Class constrained bin covering*

Leah Epstein[†] Csanád Imreh[‡] Asaf Levin[§]

Abstract

We study the following variant of the bin covering problem. We are given a set of unit sized items, where each item has a color associated with it. We are given an integer parameter $k \geq 1$ and an integer bin size $B \geq k$. The goal is to assign the items (or a subset of the items) into a maximum number of subsets of at least B items each, such that in each such subset the total number of distinct colors of items is *at least* k. We study both the offline and the online variants of this problem. We first design an optimal polynomial time algorithm for the offline problem. For the online problem we give a lower bound of $1+H_{k-1}$ (where H_{k-1} denotes the $(k-1)$-th harmonic number), and an $O(k)$-competitive algorithm. Finally, we analyze the performance of the natural heuristic First fit.

1 Introduction

Consider the following imaginary application which illustrates the problem studied in this paper. An internet provider has access to a fixed number of communication satellites, each having its own capacity. Each client is interested in getting a total bandwidth of B units. The provider needs to allocate each served client at least this requested bandwidth. In order to make the usage of communication network more reliable, in case of failures in some satellites, the bandwidth supplied to each customer needs to be coming from at least k different sources. The provider thus allocates some units of bandwidth from each satellite, so that a failure in up to $k-1$ satellites would still keep each served client connected, that is, the number of satellites with non-zero bandwidth given to each client has to be at least k. The goal is to maximize the number of served clients, without exceeding the capacity of the satellites.

In the class constrained bin covering problem of unit sized items (ccbc) we are given a set of unit sized items $I = \{1, 2, \ldots, n\}$ where each item has a color associated with it. The color of item i is denoted by c_i (i.e., if i and j have the same color then $c_i = c_j$). The set of items of one color is called a color class. We are also given (non-negative) integer parameters k and $B \geq k$. A feasible solution of profit m is a partition of I into subsets $S_1, \ldots, S_{m+t}$ such that for each $i = 1, 2, \ldots, m$ the following two conditions hold: $|S_i| \geq B$ (i.e., the items of S_i have a total size of at least B) and S_i has items from at least k color classes (i.e., $|\bigcup_{j\in S_i} c_j| \geq k$). Note that we do not require these properties from $S_{m+1}, \ldots, S_{m+t}$ and

*This research has been partially supported by the Hungarian National Foundation for Scientific Research, Grant F048587.

[†]Department of Mathematics, University of Haifa, 31905 Haifa, Israel. lea@math.haifa.ac.il

[‡]Department of Informatics, University of Szeged, 6720 Szeged, Hungary. cimreh@inf.u-szeged.hu

[§]Department of Statistics, The Hebrew University, 91905 Jerusalem, Israel. levinas@mscc.huji.ac.il

these sets do not count toward the goal function. We say that a bin is *covered* if both it has at least B items, and items of at least k color classes. Note that we assume throughout this paper that $B \geq k$ since otherwise we let $B = k$ without changing the problem.

The goal of CCBC is to find a feasible solution that maximizes the number of valid subsets m in the partition. We denote by q the total number of colors in the instance. We note that if $q = n$, or if $B = k = 1$, the resulting problem is equivalent to the bin covering problem of identical sized items, which can be easily solved by packing B items to each bin (except for at most one invalid bin). Therefore we assume $B \geq 2$ and $q < n$.

In our possible application, the satellites correspond to colors, the clients to bins, the served clients to covered bins, and the capacity of a satellite is the number of unit sized items of its color. Hence the scenario we describe is equivalent to CCBC.

For an algorithm $\mathcal{A}$, we denote its profit by $\mathcal{A}$ as well. The profit of an optimal offline algorithm (that in the case of comparison to online algorithms, knows the complete sequence of items) is denoted by OPT. For maximization problems, the (asymptotic) competitive ratio of an algorithm $\mathcal{A}$ is the infimum $\mathcal{R} \geq 1$ such that for any input, $\text{OPT} \leq \mathcal{R} \cdot \mathcal{A} + c$, where c is a constant independent of the input.

Related work. Packing problems attract much attention for many years [12, 6, 10, 9, 7]. Shachnai and Tamir [18, 19] suggested a model of packing with unit sized colorful items into (identical) integer sized bins. Each color represents a class of items, and each bin of size B has a given number of compartments, $k \leq B$, and thus can accommodate items only of this given number of colors. They call this problem *class-constrained bin packing*. They studied the offline problem in [18] and designed a dual polynomial time approximation scheme for constant values of q (which as in CCBC, denotes the total number of distinct colors in the input), that is, an algorithm that given $\varepsilon > 0$ finds in polynomial time, a packing into OPT bins of size $(1+\varepsilon)B$, where OPT is the minimum number of bins of size B required to pack all items. The problem is NP-hard for $k = 2$ [17, 14] and NP-hard in the strong sense for $k \geq 3$ [14] (these papers study a throughput version of the class-constrained bin packing). The online variant is studied in [19], where two algorithms of competitive ratio 2 were introduced, and a matching lower bound (for some types of instances) was given. The generalization of the packing problem with items of arbitrary size was studied in [20]. A different model of packing colorful unit sized items was studied in [16].

The standard bin covering problem of unit sized items (CCBC where $k = 1$) can be solved trivially in both the offline and the online scenarios. The more general bin covering problem, with arbitrary sizes of items, was first investigated in [1, 2]. Assman et al. [2] proved that the greedy algorithm (that simply keeps putting items into the same bin until it is covered and then moves on to the next bin) has a competitive ratio of 2. In the same paper, offline algorithms with approximation ratios $\frac{3}{2}$ and $\frac{4}{3}$, respectively, were presented. Csirik and Totik [11] proved that in fact the greedy algorithm is a *best possible* on-line algorithm, since no on-line algorithm can have a worst case competitive ratio that is strictly smaller than 2.

