Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

SVEN O. KRUMKE WILLEM DE PAEPE JÖRG RAMBAU
LEEN STOUGIE

Online Bin-Coloring

ZIB-Report 01-07 (April 2001)

ONLINE BIN-COLORING

SVEN O. KRUMKE [1], WILLEM E. DE PAEPE [2], JÖRG RAMBAU [1], AND LEEN STOUGIE [3]

ABSTRACT. We introduce a new problem that was motivated by a (more complicated) problem arising in a robotized assembly environment. The *bin coloring problem* is to pack unit size colored items into bins, such that the maximum number of different colors per bin is minimized. Each bin has size $B \in \mathbb{N}$. The packing process is subject to the constraint that at any moment in time at most $q \in \mathbb{N}$ bins are partially filled. Moreover, bins may only be closed if they are filled completely. An online algorithm must pack each item must be packed without knowledge of any future items.

We investigate the existence of competitive online algorithms for the bin coloring problem. We prove an upper bound of $3q - 1$ and a lower bound of $2q$ for the competitive ratio of a natural greedy-type algorithm, and show that surprisingly a trivial algorithm which uses only one open bin has a strictly better competitive ratio of $2q - 1$. Moreover, we show that any deterministic algorithm has a competitive ratio $\Omega(q)$ and that randomization does not improve this lower bound even when the adversary is oblivious.

1. INTRODUCTION

One of the commissioning departments in the distribution center of Herlitz PBS AG, Falkensee, one of the main distributors of office supply in Europe, is devoted to greeting cards. The cards are stored in parallel shelving systems. Order pickers on automated guided vehicles collect the orders from the storage systems, following a circular course through the shelves. At the loading zone, which can hold q vehicles, each vehicle is logically "loaded" with B orders which arrive online. The goal is to avoid congestion among the vehicles (see [AG$^+$98] for details). Since the vehicles are unable to pass each other and the "speed" of a vehicle is correlated to the number of different stops it must make, this motivates to assign the orders to vehicles in such a way that the vehicles stop as few times as possible.

The above situation motivated the following *bincoloring problem*: One receives a sequence of unit size items $\sigma = r_1, \ldots, r_m$ where each item has a *color* $r_i \in \mathbb{N}$, and is asked to pack them into bins with size B. The goal is to pack the items into the bins "most uniformly", that is, to minimize the maximum number of different colors assigned to a bin. The packing process is subject to the constraint that at any moment in time at most $q \in \mathbb{N}$ bins may be partially filled. Bins may only be closed if they are filled completely. (Notice that without these strict bounded space constraints the problem is trivial since in this case each item can be packed into a separate bin).

In the *online version* of the problem, denoted by OLBCP, each item must be packed without knowledge of any future items. An online algorithm is *c-competitive*, if for all possible request sequences the maximum colors in the bins packed by the algorithm and the optimal offline solution is bounded by c. Trivially, any algorithm for OLBCP is B-competitive, where B denotes the size of the bins.

---

[1]Konrad-Zuse-Zentrum für Informationstechnik Berlin, Department Optimization, Takustr. 7, D-14195 Berlin-Dahlem, Germany. Email: {krumke,rambau}@zib.de. Research supported by the German Science Foundation (DFG, grant Gr 883/5-3)

[2]Faculty of Technology Management, Technical University of Eindhoven, P. O. Box 513, 5600MB Eindhoven, The Netherlands. Email: w.e.d.paepe@tm.tue.nl

[3]Combinatorial Optimization Group, Technical University of Eindhoven, P. O. Box 513, 5600MB Eindhoven, The Netherlands. Email: leen@win.tue.nl. Supported by the TMR Network DONET of the European Community ERB TMRX-CT98-0202

The OLBCP can be viewed as a variant of the bounded space binpacking problem in (see [CGJ97, CW98] for recent surveys on binpacking problems).

Summary of Results. We investigate the existence of competitive online algorithms for the OLBCP. Our results reveal a curiosity of competitive analysis: a truly stupid algorithm achieves essentially a (non-trivial) best possible competitive ratio for the problem whereas a seemingly reasonable algorithm performs provably worse in terms of competitive analysis.

We first analyze a natural "greedy-type" strategy, and show that this strategy has a competitive ratio no greater than $3q$ but no smaller than $2q$, where q is the maximum number of bins that may be partially filled (open) at the same time. We show that a trivial strategy that only uses one open bin, has a strictly better competitive ratio of $2q - 1$. Then we show that surprisingly no deterministic algorithm can be substantially better than the trivial strategy. More specifically, we prove that no deterministic algorithm can, in general, have a competitive ratio less than q. Even more surprising, the general lower bound of q for the competitive ratio continues to hold for randomized algorithms against an oblivious adversary. Finally, not even "resource augmentation", which means that the online algorithm is allowed to use a fixed number $q' \geq q$ of open bins instead of q, can help to overcome the lower bound of $\Omega(q)$ on the competitive ratio.

Paper Outline. The paper is organized as follows. In Section 2 we formally define the OLBCP and introduce notation. In Section 3 we describe and analyze the obvious algorithm GREEDYFIT. In Section 4 we introduce and analyze the trivial algorithm ONEBIN which surprisingly obtains a better competitive ratio than GREEDYFIT. Sections 5 and 6 contain general lower bounds for deterministic and randomized algorithms.

2. PROBLEM DEFINITION

We start by defining the problem under study.

Definition 2.1 (Online Bin Coloring Problem)**.** In the *Online Bin Coloring Problem (*OLBCP$_{B,q}$*)* with parameters $B, q \in \mathbb{N}$ ($B, q \geq 2$), one is given a sequence $\sigma = r_1, \dots, r_m$ of unit size items (requests), each with a color $r_i \in \mathbb{N}$, and is asked to pack them into bins with size B, that is, each bin can accommodate exactly B items. The packing is subject to the following constraints:

(1) The items must be packed according to the order of their appearance, that is, item i must be packed before item k for all $i < k$.
(2) At most q partially filled bins may be open to further items at any point in the packing process.
(3) A bin may only be closed if it is filled completely, i.e., if it has been assigned exactly B items.

