IEICE TRANS. INF. & SYST., VOL.E104–D, NO.3 MARCH 2021

PAPER Special Section on Foundations of Computer Science — New Trends of Theory of Computation and Algorithm —

Asymptotic Approximation Ratios for Certain Classes of Online Bin Packing Algorithms**

Hiroshi FUJIWARA[†a)], **Yuta WANIKAWA**[†*], ***Nonmembers***, ***and*** **Hiroaki YAMAMOTO**[†], ***Member***

SUMMARY The performance of online algorithms for the bin packing problem is usually measured by the asymptotic approximation ratio. However, even if an online algorithm is explicitly described, it is in general difficult to obtain the exact value of the asymptotic approximation ratio. In this paper we show a theorem that gives the exact value of the asymptotic approximation ratio in a closed form when the item sizes and the online algorithm satisfy some conditions. Moreover, we demonstrate that our theorem serves as a powerful tool for the design of online algorithms combined with mathematical optimization.

key words: *bin packing problem, online algorithm, competitive analysis, mathematical optimization*

1. Introduction

The bin packing problem is a basic and important problem in theoretical computer science [1], [2]. An *input sequence* I is a sequence of items, each labeled with size between zero and one. The task is to place every item in I into a bin so that the total size of items in each bin is at most one. The objective is to minimize the number of used bins. Here, we say a bin to be *used* if the bin contains at least one item, and say a bin to be *empty* otherwise.

Online algorithms, which make decisions of placement of items one by one while processing the input sequence from the head through the tail, have been intensively studied. The performance of online algorithms for the bin packing problem is usually measured by the *asymptotic approximation ratio* [3] (also sometimes called the *asymptotic competitive ratio*). Let $ALG(I)$ denote the number of used bins by an online algorithm ALG for a given input sequence I, and let $OPT(I)$ denote the number of the fewest bins that are needed to contain all items in I. The asymptotic approximation ratio of online algorithm ALG is defined as

$$R_{ALG} = \lim_{N\to\infty} \sup\left\{ \frac{ALG(I)}{OPT(I)} \;\middle|\; OPT(I) \geq N \right\}.$$

Clearly by definition, a smaller asymptotic approximation ratio indicates that the algorithm performs better.

In the literature we can find numerous studies on seeking the value of the asymptotic approximation ratio of an optimal online algorithm, which is an online algorithm with the smallest asymptotic approximation ratio,

$$R = \inf_{ALG} R_{ALG}.$$

The current best results are $R \leq 1.57828956$ [4] and $R \geq \frac{248}{161} \approx 1.54037267$ [5].

A common method for getting an upper bound is that: An online algorithm is first constructed and then its asymptotic approximation ratio is shown to be below a bound. In contrast, we propose a different approach in this paper. With some restrictions on item sizes and online algorithms, our theorem directly gives the exact value of the asymptotic approximation ratio. Besides, we reveal that our theorem works as a simple but powerful tool for designing online algorithms.

1.1 Our Contribution

In this paper we prove that under the assumption that the item sizes and the online algorithm satisfy some conditions, the exact value of the asymptotic approximation ratio of the online algorithm can be written in a closed form. More specifically, the conditions are:

- The size of each item is an element of a finite set. Moreover, the set of item sizes is such that any non-full packing can become a full packing by adding some items.
- The number of bins used by the online algorithm can be written as the maximum of functions that are linear in the number of items of each size.

Here, a *full* (*non-full*) packing means a set of items whose total size is equal to (respectively, less than) one. Our theorem states that if the above conditions are both satisfied, the asymptotic approximation ratio can be obtained by formally plugging all full packings, each expressed as a vector of the numbers of items per size, into the expression of the number of bins and taking their maximum.

Furthermore, we demonstrate that our theorem is helpful not only for identifying the asymptotic approximation ratio, but also for the design of online algorithms by mathematical optimization. To identify the asymptotic approximation ratio, our theorem does not necessarily require all full

Manuscript received March 27, 2020.
Manuscript revised August 7, 2020.
Manuscript publicized October 12, 2020.

[†]The authors are with Department of Electrical and Computer Engineering, Shinshu University, Nagano-shi, 380–8553 Japan.
[*]Presently, with VOYAGE GROUP, Inc.

**We are grateful to participants of the Enumeration Algorithms Seminars, Ikaho, Japan, for their helpful discussions. This work was supported by KAKENHI (16K00033, 17K00013, and 17K00183).

a) E-mail: fujiwara@cs.shinshu-u.ac.jp

DOI: 10.1587/transinf.2020FCP0004

packings. This property enables us to reduce constraints in the mathematical optimization problem.

1.2 Related Work

Despite considerable research, no optimal online algorithm is known for most variants of the bin packing problem, except for a few simple cases such as [6], [7] (see Sect. 2.1 for details). This motivates our research. Starting from development of a toolkit for constructing optimal algorithms for a restricted case, our aim is to extend it to more general cases.

In order to evaluate the asymptotic approximation ratio, the technique of assigning a *weight* to each item has been used from the very beginning of studies on the bin packing problem [8], [9]. Analysis of algorithms using extended weighting techniques can be found in the paper by Seiden [10]. The purpose of our research is the same as these studies. However, the goal of weighting techniques seems to bound the asymptotic approximation ratio from above. Differently from that, our technique identifies the exact value of the asymptotic approximation ratio itself, though the set of item sizes and the online algorithm are strongly restricted.

We mention some results on *offline algorithms*, which can process items in an arbitrary order. First of all, the bin packing problem is known to be NP-hard [1]. It had interested researchers whether there exists a polynomial-time algorithm that solves the variant where the size of each item is an element of a finite set of cardinality d, as we study in this paper. The question was affirmatively answered: McCormick et al. [11] gave a polynomial-time algorithm for $d = 2$. Goemans and Rothvoß [12] showed that the variant for each $d \geq 3$ is solvable in polynomial time.

2. Restrictions on Item Sizes and Online Algorithms

In this section we discuss a sufficient condition on the set of item sizes and the online algorithm for our theorem. We begin by introducing some mathematical notations. Let $\mathbb{Z}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$ be the set of all nonnegative integers and the set of all nonnegative real numbers, respectively. We denote a vector by a lowercase bold letter, like a nonnegative integer-valued d-entry vector by $\boldsymbol{a} = (a_1, a_2, \ldots, a_d) \in \mathbb{Z}^d_{\geq 0}$. Such a vector may represent a point in $\mathbb{Z}^d_{\geq 0}$ as well. Superscripts with parentheses are used to describe indices of a sequence of vectors, as $\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \ldots$. (Please do not be confused with exponentiation.) The notation $a^{(i)}_j$ stands for the j-th entry of vector $\boldsymbol{a}^{(i)}$. The operator $\times$ is used to either multiply two scalars, or multiply a vector by a scalar.