The offline, standard bin covering problem was later studied by Csirik, Johnson and Kenyon [8], who designed an APTAS. Jansen and Solis-Oba [15] showed an AFPTAS for this problem. This settled the complexity of the problem, since it is clearly NP-hard.

We finally mention variants of packing and covering, that are related to class constrained problems, which are *cardinality constrained bin packing* (see e.g. [5, 3]) and *cardinality constrained bin covering* [13]. In these problems, a parameter p is given, where each bin must contain at most p items, in packing problems, and at least p items, in covering problems. These problems are the special case of class constrained problems where any two items have distinct colors, and $p = k$.

Paper outline. In Section 2, we consider the offline version of CCBC and we show that it is polynomially solvable. Note the difference with its dual, i.e., the packing problem, which is NP-hard in the strong sense for $k \geq 3$ and NP-hard for $k = 2$ [17, 14]. In Section 3 we turn our attention to the online problem where the items arrive one by one revealing their color and are to be packed by the algorithm before any future items arrive. For this problem we design a competitive algorithm, and provide a lower bound on the competitive ratio of any (deterministic or randomized) online algorithm. More precisely, we start by showing that the FIRST FIT algorithm is exactly B-competitive for all fixed values of $k \geq 2$. Then, we show that any (deterministic or randomized) online algorithm has a non-constant competitive ratio of at least $1 + H_{k-1}$ where for every integer m, $H_m = 1 + \frac{1}{2} + \ldots + \frac{1}{m}$ denotes the m-th harmonic number. For fixed values of B we show that the lower bound remains $\Omega(\log k)$. Finally, we present the main result of this paper, that is an $O(k)$-competitive (deterministic) algorithm for the online problem. We conclude this paper in Section 4 by discussing some directions for future research.

2 The offline problem

In this section we discuss how to solve the offline problem in polynomial time. We note that the encoding of the offline problem is as follows: we are given a set C of q colors, for each of these colors we have the number of items of this color. For color $c \in C$ we denote the number of items of color c by n_c. We require that our polynomial time algorithm is polynomial in the input length of this (compact) representation. That is, polynomial in q and $\sum\limits_{c \in C} \log n_c \geq \log \sum\limits_{c \in C} n_c = \log n$.

We first show an algorithm which guesses the value of OPT via binary search as explained later. Let $\mathbf{o}$ be an integer number such that $0 \leq \mathbf{o} \leq n$. Denote by $C_1 = \{c \in C : n_c \geq \mathbf{o}\}$ and by $C_2 = C \setminus C_1$. Denote by S_2 the total number of items of color in C_2.

Lemma 1 *Assume that $\mathbf{o} \leq$ OPT. Then the following two conditions hold:*

1. $B \cdot \mathbf{o} \leq n = \sum\limits_{c \in C} n_c$.
2. $(k - |C_1|) \cdot \mathbf{o} \leq S_2 = \sum\limits_{c \in C_2} n_c$.

Proof Condition 1 holds due to the following. Each covered bin needs at least B items, so in total there need to be at least $B \cdot \text{OPT}$ items, and thus $n \geq B \cdot \text{OPT} \geq B \cdot \mathbf{o}$. To see that Condition 2 holds, note that each covered bin must be allocated items of at least $k - |C_1|$ color classes from the items of colors in C_2 (because each such bin may have up to $|C_1|$ colors from C_1). Since there are OPT such bins, we have $S_2 \geq \text{OPT} \cdot (k - |C_1|) \geq \mathbf{o} \cdot (k - |C_1|)$, and the claim follows. □

Claim 1 *Assume that $0 \leq \mathbf{o} \leq n$. If $B \cdot \mathbf{o} \leq n$ holds, then we have $\lfloor \frac{n}{\mathbf{o}} \rfloor \geq B$ and $\lfloor \frac{n}{B} \rfloor \geq \mathbf{o}$. If $(k - |C_1|) \cdot \mathbf{o} \leq S_2$, then we have $\lfloor \frac{S_2}{\mathbf{o}} \rfloor \geq k - |C_1|$ and $\lfloor \frac{S_2}{k-|C_1|} \rfloor \geq \mathbf{o}$.*

Proof Follows from the fact that B, $k - |C_1|$ and $\mathbf{o}$ are integers. □

We next show that the combination of the two conditions of Lemma 1 is not only necessary but in fact sufficient to ensure that a solution of value $\mathbf{o}$ exists, and can be found efficiently.

Lemma 2 *Given a value $0 \leq \mathbf{o} \leq n$ with the resulting partition into C_1, C_2, such that $B \cdot \mathbf{o} \leq n$ and $(k - |C_1|) \cdot \mathbf{o} \leq S_2$, then there exists a feasible solution of profit $\mathbf{o}$. Moreover, such a solution can be created using $O(q)$ operations.*

Proof We show how to construct a solution that covers $\mathbf{o}$ bins. We first place the items of C_2 in the $\mathbf{o}$ bins such that for each color $c \in C_2$ and each bin, the bin has at most one item of color c, and also each bin has at least $\lfloor \frac{S_2}{\mathbf{o}} \rfloor$ items. Note that such allocation of items to bins can be computed in a round-robin fashion (where the items of each color are allocated consecutively).

We note that for a fixed value of $\mathbf{o}$, using $O(q)$ operations, we can find an encoding of such an allocation to bins, where this encoding indicates for each color the (cyclic) interval of bins that contains items of this color. By the second part of Claim 1, we conclude that each resulting bin has items of at least $k - |C_1|$ colors.