The objective is to minimize the maximum number of different colors assigned to a bin.

An online algorithm for OLBCP$_{B,q}$ must pack each item r_i (irrevocably) without knowledge of requests r_k with $k > i$.

In the sequel it will be occasionally helpful to use the following view on the bins used by an arbitrary algorithm ALG to process an input sequence σ. Each open bin has an *index* x, where $1 \leq x \leq q$. Each time a bin with index x is closed (since it is filled completely) and a new bin is opened the new bin will also have index x. If no confusion can occur, we will refer to a bin with index x as *bin* x.

We denote by $\mathsf{ALG}(\sigma)$ the objective function value of the solution produced by an algorithm ALG on input σ. We use OPT to denote an optimal offline algorithm. The algorithm OPT has complete knowledge about the input sequence σ in advance. However, the packing must still obey the constraints 1 to 3 specified in Definition 2.1.

Definition 2.2 (Competitive Algorithm)**.** A deterministic online algorithm ALG for OLBCP$_{B,q}$ is *c-competitive*, if there exists a constant c such that for any request sequence σ

$$\mathsf{ALG}(\sigma) \leq c \cdot \mathsf{OPT}(\sigma).$$

The competitive ratio of an algorithm ALG is the smallest number c such that ALG is c-competitive. As noted in the introduction the size of the bins B is a trivial upper bound on the competitive ratio of *any* algorithm for $\text{OLBCP}_{B,q}$.

A randomized online algorithm is a probability distribution over a set of deterministic online algorithms. The objective value produced by a randomized algorithm is therefore a random variable. In this paper we analyze the performance of randomized online algorithms only against an *oblivious adversary*. An oblivious adversary does not see the realizations of the random choices made by the online algorithm and therefore has to generate a request sequence in advance. We refer to [BEY98] for details on the various adversary models.

Definition 2.3 (Competitive Randomized Algorithm)**.** A randomized online algorithm RALG is c*-competitive against an oblivious adversary* if for any request sequence σ

$$\mathbb{E}\left[\mathsf{RALG}(\sigma)\right] \leq c \cdot \mathsf{OPT}(\sigma).$$

3. THE ALGORITHM GREEDYFIT

In this section we introduce a natural greedy-type strategy, which we call GREEDYFIT, and show that the competitive ratio of this strategy is at most $3q$ but no smaller than $2q$ (provided the capacity B is sufficiently large).

> **GREEDYFIT:** If upon the arrival of request r_i the color r_i is already contained in one of the currently open bins, say bin b, then put r_i into bin b. Otherwise put item r_i into a bin that contains the least number of different colors (which means opening a new bin if currently less than q bins are non-empty).

The analysis of the competitive ratio of GREEDYFIT is essentially via a pigeon-hole principle argumentation. We first show a lower bound on the number of bins that *any* algorithm can use to distribute a the items in a contiguous subsequence and then relate this number to the number of colors in the input sequence.

Lemma 3.1. *Let $\sigma = r_1, \dots, r_m$ be any request sequence and let $\sigma' = r_i, \dots, r_{i+\ell}$ be any contiguous subsequence of σ. Then any algorithm packs the items of σ' into at most $2q + \lfloor(\ell - 2q)/B\rfloor$ different bins.*

Proof. Let ALG be any algorithm and let $b_1, \dots, b_t$ be the set of open bins for ALG just prior to the arrival of the first item of σ'. Denote by $f(b_j) \in \{1, \dots, B-1\}$ the empty space in bin b_j at that moment in time. To close an open bin b_j, ALG needs $f(b_j)$ items. Opening and closing an additional new bin needs B items. To achieve the maximum number of bins ($\geq 2q$), ALG must first close each open bin and put at least one item into each newly opened bin. From this moment in time, opening a new bin requires B new items. Thus, it follows that the maximum number of bins ALG can use is bounded from above as claimed in the lemma. □

Theorem 3.2. *Algorithm* GREEDYFIT *is c-competitive for $\text{OLBCP}_{B,q}$ with $c = \min\{2q + \lfloor(qB - 3q + 1)/B\rfloor, B\}$.*

Proof. Let σ be any request sequence and suppose $\mathsf{GREEDYFIT}(\sigma) = w$. It suffices to consider the case $w \geq 2$. Let s be the smallest integer such that $\mathsf{GREEDYFIT}(r_1, \dots, r_{s-1}) = w - 1$ and $\mathsf{GREEDYFIT}(r_1, \dots, r_s) = w$. By the construction of GREEDYFIT, after processing $r_1, \dots, r_{s-1}$ each of the currently open bins must contain exactly $w - 1$ different colors. Moreover, since $w \geq 2$, after processing additionally request r_s, GREEDYFIT has exactly q open bins (where as an exception we count here the bin where r_s is packed as open even if by this assignment it is just closed). Denote those bins by $b_1, \dots, b_q$.

Let bin b_j be the bin among $b_1, \dots, b_q$ that has been opened last by GREEDYFIT. Let r'_s be the first item that was assigned to b_j. Then, the subsequence $\sigma' = r_{s'}, \dots, r_s$ consists of at most $qB - (q-1)$ items, since between $r_{s'}$ and r_s no bin is closed and at the moment $r_{s'}$ was processed, $q - 1$ bins already contained at least one item. Moreover, σ' contains items with at least w different colors. By Lemma 3.1 OPT distributes the items of σ' into at most $2q + \lfloor (qB - 3q + 1)/B \rfloor$ bins. Consequently,

$$\mathsf{OPT}(\sigma) \geq \frac{w}{2q + \lfloor (qB - 3q + 1)/B \rfloor},$$

which proves the claim. □

Corollary 3.3. *Algorithm* GREEDYFIT *is c-competitive for* $\mathrm{OLBCP}_{B,q}$ *with* $c = \min\{3q - 1, B\}$. □

We continue to prove a lower bound on the competitive ratio of GREEDYFIT.

