Online Colored Bin Packing

Martin Böhm* [1], Jiří Sgall* [1], and Pavel Veselý* [1]

[1]Computer Science Institute, Charles University, Prague, Czech Republic. {bohm,sgall,vesely}@iuuk.mff.cuni.cz.

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 $c \geq 2$ 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 show a matching lower bound of $\lceil 1.5 \cdot OPT \rceil$ for any value of $OPT \geq 2$. In particular, the absolute ratio of our algorithm is 5/3 and this is optimal.

For items of unrestricted sizes we give an asymptotically 3.5-competitive algorithm. When the items have sizes at most $1/d$ for a real $d \geq 2$ the asymptotic competitive ratio 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 prove that all Any Fit algorithms have absolute competitive ratio 3. When the items have sizes at most $1/d$ for a real $d \geq 2$ we show that the Worst Fit algorithm is absolutely $(1 + d/(d-1))$-competitive.

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

1 Introduction

In the *Online 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 sizes 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 so that the colors inside the bins are alternating, i.e., no two items of the same color can be next to each other in the same bin. The goal is to minimize the number of bins used.

Online 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 is that we cannot pack two items of the same color next to each other in one bin.

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.

There are several well-known and often used 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 give 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 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 5-competitive and even optimal for zero-size items. They show input instances on which FF and BF create asymptotically $3 \cdot OPT$ bins. For WF there are sequences of items witnessing that it is at least 3-competitive and $(1 + d/(d-1))$-competitive in the parametric case. Furthermore, NF is not constant competitive.

The idea of the algorithm Pseudo, on which we build as well, is that we first pack the items regardless of their size, i.e., treating their size as zero. This can be done optimally, and the optimum equals the maximal discrepancy in the sequence of colors (to be defined below). Then these bins are partitioned by NF into bins of level at most 1.

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

Very recently and independent of us Dósa and Epstein [9] studied Colored Bin Packing. They improved the lower bound for online Black and White Bin Packing to 2 for deterministic algorithms, which holds for more colors as well. For 3 colors and more they proved an asymptotic lower bound of 1.5 for zero-size items. They designed a 4-competitive algorithm based on Pseudo and a balancing algorithm for zero-size items. They also showed that Any Fit algorithms are not competitive at all (with non-zero sizes).

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. For online algorithms, 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 show a matching lower bound of $\lceil 1.5 \cdot OPT \rceil$ for any value of $OPT \geq 2$. This is significantly stronger than the asymptotic lower bound of 1.5 of Dósa and Epstein [9], in particular it shows that the absolute ratio of our algorithm is 5/3, and this is optimal.

We use this optimal algorithm for zero-size items and the algorithm Pseudo to design an (asymptotically) 3.5-competitive algorithm which is also (asymptotically) $(1.5 + d/(d-1))$-competitive in the parametric case, where the items have sizes at most $1/d$ for a real $d \geq 2$. (Note that for $d < 2$ we have $d/(d-1) > 2$ and the bound for arbitrary items is better.) It is interesting that the algorithm for zero-size items belongs to the Any Fit family, even though such algorithms cannot be competitive for general items as Dósa and Epstein [9] show. Of course, our combined algorithm for general sizes is not Any Fit.

For general sizes we show that algorithms BF, FF and WF are not constant competitive, even for instances with only three colors and very small items, in contrast to their 3-competitiveness for two colors.

For Black and White Bin Packing, we improve the upper bound on the absolute competitive ratio of Any Fit algorithms in the general case to 3 which is tight for BF, FF and WF. 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 also tight. Therefore, WF has the same competitive ratio as the Pseudo algorithm.

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 [22] and by Johnson [16] and it is known to be NP-hard. There is an asymptotic polynomial-time approximation scheme (APTAS) [14] and a fully polynomial-time approximation scheme (FPTAS) [17]. See the survey of Coffman et al. [6] 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 $248/161 \approx 1.540$-competitive [4]. Regarding AF algorithms, NF is 2-competitive and both FF and BF have the absolute competitive ratio exactly 1.7 [11, 12]. 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.

Currently best algorithms (in the worst-case behavior) are based on the *Harmonic(K)* algorithm [19] which assigns items into K size classes $(\frac{1}{2}, 1], (\frac{1}{3}, \frac{1}{2}], \ldots (\frac{1}{K}, \frac{1}{K-1}], (0, \frac{1}{K}]$. Each class of items is packed separately by NF. The asymptotic competitive ratio of Harmonic is 1.691 for K large enough. The ratio was later improved a few times by modifying size classes. The current best algorithm Harmonic++ by Seiden [21] uses 70 items classes to achieve the ratio of approximately 1.589. On the other hand, Harmonic-type algorithms cannot achieve better performance than 1.583.