2.1 Restriction on Item Sizes

In this paper we impose two levels of restrictions on item sizes. The first level is that the size of each item is an element of a sequence of $(s_1, s_2, \ldots, s_d) \in (0, 1]^d$. An item of size s_j is called an s_j-item. The second level is that the sequence is complement-closed, which will be defined soon. Our main theorem requires the second level restriction.

We here mention the studies on the first level restriction in the literature. We have already introduced in Sect. 1.2 that for any fixed d, the bin packing problem with the first level restriction can be solved in polynomial time [11], [12]. For the bin packing problem in which the size of each item is an element of (s_1, s_2) with $\frac{1}{2} < s_1 \leq 1$ and $0 < s_2 \leq s_1$, Gutin, Jensen, and Yeo [6] derived the asymptotic approximation ratio of an optimal online algorithm

$$R = \frac{\lfloor \frac{1}{s_2} \rfloor^2}{\lfloor \frac{1}{s_2} \rfloor^2 - \lfloor \frac{1-s_1}{s_2} \rfloor (\lfloor \frac{1}{s_2} \rfloor - \lfloor \frac{1-s_1}{s_2} \rfloor)}.$$

For the same problem under the setting of $\frac{1}{k+1} < s_1 \leq \frac{1}{k+1} + \frac{1}{(k+1)^2(k+2)}$ and $\frac{1}{k+2} < s_2 \leq \frac{1}{k+2} + \frac{1}{(k+1)^2(k+2)}$ for an integer $k \geq 2$, Epstein and Levin [7] proved $R = \frac{(k+1)^2}{k^2+k+1}$.

The following defines the second level restriction on item sizes.

Definition 1. *An item size sequence* $(s_1, s_2, \ldots, s_d) \in (0, 1]^d$ *is called* complement-closed *if for all* $\boldsymbol{n} \in \mathbb{Z}^d_{\geq 0}$ *with*

$$\sum_{j=1}^{d} s_j n_j < 1,$$

there exists $\boldsymbol{n}' \in \mathbb{Z}^d_{\geq 0}$ *such that*

$$\sum_{j=1}^{d} s_j n_j + \sum_{j=1}^{d} s_j n'_j = 1.$$

In short, the condition says that any non-full packing can be a full packing by adding some items. For example, an item size sequence of $(s_1, s_2, s_3) = (\frac{3}{4}, \frac{2}{4}, \frac{1}{4})$ is complement-closed. Contrarily, $(s_1, s_2) = (\frac{3}{4}, \frac{2}{4})$ is not complement-closed, since it is impossible to add any item to a $\frac{3}{4}$-item to let the total size be one.

This restriction is motivated by assignment of pre-packed items to bins. In the real world, delivery services companies, such as DHL or FedEx, offer various boxes for packing items to customers, as said on their websites [13], [14]. The dimensions of such boxes are designed for efficient packing. Thus, in the bin packing problem with a complement-closed item size sequence, each item stands for a pre-packed box.

We next introduce some related notations, which will be used in the proofs later. For a complement-closed item size sequence $(s_1, s_2, \ldots, s_d)$, we define

$$E_1 = \left\{ \boldsymbol{n} \in \mathbb{Z}^d_{\geq 0} \;\middle|\; \sum_{j=1}^{d} s_j n_j = 1 \right\}.$$

That is to say, E_1 is the set of all full packings. Obviously, E_1 is a finite set. Letting $q = |E_1|$, we assume that all full packing are enumerated as

$$E_1 = \{\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \ldots, \boldsymbol{p}^{(q)}\}.$$

For an integer $m \geq 2$, we define

$$E_m = \left\{ \boldsymbol{n} \in \mathbb{Z}^d_{\geq 0} \;\middle|\; (\exists \boldsymbol{b} \in \mathbb{Z}^q_{\geq 0})\ \boldsymbol{n} = \sum_{k=1}^{q} b_k \boldsymbol{p}^{(k)}, m = \sum_{k=1}^{q} b_k \right\}.$$

In other words, E_m is the whole set of the sums of arbitrary m full packings. We also set

$$E = \mathrm{conv}(E_1),$$

that is, the convex hull of E_1. Let

$$\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(q')}$$

be the q' ($\leq q$) vertices of E as a polytope. (See [15], for example, for the treatment of convex hulls.) From the viewpoint of geometry, it is seen that since every element of E_1 is on a hyperplane in $\mathbb{R}^d$, E is a subset of the hyperplane.

2.2 Class of Online Algorithms

In what follows we assume the first level restriction on item sizes that the size of each item is an element of a sequence of $(s_1, s_2, \ldots, s_d)$. We consider two levels of restrictions on online algorithms as well. The first level is that the number of bins used by the online algorithm is independent of the order of arrival of items. The second level is that the online algorithm belongs to class max-linear, defined below. Our main theorem needs the second level restriction.

Let $\boldsymbol{n} = (n_1, n_2, \ldots, n_d) \in \mathbb{Z}^d_{\geq 0}$ represent input sequences that consist of n_j s_j-items for each $j = 1, 2, \ldots, d$. For an online algorithm that fulfills the first level restriction, we can write the number of used bins as $ALG(\boldsymbol{n})$. Note that depending on the order of arrival, the assignment of items to bins may differ, while the number of used bins is the same. On the other hand, since the minimum number of bins is essentially independent of the order of arrival, we denote it by $OPT(\boldsymbol{n})$.

It may seem that the first level restriction is so strong that the nature of the bin packing problem changes significantly. We will claim that this perception is not true by illustrating that some existing online algorithms satisfy even the second level restriction.

We define class max-linear of online algorithms.

