

# Lower Bounds for Several Standard Bin Packing Algorithms in the Random Order Model

Leah Epstein 

Department of Mathematics, Faculty of Natural Sciences, University of Haifa, Israel

Asaf Levin 

Faculty of Data and Decisions Science, The Technion, Haifa, Israel

---

## Abstract

We consider the performance of standard bin packing algorithms in the random order model. We provide an improved lower bound of 1.15582656 on the asymptotic approximation ratio of Best Fit (BF) for randomly ordered inputs. We also show lower bounds on the asymptotic approximation ratio for two bounded space bin packing algorithms in this model, namely for 2-BF and 2-FF. These are well-studied bounded space algorithms and the first one has the same asymptotic worst-case performance as BF. However, the resulting lower bounds on their performances in the random order model are much higher than that of BF.

**2012 ACM Subject Classification** Theory of computation → Scheduling algorithms

**Keywords and phrases** Bin packing, Best Fit, Random order, Bounded space algorithms

**Digital Object Identifier** 10.4230/LIPIcs.WADS.2025.26

**Funding** *Asaf Levin*: Partially supported by ISF – Israel Science Foundation grant number 1467/22.

## 1 Introduction

Bin packing (BP) or standard bin packing [31, 30, 36, 6, 7] is a combinatorial optimization problem introduced in the early 1970's. In this problem, a set of items of positive rational sizes of at most 1 is to be partitioned into subsets of total sizes not exceeding 1. The subsets are called bins, the process of assigning an item to a bin is called packing, and the goal is to minimize the number of (non-empty) bins. The variant where the input is a set of items is called (standard or classic) offline BP. In the online variant of standard BP, items are presented one by one, such that each item has to be packed into a bin before the next item is presented. Many natural algorithms for offline BP process items as a list, and can be seen as online algorithms.

Since standard BP is NP-hard (and so are many other versions of BP), it is frequently studied with respect to approximation. The approximation ratio of an algorithm for a minimization problem and an input is the ratio between the costs of the algorithm and of an optimal solution for this input. The absolute approximate ratio of an algorithm is the worst-case approximation ratio over valid inputs. The asymptotic approximation ratio, which is the standard measure for bin packing type problems, is the superior limit of the sequence of approximation ratios with fixed optimal costs (when these fixed costs grow without bound). For online algorithms, the approximation ratio is also called *competitive ratio*, but the definitions are identical. Since the studied algorithms are of interest both as offline algorithms and as online algorithms, we use the terms *asymptotic approximation ratio* and *absolute approximation ratio*.

Here, we consider online bin packing algorithms with respect to randomly ordered inputs and the asymptotic approximation ratio, and we briefly comment on the absolute approximation ratio of one algorithm. For worst-case analysis, the best possible asymptotic approximation ratio of an online algorithm is in the interval  $[1.54278, 1.57829]$  [6, 7] and the



© Leah Epstein and Asaf Levin;

licensed under Creative Commons License CC-BY 4.0

19th International Symposium on Algorithms and Data Structures (WADS 2025).

Editors: Pat Morin and Eunjin Oh; Article No. 26; pp. 26:1–26:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

best absolute approximation ratio is  $\frac{5}{3}$  [8]. Bin packing was also analyzed with respect to average case analysis for various input distributions including  $U[0, 1]$  [17, 9, 37, 32, 5], where these models are different from our model, which is defined below.

**The algorithms that we analyze.** The Best Fit (BF) algorithm [31, 20, 38] is defined as follows. It starts with an empty set of active bins. Whenever a new item is revealed, it computes the subset of active bins in which the new item can be added without exceeding the bound of 1 on the total size of items including the new item. If this subset is non-empty, the algorithm picks a bin of maximum total size of items from the subset and packs the item there. If the subset is empty, it opens a new (empty) bin that is defined as an active bin and it packs the item there. The asymptotic (and absolute) asymptotic approximation ratio of BF is 1.7 [31, 20].

In addition to BF, we analyze two well-studied bounded space algorithms, namely 2-BF and 2-FF. Bounded space algorithms also have an active set of bins, but their number is bounded from above by a constant integer parameter. Thus, a bounded space algorithm may close a bin in order to open a new active bin, in which case the closed bin cannot be used again. Next Fit (NF) [30] is a greedy bounded space algorithm that has at most one active bin. If it has no active bin, NF opens a new bin which becomes active, and the item is packed into that bin. If NF has an active bin and the item can be packed into that bin with respect to its size, this is done. Otherwise, if there is an active bin, it is closed, and a new bin is opened, such that it receives the new item and becomes active.

Bounded space algorithms with larger numbers of active bins were studied as well [36, 43, 41, 40, 18, 44, 42]. The algorithm 2-BF has at most two active bins at all times, and initially it has no active bins. When a new item is presented, it is packed into an active bin with the maximum already packed total size (where it fits). If this is not possible and the number of active bins is 2, the most loaded active bin is closed (and it will not be able to receive new items), and (no matter whether a bin was closed or not) a new active bin is opened for the new item. A related algorithm is called 2-FF, and the difference is that in the case where there are two active bins such that each one of them can receive the new item, the bin that was opened first is used (and not the fullest bin). The latter algorithm can be seen as the bounded space version of First Fit (FF) [31], which packs every item into the bin of minimum index where it can be added. The two algorithms were analyzed with respect to the worst-case asymptotic approximation ratio. The asymptotic approximation ratio of 2-BF is 1.7 [19], as for BF (and FF) [31, 20], and the asymptotic approximation ratio of 2-FF is 2 (as for NF) [40, 44].

**Randomly ordered inputs.** The random order model was introduced for comparison and for contrast with worst-case analysis. In this model, the expected cost or profit over all permutations for an input is compared to the optimal offline cost or profit. For bin packing, this means that all  $n!$  permutations of a set of  $n$  items are considered with a uniform distribution over the permutations (which are input sequences). For an offline optimal solution, there is no meaning to the permutation, since the input of an offline bin packing algorithm is a set of items rather than a sequence. Such inputs were studied for various problems such as graph coloring [10, 25] and scheduling [2, 1], and other problems [33, 39, 14, 35, 27, 4, 28, 26]. Here, we study it with respect to bin packing [34, 16, 3, 5, 29]. We follow previous work and use a stochastic process for presenting lower bounds on the performance guarantee in the random order model. This stochastic process is a Markov chain with a finite number of states whose transition probabilities are provided by the family of permutations of a common

set of items. This Markov chain allows us to compute the equilibrium probabilities of the states using standard approaches, and these equilibrium probabilities allow us to find the resulting lower bound in a straightforward computation. Thus, the crux is to define the input set of items and the corresponding set of states and transition probabilities of the Markov chain. Note that a related model where algorithms are compared directly based on inputs with their random permutations and not with comparison to an optimal offline solution was studied for several problems including bin packing [11, 21, 15, 13, 12].