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.*

Next Fit-K that keeps the last K created bins open is the first studied bounded space algorithm with the asymptotic competitive ratio $1.7 + \frac{3}{10(K-1)}$ for $K \geq 2$ [7, 20]. 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 $K \geq 2$ [8]. Lee and Lee [19] presented Harmonic(K) which is K-bounded-space with the asymptotic ratio of 1.691 for K large enough. Lee and Lee also proved that there is no bounded space algorithm with a better asymptotic 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. [5]. 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 1.7-competitive K-Bounded Best Fit when compared to optimum without reordering; the current lower bound is $3/2$.

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. [18] who also showed that Cardinality Constrained FF has the asymptotic ratio of at most $2.7 - 2.4/k$. Interestingly, the lower bound for the asymptotic competitive ratio for large k is 1.540 [4], i.e., the same as for standard Bin Packing, while the lower bound is 1.428 for $k = 2$ and 1.5 for k from 3 to 5 [15, 1]. For $k \geq 3$, there is an asymptotically 2-competitive online algorithm [1] and better algorithms are known for small k [13]. Regarding the absolute competitive ratio there is a tight bound of 2 for any $k \geq 4$ [10].

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 a 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

Definitions and notation. There are three settings of Colored Bin Packing: In the *offline setting* we are given the items in advance and we can pack them in an arbitrary order. 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. 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.

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 $r' < r$.

For Colored Bin Packing, let C be the set of all colors. 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 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 $c_1, c_2, \dots c_k$ and sizes $s_1, s_2, \dots s_k$ by $n \times \binom{c_1}{s_1}, \binom{c_2}{s_2}, \dots \binom{c_k}{s_k}$.

Lower Bounds on the Restricted Offline Optimum. We use two lower bounds on the number of bins in any packing. The first bound LB_1 is the sum of sizes of all items.

The second bound LB_2 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 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 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

$$LB_2 = \max_{c \in C} \max_{i,j} \sum_{\ell=i}^{j} s_{c,\ell} \,.$$

For Black and White Bin Packing, equivalently $LB_2 = \max_{i,j} |\sum_{\ell=i}^{j} s_\ell|$, 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 LB_2 is a lower bound on the optimum similarly to the proof of Lemma 5 in [2], first observing that the number of bins in the optimum cannot decrease by removing a prefix or a suffix from the sequence of items.

Observation 2.1. *Let $L = L_1 L_2 L_3$ 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 into m bins is a valid packing of L without the first item. Similar holds for the last item. □

Lemma 2.2. *$OPT(L) \geq LB_2$.*

Proof. We prove that for all colors c that the optimum is at least $LB_{2,c} := \max_{i,j} \sum_{\ell=i}^{j} s_{c,\ell}$. Fix a color c and let i, j be $\arg\max_{i,j} \sum_{\ell=i}^{j} s_{c,\ell}$. Let $d = LB_{2,c}$. We may assume that $d > 0$, otherwise d 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 as colors are alternating between c and other colors in the worst case. Since we have d more c-items than non-c-items, we get $k \geq d$. Therefore $OPT \geq LB_{2,c}$ □

Note that if we consider a minimal-length sequence with the maximal discrepancy LB_2, the optimal restricted offline algorithm puts all non-c-items into c-bins whenever it is possible.

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 LB_2 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 LB_2 bins is possible in the restricted offline setting, i.e., items can be packed into LB_2 bins without reordering.*

Proof. Consider a counterexample with a minimal number of items in the sequence. Let $d = LB_2$ 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.

We define an *important interval* as a maximal interval of discrepancy d, more formally a subsequence from the i-th item to the j-th such that for some color c the discrepancy on the interval is d, i.e., $\sum_{\ell=i}^{j} s_{c,\ell} = d$, and we cannot extend the interval in either direction without decreasing its discrepancy. For an important interval, its *dominant color c* is the most frequent color inside it. At first we show that important intervals are just d items of the same color.

Observation 2.4. *Each important interval I contains only d items of its dominant color c 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 higher discrepancy. We delete a and b from the sequence and pack the rest into d bins by minimality.

