Parallel Online Algorithms for the Bin Packing Problem

Sándor P. Fekete, Jonas Grosse-Holz, Phillip Keldenich, and Arne Schmidt(✉)

Department of Computer Science, TU Braunschweig, Braunschweig, Germany
{s.fekete,j.grosse-holz,p.keldenich,arne.schmidt}@tu-bs.de

Abstract. We study *parallel* online algorithms: For some fixed integer k, a collective of k parallel processes that perform online decisions on the same sequence of events forms a *k-copy algorithm*. For any given time and input sequence, the overall performance is determined by the best of the k individual total results. Problems of this type have been considered for online makespan minimization; they are also related to optimization with *advice* on future events, i.e., a number of bits available in advance.

We develop Predictive Harmonic$_3$ (PH3), a relatively simple family of k-copy algorithms for the online Bin Packing Problem, whose joint competitive factor converges to 1.5 for increasing k. In particular, we show that $k = 6$ suffices to guarantee a factor of 1.5714 for PH3, which is better than 1.57829, the performance of the best known 1-copy algorithm Advanced Harmonic, while $k = 11$ suffices to achieve a factor of 1.5406, beating the known lower bound of 1.54278 for a single online algorithm. In the context of online optimization with advice, our approach implies that 4 bits suffice to achieve a factor better than this bound of 1.54278, which is considerably less than the previous bound of 15 bits.

Keywords: Online algorithms · Bin packing · Competitive analysis

1 Introduction

When dealing with unknown future events, optimization with incomplete information typically considers the competitive factor of an online algorithm as its performance measure; the objective becomes to develop a single strategy that performs reasonably well against the worst case. This focus on just *one* option is more restrictive than hedging strategies in a wide variety of other scientific and application fields; these typically make use of *several* parallel choices, thereby increasing the chance that one of them will yield satisfactory results. Examples include scenarios from biology, where a large and diverse progeny increases the odds of surviving offspring; finance and insurance, where a suitable combination of investment strategies is employed to balance a portfolio against extreme losses;

A full version is available on arxiv.org [10].

Phillip Keldenich was partially supported by DFG grant FE407/17-2 as part of the Research Group FOR 1800, "Controlling Concurrent Change".

E. Bampis and N. Megow (Eds.): WAOA 2019, LNCS 11926, pp. 106–119, 2020.
https://doi.org/10.1007/978-3-030-39479-0_8

and engineering, where redundancy is used to protect against catastrophic failure, either on individual components (such as parts in a machine) or on whole systems (such as automata in a robot swarm or spacecraft in a group of satellites), where it suffices that just one machine delivers a good outcome.

In this paper, we consider such *parallel* online strategies: Instead of making a single sequence of decisions, we consider k parallel processes for some fixed integer k, which we call a *k-copy algorithm*; the objective is to make the best of these k outcomes as good as possible, even in the worst case. We demonstrate the potential of this approach for the well-studied Bin Packing Problem, for which it is known that no single deterministic online algorithm can achieve a competitive factor below 1.5401.

1.1 Our Results

We define a family of k-copy algorithms for the online Bin Packing Problem, called Predictive Harmonic$_3$ (PH3), whose asymptotic competitive ratio converges to 1.5 for large k. We show that $k = 6$ suffices to guarantee a factor of 1.5714, which is better than 1.57829, the performance of the best known 1-copy algorithm Advanced Harmonic [3]. Moreover, $k = 11$ suffices to achieve a competitive ratio of 1.5406 beating the known lower bound of 1.54278 for a 1-copy algorithm [4]. In the context of online optimization with advice, our approach implies that 4 bits suffice to achieve less than 1.5401, which is considerably less than the previous bound of 16 bits of RedBlue by Angelopoulos et al. [2]; in fact, for $k = 16$ (corresponding to four bits of advice) PH3 achieves a ratio of 1.5305, compared to 3.3750 for RedBlue, while $k = 65,536$ (corresponding to 16 bits of advice) yields a factor of 1.5001 for PH3, but 1.5293 for RedBlue.

1.2 Related Work on Online Bin Packing

There is a wide range of online algorithms for bin packing. The Next Fit algorithm [9] achieves a competitive ratio of 2, whereas "Almost Any Fit" algorithms [13] like First Fit or Best Fit achieve competitive ratios of 1.7.

An important online bin packing algorithm is Harmonic$_M$, introduced by Lee and Lee [15], which achieves a competitive ratio of less than 1.692 for $M \to \infty$. Based on Harmonic$_M$, Son Of Harmonic by Heydrich and van Stee [12] achieves a competitive ratio of 1.5816. The currently best known algorithm is Advanced Harmonic, which achieves a competitive ratio of 1.57829 [3].

For lower bounds, Yao [20] established a value of 3/2 that was later improved to 1.536, independently by Brown [8] and by Liang [16]. Using a generalization of their methods, van Vliet [19] proved a lower bound of 1.5401. Balogh et al. [4] improved the lower bound to 1.54278.