Then, we continue to allocate the items of colors in C_1 in a round-robin fashion, specifically, we continue the process from the bin that was supposed to get the next item at the time in which the previous allocation was stopped. Once again, the items of each color are allocated to a consecutive set of bins. We note that since each color c of C_1 has at least $\mathbf{o}$ items, then each bin will have at least one item of color c, and therefore each bin will have items of at least k colors. Moreover, by the first part of Claim 1, at the end of the allocation of the items, each bin will have at least B items, and therefore this allocation creates a solution that covers $\mathbf{o}$ bins. To conclude that we found a polynomial time algorithm for constructing the solution, we note that (similarly to the previous case) the allocation of the items of a color $c \in C_1$ can also be identified by the cyclical consecutive interval of bins that contain $\lfloor \frac{n_c}{\mathbf{o}} \rfloor + 1$ items of color c, and the number of items of color c in the other bins (which must be $\lfloor \frac{n_c}{\mathbf{o}} \rfloor$). Hence, we can compute the allocation of the items into bins in $O(q)$ operations. □

We next show how to find the maximum value of $\mathbf{o}$, that satisfies both conditions that are shown to be necessary and sufficient due to Lemmas 1 and 2.

We create an array $\mathcal{Q}$ of size q to store the values n_c, and sort the numbers in a non-decreasing order of the values. This step can be implemented using $O(q \log q)$ operations. In one scan of the array, we create an array $\mathcal{Q}'$ of length $q + 1$, with the prefix sums of $\mathcal{Q}$. This step can be implemented using $O(q)$ operations. Each prefix sum corresponds to a fixed partition into C_1 and C_2 such that if $c_1 \in C_1(P)$ and $c_2 \in C_2(P)$, then $n_{c_2} < n_{c_1}$.

For a given partition, $P = (C_1(P) : C_2(P))$, we can find the maximum number of bins that can be covered, OPT_P as follows. By Lemma 1, we have $\text{OPT}_P \leq \lfloor \frac{n}{B} \rfloor$ and $\text{OPT}_P \leq \left\lfloor \frac{S_2(P)}{k-|C_1|)} \right\rfloor$, where $S_2(P)$ denotes the total number of items in color classes of $C_2(P)$. On the other hand, by Lemma 2, any value $\mathbf{o}$ that satisfies both conditions admits a feasible solution of this value. Therefore we have $\text{OPT}_P = \min \left\{ \lfloor \frac{n}{B} \rfloor , \left\lfloor \frac{S_2(P)}{k-|C_1|)} \right\rfloor \right\}$.

We create the values OPT_P of all $q+1$ partitions (using $O(q)$ operations for all these values) and pick the maximum value.

The output is the allocation of items for the partition P which maximizes OPT_P, which can be created using $O(q)$ operations, as established in Lemma 2. Note that the construction of Lemma 2, which we use here, is algorithmically required only for obtaining the final solution.

This gives a total of $O(q \log q)$ operations. The running time of the algorithm is in fact influenced by n, since each operation takes $O(\log n)$ time, given the size of input numbers. Thus, this algorithm would result in running time of $O(q \log q \cdot \log n)$, taking both the number of operations, and the time of each operation, into account.

Therefore, we established the following theorem.

Theorem 1 *There is an $O(q \log q \cdot \log n)$ time algorithm that solves (optimally) the offline version of* CCBC*.*

3 The online problem

In this section, we study the online problem. We start by defining and analyzing several reasonable algorithms that are all natural variants of the First fit algorithm. The last variant, which has the best performance is exactly B-competitive even for $k = 2$. Typically, First fit type algorithms have a relatively good performance for packing and covering problems, including the packing problem that is dual to CCBC (see [19]). This raises the question whether it is possible to get an improved upper bound for CCBC, using a different algorithm.

In Section 3.2 we show that any online algorithm has a non-constant competitive ratio. More precisely we show a lower bound of $1 + H_{k-1} = \Omega(\log k)$ on the competitive ratio where for every integer m, $H_m = 1 + \frac{1}{2} + \ldots + \frac{1}{m}$ denotes the m-th harmonic number. In subsection 3.3 we present the main result of this paper, that is an $O(k)$-competitive deterministic algorithm for the online problem.

3.1 Algorithm First fit (FF)

We start by considering the First fit (FF) algorithm, and showing that it performs poorly even if $k = 2$ (note that if $k = 1$ then FF is optimal).

The simplest way to define FF is as follows. Assign a new item to the first bin that is not covered yet. This algorithm is not competitive. The following example shows that a reasonable definition of FF should not keep packing items into bins that have enough items, unless these additional items have different colors. Let N be a large integer. We first have $N(B - k + 1)$ items, all of color class 1. Our algorithm assigns all these items to one bin, since it is not covered. Next, for $2 \leq i \leq k$, there are N items of color class i. After the arrival of these items, the first bin becomes covered, but no additional bin can be covered, since all additional bins would contain items of at most $k - 1$ color classes (actually, only one additional bin is created). An optimal solution would create N identical bins, each of which with $B - k + 1$ items of color class 1, and one item of each other color class.

This leads to the following possible alternative definition of FF. An item x of color c is packed into the first bin that is not covered, where x can contribute towards covering this bin.

This can happen either if the bin contains less than B items, or if the bin does not contain an item of color c.

Another reasonable definition of a FF type algorithm is the following algorithm FF(1). When a new item of color c arrives, we allocate it to the first uncovered bin that either contains less than B items, or contains at least B items, but does not contain an item of color c. If no such bin exists, we pack it in a new bin.

Note that FF(1) never packs an item into a new bin if it has a bin that contains less than B items. In the Appendix, we prove the following theorem.