Theorem 3.4. GREEDYFIT *has a competitive ratio greater or equal to* $2q$ *for the* $\mathrm{OLBCP}_{B,q}$ *if* $B \geq 2q^3 - q^2 - q + 1$.

Proof. We construct a request sequence σ that consists of a finite number M of phases in each of which qB requests are given. The sequence is constructed in such a way that after each phase the adversary has q empty bins.

Each phase consists of two steps. In the first step q^2 items are presented, each with a new color which has not been used before. In the second step $qB - q^2$ items are presented, all with a color that has occurred before. We will show that we can choose the items given in Step 2 of every phase such that the following properties hold for the bins of GREEDYFIT:

- **Property 1:** The bins with indices $1, \ldots, q-1$ are never closed.
- **Property 2:** The bins with indices $1, \ldots, q-1$ contain only items of different colors.
- **Property 3:** There is an $M \in \mathbb{N}$ such that during Phase M GREEDYFIT assigns for the first time an item with a new color to a bin that already contains items with $2q^2 - 1$ different colors.
- **Property 4:** There is an assignment of the items of σ such that no bin contains items with more than q different colors.

We analyze the behavior of GREEDYFIT by distinguishing between the items assigned to the bin (with index) q and the items assigned to bins (with indices) 1 through $q-1$. Let L_k be the set of colors of the items assigned to bins $1, \ldots, q-1$ and let R_k be the set of colors assigned to bin q during Step 1 of Phase k.

We now describe a general construction of the request sequence given in Step 2 of a phase. During Step 1 of Phase k there are items with $|R_k|$ different colors assigned to bin q. For the moment, suppose that $|R_k| \geq q$ (see Lemma 3.7 (iv)). We now partition the at most q^2 colors in $|R_k|$ into q disjoint non-empty sets $S_1, \ldots, S_q$. We give $qB - q^2 \geq 2q^2$ items with colors from $|R_k|$ such that the number of items with colors from S_j is $B - q$ for every j, and the last $|R_k|$ items all have a different color.

GREEDYFIT will pack all items given in Step 2 into bin q (Lemma 3.7 (iii)). Hence bins $1, \ldots, q-1$ only get assigned items during Step 1, which implies the properties 1 and 2.

The adversary assigns the items of Step 1 such that every bin receives q items, and the items with colors in the color set S_j go to bin j. Clearly, the items in every bin have no more than q different colors. The items given in Step 2 can by construction of the sequence be assigned to the bins of the adversary such that all bins are completely filled, and the number of different colors per bin does not increase (this ensures that property 4 is satisfied).

Lemma 3.5. *At the end of Phase $k < M$, bin q of* GREEDYFIT *contains exactly $B - \sum_{j \leq k} |L_j|$ items, and this number is at least q^2.*

Proof. After Phase k, exactly kqB items have been given. Moreover, after k phases bins 1 through $q-1$ contain exactly $\sum_{j \leq k} |L_j|$ items because the items of Step 2 are always packed into bin q by GREEDYFIT. Thus, the number of items in bin q of GREEDYFIT equals

$$kqB - \sum_{j \leq k} |L_j| \mod B = \underbrace{B - \sum_{j \leq k} |L_j|}_{<B} \mod B.$$

We show that $B - \sum_{j \leq k} |L_j| \geq q^2$. This implies that $B - \sum_{j \leq k} |L_j| \mod B = B - \sum_{j \leq k} |L_j|$.

Since $k < M$ we know that each of the bins 1 through $q-1$ contains at most $2q^2 - 1$ colors. Thus, $\sum_{j \leq k} |L_j| \leq (2q^2 - 1)(q-1) = 2q^3 - 2q^2 - q + 1$. It follows from the assumption on B that $B - \sum_{j \leq k} |L_j| \geq q^2$. $\square$

Corollary 3.6. *For any Phase $k < M$, bin q is never closed by* GREEDYFIT *before the end of Step 1 of Phase k.*

Proof. The claim clearly holds for the first phase. Hence for the remainder we consider the case $k > 1$.

Since there are exactly q^2 items presented in Step 1 of any phase, the claim is true by Lemma 3.5 as soon as $|\sum_{j \leq k} L_j| \geq q^2$ at the beginning of Phase k: in that case, there is even enough space in bin q to accommodate all items given in Step 1. We show that $|L_1| + |L_2| \geq q^2$ which implies that $|\sum_{j \leq k} L_j| \geq q^2$ for $k \geq 2$.

After Phase 1, each bin of GREEDYFIT contains q colors, which yields $|L_1| = q(q-1)$. It is easy to see that all items presented in Step 2 of the first phase are packed into bin q by GREEDYFIT: All these items have colors from R_1 where $|R_1| = q$. Either a color from R_1 is currently already present in bin q or bin q has less than q different colors, while all other bins contain q colors. In either case, GREEDYFIT packs the corresponding item into bin q.

By Lemma 3.5 at the end of Phase 1 bin q contains at least q^2 items. Since the last $|R_1| = q$ items presented in Step 2 of the first phase have all different colors (and all of these are packed into bin q as shown above) we can conclude that at the beginning of Phase 2 bin q of GREEDYFIT already contains q colors. Thus, in Step 1 of Phase 2 GREEDYFIT again puts q items into each of its bins. At this point, the total number of distinct colors in the first $q-1$ bins is at least $(q-1)q + (q-1)q = 2q^2 - 2q \geq q^2$ for $q > 1$, so that $|L_1| + |L_2| \geq q^2$. As noted above, this implies the claim. $\square$

The success of our construction heavily relies on the fact that at the beginning of each phase, bin q of GREEDYFIT contains at least q colors. We show that this is indeed true.

