Colored Bin Packing: Online Algorithms and Lower Bounds

Martin Böhm*[1], György Dósa[2], Leah Epstein[3], Jiří Sgall*[1], and Pavel Veselý*[1]

[1]Computer Science Institute of Charles University, Prague, Czech Republic.
{bohm,sgall,vesely}@iuuk.mff.cuni.cz.
[2]Department of Mathematics, University of Pannonia, Veszprém, Hungary.
dosagy@almos.vein.hu.
[3]Department of Mathematics, University of Haifa, Haifa, Israel.
lea@math.haifa.ac.il.

Abstract

In the *Colored Bin Packing* problem a sequence of items of sizes up to 1 arrives to be packed into bins of unit capacity. Each item has one of at least two colors and an additional constraint is that we cannot pack two items of the same color next to each other in the same bin. The objective is to minimize the number of bins.

In the important special case when all items have size zero, we characterize the optimal value to be equal to color discrepancy. As our main result, we give an (asymptotically) 1.5-competitive algorithm which is optimal. In fact, the algorithm always uses at most $\lceil 1.5 \cdot OPT \rceil$ bins and we can force any deterministic online algorithm to use at least $\lceil 1.5 \cdot OPT \rceil$ bins while the offline optimum is OPT for any value of $OPT \geq 2$. In particular, the absolute competitive ratio of our algorithm is 5/3 and this is optimal.

For items of arbitrary size we give a lower bound of 2.5 on the asymptotic competitive ratio of any online algorithm and an absolutely 3.5-competitive algorithm. When the items have sizes of at most $1/d$ for a real $d \geq 2$ the asymptotic competitive ratio of our algorithm is $1.5 + d/(d-1)$. We also show that classical algorithms First Fit, Best Fit and Worst Fit are not constant competitive, which holds already for three colors and small items.

In the case of two colors—the *Black and White Bin Packing* problem—we give a lower bound of 2 on the asymptotic competitive ratio of any online algorithm when items have arbitrary size. We also prove that all Any Fit algorithms have the absolute competitive ratio 3. When the items have sizes of at most $1/d$ for a real $d \geq 2$ we show that the Worst Fit algorithm is absolutely $(1 + d/(d-1))$-competitive.

1 Introduction

In the *Black and White Bin Packing* problem proposed by Balogh et al. [3, 2] as a generalization of classical *Bin Packing*, we are given a list of items of size in $[0, 1]$, each item being either black, or white. The items are coming one by one and need to be packed into bins of unit capacity. The items in a bin are ordered by their arrival time. The additional constraint to capacity is that the colors inside the bins are alternating, i.e., no two items of the same

*This work was supported by the project 14-10003S of GA ČR and by the GAUK project 548214.

color can be next to each other in the same bin. The goal is to minimize the number of bins used.

Colored Bin Packing is a natural generalization of *Black and White Bin Packing* in which items can have more than two colors. As before, the only additional condition to unit capacity is that we cannot pack two items of the same color next to each other in one bin.

Our visualization of bins are stacks. The first item packed into a bin is at the bottom of the bin and the last item is on the top. We say that an item a is on another item b in a bin if a was packed into the bin after b and no other item was packed into the bin after b and before a; in this case a and b are next to each other.

Observe that optimal offline packings with and without reordering the items differ in this model. The packings even differ by a non-constant factor: Let the input sequence have n black items and then n white items, all of size zero. The offline optimal number of bins with reordering is 1, but an offline packing without reordering (or an online packing) needs n bins, since the first n black items must be packed into different bins. Hence we need to use the offline optimum without reordering in the analysis of online colored bin packing algorithms.

We distinguish three settings of *Colored Bin Packing*:

1. In the *offline setting* we are given the items in advance and we can pack them in an arbitrary order.
2. In the *restricted offline setting* we also know sizes and colors of all items in advance, but they are given as a sequence and they need to be packed in that order.
3. In the *online setting* the items are coming one by one and we do not know what comes next or even the total number of items. Moreover, an online algorithm has to pack each incoming item immediately and it is not allowed to change its decisions later. The online algorithm does not know the total number of colors in the input.

We focus mostly on the online setting. To measure the effectiveness of online algorithms for a particular instance L, we use the restricted offline optimum denoted by $OPT(L)$ or OPT when the instance L is obvious from the context. Let $ALG(L)$ denote the number of bins used by the algorithm ALG. The algorithm is *absolutely r-competitive* if for any instance $ALG(L) \leq r \cdot OPT(L)$ and *asymptotically r-competitive* if for any instance $ALG(L) \leq r \cdot OPT(L) + o(OPT(L))$; typically the additive term is just a constant. We say that an algorithm has the (absolute or asymptotic) competitive ratio r if it is (absolutely or asymptotically) r-competitive and it is not r'-competitive for $r' < r$.

There are several well-known and often used online algorithms for classical *Bin Packing*. We investigate the Any Fit family of algorithms (AF). These algorithms pack an incoming item into some already open bin whenever it is possible with respect to the size and color constraints. The choice of the open bin (if more are available) depends on the algorithm. AF algorithms thus open a new bin with an incoming item only when there is no other possibility. Among AF algorithms, First Fit (FF) packs an incoming item into the first bin where it fits (in the order by creation time), Best Fit (BF) chooses the bin with the highest level where the item fits and Worst Fit (WF) packs the item into the bin with the lowest level where it fits.

Next Fit (NF) is more restrictive than Any Fit algorithms, since it keeps only a single open bin and puts an incoming item into it whenever the item fits, otherwise the bin is closed and a new one is opened.

Previous results. Balogh et al. [3, 2] introduced the *Black and White Bin Packing* problem. As the main result, they gave an algorithm Pseudo with the absolute competitive ratio exactly 3 in the general case and $1 + d/(d-1)$ in the parametric case, where the items have sizes of at most $1/d$ for a real $d \geq 2$. They also proved that there is no deterministic or randomized online algorithm whose asymptotic competitiveness is below $1 + \frac{1}{2\ln 2} \approx 1.721$.

Concerning specific algorithms, they proved that Any Fit algorithms are at most absolutely 5-competitive and even optimal for zero-size items. They showed input instances on which FF and BF use asymptotically 3 times more bins than an optimal offline algorithm. For WF there are sequences of items witnessing that it is at least asymptotically 3-competitive and $(1 + d/(d-1))$-competitive in the parametric case for an integer $d \geq 2$ (for a real $d \geq 2$, it is possible to show a lower bound of $(1 + d/(d-1))$ on competitiveness of WF using a similar sequence). Furthermore, NF is not constant competitive.

The idea of the algorithm Pseudo, on which we build as well, is that it first packs the items regardless of their size, i.e., treating their size as zero. This can be done optimally for two colors, e.g., by First Fit. These bins are partitioned by NF into bins of level at most 1. The algorithm can be performed online.

In the offline setting, Balogh et al. [3] gave a 2.5-approximation algorithm with $\mathcal{O}(n \log n)$ time complexity and an asymptotic polynomial time approximation scheme, both when reordering is allowed.

Our results. We completely solve the case of *Colored Bin Packing* for zero-size items. As we have seen, this case is important for constructing general algorithms. The offline optimum (without reordering) is actually not only lower bounded by the color discrepancy, but equal to it for zero-size items (see Section 2). For online algorithms, we give an (asymptotically) 1.5-competitive algorithm which is optimal (see Section 4.1). In fact, the algorithm always uses at most $\lceil 1.5 \cdot OPT \rceil$ bins and we can force any deterministic online algorithm to use at least $\lceil 1.5 \cdot OPT \rceil$ bins using only three colors while the offline optimum is OPT for any value of $OPT \geq 2$ (see Section 3.1). In particular this shows that the absolute competitive ratio of our algorithm is 5/3, which is optimal.

For items of arbitrary size and three colors, we show a lower bound of 2.5 on the asymptotic competitive ratio of any deterministic online algorithm (see Section 3.2). We use the optimal algorithm for zero-size items and the algorithm Pseudo to design an (absolutely) 3.5-competitive algorithm which is also (asymptotically) $(1.5 + d/(d-1))$-competitive in the parametric case, where the items have sizes of at most $1/d$ for a real $d \geq 2$ (see Section 4.2). (Note that for $d < 2$ we have $d/(d-1) > 2$ and the bound for arbitrary items is better.)

We show that algorithms BF, FF, WF and Pseudo (with FF for packing zero-size items) are not constant competitive, in contrast to their absolute 3-competitiveness for two colors. Their competitiveness cannot be bounded by any function of the number of colors even for only three colors and very small items (see Section 4.3).

For *Black and White Bin Packing*, we propose a lower bound of 2 on the asymptotic competitive ratio of any online algorithm improving the previous lower bound of 1.721 (see Section 3.2). We show that the absolute competitive ratio of Any Fit algorithms in the general case is at most 3 which is tight for BF, FF and WF (see Section 5.1). For WF in the parametric case, we prove that it is absolutely $(1 + d/(d-1))$-competitive for a real $d \geq 2$ which is tight (see Section 5.2). Therefore, WF has the same competitive ratio as the algorithm Pseudo.

Related work. In the classical *Bin Packing* problem, we are given items with sizes in $(0, 1]$ and the goal is to assign them into the minimum number of unit capacity bins. The problem was proposed by Ullman [25] and by Johnson [19] and it is known to be NP-hard.

See the survey of Coffman et al. [8] for the many results on classical *Bin Packing* and its many variants.

For the online problem, there is no online algorithm which is better than $248/161 \approx 1.5403$-competitive [4]. The currently best algorithm is HARMONIC++ by Seiden [24], approximately asymptotically 1.589-competitive. The best possible absolute competitive ratio of 5/3 was recently achieved by an algorithm FIVE-THIRDS [5]. Regarding AF algorithms, NF is absolutely 2-competitive and both FF and BF have the absolute competitive ratio exactly 1.7 [12, 13]. This is similar to *Black and White Bin Packing* in which FF and BF have the absolute competitive ratio of 3 and the hard instances proving tightness of the bound are the same for both algorithms.

In the context of *Colored Bin Packing*, we are interested in variants that further restrict the allowed packings. Of particular interest is *Bounded Space Bin Packing* where an algorithm can have only $K \geq 1$ open bins in which it is allowed to put incoming items. When a bin is closed an algorithm cannot pack any further item in the bin or open it again. Such algorithms are called *K-bounded-space*. The champion among these algorithms is K-BOUNDED BEST FIT, i.e., BEST FIT with at most K open bins, which is (asymptotically) 1.7-competitive for all $K \geq 2$ [9]. Lee and Lee [21] presented HARMONIC(K) which is K-bounded-space with the asymptotic competitive ratio of approximately 1.691 for sufficiently large K. Lee and Lee also proved that there is no bounded space algorithm with a smaller asymptotic competitive ratio.

The *Bounded Space Bin Packing* is an especially interesting variant in our context due to the fact that it matters whether we allow the optimum to reorder the input instance or not. If we allow reordering for *Bounded Space Bin Packing*, we get the same optimum as classical *Bin Packing*. In fact, all the bounds on online algorithms in the previous paragraph hold if the optimum with reordering is considered, which is a stronger statement than comparing to the optimum without reordering. This is a very different situation than for *Colored Bin Packing*, where no online algorithms can be competitive against the optimum with reordering, as we have noted above.

