All-Around Near-Optimal Solutions for the Online Bin Packing Problem

Shahin Kamali[1(✉)] and Alejandro López-Ortiz[2]

[1] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
skamali@mit.edu
[2] University of Waterloo, Waterloo, ON N2L 3G1, Canada
alopez-o@uwaterloo.ca

Abstract. In this paper we present algorithms with optimal average-case and close-to-best known worst-case performance for the classic online bin packing problem. It has long been observed that known bin packing algorithms with optimal average-case performance are not optimal in the worst-case. In particular First Fit and Best Fit have optimal asymptotic average-case ratio of 1 but a worst-case competitive ratio of 1.7. The competitive ratio can be improved to 1.691 using the Harmonic algorithm. Further variations of this algorithm can push down the competitive ratio to 1.588. However, these algorithms have poor performance on average; in particular, Harmonic algorithm has average-case ratio of 1.27. In this paper, first we introduce a simple algorithm which we term Harmonic Match. This algorithm performs as well as Best Fit on average, i.e., it has an average-case ratio of 1. Moreover, the competitive ratio of the algorithm is as good as Harmonic, i.e., it converges to 1.691 which is an improvement over Best Fit and First Fit. We also introduce a different algorithm, termed as Refined Harmonic Match, which achieves an improved competitive ratio of 1.636 while maintaining the good average-case performance of Harmonic Match and Best Fit. Our experimental evaluations show that our proposed algorithms have comparable average-case performance with Best Fit and First Fit, and this holds also for sequences that follow distributions other than the uniform distribution.

1 Introduction

An instance of the online bin packing problem is defined by a sequence $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$ of *items* each having a *size* in the range $(0, 1]$. Items arrive one by one, and an algorithm should take an irrecoverable decision by placing each item into a bin without any knowledge about the forthcoming items. The goal is to pack items into a minimum number of bins of uniform capacity. Next Fit (NF) algorithm keeps one open bin. If an item does not fit in the open bin, it gets closed and a new bin is opened. First Fit algorithm (FF) maintains bins in the order they are opened and places each item in the first bin with enough space. If such a bin does not exist, a new bin is opened. Best Fit (BF) performs similarly to FF, except that it maintains bins in the decreasing order of their *levels*, where the level of a bin is the total size of items in it. An alternative approach is to partition items into a fixed number of classes and pack items of each class apart from other classes. An example is the Harmonic (HA) algorithm which defines K intervals

$(1/2, 1], (1/3, 1/2], \ldots, (1/K, 1/(K-1)]$, and $(0, 1/K]$. Items with sizes in the same interval are treated separately using the Next Fit strategy.

Bin packing algorithms are usually compared through their average-case and worst-case performance. Under average-case analysis, it is assumed that item sizes are generated independently at random and follow a fixed distribution that is typically the uniform distribution over the interval $[0, 1)$. With this assumption, one can define the *asymptotic average-case performance ratio*, or simply *average ratio*, of an online algorithm $\mathbb{A}$ as $\lim_{n\to\infty} E\left[\frac{A(\sigma_{(n)})}{\mathrm{OPT}(\sigma_{(n)})}\right]$, where $\sigma_{(n)}$ is a randomly generated sequence of length n and $A(\sigma)$ denotes the number of bins used by $\mathbb{A}$ for packing σ (the same notation is used for OPT). Next Fit has average ratio of $4/3$ [5] while First Fit and Best Fit both have optimal average ratio of 1 [2]. To compare algorithms with average ratio of 1, a more precise measure of *expected waste* is defined as $E[A(\sigma_{(n)}) - s(\sigma_{(n)})]$, where $s(\sigma_{(n)})$ denotes the total size of items in $\sigma_{(n)}$. First Fit and Best Fit have expected waste of $\Theta(n^{2/3})$ and $\Theta(\sqrt{n}\lg^{3/4} n)$, respectively [21,16]. All online algorithms have expected waste of size $\Omega(\sqrt{n}\lg^{1/2} n)$ [21].