Definition 2. *Suppose that the size of each item is an element of* $(s_1, s_2, \ldots, s_d)$. *An online algorithm is said to be in* class max-linear *if there exist a sequence of vectors* $\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \ldots, \boldsymbol{a}^{(M)} \in \mathbb{R}^d$, *a sequence of functions* $\delta_1, \delta_2, \ldots, \delta_M : \mathbb{Z}^d_{\geq 0} \to \mathbb{R}$, *and a positive real number* Δ *such that for all* $\boldsymbol{n} \in \mathbb{Z}^d_{\geq 0}$,

$$ALG(\boldsymbol{n}) = \max_{1 \leq i \leq M} \left(\sum_{j=1}^{d} a^{(i)}_j n_j + \delta_i(\boldsymbol{n}) \right)$$

and $|\delta_i(\boldsymbol{n})| < \Delta$ *for all* $1 \leq i \leq M$.

We here rewrite the asymptotic approximation ratio of online algorithm ALG in class max-linear:

$$R_{ALG} = \lim_{N \to \infty} \sup_{m \geq N} \max \left\{ \frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; OPT(\boldsymbol{n}) = m \right\}.$$

The definition of class max-linear says that the number of used bins is the maximum of some linear combinations of $n_1, n_2, \ldots, n_d$, each including a bounded error $\delta_i(\boldsymbol{n})$. We remark that $\delta_i(\boldsymbol{n})$ does not affect the asymptotic approximation ratio. As long as we use the asymptotic approximation ratio as a performance measure, a constant number of wasted bins are negligible.

Under the assumption that the size of each item is an element of $(s_1, s_2, \ldots, s_d)$, the famous algorithm HARMONIC [16] belong to class max-linear. The algorithm classifies items of size in interval $(1/(k+1), 1/k]$ into one category for each $k = 1, 2, \ldots$, and place them in such a way that every bin contains only items of a single category. The number of used bins is expressed as a linear function of the number of items of each category plus a bounded term. (Please see also Sect. 3.1.)

For the original bin packing problem, algorithm HARMONIC is known to be an optimal online algorithm among *bounded-space* online algorithms [16], where an online algorithm is called *bounded-space* if the number of bins in which each item may be placed is bounded independently of the number of items [3]. Therefore, we believe it important to investigate the class that contains HARMONIC.

Algorithm REFINED FIRST FIT [17] is also in class max-linear, if there is at most one size in each of the predefined four categories. It is easy to check that the trivial algorithm "Placing every item in a new bin" is in class max-linear.

3. Theorem for Identifying the Asymptotic Approximation Ratio

Our main theorem states that if some conditions are fulfilled, the asymptotic approximation ratio of an online algorithm can be represented using the parameter and the vertices of E.

Theorem 1. *Suppose that the size of each item is an element of a complement-closed item size sequence of* $(s_1, s_2, \ldots, s_d)$. *For any online algorithm ALG in class max-linear, it follows that*

$$R_{ALG} = \max_{1 \leq i \leq M, 1 \leq k \leq q'} \sum_{j=1}^{d} a^{(i)}_j v^{(k)}_j.$$

3.1 Application to Algorithm HARMONIC

For a complement-closed item size sequence $(s_1, s_2, s_3) = (\frac{3}{4}, \frac{2}{4}, \frac{1}{4})$ as an example, let us apply Theorem 1 to algorithm HARMONIC [16]. Noting $\frac{3}{4} \in (\frac{1}{2}, 1]$, $\frac{2}{4} \in (\frac{1}{3}, \frac{1}{2}]$,

and $\frac{1}{4} \in (\frac{1}{5}, \frac{1}{4}]$, we can write the number of used bins as $HARMONIC(\boldsymbol{n}) = n_1 + \frac{n_2}{2} + \frac{n_3}{4} + \delta(\boldsymbol{n})$ with a function δ satisfying $0 \le \delta(\boldsymbol{n}) \le 2$.

Full packings for $(s_1, s_2, s_3) = (\frac{3}{4}, \frac{2}{4}, \frac{1}{4})$ are enumerated as: $\boldsymbol{p}^{(1)} = (1, 0, 1)$, $\boldsymbol{p}^{(2)} = (0, 2, 0)$, $\boldsymbol{p}^{(3)} = (0, 1, 2)$, and $\boldsymbol{p}^{(4)} = (0, 0, 4)$. Then, the vertices of the convex hull E are: $\boldsymbol{v}^{(1)} = (1, 0, 1)$, $\boldsymbol{v}^{(2)} = (0, 2, 0)$, and $\boldsymbol{v}^{(3)} = (0, 0, 4)$. Note that $\boldsymbol{p}^{(3)} = (0, 1, 2)$ is redundant because $(0, 1, 2) = \frac{1}{2} \times (0, 2, 0) + \frac{1}{2} \times (0, 0, 4)$. By applying Theorem 1 we calculate $R_{HARMONIC} = \max\{1\times1+\frac{1}{2}\times0+\frac{1}{4}\times1, 1\times0+\frac{1}{2}\times2+\frac{1}{4}\times0, 1\times0+\frac{1}{2}\times0+\frac{1}{4}\times4\} = \frac{5}{4}$ $(= 1.25)$.

3.2 Proofs

We first show two helper lemmas for Theorem 1.

Lemma 1. *Suppose that the size of each item is an element of a complement-closed item size sequence of $(s_1, s_2, \ldots, s_d)$. For any positive integer m and any online algorithm ALG such that the number of used bins is independent of the order of arrival of items, it holds that*

$$\max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; OPT(\boldsymbol{n}) = m\right\} = \max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; \boldsymbol{n} \in E_m\right\}.$$

Proof. Let m be a positive integer and ALG be an online algorithm. Since every element of E_m is the sum of m full packings, $\boldsymbol{n} \in E_m$ implies $OPT(\boldsymbol{n}) = m$. Hence,

$$\max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; OPT(\boldsymbol{n}) = m\right\} \ge \max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; \boldsymbol{n} \in E_m\right\}$$

holds true. Then, it remains to prove

$$\max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; OPT(\boldsymbol{n}) = m\right\} \le \max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; \boldsymbol{n} \in E_m\right\}. \quad (1)$$

To prove (1), it is sufficient to claim that for all positive integer m and all $\boldsymbol{n} \in \mathbb{Z}^d_{\ge 0}$ such that $OPT(\boldsymbol{n}) = m$, there exists $\boldsymbol{n}'$ such that $\boldsymbol{n}' \in E_m$ and $\frac{ALG(\boldsymbol{n}')}{OPT(\boldsymbol{n}')} \ge \frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})}$. The rest of the proof is devoted to showing this.