Consider the situation after packing the item prior to a. There must be a c-bin B, otherwise the subsequence of I from the beginning up to a (including a) has strictly more non-c-items than c-items (each c-item from I is under a non-c-item and a is the extra non-c-item), so the rest of I has discrepancy more than d. By putting a and b into B we pack the whole sequence into d bins, thus it is not a counterexample. □

From the previous observation it follows that important intervals are disjoint. Clearly, there must be an important interval in any nonempty sequence. Let $I_1, I_2, \ldots I_k$ be important intervals in the counterexample sequence and let $J_1, J_2, \ldots J_{k-1}$ 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., $J_0, I_1, J_1, I_2, J_2, \ldots\ J_{k-1}, I_k, J_k$ 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 $I_2, J_2, I_3, \ldots I_k, J_k$ 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 nonempty: 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 nonempty: 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 nonempty 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 nonempty in the counterexample. If J_0 is nonempty, we merge packings of J_0, I_1 and I_1, J_1, I_2, J_2, and if J_2 is nonempty, 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 nonempty. 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 the important intervals are maximal, 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 and at least the first two items in J_1 have colors different from c.

We delete the first c-item p from I_1, the first non-c-item q from J_1 and the last c-item r from I_2. Suppose for a contradiction that there is an interval I of discrepancy d in the rest of the sequence. As I (possibly with q) has lower discrepancy in the original sequence it must intersect I_1 and J_1, hence its dominant color is c. If I intersects also I_2, we add the items p, q and r into I (and possibly some other items from I_1 or I_2) to obtain an interval of discrepancy at least $d + 1$ in the original sequence which is a contradiction. Otherwise I intersects only I_1 and J_1, but any prefix of the rest of J_1 still contains at least as many non-c-items as c-items, so $I \setminus J_1$ has discrepancy at least d. But $I \setminus J_1$ is contained in the rest of I_1 that have only $d - 1$ items and we get a contradiction. Therefore the maximal discrepancy decreases after deleting the three items, so we can pack the rest into $d - 1$ bins and the items p, q and r are put into a separate bin. Note that important intervals of discrepancy $d - 1$ may change after deleting the three items.

In all cases we can pack the sequence into d bins, therefore no such counterexample exists. □

3 Algorithms for Zero-size Items

3.1 Lower Bound on Competitiveness of Any Online Algorithm

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, while the optimal number of bins is n.*

Proof. We show that if an algorithm uses less than $\lceil 1.5n \rceil$ bins, we 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 maximal discrepancy stays n. Applying Theorem 2.3 we know that $OPT = n$, but the algorithm is forced to open $\lceil 1.5n \rceil$ bins using finitely many items as the number of black bins is increasing. Moreover, we use only three colors throughout the whole proof, denoted by black, white and red and abbreviated by b, w and r in formulas.

We introduce the current discrepancy of a color c which basically tells us how many c-items has come recently and thus how many c-items may arrive without increasing the overall discrepancy. Formally, we define the current discrepancy after packing the k-th item as $CD_{c,k} = \max_{i \leq k+1} \sum_{\ell=i}^{k} s_{c,\ell}$, i.e., the discrepancy on an interval which ends with the last packed item (the k-th). Note that $CD_{c,k}$ is at least zero as we can set $i = k+1$. We omit the k index in $CD_{c,k}$ when it is obvious from the context.

Initially we send n black items, then we continue by phases and end the process whenever the algorithm uses $\lceil 1.5n \rceil$ bins. When a phase starts, there are less than $\lceil 1.5n \rceil$ black bins and possibly some other white or red bins. We also guarantee $CD_{\mathrm{w}} = 0$, $CD_{\mathrm{r}} = 0$, and $CD_{\mathrm{b}} \leq n$. Let N_{b} be the number of black bins when a phase starts. In each phase we force the algorithm to use $\lceil 1.5n \rceil$ bins or to have more than N_{b} black bins, while $CD_{\mathrm{w}} = 0$, $CD_{\mathrm{r}} = 0$, and $CD_{\mathrm{b}} \leq n$ at the end of each phase.

We now present how a phase works. Let new items be items from the current phase and old items be items from previous phases. We begin the phase by sending n new items of colors alternating between white and red, starting by white, so we send $\lceil n/2 \rceil$ white items and $\lfloor n/2 \rfloor$ red items. After these new items, the current discrepancy is one either for red if n is even, or for white if n is odd, and it is zero for the other colors.