Theorem 2 *The competitive ratio of* FF*(1) is exactly $B + k - 1$ for all values of k such that $k \geq 2$.*

Algorithm FF(2) is a modification of FF(1) that takes into account the fact that if a bin already has at least one item, but is lacking $t \leq k - 1$ colors, then in order to get covered, it must receive at least t additional items of other colors, and thus if a bin received a total of $k - t$ colors and it is filled by $B - t$ items, an additional item of one of the colors that are already packed in this bin would not be helpful towards covering the bin, and therefore in this case we do not see the bin as suitable for the item. For a bin that contains at least one item, we define the notion of *being useful* for adding an item of color c to the bin as follows. If the bin is covered, that is, it contains at least B items of at least k colors, then clearly adding an additional item is not useful. If the bin contains items of at most $k - t$ colors, for some $1 < t < k$, and c is one of these colors, then adding the new item is useful if the number of items is no larger than $B - t - 1$, and otherwise, not useful. In any other case (an uncovered bin that does not contain an item of color c, or an uncovered bin that already contains items of k different colors), the packing is useful. The algorithm is defined as follows. When a new item of color c arrives, we allocate it to the first bin for which adding the new item would be useful. If no such bin exists, we pack it into a new bin.

Theorem 3 *The competitive ratio of* FF*(2) is exactly B for all values of k such that $k \geq 2$.*

Proof To show the lower bound consider the following sequence consisting of four parts. Let N be a large integer and fix the value of k. The first part is $(B-k+1)N$ items of color 1. At the end of this part FF(2) will pack $B-k+1$ items in each of the first N bins. The second part consists of N items of each of the colors $2, 3, \ldots, k$. At this stage, FF(2) pack these items one of each color per each bin of the first N bins (and hence it will cover the first N bins). The third part is $(B - k + 1)BN$ items of color $k + 1$. The fourth part is BN items of each one of the colors $k + 2, \ldots, 2k - 1$. Since the third and fourth parts of the input (together) consist of only $k - 1$ colors, no additional bins can be covered by FF(2), except for the first N bins. Therefore the value of its solution is exactly N.

The optimal solution has exactly BN covered bins. Each such bin has $B - k + 1$ items of color $k + 1$, one item of the BN items of each of the colors $k + 2, k + 3, \ldots, 2k - 1$, and one item of just one of the colors $1, \ldots, k$ (in total, there are BN items of these colors). The ratio between the value of the two solutions is exactly B, and since N can be arbitrary large, we conclude that the asymptotic competitive ratio of FF(2) is at least B.

It remains to show the upper bound. Note that FF(2) never packs more than B items in one bin.

By definition of FF(2), an item is packed into a new bin only after all other bins have a total of at least $B-k+1$ items each. Moreover, we show that a bin that contains $B-t$ items for some $0 \le t < k$ contains items of at least $k-t$ colors. This can be proved by induction on $k-t$. If $k-t=1$, $B-t = B-k+1 \ge 1$, and thus it contains items of at least one color. Assume that the claim is true for $k-t=s$. $k-t$ increases to $s+1$ when a new item is assigned. If the bin already contains $s+1$ colors, we are done. Otherwise, FF(2) assigns a new item into a bin with $B-t$ items and $k-t$ colors. This is done only if the new item has a different color from the previous items in this bin.

We show that every bin that contains at least k colors, except for possibly the last bin ever opened, is covered. Assume that bin i has items of k colors but is not covered, and is not the last bin ever opened. Let $B-d$ be the number of items in bin i at the time that the next bin receives the first item. At this time, by our claim, bin i already has items of $k-d$ different colors. By definition of FF(2), since an item is assigned to bin i', bin i has exactly $k-d$ colors at this time. Since it eventually has at least k colors, it means that it receives d additional items, and thus becomes covered.

Let j be the index of the first bin (in the order in which bins were opened) that has less than k colors. If no such bin was opened, and j' bins were opened in total, by our claim, FF(2) has at least $j'-1$ covered bins, with B items each, and at most B items in bin j', therefore, OPT $\le j'$ and FF(2) $\ge j'-1$, which gives a ratio of at most 2 if $j' \ge 1$. If $j'=1$ then FF(2) $=$ OPT.

By our claim, all bins $1,\ldots,j-1$ are covered. The proof in this case is similar to the proof for FF(1). We have FF(2) $= j-1$, and bins $j, j+1, \ldots$ have items of at most $k-1$ distinct colors. Thus each bin of any optimal solution has at least one item from one of the first $j-1$ bins, and there are at most $B(j-1)$ such items, therefore OPT $\le B(j-1)$. □

3.2 Lower bound

In this section we show that any online algorithm has an unbounded competitive ratio if k can grow to infinity.

Theorem 4 *The competitive ratio of any deterministic online algorithm is at least* $1+H_{k-1} = \Omega(\log k)$. *Moreover, given a fixed value of* $B > k$, *the competitive ratio of any online algorithm is at least* $\frac{(B-1)(B-k+1)}{B(B-k)+B-1} \cdot \left(\frac{B-k}{B-1} + H_{k-1}\right) = \Omega(\log k)$. *For* $B = k$, *the competitive ratio of any online algorithm is at least* H_{k-1}.

Proof We fix k and B, and we first assume that $B > k$. Let N be a large integer, divisible by $B!$, thus N is divisible by $k-i$ for any $0 \le i \le k-1$ (since $B \ge k$), and by $B-1$. Our lower bound input sequence consists of two phases. In the first phase we have a set of N items of each of the colors $1,2,\ldots,k-1$. The second phase has k different scenarios. In the first scenario this phase consists of $\frac{N(k-1)}{B-1}$ items of color k. In this scenario the optimal solution has exactly $\frac{N(k-1)}{B-1}$ covered bins, each of them has $B-1$ items of colors $1,2,\ldots k-1$ (with at least one representative for each color class) and one additional item of color k.

For $i = 1, 2, \ldots, k-1$, the $(i+1)$-th scenario consists of NBk items of each of the i colors $k, k+1, \ldots, k+i-1$. The number of items from each color is chosen such that there can never be a shortage of items of these colors. In the $(i+1)$-th scenario the optimal solution consists of allocating exactly $k-i$ items of the first phase to each bin (with different colors) and additional $B-k+i$ items of the additional i colors, with at least one representative of each color. Hence $\frac{N(k-1)}{(k-i)}$ bins are covered by OPT.