The bounded space offline optimum without reordering was studied by Chrobak et al. [7]. It turns out that the computational complexity is very different: There exists an offline $(1.5+\varepsilon)$-approximation algorithm for *2-Bounded Space Bin Packing* with polynomial running time for every constant $\varepsilon > 0$, but exponential in ε. No polynomial time 2-bounded-space algorithm can have its approximation ratio better than 5/4 (unless $P = NP$). In the online setting it is open whether there exists a better algorithm than asymptotically 1.7-competitive K-BOUNDED BEST FIT when compared to the optimum without reordering; the current lower bound is 3/2.

Other variants of bin packing where the sequence of items must remain ordered even for offline solutions include *Bin Packing with LIB* (largest item in the bottom) constraints, where an item can be packed into a bin with sufficient space if it is no larger than any item packed into this bin [22, 17, 23, 16, 14].

Another interesting variant with restrictions on the contents of a bin is *Bin Packing with Cardinality Constraints*, which restricts the number of items in a bin to at most k for a parameter $k \geq 2$. It was introduced by Krause et al. [20] who also showed that CARDINALITY CONSTRAINED FF has the asymptotic competitive ratio of at most $2.7 - 2.4/k$. Interestingly, the lower bound for the asymptotic competitive ratio of any online algorithm for large k is 1.5403 [4], i.e., the same as for standard *Bin Packing*. For $k \geq 3$, there is an asymptotically 2-competitive online algorithm [1] and the absolute competitive ratio of any online algorithm is at least 2 for any $k \geq 4$ [11]. Better algorithms and various lower bounds are known for small k [15, 1, 18].

Motivation. Suppose that a television or a radio station maintains several channels and wants to assign a set of programs to them. The programs have types like "documentary", "thriller", "sport" on TV, or music genres on radio. To have a fancy schedule of programs, the station does not want to broadcast two programs of the same type one after the other. *Colored Bin Packing* can be used to create such a schedule. Items here correspond to programs, colors to genres and bins to channels. Moreover, the programs can appear online and have to be scheduled immediately, e.g., when listeners send requests for music to a radio station via the Internet.

Another application of *Colored Bin Packing* comes from software which renders user-generated content (for example from the Internet) and assigns it to columns which are to be displayed. The content is in boxes of different colors and we do not want two boxes of the same color to be adjacent in a column, otherwise they would not be distinguishable for the user.

Moreover, *Colored Bin Packing* with all items of size zero corresponds to a situation in which we are not interested in loads of bins (lengths of the schedule, sizes of columns, etc.), but we just want some kind of diversity or colorfulness.

2 Preliminaries and Offline Optimum

Notation. Let C be the set of all colors in the input sequence. For $c \in C$, the items of color c are called c-items and bins with the top (last) item of color c are called c-bins. By a non-c-item we mean an item of color $c' \neq c$ and similarly a non-c-bin is a bin with the top item of color $c' \neq c$. The *level of a bin* means the cumulative size of all items in the bin.

We denote a sequence of nk items consisting of n groups of k items of colors $c_1, c_2, \ldots c_k$ and sizes $s_1, s_2, \ldots s_k$ by $n \times \left(\begin{smallmatrix} c_1 \\ s_1 \end{smallmatrix}, \begin{smallmatrix} c_2 \\ s_2 \end{smallmatrix}, \ldots \begin{smallmatrix} c_k \\ s_k \end{smallmatrix} \right)$.

Lower Bounds on the Restricted Offline Optimum. We use two lower bounds on the number of bins in any packing. The first bound *Vol* is the sum of sizes of all items (the total volume).

The second bound *Dis* is the maximal color discrepancy inside the input sequence. In *Black and White Bin Packing*, the color discrepancy introduced by Balogh et al. [2] is simply the difference of the number of black and white items in a segment of the input sequence, maximized over all segments. It is easy to see that it is a lower bound on the number of bins.

In the generalization of the color discrepancy for more than two colors we count the difference between c-items and non-c-items for all colors c and segments. It is easy to see that this is a lower bound as well. Formally, let $s_{c,i} = 1$ if the i-th item from the input sequence has color c, and $s_{c,i} = -1$ otherwise. We define

$$Dis = \max_{c \in C} \max_{i,j} \sum_{\ell=i}^{j} s_{c,\ell} \,.$$

For *Black and White Bin Packing*, equivalently $Dis = \max_{i,j} \left| \sum_{\ell=i}^{j} s_\ell \right|$, where $s_i = 1$ if the i-th item is white, and $s_i = -1$ otherwise; the absolute value replaces the maximization over colors.

We prove that *Dis* is a lower bound on the optimum similarly to the proof of Lemma 5 in [2]. First we observe that the number of bins in the optimum cannot increase by removing a prefix or a suffix from the sequence of items.

Observation 2.1. *Let $L = L_1L_2L_3$ be a sequence of items partitioned into three subsequences (some of them can be empty). Then $OPT(L) \geq OPT(L_2)$.*

Proof. It is enough to show that the removal of the first or the last item does not increase the optimum. By iteratively removing items from the beginning and the end of the sequence we obtain the subsequence L_2 and consequently $OPT(L) \geq OPT(L_2)$.

The first item of the sequence is clearly the first item in a bin. By removing the first item from the bin we do not violate any condition. Hence any packing of L is a valid packing of L without the first item. A similar claim holds for the last item. □

Lemma 2.2. *$OPT(L) \geq Dis$.*

Proof. We prove that for every color c, the optimum is at least $Dis_c := \max_{i,j} \sum_{\ell=i}^{j} s_{c,\ell}$. Fix a color c and let i, j be $\arg\max_{i,j} \sum_{\ell=i}^{j} s_{c,\ell}$. Let $\delta = Dis_c$. We may assume that $\delta > 0$, otherwise δ is trivially at most the optimum. By the previous observation we may assume $i = 1$ and $j = n$.

Consider any packing of the sequence and let k be the number of bins used. Any bin contains at most one more c-item than non-c-items, since colors are alternating between c and other colors in the extreme case. Since we have δ more c-items than non-c-items, we get $k \geq \delta$. Therefore $OPT \geq Dis_c$. □

In *Black and White Bin Packing*, when all the items are of size zero, all Any Fit algorithms create a packing into the optimal number of bins [2]. For more than two colors this is not true and in fact no deterministic online algorithm can have a competitive ratio below 1.5. However, in the restricted offline setting a packing into *Dis* bins is still always possible, even though this fact is not obvious. This shows that the color discrepancy fully characterizes the combinatorial aspect of the color restriction in *Colored Bin Packing*.

Theorem 2.3. *Let all items have size equal to zero. Then a packing into Dis bins is possible in the restricted offline setting, i.e., items can be packed into Dis bins without reordering.*

Proof. Consider a counterexample with a minimal number of items in the sequence. Let $D = Dis$ be the maximal discrepancy in the counterexample and $n \geq D$ be the number of items. The minimality implies that the theorem holds for all sequences of length $n' < n$. Moreover, $D > 1$, since for $D = 1$ we can pack the sequence trivially into a single bin as there are no two adjacent items with the same color in the sequence.

We define an *important interval* as a maximal interval of discrepancy D. More precisely, consider a subsequence I of the input of consecutive items starting from the i-th item and ending with the j-th item, such that for some color c the discrepancy for this interval is D, i.e., $\sum_{\ell=i}^{j} s_{c,\ell} = D$. Assume that there is no longer subsequence I' of consecutive input items that has discrepancy D and that contains the subsequence I (that is, a subsequence I' which starts no later than the i-th item and ends no earlier than the j-th item). A subsequence I of this kind is called an important interval. For an important interval, its *dominant color c* is the most frequent color in it. We start with showing that important intervals are just D items of the same color in the minimal counterexample.

Observation 2.4. *Each important interval I contains D items of its dominant color c and no other items in the minimal counterexample.*

Proof. Suppose there is a non-c-item in I and let a be the last such item in I. Then a must be followed by a c-item b in I, otherwise I without a would have discrepancy higher than D. We temporarily delete a and b from the sequence and pack the rest into D bins by minimality.

We interrupt the packing of the rest of the sequence just after the item prior to a is put into a bin. There must be a c-bin B, otherwise the subsequence of I from the beginning up to a (including a) would have strictly more non-c-items than c-items (for each c-item in the subsequence of I there is a non-c-item packed on it and a is the extra non-c-item), and hence the rest of I after a would have discrepancy of more than D. We put a and b into B and the bin B is still a c-bin. Then we continue in the packing of the rest of the sequence which yields a packing of the whole sequence into D bins, therefore it is not a counterexample. □

Next, we show that important intervals are disjoint in the minimal counterexample. Suppose that two important intervals I_1 and I_2 with dominant colors c_1 and c_2 intersect on an interval J. If $c_1 \neq c_2$ we use the previous observation, since I_1 or I_2 has to contain an item from the other interval. Otherwise $c_1 = c_2$, but then their union has discrepancy of more than D which cannot happen.

Clearly, there must be an important interval in any non-empty sequence. Let $I_1, I_2, \dots I_k$ be important intervals in the counterexample sequence and let $J_1, J_2, \dots J_{k-1}$ be the intervals between the important intervals (J_i between I_i and I_{i+1}), J_0 be the interval before I_1 and J_k be the interval after I_k. These intervals are disjoint and form a complete partition of the sequence, i.e., $J_0, I_1, J_1, I_2, J_2, \dots\ J_{k-1}, I_k, J_k$ is the whole sequence of items. Note that some of J_ℓ's can be empty.

If $k > 2$, we can create a packing P_1 of the sequence containing only intervals J_0, I_1, J_1, I_2 into D bins by minimality of the counterexample. Also there exists a packing P_2 of intervals $I_2, J_2, I_3, \dots I_k, J_k$ into D bins. Any bin from P_1 must end with an item from the important interval I_2 and any bin from P_2 must start with an item from I_2. Therefore we can merge both packings by items from I_2 and obtain a valid packing of the whole sequence into D bins. Hence $k \leq 2$.

In the case $k = 1$, there are four subcases depending on whether J_0 and J_1 are empty or not:

- J_0 and J_1 are non-empty: We create packings of J_0, I_1 and I_1, J_1 into D bins and merge them as before.
- J_0 is empty and J_1 non-empty: We delete the first item from I_1, pack the rest into $D-1$ bins (the maximal discrepancy decreases after deleting) and put the deleted item into a separate bin.
- J_0 is non-empty and J_1 empty: Similarly, we delete the last item from I_1 and pack the rest into $D-1$ bins.
- both are empty: I_1 can be trivially packed into D bins.

For $k = 2$, we first show that J_0 and J_2 are empty and J_1 is non-empty in the counterexample. If J_0 is non-empty, we merge packings of J_0, I_1 and I_1, J_1, I_2, J_2, and if J_2 is non-empty, we put together packings of J_0, I_1, J_1, I_2 and I_2, J_2. When J_1 is empty, the sequence consists only of intervals I_1 and I_2 which must have different dominant colors. Thus they can be easily packed one on the other into D bins.

The last case to be settled has only I_1, J_1 and I_2 non-empty. If the dominant colors c_1 for I_1 and c_2 for I_2 are different, we delete the first item from I_1 and the last item from I_2, so the discrepancy decreases. We pack the rest into $D-1$ bins and put the deleted items into a separate bin, so the whole sequence is in D bins again.