1.3 Related Work on Online Bin Packing with Advice

In the context of online algorithms with advice, Boyar et al. [7] showed that an online algorithm with $n\lceil \log(OPT(I))\rceil$ bits of advice is sufficient and that at least $(n - 2OPT(I)) \cdot \log(OPT(I))$ bits of advice are necessary to achieve optimality. In the same paper, they presented an online bin packing algorithm, namely ReserveCritical, with $O(\log(n)) + o(\log(n))$ bits of advice that is 1.5-competitive and an algorithm with $2n + o(n)$ bits of advice that is $\frac{4}{3}$-competitive. Zhao and Shen [21] developed an algorithm using $3n + o(n)$ bits of advice achieving a competitive ratio of $\frac{5}{4}\text{OPT} + 2$. Renault et al. [18] developed an $(1 + \varepsilon)$-competitive algorithm using $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ bits of advice per request.

Based on ReserveCritical, Angelopoulos et al. [2] developed the algorithm RedBlue with constant advice that is 1.5-competitive. Their second algorithm achieves a competitive ratio of $1.47012 + \varepsilon$ with finite advice that is exponentially dependent of ε. However, to beat the competitive ratio of 1.5 already an enormous amount of advice is needed, which makes the algorithm impractical.

In terms of lower bounds, Boyar et al. [7] proved that no competitive ratio better than 9/8 can be reached by any algorithm that uses sub-linear advice. Angelopoulos et al. [2] improved this bound to 7/6.

1.4 Related Work on Parallel Online Algorithms

Parallel algorithms have already been considered in the field of online algorithms with advice. Boyar et al. [6] presented an algorithm for the online list update problem, making use of 2 bits of advice to choose one out of three algorithms. This algorithm achieves a competitive ratio of 5/3, beating the lower bound for conventional online algorithms of 2. A practical application of this algorithm was shown by Kamali and Ortiz [14], who applied it in the Burrows-Wheeler transform compression. More work on parallel online algorithms include parallel scheduling [1], finding independent sets [11] and the "multiple-cow" version of the linear search problem [17].

While online algorithms with advice mostly focus on the amount of advice to allow classification of online algorithms and problems, k-copy online algorithms focus on small finite values for k and thus small finite amounts of advice, with more emphasis on practical application. The perspective on different algorithms running in parallel instead of abstract arbitrary information facilitates finer optimization in some cases.

Also, when considering online algorithms with advice, the number of algorithms can only be doubled by increasing the amount of advice by one bit. The perspective of k-copy algorithms allows arbitrary $k \in \mathbb{N}$ for the number of algorithms.

2 Preliminaries

2.1 *k*-Copy Online Algorithms

In this paper, we consider k online algorithms $A_1, \ldots, A_k$, each of them processing the same input list I in parallel. We call the set $\mathcal{A} := \{A_1, \ldots, A_k\}$ a *k-copy online algorithm*.

For an input list I and an online algorithm A, let $A(I)$ denote the number of bins used by A and $\textsc{opt}(I)$ denote the number of bins used in an optimal offline solution. The absolute competitive ratio $R_{\mathcal{A}}$ for a k-copy online algorithm $\mathcal{A}$ is defined as

$$R_{\mathcal{A}} = \sup_{I} \left\{ \frac{\min_{A \in \mathcal{A}} A(I)}{\textsc{opt}(I)} \right\}.$$

The asymptotic competitive ratio $R_{\mathcal{A}}^{\infty}$ for algorithm $\mathcal{A}$ is defined as

$$R_{\mathcal{A}}^{\infty} = \lim_{n \to \infty} \sup_{I} \left\{ \frac{\min_{A \in \mathcal{A}} A(I)}{\textsc{opt}(I)} \;\middle|\; \textsc{opt}(I) = n \right\}$$

As already stated by Boyar et al. [5], any k-copy online algorithm can be converted into an online algorithm with advice, and vice versa.

Lemma 1. *Any k-copy online algorithm can be converted into an online algorithm with $l = \lceil \log_2(k) \rceil$ bits of advice that achieves the same competitive ratio. Conversely, any online algorithm with $l \in \mathbb{N}$ bits of advice can be converted into a k-copy online algorithm without advice with $k = 2^l$ that achieves the same competitive ratio.*

Proof. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ be a k-copy algorithm. Construct the online algorithm A' that gets a value $i \in \{1, 2, \ldots, k\}$ as advice, specifying the index i of the algorithm $A_i \in \mathcal{A}$ that performs best on the given input sequence. The value i can be encoded using $\lceil \log_2(k) \rceil$ bits. A' then behaves like A_i and thus achieves the same competitive ratio as $\mathcal{A}$.

Let A be an online algorithm that gets $l \in \mathbb{N}$ bits of advice. Construct the online k-copy algorithm $\mathcal{A}'$ with $k = 2^l$ algorithms $A_i, i \in \{1, 2, \ldots, k\}$. For each $i \in \{1, 2, \ldots, k\}$, the algorithm A_i behaves like A given i encoded in binary as advice. As the values $i \in \{1, 2, \ldots, k\}$ cover every possible configuration of the advice bits, for any advice given to A, there is an algorithm $A_i \in \mathcal{A}'$, that assumes this advice. Accordingly, there is an algorithm $A_i \in \mathcal{A}'$, that performs as well as A, i.e., the best algorithm $A_j \in \mathcal{A}$ that performs at least as well as A. Thus, $\mathcal{A}'$ performs at least as well as A.