Consider an online algorithm. Since the input consists of one of these k scenarios, we can assume without loss of generality that at the end of the first phase the algorithm has bins of several types. The first type is bins with $B-1$ items, and at least one representative of each color $1, 2, \ldots, k-1$. We denote the number of these bins by y. For every $j = 1, 2, \ldots, k-1$, there can be bins with exactly one item from each of (exactly) j colors (we denote the number of bins of this type for each value of j by x_j). To show that we may assume that no other types of bins need to be considered, note first that we can allow the variables $y, x_1, \ldots, x_{k-1}$ to be rational (but non-negative), rather than integral. We may also assume that a bin which contains i items, contains $\min\{i, k-1\}$ colors, since this is the maximum number of colors that such a bin can contain, and assuming the largest possible number of colors for each bin may only help the algorithm. Since the number of colors at this time is only $k-1$, every bin must receive at least one additional item, thus we may assume that no bin contains more than $B-1$ items. Now assume that the algorithm created a bin with $B-r$ items, such that $1 < r \leq B-k$. We replace it by two fractions of bins, one is a $\frac{r-1}{B-k}$-fraction of a bin with $k-1$ items, and the other one is a $\frac{B-k-r+1}{B-k}$-fraction of a bin with $B-1$ items. If the second phase is according to the first scenario, the algorithm may only benefit from the change, since r items that would cover a bin with $B-r$ items, are sufficient to cover the bin fractions; the fraction with $k-1$ items needs a $\frac{r-1}{B-k}$-fraction of $B-k+1$, and the fraction with $B-1$ items needs a $\frac{B-k-r+1}{B-k}$-fraction of a single item. Together this results in r items. In all other cases, both the original bin and the fractional bins created from it already have k colors and can be covered in all the other scenarios.

Then, in the first scenario the algorithm is able to cover only bins with either $B-1$ or $k-1$ items, i.e., at most $y + \frac{N(k-1)}{(B-1)(B-k+1)}$ bins. In the $(i+1)$-th scenario (for $i = 1, 2, \ldots, k-1$) the algorithm is able to cover at most $y + \sum_{j=k-i}^{k} x_j$ bins, as the algorithm can cover only bins having items of at least $k-i$ colors. Denote by R the competitive ratio of the algorithm, then the following are satisfied:

$$y(B-1) + \sum_{j=1}^{k-1} j x_j = (k-1)N \tag{1}$$

$$y \geq \frac{1}{R} \cdot \frac{N(k-1)}{B-1} - \frac{N(k-1)}{(B-1)(B-k+1)} \tag{2}$$

$$y + \sum_{j=k-i}^{k-1} x_j \geq \frac{1}{R} \cdot \frac{N(k-1)}{(k-i)} \quad \forall i = 1, 2, \ldots, k-1 \tag{3}$$

$$y, x_1, x_2, \ldots, x_{k-1} \geq 0, \tag{4}$$

where (1) holds by counting the number of elements in the first phase, (2) holds by the performance guarantee in case scenario 1 holds, (3) holds for all i by the performance guarantee in case the $(i+1)$-th scenario holds.

We sum up all the constraints (3) together with $B-k$ times constraint (2). In the resulting inequality we get that the left hand side is equal to the left hand side of equation (1) and therefore we obtain the following:

$$(k-1)N \geq \frac{1}{R} \cdot \left(\frac{N(k-1)(B-k)}{B-1} + \sum_{i=1}^{k-1} \frac{N(k-1)}{(k-i)} \right) - \frac{N(k-1)(B-k)}{(B-1)(B-k+1)}$$

$$= \frac{N(k-1)}{R} \cdot \left(\frac{B-k}{B-1} + H_{k-1} \right) - \frac{N(k-1)(B-k)}{(B-1)(B-k+1)},$$

and so $R \geq \frac{(B-1)(B-k+1)}{B(B-k)+B-1} \cdot \left(\frac{B-k}{B-1} + H_{k-1} \right)$. When B goes to infinity the last expression approaches $1 + H_{k-1}$, and we get that $R \geq 1 + H_{k-1}$ when B goes to infinity.

For fixed values of B, if $B > k$, the proof still holds and results in a lower bound of $\frac{(B-1)(B-k+1)}{B(B-k)+B-1} \cdot \left(\frac{B-k}{B-1} + H_{k-1} \right) = \Omega(\log k)$. If $B = k$, we apply a small modification to the sequence, and do not use the first scenario. Note that in the case $B = k$, the variables y and x_{k-1} correspond both to the same number, that is, to the number of bins with $k-1$ items of different colors. Thus we do not use a variable y, we substitute $y = 0$ in all conditions, and omit condition (2). This yields a lower bound of H_{k-1}, as required. □

Remark 1 *The competitive ratio of any randomized online algorithm is at least* $1 + H_{k-1} = \Omega(\log k)$. *Moreover, given a fixed value of* $B > k$, *the competitive ratio of any online algorithm is at least* $\frac{(B-1)(B-k+1)}{B(B-k)+B-1} \cdot \left(\frac{B-k}{B-1} + H_{k-1} \right) = \Omega(\log k)$. *For* $B = k$, *the competitive ratio of any online algorithm is at least* H_{k-1}.

Proof We note that the proof of Theorem 4 holds also for randomized algorithms. The variables y and x_i can denote the expected number of bins rather than the exact number. We can apply Yao's principle [21] (see also Chapter 8.3.1 in [4]). Yao's principle states that given a probability measure, defined over a set of input sequences, a lower bound the competitive ratio of any online algorithm (for a maximization problem) is implied by a lower bound on the ratio between the the expected value of an optimal solution divided by the expected value of a deterministic algorithm (both expectations are taken with respect to the probability distribution defined for the random choice of the input sequence).