Otherwise c_1 is equal to c_2 and let c be c_1. Since $I_1 \cup J_1 \cup I_2$ is not important, there must be at least $D+1$ more non-c-items than c-items in J_1. Also any prefix of J_1 contains strictly more non-c-items than c-items, thus at least the first two items in J_1 have colors different from c.

We now delete three items from the sequence, pack those items into one bin and show that the rest of the sequence can be packed into $D-1$ bins. Let I_1' be I_1 without the first c-item p in I_1, let J_1' be J_1 without the first non-c-item q in J_1 and let I_2' be I_2 without the last c-item r in I_2. In other words, we delete items p, q, r from the sequence. See Figure 1 for an illustration of the original and modified sequence.

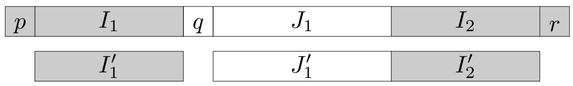

Figure 1: An illustration of the original sequence and the modified sequence $I_1' \cup J_1' \cup I_2'$ in the last case of the proof of Theorem 2.3.

Lemma 2.5. *The discrepancy of* $I_1' \cup J_1' \cup I_2'$ *is* $D-1$.

Proof. Note that the discrepancy of I_1' is $D-1$, thus it remains to show the upper bound. Let I' be an important interval in $I_1' \cup J_1' \cup I_2'$. Suppose for a contradiction that the discrepancy of I' is D and let a be the first item in I' and b be the last item in I'. Let I be the interval from a to b in $I_1 \cup J_1 \cup I_2$, i.e., the original sequence.

I must contain at least one of deleted items p, q, r, since otherwise $I = I'$ and we have found another important interval in the original sequence, because both I_1 and I_2 contain a deleted item. The deleted item in I cannot be p or r, since they are the first and the last item, respectively, of the original sequence.

Thus I contains q, b is after q and a is before q in the sequence. It follows that the dominant color of I' is c, otherwise deleting a would increase the discrepancy of I'. Note that I does not contain the deleted item p and that the discrepancy of I is at least $D-1$, since I and I' differ only by q.

Consider the interval J from p, the first item in the sequence, to b. It contains I and additionally at least one c-item p. Hence J has the discrepancy at least D, but it contains the whole I_1 and some other items not in I_1 (at least q and b) which is a contradiction, because I_1 is an important interval and cannot be extended without decreasing its discrepancy. □

By the previous lemma and the minimality of the counterexample, we pack the items in $I_1' \cup J_1' \cup I_2'$ into $D-1$ bins and we put the deleted items p, q and r into another bin. Thus in all cases we can pack the sequence into D bins, therefore no such counterexample exists. □

By Theorem 2.3 we can compute the restricted offline optimum of an instance with zero-size items only in polynomial time by computing Dis.

3 Lower Bounds on Competitiveness of Any Online Algorithm

In our proofs of lower bounds for any deterministic online algorithm, an input is presented to an arbitrary fixed online algorithm. The next item in the input depends on what has the algorithm done with previous items. A natural way to describe such inputs uses a malicious *adversary* that chooses the next item in the input based on the current packing of the algorithm. The adversary tries to maximize the number of bins used by the algorithm, while keeping the offline optimum relatively small.

3.1 Lower Bound for Zero-size Items

Theorem 3.1. *For zero-size items of at least three colors, there is no deterministic online algorithm with an asymptotic competitive ratio less than* 1.5. *Precisely, for each* $n > 1$ *we can force any deterministic online algorithm to use at least* $\lceil 1.5n \rceil$ *bins using three colors, while the optimal number of bins is* n.

Proof. The adversary uses only three colors throughout the proof, denoted by black, white and red and abbreviated by b, w and r in formulas. We show that if an online algorithm uses less than $\lceil 1.5n \rceil$ bins, the adversary can send some items and force the algorithm to increase the number of black bins or to use at least $\lceil 1.5n \rceil$ bins, while the optimal number of bins stays n. This way the algorithm is forced to open $\lceil 1.5n \rceil$ bins using finitely many items as the number of black bins is increasing.

Initially the adversary sends n black items, then it continues by phases and ends the process whenever the algorithm uses $\lceil 1.5n \rceil$ bins at the end of a phase. When a phase starts, the algorithm has $N_{\mathrm{b}} < \lceil 1.5n \rceil$ black bins and possibly some other white or red bins. In each phase the adversary forces the algorithm to use $\lceil 1.5n \rceil$ bins or to have more than N_{b} black bins. Note that the number of black bins increases in all phases except possibly the last one.

The adversary also guarantees that there is a restricted offline packing of the items into n bins at the beginning of each phase and moreover all bins in the offline packing of the adversary are black after each phase in which N_{b} increases.

We now present how a phase works. Let *new items* be items from the current phase and *old items* be items from previous phases. The adversary begins the phase by sending n new items of colors alternating between white and red, starting by white, so it sends $\lceil n/2 \rceil$ white items and $\lfloor n/2 \rfloor$ red items.

If the algorithm has not put some new item on an old black item, the adversary sends n black items. Since the new items are packed into less than n black bins (more precisely, black at the beginning of the phase), the number of black bins increases. See Figure 2 for an example of such situation. In the offline packing the adversary packs the first n new items of colors alternating between white and red into n bins (one item into each bin) which is allowed since all the bins were black at the beginning of the phase. Then the adversary puts n new black items into n bins and all the bins are black as desired. The adversary finishes the phase and continues with the next phase if the algorithm has less than $\lceil 1.5n \rceil$ black bins.

Otherwise the algorithm put all new red and white items on old black items. If n is even, the adversary sends n additional white items. After that the algorithm has at least $1.5n$ white bins. The adversary packs the first n new items into a single bin which now has a red item on the top. Since all other bins are black, the next n new white items are packed into n bins. Therefore the adversary reaches its goal and stops the process.

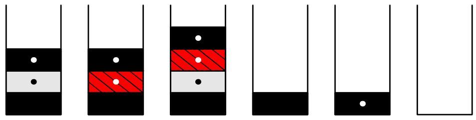

Figure 2: An illustration of the case when the algorithm has not put some new item on an old black item for $n = 4$. New items are depicted with a dot. Note that an algorithm packed the first n new items into less than n black bins, thus the adversary sent n black items and forced an increase of the number of black bins.

If n is odd, the adversary has a white bin in the offline packing, thus it can send only $n - 1$ white items forcing the algorithm to have $\lceil 1.5n \rceil - 1$ white bins. This suffices to prove the result in the asymptotic sense, but for the precise lower bound of $\lceil 1.5n \rceil$ for an odd n we need a somewhat more complicated construction.

Therefore if all new red and white items are put on old black items and n is odd, the adversary sends a black item e. We split our analysis depending on where e is packed by the algorithm:

1. If the algorithm does not pack e on a new white item, the adversary sends n white items forcing $\lceil n/2 \rceil + n$ white bins. The offline packing is created similarly to the case of even n: Put the first n new items and e into one bin and the next n new white items into n bins. Thus the adversary is done and stops the process.
2. The black item e is put on a new white item. There are $\lfloor n/2 \rfloor$ white and $\lfloor n/2 \rfloor$ red new items on the top of algorithm's bins and the adversary sends another black item f. Since red and white are equivalent colors (considering only new items), without loss of generality the algorithm packs f into a red bin or into newly opened bin.
   Next the adversary sends a white item g and a red item h. After packing g there are $\lceil n/2 \rceil$ bins with a new white item on the top and at least one bin with a new black item on the top. If h is not put on a new white item (i.e., it is put into a black bin, a new bin or on an old white item), the adversary sends n white items and the algorithm must use $\lceil 1.5n \rceil$ bins. In this case the adversary packs the first $n-1$ new items together with e, g and h into one bin and the n-th new white item with f into another bin. Then all the bins are black and the last n new white items are put into them. The adversary stops sending items again.
   Otherwise the algorithm packs h on a new white item and the adversary sends n black items. (See Figure 3.) The number of black bins increases, because the adversary sent $n + 2$ new black items and $n + 2$ new non-black items, and the red item h is put on a new white item. In the offline packing the first n new non-black items are packed into n bins, black items e and f into two arbitrary bins and non-black items g and h are put on e and f. Since no bin is black, the adversary puts the last n new black items into n bins and all the bins are black, thus the adversary continues with the next phase.

□

The lower bound has additional properties that we use later in our lower bound for items of arbitrary size. Most importantly, we have at least $\lceil 1.5 \cdot OPT \rceil$ of c-bins at the end (and possibly some additional bins of other colors).

Lemma 3.2. *After packing the instance from Theorem 3.1 by an online algorithm there is a color c for which we have $\lceil 1.5 \cdot OPT \rceil$ of c-bins. Moreover, in each restricted offline optimal*

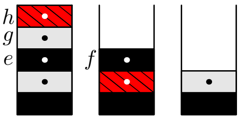

Figure 3: An illustration of the packing for $n = 3$ in the last case of the proof of Theorem 3.1, i.e., the algorithm put the black item e on a new white item, the black item f is packed into a red bin, and the red item h is put on a new white item (note that it does not matter where the white item g is packed). New items are depicted with a dot. Then the adversary sends 3 black items and the algorithm must pack one of them into a newly created bin.

packing of the instance all the bins have a c-item on the top.

Proof. Let $n = OPT$ as in the previous proof. The adversary stops sending items when it finishes the last phase. In the last phase either the number of black bins increases to $\lceil 1.5n \rceil$, or the adversary forces $\lceil 1.5n \rceil$ white or red bins by sending n white or red items. In both cases we have $\lceil 1.5n \rceil$ bins of the same color.

Since an optimal packing uses n bins and the last n items are of the same color (in each case of the construction), they must be packed into different bins. Hence each bin of a restricted offline optimal packing has a c-item on the top. □

3.2 Lower Bound for Items of Arbitrary Size

We show a lower bound of 2 for two colors, i.e., for *Black and White Bin Packing*, and a lower bound of 2.5 for at least three colors. Both bounds follow a similar adversarial construction that has two parts: The first part uses only zero-size items to create a lot of bins of the same color, say white. The second part is the same for both lower bounds and it is defined in the next lemma.

As in the lower bound for zero-size items, the lower bound for at least three colors uses exactly three colors, denoted by black, white and red and abbreviated by b, w and r in formulas.

The next lemma constitutes the second part of the adversarial construction.

Lemma 3.3. *Suppose that a deterministic online algorithm A has created $k \geq n$ bins of the same color, without loss of generality white, on a sequence L of zero-size items (A may create some bins of other colors which we do not take into account). Suppose also that:*

- *$OPT(L) = n > 1$,*
- *in each restricted offline optimal packing of L all the bins have a white item on the top.*

Then for the online algorithm A there exists a sequence L' of black and white items such that A uses $k + n$ bins on the whole sequence LL', while an optimal restricted offline algorithm packs LL' into $n + 1$ bins.

Note that the preconditions of the lemma are exactly satisfied by Lemma 3.2 for $k = \lceil 1.5n \rceil$.

Proof. Let W be the set of k white bins opened by A on the sequence L.