Lemma 3.7. *For $k \geq 1$ the following statements are true:*

(i) *At the beginning of Phase k bin q of* GREEDYFIT *contains exactly the colors from R_{k-1} (where $R_0 := \emptyset$).*
(ii) *After Step 1 of Phase k, each of the bins $1, \ldots, q-1$ of* GREEDYFIT *contains at least $|R_k| + |R_{k-1}| - 1$ different colors.*
(iii) *In Step 2 of Phase k* GREEDYFIT *packs all items into bin q.*
(iv) $|R_k| \geq q$.

Proof. The proof is by induction on k. All claims are easily seen to be true for $k = 1$. Hence, in the inductive step we assume that statements (i)–(iv) are true for some $k \geq 1$ and we consider Phase $k+1$.

(i) By the induction hypothesis (iii) all items from Step 2 presented in Phase k were packed into bin q by GREEDYFIT. Since at the end of Phase k bin q contains at least $q^2 \geq |R_k|$ items (see Lemma 3.5) and the last R_k items presented in Phase k had different colors, it follows that at the beginning of Phase $k+1$ bin q contains *at least* all colors from R_k. On the other hand, since all the $Bq - q^2 > B$ items from Step 2 were packed into bin q by GREEDYFIT, this bin was closed during this process and consequently can only contain colors from R_k.
(ii) By Corollary 3.6 bin q is not closed before the end of Step 1. After Step 1 all colors from R_{k+1} are already in bin q by construction. Since by (i) before Step 1 also all colors from R_k were contained in bin q, it follows that bin q contains at least $|R_k| + |R_{k+1}|$ different colors at the end of Step 1. By construction of GREEDYFIT each of the bins $1, \ldots, q-1$ must then contain at least $|R_k| + |R_{k+1}| - 1$ different colors.

(iii) When Step 2 starts then all colors from R_{k+1} are already in bin q by construction. Therefore, GREEDYFIT will initially pack items with colors from R_{k+1} into bin q as long as this bin is not yet filled up. We have to show that after bin q has been closed the number of colors in any other bin is always larger than in bin q. This follows from (ii), since by (ii) each of the bins $1, \dots, q-1$ has at least $|R_k| + |R_{k+1}| \geq |R_{k+1}| + q - 1 > |R_{k+1}|$ colors after Step 2 of Phase $k+1$.

(iv) At the beginning of Phase $k+1$ bin q contains exactly $|R_k|$ colors by (i). By the induction hypothesis (ii) and (iii) each of the bins $1, \dots, q-1$ contains at least $|R_k| + |R_{k-1}| - 1 \geq |R_k|$ colors. Hence, at the beginning of Phase $k+1$, the minimum number of colors in bins $1, \dots, q-1$ is no smaller than the number of colors in bin q. It follows from the definition of GREEDYFIT that during Step 1 of Phase $k+1$, bin q is assigned at least the $q^2/q = q$ colors. In other words, $|R_{k+1}| \geq q$.

□

To this point we have shown that we can actually construct the sequence as suggested, and that the optimal offline cost on this sequence is no more than q. Now we need to prove that there is a number $M \in \mathbb{N}$ such that after M phases there is a bin from GREEDYFIT that contains items with $2q^2$ different colors. We will do this by establishing the following lemma:

Lemma 3.8. *In every two subsequent Phases k and $k+1$, either $|L_k \cup L_{k+1}| > 0$ or bin q contains items with $2q^2$ different colors during one of the two phases.*

Proof. Suppose that there is a Phase k in which $|L_k| = 0$. This means that all q^2 items given in Step 1 are assigned to bin q ($|R_k| = q^2$). By Lemma 3.7 (i), at the beginning of Phase $k+1$, bin q still contains q^2 different colors. If in Step 1 of Phase $k+1$ again all q^2 items are assigned to bin q, bin q contains items with $2q^2$ different colors (recall that bin q is never closed before the end of Step 1 by Corollary 3.6). If less than q^2 items are assigned to bin q then one of the other bins gets at least one item, and $|L_{k+1}| > 0$. □

We can conclude from Lemma 3.8 that at least once every two phases the number of items in the bins 1 through $q-1$ grows. Since these bins are never closed (property 1), and all items have a unique color (property 2), after a finite number M of phases, one of the bins of GREEDYFIT must contain items with $2q^2$ different colors. This completed the proof of the Theorem. □

4. THE TRIVIAL ALGORITHM ONEBIN

This section is devoted to arguably the simplest (and most trivial) algorithm for the OLBCP, which surprisingly has a better competitive ratio than GREEDYFIT. Moreover, as we will see later that this algorithm achieves essentially the best competitive ratio for the problem.

Algorithm ONEBIN: The next item r_i is packed into the (at most one) open bin. A new bin is opened only if the previous item has closed the previous bin by filling it up completely.

The proof of the upper bound on the competitive ratio of ONEBIN is along the same lines as that of GREEDYFIT.

Lemma 4.1. *Let $\sigma = r_1, \dots, r_m$ be any request sequence. Then for $i \geq 0$ any algorithm packs the items $r_{iB+1}, \dots, r_{(i+1)B}$ into at most $\min\{2q-1, B\}$ bins.*

Proof. It is trivial that the B items $r_{iB+1}, \dots, r_{(i+1)B}$ can be packed into at most B different bins. Hence we can assume that $2q-1 \leq B$, which means $q \leq (B-1)/2 \leq B$.

Consider the subsequence $\sigma' = r_{iB+1}, \dots, r_{(i+1)B}$ of σ. Let ALG be any algorithm and suppose that just prior to the arrival of the first item of σ', algorithm ALG has l open bins. Denote these bins by $b_1, \dots, b_t$. Let $f(b_j) \in \{1, \dots, B-1\}$ be the number of empty places in bin b_j, $j = 1, \dots, t$. Notice that

$$\sum_{j=1}^{t} f(b_j) \equiv 0 \mod B. \tag{1}$$

Suppose that ALG uses at least $2q$ bins to distribute the items of σ'. By arguments similar to those given in Lemma 3.1, ALG can maximize the number of bins used only by closing each currently open bin and put at least one item into each of the newly opened bins. To obtain at least $2q$ bins at least $\sum_{j=1}^{t} f(b_j)+(q-t)+q$ items are required. Since σ' contains B items and $t \le q$ it follows that that

$$\sum_{j=1}^{t} f(b_j) + q \le B. \tag{2}$$

Since by (1) the sum $\sum_{j=1}^{l} f(b_j)$ is a multiple of B and $q \ge 1$, the only possibility that the left hand side of (2) can be bounded from above by B is that $\sum_{j=1}^{t} f(b_j) = 0$. However, this is a contradiction to $f(b_j) \ge 1$ for $j = 1, \dots, t$. □

As a consequence of the previous lemma we obtain the following bound on the competitive ratio of ONEBIN.