Kenyon [34] started the study of BF with respect to randomly ordered inputs. She proved a lower bound of approximately 1.08 on the asymptotic approximation ratio of BF with randomly ordered inputs, and wrote that 1.15 may be a lower bound. Many years later, Albers, Khan, and Ladewig [3] improved the lower bound to 1.1037, and finally Hebbar, Khan, and Sreenivas [29] improved the lower bound to 1.144. Kenyon [34] also proved the surprising result that the asymptotic approximation ratio of BF in this model is at most 1.5. By using various ingredients, this result was improved to slightly below 1.5 [29]. For the special case of items of sizes in  $(\frac{1}{4}, \frac{1}{2}]$ , an upper bound of approximately 1.4941 on the asymptotic approximation ratio of BF for randomly ordered inputs was shown [5].

Randomly ordered inputs with items larger than  $1/3$  (inputs with large items) were studied by Albers, Khan, and Ladewig [3], who showed that the asymptotic approximation for this case is at most 1.25. It was proved afterwards [5] that the asymptotic approximation ratio for this case is in fact 1. Interestingly, the absolute ratio for general inputs and even for inputs with large items is not 1. Lower bounds of 1.3 and 1.2, respectively, were shown [3] for the absolute approximation ratio for general inputs and for items of sizes above  $\frac{1}{3}$ , respectively (where for the last case an upper bound of  $\frac{21}{16} = 1.3125$  on the absolute approximation ratio of BF was proved in the same work).

Coffman et al. analyzed NF for BP with random order [16], and showed that the asymptotic competitive ratio is unchanged (as opposed to the situation with BF [34]). The main idea for the lower bound proof is to use the worst-case input of NF that alternates between items of size  $\frac{1}{2}$  and very small items of size  $\epsilon > 0$ . It is possible to add multiple items of size  $\epsilon$  instead of just one between every two items of size  $\frac{1}{2}$ , and applying NF on almost every permutation of this input will result in bins with one item of size  $\frac{1}{2}$  and many items of size  $\epsilon$ , and a very small number of bins with two items of size  $\frac{1}{2}$ .

Bin packing has different variants. A dual problem to BP is bin covering (BC) (see for example [13, 22, 23]), which is a maximization problem. BC can also be offline or online. The input is the same as for bin packing, but subsets should be covered in the sense that total sizes of bins that count towards the objective should be at least 1. Bin covering and other related problems were studied with respect to inputs permuted in a random order [13, 22, 23, 24]. Fischer [24] presented algorithms of asymptotic approximation ratios that are 1 or close to 1 using definitions of the analysis of the type that we perform here, though those are not natural algorithms, and they are randomized.

**Our results.** We provide lower bounds on the asymptotic approximation ratio for three online algorithms for standard bin packing. Two of these algorithms are the bounded space algorithms 2-BF and 2-FF, and the third algorithm is BF, which was already studied with respect to random orders. We provide an improved lower bound for BF, and much higher lower bounds for algorithms whose performance is the same as BF (2-BF) and not much worse than BF (2-FF) in terms of worst-case analysis. Our lower bound for Best Fit is 1.15582656, while the previous bound was 1.1440 [29]. Our lower bound for 2-BF is  $\frac{4}{3}$ , and our lower bound for 2-FF is above 1.41. The last two bounds for the bounded space

algorithms are based on common input sets of items. We start our study by considering this type of input and the resulting bounds for 2-BF and 2-FF in Section 2. Then, we turn our attention to BF, and present our lower bound in Section 3. Our lower bound for the asymptotic approximation ratio of BF is based on Markov chains on finite sets of item sizes (and so are the other lower bounds). The cardinalities of the sets of item sizes are (positive) integer parameters of our construction. Previous lower bounds for BF were also based on the approach of Markov chains [34, 3, 29] (as described above), but there was a single (constant cardinality) set of items for each proof (with at least two and at most seven item sizes) rather than a class of sets of sizes. Our lower bound for BF holds even for items of sizes in  $(\frac{1}{4}, \frac{1}{2}]$ , which is the special case that was studied by Ayyadevara et al. [5] with respect to an upper bound for BF, while the current best lower bound for this special case is the one of Kenyon [34]. In Section 4, we prove a new lower bound on the absolute approximation ratio of BF for randomly ordered inputs, improving the lower bound of 1.3 [3] to 1.5. A forthcoming full version will contain additional details and omitted proofs.

## 2 Lower Bounds for Two Bounded Space Algorithms

We start with defining the common new input for the lower bounds of 2-BF and 2-FF and common characteristics of the two Markov chains. Then we will show how to construct a chain for proving a lower bound on the asymptotic approximation ratio for this input with random order for 2-BF. The resulting proof for 2-BF is a complete analytical analysis of the state probabilities at equilibrium. Then, we turn our attention to the analysis of a corresponding lower bound for 2-FF. This last bound is computed using a computer assisted program to solve a (large) system of linear equations. We augment the last computer assisted proof with a very close bound derived using an analytic (and complete) proof.

### 2.1 The Common Input and Common Aspects of the Two Markov Chains

We define an input for these algorithms. There will be large, medium, and small items. Let  $n > 0$  be a large integer, and let  $N = 2kn$ , where  $k \geq 1$  is an integral parameter. The total number of items will be  $N^N$ . Let  $\varepsilon < \frac{1}{N^{N+3}}$ . A small item has size  $N^2 \cdot \varepsilon$  and the probability for such an item is  $\frac{N^{N-1}-1}{N^{N-1}}$ . The expected number of items that are not small is  $N$ . The remaining probabilities (for medium and large items) will be equal, and there are  $2k$  item sizes, where there are  $k$  sizes of large items and  $k$  sizes of medium items. For  $1 \leq i \leq k$ , an  $i$ -large item has size  $\frac{1}{2} + i \cdot \varepsilon$ , and these items (for all values of  $i$ ) are called large. For  $1 \leq i \leq k$ , an  $i$ -medium item has size  $\frac{1}{2} - i \cdot \varepsilon$ , and these items (for all values of  $i$ ) are called medium. A feasible solution (that provides an upper bound on the optimal cost) assigns all small items to a single bin. It also assigns every  $i$ -medium item with a  $i$ -large item, whenever this is possible. The remaining items are packed into bins alone, and the resulting number of bins is approximately  $k \cdot n + o(n)$ . See [29] for details regarding the property that this is indeed the optimal cost for inputs with a constant number of input sizes. The articles [3, 29] also show precisely that using an input drawn from a distribution on a discrete set of item sizes also gives a lower bound on the asymptotic approximation ratio of randomly ordered items.

Given an integer parameter  $k \geq 1$ , the Markov chains we define in what follows consist of  $2k + 1$  states. There is an initial state denoted by  $A$ ,  $k$  states denoted by  $B_1, B_2, \dots, B_k$  and  $k$  states denoted by  $C_1, C_2, \dots, C_k$ . For each state, the transition probabilities are integer multiples of  $\frac{1}{2k}$  such that the sum over all transitions out of each state is 1 for the state.