The proof is based on the following idea: The adversary sends the instance in phases, each starting with two small items, white and black. If the black item is put into an already opened bin with a non-zero level, we send a huge white item that can be put only on the small black item. Therefore the algorithm has to put the huge white item into a bin with level zero (and not from the set W), but an optimal offline algorithm puts the small black item into a level zero bin and the huge white item on it. Otherwise, if the small black item is put into a new bin or a level zero bin, the phase is finished: The online algorithm opened a bin in the phase, while an optimal offline algorithm does not need to. This way an online algorithm is forced to behave oppositely to an optimal offline algorithm. Note that the first option (the first black item from the phase is put into an already opened bin) is better for the online algorithm.

We formalize this idea by the following adversarial algorithm. Let $\varepsilon = 1/(4k)$ and $\delta_i = 1/5^i \cdot \varepsilon = 1/(5^i \cdot 4k)$. The adversary uses the items of the following types:

- regular white items of size ε,
- regular black items of size δ_i for some $i \geq 1$,
- special black items of size $3\delta_i$ for some $i \geq 1$,
- huge white items of size $1 - 2\delta_i$ for some $i \geq 1$.

Note that $3\delta_i < \varepsilon$, i.e., all black items are smaller than ε, and that a huge white item of size $1 - 2\delta_i$ cannot be packed with a black item of size at least δ_j for any $j < i$.

Let i be the index of the current phase and let j be the number of huge white items in the instance so far. The adversarial algorithm is as follows:

1. Let $i = 0$ and $j = 0$.
2. If $j = n$ or if $i = k + n$, then stop.
3. Let $i = i + 1$. Send $\binom{\text{white}}{\varepsilon}, \binom{\text{black}}{\delta_i}$, i.e., a group consisting of a regular white item and a regular black item.
4. If the regular black item is packed in a new bin or in a bin with level zero, go to step 2 (continue with the next phase).
5. Let $j = j + 1$. Send $\binom{\text{black}}{3\delta_i}, \binom{\text{white}}{1-2\delta_i}, \binom{\text{black}}{\delta_i}$. Then go to step 2 (continue with the next phase).

See Figure 4 for an example of the situation after two phases of the adversarial algorithm.

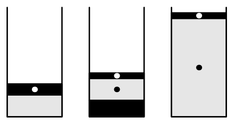

Figure 4: A situation after two phases of the algorithm (for simplicity, zero-size items are not shown). Items from the second phase are marked with a dot. In the first phase, the second item went into a bin of level zero, so the phase ended immediately. However, in the second phase the second item went into a bin with a non-zero level, thus a huge white item arrived.

First we show that we can pack the whole list of items into $n + 1$ bins and then that no huge white item can be packed by an online algorithm into a bin from the set W, i.e., one of k bins which are white after the first part with zero-size items.

Lemma 3.4. $OPT(LL') = n + 1$.

Proof. We create n white bins of level zero from the list L by the preconditions of the lemma. Each of $j \leq n$ huge white items is packed with the two regular black items from the same phase, thus creating j full bins with a black item at the bottom. All these bins are combined with the bins created from L. The remaining items, i.e., for each phase the regular white item and the special or regular black item, have alternating colors and the total size of at most $2i \cdot \varepsilon \leq 2(k+n) \cdot /(4k) \leq 1$ (recall that $k \geq n$, i is the index of the current phase and all black items are smaller than ε). Therefore all remaining items can be put into an additional $(n+1)$-th bin.

Since all bins in each optimal packing of L are white and L' begins by a regular white item, we get that $OPT(LL') = n + 1$. □

We now analyze how the online algorithm A behaves on the sequence L'.

Lemma 3.5. *After the i-th phase the number of bins with a non-zero level is at least i. Moreover, A packs no huge white item into a bin from the set W.*

Proof. We show that in each phase the number of bins with a non-zero level increases by at least one. This holds trivially, if the second item in a phase, denoted by s, is put into a new bin or to a bin with level zero. Otherwise, if s is put into a bin of non-zero level, the adversary continues the phase by sending three other items, most importantly a huge white item h. The item s is the only one from L' that is sent before h and that is sufficiently small to be packed into a single bin with h, but s is in a bin with another item of non-zero size. Therefore h must be packed into a new bin or into a bin with level zero. This proves the first statement of the lemma.

For the second statement, note that if the algorithm puts h into a bin with zero-size items only, the bin cannot be white, but all the bins from the set W that have still level zero (while packing h) are white. As we already observed, h is not put into a bin from W that has a non-zero level. □

We now finish the proof of Lemma 3.3. By the previous lemma we know that if the adversarial algorithm ends with $i = k+n$, there are $k+n$ bins with a non-zero level. Otherwise, if the instance stops by $j = n$, the online algorithm has at least $|W| + n = k + n$ open bins, since it opens bins in W on L and it must put n huge white items into other bins. □

We use the lemma to prove lower bounds for *Black and White Bin Packing* and for *Colored Bin Packing*.

Theorem 3.6. *For items of two colors and arbitrary size, there is no deterministic online algorithm with an asymptotic competitive ratio of less than* 2.

Proof. Let $n > 1$ be a large integer. The adversary starts the instance by sending n zero-size white items and the online algorithm must open n white bins, one for each item.

Observe that the preconditions of Lemma 3.3 are satisfied for $k = n$ and L being the n zero-size white items. By the lemma the adversary forces the algorithm to use $k + n = 2n$ bins, while the restricted offline optimum equals $n+1$. Thus we get that the ratio between the number of bins by the online algorithm and the optimum tends to 2 as n goes to infinity. □

By combining our lower bound of 1.5 for zero-size items of at least three colors and Lemma 3.3 we obtain a general lower bound of 2.5 for items of arbitrary size and at least three colors.

Theorem 3.7. *For items of at least three colors and arbitrary size, there is no deterministic online algorithm with an asymptotic competitive ratio of less than* 2.5.

Proof. Let $n > 1$ be a large integer. The adversary starts with the hard instance for zero-size items from the proof of Theorem 3.1 with the optimum equal to n. By Lemma 3.2 there are at least $\lceil 1.5n \rceil$ bins of the same color, without loss of generality white, and each optimal packing has all bins of the same color. This satisfies the preconditions of Lemma 3.3 with $k = \lceil 1.5n \rceil$ and L being the sequence of items from the lower bound for zero-size items.

We now use Lemma 3.3 again and get that the algorithm must use at least $k+n = \lceil 2.5n \rceil$ bins. As $OPT = n + 1$, we get that the ratio between the number of bins by the online algorithm and the optimum tends to 2.5 as n goes to infinity. □

4 Algorithms for Arbitrarily Many Colors

4.1 Optimal Algorithm for Zero-size Items

The main problem of FF, BF and WF is that they pack items regardless of the colors of bins, only keeping the packing valid. We address the problem by balancing the colors of top items in bins – we mostly put an incoming c-item into a bin of the most frequent other color. When there are more most frequent colors other than c or we have more choices of bins of the most frequent other color where to put an item we can choose arbitrarily among these colors or bins, e.g., by First Fit. We call this algorithm Balancing Any Fit (BAF).

We define BAF for items of size zero and show that it opens at most $\lceil 1.5Dis \rceil$ bins which is optimal in the worst case by Theorem 3.1. Then we combine BAF with the algorithm Pseudo by Balogh et al. [2] for items of arbitrary size and prove that the resulting algorithm is absolutely 3.5-competitive.

Let D_k be the maximal discrepancy on the subsequence of the input from the first item up to the k-th item, i.e., $D_k = \max_{c \in C} \max_{i,j \leq k} \sum_{\ell=i}^{j} s_{c,\ell}$, and let $N_{c,k}$ be the number of c-bins after packing the k-th item. We define the current discrepancy as $CD_{c,k} = \max_{i \leq k+1} \sum_{\ell=i}^{k} s_{c,\ell}$, i.e., the discrepancy of color c on an interval which ends with the last packed item (the k-th). The current discrepancy basically tells us how many c-items have come recently and thus how many c-items may arrive without increasing the overall discrepancy. Note that $CD_{c,k} \leq D_k$ and that $CD_{c,k}$ is at least zero as we can set $i = k + 1$.

Let $\alpha_{c,k} = N_{c,k} - \lceil D_k/2 \rceil$ be the difference between the number of c-bins and the half of the maximal discrepancy so far. Observe that $\lceil D_k/2 \rceil$ is the number of bins which BAF may use in addition to OPT bins, since the current value of OPT is D_k by Theorem 2.3. We omit the index k in D_k, $N_{c,k}$, $CD_{c,k}$ and $\alpha_{c,k}$ when it is obvious from the context.

While processing the items, if D is the maximal discrepancy so far, the algorithm may receive $D - CD_c$ of c-items without changing the maximal discrepancy; this forces the algorithm to use $N_c + D - CD_c$ bins. Hence, to terminate with at most $\lceil 1.5D \rceil$ bins we try to keep $N_c - CD_c \leq \lceil D/2 \rceil$ for all colors c. For simplicity, we use an equivalent inequality of

$$\alpha_c = N_c - \left\lceil \frac{D}{2} \right\rceil \leq CD_c. \quad (1)$$

If we can keep the inequality valid, there can be no color c with $N_c > \lceil 1.5D \rceil$, else we get $CD_c \geq N_c - \lceil D/2 \rceil > \lceil 1.5D \rceil - \lceil D/2 \rceil = D$ which contradicts $CD_c \leq D$. Let the **main invariant** for a color c be Equation 1.

As $CD_c \geq 0$, keeping the invariant is easy for all colors with at most $\lceil D/2 \rceil$ bins. Also when there is only one color c with $N_c > \lceil D/2 \rceil$, we just put any non-c-item into a c-bin. Therefore, if a non-c-item comes, the number of c-bins N_c decreases and the current discrepancy CD_c decreases by at most one. (CD_c stays the same when it is zero.) Since both increase with an incoming c-item, we are keeping our main invariant (1) for the color c.

Moreover, there are at most two colors with strictly more than $\lceil D/2 \rceil$ bins, given that we have at most $\lceil 1.5D \rceil$ open bins. Thus we only have to deal with two colors having $N_c > \lceil D/2 \rceil$. We state the algorithm Balancing Any Fit for items of size zero.

Balancing Any Fit (BAF):

1. For an incoming c-item, if there are no bins or c-bins only, open a new bin and put the item into it.
2. Otherwise, if there is at most one color with the number of bins strictly more than $\lceil D/2 \rceil$, put an incoming c-item into a bin of color $c' = \arg\max_{c'' \neq c} N_{c''}$. If more colors have the same maximal number of bins, choose color c' arbitrarily among them, e.g., by First Fit. Among c'-bins, choose again arbitrarily, e.g., by First Fit.
3. Suppose that there are two colors b and w such that $N_{\mathrm{b}} > \lceil D/2 \rceil$ and $N_{\mathrm{w}} > \lceil D/2 \rceil$. If $c = \mathrm{w}$, put the item into a bin of color b. If $c = \mathrm{b}$, put the item into a bin of color w. Otherwise $c \notin \{\mathrm{b}, \mathrm{w}\}$; if $N_{\mathrm{b}} - \lceil D/2 \rceil < CD_{\mathrm{b}}$, put the item into a bin of color w, otherwise into a bin of color b.

As we discussed, keeping the main invariant (1) is easy in the first and the second case of the algorithm. Therefore we can conclude the following lemma.