Theorem 4.2. *Algorithm* ONEBIN *is c-competitive for the* OLBCP$_{B,q}$ *with* $c = \min\{2q - 1, B\}$.

Proof. Let $\sigma = r_1, \dots, r_m$ be any request sequence and suppose that ONEBIN$(\sigma) = w$. Let $\sigma' = r_{iB+1}, \dots, r_{(i+1)B}$ of σ be the subsequence on which ONEBIN gets w different colors. Clearly, σ' contains items with exactly w colors. By Lemma 4.1 OPT distributes the items of σ' into at most $\min\{2q-1, B\}$ different bins. Hence, one of those bins must be filled with at least $\frac{w}{\min\{2q-1,B\}}$ colors. □

The competitive ratio proved in the previous theorem is tight as the following example shows. Let $B \ge 2q - 1$. First we give $(q-1)B$ items, after which by definition ONEBIN has only empty bins. The items have q different colors, every color but one occurs $B - 1$ times, one color occurs only $q - 1$ times. The adversary assigns all items of the same color to the same bin, using one color per bin. After this, q items with all the different colors used before are requested. The adversary can now close $q - 1$ bins, still using only one color per bin. ONEBIN ends up with q different colors in its bin. Then $q - 1$ items with new (previously unused) colors are given. The adversary can assign every item to an empty bin, thus still having only one different color per bin, while ONEBIN puts these items in the bin where already q different colors where present.

5. A General Lower Bound for Deterministic Algorithms

In this section we prove a general lower bound on the competitive ratio of any deterministic online algorithm for the OLBCP. We establish a lemma which immediately leads to the desired lower bound but which is even more powerful. In particular, this lemma will allow us to derive essentially the same lower bound for randomized algorithms in Section 6.

In the sequel we will have to refer to the “state” of (the bins managed by) an algorithm ALG after processing a prefix of a request sequence σ. To this end we introduce the notion of a $\mathcal{C}$*-configuration*.

Definition 5.1 ($\mathcal{C}$-configuration)**.** Let $\mathcal{C}$ a set of colors. A $\mathcal{C}$*-configuration* is a packing of items with colors from $\mathcal{C}$ into at most q bins. More formally, a $\mathcal{C}$-configuration can be defined as a mapping $K\colon \{1, \dots, q\} \to \mathcal{S}_{\le B}$, where

$$\mathcal{S}_{\le B} := \{\, S : S \text{ is a multiset over } \mathcal{C} \text{ containing at most } B \text{ elements from } \mathcal{S} \,\}$$

with the interpretation that $K(j)$ is the multiset of colors contained in bin j. We omit the reference to the set $\mathcal{C}$ if it is clear from the context.

Lemma 5.2. *Let $B, q, s \in \mathbb{N}$ such that $s \geq 1$ and the inequality $B/q \geq s-1$ holds. There exists a finite set $\mathcal{C}$ of colors and a constant $L \in \mathbb{N}$ with the following property. For any deterministic algorithm* ALG *and any $\mathcal{C}$-configuration K there exists an input sequence $\sigma_{\mathsf{ALG},K}$ of* $\mathrm{OLBCP}_{B,q}$ *such that*

(i) The sequence $\sigma_{\mathsf{ALG},K}$ uses only colors from $\mathcal{C}$ and $|\sigma_{\mathsf{ALG},K}| \leq L$, that is, $\sigma_{\mathsf{ALG},K}$ consists of at most L requests.
(ii) If ALG *starts with initial $\mathcal{C}$-configuration K then $\mathsf{ALG}(\sigma_{\mathsf{ALG},K}) \geq (s-1)q$.*
(iii) If OPT *starts with the empty configuration (i.e., all bins are empty), then $\mathsf{OPT}(\sigma_{\mathsf{ALG},K}) \leq s$. Additionally,* OPT *can process the sequence in such a way that at the end again the empty configuration is attained.*

Moreover, all of the above statements remain true even in the case that the online algorithm is allowed to use $q' \geq q$ bins instead of q (while the offline adversary still only uses q bins). In this case, the constants $|\mathcal{C}|$ and K depend only on q' but not on the particular algorithm ALG.

Proof. Let $\mathcal{C} = \{c_1, \dots, c_{(s-1)^2q^2q'}\}$ be a set of $(s-1)^2q^2q'$ colors and ALG be any deterministic online algorithm which starts with some initial $\mathcal{C}$-configuration K.

The construction of the request sequence $\sigma_{\mathsf{ALG},K}$ works in *phases*, where at the beginning of each phase the offline adversary has all bins empty. During the run of the request sequence, a subset of the currently open bins of ALG will be *marked*. We will denote by P_k the subset of marked bins at the beginning of Phase k. $P_1 = \emptyset$ and during some Phase M, one bin in P_M will contain at least $(s-1)q$ colors. In order to assure that this goal can in principle be achieved, we keep the invariant that each bin $b \in P_k$ has the property that the number of different colors in b plus the free space in b is at least $(s-1)q$. In other words, each bin $b \in P_k$ could potentially still be forced to contain at least $(s-1)q$ different colors. For technical reasons, P_k is only a subset of the bins with this property.

For bin j of ALG we denote by $n(j)$ the number of different colors currently in bin j and by $f(j)$ the space left in bin j. Then every bin $j \in P_k$ satisfies $n(j) + f(j) \geq (s-1)$. By $\min P_k := \min_{j \in P_k} n(j)$ we denote the minimum number of colors in a bin from P_k.

