [7] and also beats the known lower bound for arbitrary deterministic algorithms of 1.717 shown in [2] for the special case $k = 2$.

When we consider the random-order model, we find out that several algorithms behave worse than in the case of i.i.d. sampling. Especially, we establish a lower bound of $10/9$ for the random-order ratio of all deterministic online-algorithms. As far as we know, this is the first lower bound for arbitrary algorithms w.r.t. to a probabilistic performance measure in the area of bin packing and bin covering and its variants.

Furthermore, we consider the special case of unit sized items. We observe again different behaviors of heuristics w.r.t. to the two considered performance measures. Especially, a large class of "natural" algorithms performs asymptotically optimal, if the items are drawn i.i.d.

For class constrained bin covering we investigate the behavior of DNF and FF2. We observe that the algorithms benefit a lot from random input – independently of the considered probabilistic performance measure. We provide bounds, which are logarithmic in k, for the performance of DNF w.r.t. both models. In the case of unit sized items we show that FF2 behaves asymptotically optimal in the random-order model, and therefore also for i.i.d. sampling. We use this result to establish a $1/3$-competitive algorithm in the random-order model for general item sizes.

The main tools for proving these results are

– Markov chain arguments (e.g. estimates for the stationary distribution and growth bounds for trajectories);
– couplings to compare stochastic processes and relating i.i.d. sampling with the random-order model;
– concentration inequalities for – possibly dependent – random variables.

An overview on the used concentration bounds will be given in the full version of the paper.

Intuitively one main reason why the two probabilistic measures lead to different results in the case of class constrained bin packing is that in the random-order model the number of different colors can grow with the length of the input sequence while it cannot grow arbitrarily with the input length if the items are drawn i.i.d. with respect to some fixed distribution.

1.4 Algorithms

Let us describe the algorithms that we analyze in more detail. We start with the (class constrained) bin packing problem. For this problem, the following algorithms are relevant for our results.

– Next-Fit (NF): At each point of time one bin is open. NF assigns each arriving item to the currently open bin if it can accommodate the item. Otherwise it closes the currently open bin and opens a new bin to which the

item is added. Here closing a bin means that no item will be assigned to this bin in the future anymore.
- First-Fit (FF): FF never closes a bin, i.e., it keeps all bins open and assigns each arriving item to the first bin that can accommodate it if such a bin exists. Otherwise it opens a new bin and adds the item to it.
- Best-Fit (BF): BF never closes a bin, i.e., it keeps all bins open and assigns each arriving item to the fullest bin that can accommodate it if such a bin exists. Otherwise it opens a new bin and adds the item to it.
- Worst-Fit (WF): WF never closes a bin, i.e., it keeps all bins open and assigns each arriving item to the bin with the most space remaining if this bin can accommodate it. Otherwise it opens a new bin and adds the item to it.
- Smart-Next-Fit (SNF): SNF works similarly to NF. It assigns each arriving item to the currently open bin Z if this bin can accommodate the item. Otherwise it opens a new bin Z' and adds the item to it. It retains as new current bin whichever of Z and Z' has the most space remaining.
- HARMONIC$_M$: HARMONIC$_M$ is an algorithm designed for classical bin packing. It partitions the interval $(0, 1]$ into the subintervals

  $$(0, 1/M], (1/M, 1/(M-1], \ldots, (1/2, 1].$$

  This partition induces also a partition of the set of items into M classes. HARMONIC$_M$ packs items from different classes into different bins and it runs NF independently for each class. That is, it packs exactly j items from the interval $(1/j + 1, 1/j]$ into a bin.
- CS [A]: A technique often used to generate algorithms for class constrained bin packing is the ColorSets-approach. The ColorSets-approach wants to apply an algorithm A, which is designed for classical bin packing, to class constrained bin packing. In order to do this, it groups the colors according to their first arrival in groups of size k and then applies A separately to each group. Popular examples are CS [NF], CS [FF] and CS [BF] (see e.g. [14, 16]).

While in NF there is only one and in HARMONIC$_M$ only M open bins at each point of time, in FF, BF, and WF all bins are kept open during the whole input sequence. We say that an algorithm is an *ℓ-bounded space algorithm* if on any input and at each point of time it has at most ℓ open bins.