We define the transitions of each state where we write these values multiplied by  $2k$ , then we write the probability of each state explicitly and show that they are consistent with the transitions. Since every transition corresponds to the arrival of one item out of specific types in the constructions, we discuss even transitions for which the state is unchanged.

Our Markov chains will have a well-defined equilibrium. We use the notation  $p(A)$ ,  $p(B_i)$  (for  $1 \leq i \leq k$ ), and  $p(C_i)$  (for  $1 \leq i \leq k$ ) for the equilibrium probabilities of the states. We also use  $p(B) = \sum_{i=1}^k p(B_i)$  and  $p(C) = \sum_{i=1}^k p(C_i)$ .

## 2.2 Analysis for 2-BF

We first define the transition probabilities of the Markov chain that we use. Next, we will use those transition probabilities to obtain the corresponding equilibrium probabilities. Last, we show that this Markov chain allows us to prove our lower bound of the performance guarantee of 2-BF in the random order model.

**The transition probabilities.** These probabilities are given as follows.

- For  $A$ , there is one transition to each other state with value 1 (that is, with probability  $\frac{1}{2k}$ ). The value of the transition to  $A$  is zero.
- For  $B_i$  (where  $1 \leq i \leq k$ ) the value for the transition to  $A$  is  $i + k$ . The value for the transition back to  $B_i$  is  $k - i$  (so for  $i = k$  it is zero), and for  $B_j$  such that  $j \neq i$  the value of the transition to  $B_j$  is zero. The values of all remaining transitions (to states of the form  $C_\ell$ ) are also equal to zero.
- For  $C_i$  (where  $1 \leq i \leq k$ ) the value for the transition to  $A$  is  $k - i + 1$ . The corresponding value for  $B_j$  is 1 if  $j < i$ , and otherwise it is zero. The corresponding value for  $C_j$  is 1 if  $j < i$ , it is  $k - i + 1$  if  $j = i$ , and it is zero if  $j > i$ .

Observe that this chain has a unique equilibrium (i.e., the equilibrium the chain tends to when the number of transitions grows without bound does not depend on the initial state).

**The equilibrium probabilities.** We show that the following probabilities satisfy the requirements, and since there is a unique solution, they give this solution. We define the probabilities based on  $p(A)$  and ensure that the sum of probabilities is 1.

► **Lemma 1.** Let  $\lambda_k = \sum_{j=1}^k \frac{1}{(k+j)^2}$ . We have the three following claims.

1. It holds that

$$p(A) = p(C) = \frac{1}{2 + 2k\lambda_k}.$$

2. For  $1 \leq i \leq k$ , it holds that

$$p(C_i) = \frac{2k \cdot p(A)}{k + i - 1} - \frac{2k \cdot p(A)}{k + i} = \frac{2k \cdot p(A)}{(k + i - 1) \cdot (k + i)}.$$

3. For  $1 \leq i \leq k$ , it holds that

$$p(B_i) = \frac{2k \cdot p(A)}{(k + i)^2} > 0.$$

Observe that for large values of  $k$  it holds that  $\lambda_k \approx \int_k^{2k} \frac{1}{x^2} dx$ , and the value of this definite integral is  $\frac{1}{2k}$ . Thus,  $p(A) = \frac{1}{2 + 2k\lambda_k}$  is approximately equal to  $\frac{1}{3}$  when  $k$  grows without bound.

**The usage of the Markov chain for the analysis of 2-BF.** Our goal is to prove the following theorem.

► **Theorem 2.** *The asymptotic approximation ratio of 2-BF on randomly permuted inputs is at least  $\frac{4}{3} \approx 1.33333$ .*

**Proof.** The small items are sufficiently small such that all of them would fit into a bin together with one medium item or one large item. However, once a small item is packed with an item that is not small, since  $N^2\epsilon > 2k\epsilon$ , the bin cannot contain another item that is not small. We use the property of the packing that after a bin receives one medium or large item and at least one small item, it will only receive small items. On the other hand, by the same argument, a bin that contains two items that are not small (two medium items or a medium item and a large item) cannot contain any small items. Thus, such a bin cannot contain any additional items, and with a slight abuse of notation we assume that it is closed when it receives the second item.

With high probability it holds that there is at least one small item between every two items that are not small, and there is at least one small item in the beginning of the input. We assume the process of arrival of items and packing them by 2-BF has the following form. Consider the prefix of the input before and including the first item that is not small. This bin has a total size above  $\frac{1}{2} + k\epsilon$  after the prefix arrives. At this time, there is one active bin, and this bin can only receive small items. From this time on, there will be either one active bin or two active bins. There will always be an active bin that can only receive small items, and if there is a second active bin, it will have one item, where this item is not small. State  $A$  will correspond to having a single active bin. Transitions will be defined for times that an item that is not small arrives (after the prefix).

The other  $2k$  states are defined as follows. For each such state there is one bin that already has one item that is not small and at least one small item, so it can only receive small items, and one bin that contains a single item, where this item is medium or large. State  $B_i$  (for  $1 \leq i \leq k$ ) is the state where the second bin has an  $i$ -medium item, and state  $C_i$  (for  $1 \leq i \leq k$ ) is the state where the second bin has an  $i$ -large item.

If the stochastic process is currently in state  $B_i$  and a  $j$ -medium item arrives, the new item is packed into the bin that has no small items, and there is a transition to state  $A$ . If the process is currently in state  $B_i$  and a  $j$ -large item arrives, in the case  $j \leq i$ , the new item is packed into the bin that has no small items, and there is a transition to state  $A$ . In the case  $j > i$ , the bin with small items (which has the largest total size) is closed and the new item is packed into a new bin. One of the two bins will receive at least one small item (with high probability) before the next medium or large item arrives and this is the bin with the large item. Thus, in this case the transition is to the same state ( $B_i$ ).

If the process is currently in state  $C_i$  and a  $j$ -large item arrives, the bin with small items (which has the largest total size) is closed and the new item is packed into a new bin. One of the two bins will receive at least one small item (with high probability) before the next medium or large item arrives and this is the bin with the larger large item. This is the new bin if  $j \geq i$ , and otherwise it is the bin with the  $i$ -large item. The state remains  $C_i$  if  $j \geq i$ , and otherwise the transition is to the state  $C_j$ . If the process is currently in state  $C_i$  and a  $j$ -medium item arrives, the transition is to state  $A$  if  $j \geq i$ . Otherwise, the bin with at least one small item is closed, and the new item is packed into a new bin. The larger item packed alone in a bin is the large item, and therefore in the case  $j < i$  the new state is  $B_j$ .