In order to apply this method for any $B \geq k$, we choose the first scenario with probability $\frac{B-k}{B-1}$ and each other scenario with probability $\frac{1}{B-1}$. The analysis reduces to the analysis of the proof of Theorem 4. □

3.3 Algorithm Color&Size (CnS)

In this section, we consider the case $B \geq 2k$ and design an algorithm of competitive ratio of $O(k)$ for this case. If $B < 2k$, it is possible to use FF(2), which has a competitive ratio of at most $2k$.

Our algorithm is based on an online partition of the items into *color items* and *size items*. The first set of items is denoted by C and the second set is denoted by S. Afterwards we pack the items using the First fit algorithm into a joint set of bins, but the two types of items are packed independently of each other and even obliviously of the contents of the bins with respect to the other type of items. For items of S, the FIRST FIT algorithm packs an item in the first bin which contains less than B items of S. If no such bin exists, a new bin is opened. Note that an open bin may contain zero items of S, if it was opened to accommodate some item of C. The FIRST FIT algorithm packs an item of C into the first bin that has items of C of at most $k-1$ different colors, provided that the color of the current item is different from all the colors of items of C that the bin contains. If no such bin exists, a new bin is opened.

Note that it is possible to define the packing of items of S so that every bin would contain $B-k$ such items rather than B such items. Since a bin must contain at least k items of C to be covered, this would result in a sufficient number of items in a covered bin. However, since we assume $B \geq 2k$, this may only change the competitive ratio by a constant multiplicative factor, and therefore for simplicity, we analyze the algorithm in which each covered bin receives B items of S.

It remains to describe the online partition of the items into the sets C and S. Algorithm COLOR&SIZE (CnS) has an integer parameter p that we will select afterwards. Each color is assigned an index by the algorithm, according to its appearance order in the sequence. That is, a color is assigned index i if items of $i-1$ different colors appeared in the sequence before the 1st appearance of this color. Assume now that a new item that arrives, is the j-th item of color i. If one of the following conditions holds, the item belongs to C, and otherwise to S. The first condition is $i \bmod p \neq 0$ and $j \bmod p \neq 0$ and the second condition is $i \bmod p = 0$ and $j \bmod p \neq 1$. Thus, the very first item belongs to the set C.

Claim 2 *Given the input that contains at least p^2 items, the ratio of the numbers of items in C and S satisfies $p-2 \leq \frac{|C|}{|S|} \leq 2p(p-1)$.*

Proof First note that in a sequence of the first p^2 items there is at least one item of S. This holds since if there are at least p colors, then already the first item of color p is defined to be an item of S. Otherwise, there are at most $p-1$ colors and in this case there is at least one color with at least p items, and this color has an item in S. Therefore, the ratio in the claim is well-defined.

We first show that $(p-2)|S| \leq |C|$. We partition a subset of C into subsets of at least $p-2$ items, and each subset is assigned to some item of S, so that every item of S is assigned some subset of items of C.

We partition the colors into blocks of p consecutive colors. For every block that contains p colors, we consider the first item of each color. Out of these items, the first $p-1$ are in C and the last one is in S. From these $p-1$ items of C, we create a set and assign it to belong to the item of S. If there exists a block with less than p colors, the first items of its colors remain unassigned. We removed the first item of every color and thus we are left with colors which have more than one item.

The items of each color are now sequences of items of C, some of which are followed by an item of S. We put every subset of items of C that appear consecutively in a set, and assign it to belong to the item of S that appears after them in the same color. If there is no such item in S, these items of C remain unassigned. The subsets that are followed by an item of S in the sequences are of size $p-2$ (if they are items of indices $2,\ldots,p-1$ of some color whose first item is in C), or otherwise of size $p-1$.

Since every item of S was assigned at least $p-2$ items of C, and the sets of such items of C are disjoint, the claim follows.

We next show that $|C| \leq 2p(p-1)|S|$. We make a list of the items, such that we first write down the items of the first color, then the items of the second color and so on, for each color, the items are sorted according to their arrival order. In this list it suffices to show that there is no set of $p(p-1)+1$ consecutive items that are all in C. Assume the contrary, so there exists a set of $p(p-1)+1$ consecutive items of C. Note that after $p-1$ items of this list, if there is no item of S, it means that we must have moved to a new color (as for colors that start with an item of C, the p-th item is in S). Starting from the beginning of this list and after at most $p(p-1)$ items we considered at least p colors, and out of them, at least $p-1$ complete lists of colors, we must reach a color whose first item is in S. $\square$

Consider the bins that algorithm CnS creates with less than k colors of items of C. Let $C_1, C_2, \ldots, C_s$ be the sets of colors of the items of C in these bins (when ordered according to the order of the bins).

Claim 3 *We have $C_s \subseteq C_{s-1} \subseteq \cdots \subseteq C_1$. Moreover, the set of bins containing k items of C is a consecutive set of bins. If it is non-empty, it starts from the first bin ever opened.*

Proof We prove the first part of the claim. Assume by contradiction that for some t we do not have $C_{t+1} \subseteq C_t$. There is an item of C of a color $c \notin C_t$ that is assigned to the bin with the set of colors C_{t+1}. By definition of the FIRST FIT rule, this can only happen if $|C_t| = k$, which is not the case, contradiction. Assume now that there exists a bin j that received less than k colors, while bin $j+1$ received k colors. Similarly to the first part of the claim, since bin $j+1$ receives at least one item of C of a color that bin j does not have, we get a contradiction to the action of the algorithm. $\square$

Note that the items of S are assigned to bins consecutively, the first B items of S to the first bin, the next B items of S to the next bin etc. Therefore the set of bins containing B items of S is a prefix of the bins. By Claim 3 we find that the set of bins containing k items of C is a prefix of the bins as well. Since one of these prefixes may be longer than the other, we need to consider these two cases.