Let $\boldsymbol{n}$ be a vector in $\mathbb{Z}^d_{\ge 0}$ such that $OPT(\boldsymbol{n}) = m$. If $\sum_{j=1}^d s_j n_j = m$, we are done by setting $\boldsymbol{n}' = \boldsymbol{n}$.

Now assume that $\sum_{j=1}^d s_j n_j < m$. Consider an assignment of items represented by $\boldsymbol{n}$ to m bins. Such an assignment always exists since $OPT(\boldsymbol{n}) = m$. Some bins are of non-full packings. Set $\boldsymbol{n}'$ to $\boldsymbol{n}$ plus additional items so that all of those bins become bins of full packings. This is possible since we have assumed that the size of each item is chosen from a complement-closed item size sequence. Then, we have $\boldsymbol{n}' \in E_m$ and $OPT(\boldsymbol{n}') = m$.

On the other hand, consider an assignment of items represented by $\boldsymbol{n}$ to bins by ALG. Next, give additional items of the difference of $\boldsymbol{n}'$ from $\boldsymbol{n}$ to ALG. Since ALG is an online algorithm, it cannot change the original assignment and hence the number of used bins increases or remains the same. Together with the fact that the order of arrival does not affect the number of used bins, we obtain $ALG(\boldsymbol{n}') \ge ALG(\boldsymbol{n})$. Consequently, we have $\frac{ALG(\boldsymbol{n}')}{OPT(\boldsymbol{n}')} \ge \frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})}$. □

Lemma 2. *For all integers i and m with $1 \le i \le M$ and $1 \le m$, it holds that*

$$\max\left\{\sum_{j=1}^d a_j^{(i)} \times \frac{n_j}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\} = \max_{1\le k\le q'} \sum_{j=1}^d a_j^{(i)} v_j^{(k)}. \quad (2)$$

Proof. We first give a sketch of the proof. We extend the domain of the mapping $(u_1, u_2, \ldots, u_d) \mapsto \sum_{j=1}^d a_j^{(i)} u_j$ to E. Then, it turns out that the maximum of images is achieved on a vertex of E. On the other hand, every vertex of E is contained in the original domain. Hence, the maximum over the original domain is also achieved on a vertex of E, which states the lemma.

Let i and m be integers such that $1 \le i \le M$ and $1 \le m$. Define

$$F = \left\{\left(\frac{n_1}{m}, \frac{n_2}{m}, \ldots, \frac{n_d}{m}\right) \;\middle|\; \boldsymbol{n} \in E_m\right\}.$$

The left-hand side of (2) is the maximum of images by

$$\phi : F \ni \boldsymbol{u} \mapsto \sum_{j=1}^d a_j^{(i)} u_j \in \mathbb{R}.$$

We begin by showing $F \subseteq E$. By definition of E_m, for all $\boldsymbol{n} \in E_m$, there exists $\boldsymbol{b} \in \mathbb{Z}^q_{\ge 0}$ such that $\boldsymbol{n} = \sum_{k=1}^q b_k \boldsymbol{p}^{(k)}$ and $m = \sum_{k=1}^q b_k$. Dividing the both-hand sides of these equalities by m, we get $(\frac{n_1}{m}, \frac{n_2}{m}, \ldots, \frac{n_d}{m}) = \sum_{k=1}^q \frac{b_k}{m} \times \boldsymbol{p}^{(k)}$ and $1 = \sum_{k=1}^q \frac{b_k}{m}$. This implies that $(\frac{n_1}{m}, \frac{n_2}{m}, \ldots, \frac{n_d}{m})$ is in the convex hull of $\{\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \ldots, \boldsymbol{p}^{(q)}\}$, which is E.

We next calculate the maximum of images by ϕ when the domain is extended to E. Recall that convex hull E is spanned by $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(q')}$. Every $\boldsymbol{u} \in E$ is displayed as $\boldsymbol{u} = \sum_{k=1}^{q'} t_k \boldsymbol{v}^{(k)}$ using $\boldsymbol{t} \in \mathbb{R}^{q'}_{\ge 0}$ such that $\sum_{k=1}^{q'} t_k = 1$. $\boldsymbol{u}$ is mapped by ϕ to

$$\sum_{j=1}^d a_j^{(i)} u_j = \sum_{j=1}^d a_j^{(i)} \Big(\sum_{k=1}^{q'} t_k v_j^{(k)} \Big) = \sum_{k=1}^{q'} t_k \Big(\sum_{j=1}^d a_j^{(i)} v_j^{(k)} \Big) = \sum_{k=1}^{q'} t_k A_k,$$

where $A_k = \sum_{j=1}^d a_j^{(i)} v_j^{(k)}$. We below show that the maximum value of $\sum_{k=1}^{q'} t_k A_k$ equals $\max_{1\le k\le q'} A_k$. Without loss of generality, assume that A_1 is the maximum among $A_1, A_2, \ldots, A_k$. Then, we derive

$$A_1 - \sum_{k=1}^{q'} t_k A_k = A_1 - t_1 A_1 - \sum_{k=2}^{q'} t_k A_k$$
$$= A_1 - \left(1 - \sum_{k=2}^{q'} t_k\right) A_1 - \sum_{k=2}^{q'} t_k A_k = \sum_{k=2}^{q'} t_k (A_1 - A_k) \geq 0,$$

which shows that the maximum value of $\sum_{k=1}^{q'} t_k A_k$ is A_1. In this way, we have that the maximum of images of E by ϕ is $\max_{1 \leq k \leq q'} A_k$ and it is achieved by one of the vertices $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(q')}$ of E. (From the viewpoint of geometry, this argument can be understood more simply: The convex hull E is mapped by ϕ, as a projection from $\mathbb{R}^d$ to a line, to a line segment. Its both endpoints correspond to any two of the vertices of E.)

We finally claim that the vertices $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots, \boldsymbol{v}^{(q')}$ of E are all in F. Let k be an integer with $1 \leq k \leq q'$. Since E is the convex hull of $\{\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \ldots, \boldsymbol{p}^{(q)}\}$, $\boldsymbol{v}^{(k)}$ is equal to one of $\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \ldots, \boldsymbol{p}^{(q)}$. Without loss of generality, assume $\boldsymbol{v}^{(k)} = \boldsymbol{p}^{(1)}$. Then, by taking $\boldsymbol{b} = (m, 0, 0, \ldots, 0)$, we have $\sum_{k=1}^{q} b_k \boldsymbol{p}^{(k)} = m\boldsymbol{p}^{(1)}$ and $m = \sum_{k=1}^{q} b_k$, which means $m\boldsymbol{p}^{(1)} \in E_m$ by definition of E_m. Furthermore, this implies $\boldsymbol{p}^{(1)} \in F$ by definition of F. Therefore, $\boldsymbol{v}^{(k)} \in F$.