The resulting chain is thus the one analyzed earlier in Lemma 1. The expected cost of 2-BF is  $1 - p(A)$  times the number of items that are not small. The expected cost of an optimal solution is  $\frac{1}{2}$  times the number of items (that are not small). For  $k$  growing without bound we find a lower bound of  $\frac{4}{3}$ . ◀

### 2.3 Analysis for 2-FF

Next, we discuss the difference with 2-FF. Since the closing rule is unchanged, the algorithm will close the bin that has a small item for this algorithm as well. The states are therefore unchanged, but the bin which will start receiving small items out of the two bins that have one item each (where these items are not small) is the older one. There is no change in the transitions to state  $A$ , but in cases where the transition is to another state, it is always to  $C_j$  when the new item is  $j$ -large, and to  $B_j$  when the new item is  $j$ -medium. These changes are reflected in the proof of the following theorem.

► **Theorem 3.** *The asymptotic approximation ratio of 2-FF on randomly permuted inputs is at least 1.41318399.*

The bound in the last theorem is proved using a computer assisted solution of a system of linear equations. In the full version of this work, we provide an analytic proof showing a slightly weaker bound of  $\frac{8355970560}{5918778241} \approx 1.41177$ .

**Proof.** We again use a Markov chain with  $2k + 1$  states. The transition probabilities of this chain are given as follows.

- For  $A$ , there is one transition to each other state with probability  $\frac{1}{2k}$ . The probability of the transition to  $A$  is zero.
- For  $B_i$  (where  $1 \leq i \leq k$ ) the probability for the transition to  $A$  is  $\frac{i+k}{2k}$ . The probability for the transition to  $C_j$  for  $j > i$  is  $\frac{1}{2k}$ . All other transition probabilities are equal to zero.
- For  $C_i$  (where  $1 \leq i \leq k$ ) the probability for the transition to  $A$  is  $\frac{k-i+1}{2k}$ . The probability for the transition to  $C_j$  (for any  $j$ ) is  $\frac{1}{2k}$ . The probability for the transition to  $B_j$  is  $\frac{1}{2k}$  if  $j < i$  and otherwise it is zero (thus the probability for the transition to  $B_k$  is always zero).

We solved this chain numerically for  $k = 10000000$  and found that  $p(A) \approx 0.29340800445$ . This shows a lower bound of 1.41318399 for 2-FF with random orders. ◀

## 3 An Improved Lower Bound for Best Fit

In this section, we turn our attention to Best Fit in the random order model. We consider a class of inputs parameterized by an integer parameter  $k \geq 1$ . Item sizes are slightly smaller than  $\frac{1}{3}$  ( $k$  such sizes) and slightly larger than  $\frac{1}{3}$  ( $k$  such sizes), and there is also a size of exactly  $\frac{1}{2}$  (so there are  $2k + 1$  different sizes). Let  $\varepsilon > 0$  be a small constant such that  $\varepsilon < \frac{1}{100k}$ . Let  $Z = \frac{1}{2}$ . For  $i = 1, 2, \dots, k$ , let

$$U_i = \frac{1}{3} - \frac{\sum_{j=i}^k \varepsilon^{k-j+1}}{2} \text{ and } V_i = \frac{1}{3} + \sum_{j=i}^k \varepsilon^{k-j+1}.$$

It holds that the sequence  $U_i$  is monotonically increasing as a function of  $i$  and the sequence  $V_i$  is monotonically decreasing as a function of  $i$ .

We say that a bin is *ready* if it cannot receive an additional item. Since the smallest item size is  $U_1$ , this means that its total size of items is above  $1 - U_1$ . Note that  $U_1 > \frac{1}{3} - k\varepsilon > \frac{1}{4}$ , and therefore no bin can contain more than three items. On the other hand, no item is larger than  $\frac{1}{2}$  since  $V_1 \leq \frac{1}{3} + k\varepsilon < \frac{1}{2}$ . A *packing pattern* is a non-empty set of three item sizes out of the  $2k + 1$  item sizes which can be packed into a bin. A multiset of items that cannot be packed into a bin is called an *invalid pattern* (and a pattern is sometimes called a valid pattern). For example, it is not possible to pack two items of size  $U_k$  and an item



of size  $V_{k-1}$  into a bin since  $2 \cdot U_k + V_{k-1} = 1 - \varepsilon + (\varepsilon + \varepsilon^2) > 1$  (so this is an invalid pattern), but it is possible to pack two items of size  $U_k$  and an item of size  $V_k$  into a bin since  $2 \cdot U_k + V_k = 1 - \varepsilon + \varepsilon = 1$  (so this is a pattern or a valid pattern).

- **Lemma 4.** *The following packing patterns are a complete list of packing patterns.*
- Any pattern consisting of a single item ( $2k + 1$  patterns in total).
  - Any pattern consisting of two items (less than  $(2k + 1)^2$  patterns in total).
  - Any pattern consisting of three items of sizes below  $\frac{1}{3}$ .
  - Any pattern consisting of three items which are  $U_{j_1}$ ,  $U_{j_2}$ , and  $V_{j_3}$  such that  $j_3 \geq \max\{j_1, j_2\}$ .

► **Corollary 5.** *Any bin with one of the following packing patterns is ready.*

*Any valid pattern consisting of three items.*

*Any pattern consisting of two items of sizes above  $\frac{1}{3}$ .*

*Any pattern with an item of size  $U_{j_1}$  and an item of size  $V_{j_2}$  such that  $j_2 < j_1$ .*

**Proof.** For patterns with three items the claim holds since there are no patterns with four items. For patterns with two items the claim holds by Lemma 4 since there is no valid pattern containing such a pair of items and an additional item. ◀

Thus, bins that are not ready are those with a single item, with two items smaller than  $\frac{1}{3}$ , and with a pair of items of sizes  $U_{j_1}, V_{j_2}$  such that  $j_2 \geq j_1$ . The following lemma establishes that although BF is not a bounded space algorithm, when we restrict our attention to this class of inputs, BF is in fact bounded space.

► **Lemma 6.** *BF never encounters a situation where it has two bins with the same pattern that are not ready. Moreover, it never has two bins with a single item each, it has at most one bin with two items smaller than  $\frac{1}{3}$ , and if it has such a bin, then there is no bin with a single item of size below  $\frac{1}{3}$ .*

**Proof.** The claim holds for bins with a single item since any two items can be packed into a bin together and BF does not open a new bin when it can pack an item into an existing bin.

Assume that there are two bins with one or two items smaller than  $\frac{1}{3}$  packed into each of these bins. When the second one of these bins was opened, its first item could have been packed into the first one of these two bins, while BF does not open a new bin when it can pack an item into an existing bin.