We first consider the case where the prefix of bins with k items of C is longer than the prefix of bins with B items of S. That is, there is a bin (in the output of CnS) that has t items of S where $0 \leq t < B$ and exactly k items of C. In this case $\mathrm{CnS} = \lfloor \frac{|S|}{B} \rfloor$ and OPT is at most the total number of items divided by B. This amount is $\frac{|S|+|C|}{B} \leq \frac{|S|\cdot(1+2p(p-1))}{B} \leq (\mathrm{CnS}+1)(2p^2-2p+1)$.

In the remaining case we can assume that every bin of CnS with k items of C has exactly B items of S. We define C' to be the subset of C of all items whose color does not belong to C_1. Similarly we define S' as the subset of S of all items whose color does not belong to C_1. Since each bin of the optimal solution must contain at least one item whose color is not in C_1 (as $|C_1| \leq k-1$), we conclude that $\mathrm{OPT} \leq |S'| + |C'|$.

We next argue that $\mathrm{CnS} \geq \frac{|C'|}{k}$. This is so because the number of covered bins in the solution returned by CnS is exactly the number of bins that have k items of C. If a bin has k items of C', then it is clearly covered by CnS. Since each item of C' contributes to one of the covered bins, we conclude that the number of covered bins is at least $\frac{|C'|}{k}$ and hence $|C'| \leq k \cdot \mathrm{CnS}$.

Claim 4 $(p-2)|S'| \le |C'| + k - 1$

Proof We use the additive constant of $k-1$ as a way to make up deleted colors. We consider this as a global budget.

We apply a similar proof to the proof of the first part of Claim 2. We partition all colors (including the colors of C_1) into blocks of p consecutive colors, and assign the subsets of C to items of S exactly as in that proof. That is, for every block with exactly p colors, we assign the set of first items of the $p-1$ first colors to the item of S which is the first item of the p-th color of this block. Each other item of S receives a subset of items of C of the same color. Thus, when the C_1 colors are removed, the only case where an item of S was not removed while the subset of items of C assigned to it was changed, is the case where all these items are each a first item of some color. Thus, the number of items of C removed in this way without an item of S they were assigned to, is at most the number of removed colors, i.e., at most $k-1$. Hence this claim follows. □

We conclude that in this case $\text{OPT} \le |S'| + |C'| \le \frac{|C'|+k-1}{p-2} + |C'| = \frac{p-1}{p-2} \cdot |C'| + \frac{k-1}{p-2} \le \frac{(p-1)k}{p-2}\text{CnS} + \frac{k-1}{p-2}$. We conclude that the competitive ratio of CnS is at most $\max\{2p^2 - 2p + 1, \frac{(p-1)k}{p-2}\}$.

If $k \ge 36$, we use $p = \lfloor \frac{\sqrt{k}}{2} \rfloor$. Then $p \ge 3$ and $\frac{p-1}{p-2} \le 2$, so the second term in the maximum is at most $2k$ (but is approximately k for large k). The first term in this case is smaller than k. Otherwise, we use $p = 5$, and get the competitive ratio of at most $\max\{41, 140/3\} = O(1)$. The competitive ratio is therefore $O(k)$. Note that the additive constant in all cases is $O(k)$. Hence we established the following.

Theorem 5 *The algorithm CnS has competitive ratio of $O(k)$ (with an additive constant of $O(k)$).*

We summarize the algorithm as follows.

Algorithm CnS

1. If $k \ge 36$ let $p = \lfloor \frac{\sqrt{k}}{2} \rfloor$, and otherwise let $p = 5$.
2. Let $C = S = \emptyset$.
3. Upon arrival of a new item, which is the j-th item of the i-th color.
   (a) If ($i \bmod p \ne 0$ and $j \bmod p \ne 0$) or ($i \bmod p = 0$ and $j \bmod p \ne 1$), augment the set C with the new item. Otherwise, augment S with the new item.
   (b) If the new item is in C, pack it into the first bin that contains items of C of at most $k-1$ distinct colors, none of which is color i.
   (c) If the new item is in S, pack it into the first bin that contains less than B items of the set S.

4 Concluding remarks

In this paper we studied the case of equal sized items. The study of the generalization of the problem where each item has an arbitrary non-negative size is left for future research. Moreover, closing the gaps of the competitive ratio of the online algorithms (for the identical size case) between the lower bound and the upper bound is also left for future research.

Note that our results for the online algorithms hold for any value of B, including large values. When B is of the same order as the size of k, one can hope to get better competitive ratios. This line of research is also left for future investigation.

References

[1] S.F. Assmann. *Problems in discrete applied mathematics.* PhD thesis, Mathematics Department, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1983.

[2] S.F. Assmann, D.S. Johnson, D.J. Kleitman, and J.Y.-T. Leung. On a dual version of the one-dimensional bin packing problem. *Journal of Algorithms*, 5:502–525, 1984.

[3] L. Babel, B. Chen, H. Kellerer, and V. Kotov. Algorithms for on-line bin-packing problems with cardinality constraints. *Discrete Applied Mathematics*, 143(1-3):238–251, 2004.

[4] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[5] A. Caprara, H. Kellerer, and U. Pferschy. Approximation schemes for ordered vector packing problems. *Naval Research Logistics*, 92:58–69, 2003.

[6] E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In D. Hochbaum, editor, *Approximation algorithms.* PWS Publishing Company, 1997.

[7] E. G. Coffman Jr. and J. Csirik. Performance guarantees for one-dimensional bin packing. In T. F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics*, chapter 32, pages (32–1)–(32–18). Chapman & Hall/Crc, 2007.

[8] J. Csirik, D. S. Johnson, and C. Kenyon. Better approximation algorithms for bin covering. In *Proc.of the 12th Annual Symposium on Discrete Algorithms (SODA2001)*, pages 557–566, 2001.