Lemma 4.1. *Suppose that the main invariant holds for all colors before packing the t-th item and that there is at most one color c with $N_{c,t-1} > \lceil D_{t-1}/2 \rceil$ before the t-th item, i.e., the t-th item is packed using the first or the second case of the algorithm. Then the main invariant holds for all colors also after packing the t-th item.*

Most of the proof of 1.5-competitiveness of BAF thus deals with two colors having more than $\lceil D/2 \rceil$ bins. Without loss of generality let these two colors be black and white in the following and let us abbreviate them by b and w.

In the third case of the algorithm we have to choose either black or white bin for items of other colors than black and white, but the current discrepancy decreases for both black and white, while the number of bins stays the same for the color which we do not choose. So if $\alpha_{\mathrm{b}} = CD_{\mathrm{b}}$ and $\alpha_{\mathrm{w}} = CD_{\mathrm{w}}$, it is possible to force the algorithm to open more than $\lceil 1.5D \rceil$ bins. See Figure 5 for an example of such situation.

Therefore we need to prove that in the third case, i.e., when $N_{\mathrm{b}} > \lceil D/2 \rceil$ and $N_{\mathrm{w}} > \lceil D/2 \rceil$, at least one of inequalities $\alpha_{\mathrm{b}} \leq CD_{\mathrm{b}}$ and $\alpha_{\mathrm{w}} \leq CD_{\mathrm{w}}$ is strict. This motivates the following **secondary invariant:**

$$2\alpha_{\mathrm{b}} + 2\alpha_{\mathrm{w}} \leq CD_{\mathrm{b}} + CD_{\mathrm{w}} + 1\,. \tag{2}$$

If the secondary invariant holds, it is not hard to see that in the third case of the algorithm the choice of the bin maintains the main invariant. The tricky part of the proof is to prove the *base case* of the inductive proof of the secondary invariant. A natural proof would show the base case whenever b and w become the two colors with $N_{\mathrm{b}}, N_{\mathrm{w}} > \lceil D/2 \rceil$. However, we are not able to do that. Instead, we prove that the secondary invariant holds already at the moment when b and w become the two strictly most frequent colors on the top of the bins, i.e., $N_{\mathrm{b}} > N_c$ and $N_{\mathrm{w}} > N_c$ for all other colors c, which may happen much earlier, when the

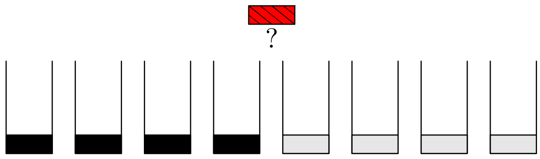

Figure 5: An example with $D = 5$ and $\lceil 1.5 \cdot D \rceil = 8$ (note that only top items in bins are shown). Suppose that $CD_{\mathrm{b}} = 1$ and $CD_{\mathrm{w}} = 1$, thus $N_{\mathrm{b}} = CD_{\mathrm{b}} + \lceil D/2 \rceil$ and $N_{\mathrm{w}} = CD_{\mathrm{w}} + \lceil D/2 \rceil$, i.e., the main invariant does not hold strictly for both black and white. If the next incoming red item goes into a black bin, then the adversary sends five white items and $D = 5$ after that, since CD_{w} decreases to zero. Hence the adversary forces nine open bins. The case in which the red item is packed into a white bin is symmetric.

number of their bins is significantly below $D/2$. After that, maintaining both invariants is relatively easy.

Theorem 4.2. Balancing Any Fit *is* 1.5*-competitive for items of size zero and an arbitrary number of colors. Precisely, it uses at most* $\lceil 1.5 \cdot OPT \rceil$ *bins.*

Proof. First we show that keeping the main invariant (1) for each color c, i.e., $\alpha_c \leq CD_c$, is sufficient for the algorithm to create at most $\lceil 1.5D \rceil$ bins. This implies both that the algorithm is well defined since there are at most two colors with $N_c > \lceil D/2 \rceil$, and that the algorithm is 1.5-competitive, since the maximal discrepancy equals the optimum.

Lemma 4.3. *After packing the t-th item, if we suppose that $N_{c,i} - \lceil D_i/2 \rceil \leq CD_{c,i}$ for all colors c and for all $i < t$, the algorithm uses at most $\lceil 1.5D_t \rceil$ bins.*

Proof. We prove the lemma by contradiction: Suppose that BAF opens a bin with the k-th item in the sequence (for $k \leq t$) and we exceed the $\lceil 1.5D_k \rceil$ limit, but before the k-th item there were at most $\lceil 1.5D_{k-1} \rceil$ bins. Thus $D_k = D_{k-1}$, since if $D_k = D_{k-1} + 1$, then the bound also increases with the k-th item.

Let c be the color of the k-th item. Let the ℓ-th item be the last non-c-item before the k-th, so only c-items come after the ℓ-th item. None of c-items from the $(\ell + 1)$-st to the k-th increase the maximal discrepancy D, otherwise if one such item increases D, then all following such items also do. Thus $D_\ell = D_k$.

The algorithm must have received $\lceil 1.5D_\ell \rceil + 1 - N_{c,\ell}$ of c-items after the ℓ-th item to open $\lceil 1.5D_\ell \rceil + 1$ bins, but then

$$CD_{c,k} = CD_{c,\ell} + \lceil 1.5D_\ell \rceil + 1 - N_{c,\ell} \geq N_{c,\ell} - \left\lceil \frac{D_\ell}{2} \right\rceil + \lceil 1.5D_\ell \rceil + 1 - N_{c,\ell} = D_\ell + 1$$

where we used the main invariant for the inequality which holds, because $\ell < k \leq t$. We get a contradiction, since $CD_{c,k} \leq D_k = D_\ell$. □

We have to deal with the case in which $N_{\mathrm{b}} > \lceil D/2 \rceil$ and $N_{\mathrm{w}} > \lceil D/2 \rceil$. We show that we can maintain the secondary invariant (2), while black and white are the two strictly most frequent colors of bins (even if $N_{\mathrm{b}} \leq \lceil D/2 \rceil$ or $N_{\mathrm{w}} \leq \lceil D/2 \rceil$). Then we prove that the secondary invariant starts to hold when black and white become the two strictly most frequent colors, i.e., $N_c < N_{\mathrm{b}}$ and $N_c < N_{\mathrm{w}}$ for all other colors c; this step must precede the time when the number of bins for the second color gets over the $\lceil D/2 \rceil$ limit. Therefore we prove by induction that the secondary invariant holds in certain intervals of the input sequence.

Lemma 4.4. *Suppose that black and white are the two strictly most frequent colors of bins before packing the t-th item and that the main invariant (1) holds for all colors and the secondary invariant (2) also holds before packing the t-th item, i.e., $N_{c,t-1} - \lceil D_{t-1}/2 \rceil \le CD_{c,t-1}$ for all colors c and $2\alpha_{\mathrm{b},t-1} + 2\alpha_{\mathrm{w},t-1} \le CD_{\mathrm{b},t-1} + CD_{\mathrm{w,t-1}} + 1$. Then the main invariant for all colors and the secondary invariant for black and white hold also after packing the t-th item.*

Proof. First we suppose that the maximal discrepancy D is not changed by the t-th item. We start with showing that the main invariant holds after packing the t-th item. If the t-th item is packed using the second case of BAF, the main invariant holds by Lemma 4.1. (Note that the t-th item cannot be packed using the first case of the algorithm, since $N_{\mathrm{b},t-1} > 0$ and $N_{\mathrm{w},t-1} > 0$.)

Otherwise, if the t-th item is packed using the third case, it holds that $\alpha_{\mathrm{b},t-1} > 0$ and $\alpha_{\mathrm{w},t-1} > 0$. The main invariant holds for any color c other than black and white, because $N_{c,t-1} < \lceil D_{t-1}/2 \rceil$ which implies $N_{c,t} \le \lceil D_t/2 \rceil$.

To prove the main invariant for black and white, we show by contradiction that the secondary invariant (2) guarantees that $\alpha_{\mathrm{b},t-1} < CD_{\mathrm{b},t-1}$ or $\alpha_{\mathrm{w},t-1} < CD_{\mathrm{w},t-1}$. Otherwise, if $\alpha_{\mathrm{b},t-1} \ge CD_{\mathrm{b},t-1}$ and $\alpha_{\mathrm{w},t-1} \ge CD_{\mathrm{w},t-1}$, the secondary invariant becomes $2\alpha_{\mathrm{b},t-1} + 2\alpha_{\mathrm{w},t-1} \le CD_{\mathrm{b},t-1} + CD_{\mathrm{w},t-1} + 1 \le \alpha_{\mathrm{b},t-1} + \alpha_{\mathrm{w},t-1} + 1$ which is a contradiction. Note that we used that $\alpha_{\mathrm{w},t-1}$ and $\alpha_{\mathrm{b},t-1}$ are integral and positive.

We now distinguish three cases according to the color of the t-th item:

- The t-th item is black: Then it is packed into a white bin. The main invariant for black holds after packing the item, because both N_{b} and CD_{b} increase, and the main invariant for white holds, since N_{w} decreases and CD_{w} decreases by at most one. (CD_{w} stays the same when it is zero.)
- The t-th item is white: The situation is symmetric to the previous case.
- The t-th item has some other color: We pack it into a white bin if $N_{\mathrm{b},t-1} - \lceil D_{t-1}/2 \rceil < CD_{\mathrm{b},t-1}$, otherwise into a black bin. If it is packed into a white bin, N_{w} decreases and CD_{w} decreases by at most one, thus the main invariant holds for white. The main invariant holds for black too, since N_{b} stays the same and CD_{b} decreases by at most one, but the main invariant held strictly for black before packing the t-th item.
  If the t-th item is packed into a black bin, we have $N_{\mathrm{w},t-1} - \lceil D_{t-1}/2 \rceil < CD_{\mathrm{w},t-1}$ and the situation is symmetric as if the t-th item is packed into a white bin.

It remains to show that the t-th item does not violate the secondary invariant. There are again three cases according to the color of the t-th item:

- The t-th item is black: Then it is packed into a white bin in both the second and third cases of the algorithm. Thus α_{b} increases and α_{w} decreases, so the left-hand side of the inequality (2) stays the same. Also the right-hand side does not change or even increases as CD_{b} increases and CD_{w} decreases by at most one. (CD_{w} stays the same when it is zero.)
- The t-th item is white: The situation is symmetric to the previous case.
- The t-th item has another color than black and white: Then it is packed into a white or black bin in both the second and third cases of the algorithm. Thus one of α_{w} and α_{b} decreases and the other one stays the same, while both CD_{b} and CD_{w} decrease by

at most one. The secondary invariant holds as the left-hand side decreases by two and the right-hand side decreases by at most two.

Otherwise D increases with an incoming item, thus $\alpha_{c'}$ for each color c' decreases if D becomes odd. We follow the same proof as if D stays the same, and the eventual additional decrease of $\alpha_{c'}$ can only decrease the left-hand sides of the main and secondary invariants. □

Note that in the previous proof, α_{b} or α_{w} can be negative in the secondary invariant. We show the base case of the secondary invariant, i.e., that it starts to hold when two colors become the two strictly most frequent colors of bins.