If some new item is not put on an old black item, we send n black items. Since the new items are on the top of less than n bins that were black item at the start of the phase, the number of black bins increases. Moreover, $CD_{\mathrm{w}} = 0$, $CD_{\mathrm{r}} = 0$, and $CD_{\mathrm{b}} = n$, hence we finish the phase and continue with the next phase if we have less than $\lceil 1.5n \rceil$ bins.

Otherwise all new red and white items are put on old black items. If n is even, $CD_{\mathrm{w}} = 0$ and we send additional n white items. After that we have at least $1.5n$ white bins, so we reach our goal.

If n is odd, $CD_{\mathrm{w}} = 1$ and we send a black item p. If p does not go on a new white item, we send n white items forcing $\lceil n/2 \rceil + n$ white bins and we are done. Otherwise the black item p is put on a new white item. White and red have $\lfloor n/2 \rfloor$ new items on the top of bins, $CD_{\mathrm{w}} = 0$, and $CD_{\mathrm{r}} = 0$. We send another black item q. Since red and white are equivalent colors (considering only new items), w.l.o.g. q goes into a red bin or into newly opened bin.

Next we send a white item r and a red item s. After packing r 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. Moreover, after packing the red item s we have $CD_{\mathrm{b}} = 0$ and $CD_{\mathrm{w}} = 0$. So if s 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), we send n white items and the algorithm must use $\lceil 1.5n \rceil$ bins. Otherwise s is packed on a new white item and we send n black items. We increase the number of black bins, because we sent $n+2$ new black items and at most $n+1$ new non-black items were put into a black bin (at most n items at the beginning of the phase plus the item r). Since $CD_{\mathrm{w}} = 0$, $CD_{\mathrm{r}} = 0$, and $CD_{\mathrm{b}} = n$, we continue with the next phase. □

3.2 Optimal Algorithm for Zero-size Items

The overall problem of FF, BF and WF is that they pack items regardless of the colors of bins. We address the problem by balancing the colors of top items in bins – we mostly put an incoming c-item into a bin of color $c' \neq c$ such that there is the maximal number of c'-bins. When we have more choices of bins where to put an item we use 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 $\lceil 1.5LB_2 \rceil$ 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 any size and prove that the resulting algorithm is (asymptotically) 3.5-competitive.

After packing the k-th item from the sequence, let d_k be the maximal discrepancy so far, i.e., the discrepancy on an interval before the $(k+1)$-st item, and let $N_{c,k}$ be the number of c-bins after packing the k-th item. As in the proof of Theorem 3.1, 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 on an interval which ends with the last packed item (the k-th). Note 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. We omit the k index in d_k, $N_{c,k}$, $CD_{c,k}$ and $\alpha_{c,k}$ when it is obvious from the context.

During processing the items, if we have m open bins, $d \leq m$ is the maximal discrepancy so far, and for some color c it holds that $d - CD_c > m - N_c$, we can send $d - CD_c$ of c-items without increasing the discrepancy and force the algorithm to open a new bin.

Hence, to end with at most $\lceil 1.5d \rceil$ bins we try to keep $\alpha_c = N_c - \lceil d/2 \rceil \leq CD_c$ for all colors c. If we can keep that and there is a color c with $N_c > \lceil 1.5d \rceil$, 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

$$N_c - \left\lceil \frac{d}{2} \right\rceil \leq CD_c. \tag{1}$$

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 all non-c-items into c-bins, therefore both the number of c-bins N_c and the current discrepancy CD_c decrease with an incoming non-c-item and both increase with a c-item, so 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 $N_c > \lceil d/2 \rceil$. In the following let these two colors be black and white w.l.o.g., we abbreviate them b and w in formulas. We state the algorithm Balancing Any Fit for items of size zero.

Balancing Any Fit (BAF):

1. When there is at most one color with the number of bins strictly more than $\lceil d/2 \rceil$, we 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 we can choose color c' arbitrarily, e.g., by First Fit. Between c'-bins we can choose again by FF or arbitrarily.
2. If $N_{\mathrm{b}} > \lceil d/2 \rceil$ and $N_{\mathrm{w}} > \lceil d/2 \rceil$, we put black items into white bins and white items into black bins. We pack items of other colors into a white bin if $N_{\mathrm{b}} - \lceil d/2 \rceil < CD_{\mathrm{b}}$, otherwise into a black bin.