2.2 Bin Packing

In the online version of bin packing, we are given a list of items $I := \langle a_1, \ldots, a_n \rangle$ with $a_i \in (0, 1]$ for $i \in \{1, \ldots, n\}$. These items must be packed by an algorithm, one at a time, without any information on subsequent items and without the possibility to change previous decisions. The goal is to pack all items into a minimum number of bins with unit capacity.

Definition 1 (Item size).
Let $S = \left[0, \frac{1}{3}\right]$, $M = \left(\frac{1}{3}, \frac{1}{2}\right]$, $L = \left(\frac{1}{2}, \frac{2}{3}\right)$ and $XL = \left[\frac{2}{3}, 1\right]$. We call items in S small, *items in M* medium, *items in L* large *and items in XL* extra large. *For a list $I = \langle a_1, a_2, \ldots a_n \rangle$, the set of items $Set(I) \cap XL$ is noted as I_{XL} for improved readability. The subsets I_L, I_M and I_S are used analogously.*

Definition 2 (Size function). *Let S be a set (or list) of items. Then,* $\text{size}(S) := \sum_{i \in S} i$. *For a bin b, we refer to* $\text{size}(b)$ *as the size of the bin, i.e., the sum of items already packed in b.*

Definition 3 (Sub-bins).
Given a bin b, it can be split into two parts b_1 and b_2, such that the sum of their capacities is equal to the capacity of b. We refer to b_1 and b_2 as sub-bins. We call a sub-bin with capacity C a C-sub-bin.

As sub-bins are not packed with an amount larger than their capacity, each sub-bin can be packed independently from the other.

3 Predictive Harmonic$_3$

Now we introduce the algorithm Predictive Harmonic$_3$ (PH3). Although developed independently, it bears many similarities to ReserveCritical and RedBlue. PH3 uses the same classifications as the other two algorithms and tries to pack all large items with small items, such that the corresponding bins are packed to a level of at least 2/3. However, in contrast to RedBlue, the information needed by PH3 does not depend on the result of ReserveCritical, but only on the number and size of certain item types, and can be calculated in linear time.

The main idea of PH3 is to guess the ratio of how many small items must be packed with large items to obtain a packing density of 2/3. Having multiple instances of PH3, every instance can guess a different ratio to get close to a competitive ratio of 1.5.

Algorithm 1 Predictive Harmonic$_3$. Given a list $I = \langle a_1, a_2, \ldots, a_n \rangle$ of items $a_i \in (0, 1], i \in 1, \ldots, n$, and a ratio $r_L \in [0, 1]$, the algorithm packs the items as follows:

- Extra large items are packed into individual bins. These bins are called XL-bins, the set of all XL-bins is called B_{XL}.
- Large items are packed into individual bins. These bins are called L-bins, the set of all L-bins is called B_L. Furthermore, we split each L-bin into a $\frac{2}{3}$-sub-bin (for large items) and a $\frac{1}{3}$-sub-bin (for small items).
- Medium items are packed into separate bins together with other medium items (note that at most two of them fit into one bin). These bins are called M-bins, the set of all M-bins is called B_M.
- Small items are packed into a $\frac{1}{3}$-sub-bin of L-bins in a next fit manner, if the size of small items packed into L-bins is smaller than r_L times the total size of small items packed so far; otherwise we pack the small item into S-bins.

3.1 Competitive Ratio

Using simple bounds for an optimal solution and performing a case analysis, we can prove the following theorem. Due to space constraints, the proof can be found in the full version [10].

Theorem 2. *Let* $r_L^* = min\left\{\frac{|I_L|}{6\ \text{size}(I_S)}, 1\right\}$[1] *and* $\delta = r_L - r_L^*$. *PH3 achieves the asymptotic competitive ratio*

$$R_{PH3}^{\infty} \leq \begin{cases} \dfrac{3}{2} + min\left\{\dfrac{1}{4r_L^*}, \dfrac{3}{6r_L^*+2}\right\}(-\delta) & \textit{for } \delta \leq 0 \\ \dfrac{3}{2} + min\left\{\dfrac{3}{4r_L^*}, \dfrac{9}{6r_L^*+2}\right\}\delta & \textit{for } \delta \geq 0. \end{cases}$$

3.2 Tightness

Theorem 3. *For any* $r_L, r_L^* \in [0,1]$*, the asymptotic competitive ratio given in Theorem 2 is tight.*