The lemma is thus proved. □

We are ready to prove the main theorem.

Proof of Theorem 1. Let m be a positive integer and ALG be an online algorithm. Using Lemma 1, we derive

$$\max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; OPT(\boldsymbol{n}) = m\right\}$$
$$= \max\left\{\frac{ALG(\boldsymbol{n})}{OPT(\boldsymbol{n})} \;\middle|\; \boldsymbol{n} \in E_m\right\}$$
$$= \max\left\{\frac{\max_{1 \leq i \leq M}\left\{\sum_{j=1}^{d} a_j^{(i)} n_j + \delta_i(\boldsymbol{n})\right\}}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\}$$
$$= \max\left\{\max_{1 \leq i \leq M}\left\{\sum_{j=1}^{d} a_j^{(i)} \times \frac{n_j}{m} + \frac{\delta_i(\boldsymbol{n})}{m}\right\} \;\middle|\; \boldsymbol{n} \in E_m\right\}$$
$$= \max_{1 \leq i \leq M}\left\{\max\left\{\sum_{j=1}^{d} a_j^{(i)} \times \frac{n_j}{m} + \frac{\delta_i(\boldsymbol{n})}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\}\right\}.$$

We define

$$r_{i,m} = \max\left\{\sum_{j=1}^{d} a_j^{(i)} \times \frac{n_j}{m} + \frac{\delta_i(\boldsymbol{n})}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\}$$

for $1 \leq i \leq M$.

Let N be a positive integer. We next bound $r_{i,m}$ for $m \geq N$ using Lemma 2. We obtain

$$r_{i,m} < \max\left\{\sum_{j=1}^{d} a_j^{(i)} \times \frac{n_j}{m} + \frac{\Delta}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\}$$
$$= \max\left\{\sum_{j=1}^{d} a_j^{(i)} \times \frac{n_j}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\} + \frac{\Delta}{m}$$
$$\leq \max_{1 \leq k \leq q'} \sum_{j=1}^{d} a_j^{(i)} v_j^{(k)} + \frac{\Delta}{N}.$$

On the other hand,

$$r_{i,m} > \max\left\{\sum_{j=1}^{d} a_j^{(i)} \times \frac{n_j}{m} - \frac{\Delta}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\}$$
$$= \max\left\{\sum_{j=1}^{d} a_j^{(i)} \times \frac{n_j}{m} \;\middle|\; \boldsymbol{n} \in E_m\right\} - \frac{\Delta}{m}$$
$$\geq \max_{1 \leq k \leq q'} \sum_{j=1}^{d} a_j^{(i)} v_j^{(k)} - \frac{\Delta}{N}.$$

Thus, the supremum value is also bounded as

$$\max_{1 \leq k \leq q'} \sum_{j=1}^{d} a_j^{(i)} v_j^{(k)} - \frac{\Delta}{N}$$
$$< \sup_{m \geq N} r_{i,m} \leq \max_{1 \leq k \leq q'} \sum_{j=1}^{d} a_j^{(i)} v_j^{(k)} + \frac{\Delta}{N}.$$

As $N \to \infty$, $\frac{\Delta}{N}$ converges to zero. Hence, by the squeeze theorem, we conclude that

$$R_{ALG} = \lim_{N \to \infty} \sup_{m \geq N} \max_{1 \leq i \leq M} r_{i,m}$$
$$= \max_{1 \leq i \leq M} \lim_{N \to \infty} \sup_{m \geq N} r_{i,m} = \max_{1 \leq i \leq M, 1 \leq k \leq q'} \sum_{j=1}^{d} a_j^{(i)} v_j^{(k)}.$$

□

4. Application to Designing Online Algorithms

In this section we give an application of Theorem 1 to the design of online algorithms for the bin packing problem with a complement-closed item size sequence. We first describe a simple online algorithm that belongs to class max-linear.

4.1 Algorithm DAR

Algorithm DAR (Distribute According to the Ratio) takes a vector $(z_1, z_2, \ldots, z_q) \in \mathbb{Z}_{\geq 0}^{q}$ as a parameter. Roughly speaking, the algorithm tries to place items so that for each $k = 1, 2, \ldots, q$, the number of bins that realize full packing $\boldsymbol{p}^{(k)} \in E_1$ is proportional to z_k. We divide the whole set of bins into groups $1, 2, \ldots, q$. In each group, all bins are further indexed $1, 2, \ldots$. Every bin in group k will be eventually filled by items as the same as full packing $\boldsymbol{p}^{(k)}$.

For each $j = 1, 2, \ldots, d$, we construct a mapping for algorithm DAR to distribute items, one by one, according to the parameter $(z_1, z_2, \ldots, z_q)$. Set $D_j = \sum_{l=1}^{q} z_l p_j^{(l)}$. We define a mapping $\psi_j : \{0, 1, \ldots, D_j - 1\} \to \{1, 2, \ldots, q\}$ by: For $x = 0, 1, \ldots, D_j - 1$ and $k = 1, 2, \ldots, q$,

Algorithm 1: DAR (Distribute According to the Ratio).

1 Reset counters $c_j \leftarrow 0$ and $c_j^{(k)} \leftarrow 0$ for $j = 1, 2, \ldots, d$ and $k = 1, 2, \ldots, q$;
2 **foreach** *item* a **do**
3 $\quad j \leftarrow$ the index such that the size of a is s_j;
4 $\quad k \leftarrow \psi_j(c_j \bmod D_j)$;
5 $\quad c_j \leftarrow c_j + 1$;
6 $\quad c_j^{(k)} \leftarrow c_j^{(k)} + 1$;
7 $\quad$ place a in the $\lceil \frac{c_j^{(k)}}{p_j^{(k)}} \rceil$-th bin in group k;
8 **end**

$$\psi_j(x) = k \Leftrightarrow \sum_{l=1}^{k-1} z_l p_j^{(l)} \le x \le \sum_{l=1}^{k} z_l p_j^{(l)} - 1.$$

The value of ψ_j tells to which group each s_j-item should be sent. Specifically, the first $z_1 p_j^{(1)}$ items are sent to group 1 for creating bins of full packing $\boldsymbol{p}^{(1)}$. The next $z_2 p_j^{(2)}$ items are sent to group 2, and so on. Algorithm DAR repeatedly does the same for every D_j s_j-items, as long as the sequence continues.