In the second case 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 $\alpha_{\mathrm{w}} = CD_{\mathrm{w}}$, we can force the algorithm to open more than $\lceil 1.5d \rceil$ bins. Therefore we need to prove that in the second 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.

Theorem 3.2. *Balancing Any Fit algorithm 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 enough for the algorithm to create at most $\lceil 1.5d \rceil$ bins and thus to be 1.5-competitive, since the maximal discrepancy equals the optimum. As we discussed, keeping these inequalities can be done by the first case of the algorithm, if there is at most one color c with $N_c > \lceil d/2 \rceil$, thus most of the proof deals with two colors having more than $\lceil d/2 \rceil$ bins.

Claim 3.3. *Given that* $N_c - \lceil d/2 \rceil \leq CD_c$ *for all colors* c *during the whole process, the algorithm opens at most* $\lceil 1.5d \rceil$ *bins.*

Proof. We prove the claim by contradiction: Suppose that the main invariant (1) holds for all colors all the time and BAF opens a bin with a c-item which is the t-th item in the sequence and we exceed the $\lceil 1.5d_t \rceil$ limit, but before the t-th item there were at most $\lceil 1.5d_{t-1} \rceil$ bins. Thus $d_t = d_{t-1}$, since if $d_t = d_{t-1} + 1$, then the bound also increases with the t-th item.

Let the k-th item be the last non-c-item before the t-th, so only c-items come after the k-th item. None of c-items from the $(k + 1)$-st to the t-th increase the maximal discrepancy d, otherwise if one such item increases d, then all following such items also do. Thus $d_k = d_t$.

The algorithm must receive $\lceil 1.5d_k \rceil + 1 - N_{c,k}$ of c-items after the k-th item, but then

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

We get a contradiction, since it holds $CD_{c,t} \leq d_t = d_k$. □

We now focus on the case in which $N_{\mathrm{b}} > \lceil d/2 \rceil$ and $N_{\mathrm{w}} > \lceil d/2 \rceil$. Let the **secondary invariant** be

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

We show that we can maintain the secondary invariant, while black and white are the two strictly most common colors of bins. Then we prove that the secondary invariant holds when black and white become the two strictly most common colors which must precede the time when the number of bins for the second color gets over the $\lceil d/2 \rceil$ limit.

Observe that keeping the secondary invariant (2) is enough to have $\alpha_{\mathrm{b}} < CD_{\mathrm{b}}$ or $\alpha_{\mathrm{w}} < CD_{\mathrm{w}}$ if both $\alpha_{\mathrm{b}} > 0$ and $\alpha_{\mathrm{w}} > 0$. If $\alpha_{\mathrm{b}} \geq CD_{\mathrm{b}}$ and $\alpha_{\mathrm{w}} \geq CD_{\mathrm{w}}$, the secondary invariant becomes $2\alpha_{\mathrm{b}} + 2\alpha_{\mathrm{w}} \leq CD_{\mathrm{b}} + CD_{\mathrm{w}} + 1 \leq \alpha_{\mathrm{b}} + \alpha_{\mathrm{w}} + 1$ which is a contradiction. Note that we used that α_{w} and α_{b} are integral. Therefore we can choose either black bin, or white bin for an item of another color and keep the main invariant (1) for black and white colors in the second case of the algorithm.

Claim 3.4. *Suppose that the main invariant (1) holds for all colors and the secondary invariant (2) also holds when black and white became the two strictly most common colors of bins, i.e., $N_c < N_{\mathrm{b}}$ and $N_c < N_{\mathrm{w}}$ for all other colors c. Then these invariant inequalities hold as long as black and white are still the two strictly most common colors of bins.*

Proof. We show that an incoming item does not violate the secondary invariant, therefore we also keep the main invariant for all colors by the definition of the algorithm.

First we suppose that the maximal discrepancy d is not changed by the incoming item. There are three cases according to the color of the item:

- The item is black: Then α_b increases and α_w decreases, because black items are put into white bins. Also the right-hand side of the inequality 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.)
- When the item is white, the situation is symmetric to the previous case.
- Otherwise we pack the item into a white bin if $N_\mathrm{b} - \lceil d/2 \rceil < CD_\mathrm{b}$, otherwise into a black bin. If it is packed into a white bin, α_w decreases and α_b stays the same, while both CD_b and CD_w decrease by at most one, so the secondary invariant holds as the left-hand side decreases by two and the right-hand side decreases by at most two. The main invariant also holds for both colors, since for black the inequality holds strictly. If the item is packed into a black bin, we have $N_\mathrm{w} - \lceil d/2 \rceil < CD_\mathrm{w}$, because the secondary invariant held before packing the item, thus either for black, or white the main invariant inequality held strictly. The secondary invariant holds too as its left-hand side decreases by two and the right-hand side decreases by at most two.