Now we describe the relevant algorithms for the (class constrained) bin covering problem.

- Dual Next-Fit (DNF): DNF packs all arriving items into the same bin until the bin is filled. Then the next items are packed into a new bin until this bin is filled, and so on.
- FF2: The algorithm FF2 is for the class constrained bin covering problem with unit sized items only. It adds each arriving item to the first bin for which it is suitable. To define the notion of suitable, consider a bin that contains already items with $k - t$ different colors. If this bin contains fewer than $B - t$ items, every item is suitable. Otherwise, if the number of items

is exactly $B - t$, an item is only suitable if it has a color that is not yet contained in the bin.

- Dual Harmonic DH_M: The algorithm DH_M is the adaption of $\mathrm{HARMONIC}_M$ to classical bin covering. The interval $(0, 1]$ is partitioned into the subintervals $(0, 1/M), [1/M, 1/(M-1)), \ldots, [1/2, 1)$. This partition again induces also a partition of the set of items into M classes. DH_M packs items from different classes into different bins and it runs DNF independently for each class. That is, it uses exactly j items from the interval $[1/j, 1/(j-1))$ to cover a bin.

In Section 2 we discuss our results on classical bin packing and bin covering in detail, followed by Section 3 and 4 on class constrained bin packing and bin covering, respectively.

2 Classical Bin Packing and Covering

We start by showing that the trivial greedy algorithm for the classical bin covering problem performs strictly better in the random-order model than in the worst case. This statement confirms the conjecture given in [4]. Furthermore, to the best of our knowledge this is the first positive result on the random-order ratio of bin packing and covering (and its variations) since the celebrated result of Kenyon [10].

Theorem 1. *We have* $\mathrm{RR}(\mathrm{DNF}) \geq 1/2 + 1/1000$.

DNF is a monotone algorithm in the sense that decreasing the size of items or deleting them only does harm to the algorithm. We see this as follows: Let $I = (a_1, \ldots, a_n)$ and $I' = (a_1, \ldots, a_{i-1}, a'_i, a_{i+1}, \ldots, a_n)$ with $a'_i < a_i$. We simulate deleting an item by setting a'_i equal to zero. Let $f(a_j)$ denote the number of the bin a_j is assigned to if DNF performs on I and $f'(a_j)$ if DNF performs on I', respectively. If $j \leq i$ it is obvious that we have $f(a_j) = f'(a_j)$. If $j > i$ we can show via induction that $f(a_j) \geq f'(a_j)$. Therefore, we have $\mathrm{DNF}(I') \leq \mathrm{DNF}(I)$.

It follows from the monotonicity of DNF that we can assume without loss of generality that we deal with instances I that can be packed perfectly into $\mathrm{OPT}(I)$ bins. Especially, we have $\mathrm{OPT}(I) = S(I)$, where $S(I)$ denotes the total size of all items in I.

Since DNF is $1/2$-competitive, we know that the algorithm covers, independently of I^σ, at least $\lfloor \mathrm{OPT}(I)/2 \rfloor$ many bins. For $i \in [1 : \lfloor \mathrm{OPT}(I)/2 \rfloor]$ let $S_i(I^\sigma)$ denote the total size of items in the i-th covered bin if we apply DNF to I^σ. We define the *overshoot* for the i-th bin as $R_i(I^\sigma) := S_i(I^\sigma) - 1$.

Then, the proof of the theorem is based on two pillars. At first, we show that the overshoot is identically distributed:

Lemma 2. *The random variables* $R_i(I^\sigma)$*, where* $1 \leq i \leq \lfloor \mathrm{OPT}(I)/2 \rfloor$*, are identically distributed.*