Finally, consider two bins, each with a pair of items of sizes  $U_{j_1}, V_{j_2}$  such that  $j_2 \geq j_1$ . When the second such bin is opened, the first such bin already has two items, since any two items can be packed together into a bin. When the item of size  $U_{j_1}$  of the second bin arrives, its bin has at most one item. Thus, the first bin has a total size  $U_{j_1} + V_{j_2}$  while the second bin has at most  $V_{j_2}$ . Since a pattern with one item of size  $V_{j_2}$  and two items of size  $U_{j_1}$  is valid, the item of size  $U_{j_1}$  should have preferred to be packed into the first bin by BF. ◀

We use the following names for bins that are not ready.

- A bin with a single item is called *OH* if it has an item of size  $Z$ , it is called *OS<sub>i</sub>* if it has an item of size  $U_i$  and it is called *OL<sub>i</sub>* if it has an item of size  $V_i$ . Recall that there is at most one such bin (out of all  $2k + 1$  types for such a bin) at every time.
- A bin with two items of sizes  $U_{j_1}$  and  $U_{j_2}$  where  $j_2 \geq j_1$  is called *TS<sub>j<sub>2</sub></sub>*. Note that we only remember the size of the larger item of the two, and recall that there is at most one such bin (out of all  $k$  types for such a bin) at every time.
- A bin with items of sizes  $U_{j_1}$  and  $V_{j_2}$  where  $j_2 \geq j_1$  is called *SL<sub>j<sub>2</sub></sub>*. Note that we only remember the size of the larger item of the two.



To summarize we recall the number of items in a bin as well as the size of its largest item.

► **Lemma 7.** *BF never encounters a situation where it has the following pairs of bins that are not ready.*

- An  $SL_{j_1}$  bin and a  $TS_{j_2}$  bin such that  $j_2 \leq j_1$ .
- An  $SL_{j_1}$  bin and an  $OS_{j_2}$  bin such that  $j_2 \leq j_1$ .
- A  $TS_{j_1}$  bin and an  $OL_{j_2}$  bin such that  $j_2 \geq j_1$ .

**Proof.** We assume by contradiction that one of the stated pairs of bins exists, and show that none of the two orders of creation (by BF) for the two bins is possible. Since every pair of items can be packed into a bin together, the first bin of the two already has two items when the second one receives its first item. This excludes the option that an  $SL_{j_1}$  bin receives its first item though an  $OS_{j_2}$  bin exists for  $j_2 \leq j_1$ , and that a  $TS_{j_1}$  bin receives its first item while an  $OL_{j_2}$  bin exists for  $j_2 \geq j_1$ . We are left with four cases where a new bin is opened when the other bin already has two items. Moreover, we use the property that no item is larger than  $\frac{1}{2}$  while every item is larger than  $\frac{1}{4}$ , so the total size of two items is always larger than that of one item.

Consider an  $SL_{j_1}$  bin and a  $TS_{j_2}$  bin such that  $j_2 \leq j_1$  that were created in this order. Neither the first item nor the second item of the  $TS_{j_2}$  could be packed into the  $SL_{j_1}$  bin (since two items are larger than one item, BF would have preferred the  $SL_{j_1}$  bin). Since the  $SL_{j_1}$  bin is not ready, it has an item of size  $V_{j_1}$ , and an item of size at most  $U_{j_1}$ . The  $TS_{j_2}$  bin has an item at most  $U_{j_2} \leq U_{j_1}$  (in fact it has two such items). This contradicts the fact that no item of the  $TS_{j_2}$  could be packed into the  $SL_{j_1}$  bin. Similarly, consider the case of an  $SL_{j_1}$  bin and an  $OS_{j_2}$  bin such that  $j_2 \leq j_1$ , that were created in this order. A contradiction is reached in the same way (the second bin has just one item of size  $U_{j_2}$  but this is sufficient for the contradiction).

Consider a  $TS_{j_2}$  bin and an  $SL_{j_1}$  bin that were created in this order. The smaller item of the  $SL_{j_1}$  bin could have been packed into the  $TS_{j_2}$  bin (no matter what the values  $j_1, j_2$  are). Consider a  $TS_{j_1}$  bin and an  $OL_{j_2}$  bin such that  $j_2 \geq j_1$  that were created in this order. The total size of two items of size at most  $U_{j_1}$  each and one item of size  $V_{j_2} \leq V_{j_1}$  can be packed together. ◀

We have seen that there can be at most one  $TS_i$  bin that is not ready (for some  $i$ ). We extend this claim for  $SL_{j_1}$  bins that are not ready (for some  $j_1$ ).

► **Lemma 8.** *BF never encounters a situation where there are two  $SL_{j_1}$  bins that are not ready.*

**Proof.** Let  $U_{j_2}, U_{j_3}$  be the sizes of items packed with items of sizes  $V_{j_1}$  into these bins. Since the bins are not ready, it holds that  $j_1 \geq j_2, j_3$ . Assume that the item of size  $U_{j_3}$  is packed into the bin that was created later by BF. Since  $U_{j_2}, U_{j_3} \leq U_{j_1}$ , the item of size  $U_{j_3}$  could be packed into the first  $SL_{j_1}$  bin, and BF would have packed it there. ◀

Note that we did not exclude the option of a pair of bins that are an  $SL_{j_1}$  bin and a  $SL_{j_2}$  bin (for  $j_1 \neq j_2$ ) that are not ready, and this is in fact possible. For example, if an item of size  $V_1$  was packed with an item of size  $U_1$  and then an item of size  $V_2$  arrives followed by an item of size  $U_2$ , BF will have one  $SL_1$  bin and one  $SL_2$  bin, none of which is ready.

To define a suitable Markov chain for the analysis of BF on our family of instances, we discuss its states and transition probabilities. Every state has a subset of bins that is not ready. The initial state has an empty set of such bins. Since there are  $k$  options for  $SL_i$  bins where each state may have at most one bin for every index, every state has a subset of

indexes of  $\{1, 2, \dots, k\}$  of such bins. We discuss the set of states as a function of the larger index  $i$  such that the state has an  $SL_i$  bin (and also states without such bins for which we let  $i = 0$ ).

For a fixed value of  $i$  and any set of values for  $j < i$  ( $2^{i-1}$  possible subsets of  $SL_j$  bins for  $j < i$ ) the following states exist (where the selected subset is called *the specific subset*), such that they are partitioned into seven types:

1. In addition to the specific subset there are no other bins that are not ready.
2. In addition to the specific subset there is an  $OH$  bin and no other bins.
3. In addition to the specific subset there is an  $OL_\ell$  bin and no other bins. Such a state exists for any  $1 \leq \ell \leq k$ .
4. In addition to the specific subset there is an  $OS_\ell$  bin and no other bins. Such a state exists for any  $i + 1 \leq \ell \leq k$ .
5. In addition to the specific subset there is a  $TS_\ell$  bin and no other bins. Such a state exists for any  $i + 1 \leq \ell \leq k$ .
6. In addition to the specific subset and a  $TS_\ell$  for a fixed  $\ell$ , there is also an  $OH$  bin. Such a state exists for any  $i + 1 \leq \ell \leq k$ .
7. In addition to the specific subset and a  $TS_\ell$  for a fixed  $\ell$ , there is also an  $OL_q$  bin. Such a state exists for any  $i + 1 \leq \ell \leq k$  and  $1 \leq q \leq \ell - 1$ .