Otherwise d increases with an incoming c-item, thus also CD_c increases and $\alpha_{c'}$ 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 the secondary invariants. □

Note that in the previous proof, α_b or α_w can be negative in the secondary invariant. We complete the analysis of BAF by showing that the secondary invariant starts to hold when two colors become the two strictly most common colors of bins.

Claim 3.5. *When it starts to hold that $N_c < N_\mathrm{b}$ and $N_c < N_\mathrm{w}$ for all other colors c, then $2\alpha_\mathrm{b} + 2\alpha_\mathrm{w} \leq CD_\mathrm{b} + CD_\mathrm{w} + 1$.*

Proof. Let the k-th item be the one after which black and white became the two strictly most common colors and assume w.l.o.g. also $N_{\mathrm{b},k} \geq N_{\mathrm{w},k}$. Note that the k-th item can be only black or white. Before the k-th item the number of non-black bins is at most $\lceil 1.5 d_{k-1} \rceil - N_{\mathrm{b},k-1} = d_{k-1} - \alpha_{\mathrm{b},k-1}$.

If the k-th item is white, then after packing it the number of non-black bins is at most $d_{k-1} - \alpha_{\mathrm{b},k-1} + 1 \leq d_k - \alpha_{\mathrm{b},k}$ (note that there is an inequality only because of a possible increase of d). As white was not the second strictly most common color of bins before the k-th item we have

$$\alpha_{\mathrm{w},k} = N_{\mathrm{w},k} - \left\lceil \frac{d_k}{2} \right\rceil = N_{\mathrm{w},k-1} + 1 - \left\lceil \frac{d_k}{2} \right\rceil \le \frac{d_{k-1} - \alpha_{\mathrm{b},k-1}}{2} + 1 - \left\lceil \frac{d_k}{2} \right\rceil$$
$$\le \frac{d_k - \alpha_{\mathrm{b},k} - 1}{2} + 1 - \left\lceil \frac{d_k}{2} \right\rceil = -\frac{\alpha_{\mathrm{b},k}}{2} + 0.5 + \frac{d_k}{2} - \left\lceil \frac{d_k}{2} \right\rceil \le -\frac{\alpha_{\mathrm{b},k}}{2} + 0.5.$$

Otherwise the k-th item is black, so after it the number of non-black bins is at most $d_k - \alpha_{\mathrm{b},k} \ge d_{k-1} - \alpha_{\mathrm{b},k-1} - 1$. Since the number of white bins does not change, otherwise white would not become the second strictly most common color of bins, we get

$$\alpha_{\mathrm{w},k} = N_{\mathrm{w},k} - \left\lceil \frac{d_k}{2} \right\rceil = N_{\mathrm{w},k-1} - \left\lceil \frac{d_k}{2} \right\rceil \le \frac{d_{k-1} - \alpha_{\mathrm{b},k-1}}{2} - \left\lceil \frac{d_k}{2} \right\rceil$$
$$\le \frac{d_k - \alpha_{\mathrm{b},k} + 1}{2} - \left\lceil \frac{d_k}{2} \right\rceil = -\frac{\alpha_{\mathrm{b},k}}{2} + 0.5 + \frac{d_k}{2} - \left\lceil \frac{d_k}{2} \right\rceil \le -\frac{\alpha_{\mathrm{b},k}}{2} + 0.5.$$

In both cases of the color of the k-th item we get $\alpha_{\mathrm{w},k} \le -\alpha_{\mathrm{b},k}/2 + 0.5$. 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 which BAF keeps, since it uses the first case of the algorithm for packing the k-th item, because black can be the only color with more than $\lceil d_{k-1}/2 \rceil$ bins. Hence the secondary invariant (2) holds. □

Therefore we can keep the main invariant $N_c - \lceil d/2 \rceil \le CD_c$ for all colors c during the whole run of the algorithm and the theorem follows by Claim 3.3. □