We now describe Phase k with $1 \leq k \leq q(s-1)q'$. The adversary selects a set of $(s-1)q$ new colors $C_k = \{c_1, \dots, c_{(s-1)q}\}$ from $\mathcal{C}$ not used in any phase before and starts to present one item of each color in the order

$$c_1, c_2, \dots, c_{(s-1)q}, c_1, c_2, \dots, c_{(s-1)q}, c_1, c_2, \dots \tag{3}$$

until one of the following cases appears:

Case 1: ALG puts an item into a bin $p \in P_k$. In this case we let $Q := P_k \setminus \{\, j \in P_k : n(j) < n(p) \,\}$, that is, we remove all bins from P_k which have less than $n(p)$ colors. Notice that $\min_{j \in Q} n(j) > \min P_k$, since the number of different colors in bin p increases.

Case 2: ALG puts an item into some bin $j \notin P_k$ which satisfies

$$n(j) + f(j) \geq (s-1)q. \tag{4}$$

In this case we set $Q := P_k \cup \{j\}$ (we tentatively add bin j to the set P_k).

Notice that after a finite number of requests one of these two cases must occur: Let $b_1, \dots, b_t$ be the set of currently open bins of ALG. If ALG never puts an item into a bin from P_k then at some point all bins of $\{b_1, \dots, b_t\} \setminus P_k$ are filled and a new bin, say bin j, must be opened by ALG by putting the new item into bin j. But at this moment bin j satisfies satisfies $n(j) = 1$, $f(j) = B - 1$ and hence $n(j) + f(j) = B \geq (s-1)q$ which gives (4).

Since the adversary started the phase with all bins empty and during the current phase we have given no more than $(s-1)q$ colors, the adversary can assign the items to bins such that no bin contains more than $s-1$ different colors (we will describe below how this is done precisely). Notice that due to our stopping criterions from above (case 1 and case 2) it might be the case that in fact so far we have presented less than $(s-1)q$ colors.

In the sequel we imagine that each currently open bin of the adversary has an index x, where $1 \leq x \leq q$. Let $\beta\colon C_k \to \{1,\dots,q\}$ be any mapping of the colors from C_k to the offline bin index such that $|\beta^{-1}(\{x\})| \leq s-1$ for $j = 1,\dots,q$. We imagine color c_r to "belong" to the bin with index $\beta(c_r)$ even if no item of this color has been presented (yet). For those items presented already in Phase k, each item with color c_r goes into the currently open bin with index $\beta(c_r)$. If there is no open bin with index $\beta(c_r)$ when the item arrives a new bin with index $\beta(c_r)$ is opened by the adversary to accommodate the item.

Our goal now is to clear all open offline bins so that we can start a new phase. During our clearing loop the offline bin with index x might be closed and replaced by an empty bin multiple times. Each time a bin with index x is replaced by an empty bin, the new bin will also have index x. The bin with index x receives a color not in $\beta^{-1}(\{x\})$ at most once, ensuring that the optimum offline cost still remains bounded from above by s. The clearing loop works as follows:

(1) (Start of clearing loop iteration) Choose a color $c^* \in C_k$ which is not contained in any bin from Q. If there is no such color, goto the "good end" of the clearing loop (Step 4).

(2) Let $F \leq qB$ denote the current total empty space in the open offline bins. Present items of color c^* until one of the following things happens:

Case (a): At some point in time ALG puts the ℓth item with color c^* into a bin $j \in Q$ where $1 \leq \ell < F$. Notice that the number of different colors in j increases. Let

$$Q' := Q \setminus \{b \in Q : n(b) < n(j)\},$$

in other words, we remove all bins b from Q which currently have less than $n(j)$ colors. This guarantees that

$$\min_{b \in Q'} n(b) > \min_{b \in Q} n(b) \geq \min P_k. \tag{5}$$

The adversary puts all t items of color c^* into bins with index $\beta(c^*)$. Notice that during this process the open bin with index $\beta(c^*)$ might be filled up and replaced by a new empty bin with the same index.

Set $Q := Q'$ and go to the start of the next clearing loop iteration (Step 1). Notice that the number of colors from C_k which are contained in Q decreases by one, but $\min_{b \in Q} n(b)$ increases.

Case (b): F items of color c^* have been presented, but ALG has not put any of these items into a bin from Q.

In this case, the offline adversary processes these items differently from Case (a): The F items of color c^* are used to fill up the exactly F empty places in all currently open offline bins. Since up to this point, each offline bin with index x had received colors only from the $s-1$ element set $\beta^{-1}(\{x\})$, it follows that no offline bin has contained more than s different colors.

We close the clearing loop by proceeding as specified at the "standard end" (Step 3).

(3) (Standard end of clearing loop iteration)

In case we have reached this step, we are in the situation that all offline bins have been cleared (we can originate only from Case (b) above). We set $P_{k+1} := Q$ and end the clearing loop and the current Phase k.

(4) (Good end of clearing loop iteration)

We have reached the point that all colors from C_k are contained in a bin from Q. Before the first iteration, exactly one color from C_k was contained in Q. The number of colors from C_k which are contained in bins from Q can only increase by one (which is in Case (a) above) if $\min_{b \in Q} n(b)$ increases. Hence, if all colors from C_k are contained in bins from Q, $\min_{b \in Q} n(v)$ must have increased $(s -$

$1)q-1$ times, which implies $\min_{b\in Q} n(b) = (s-1)q$. In other words, one of ALG's bins in Q contains at least $(s-1)q$ different colors.

The only thing left to do is append a suitable suffix to our sequence constructed so far such that all open offline bins are closed. Clearly this can be done without increasing the offline-cost.

In case the clearing loop finished with a "good end" we have achieved our goal of constructing a sufficiently bad sequence for ALG. What happens if the clearing loop finishes with a "standard end"?