See Algorithm 1 for a pseudocode of algorithm DAR. During the execution, counters of the number of items are maintained individually for each size and each group. At the beginning of each iteration, c_j equals the number of s_j-items that have arrived so far, while $c_j^{(k)}$ is the number of s_j-items that have arrived and been placed in bins in group k so far. Upon arrival of an item, algorithm DAR determines a group to which the item belongs by ψ_j and c_j, and then a bin in which the item is placed by $c_j^{(k)}$. Note that every placement is feasible, since it is done for realizing some full packing.

It is confirmed that algorithm DAR is bounded-space since each item may be placed in any of at most $d \times q$ bins.

4.2 Number of Bins Used by Algorithm DAR

We evaluate the number of bins used by algorithm DAR and confirm that the algorithm is in class max-linear. Suppose that algorithm DAR is run for $\boldsymbol{n}$. Recall that $\boldsymbol{n}$ represents input sequences consisting of n_j s_j-items for each $j = 1, 2, \ldots, d$. Let $n_j^{(k)}$ denote the value of counter $c_j^{(k)}$ when the algorithm finishes processing the entire input sequence. In other words, $n_j^{(k)}$ is the total number of s_j-items that have been placed in bins in group k. Since algorithm DAR places each item in the $\lceil \frac{c_j^{(k)}}{p_j^{(k)}} \rceil$-th bin, the maximum index of used bins in group k is written as

$$\max_{j_k : 1 \le j_k \le d, z_k p_{j_k}^{(k)} \neq 0} \left\lceil \frac{n_{j_k}^{(k)}}{p_{j_k}^{(k)}} \right\rceil,$$

which also equals the number of used bins in group k. Here we employ j_k rather than j because in the following discussion we want to move this index independently for each k. The condition of the maximum operation is valid, because if $z_k = 0$ or $p_{j_k}^{(k)} = 0$ then s_{j_k}-items are never sent to group k. By summing up the above number for each group $1, 2, \ldots, q$, the number of bins used by algorithm DAR is derived as

$$\begin{aligned} DAR(\boldsymbol{n}) &= \sum_{k=1}^{q} \max_{j_k : 1 \le j_k \le d, z_k p_{j_k}^{(k)} \neq 0} \left\lceil \frac{n_{j_k}^{(k)}}{p_{j_k}^{(k)}} \right\rceil \\ &= \max_{(j_1, j_2, \ldots, j_q) \in J} \sum_{k=1}^{q} \left\lceil \frac{n_{j_k}^{(k)}}{p_{j_k}^{(k)}} \right\rceil, \end{aligned}$$

where

$$J = \{(j_1, j_2, \ldots, j_q) \mid (\forall k = 1, 2, \ldots, q)\ 1 \le j_k \le d, z_k p_{j_k}^{(k)} \neq 0\}.$$

For each $j = 1, 2, \ldots, d$, algorithm DAR periodically distributes $z_k p_j^{(k)}$ items to group k out of D_j s_j-items. Therefore, it holds that

$$\frac{z_k p_j^{(k)}}{D_j} \times n_j - z_k p_j^{(k)} \le n_j^{(k)} \le \frac{z_k p_j^{(k)}}{D_j} \times n_j + z_k p_j^{(k)} \tag{3}$$

for $j = 1, 2, \ldots, d$ and $k = 1, 2, \ldots, q$.

We are ready to check that algorithm DAR is in class max-linear. We use the fact that for all real number x, $\lceil x \rceil = x + (\lceil x \rceil - x)$ and $0 \le (\lceil x \rceil - x) < 1$ hold. Also, by inequalities (3), we have that: There exist a sequence of functions $\delta_{(j_1, \ldots, j_d)} : \mathbb{Z}_{\ge 0}^d \to \mathbb{R}$ for $(j_1, \ldots, j_q) \in J$ and a positive real number Δ such that for all $\boldsymbol{n} \in \mathbb{Z}_{\ge 0}^d$,

$$\begin{aligned} &DAR(\boldsymbol{n}) \\ &= \max_{(j_1, \ldots, j_q) \in J} \left(\sum_{k=1}^{q} \frac{1}{p_{j_k}^{(k)}} \times \frac{z_k p_{j_k}^{(k)}}{D_{j_k}} \times n_{j_k} + \delta_{(j_1, \ldots, j_d)}(\boldsymbol{n}) \right) \\ &= \max_{(j_1, \ldots, j_q) \in J} \left(\sum_{k=1}^{q} \frac{z_k}{D_{j_k}} \times n_{j_k} + \delta_{(j_1, \ldots, j_d)}(\boldsymbol{n}) \right). \end{aligned} \tag{4}$$

and $|\delta_{(j_1, \ldots, j_d)}(\boldsymbol{n})| < \Delta$ for all $(j_1, \ldots, j_q) \in J$. By assigning a positive integer to each element of J, we know that algorithm DAR is in class max-linear. It may happen that $k_1 \neq k_2$ but $j_{k_1} = j_{k_2}$. Then, some n_j appears more than once in the sum $\sum_{k=1}^{q} \frac{z_k}{D_{j_k}} \times n_{j_k}$. Nevertheless, the sum is still linear in n_j's.

4.3 An Example of the Behavior of Algorithm DAR

We use $(s_1, s_2, s_3) = \left(\frac{3}{4}, \frac{2}{4}, \frac{1}{4}\right)$ again. We fix a parameter of $(z_1, z_2, z_3, z_4) = (2, 1, 2, 1)$ for algorithm DAR and observe its behavior. We get $D_1 = 2$, $D_2 = 4$, and $D_3 = 10$. Mappings ψ_1, ψ_2, and ψ_3 are defined as Table 1.

We first see the behavior for an input sequence of 12 $\frac{2}{4}$-items and 17 $\frac{1}{4}$-items, followed by 6 $\frac{3}{4}$-items, which is represented by $\boldsymbol{n} = (12, 6, 17)$. Figure 1 illustrates how algorithm DAR places the items to bins. The resulting number of used bins $DAR(\boldsymbol{n}) = 17$, while $OPT(\boldsymbol{n}) = 15$.