This set of states includes all possible states based on the properties proved above. The next lemma enumerates the possible states.

► **Lemma 9.** *The set of states has exactly  $2^k \cdot (2k + 3) - 1$  states.*

Next, we explain the transitions for all states with a new item of each one of the sizes. The probability of the transition is equal to the probability of the item, as explained later. For an item of size  $Z$ , given a state which is a collection of bins that are not ready, there are two cases. If there is a bin with a single item, the new item is packed there, and the transition is to the state which has the same set of bins excluding the bin with the single item that became ready. Since there is at most one bin with a single item for each state, the transition is well-defined. If the state has no bin with a single item, the transition is to a state that contains the same collection of bins and also  $OH$ .

For an item of size  $V_i$ , the transition is as follows. If the collection of bins of the current state has a  $TS_j$  bin such that  $j \leq i$ , the transition is to a state with the same collection of bins excluding the  $TS_j$  bin. Otherwise, if the collection of bins of the state has a bin with a single item, the new item is packed there and we find whether the bin becomes ready. If it is a  $OH$  bin or a  $OL_\ell$  bin for some value of  $\ell$ , it becomes ready. If the bin with a single item is  $OS_\ell$  it becomes ready if  $\ell > i$ , and otherwise it becomes a  $SL_i$  bin. Thus the transition is to a state that contains the same collection of bins excluding the bin that was used to pack the item. If this bin did not become ready, an  $SL_i$  bin is added to the collection of the state. Note that in the case that the state has an additional bin there was no  $SL_i$  bin in the state because in the case  $\ell \leq i$  there cannot be an  $OS_\ell$  bin and an  $SL_i$  bin with  $i \geq \ell$  at the same time by Lemma 7. If there is also no bin with a single item, an  $OL_i$  bin is created, and the transition is to a state that contains the same collection of bins as the previous state, together with an  $OL_i$  bin.

For an item of size  $U_i$ , the transition is as follows. If there is an  $SL_j$  bin such that  $i \leq j$ , the item is packed there and the bin becomes ready. Since there may be multiple such bins of the form  $SL_j$  (with  $i \leq j$  and different values of  $j$ ), the one with the smallest value of  $j$  is chosen. In this case we show that this is indeed the selection of BF. For two bins that

are  $SL_{j_1}$  and  $SL_{j_2}$  bins for  $j_1 > j_2$ , it holds that the total size for the first bin is at most  $V_{j_1} + U_{j_1}$  and the total size of the second bin is at least  $V_{j_2} + U_1$ . It holds that the difference is at least

$$V_{j_2} + U_1 - V_{j_1} - U_{j_1} = \sum_{\ell=j_2}^{j_1-1} \epsilon^{k-\ell+1} - \left( \sum_{\ell=1}^{j_1-1} \epsilon^{k-\ell+1} \right) / 2 = \left( \sum_{\ell=j_2}^{j_1-1} \epsilon^{k-\ell+1} - \sum_{\ell=1}^{j_2-1} \epsilon^{k-\ell+1} \right) / 2 > 0,$$

since  $j_1 \geq j_2 + 1$  and  $j_2 \geq 1$ , and similarly to previous calculations. If there is no  $SL_j$  bin with  $i \leq j$ , but there is an  $TS_j$  bin, that bin is used and becomes ready. Otherwise, if there is a bin with a single item, the new item is packed there. The bin becomes ready if it is an  $OH$  bin. If it is an  $OS_j$  bin, it does not become ready and it turns into an  $TS_p$  bin where  $p = \max\{i, j\}$  (and there was no bin with two items of size below  $\frac{1}{3}$  before this). If it is an  $OL_j$  bin, it becomes ready if  $j < i$  and otherwise it becomes an  $SL_j$  bin (which did not exist because there was no  $SL_j$  bin for  $j \geq i$  if we reach this stage). If there was also no bin with a single item, a new  $OS_i$  bin is created. The collection of bins of the state is modified as follows. If an existing bin was used for the item and the bin became ready, it is removed from the collection. If it did not become ready, the previous bin is removed from the collection and the new one is added instead. If the item is packed into a new bin, the new bin is added to the collection.

Using the set of states and the corresponding transitions, we construct the Markov chain, and define the system of equations corresponding to the equilibrium as a system of linear equations. We solved it for different sets of probabilities for input items using online tools. We used the property that a lower bound for a randomly created input can be used as a lower bound for randomly permuted inputs.

For inputs with  $N$  items, we use probabilities  $p_0$  for an item of size  $Z$  and  $p_1, p_2, \dots, p_k$  such that an item of size  $U_i$  has probability of  $p_i$  while an item of size  $V_i$  has probability  $p_{i+k} \leq p_i/2$ . Obviously we ensure that  $p_0 + \sum_{i=1}^{2k} p_i = 1$ . An optimal solution has cost that is very close to

$$N \cdot \left( \frac{p_0}{2} + \frac{\sum_{i=1}^{2k} p_i}{3} \right) = \frac{N}{3} + \frac{p_0 \cdot N}{6}.$$

This holds since for every  $1 \leq i \leq k$ , if there are exactly  $p_i \cdot N$  items of size  $U_i$  and  $p_{i+k} \cdot N$  items of size  $V_i$ , it is possible to pack the items in triples with at most one item of size  $V_i$  in each triple (if  $p_i > 2p_{i+k}$  some triples have three items of size  $U_i$ ).

The (expected) cost of the algorithm is the sum of probabilities of transitions from a state without a bin consisting of a single item to a state with such a bin multiplied by  $N$ . The calculation requires the equilibrium probabilities of states, and transition probabilities are calculated using the probabilities for different items.

We have compared the results of our program on multiple inputs to the outputs of the program of [29] and always obtain the same result (at least eight first digits after the decimal point). However, for our largest inputs (for  $k = 9$ ) we could only run our program.