Claim 5.3. *If Phase k completes with a "standard end", then $\min P_{k+1} > \min P_k$ or $|P_{k+1}| > |P_k|$.*

Before we prove Claim 5.3, let us show how this claim implies the result of the lemma. Since the case $|P_{k+1}| > |P_k|$ can happen at most q' times, it follows that after at most q' phases $\min P_k$ must increase. On the other hand, since $\min P_k$ never decreases by our construction and the offline costs remain bounded from above by s, after at most $q(s-1)q'$ phases we must be in the situation that $\min P_k \geq (s-1)q$, which implies a "good end". Since in each phase at most $(s-1)q$ new colors are used, it follows that our initial set $\mathcal{C}$ of $(s-1)^2q^2q'$ colors suffices to construct the sequence $\sigma_{\mathsf{ALG},K}$. Clearly, the length of $\sigma_{\mathsf{ALG},K}$ can be bounded by a constant L independent of ALG and K.

Proof of Claim 5.3. Suppose that the sequence (3) at the beginning of the phase was ended because Case 1 occurred, i.e., ALG put one of the new items into a bin from P_k. In this case $\min_{b\in Q} n(b) > \min P_k$. Since during the clearing loop $\min_{b\in Q} n(b)$ can never decrease and P_{k+1} is initialized with the result of Q at the "standard end" of the clearing loop, the claim follows.

The remaining case is that the sequence (3) was ended because of a Case 2-situation. Then $|Q| = |P_k \cup \{j\}|$ for some $j \notin P_k$ and hence $|Q| > |P_k|$. During the clearing loop Q can only decrease in size if $\min_{i\in Q} n(i)$ increases. It follows that either $|P_{k+1}| = |P_k|+1$ or $\min P_{k+1} > \min P_k$ which is what we claimed. □

This completes the proof of the lemma. □

As an immediate consequence of Lemma 5.2 we obtain the following lower bound result for the competitive ratio of any deterministic algorithm:

Theorem 5.4. *Let $B, q, s \in \mathbb{N}$ such that $s \geq 1$ and the inequality $B/q \geq s-1$ holds. No deterministic algorithm for $\text{OLBCP}_{B,q}$ can achieve a competitive ratio less than $(s-1)/s \cdot q$. Hence, the competitive ratio of any deterministic algorithm for fixed B and q is at least $\left(1-\frac{q}{B+q}\right) q$. In particular, for the general case with no restrictions on the relation of the capacity B to the number of bins q, there can be no deterministic algorithm for $\text{OLBCP}_{B,q}$ that achieves a competitive ratio less than q.*

All of the above claims remain valid, even if the online algorithm is allowed to use an arbitrary number $q' \geq q$ of open bins. □

6. A General Lower Bound for Randomized Algorithms

In this section we show lower bounds for the competitive ratio of any randomized algorithm against an oblivious adversary for $\text{OLBCP}_{B,q}$. The basic method for deriving such a lower bound is Yao's principle (see also [BEY98, MR95]). Let X be a probability distribution over input sequences $\Sigma = \{\, \sigma_x : x \in \mathcal{X} \,\}$. We denote the *expected cost* of the deterministic algorithm ALG according to the distribution X on Σ by $\mathbb{E}_X[\mathsf{ALG}(\sigma_x)]$. Yao's principle can now be stated as follows.

Theorem 6.1 (Yao's principle). *Let $\{\, \mathsf{ALG}_y : y \in \mathcal{Y} \,\}$ denote the set of deterministic online algorithms for an online minimization problem. If X is a probability distribution over input* *sequences* $\{ \sigma_x : x \in \mathcal{X} \}$ *such that*

$$\inf_{y\in\mathcal{Y}} \mathbb{E}_X \left[\mathsf{ALG}_y(\sigma_x)\right] \geq \bar{c}\,\mathbb{E}_X \left[\mathsf{OPT}(\sigma_x)\right]. \tag{6}$$

for some real number $\bar{c} \geq 1$, *then* $\bar{c}$ *is a lower bound on the competitive ratio of any randomized algorithm against an oblivious adversary.* □

Theorem 6.2. *Let* $B, q, s \in \mathbb{N}$ *such that* $s \geq 1$ *and the inequality* $B/q \geq s - 1$ *holds. Then no randomized algorithm for* $\mathrm{OLBCP}_{B,q}$ *can achieve a competitive ratio less than* $(s-1)/s \cdot q$ *against an oblivious adversary.*

In particular for fixed B *and* q*, the competitive ratio against an oblivious adversary is at least* $\left(1 - \frac{q}{B+q}\right) q$*.*

All of the above claims remain valid, even if the online algorithm is allowed to use an arbitrary number $q' \geq q$ *of open bins.*

Proof. Let $\mathcal{A} := \{ \mathsf{ALG}_y : y \in \mathcal{Y} \}$ the set of deterministic algorithms for the Uniform Bin Packing Problem with capacity B and q open bins. We will show that there is a probability distribution X over a certain set of request sequences $\{ \sigma_x : x \in \mathcal{X} \}$ such that for any $\mathsf{ALG}_y \in \mathcal{A}$

$$\mathbb{E}_X \left[\mathsf{ALG}_y(\sigma_x)\right] \geq (s-1)q,$$

and, moreover,

$$\mathbb{E}_X \left[\mathsf{OPT}(\sigma_x)\right] \leq s.$$

The claim of the theorem then follows by Yao's principle.

Let us recall the essence of Lemma 5.2. The lemma establishes the existence of a finite color set $\mathcal{C}$ and a constant L such that for a fixed configuration K any deterministic algorithm can be "fooled" by one of at most $|\mathcal{C}|^L$ sequences. Since there are no more than $|\mathcal{C}|^{qB}$ configurations, a *fixed finite* set of at most $N := |\mathcal{C}|^{L+qB}$ sequences $\Sigma = \{\sigma_1, \ldots, \sigma_N\}$ suffices to "fool" *any* deterministic algorithm provided the initial configuration is known.