Proof sketch: Let $\langle a_1, a_2, \dots a_k\rangle \times n$ with $n \in \mathbb{N}$ denote n repetitions of the sequence $\langle a_1, a_2, \dots a_k\rangle$. Let I be a sequence consisting of concatenated sub-sequences I_S, I_M and I_L, where I_S is a sequence consisting of two interleaved sub-sequences I_{SL} and I_{LL}. With $N \in \mathbb{N}$ and $\varepsilon = 1/(12N+2)$, we define

$$\begin{aligned}
I_L &= \left(\frac{1}{2} + \frac{\varepsilon}{2}\right) \times n_L \text{ with } n_L = \lceil 4r_L^* N\rceil \\
I_M &= \left(\frac{1}{3} + \frac{\varepsilon}{2}\right) \times n_M \text{ with } n_M = \begin{cases} 0 & \text{for } r_L^* \leq 1/3 \\ \lfloor (6r_L^* - 2)N\rfloor & \text{for } r_L^* \geq 1/3 \end{cases} \\
I_{SS} &= \left(\frac{1}{3} - 2\varepsilon, \frac{1}{6} - \varepsilon, \frac{1}{6} - \varepsilon, 12\varepsilon\right) \times n_{SS} \text{ with } n_{SS} = \lceil n'_{SS}\rceil = \lceil (1-r_L)N\rceil \\
I_{SL} &= \left(\frac{1}{6} - \varepsilon, 3\varepsilon\right) \times n_{SL} \text{ with } n_{SL} = \lceil n'_{SL}\rceil = \lceil 4r_L N\rceil
\end{aligned}$$

The proof is based on a case analysis of which item appears next and in which bin this item is packed by PH3. Due to space constraints, a full proof can be found in the full version [10].

4 Parallel Predictive Harmonic$_3$

4.1 Competitive Ratio for PH3 as 1-Copy Online Algorithm

To optimize the performance for PH3 as a 1-copy algorithm, we determine the optimal value for r_L with respect to minimizing the asymptotic competitive ratio over all $r_L^* \in [0,1]$.

[1] The intuition of this value is that at least $1/2$ of each $1/3$-sub-bin must be filled to guarantee a packing density of $2/3$. Therefore, for $|I_L|$ bins, we have to fill up a total capacity of $\frac{|I_L|}{6}$ with small items.

Lemma 2 (Monotonicity of competitive ratio of PH3). *For any fixed $r_L \in [0,1]$, the competitive factor is monotonically decreasing for $r_L^* \in [0, r_L]$ and monotonically increasing for $r_L^* \in [r_L, 1]$.*

Proof. Assume r_L to be fixed. Let $r_{+,<}, r_{-,<} : [0, 1/3] \to \mathbb{R}$ and $r_{+,>}, r_{-,>} : [1/3, 1] \to \mathbb{R}$ with

$$\begin{aligned}
r_{-,<}(r_L^*) &= \frac{3}{2} + \frac{3}{6r_L^* + 2}(-\delta) && = R_{PH3}^{\infty} \text{ for } \delta \le 0, r_L^* \le \frac{1}{3} \\
r_{-,>}(r_L^*) &= \frac{3}{2} + \frac{1}{4r_L^*}(-\delta) && = R_{PH3}^{\infty} \text{ for } \delta \le 0, r_L^* \ge \frac{1}{3} \\
r_{+,<}(r_L^*) &= \frac{3}{2} + \frac{9}{6r_L^* + 2}\delta && = R_{PH3}^{\infty} \text{ for } \delta \ge 0, r_L^* \le \frac{1}{3} \\
r_{+,>}(r_L^*) &= \frac{3}{2} + \frac{3}{4r_L^*}\delta && = R_{PH3}^{\infty} \text{ for } \delta \ge 0, r_L^* \ge \frac{1}{3}
\end{aligned}$$

Consider the derivative of $r_{-,<}$ and $r_{-,>}$.

$$\begin{aligned}
\frac{\partial}{\partial r_L^*} r_{-,<}(r_L^*) &= \frac{\partial}{\partial r_L^*}\left(\frac{3}{2} + \frac{3}{6r_L^* + 2}(-\delta)\right) \\
&= \frac{\partial}{\partial r_L^*}\left(\frac{3(r_L^* - r_L)}{6r_L^* + 2}\right) \\
&= \frac{18r_L + 6}{(6r_L^* + 2)^2} \ge 0 \text{ for } 0 \le r_L \le r_L^* \le \frac{1}{3} \\
\frac{\partial}{\partial r_L^*} r_{-,>}(r_L^*) &= \frac{\partial}{\partial r_L^*}\left(\frac{3}{2} + \frac{1}{4r_L^*}(-\delta)\right) \\
&= \frac{\partial}{\partial r_L^*}\left(\frac{r_L^* - r_L}{4r_L^*}\right) \\
&= \frac{r_L}{4(r_L^*)^2} \ge 0 \text{ for } 0 \le r_L \le r_L^* \text{ and } \frac{1}{3} \le r_L^* \le 1
\end{aligned}$$

As the derivatives of $r_{-,<}$ and $r_{-,>}$ are both non-negative in their respective domains, they are both monotonically increasing. Because $r_{-,<}(\frac{1}{3}) = r_{-,>}(\frac{1}{3})$, we conclude that the competitive ratio is monotonically increasing for $r_L^* \in [r_L, 1]$.