Table 1 Mappings ψ_1, ψ_2, and ψ_3 for distributing s_1-, s_2-, and s_3-items, respectively. A hyphen means "not defined".

x	0	1	2	3	4	5	6	7	8	9
$\psi_1(x)$	1	1	-	-	-	-	-	-	-	-
$\psi_2(x)$	2	2	3	3	-	-	-	-	-	-
$\psi_3(x)$	1	1	3	3	3	3	4	4	4	4

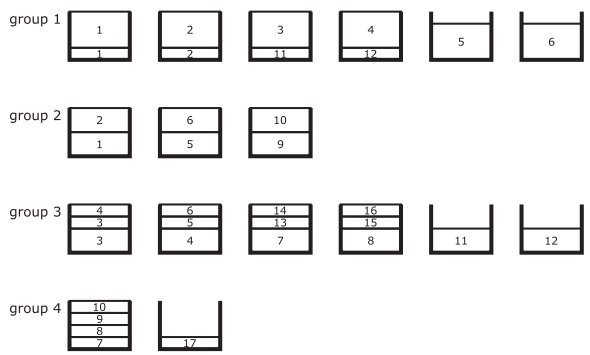


Fig. 1 Behavior of algorithm DAR with a parameter of $(z_1, z_2, z_3, z_4) = (2, 1, 2, 1)$, under the assumption that the size of each item is an element of a complement-closed item size sequence of $(s_1, s_2, s_3) = (\frac{3}{4}, \frac{2}{4}, \frac{1}{4})$. The input sequence here is of 12 $\frac{2}{4}$-items and 17 $\frac{1}{4}$-items, followed by 6 $\frac{3}{4}$-items, which is represented by $\boldsymbol{n} = (12, 6, 17)$. The number on the items indicates the order in which they are placed, numbered separately by size. Bins in each group are indexed $1, 2, \ldots$ from left to right. The resulting number of used bins $DAR(\boldsymbol{n}) = 17$, while $OPT(\boldsymbol{n}) = 15$.

We then evaluate the performance of algorithm DAR for $(z_1, z_2, z_3, z_4) = (2, 1, 2, 1)$. The vertices of the convex hull E are $\boldsymbol{v}^{(1)} = (1, 0, 1)$, $\boldsymbol{v}^{(2)} = (0, 2, 0)$, and $\boldsymbol{v}^{(3)} = (0, 0, 4)$ as we got in Sect. 3.1. Omitting the details, we conclude that for $(z_1, z_2, z_3, z_4) = (2, 1, 2, 1)$, $R_{DAR} = \max\{\frac{5}{4}, \frac{4}{3}, \frac{5}{3}\} = \frac{5}{3}$ (≈ 1.67), which is much worse than $R_{HARMONIC} = 1.25$.

4.4 Optimizing the Parameter by Linear Optimization

We demonstrate an optimization of the parameter of algorithm DAR with the help of Theorem 1 and linear optimization (linear programming). We still continue to use the complement-closed item size sequence $(s_1, s_2, s_3) = (\frac{3}{4}, \frac{2}{4}, \frac{1}{4})$.

Consider (z_1, z_2, z_3, z_4) to be variables. From (4), we have that $DAR(\boldsymbol{n})$ is the maximum value of the following four values:

$$n_1 + x_{22}n_2 + x_{23}n_2 + x_{34}n_3 + \delta_{(1,2,2,3)}(\boldsymbol{n}), \tag{5}$$

$$n_1 + x_{22}n_2 + x_{33}n_3 + x_{34}n_3 + \delta_{(1,2,3,3)}(\boldsymbol{n}), \tag{6}$$

$$x_{31}n_3 + x_{22}n_2 + x_{23}n_2 + x_{34}n_3 + \delta_{(3,2,2,3)}(\boldsymbol{n}), \text{ and} \tag{7}$$

$$x_{31}n_3 + x_{22}n_2 + x_{33}n_3 + x_{34}n_3 + \delta_{(3,2,3,3)}(\boldsymbol{n}), \tag{8}$$

where we have replaced (z_1, z_2, z_3, z_4) by a new sequence of variables $(x_{22}, x_{23}, x_{31}, x_{33}, x_{34}) \in \mathbb{R}^5_{\geq 0}$ as

$$\frac{z_2}{2z_2 + z_3} = x_{22},$$
$$\frac{z_3}{2z_2 + z_3} = x_{23},$$
$$\frac{z_1}{z_1 + 2z_3 + 4z_4} = x_{31},$$
$$\frac{z_3}{z_1 + 2z_3 + 4z_4} = x_{33}, \text{ and}$$
$$\frac{z_4}{z_1 + 2z_3 + 4z_4} = x_{34}.$$

We get natural constraints between the new variables:

$$2x_{22} + x_{23} = 1 \text{ and}$$
$$x_{31} + 2x_{33} + 4x_{34} = 1.$$

Set $f(n_1, n_2, n_3)$ to the maximum of (5), (6), (7), and (8), dropping $\delta_{\cdot}(\cdot)$ from each. We have

$$f(n_1, n_2, n_3) = \max\{n_1, x_{31}n_3\} + x_{22}n_2 + \max\{x_{23}n_2, x_{33}n_3\} + x_{34}n_3.$$

Using $\boldsymbol{v}^{(1)} = (1, 0, 1)$, $\boldsymbol{v}^{(2)} = (0, 2, 0)$, and $\boldsymbol{v}^{(3)} = (0, 0, 4)$, we obtain

$$R_{DAR} = \max\{f(1, 0, 1), f(0, 2, 0), f(0, 0, 4)\}$$

by Theorem 1, where

$$f(1, 0, 1) = \max\{1, x_{31}\} + x_{33} + x_{34},$$
$$f(0, 2, 0) = 2x_{22} + 2x_{33}, \text{ and}$$
$$f(0, 0, 4) = 4x_{31} + 4x_{33} + 4x_{34}.$$

This leads us to a linear optimization problem (linear program) ($\mathcal{P}$) for finding $(x_{22}, x_{23}, x_{31}, x_{33}, x_{34})$ that minimizes R_{DAR}. The value of R_{DAR} is represented by variable r.