Then, we can express the random-order ratio in terms of the overshoot. Let $W(I^\sigma)$ denote the total size of the items in the last bin, which is not covered. We have

$$\begin{aligned}\mathrm{OPT}(I) &= \mathrm{DNF}(I^\sigma)\cdot 1 + \sum_{i=1}^{\mathrm{DNF}(I^\sigma)} R_i(I^\sigma) + W(I^\sigma)\\ &= \mathrm{DNF}(I^\sigma) + \sum_{i=1}^{\lfloor \mathrm{OPT}(I)/2\rfloor} R_i(I^\sigma) + \sum_{i=\lfloor \mathrm{OPT}(I)/2\rfloor+1}^{\mathrm{DNF}(I^\sigma)} R_i(I^\sigma) + W(I^\sigma)\\ &\leq \mathrm{DNF}(I^\sigma) + \sum_{i=1}^{\lfloor \mathrm{OPT}(I)/2\rfloor} R_i(I^\sigma) + (\mathrm{DNF}(I^\sigma) - \lfloor \mathrm{OPT}(I)/2\rfloor) + 1\\ &\leq 2\,\mathrm{DNF}(I^\sigma) - \mathrm{OPT}(I)/2 + 2 + \sum_{i=1}^{\lfloor \mathrm{OPT}(I)/2\rfloor} R_i(I^\sigma).\end{aligned}$$

Applying expectation values to both sides and using the previous lemma, we obtain

$$\mathrm{OPT}(I) \leq 2\mathbb{E}\left[\mathrm{DNF}(I^\sigma)\right] - \mathrm{OPT}(I)/2 + 2 + \frac{1}{2}\,\mathrm{OPT}(I)\mathbb{E}\left[R_1(I^\sigma)\right].$$

It follows that

$$\frac{\mathbb{E}\left[\mathrm{DNF}(I^\sigma)\right]}{\mathrm{OPT}(I)} \geq \frac{3}{4} - \frac{1}{4}\cdot\mathbb{E}\left[R_1(I^\sigma)\right] - \frac{1}{\mathrm{OPT}(I)}. \tag{1}$$

The second pillar is to give an upper bound for the overshoot. A similar statement in case of items that are drawn i.i.d. was shown in our paper [8]. At that time we used elementary counting and covering arguments. This time we apply concentration inequalities that lead to a simplified proof with a stronger bound – even if the resulting bound is still close to the worst case.

Lemma 3. *Let $(I_j)_j$ be an arbitrary sequence of instances with*

$$\lim_{j\to\infty} \mathrm{OPT}(I_j) = \infty \quad \textit{and} \quad \mathrm{RR}(\mathrm{DNF}) = \liminf_{j\to\infty} \frac{\mathbb{E}\left[\mathrm{DNF}(I_j^\sigma)\right]}{OPT(I_j)}.$$

Then, if j is sufficiently large we have

$$\mathbb{E}\left[R_1(I_j^\sigma)\right] \leq 1 - \frac{34}{100e^3}\cdot\left(1-\exp\left(-\frac{121}{420}\right)\right) \approx 0.99576.$$

Combining this upper bound with (1) yields a lower bound of approximately 0.501 for DNF if the items arrive in random order. The lower bound is complemented by an upper bound of 2/3 for the random-order ratio in [8].

The behavior of DNF in the random-order model is in contrast to the behavior of NF in classical bin packing: Coffman et al. [9] showed in 2008 that

RR(NF) = 2, which is equal to its worst-case performance. We will refine this statement and show that the algorithms NF, SNF and WF for classical bin packing do not behave better than in the worst case even if the items are sampled in an i.i.d. manner.

Proposition 4. *For* $\mathrm{A} \in \{\mathrm{NF}, \mathrm{SNF}, \mathrm{WF}\}$ *we have* $\mathrm{AAPR(A)} = \mathrm{RR}(A) = 2$.

Furthermore, also the dual harmonic algorithm DH_k for bin covering does not improve on the worst case if the items are drawn i.i.d.

Proposition 5. *We have* $\mathrm{AAPR}(\mathrm{DH}_k) = 1/2$.

3 Class Constrained Bin Packing

3.1 Results for General Item Sizes

As already mentioned in the introduction a popular approach to deal with class constrained bin packing is the ColorSets-approach. We show that there are algorithms based on this approach, that behave remarkably well in the case of items that are drawn i.i.d. Let $t_1 = 1$ and $t_{i+1} = t_i(t_i + 1)$. We set $\sum_{i=1}^{\infty} \frac{1}{t_i} =: h_\infty \approx 1.691$. h_∞ is known in bin packing as the famous lower bound for bounded-space online algorithms for the classical bin packing problem proved by C.C. Lee and D.T. Lee in [12]. The following statement shows that there is a sequence of ColorSets-based algorithms whose performance tends to h_∞. Furthermore, no algorithm based on this idea could behave better.