Now consider the derivative of $r_{+,<}$ and $r_{+,>}$.

$$\begin{aligned}
\frac{\partial}{\partial r_L^*} r_{+,<}(r_L^*) &= \frac{\partial}{\partial r_L^*}\left(\frac{3}{2} + \frac{9}{6r_L^* + 2}\delta\right) \\
&= \frac{\partial}{\partial r_L^*}\left(\frac{9(r_L - r_L^*)}{6r_L^* + 2}\right) \\
&= \frac{-54r_L - 18}{(6r_L^* + 2)^2} \leq 0 \text{ for } r_L^* \leq r_L \leq 1 \text{ and } 0 \leq r_L^* \leq \frac{1}{3} \\
\frac{\partial}{\partial r_L^*} r_{+,>}(r_L^*) &= \frac{\partial}{\partial r_L^*}\left(\frac{3}{2} + \frac{3}{4r_L^*}\delta\right) \\
&= \frac{\partial}{\partial r_L^*}\left(\frac{3(r_L - r_L^*)}{4r_L^*}\right) \\
&= \frac{-3r_L}{4(r_L^*)^2} \leq 0 \text{ for } \frac{1}{3} \leq r_L^* \leq r_L \leq 1
\end{aligned}$$

As the derivatives of $r_{+,<}$ and $r_{+,>}$ are both non-positive in their respective domains, they are both monotonically decreasing. Because $r_{+,<}(\frac{1}{3}) = r_{+,>}(\frac{1}{3})$, we conclude that the competitive ratio is monotonically decreasing for $r_L^* \in [0, r_L]$.

Because of Lemma 2, the competitive ratio does not decrease with r_L^* increasing for $\delta \leq 0$. Thus, as an upper bound on the competitive ratio for $\delta \leq 0$, only the competitive ratio for $r_L^* = 1$ has to be considered.

$$\begin{aligned}
R_{PH3}^{\infty} &\leq \frac{3}{2} + \frac{1}{4}(-\delta) \text{ for } \delta \leq 0 \\
&= \frac{3}{2} + \frac{1}{4}(1 - r_L) \\
&= \frac{7}{4} - \frac{r_L}{4}
\end{aligned}$$

For $\delta \geq 0$, the competitive ratio does not decrease with r_L^* decreasing. In this case, the competitive ratio for $r_L^* = 0$ is an upper bound on the competitive ratio.

$$\begin{aligned}
R_{PH3}^{\infty} &\leq \frac{3}{2} + \frac{9}{2}\delta \text{ for } \delta \geq 0 \\
&= \frac{3}{2} + \frac{9}{2}(r_L - 0) \\
&= \frac{3}{2} + \frac{9}{2}r_L
\end{aligned}$$

At the same time, these values are lower bounds on the overall competitive ratio. Given these bounds, this linear program can be formulated to minimize the competitive ratio:

$$\begin{aligned} \text{Minimize } & R^{\infty}_{PH3} \\ \text{Subject to } & R^{\infty}_{PH3} \geq \frac{7}{4} - \frac{r_L}{4} \\ & R^{\infty}_{PH3} \geq \frac{3}{2} + \frac{9}{2} r_L \\ & r_L \geq 0 \\ & r_L \leq 1 \end{aligned}$$

The optimal solution for this linear program is $r_L = 1/19$ and $R^{\infty}_{PH3} = 33/19 < 1.7369$. Figure 1 shows the asymptotic competitive ratio of PH3 over r^*_L for $r_L = 1/19$.

Compared to other known algorithms for online bin packing, PH3 is not a good choice for worst-case behavior. Among the classical algorithms, only NF and WF, both of which are 2-competitive, are worse than PH3. Any AAF algorithm achieves an asymptotic competitive ratio $R^{\infty}_{AAF} = 1.7$ [9] and thus performs slightly better than PH3. The best-performing online algorithm for bin packing currently known, Son Of Harmonic, is 1.5816-competitive and thus clearly superior to PH3 [12].

However, if we know in advance that r^*_L is restricted to some interval $I_r = [a, b] \subset [0, 1]$, the above argument can be used to prove a better competitive ratio.

4.2 Competitive Ratio for PH3 as *k*-Copy Online Algorithm

PH3's property of achieving a better competitive ratio for r^*_L being further restricted can be used to create a set of $k \in \mathbb{N}$ algorithms achieving a better competitive ratio. For this purpose, the interval $[0, 1]$ is split into k sub-intervals $I_1, \ldots, I_k \subset [0, 1]$ with $\cup_{i \in \{1,\ldots,k\}} I_i = [0, 1]$. Each interval I_i is covered by one instance of the algorithm PH3 A_i, such that A_i achieves a targeted competitive ratio $R \in (3/2, 33/19)$ for $r^*_L \in I_i$.

R is restricted to $(3/2, 33/19)$, because any competitive ratio above or equal to $33/19$ can be achieved with the instance of PH3 shown above, and k-copy PH3 cannot achieve a competitive ratio of $3/2$ or less with finitely many algorithms.

To calculate the number k of algorithms needed to achieve a given competitive ratio R, the following iterative approach can be used.