In this way we get the lower bound 1.15539228 using both programs (with a cost of approximately  $0.40092127 \cdot N$  for BF) using the probabilities

$$\begin{aligned} p_0 &= 0.0820008, \quad p_1 = 0.0848666, \quad p_2 = 0.0802666, \quad p_3 = 0.0762666, \quad p_4 = 0.0724666, \\ p_5 &= 0.0688666, \quad p_6 = 0.0650666, \quad p_7 = 0.0610666, \quad p_8 = 0.0566666, \quad p_9 = 0.0697, \\ p_{10} &= 0.0424333, \quad p_{11} = 0.0401333, \quad p_{12} = 0.0381333, \quad p_{13} = 0.0362333, \\ p_{14} &= 0.0344333, \quad p_{15} = 0.0325333, \quad p_{16} = 0.0305333, \quad p_{17} = 0.0283333, \quad \text{and } p_{18} = 0. \end{aligned}$$

## 26:12 Lower Bounds for Bin Packing Algorithms in the Random Order Model

The improved bound for  $k = 9$  is a lower bound of 1.15582656 (with a cost of approximately  $0.398760164 \cdot N$  for BF) using the probabilities (where for  $1 \leq i \leq 9$ , we let  $p_i = 2 \cdot p_{i+9}$ ):

$$\begin{aligned} p_0 &= 0.07, \quad p_{10} = 0.0411, \quad p_{11} = 0.0386, \quad p_{12} = 0.0376, \quad p_{13} = 0.0361, \quad p_{14} = 0.0346, \\ p_{15} &= 0.0329, \quad p_{16} = 0.0311, \quad p_{17} = 0.0295, \quad p_{18} = 0.0285. \end{aligned}$$

We have thus proved the next theorem.

► **Theorem 10.** *The asymptotic approximation ratio of BF on randomly permuted inputs is at least 1.15582656.*

### 4 The Absolute Approximation Ratio for Best Fit

In this section we improve the lower bound on the absolute approximation ratio of BF with random order from 1.3 to 1.5.

► **Proposition 11.** *The absolute approximation ratio of BF with random order is at least 1.5.*

**Proof.** We use a direct proof, considering all permutations of a fixed set of items. Let  $N > 0$  be a large integer. The input consists of  $N$  items of size  $\frac{1}{N}$  each (larger items) and  $N + 1$  items of size  $\frac{1}{N+1}$  each (smaller items). An optimal offline solution packs two completely full bins, such that one bin has all larger items and the other bin has all smaller items.

In order for a solution to use two bins rather than three bins, each bin has to contain items of total size exactly 1. We claim that a bin that has at least one item of each size cannot have a total size of exactly 1. Assume that a bin has  $x \geq 1$  larger items and  $y \geq 1$  smaller items, and it has a total size of 1. We have  $\frac{x}{N} + \frac{y}{N+1} = 1$ , and equivalently  $(N+1) \cdot x + N \cdot y = N(N+1)$ . Since the two variables  $x, y$  are integral, we use simple number theory arguments. The numbers  $N+1$  and  $N$  are coprime. Since  $(N+1) \cdot x = N(N+1-y)$ , we find that  $x$  is divisible by  $N$ . Since  $N \cdot y = (N+1)(N-x)$ , we conclude that  $y$  is divisible by  $N+1$ . Thus,  $x \geq N$  and  $y \geq N+1$ , and we get a total size of at least 2, which is a contradiction since these items fit into one bin.

There are  $(2N+1)!$  possible input permutations. We find those permutations that BF packs two bins for. If the first item of the permutation is larger, all larger items have to appear consecutively in the permutation such that the first bin will not contain a smaller item. If the first item of the permutation is smaller, the first bin has to receive at least  $N$  smaller items that have to appear consecutively. This property guarantees that there is no space for a larger item (and one additional smaller item may appear later). The number of permutations where all larger items appear as the first  $N$  items is  $N! \cdot (N+1)!$ , due to permuting the prefix of larger items and permuting the suffix of smaller items. The permutations where there are  $N$  smaller items at the beginning differ not only by the order but also by the smaller item that appears in the suffix of length  $N+1$ . Thus, this number is  $(N+1) \cdot N! \cdot (N+1)!$ . In total, for  $(N+1)(N+2) \cdot (N!)^2 = (N+2)! \cdot N!$  permutations, the number of bins is 2, while any other permutation results in (at least) three bins for BF.

We would like to show that the expected number of bins tends to 3 as  $N$  grows. Thus, we find an upper bound on  $\frac{(N+2)! \cdot N!}{(2N+1)!}$ :

$$\frac{(N+2)! \cdot N!}{(2N+1)!} = \frac{\prod_{i=1}^{N-1} (i+1)}{\prod_{i=1}^{N-1} (N+i+2)} \leq \frac{1}{2^{N-1}},$$

since  $\frac{i+1}{N+i+2} \leq \frac{1}{2}$  for  $i < N$ . The expected number of bins of BF for the given input is therefore at least  $3 - \frac{1}{2^{N-1}}$ . ◀

## References

- 1 S. Albers, W. Gálvez, and M. Janke. Machine covering in the random-order model. *Algorithmica*, 85(6):1560–1585, 2023. doi:10.1007/S00453-022-01011-0.
- 2 S. Albers and M. Janke. Scheduling in the random-order model. *Algorithmica*, 83(9):2803–2832, 2021. doi:10.1007/S00453-021-00841-8.
- 3 S. Albers, A. Khan, and L. Ladewig. Best Fit bin packing with random order revisited. *Algorithmica*, 83(9):2833–2858, 2021. doi:10.1007/S00453-021-00844-5.
- 4 S. Albers, A. Khan, and L. Ladewig. Improved online algorithms for knapsack and GAP in the random order model. *Algorithmica*, 83(6):1750–1785, 2021. doi:10.1007/S00453-021-00801-2.
- 5 N. Ayyadevara, R. Dabas, A. Khan, and K. V. N. Sreenivas. Near-optimal algorithms for stochastic online bin packing. In *Proc. of the 49th International Colloquium on Automata, Languages, and Programming, (ICALP'22)*, volume 229, pages 12:1–12:20, 2022. doi:10.4230/LIPICS.ICALP.2022.12.
- 6 J. Balogh, J. Békési, Gy. Dósa, L. Epstein, and A. Levin. A new and improved algorithm for online bin packing. In *Proc. of the 26th European Symposium on Algorithms (ESA2018)*, pages 5:1–5:14, 2018. doi:10.4230/LIPICS.ESA.2018.5.
- 7 J. Balogh, J. Békési, Gy. Dósa, L. Epstein, and A. Levin. A new lower bound for classic online bin packing. *Algorithmica*, 83(7):2047–2062, 2021. doi:10.1007/S00453-021-00818-7.
- 8 J. Balogh, J. Békési, Gy. Dósa, J. Sgall, and R. van Stee. The optimal absolute ratio for online bin packing. *Journal of Computer and System Sciences*, 102:1–17, 2019. doi:10.1016/J.JCSS.2018.11.005.
- 9 J. L. Bentley, D. S. Johnson, F. T. Leighton, C. C. McGeoch, and L. A. McGeoch. Some unexpected expected behavior results for bin packing. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing (STOC'84)*, pages 279–288, 1984.
- 10 B. Bosek, G. Gutowski, M. Lasoń, and J. Przybyło. First-Fit coloring of forests in random arrival model. In *Proc. of the 49th International Symposium on Mathematical Foundations of Computer Science (MFCS'24)*, pages 33:1–33:10, 2024. doi:10.4230/LIPICS.MFCS.2024.33.
- 11 J. Boyar and L. M. Favrholt. The relative worst order ratio for online algorithms. *ACM Transactions on Algorithms*, 3(2):22, 2007. doi:10.1145/1240233.1240245.
- 12 J. Boyar, L. M. Favrholt, and K. S. Larsen. Relative worst-order analysis: A survey. *ACM Computing Surveys*, 54(1):8:1–8:21, 2021. doi:10.1145/3425910.
- 13 J. Boyar, S. Irani, and K. S. Larsen. A comparison of performance measures for online algorithms. *Algorithmica*, 72(4):969–994, 2015. doi:10.1007/S00453-014-9884-6.
- 14 J. Boyar, K. S. Larsen, and A. Maiti. A comparison of performance measures via online search. *Theoretical Computer Science*, 532:2–13, 2014. doi:10.1016/J.TCS.2013.07.022.
- 15 M. G. Christ, L. M. Favrholt, and K. S. Larsen. Online bin covering: Expectations vs. guarantees. *Theoretical Computer Science*, 556:71–84, 2014. doi:10.1016/J.TCS.2014.06.029.
- 16 E. Coffman Jr., J. Csirik, L. Rónyai, and A. Zsbán. Random-order bin packing. *Discrete Applied Mathematics*, 156(14):2810–2816, 2008. doi:10.1016/J.DAM.2007.11.004.
- 17 E. G. Coffman Jr., K. So, M. Hofri, and A. C.-C. Yao. A stochastic model of bin-packing. *Information and control*, 44(2):105–115, 1980. doi:10.1016/S0019-9958(80)90050-9.
- 18 J. Csirik and B. Imreh. On the worst-case performance of the  $NkF$  bin-packing heuristic. *Acta Cybernetica*, 9(2):89–105, 1989. URL: <https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/3358>.
- 19 J. Csirik and D. S. Johnson. Bounded space on-line bin packing: Best is better than first. *Algorithmica*, 31(2):115–138, 2001. doi:10.1007/S00453-001-0041-7.
- 20 Gy. Dósa and J. Sgall. Optimal analysis of Best Fit bin packing. In *Proc. of the 41st International Colloquium on Automata, Languages, and Programming (ICALP'14) Part I*, pages 429–441, 2014.