Theorem 6. *Let $\epsilon > 0$ be arbitrary. Choosing M sufficiently large, we have*

$$\mathrm{AAPR}(\mathrm{CS}\,[\mathrm{HARMONIC}_M]) \leq h_\infty + \epsilon.$$

Furthermore, let A be an arbitrary algorithm for classical bin packing, then we have

$$\mathrm{AAPR}(\mathrm{CS}\,[A]) \geq h_\infty.$$

To show the upper bound we want to compare the performance of the algorithm $\mathrm{CS}\,[\mathrm{HARMONIC}_M]$ with the performance of the algorithm $\mathrm{HARMONIC}_M$ in the case we ignore the colors. Lee and Lee proved in [12] that the asymptotic competitive ratio of $\mathrm{HARMONIC}_M$ for classical bin packing is upper bounded by $h_\infty + \epsilon$ for $\epsilon > 0$ arbitrary, if we choose M sufficiently large.

We observe that the number of opened bins differs by at most $M \cdot Q_F(n)$ many bins. Here, $Q_F(n)$ denotes the number of different drawn colors among the first n drawn items. But $Q_F(n)$ grows sublinearly in expectation. Therefore, asymptotically the performance of $\mathrm{CS}\,[\mathrm{HARMONIC}_M]$ coincides with the performance of $\mathrm{HARMONIC}_M$ in classical bin packing.

For the lower bound we construct a distribution F as follows: The multiset of items $\mathcal{I}$ contains *large* items and *small* items. The large items are as follows: For each $(i, j) \in [k]^2$ there will be an item of size $\frac{1}{t_i+1} + \beta$ of color $(i-1)k + j$, where $\beta > 0$ is sufficiently small. Furthermore, $\mathcal{I}$ contains lots of small items of different colors. Choosing the small items appropriately the order of the first arrival of the colors is $1, \ldots, k^2$ with high probability. Then, $\mathrm{CS}\,[A]$ will pack the items of size $\frac{1}{t_i+1} + \beta$ separately. The statement then follows from the work of Lee and Lee.

In the random-order model things are more complicated: We can show that it is not possible for ColorSets-based algorithms and FF to achieve a performance of h_∞.

Proposition 7. *Let A be an arbitrary algorithm for classical bin packing. Then we have* $\mathrm{RR}(\mathrm{CS}\,[A]) \geq 2$*, even in the special case $k = 2$. Furthermore, we show that* $\mathrm{RR}(\mathrm{FF}) \geq 2$.

Furthermore, we are able to establish a non-trivial lower bound for the performance of an arbitrary online-algorithm in the random-order model. As far as we know this is the first asymptotic lower bound for a probabilistic performance measure in the field of bin packing/bin covering and its variants.

Theorem 8. *Let A be an arbitrary online-algorithm for class constrained bin packing. Then we have* $\mathrm{RR}(A) \geq 10/9$.

The idea of the proof is to construct an instance I that contains items of colors of the two types *small* and *large*. The total size of all items of a small color is close to zero, while the total size of all items of a large color is close to 1. Furthermore, for each color there are lots of tiny items. If the items arrive in random order, there will be lots of tiny items in the beginning. This forces the algorithm to decide which colors to put in the same bin without the possibility to learn, which color is small and large. Therefore, there will be a constant fraction of bins opened by the algorithm that are nearly empty.

3.2 The Special Case of Unit Sized Items

Now we want to consider the case of unit sized items. We observe the same behavior of algorithms as in the case of general item sizes. If the items are drawn i.i.d. a large class of natural algorithms performs asymptotically optimal, but in the random-order model their performance is worse.