Let $\mathcal{A}$ be a set of algorithms. Initially, $\mathcal{A} := \emptyset$. We initialize our iterative approach with $i = 0$ and set $r^0_{max} = 0$. Then, while $r^i_{max} < 1$, we increase i by one and we compute three values r^i_{min}, r^i_L and r^i_{max}. With these three values we can define algorithm A_i for which r^i_L denotes the value of r_L, r^i_{min} denotes the minimal and r^i_{max} denotes the maximal value for r^*_L for which A_i is still R-competitive. By Lemma 2, A_i will be R-competitive for the interval $[r^i_{min}, r^i_{max}]$. All three values are computed as follows. We set $r^i_{min} = r^{i-1}_{max}$. Given r^i_{min}, r^i_L can be computed:

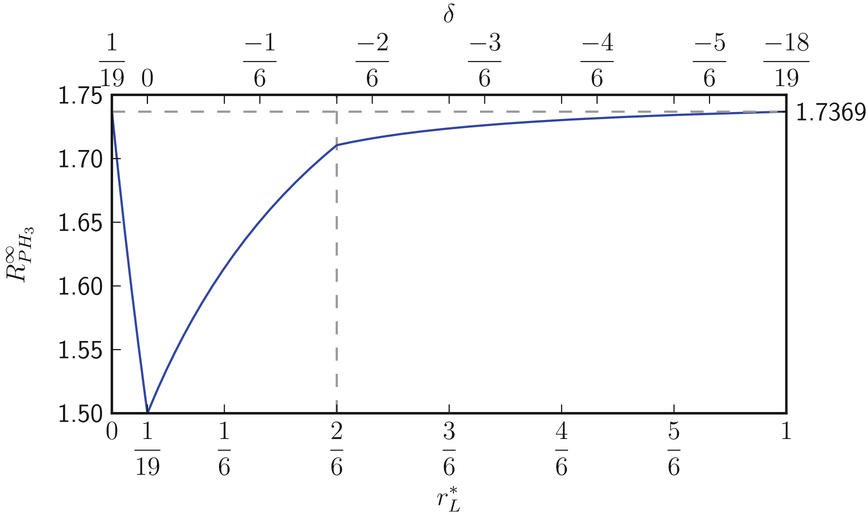

Fig. 1. Competitive ratio of the optimal 1-copy PH3 algorithm dependent on r^*_L for a fixed r_L.

If $r^i_{min} \leq 1/3$, we have $R = \frac{3}{2} + \frac{9}{2+6r^i_{min}}(r^i_L - r^i_{min})$. Solving this equation for r^i_L we get $r^i_L = r^i_{min} + \left(R - \frac{3}{2}\right)\left(\frac{2+6r^i_{min}}{9}\right)$. If $r^i_{min} \geq 1/3$, we have $R = \frac{3}{2} + \frac{3}{4r^i_{min}}(r^i_L - r^i_{min})$. Solving this equation for r^i_L yields $r^i_L = r^i_{min} + \left(R - \frac{3}{2}\right)\left(\frac{4r^i_{min}}{3}\right)$.

Having r^i_L, we can compute r^i_{max}. Because the competitive ratio is the minimum of two values, we get two candidates $r^i_{max,1}$ and $r^i_{max,2}$ for r^i_{max}. We can take the maximum of those two candidates, i.e., $r^i_{max} = \max(r^i_{max,1}, r^i_{max,2})$, because it is sufficient to be R-competitive in one case. In the first case $(\frac{3}{6r^*_L+2} < \frac{1}{4r^*_L})$ we obtain $r^i_{max,1} = \frac{3r^i_L-3+2R}{12-6R}$ and in the second case we get $r^i_{max,2} = \frac{r^i_L}{7-4R}$.

Now consider the case when $r^i_{max} \geq 1$. Because each algorithm A_ℓ with $1 \leq \ell \leq i$ is R-competitive for the interval $[r^\ell_{min}, r^\ell_{max}] = [r^{\ell-1}_{max}, r^\ell_{max}]$ with $r^0_{min} = 0$, there is an algorithm A_m for any $r^*_L \in [0,1]$ that is R-competitive. Therefore, we have a i-copy online algorithm for bin packing achieving the competitive factor R.

Following this method, we see that $k = 6$ algorithms are sufficient to guarantee a competitive ratio $R = 1.5815$. This beats the currently best 1-copy online algorithm Son Of Harmonic with a competitive ratio of 1.5816. Figure 2 shows the competitive ratio achieved by the individual algorithms over $r^*_L \in [0,1]$ for $R = 1.5815$. Note that 1.5815 is not the best competitive ratio achievable by 6-copy PH3, as shown below in Fig. 3. Using $k = 12$ algorithms, a competitive

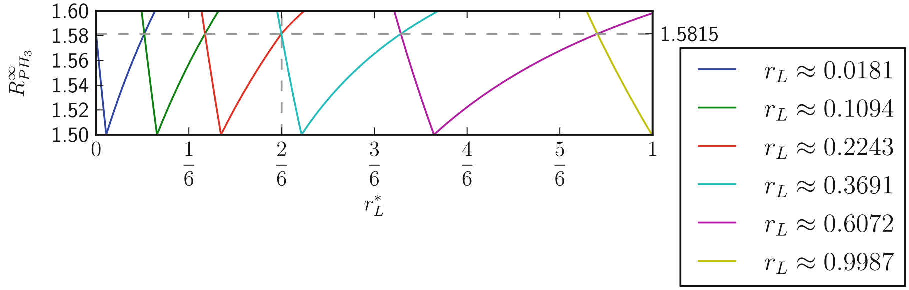

Fig. 2. 6-copy PH3 beats the best 1-copy online algorithm known to date, achieving an asymptotic competitive ratio $R^{\infty}_{PH3} < 1.5815$.

ratio $R = 1.5402 < 1.5403$ can be achieved, beating the highest known lower bound for 1-copy online algorithms.