4 Algorithms for Items of Any Size

4.1 Constant 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 4.1. *The algorithm Pseudo-BAF for Colored Bin Packing is asymptotically* 3.5*-competitive. Precisely, it uses at most* $\lceil 3.5 \cdot OPT \rceil$ *bins. 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 at least one, since no two consecutive items of the same color are in a pseudo bin. In each pseudo bin we pair each bin with an odd index with the following bin with an even index, therefore we pair all bins except at most one in each pseudo bin. Moreover, the total size of a pair of bins is at least one. Therefore the number of paired bins is at most $2 \cdot LB_1 \leq 2 \cdot OPT$. The number of unpaired bins is at most the number of pseudo bins created by the algorithm BAF which uses at most $\lceil 1.5 \cdot LB_2 \rceil \leq \lceil 1.5 \cdot OPT \rceil$ bins as the maximal discrepancy LB_2 is also a lower bound on the optimum. Overall the algorithm Pseudo-BAF creates at most $\lceil 3.5 \cdot OPT \rceil$ bins.

For the parametric case, inside each pseudo bin all real bins except the last one have level at least $(d-1)/d$, so their number is at most $d/(d-1) \cdot OPT$. 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 $\varepsilon = \frac{1}{2n}$. We send $n-1$ groups of three items, specifically $(n-1) \times \left(\substack{\text{white} \\ \varepsilon}, \substack{\text{black} \\ 1}, \substack{\text{black} \\ \varepsilon} \right)$.

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 we have $3 \cdot (n-1)$ bins. The optimum for $n-1$ groups is n, because we can pack all tiny items together in one bin and $LB_1 = n$.

Then we send 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 $OPT = n$.

For the parametric case, 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. Moreover, we can continue with the hard instance with zero-size items like in the general case 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. $\square$

4.2 Classical Any Fit Algorithms

We analyze classical Any Fit algorithms, namely 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.

Proposition 4.2. *First Fit and Best Fit are not constant competitive.*

Proof. We send an instance of $4n$ items which can be packed into two bins, but FF and BF create $n+1$ bins where n is an arbitrary integer.

Let $\varepsilon = \frac{1}{4n}$. The instance is $n \times \binom{\text{black}}{\varepsilon}, \binom{\text{black}}{\varepsilon}, \binom{\text{white}}{\varepsilon}, \binom{\text{red}}{\varepsilon})$. 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 in the first bin. The first black item, the white item and red items 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.

Hence FF and BF create $\frac{n+1}{2}OPT$ bins. $\square$

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 4.3. *Worst Fit is not constant competitive.*

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

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 $\frac{3n}{6n^2+1}$, which is less than $\frac{1}{2n}$ as $\frac{\varepsilon}{\delta} > 3n$. Therefore all second black items are packed into separate bins and WF creates $n+1$ bins, while the optimum is two. $\square$

5 Black and White Bin Packing

We now focus on Colored Bin Packing with two colors,i.e., Black and White Bin Packing, studied previously by Balogh et al. [3, 2]. 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 of sizes at most $1/d$ (for $d \geq 2$) as it is $(1 + d/(d-1))$-competitive in this case. Note that for infinitesimally small items WF is 2-competitive, while BF and FF are 3-competitive.

Both bounds are tight, since there are instances for FF and BF on which the competitive ratio is asymptotically 3 and an instance for WF in the parametric case on which WF uses asymptotically $(1+d/(d-1))\,OPT$ bins [2].

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, 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.

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 some small bins are separated. Moreover, our chains can have at most two bins, so the average level of bins in each chain is clearly 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 LB_2 which yields the 3-competitiveness of AF. Therefore we want to have the number of separated bins as small as possible.

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 last bin in the chain has another color than the bottom item of the added bin.

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 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 in which the last bin has an item of another color on the top, i.e., black for a white incoming item and white for a black incoming item, we add the newly created bin into the chain. (Note that the last bin in the chain must be big.) Otherwise 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 indexes 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 w.l.o.g. 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. 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 $k \leq t - |W|$).

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 for separated bins with a black item at the bottom the number of black items is greater by one than the number of white items and separated bins created between the k-th and the t-th 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 black item than white items with index i such that $k < i \leq t$, since the first and the last such items are black.

Now we look at items with index i such that $k < i < t$ 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 claim.

Claim 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. When the t-th item comes and creates a new separated bin, the last item in each chain must be black. Therefore the claim holds for the chains with only one bin.