Lemma 4.5. *Suppose that after packing the k-th item it starts to hold that $N_c < N_{\mathrm{b}}$ and $N_c < N_{\mathrm{w}}$ for all other colors c, i.e., black and white become the two strictly most frequent colors. Suppose also that the main invariant holds all the time before packing the k-th item. Then $2\alpha_{\mathrm{b},k} + 2\alpha_{\mathrm{w},k} \le CD_{\mathrm{b},k} + CD_{\mathrm{w},k} + 1$, i.e., the secondary invariant holds after packing the k-th item.*

Proof. Assume without loss of generality that $N_{\mathrm{b},k} \ge N_{\mathrm{w},k}$. If $N_{\mathrm{b},k} = N_{\mathrm{w},k}$, we also suppose without loss of generality that $N_{\mathrm{b},k-1} \ge N_{\mathrm{w},k-1}$.

First we show by contradiction that always $N_{\mathrm{b},k-1} \ge N_{\mathrm{w},k-1}$. Otherwise if $N_{\mathrm{b},k-1} < N_{\mathrm{w},k-1}$, then $N_{\mathrm{b},k} > N_{\mathrm{w},k}$ (note that $N_{\mathrm{b},k} = N_{\mathrm{w},k}$ would imply $N_{\mathrm{b},k-1} \ge N_{\mathrm{w},k-1}$). This can happen only when a black item is packed into a white bin, but then the numbers of black and white bins are swapped, hence black and white were already the two strictly most frequent colors before the k-th item which contradicts the assumption of the lemma. We conclude that $N_{\mathrm{b},k} \ge N_{\mathrm{w},k}$ and $N_{\mathrm{b},k-1} \ge N_{\mathrm{w},k-1}$.

We bound the number of non-black bins before the k-th item arrive from above by $\lceil 1.5D_{k-1}\rceil - N_{\mathrm{b},k-1} = D_{k-1} - \alpha_{\mathrm{b},k-1}$, since there are at most $\lceil 1.5D_{k-1}\rceil$ bins by Lemma 4.3 (we use that the main invariant holds before packing the k-th item). As we have $N_{\mathrm{b},k-1} \ge N_{\mathrm{w},k-1}$ and black and white are not the two strictly most frequent colors before the k-th item, there must be a color $\mathrm{r} \notin \{\mathrm{b}, \mathrm{w}\}$ such that $N_{\mathrm{r},k-1} \ge N_{\mathrm{w},k-1}$ (let the color be red without loss of generality). Therefore the number of white bins is at most half of the number of non-black bins, i.e., $N_{\mathrm{w},k-1} \le (D_{k-1} - \alpha_{\mathrm{b},k-1})/2$.

We show by contradiction that the k-th item must be packed using the second case of the algorithm. (Note that BAF cannot use the first case, since otherwise all bins would have the same color after packing the k-th item.) If the item is packed using the third case, it must hold that $N_{\mathrm{b},k-1} \ge \lceil D_{k-1}/2\rceil + 1$ and $N_{\mathrm{r},k-1} \ge \lceil D_{k-1}/2\rceil + 1$. Since there are at most $\lceil 1.5D_{k-1}\rceil$ bins by Lemma 4.3, we get $N_{\mathrm{w},k-1} \le \lfloor D_{k-1}/2\rfloor - 2$, but then the k-th item cannot cause $N_{\mathrm{w},k} > N_{\mathrm{r},k}$.

Therefore BAF packs the k-th item using the second case and it follows by Lemma 4.1 that the main invariant holds for all colors after packing the k-th item.

Observe that by packing the k-th item, the number of white bins must increase, or the number of red bins must decrease, or both. Note that the k-th item can have any color, not only white. We distinguish two cases: The k-th item is white and the k-th item is not white.

If the k-th item is white, we have $\alpha_{\mathrm{b},k} \le \alpha_{\mathrm{b},k-1}$, as the number of black bins does not increase (note that there is an inequality because of a possible increase of D or a decrease of N_{b}). We get

$$\alpha_{\mathrm{w},k} = N_{\mathrm{w},k} - \left\lceil \frac{D_k}{2} \right\rceil = N_{\mathrm{w},k-1} + 1 - \left\lceil \frac{D_k}{2} \right\rceil \le \frac{D_{k-1} - \alpha_{\mathrm{b},k-1}}{2} + 1 - \left\lceil \frac{D_k}{2} \right\rceil$$
$$\le \frac{D_k - \alpha_{\mathrm{b},k}}{2} + 1 - \left\lceil \frac{D_k}{2} \right\rceil \le -\frac{\alpha_{\mathrm{b},k}}{2} + 1.$$

where we used $N_{\mathrm{w},k-1} \le (D_{k-1} - \alpha_{\mathrm{b},k-1})/2$ for the first inequality and $D_{k-1} - \alpha_{\mathrm{b},k-1} \le D_k - \alpha_{\mathrm{b},k}$ for the second inequality which follows from $\alpha_{\mathrm{b},k} \le \alpha_{\mathrm{b},k-1}$.

We know that $\alpha_{\mathrm{w},k} \le -\alpha_{\mathrm{b},k}/2 + 1$. Therefore

$$2\alpha_{\mathrm{w},k} + 2\alpha_{\mathrm{b},k} \le -\alpha_{\mathrm{b},k} + 2 + 2\alpha_{\mathrm{b},k} = \alpha_{\mathrm{b},k} + 2 \le CD_{\mathrm{b},k} + 2 \le CD_{\mathrm{w},k} + CD_{\mathrm{b},k} + 1$$

where we use the main invariant (1) for black color for the second inequality and $CD_{\mathrm{w},k} \ge 1$ for the third inequality which holds, because the k-th item is white.

Otherwise the k-th item is not white and it is packed into a bin of another color than black and white, otherwise N_b or N_w decreases, thus black and white cannot become the two strictly most frequent colors. After packing the k-th item we have $\alpha_{\mathrm{b},k} \le \alpha_{\mathrm{b},k-1} + 1$, as the k-th item may be black, therefore $D_{k-1} - \alpha_{\mathrm{b},k-1} \le D_k - \alpha_{\mathrm{b},k} + 1$. Since the number of white bins does not change, we get

$$\alpha_{\mathrm{w},k} = N_{\mathrm{w},k} - \left\lceil \frac{D_k}{2} \right\rceil = N_{\mathrm{w},k-1} - \left\lceil \frac{D_k}{2} \right\rceil \le \frac{D_{k-1} - \alpha_{\mathrm{b},k-1}}{2} - \left\lceil \frac{D_k}{2} \right\rceil$$
$$\le \frac{D_k - \alpha_{\mathrm{b},k} + 1}{2} - \left\lceil \frac{D_k}{2} \right\rceil \le -\frac{\alpha_{\mathrm{b},k}}{2} + \frac{1}{2}.$$

In this case we have $\alpha_{\mathrm{w},k} \le -\alpha_{\mathrm{b},k}/2 + \frac{1}{2}$. Therefore

$$2\alpha_{\mathrm{w},k} + 2\alpha_{\mathrm{b},k} \le -\alpha_{\mathrm{b},k} + 1 + 2\alpha_{\mathrm{b},k} = \alpha_{\mathrm{b},k} + 1 \le CD_{\mathrm{b},k} + 1 \le CD_{\mathrm{w},k} + CD_{\mathrm{b},k} + 1$$

where we use the main invariant (1) for black color for the second inequality. Hence the secondary invariant (2) holds. □

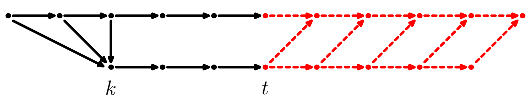

Figure 6: An illustration of dependencies of the main and secondary invariants. The horizontal axis represents time. An invariant at a certain time represented by a point P follows from invariants from which there is an arrow to P. After packing the k-th item (time k) black and white become the two strictly most frequent and after the t-th item (time t) it starts to hold that $N_\mathrm{b} > \lceil D/2 \rceil$ and $N_\mathrm{w} > \lceil D/2 \rceil$. Thus in the black part of the figure, BAF uses the first or the second case of the algorithm, while in the dashed (red) part BAF uses the third case of the algorithm.

We now complete the proof of the theorem by putting everything together. Precisely, we prove that the main invariant holds during the whole run of the algorithm by induction. The main invariant for each color holds trivially at the beginning before any item comes. When the t-th item is packed, there are two cases:

- No two colors were the strictly most frequent before the t-th item: BAF keeps the main invariant for all colors by Lemma 4.1, since it must pack the t-th item with the first or the second case of the algorithm. If two colors become the two strictly most frequent after packing the t-th item, the secondary invariant starts to hold by Lemma 4.5; otherwise the secondary invariant is irrelevant in this case.
- Two colors were the strictly most frequent: Let these two colors be black and white without loss of generality. Then the main invariant for all colors and the secondary invariant for black and white are kept by Lemma 4.4 (even if black and white are not the two strictly most frequent after the t-th item).
  It may happen that the two strictly most frequent colors change after packing the t-th item (e.g., to black and red). The main invariant for all colors still follows by Lemma 4.4, but the secondary invariant for the new strictly most frequent colors follows by Lemma 4.5.

See Figure 6 for an illustration of dependencies of the invariants.

Therefore we can keep the main invariant $N_c - \lceil D/2 \rceil \leq CD_c$ for all colors c during the whole run of the algorithm and the theorem follows by Lemma 4.3. □

4.2 3.5-competitive Algorithm

We now show that there is a constant competitive online algorithm even for items of sizes between 0 and 1. We combine algorithms Pseudo from [2] and our algorithm BAF that is 1.5-competitive for zero-size items. The algorithm Pseudo uses *pseudo bins* which are bins of unbounded capacity.

> Pseudo-BAF:
> 1. First pack an incoming item into a pseudo bin using the algorithm BAF (treat the item as a zero-size item).
> 2. In each pseudo bin, items are packed into unit capacity bins using Next Fit.

Theorem 4.6. *The algorithm* Pseudo-BAF *for Colored Bin Packing is absolutely* 3.5*-competitive. In the parametric case when items have size at most* $1/d$*, for a real* $d \geq 2$*, it uses at most* $\lceil (1.5 + d/(d-1))OPT \rceil$ *bins. Moreover, the analysis is asymptotically tight.*

Proof. In the general case for items between 0 and 1 we know that two consecutive bins in one pseudo bin have total size strictly more than one, since no two consecutive items of the same color are in a pseudo bin. In each pseudo bin we match each bin with an odd index with the following bin with an even index, therefore we match all bins except at most one in each pseudo bin. Moreover, the total size of a pair of matched bins is more than one. Therefore the number of matched bins is strictly less than $2 \cdot Vol \leq 2 \cdot OPT$ (where Vol is the total size of all items), thus at most $2 \cdot OPT - 1$. The number of unmatched bins is at most the number of pseudo bins created by the algorithm BAF which uses at most $\lceil 1.5 \cdot Dis \rceil \leq \lceil 1.5 \cdot OPT \rceil \leq 1.5 \cdot OPT + 0.5$ bins. Summing both bounds, the algorithm Pseudo-BAF creates at most $3.5 \cdot OPT$ bins.

For the parametric case, inside each pseudo bin all real bins except the last one have level strictly more than $(d-1)/d$, so their number is strictly less than $d/(d-1) \cdot OPT$, i.e., at most $\lceil d/(d-1) \cdot OPT \rceil - 1$. The number of pseudo bins is still bounded by $\lceil 1.5 \cdot OPT \rceil$, thus the algorithm Pseudo opens at most $\lceil (1.5 + d/(d-1))OPT \rceil$ bins.