There are algorithms which are based on matching a "large" item with a "small". Throughout the paper, we call an item large if it is larger than 1/2 and small otherwise. Interval First Fit (IFF) algorithm [9] divides the unit interval into K intervals of equal length, namely $I_t = (\frac{t-1}{K}, \frac{t}{K}]$ for $t = 1, 2, \ldots, K$, where $K = 2j + 1$ is an odd integer. The algorithm defines $j + 1$ classes so that intervals I_τ and $I_{K-\tau}$ form class τ ($1 \leq \tau \leq j$) and interval I_K forms class $j + 1$. Items in each class are packed separately using a strategy similar to First Fit. Algorithm Online Match (OM) [7] also has a parameter K and declares two items as being *companions* if their sum is in the range $[1-\frac{1}{K}, 1]$. A new bin is opened for each large item. For placing a small item x, the algorithm checks whether there is an open bin β with a large companion of x; in case there is, it places x in β and closes β. Otherwise, it packs x using the NF strategy in a separate list of bins. Matching Best Fit (MBF) algorithm is similar to Best Fit except that it closes a bin as soon as it receives the first small item. There is an online algorithm with expected waste of size $\Theta(\sqrt{n}\lg^{1/2} n)$ [22] which matches the lower bound of [21]. The above matching algorithms have promising average-case performance; however, they perform poorly in the worst case (see Table 1).

Competitive analysis is the standard worst-case measure for comparing online algorithms. Throughout the paper, by 'competitive ratio' of an online algorithm $\mathbb{A}$, we mean '*asymptotic* competitive ratio' of $\mathbb{A}$, which is defined as $inf\{r \geq 1 : \text{for some } N > 0, A(\sigma)/\mathrm{OPT}(\sigma) \leq r \text{ for all } \sigma \text{ with } \mathrm{OPT}(\sigma) \geq N\}$. Next Fit has a competitive ratio of 2 while First Fit and Best Fit have the same ratio of 1.7 [11]. For large values of K, the competitive ratio of HA approaches to $T_\infty = \sum_{i=1}^{\infty} \frac{1}{t_i - 1}$, where $t_1 = 2$ and $t_{i+1} = t_i(t_i - 1) + 1, i \geq 1$. Members of a general framework of Super Harmonic algorithms [20] have even better competitive ratios. Similar to HA, these algorithms classify items by their sizes and pack items of the same class together. To improve over HA, a fraction of opened bins include items from different classes. These bins are opened with items of small sizes in the hopes of subsequently adding items of larger sizes. At the time of opening a bin, it is pre-determined how many items from each class should be placed in the bin, and it is guaranteed that the reserved spot is enough for any member of the class. Hence, the expected total size of items in the bin is less than 1, and the ex-

Algorithm	Average Ratio	Expected waste	Competitive Ratio
Next Fit (NF)	$1.\bar{3}$ [5]	$\Omega(n)$	2
Best Fit (BF)	1 [2]	$\Theta(\sqrt{n}\lg^{3/4} n)$ [21,16]	1.7 [11]
First Fit (FF)	1 [16]	$\Theta(n^{2/3})$ [21,6]	1.7 [11]
Harmonic (HA)	1.2899 [15]	$\Omega(n)$	$\to T_\infty \approx 1.691$ [14]
Refined First Fit (RFF)	> 1	$\Omega(n)$	1.66 [23]
Refined Harmonic (RH)	1.2824 [10]	$\Omega(n)$	1.636 [14,10]
Modified Harmonic (MH)	1.189 [17]	$\Omega(n)$	1.615[18]
Harmonic++	> 1	$\Omega(n)$	1.588 [20]
Harmonic Match HM	$\mathbf{1}$	$\mathbf{\Theta(\sqrt{n}\lg^{3/4} n)}$	$\mathbf{\to T_\infty \approx 1.691}$
Refined Harmonic Match (RHM)	$\mathbf{1}$	$\mathbf{\Theta(\sqrt{n}\lg^{3/4} n)}$	$\mathbf{1.636}$

Table 1: Average ratio, expected waste (under continuous uniform distribution), and competitive ratios for bin packing algorithms. Results in bold are our contributions.

pected waste is linear to the number of opened bins. This implies that the average ratio of Super Harmonic algorithms is strictly larger than 1. Regarding the lower bound for competitive ratio of online algorithms, Balogh et al. [1] proved that no online algorithm can have a competitive ratio better than 1.54037. Table 1 includes a summary of the performance of bin packing algorithms.