For the chains with two bins (the first big and the second small) we observe that a bin created with a link has either a black item, or a big white item at the bottom. If it would have a small white link at the bottom, there cannot be a small black bin and $b_i = 0$ for $k < i < t$ which is a contradiction with the definition of k. Since a big white item starts a new chain, the second bin in a chain cannot have a white link at the bottom.

Moreover, the first link in the second bin of a chain must be black, because either the second bin was created after the k-th item and we use the observation from the previous paragraph, or it was created before the k-th item and then it must had a white item on the top when the k-th item came, since there was no small bin with a black item on the top.

So it cannot happen in a chain with two bins that there are two white links next to each other, or separated by some items that are not links. Note that for black links this situation can happen. Since there must be a link in the second bin and the last such link is black, the claim holds for chains with one big and one small bin.

The process of assigning bins into chains does not allow chains with more than two bins or with two big bins. Hence in each chain the number of black links is at least the number of white links. $\square$

Let s be the number of separated bins. We found out that when we focus on items with 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 s can only decrease, since no separated bin is created. So we have bounded the value of s at the end from above by the color discrepancy between the $(k+1)$-st and the t-th item (s_i is 1 when the i-th item is white and -1 otherwise):

$$s \leq \left| \sum_{\ell=k+1}^{t} s_\ell \right| \leq LB_2$$

Note that some items after the t-th item can create a bin, but such bins are put into chains. □

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 sizes 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. Let OPT be the number of bins used in an optimal packing. 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 $|B| \leq d/(d-1) \cdot OPT$. We show that $|S|$ is bounded by the maximal color discrepancy LB_2 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 $(d-1)/d$. Let b_i be the number of small black bins after adding the i-th item from the sequence. Similarly let w_i be the number of white bins with level less than $(d-1)/d$ after adding the i-th item.

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 w.l.o.g. that the last created bin has a black item at the bottom. Let t be the index of a black item that created the last bin. It holds that $w_t = 0$, since otherwise the t-th item would go into a small white bin.

Let k be the last index smaller than t for which $b_k = 0$ (if $b_i > 0$ for all $i \geq 1$, 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 goes 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 $k \leq t - |W|$).

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 black item than white items with index i such that $k < i \leq t$, since the first such item is black.

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

Note that this pairing 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, but not contained in W.

We found out that when we focus on items with index 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_i is 1 when the i-th item is white and -1 otherwise):

$$|S| \leq \left| \sum_{\ell=k+1}^{t} s_\ell \right| \leq LB_2$$

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 general items, our online algorithm still leaves a gap between the lower bound 2 [9] and our upper bound of 3.5 and a corresponding gap in the parametric setting. 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.

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*, pages 1–19, 2014.

[3] J. Balogh, J. Békési, G. Dósa, H. Kellerer, and Z. Tuza. Black and white bin packing. In *Approximation and Online Algorithms*, volume 7846 of *Lecture Notes in Computer Science*, 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] M. Chrobak, J. Sgall, and G. J. Woeginger. Two-bounded-space bin packing revisited. In C. Demetrescu and M. Halldórsson, editors, *Algorithms - ESA 2011*, volume 6942 of *Lecture Notes in Computer Science*, pages 263–274. Springer Berlin Heidelberg, 2011.

[6] 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 New York, 2013.

[7] J. Csirik and B. Imreh. On the worst-case performance of the nkf bin-packing heuristic. *Acta Cybern.*, 9:89–105, Dec. 1989.

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

[9] G. Dósa and L. Epstein. Colorful bin packing. *To appear in proceedings of 14th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2014)*, 2014. Also ArXiv 1404.3990.

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

[11] G. Dósa and J. Sgall. First Fit bin packing: A tight analysis. In N. Portier and T. Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 538–549, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[12] G. Dósa and J. Sgall. Optimal analysis of Best Fit bin packing. *To appear in proceedings of 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014)*, 2014.

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

[14] W. Fernandez de la Vega and G. Lueker. Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[15] H. Fujiwara and K. Kobayashi. Improved lower bounds for the online bin packing problem with cardinality constraints. *Journal of Combinatorial Optimization*, pages 1–21, 2013.

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

[17] N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Foundations of Computer Science, 1982. SFCS '08. 23rd Annual Symposium on*, pages 312–320, Nov 1982.

[18] 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, Oct. 1975.

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

[20] W. Mao. Tight worst-case performance bounds for next-k-fit bin packing. *SIAM J. Comput.*, 22:46–56, Feb. 1993.

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

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