To compute the best competitive ratio achievable by $k \in \mathbb{N}$ algorithms, we use binary search on R starting in the interval $[3/2, 33/19]$ and test in each iteration if we can guarantee R-competitiveness with at most k algorithms. Figure 3 shows the best competitive ratios achievable by k-copy PH3.

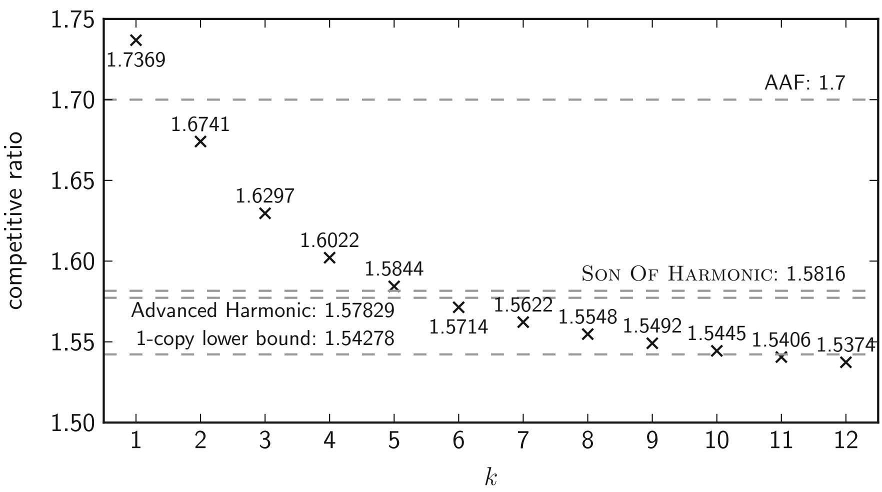

Fig. 3. k-copy PH3 performance dependent on k.

4.3 Comparison to Related Algorithms

Because k-copy online algorithms can be translated to an online algorithm with advice and vice versa (see Lemma 1), it seems natural to compare these two variants, even though k-copy allows a more precise analysis on the competitive ratio. In this subsection we compare our algorithm to the best known online algorithm with constant advice, namely RedBlue introduced by Angelopoulos et al. [2]. Their second algorithm is 1.47012-competitive (and thus beats our algorithm), but the amount of advice needed by this algorithm is too large. As the focus of k-copy algorithms is to provide good solutions for small k, it is reasonable to only compare k-copy PH3 to RedBlue.

Table 1 shows a comparison between RedBlue and k-copy PH3 for small amounts of advice. The competitive ratios given are rounded up to the fourth decimal place. The competitive ratios for RedBlue are computed using the upper bound on the competitive ratio $1.5 + 15/(2^{\ell/2+1})$. The competitive ratios for k-copy PH3 are calculated using binary search as described above.

Table 1. Comparison of the performance of k-copy PH3 and RedBlue.

Advice in bits	k	$R^{\infty}_{\text{RedBlue}}$	R^{∞}_{PH3}
4	16	3.3750	1.5305
5	32	2.8258	1.5155
6	64	2.4375	1.5078
7	128	2.1629	1.5040
8	256	1.9688	1.5020
9	512	1.8315	1.5010
10	1024	1.7344	1.5005
11	2048	1.6657	1.5003
12	4096	1.6172	1.5002
13	8192	1.5829	1.5001
14	16384	1.5586	1.5001
15	32768	1.5414	1.5001
16	65536	1.5293	1.5001

Table 1 clearly shows the advantage of k-copy PH3 over RedBlue for few bits of advice. With as few as 5 bits of advice, or $k = 32$, k-copy PH3 achieves a better competitive ratio than RedBlue with 16 bits of advice, which corresponds to $k = 65536$ algorithms when used as k-copy algorithm.

Although RedBlue and k-copy PH3 work in a similar way, k-copy PH3 achieves a better competitive ratio due to the more precise analysis of the intervals for r^*_L, in which each algorithm achieves the competitive ratio. By avoiding overlaps in these intervals, fewer algorithms are needed.

On the other hand, RedBlue simply splits an interval for its parameter β evenly into $2^{\ell/2}$ intervals; translated into a k-copy setting, this leads to overlaps in the intervals covered by each algorithm.