Proposition 9. $\mathrm{CS}\,[\mathrm{NF}]$ *and every algorithm that opens a new bin only if it is forced, is optimal if the items are drawn i.i.d.*

Proposition 10. *We have* $\mathrm{RR}(\mathrm{CS}\,[\mathrm{NF}]) = 2$.

Proposition 11. *We have* $\mathrm{RR}(\mathrm{FF}) \geq 1.5$.

Finally, we want to mention that bounded-space algorithms perform poorly for class constrained bin packing, even on random input with unit sized items. This is in contrast to classical bin packing.

Proposition 12. *Consider class constrained bin packing with unit sized items and parameters B and k. Let A be an arbitrary bounded-space online-algorithm. Then we have* $\mathrm{RR}(A), \mathrm{AAPR}(A) \in \Omega(B/k)$.

4 Class Constrained Bin Covering

We start with a simple result on bounded-space algorithms for the class constrained bin covering problem. While in the classical bin covering problem, even the trivial 1-bounded space algorithm is best possible w.r.t. the competitive ratio, in the class constrained variant those algorithms behave poorly.

Proposition 13. *Let A be a bounded-space algorithm. Then A is not competitive w.r.t. the competitive ratio.*

In general, there is a logarithmic upper bound in k for the performance of online algorithms. That is, the online version of this problem is strictly more difficult than the classical problem. The following statement is a slight improvement on the corresponding result in [6]. The proof uses the same technique, but adjusts the choice of scenarios.

Proposition 14. *The competitive ratio of any deterministic online algorithm is at most $\left(H_{k-1} + 1 - \frac{k-1}{B}\right)^{-1}$. If $B = k$ this yields an upper bound of H_k^{-1}.*

Now we begin to investigate the performance of heuristics w.r.t. probabilistic performance measures. We start with the simple 1-bounded space algorithm DNF.

Theorem 15. *For unit sized items we have* $\mathrm{RR}(\mathrm{DNF}) \in \Theta(\log(k)^{-1})$*. For general item sizes we have* $\mathrm{AAPR}(\mathrm{DNF}) \in \Theta(\log(k)^{-1})$*.*

We see that in class constrained bin covering DNF benefits a lot from probabilistic input. We have seen that bounded-space algorithms cannot be competitive in the worst case and that there exists a logarithmic upper bound in k for the performance of arbitrary online algorithms. If unit sized items arrive in random order, even the simple 1-bounded-space algorithm DNF achieves a competitive ratio that matches this bound.

Surprisingly it turns out that a FirstFit-approach is even optimal if unit sized items arrive in random order:

Theorem 16. *We have* $\mathrm{RR}(\mathrm{FF2}) = 1$*.*

To prove the theorem at first we observe that the algorithm is monotone. Therefore, we can assume that we deal with instances I that cover $\mathrm{OPT}(I)$ bins perfectly. Furthermore, the monotonicity of FF2 allows us to restart the algorithm several times starting again with a single empty bin. Using this technique, we divide the input I into $|I|^{2/3}$ many sub-inputs containing each $|I|^{1/3}$ many items. Then we show that we can assume that the items in the sub-input are drawn i.i.d. So we reduce the analysis in the random-order model to the case of i.i.d. sampling.

Then, we construct a comparison Markov chain, which lower bounds the number of covered bins of FF2 on the sub-inputs. The idea of the comparison chain is as follows: We simulate the behavior of a modified FF2-algorithm on a special distribution F. We obtain F from drawing an item with color 1 with probability $(B-k+1)/B$ and an item with color i with probability $1/B$, where $i \in \{2,\dots,k\}$. The modified FF2-algorithm treats the first $B-k+1$ items in a bin as items of color 1. We can show that FF2 on an arbitrary (perfect-packing) distribution F covers in expectation at least as many bins as the modified algorithm in our comparison chain.

Finally, we use tools from the field of Markov chains to show that the growth of open bins in the comparison chain is only sublinear in the number of items. Plugging the pieces together we obtain that $\mathrm{RR}(\mathrm{FF2}) = 1$.

Theorem 16 allows us to give a simple online algorithm, which is $1/3$-competitive for general item sizes if the items arrive in random-order.