[9] J. Csirik and J. Y.-T. Leung. Variable-sized bin packing and bin covering. In T. F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics*, chapter 34, pages (34–1)–(34–11). Chapman & Hall/Crc, 2007.

[10] J. Csirik and J. Y.-T. Leung. Variants of classical one-dimensional bin packing. In T. F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics*, chapter 33, pages (33–1)–(33–13). Chapman & Hall/Crc, 2007.

[11] J. Csirik and V. Totik. On-line algorithms for a dual version of bin packing. *Discrete Applied Mathematics*, 21:163–167, 1988.

[12] J. Csirik and G. J. Woeginger. On-line packing and covering problems. In A. Fiat and G. J. Woeginger, editors, *Online Algorithms: The State of the Art*, chapter 7, pages 147–177. Springer, 1998.

[13] L. Epstein, C. Imreh, and A. Levin. Bin covering with cardinality constraints. Manuscript, 2007.

[14] L. Golubchik, S. Khanna, S. Khuller, R. Thurimella, and A. Zhu. Approximation algorithms for data placement on parallel disks. In *Proc. of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA2000)*, pages 223–232, 2000.

[15] K. Jansen and R. Solis-Oba. An asymptotic fully polynomial time approximation scheme for bin covering. *Theoretical Computer Science*, 306(1-3):543–551, 2003.

[16] S. O. Krumke, W. de Paepe, J. Rambau, and L. Stougie. Online bin coloring. In *Proc. of the 9th Annual European Symposium on Algorithms (ESA2001)*, pages 74–85, 2001.

[17] H. Shachnai and T. Tamir. On two class-constrained versions of the multiple knapsack problem. *Algorithmica*, 29(3):442–467, 2001.

[18] H. Shachnai and T. Tamir. Polynomial time approximation schemes for class-constrained packing problems. *Journal of Scheduling*, 4(6):313–338, 2001.

[19] H. Shachnai and T. Tamir. Tight bounds for online class-constrained packing. *Theoretical Computer Science*, 321(1):103–123, 2004.

[20] E. C. Xavier and F. K. Miyazawa. The class constrained bin packing problem with applications to video-on-demand. In *Proc. of the 12th Annual International Conference on Computing and Combinatorics (COCOON 2006)*, pages 439–448, 2006.

[21] A. C. C. Yao. Probabilistic computations: towards a unified measure of complexity. In *Proc. of the 18th Symposium on Foundations of Computer Science (FOCS'77)*, pages 222–227, 1977.

A Proof of Theorem 2

To show the lower bound consider the following sequence consisting of four parts. Let N be a large integer and fix the value of k. The first part is BN items of color 1. At the end of this part $\text{FF}(1)$ will pack B items in each of the first N bins. The second part consists of N items of each of the colors $2, 3, \ldots, k$. At this stage, $\text{FF}(1)$ pack these items one of each color per each bin of the first N bins (and hence it will cover the first N bins). The third part is $(B-k+1)(B+k-1)N$ items of color $k+1$. The fourth part is $(B+k-1)N$ items of each one of the colors $k+2, \ldots, 2k-1$. Since the third and fourth parts of the input (together) consist of only $k-1$ colors, no additional bins can be covered by $\text{FF}(1)$, except for the first N bins. Therefore the value of its solution is exactly N.

The optimal solution has exactly $(B+k-1)N$ covered bins. Each such bin has $B-k+1$ items of color $k+1$, one item of the $(B+k-1)N$ items of each of the colors $k+2, k+3, \ldots, 2k-1$, and one item of just one of the colors $1, \ldots, k$ (in total, there are $(B+k-1)N$ items of these colors). The ratio between the value of the two solutions is exactly $B+k-1$, and since N can be arbitrary large, we conclude that the asymptotic competitive ratio of $\textsc{ff}(1)$ is at least $B+k-1$.

It remains to show the upper bound. By definition, a bin can receive additional items after it has B items only if it receives an item of a new color that this bin does not have, and this can only happen $k-1$ times. Therefore, no bin contains more than $B+k-1$ items.

If all created bins are covered, if there are j bins, there are at most $j(B+k-1) < 2jB$ items (using $k \leq B$), and the competitive ratio is at most 2, since $\textsc{opt} < 2j$ and $\textsc{ff}(1) = j$.

Otherwise, let $j \geq 1$ be the index of the first bin (in the order in which bins were opened) that was created but is not covered. If bin j contains less than B items, then by definition of $\textsc{ff}(1)$, bin $j+1$ does not exist. In this case, consider bins $1, \ldots, j-1$, that contain at most $B+k-1$ items each. In total, there are at most $(j-1)(B+k-1)+B-1 \leq (2B-1)(j-1)+B-1 \leq (2j-1)B-j$ items (using $k \leq B$). For $j \geq 2$, this gives at most $3(j-1)B$ items. Therefore $\textsc{opt} \leq 3(j-1)$, whereas $\textsc{ff}(1) = j-1$. In this case, the competitive ratio is at most 3. If $j=1$, there are less than B items, and $\textsc{ff}(1) = \textsc{opt} = 0$.

In the case that bin j has at least B items, and since it is uncovered, it has items of at most $k-1$ distinct colors. Denote the set of colors in bin j by C'. Any later bin, $j' > j$ can only have items of colors from C', otherwise $\textsc{ff}(1)$ would have assigned an item of an additional color to bin j. Thus, since any bin of an optimal solution must have items of at least k colors, it must have at least one item from bins $1, \ldots, j-1$ of $\textsc{ff}(1)$. As we saw, the number of items in these bins is at most $(j-1)(B+k-1)$ and thus $\textsc{opt} \leq (j-1)(B+k-1)$. Since $\textsc{ff}(1) = j-1$, the ratio follows. $\square$