$$\begin{aligned}
(\mathcal{P}) \quad \text{find} \quad & (x_{22}, x_{23}, x_{31}, x_{33}, x_{34}, r) \\
\text{minimize} \quad & r \\
\text{subject to} \quad & 1 + x_{33} + x_{34} \leq r \\
& x_{31} + x_{33} + x_{34} \leq r \\
& 2x_{22} + 2x_{23} \leq r \\
& 4x_{31} + 4x_{33} + 4x_{34} \leq r \\
& 2x_{22} + x_{23} = 1 \\
& x_{31} + 2x_{33} + 4x_{34} = 1 \\
& x_{22}, x_{23}, x_{31}, x_{33}, x_{34}, r \geq 0
\end{aligned}$$

Solving ($\mathcal{P}$), we get an optimal solution $(\bar{x}_{22}, \bar{x}_{23}, \bar{x}_{31}, \bar{x}_{33}, \bar{x}_{34}, \bar{r}) = \left(\frac{1}{2}, 0, \frac{1}{13}, 0, \frac{3}{13}, \frac{16}{13}\right)$. From this, we have an optimal parameter, for example, $(\bar{z}_1, \bar{z}_2, \bar{z}_3, \bar{z}_4) = (1, 1, 0, 3)$. Then, $R_{DAR} = \frac{16}{13} \approx 1.23 < 1.25 = R_{HARMONIC}$, which means that algorithm DAR with an optimal parameter is slightly better than algorithm HARMONIC. The result establishes an upper bound on the asymptotic approximation ratio of an optimal online algorithm.

Theorem 2. *For the bin packing problem in which the size of each item is an element of* $\left(\frac{3}{4}, \frac{2}{4}, \frac{1}{4}\right)$*, it holds that* $R \leq \frac{16}{13}$.

5. Concluding Remarks

We have shown that our theorem that identifies the asymptotic approximation ratio serves as a powerful tool for designing online algorithms. Our next work is to enlarge the

applicable family of item sizes and online algorithms. It seems easier to weaken the condition for online algorithms. It would be interesting to consider the case where the number of used bins has some good properties as a function, such as convexity, homogeneity, or scalability.

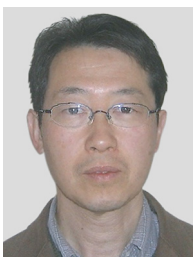
We remark that algorithm DAR is a hopeless candidate for the optimal online algorithm for the original bin packing problem that admits arbitrary item sizes. This is because, as we mentioned, algorithm DAR is a bounded-space algorithm and the asymptotic approximation ratio of any bounded-space algorithm is known to be at least 1.69103 [16]. Therefore, to this end, some other algorithm should be considered.

References

[1] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman & Co., New York, NY, USA, 1979.
[2] B. Korte and J. Vygen, Combinatorial Optimization: Theory and Algorithms, 5th ed., Springer, 2012.
[3] J. Csirik and G.J. Woeginger, "On-line packing and covering problems," Online Algorithms, ed. A. Fiat and G.J. Woeginger, LNCS, vol.1442, pp.147–177, Springer, 1998.
[4] J. Balogh, J. Békési, G. Dósa, L. Epstein, and A. Levin, "A new and improved algorithm for online bin packing," Proc. ESA '18, pp.5:1–5:14, 2018.
[5] J. Balogh, J. Békési, and G. Galambos, "New lower bounds for certain classes of bin packing algorithms," Theor. Comput. Sci., vol.440–441, pp.1–13, 2012.
[6] G. Gutin, T. Jensen, and A. Yeo, "On-line bin packing with two item sizes," Algorithmic Operations Research, vol.1, no.2, pp.72–78, 2006.
[7] L. Epstein and A. Levin, "More on online bin packing with two item sizes," Discret. Optim., vol.5, no.4, pp.705–713, 2008.
[8] D.S. Johnson, Near-optimal Bin Packing Algorithms, PhD thesis, Massachusetts Institute of Technology, 1973.
[9] D.S. Johnson, A. Demers, J.D. Ullman, M.R. Garey, and R.L. Graham, "Worst-case performance bounds for simple one-dimensional packing algorithms," SIAM J. Computing, vol.3, no.4, pp.299–325, 1974.
[10] S.S. Seiden, "On the online bin packing problem," J. ACM, vol.49, no.5, pp.640–671, 2002.
[11] S.T. McCormick, S.R. Smallwood, and F.C.R. Spieksma, "A polynomial algorithm for multiprocessor scheduling with two job lengths," Math. Oper. Res., vol.26, no.1, pp.31–49, 2001.
[12] M.X. Goemans and T. Rothvoß, "Polynomiality for bin packing with a constant number of item types," Proc. SODA '14, pp.830–839, 2014.
[13] "Packing tips: Packing material and tips." https://www.dhl.de/en/privatkunden/pakete-versenden/pakete-abgeben/verpacken.html
[14] "Boxes for packing, shipping & moving." https://www.fedex.com/en-us/shipping/packing/supplies/boxes.html
[15] G.M. Ziegler, Lectures on Polytopes, Graduate Texts in Mathematics, vol.152, Springer New York, 2012.
[16] C.C. Lee and D.T. Lee, "A simple on-line bin-packing algorithm," J. ACM, vol.32, no.3, pp.562–572, 1985.
[17] A.C.-C. Yao, "New algorithms for bin packing," J. ACM, vol.27, no.2, pp.207–227, 1980.

Hiroshi Fujiwara received his B.E., Master of Informatics, and Ph.D. in Informatics degrees from Kyoto University in 2001, 2003, and 2006, respectively. Since 2014 he has been an associate professor in Department of Electrical and Computer Engineering at Shinshu University, Japan, after working at Kwansei Gakuin University and Toyohashi University of Technology. During his career, he worked as a research assistant at University of Freiburg, Germany, from 2004 to 2005, and worked as a visiting associate professor at University of Electronic Science and Technology, China, in 2012. His research interests include analysis of algorithms, online optimization, and functional optimization.

Yuta Wanikawa received his B.E. and M.E. degrees from Shinshu University in 2017 and 2019, respectively. He joined VOYAGE GROUP, Inc. in 2019. His research interests include design and analysis of algorithms and combinatorial optimization.



Hiroaki Yamamoto received the B.E. degree in Information Engineering from Shinshu University in 1980, and the M.E. and Ph.D. degrees in Information Engineering from Tohoku University in 1982 and 1985, respectively. In 1988, he joined Shinshu University, and is currently a professor at the same university. His research interests include algorithm theory, automata theory, information security, and information retrieval. He is a member of the IEICE and the IPSJ.