5 Conclusion

We studied the concept of parallel online algorithms for the Bin Packing Problem. We developed a k-copy online algorithm named PH3 and showed that PH3 has an asymptotic competitive ratio of 1.5 for large k; in particular, $k = 11$ suffices to break through the lower bound of a single online algorithm. We also considered the relationship to online algorithms with advice and achieved a considerable improvement compared to a previous algorithm.

There are various directions for future work. We saw that PH3 is $(1.5+\varepsilon)$-competitive if $\frac{|I_L|}{6\,\mathrm{size}(I_S)} \leq 1$, i.e., when there is a surplus of small items. If there are too few small items, PH3 is asymptotically $(1.5+\varepsilon)$-competitive. Can we make better use of the second case for an improvement? Can we guarantee an absolute competitive ratio of $1.5(+\varepsilon)$?

How does the asymptotic competitive ratio of PH3 depend on k? It seems to be something like $\frac{3}{2} + O\left(\frac{1}{k+\log_2(k+1)}\right)$. Translated to an online algorithm with ℓ bits of advice, this would yield an asymptotic competitive ratio of $\frac{3}{2} + O\left(\frac{1}{2^\ell+\ell}\right)$.

We also believe that the concept of k-copy algorithms is useful for a wide range of other problems.

References

1. Albers, S., Hellwig, M.: Online makespan minimization with parallel schedules. Algorithmica **78**(2), 492–520 (2017)
2. Angelopoulos, S., Dürr, C., Kamali, S., Renault, M., Rosén, A.: Online bin packing with advice of small size. In: Dehne, F., Sack, J.-R., Stege, U. (eds.) WADS 2015. LNCS, vol. 9214, pp. 40–53. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21840-3_4
3. Balogh, J., Békési, J., Dósa, G., Epstein, L., Levin, A.: A new and improved algorithm for online bin packing. In: 26th Annual European Symposium on Algorithms (ESA 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018)
4. Balogh, J., Békési, J., Dósa, G., Epstein, L., Levin, A.: A new lower bound for classic online bin packing. arXiv preprint arXiv:1807.05554 (2018). (To appear at 17th Workshop on Approximation and Online Algorithms (WAOA))
5. Boyar, J., Favrholdt, L.M., Kudahl, C., Larsen, K.S., Mikkelsen, J.W.: Online algorithms with advice: a survey. ACM Comput. Surv. (CSUR) **50**(2), 19 (2017)
6. Boyar, J., Kamali, S., Larsen, K.S., López-Ortiz, A.: On the list update problem with advice. In: 8th Conference on Language and Automata Theory and Applications (LATA), pp. 210–221 (2014)
7. Boyar, J., Kamali, S., Larsen, K.S., López-Ortiz, A.: Online bin packing with advice. Algorithmica **74**(1), 507–527 (2016)
8. Brown, D.M.: A lower bound for on-line one-dimensional bin packing algorithms. Technical report (1979)
9. Csirik, J., Woeginger, G.J.: On-line packing and covering problems. In: Fiat, A., Woeginger, G.J. (eds.) Online Algorithms: The State of the Art. LNCS, vol. 1442, pp. 147–177. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0029568
10. Fekete, S.P., Grosse-Holz, J., Keldenich, P., Schmidt, A.: Parallel online algorithms for the bin packing problem (2019). arXiv preprint (1910.03249)

11. Halldórsson, M.M., Iwama, K., Miyazaki, S., Taketomi, S.: Online independent sets. Theor. Comput. Sci. **289**(2), 953–962 (2002)
12. Heydrich, S., van Stee, R.: Beating the harmonic lower bound for online bin packing. In: The 43rd International Colloquium on Automata, Languages, and Programming (ICALP), pp. 41:1–41:14 (2016)
13. Johnson, D.S.: Fast algorithms for bin packing. J. Comput. Syst. Sci. **8**(3), 272–314 (1974)
14. Kamali, S., Ortiz, A.L.: Better compression through better list update algorithms. In: 2014 Data Compression Conference, pp. 372–381 (2014)
15. Lee, C.C., Lee, D.T.: A simple on-line bin-packing algorithm. J. ACM **32**, 562–572 (1985)
16. Liang, F.M.: A lower bound for on-line bin packing. Inf. Process. Lett. **10**(2), 76–79 (1980)
17. López-Ortiz, A., Schuierer, S.: On-line parallel heuristics, processor scheduling and robot searching under the competitive framework. Theor. Comput. Sci. **310**(1–3), 527–537 (2004)
18. Renault, M.P., Rosén, A., van Stee, R.: Online algorithms with advice for bin packing and scheduling problems. Theor. Comput. Sci. **600**, 155–170 (2015)
19. van Vliet, A.: An improved lower bound for on-line bin packing algorithms. Inf. Proc. Lett. **43**(5), 277–284 (1992)
20. Yao, A.C.-C.: New algorithms for bin packing. J. ACM **27**(2), 207–227 (1980)
21. Zhao, X., Shen, H.: On the advice complexity of one-dimensional online bin packing. In: Chen, J., Hopcroft, J.E., Wang, J. (eds.) FAW 2014. LNCS, vol. 8497, pp. 320–329. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08016-1_29