Let X be a probability distribution over the set of finite request sequences

$$\{ \sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k} : k \in \mathbb{N}, 1 \leq i_j \leq N \}$$

such that σ_{i_j} is chosen from Σ uniformly and independently of all previous subsequences $\sigma_{i_1}, \ldots, \sigma_{i_{j-1}}$. We call subsequence σ_{i_k} the *kth phase*.

Let $\mathsf{ALG}_y \in \mathcal{A}$ be arbitrary. Define ϵ_k by

$$\epsilon_k := \mathrm{Pr}_X \left[\mathsf{ALG}_y \text{ has one bin with at least } (s-1)q \text{ colors during Phase } k\right]. \tag{7}$$

The probability that ALG_y has one bin with at least $(s-1)q$ colors on any given phase is at least $1/N$, whence $\epsilon_k \geq 1/N$ for all k. Let

$$p_k := \mathrm{Pr}_X \left[\mathsf{ALG}_y(\sigma_{i_1} \ldots \sigma_{i_{k-1}} \sigma_{i_k}) \geq (s-1)q\right]. \tag{8}$$

Then the probabilities p_k satisfy the following recursion:

$$p_0 = 0 \tag{9}$$

$$p_k = p_{k-1} + (1 - p_{k-1})\epsilon_k \tag{10}$$

The first term in (10) corresponds to the probability that ALG_y has already cost at least $(s-1)q$ after Phase $k-1$, the second term accounts for the probability that this is not the case but cost at least $(s-1)q$ is achieved in Phase k. By construction of X, these events are independent. Since $\epsilon_k \geq 1/N$ we get that

$$p_k \geq p_{k-1} + (1 - p_{k-1})/N. \tag{11}$$

It is easy to see that any sequence of real numbers $p_k \in [0, 1]$ satisfying (9) and (11) must converge to 1. Hence, also the expected cost $\mathbb{E}_X \left[\mathsf{ALG}_y(\sigma_x)\right]$ converges to $(s-1)q$. On the other hand, the offline costs remain bounded by s by the choice of the σ_{i_j} according to Lemma 5.2. □

7. CONCLUSIONS

We have studied the online bin coloring problem OLBCP, which was motivated by applications in a robotized assembly environment. The investigation of the problem from a competitive analysis point of view revealed a number of odds. A natural greedy-type strategy (GREEDYFIT) achieves a competitive ratio strictly worse than arguably the most stupid algorithm (ONEBIN). Moreover, no algorithm can be substantially better than the trivial strategy (ONEBIN). Even more surprising, neither randomization nor "resource augmentation" helps to overcome the $\Omega(q)$ lower bound on the competitive ratio (see [PK95, PS+97] for successful applications to scheduling problems) can help to overcome the $\Omega(q)$ lower bound on the competitive ratio. Intuitively, the strategy GREEDYFIT should perform well "on average" (which we could sort of confirm by preliminary experiments with random data).

An open problem remains the existence of a deterministic (or randomized) algorithm which achieves a competitive ratio of q (matching the lower bound of Theorems 5.4 and 6.2). However, the most challenging issue raised by our work seems to be an investigation of OLBCP from an average-case analysis point of view.

REFERENCES

[AG+98] N. Ascheuer, M. Grötschel, S. O. Krumke, and J. Rambau, *Combinatorial online optimization*, Proceedings of the International Conference of Operations Research (OR'98), Springer, 1998, pp. 21–37.

[BEY98] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*, Cambridge University Press, 1998.

[CGJ97] E. G. Coffman, M. R. Garey, and D. S. Johnson, *Approximation algorithms for bin packing: a survey*, In Hochbaum [Hoc97].

[CW98] J. Csirik and G. J. Woeginger, *On-line packing and covering problems*, In Fiat and Woeginger [FW98].

[FW98] A. Fiat and G. J. Woeginger (eds.), *Online algorithms: The state of the art*, Lecture Notes in Computer Science, vol. 1442, Springer, 1998.

[Hoc97] D. S. Hochbaum (ed.), *Approximation algorithms for* NP*-hard problems*, PWS Publishing Company, 20 Park Plaza, Boston, MA 02116–4324, 1997.

[MR95] R. Motwani and P. Raghavan, *Randomized algorithms*, Cambridge University Press, 1995.

[PK95] K. Pruhs and B. Kalyanasundaram, *Speed is as powerful as clairvoyance*, Proceedings of the 36th Annual IEEE Symposium on the Foundations of Computer Science, 1995, pp. 214–221.

[PS+97] C. Phillips, C. Stein, E. Torng, and J. Wein, *Optimal time-critical scheduling via resource augmentation*, Proceedings of the 29th Annual ACM Symposium on the Theory of Computing, 1997, pp. 140–149.

SVEN O. KRUMKE, KONRAD-ZUSE-ZENTRUM FÜR INFORMATIONSTECHNIK BERLIN, DEPARTMENT OPTIMIZATION, TAKUSTR. 7, D-14195 BERLIN-DAHLEM, GERMANY.
E-mail address: `krumke@zib.de`
URL: `http://www.zib.de/krumke`

WILLEM E. DE PAEPE, FACULTY OF TECHNOLOGY MANAGEMENT, TECHNICAL UNIVERSITY OF EINDHOVEN, P. O. BOX 513, 5600MB EINDHOVEN, THE NETHERLANDS.
E-mail address: `w.e.d.paepe@tm.tue.nl`

JÖRG RAMBAU, KONRAD-ZUSE-ZENTRUM FÜR INFORMATIONSTECHNIK BERLIN, DEPARTMENT OPTIMIZATION, TAKUSTR. 7, D-14195 BERLIN-DAHLEM, GERMANY.
E-mail address: `rambau@zib.de`

LEEN STOUGIE, COMBINATORIAL OPTIMIZATION GROUP, TECHNICAL UNIVERSITY OF EINDHOVEN, P. O. BOX 513, 5600MB EINDHOVEN, THE NETHERLANDS.
E-mail address: `leen@win.tue.nl`