Corollary 17. *There exists an $1/3$-competitive algorithm in the random-order model for the class constrained bin covering problem with general item sizes.*

This also gives us an easy randomized algorithm for the offline case. To the best of our knowledge this is the first offline algorithm presented for this problem.

Corollary 18. *There is a randomized asymptotic $1/3$-competitive algorithm for class constrained bin covering in the offline case.*

5 Conclusion and Further Research

We showed that the DNF algorithm for bin covering performs better in the random-order model than in the worst case by providing a lower bound of 0.501 on its random-order ratio. This is the first bound better than the trivial bound of 0.5. We think that it is an interesting open problem to close the gap between the lower and upper bounds and we conjecture that the random-order ratio of DNF equals the upper bound 2/3.

Furthermore, we studied class constrained bin packing and class constrained bin covering in the random-order model and i.i.d. sampling. We saw that in many cases heuristics benefit from the probabilistic input and can beat several worst-case bounds. In class constrained bin packing we observed different performances of algorithms in the two probabilistic models. The random-order model allows us to restrict the number of similar items and to force a linear number of different item types, while in the i.i.d. model the number of different item types grows only sublinearly. This difference plays an important role in class constrained bin packing, while it is not relevant in class constrained bin covering. As far as we know in bin packing and bin covering there are no other results known, where the performance of algorithms differ with respect to the investigated performance measures. It would be interesting to find further examples in this area, where algorithms perform differently, and to give formal explanations why both performance measures coincide in other bin packing/bin covering variants.

References

1. Luitpold Babel, Bo Chen, Hans Kellerer, and Vladimir Kotov. Algorithms for on-line bin-packing problems with cardinality constraints. *Discrete Applied Mathematics*, 143:238–251, 2004.
2. János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. Lower bounds for several online variants of bin packing. *CoRR*, abs/1708.03228, 2017.
3. János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. Online Bin Packing with Cardinality Constraints Resolved. In *Proceedings of the 25th European Symposium on Algorithms (ESA)*, volume 87, pages 10:1–10:14, 2017.
4. Marie G. Christ, Lene M. Favrholdt, and Kim S. Larsen. Online bin covering: Expectations vs. guarantees. *Theoretical Computer Science*, 556:71–84, 2014.
5. György Dósa and Jirí Sgall. Optimal analysis of best fit bin packing. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 429–441, 2014.
6. Leah Epstein, Csanád Imreh, and Asaf Levin. Class constrained bin covering. *Theory of Computing Systems*, 46(2):246–260, 2010.
7. Leah Epstein, Csanád Imreh, and Asaf Levin. Class constrained bin packing revisited. *Theoretical Computer Science*, 411(34-36):3073–3089, 2010.
8. Carsten Fischer and Heiko Röglin. Probabilistic analysis of the dual next-fit algorithm for bin covering. In *Proceedings of the 12th Latin American Symposium on Theoretical Informatics (LATIN)*, pages 469–482, 2016.
9. Edward G. Coffman Jr., János Csirik, Lajos Rónyai, and Ambrus Zsbán. Random-order bin packing. *Discrete Applied Mathematics*, 156(14):2810–2816, 2008.
10. Claire Kenyon. Best-fit bin-packing with random order. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 359–364, 1996.
11. Kenneth L. Krause, Vincent Shen, and Herb D. Schwetman. Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems. *J. ACM*, 22(4).
12. C. C. Lee and D. T. Lee. A simple on-line bin-packing algorithm. *J. ACM*, 32(3).
13. Hadas Shachnai and Tami Tamir. Polynomial time approximation schemes for class-constrained packing problems. *Journal of Scheduling*, 4(6).
14. Hadas Shachnai and Tami Tamir. Tight bounds for online class-constrained packing. *Theoretical Computer Science*, 321(1):103–123, 2004.
15. Jeffrey D. Ullman. *The Performance of a Memory Allocation Algorithm.* Technical report 100 (Princeton University. Dept. of Electrical Engineering. Computer Sciences Laboratory). Princeton University, 1971.
16. Eduardo C. Xavier and Flávio K. Miyazawa. The class constrained bin packing problem with applications to video-on-demand. *Theoretical Computer Science*, 393(1-3):240–259, 2008.