We show the tightness of the analysis by combining hard instances for Pseudo by Balogh et al. [2] and for BAF from the proof of Theorem 3.1. More concretely, for n (a big integer) let $\varepsilon = 1/(2n)$. The input consists of $n-1$ groups of three items, specifically $(n-1) \times \left(\overset{\text{white}}{\varepsilon}, \overset{\text{black}}{1}, \overset{\text{black}}{\varepsilon}\right)$.

The algorithm creates one pseudo bin containing every first and second item from each group and $n-1$ pseudo bins, each containing only the third item from a group. Moreover, the first pseudo bin is split into $2 \cdot (n-1)$ unit capacity bins (each item is in a separate bin), so there are $3 \cdot (n-1)$ bins. The optimum for $n-1$ groups is n, because we can pack all tiny items together in one bin and the total size of all items is strictly more than $n-1$.

Then the input continues by the hard instance with zero-size items from the proof of Theorem 3.1 and BAF creates additional $\lceil (n-1)/2 \rceil$ pseudo bins, while the optimum on the instance is $n-1$. Pseudo-BAF now have $\lceil 3.5 \cdot (n-1) \rceil$ bins. Observe that the optimal packing for $n-1$ groups does not need to be changed to put there zero-size items, thus $OPT = n$.

To prove tightness of the analysis for the parametric case for an integer $d \geq 2$, we use a modification of the first part of the hard instance by Balogh et al. [2] on which Pseudo creates at least $(d-1)n + dn$ bins, while its optimal packing needs $(d-1)n + 1$ bins. The input continues by the hard instance with zero-size items like in the case of items of arbitrary size and force Pseudo-BAF to create additional $\lceil (d-1)n/2 \rceil$ bins without increasing the optimum. Therefore Pseudo-BAF ends up with asymptotically $(1.5 + d/(d-1))OPT$ bins.

For a real $d \geq 2$, it is possible to use a similar sequence and show a lower bound of $(1.5 + d/(d-1))$ on the competitive ratio of Pseudo. □

4.3 Classical Any Fit Algorithms and Pseudo

We analyze algorithms First Fit, Best Fit and Worst Fit and we find that they are not constant competitive. Their competitiveness cannot be bounded by any function of the number of colors even for only three colors, in contrast to their good performance for two colors.

We also show the same negative result for the algorithm Pseudo from [3, 2] which first packs items by First Fit into pseudo bins and then apply Next Fit in each pseudo bin and which is 3-competitive for *Black and White Bin Packing*.

Proposition 4.7. First Fit*,* Best Fit *and* Pseudo *are not constant competitive.*

Proof. The input consists of $4n$ items which can be packed into two bins, but FF, BF and Pseudo create $n+1$ bins where n is an arbitrary integer.

Let $\varepsilon = 1/(4n)$. The instance is $n \times \left(\overset{\text{black}}{\varepsilon}, \overset{\text{black}}{\varepsilon}, \overset{\text{white}}{\varepsilon}, \overset{\text{red}}{\varepsilon}\right)$. An optimal packing can be obtained by putting black items from each group into the first and the second bin, the white item into the first bin and the red item into the second bin.

FF and BF pack the first group into two bins, both with a black bottom item, and white and red items are assigned to the first bin. The first black item, the white item and the red item from each following group are packed into the first bin, while the second black item is packed into a new bin. Therefore these algorithms create one bin with all white and red items and all first black items from each group and n bins with a single black item.

The packing created by Pseudo on this instance is the same as the packing by FF, since each pseudo bin contains only a single bin (the total size of all items is 1) and items are put into pseudo bins by FF.

Hence FF, BF and PSEUDO create $(n+1)/2 \cdot OPT$ bins. □

Note that WF on such instance creates an optimal packing, but the instance can be modified straightforwardly to obtain a bad behavior for WF.

Proposition 4.8. WORST FIT *is not constant competitive.*

Proof. The instance is similar to the one in the previous proof, but sizes of items are different in each group. Let $\varepsilon = 1/(2n)$ and let $\delta = 1/(6n^2+1)$. The instance is $n \times \left(\substack{\text{black} \\ \delta}, \substack{\text{black} \\ \varepsilon}, \substack{\text{white} \\ \delta}, \substack{\text{red} \\ \delta}\right)$.

We observe that the optimal packing does not change with other sizes. However, WF packs all δ-items into the first bin, i.e., first black items from each group and all white and red items, since the level of the first bin stays at most $(3n)/(6n^2+1)$, which is less than $\varepsilon = 1/(2n)$. Therefore all second black items are packed into separate bins and WF creates $n+1$ bins, while the optimum is two. □

5 Any Fit Algorithms for Two Colors

For *Black and White Bin Packing*, we improve the upper bound on the absolute competitive ratio of ANY FIT algorithms from 5 to 3. Then we show that WORST FIT performs even better for items with size of at most $1/d$ (for $d \geq 2$) as it is absolutely $(1 + d/(d-1))$-competitive in this case. Both bounds are tight by the results of Balogh et al. [2] (they show a lower bound on competitiveness of WF only for an integer d, but it is possible to apply a similar proof in the case of a real d). Therefore WF matches the performance of PSEUDO, the online algorithm with the best competitive ratio known so far. Note that for infinitesimally small items WF is 2-competitive, while BF and FF remain 3-competitive.

5.1 Competitiveness of Any Fit Algorithms

Theorem 5.1. *Any algorithm in the* ANY FIT *family is absolutely* 3*-competitive for Black and White Bin Packing.*

Proof. We use the following notation: An item is *small* when its size is less than 0.5 and *big* otherwise. Similarly *small bins* have level less than 0.5 and *big bins* have level at least 0.5.

We assign bins into *chains* — sequences of bins in which all bins except the last must be big. If there is only one bin in a chain, it must be big. Moreover, it is required that the bottom item in the i-th bin of a chain cannot be added into the $(i-1)$-st bin, even if it would have the right color, i.e., it is too big to be put into the $(i-1)$-st bin. We will split chains such that our chains will have at most two bins, so the average level of bins in each chain is clearly at least 0.5.

A bin is contained in at most one chain. We call a bin that is not in a chain a *separated* bin. We create chains such that all big bins are in a chain and only as few small bins as possible remain separated.

Since the average level of bins in chains is at least 0.5, it follows that the total number of bins in all chains is bounded from above by $2 \cdot OPT$. We want to bound the number of separated bins from above by the maximal color discrepancy *Dis* which yields the 3-competitiveness of AF.

We define a process of assigning bins into chains. We simply try to put as many bins into chains as possible, but we add a bin into a chain only when the first item in the bin cannot be added into the last bin of the chain regardless of the color of the item. Note that the top item in the first bin and the bottom item in the second bin may have the same color.

Formally, when an item from the input sequence is added we do the following:

- The item is added into a bin in a chain: Nothing happens with chains or separated bins.
- The item is added into a small separated bin: If the bin becomes big, we create a new chain from the bin, otherwise the bin stays separated.
- The item is big and creates a new bin: The newly created bin forms a new chain.
- The item is small and creates a new bin: If there is a chain such that the incoming item cannot be packed into the last bin of the chain by capacity, i.e., even if it would have the right color, we add the newly created bin into the chain. (Note that the last bin in the chain must be big.) If there is no such chain, the new bin is separated.

Moreover, whenever a chain has two big bins we split it into two chains, each containing one big bin. Therefore each chain is either one big bin, or a big bin and a small bin. The intuitive reason for splitting chains is that we can put more newly created small bins into chains.

If there is no separated bin at the end (after the last item is added), we have created at most $2 \cdot OPT$ bins. Otherwise we define k and t as indices of incoming items and show that the color discrepancy of items between the k-th and the t-th item is at least the number of separated bins at the end.

Let t be the index of an item that created the last bin that is separated when it is created (the t-th item must be small). Suppose without loss of generality that the t-th item is black. Note that a small item that comes after the t-th item can create a bin, but we put the bin into a chain immediately, therefore the number of separated bins can only decrease after adding the t-th item.

Let b_i be the number of small black bins, i.e., bins with a black item on the top, and w_i be the number of small white bins after adding the i-th item from the sequence. From the definition of t we know that $w_t = 0$.

We define k as the biggest $i \leq t$ such that $b_i = 0$, i.e., there is no small black bin (if $b_i > 0$ for all $i \geq 1$ we set $k = 0$). Clearly the $(k+1)$-st item must be small and black. Note that there can be some separated white bins and possibly some other small white bins in chains, but there is no separated black bin when the $(k+1)$-st item arrives. Let W be the set of white bins that are separated after adding the k-th item. Before adding the t-th item and creating the last bin, all bins in W must have a black item on the top, or become big bins in chains (thus $k \leq t - |W|$). See Figure 7 for an example of the situation after packing the k-th item.

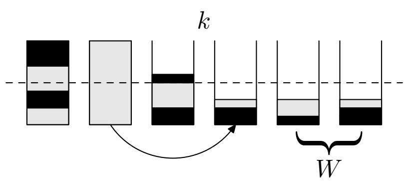

Figure 7: An example of the situation after packing the k-th item. A chains with two bins is depicted by an arrow (the arrow goes from the first to the second bin). The dashed line is at height 0.5.

Let *new items* be items with an index i for some value of i such that $k < i \leq t$. We want to bound the number of separated bins after adding the t-th item by the color discrepancy. Note that these bins are small by the process of assigning bins into chains. We observe that all separated bins must have a black item on the top before adding the t-th item and also all chains have a black item on the top in the last bin, otherwise the bin created by the t-th item would be added in a chain.

Hence the number of black items is greater by one than the number of white items in separated bins with a black item at the bottom. Separated bins created with a new item must have a black item at the bottom, since otherwise there cannot be a small black bin and $b_i = 0$ for $k < i < t$.

Separated bins from the set W can have the same number of black and white items before adding the t-th item, but in each such bin there is one more new black item than new white items, since the first and the last such items are black.

Now we look at new items which are packed into bins that are in chains after adding the t-th item. We call such an item a *link*. Note that some links can be at first packed into separated bins, but these bins are put into chains before adding the t-th item. It suffices to show the following lemma.

Lemma 5.2. *In each chain the number of black links is at least the number of white links after adding the t-th item.*

Proof. Since the t-th item is small, it cannot be placed into any existing bin and the bin created by the t-th item is not put into an existing chain, no chain ends with a white item (not even if that item is in a big bin). Since new items have alternating colors in any bin, the claim holds for chains with only one bin.

All other chains have two bins by construction. Consider such a chain. If the second bin which is small was created before the $(k+1)$-st item arrived, it had a white item on the top when the $(k+1)$-st item arrived, since $b_k = 0$. Thus the second bin has a link and the first link packed into it is black.

If the second bin was created later, at some time i such that $k < i < t$, then the first link in it is small since otherwise we would start a new chain, and it is black since otherwise we have $b_i = 0$ for $i > k$ which contradicts the choice of k.

In all cases, the second bin contains a link and the first link in it is black. Since the chain ends with a black item, the second bin has one more black link than white links, and the first bin has at most one more white link than black links. The lemma follows. □

See Figure 8 for an example of the situation after packing the t-th item.