In their survey of bin packing, Coffman et al. [4] state that *'All algorithms that do better than First Fit in the worst-case seem to do much worse in the average-case.'* In this paper, however, we show that this is not necessarily true and introduce an algorithm whose competitive ratio, average ratio, and expected wasted space are all at or near the top of each class. This also addresses a conjecture by Gu et al. [10] stated as *'Harmonic is better than First Fit in the worst-case performance, and First Fit is better than Harmonic in the average-case performance. Maybe there exists an on-line algorithm with the advantages of both First Fit and Harmonic.'*

1.1 Contribution

We introduce an algorithm called Harmonic Match (HM) which has a competitive ratio similar to Harmonic, i.e., approaches $T_\infty \approx 1.691$ for large values of K, where K is a parameter of the algorithm. For sequences generated uniformly and independently at random, Harmonic Match has an optimal average ratio of 1 and expected waste of $\Theta(\sqrt{n}\lg^{3/4} n)$ which is as good as Best Fit and better than First Fit. The idea behind Harmonic Match can be used in a general way to improve Super Harmonic algorithms. We illustrate this for the simplest member of this family, namely the Refined Harmonic algorithm of Lee and Lee [14]. We introduce a new algorithm called Refined Harmonic Match (RHM), which has a competitive ratio of at most 1.636. At the same time, the average ratio and expected waste of RHM are as good as those of Best Fit.

Harmonic Match and Refined Harmonic Match are easy-to-implement, and their running time is as good as Best Fit. This makes them useful in practical scenarios in which the worst-case scenarios might indeed happen. One example is the denial of service attacks in cloud [13] in which an adversary sends items (jobs or 'tenants) that form a worst-case sequence. In these cases, the advantage of RHM over Best Fit is significant from the perspective of cloud service providers. Although the analysis techniques used in this paper are straightforward, we use them to prove an important result that shows the average performance does not need to be compromised for better competitive ratios. For the bulk of this paper, we assume item sizes are distributed uniformly and independently in the interval $(0, 1]$. However, for a better picture on the average-case performance, we test them on sequences that follow other distributions. The results of our experiments suggest that Harmonic Match and Refined Harmonic Match have comparable performance with Best Fit and First Fit. At the same time, they have a considerable advantage over other members of the Harmonic family of algorithms. Due to space restrictions, many proofs have been removed. They will appear in the long version of the paper.

2 Harmonic Match Algorithm

Similarly to Harmonic algorithm, Harmonic Match has a parameter K and divides items into K classes based on their sizes. We use $\textsc{Hm}_K$ to refer to Harmonic Match with parameter K. The algorithm defines K pairs of intervals as follows. The i-th pair $(1 \leq i \leq K-1)$ contains intervals $(\frac{1}{i+2}, \frac{1}{i+1}]$ and $(\frac{i}{i+1}, \frac{i+1}{i+2}]$. The K-th pair includes intervals $(0, \frac{1}{K+1}]$ and $(\frac{K}{K+1}, 1]$. An item x belongs to class i if the size of x lies in any of the two intervals associated with the i-th pair. Note that the intervals in $\textsc{Hm}_K$ are the same as Harmonic with parameter $K+1$ except that the interval $(\frac{1}{2}, 1]$ in the Harmonic algorithm is further divided into $K+1$ more intervals in Harmonic Match. This division enables "matching" large items with proportionally smaller items. The pair of intervals which form a class have the same length. This is essential for a good average-case performance for our uniform distribution on $(0, 1]$. The algorithm applies a strategy similar to Best Fit to place items inside each class. The Harmonic-type classification of items allows improvement on the competitive ratio.