- 21 L. Epstein, L. M. Favrholdt, and J. S. Kohrt. Comparing online algorithms for bin packing problems. *Journal of Scheduling*, 15(1):13–21, 2012. doi:10.1007/S10951-009-0129-5.
- 22 C. Fischer and H. Röglin. Probabilistic analysis of the dual next-fit algorithm for bin covering. In *Proc. of the 12th Latin American Symposium on Theoretical Informatics (LATIN'16)*, pages 469–482, 2016.
- 23 C. Fischer and H. Röglin. Probabilistic analysis of online (class-constrained) bin packing and bin covering. In *Proc. of the 13th Latin American Symposium on Theoretical Informatics (LATIN'18)*, pages 461–474, 2018.
- 24 C. O. Fischer. *New results on the probabilistic analysis of online bin packing and its variants*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2019.
- 25 F. Frei, M. Gehnen, D. Komm, R. Královic, R. Královic, P. Rossmanith, and M. Stocker. Tree coloring: Random order and predictions. *CoRR*, abs/2405.18151, 2024. doi:10.48550/arXiv.2405.18151.
- 26 M. Garg, D. Kar, and A. Khan. Random-order online independent set of intervals and hyperrectangles. In *Proc. of the 32nd Annual European Symposium on Algorithms (ESA'24)*, volume 308, pages 58:1–58:18, 2024. doi:10.4230/LIPICS.ESA.2024.58.
- 27 A. Gupta, G. Kehne, and R. Levin. Random order online set cover is as easy as offline. In *Proc. of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS'22)*, pages 1253–1264, 2021.
- 28 A. Gupta and S. Singla. Random-order models. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 234–258. Cambridge University Press, 2021. doi:10.1017/9781108637435.015.
- 29 A. Hebbbar, A. Khan, and K. V. N. Sreenivas. Bin packing under random-order: Breaking the barrier of  $3/2$ . In *Proc. of the 35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'24)*, pages 4177–4219, 2024. Also in CoRR:abs/2401.04714.
- 30 D. S. Johnson. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8:272–314, 1974. doi:10.1016/S0022-0000(74)80026-7.
- 31 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(4):299–325, 1974. doi:10.1137/0203025.
- 32 S. Kamali and A. López-Ortiz. All-around near-optimal solutions for the online bin packing problem. In *Proc. of the 26th International Symposium on Algorithms and Computation (ISAAC'15)*, pages 727–739, 2015.
- 33 A. R. Karlin, S. J. Phillips, and P. Raghavan. Markov paging. *SIAM Journal on Computing*, 30(3):906–922, 2000. doi:10.1137/S0097539794268042.
- 34 C. Kenyon. Best-Fit bin-packing with random order. In *Proc. of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'96)*, pages 359–364, 1996. Also in research report no 95-19, Ecole Normale Supérieure de Lyon, 1996.
- 35 T. Kesselheim, K. Radke, A. Tönnis, and B. Vöcking. Primal beats dual on online packing LPs in the random-order model. *SIAM Journal on Computing*, 47(5):1939–1964, 2018. doi:10.1137/15M1033708.
- 36 C. C. Lee and D. T. Lee. A simple online bin packing algorithm. *Journal of the ACM*, 32(3):562–572, 1985.
- 37 T. Leighton and P. W. Shor. Tight bounds for minimax grid matching with applications to the average case analysis of algorithms. *Combinatorica*, 9(2):161–187, 1989. doi:10.1007/BF02124678.
- 38 A. Levin. Comparing the costs of any fit algorithms for bin packing. *Operations Research Letters*, 50(6):646–649, 2022. doi:10.1016/J.ORL.2022.09.006.
- 39 M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC'11)*, pages 597–606, 2011.
- 40 W. Mao. Best- $k$ -fit bin packing. *Computing*, 50(3):265–270, 1993.

- 41 W. Mao. Tight worst-case performance bounds for Next- $k$ -Fit bin packing. *SIAM Journal on Computing*, 22(1):46–56, 1993. doi:10.1137/0222004.
- 42 J. Sgall. A new analysis of best fit bin packing. In *Proc. of the 6th International Conference on Fun with Algorithms (FUN'12)*, pages 315–321, 2012.
- 43 G. J. Woeginger. Improved space for bounded-space online bin packing. *SIAM Journal on Discrete Mathematics*, 6:575–581, 1993. doi:10.1137/0406045.
- 44 G. Zhang and M. Yue. Tight performance bound of  $AFB_k$  bin packing. *Acta Mathematicae Applicatae Sinica*, 13:443–446, 1997.