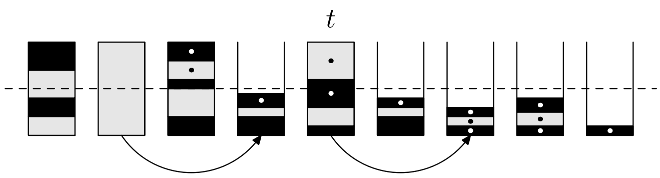

Figure 8: The situation after packing the t-th item into a new bin. New items, i.e., items with an index i such that $k < i \leq t$, are marked with a dot.

Let S be the set of separated bins at the end. We found out that when we focus on new items, i.e., items with an index i such that $k < i \leq t$, there is one more such black item than such white items in all separated bins and at least the same number of such items of both colors in bins in all chains, i.e., links. Moreover, after the t-th item comes the number of separated bins can only decrease, since no separated bin is created. So we have bounded the size of S from above by the color discrepancy between the $(k+1)$-st and the t-th item:

$$|S| \leq \left| \sum_{\ell=k+1}^{t} s_\ell \right| \leq Dis$$

where s_i is 1 when the i-th item is white and -1 otherwise. Note that some items after the t-th item can create a bin, but such bins are put into chains right away by the definition of t. □

5.2 Competitiveness of the Worst Fit Algorithm

The Worst Fit algorithm performs in fact even better when all items are small which we prove similarly to the proof of Theorem 5.1.

Theorem 5.3. *Suppose that all items in the input sequence have size of at most $1/d$, for a real $d \geq 2$. Then* Worst Fit *is absolutely $(1 + d/(d-1))$-competitive for Black and White Bin Packing.*

Proof. We divide bins created by WF into sets B (big bins) and S (small bins). Each *big bin* has level at least $(d-1)/d$, thus $|B| \leq d/(d-1) \cdot OPT$. *Small bins* are smaller than $(d-1)/d$, thus they can receive any item of the right color. Note that a newly created bin is always small for any $d \geq 2$. We show that $|S|$ is bounded by the maximal color discrepancy *Dis* and we obtain that WF is $(1 + d/(d-1))$-competitive.

As items are arriving, we count the number of small black bins, i.e., bins with a black item on the top and with level less than $(d-1)/d$, and the number of small white bins. Let b_i and w_i be the number of small black and white bins, respectively, after adding the i-th item from the sequence.

If $b_n = 0$ and $w_n = 0$, i.e., there is no small bin at the end, WF created at most $d/(d-1) \cdot OPT$ bins. Otherwise suppose without loss of generality that the last created bin has a black item at the bottom. Let t be the index of the black item that created the last bin. It holds that $w_t = 0$, since otherwise the t-th item would be packed into a small white bin.

Let k be the largest index smaller than t for which $b_k = 0$ (if $b_i > 0$ for all $i \in \{1, \ldots, t\}$, we set $k = 0$). The $(k+1)$-st item must be black. We observe that any bin created after this point has a black item at the bottom, otherwise $b_i = 0$ for some i such that $k < i < t$. Note that w_k can be greater than 0, i.e., there can be some small white bins and the $(k+1)$-st item is packed into one of them. Let W be the set of these bins. Before adding the t-th item and creating the last bin, all bins in W must have a black item on the top, or become big bins (thus $k \leq t - |W|$). See Figure 9 for an example of the situation after packing the k-th item.

Let *new items* be items with an index i for some value of i such that $k < i \leq t$. We want to bound the number of small bins after adding the t-th item by the color discrepancy. We already observed that all these bins must have a black item on the top. Hence for small bins with a black item at the bottom the number of black items is greater by one than the number of white items. Small bins from the set W can have the same number of black and white items, but in each such bin there is one more new black item than new white items, since the first such item is black.

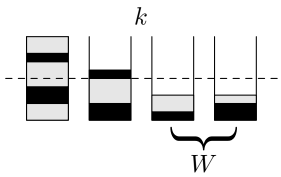

Figure 9: An example of the situation after packing the k-th item for $d = 2$. Bins under the dashed line at height of $1/d = 0.5$ are small.

Now we look at new items which are packed into bins that are big after the t-th item comes. It suffices to show that the number of such new black items is at least the number of such new white items. We observe that WF packs any new white item into an existing small bin, otherwise $b_\ell = 0$ for some ℓ such that $k < \ell < t$. Hence any new white item must be packed into a bin created after the k-th item (therefore with a new black item at the bottom), or into a bin from the set W. Since the first item that is assigned to a bin from W after the k-th item is black, our claim holds. See Figure 10 for an example of the situation after packing the t-th item.

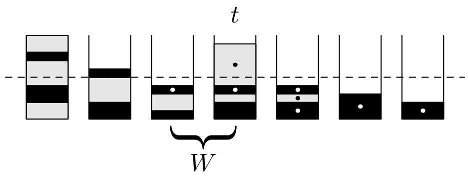

Figure 10: The situation after packing the t-th item into a new bin. New items, i.e., items with an index i such that $k < i \leq t$, are marked with a dot.

Note that this matching of black and white items in big bins would fail for algorithms like Best Fit or First Fit, since they can put a white item into a big bin created before the k-th item and not contained in W.

We found out that when we focus on new items, i.e., items with an index i for some value of i such that $k < i \leq t$, there is one more such black item than such white items in all small bins and at least as many such black items as such white items in all big bins. Moreover, after the t-th item comes the number of small bins $|S|$ can only decrease, since no bin is created. So we bound $|S|$ from above by the color discrepancy between the $(k+1)$-st and the t-th item:

$$|S| \leq \left| \sum_{\ell=k+1}^{t} s_\ell \right| \leq Dis$$

where s_i is 1 when the i-th item is white and -1 otherwise.

Note that the last bin is already counted in the color discrepancy, since its bottom item is black and has index t. □

Conclusions and Open Problems

The *Colored Bin Packing* for zero-size items is completely solved.

For items of arbitrary size, our online algorithm still leaves a gap between our lower bound 2.5 and our upper bound of 3.5. The upper bounds are only 0.5 higher than for two colors (*Black and White Bin Packing*) where a gap between 2 and 3 remains for general items.

Classical algorithms FF, BF and WF, although they maintain a constant approximation for two colors, start to behave badly when we introduce the third color. For two colors, we now know their exact behavior. In fact, all Any Fit algorithms are absolutely 3-competitive which is a tight bound for FF, BF and WF. However, for items of size up to $1/d$, $d \geq 2$, FF and BF remain 3-competitive, while WF has the absolute competitive ratio $1 + d/(d-1)$. Thus we now know that even the simple Worst Fit algorithm matches the performance of Pseudo, the online algorithm with the best competitive ratio known so far. It is also an interesting question whether it holds that Any Fit algorithms cannot be better than 3-competitive for two colors.

Acknowledgments

We would like to thank anonymous reviewers for careful reading and many useful suggestions.

References

[1] L. Babel, B. Chen, H. Kellerer, and V. Kotov. Algorithms for on-line bin-packing problems with cardinality constraints. *Discrete Applied Mathematics*, 143:238–251, 2004.

[2] J. Balogh, J. Békési, G. Dósa, L. Epstein, H. Kellerer, and Z. Tuza. Online results for black and white bin packing. *Theory of Computing Systems*, volume 56, issue 1, pages 137–155, 2015.

[3] J. Balogh, J. Békési, G. Dósa, H. Kellerer, and Z. Tuza. Black and white bin packing. In *Approximation and Online Algorithms*, LNCS 7846, pages 131–144. Springer, 2013.

[4] J. Balogh, J. Békési, and G. Galambos. New lower bounds for certain classes of bin packing algorithms. *Theoretical Computer Science*, 440-441:1–13, 2012.

[5] J. Balogh, J. Békési, G. Dósa, J. Sgall, and R. van Stee. The optimal absolute ratio for online bin packing. In Proc. of the 26th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA), pages 1425–1438. ACM-SIAM, 2015.

[6] M. Böhm, J. Sgall, and P. Veselý. Online Colored Bin Packing. In *Approximation and Online Algorithms*, LNCS 8952, pages 35–46. Springer, 2015.

[7] M. Chrobak, J. Sgall, and G. J. Woeginger. Two-bounded-space bin packing revisited. In *European Symposium on Algorithms (ESA)*, LNCS 6942, pages 263–274. Springer, 2011.

[8] E. Coffman Jr., J. Csirik, G. Galambos, S. Martello, and D. Vigo. Bin packing approximation algorithms: Survey and classification. In P. M. Pardalos, D.-Z. Du, and R. L. Graham, editors, *Handbook of Combinatorial Optimization*, pages 455–531. Springer, 2013.

[9] J. Csirik and D. S. Johnson. Bounded space on-line bin packing: Best is better than first. *Algorithmica*, 31(2):115–138, 2001.

[10] G. Dósa and L. Epstein. Colorful bin packing. In *Algorithm Theory – SWAT*, LNCS 8503, pages 170–181. Springer, 2014.

[11] G. Dósa and L. Epstein. Online bin packing with cardinality constraints revisited. *ArXiv e-prints 1404.1056*, Apr. 2014.

[12] G. Dósa and J. Sgall. First Fit bin packing: A tight analysis. In *30th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 538–549, Dagstuhl, Germany, 2013.

[13] G. Dósa and J. Sgall. Optimal analysis of Best Fit bin packing. In *Automata, Languages, and Programming (ICALP)*, LNCS 8572, pages 429–441. Springer, 2014.

[14] G. Dósa, Z. Tuza, and D. Ye. Bin packing with "largest in bottom" constraint: tighter bounds and generalizations. *Journal of Combinatorial Optimization*, 26(3):416–436, 2013.

[15] L. Epstein. Online bin packing with cardinality constraints. *SIAM Journal on Discrete Mathematics*, 20, 2006.

[16] L. Epstein. On online bin packing with LIB constraints. *Naval Research Logistics*, 56(8):780–786, 2009.

[17] L. Finlay and P. Manyem. Online LIB problems: Heuristics for bin covering and lower bounds for bin packing. *RAIRO Operetions Research*, 39(3):163–183, 2005.

[18] H. Fujiwara and K. Kobayashi. Improved Lower Bounds for the Online Bin Packing Problem with Cardinality Constraints In *Computing and Combinatorics*, LNCS 7936, pages 518–530. Springer, 2013.

[19] D. Johnson. *Near-optimal Bin Packing Algorithms.* Massachusetts Institute of Technology, project MAC. Massachusetts Institute of Technology, 1973.

[20] K. L. Krause, V. Y. Shen, and H. D. Schwetman. Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems. *J. ACM*, 22:522–550, 1975.

[21] C. C. Lee and D. T. Lee. A simple on-line bin-packing algorithm. *J. ACM*, 32:562–572, 1985.

[22] P. Manyem. Bin packing and covering with longest items at the bottom: Online version. *The ANZIAM Journal*, 43(E):E186–E232, 2002.

[23] P. Manyem, R. L. Salt, and M. S.Visser. Approximation lower bounds in online LIB bin packing and covering. *Journal of Automata, Languages and Combinatorics*, 8(4):663–674, 2003.

[24] S. S. Seiden. On the online bin packing problem. *J. ACM*, 49:640–671, 2002.

[25] J. Ullman. The Performance of a Memory Allocation Algorithm. *Technical Report 100*, 1971.