The packing maintained by Harmonic Match includes two types of bins: the "mature" bins which are almost full and "normal" bins which become mature by receiving more items. For placing an item x, HM detects the class that x belongs to and applies the following strategy to place x. If x is a large item ($x > 1/2$), the algorithm opens a new bin and declares it as a normal bin. If x is small ($x \leq 1/2$), the algorithm applies the Best Fit (BF) strategy to place x in a mature bin. If there is no mature bin with enough space, the BF strategy is applied one more time to place x in a normal bin that contains the largest "companion" of x. A companion of x is a large item of the same class that fits with x in the same bin. In case x is placed in a bin (i.e., there is a normal bin with a companion of x) the selected bin is declared as a mature bin. Otherwise, the algorithm applies the Next Fit (NF) strategy to place x in a single normal bin maintained for that class; such a bin includes small items of the same class. If the bin maintained by NF does not have enough space, it is declared as a mature bin and a new NF-bin is opened.

Harmonic Match treats items of the same class in a similar way that Online Match does except that there is no restriction on the sum of the sizes of two companion items. To facilitate our analysis, we introduce the Relaxed Online Match (ROM) algorithm as a subroutine of HM. To place a large item, ROM opens a new bin. To place a small item x, it applies the Best Fit strategy to place x in an open bin with a single large item and closes the bin. If such a bin does not exists, ROM places x using the Next Fit strategy (and opens a new bin if necessary). Using ROM, we can describe the Harmonic Match algorithm in the following way. To place a small item, $\textsc{Hm}_K$ applies the Best Fit strategy to place it in a mature bin. Large items and the small items which do not fit in mature bins are treated using the ROM strategy along with other items of their classes. The bins which are closed by the ROM strategy are declared as mature bins.

2.1 Worst-Case Analysis

To analyze Harmonic Match, we observe that the classic Harmonic algorithm is *monotone* in the sense that removing an item does not increase the number of bins it opens.

Lemma 1. *Removing any item from an input sequence σ does not increase the number of bins used by the Harmonic algorithm for packing σ.*

Using the above lemma, we show that the number of bins used by $\textsc{Hm}_K$ for any sequence is no larger than that of Harmonic with parameter $K+1$ ($\textsc{Ha}_{K+1}$). Informally speaking, the small items which are placed with large items in $\textsc{Hm}_K$ can be thought as being "removed" from the packing of Harmonic.

Lemma 2. *The number of bins used by Harmonic Match with parameter K ($\textsc{Hm}_K$) to pack any sequence σ is no larger than that of Harmonic with parameter K ($\textsc{Ha}_{K+1}$).*

Proof. We say a small item is *red* if it is placed in a bin with a large item in the packing of $\textsc{Hm}_K$, and call it *white* otherwise. Consider a subsequence σ^- of σ in which red items are removed. We show $\textsc{Hm}_K(\sigma) = \textsc{Ha}_{K+1}(\sigma^-)$. Let σ_i denote the sequence formed by items of class i in $\textsc{Hm}_K$ ($1 \leq i \leq K$). The number of bins opened by $\textsc{Hm}_K$ for σ_i is $l_i + \textsc{Nf}(W_i)$ where l_i is the number of large items of σ_i and W_i is the sequence formed by the white items in σ_i. Let σ_i^- be a subsequence of σ_i in which red items are removed. Since small and large items are treated separately by $\textsc{Ha}_{K+1}$, the number of bins used by $\textsc{Ha}_{K+1}$ for σ_i^- is also $l_i + \textsc{Nf}(W_i)$, and we have $\textsc{Hm}_K(\sigma_i) = \textsc{Ha}_{K+1}(\sigma_i^-)$. Taking the sum over all classes, we get $\textsc{Hm}_K(\sigma) = \textsc{Ha}_{K+1}(\sigma^-)$. Since HA is monotone by Lemma 1, we have $\textsc{Ha}_{K+1}(\sigma^-) \leq \textsc{Ha}_{K+1}(\sigma)$, and $\textsc{Hm}_K(\sigma) \leq \textsc{Ha}_{K+1}(\sigma)$. □

For large values of K, the competitive ratio of Harmonic Match approaches $T_\infty \approx 1.691$. Indeed, the above upper bound is tight and we get the following result.

Theorem 1. *The competitive ratio of $\textsc{Hm}_K$ is equal to that of $\textsc{Ha}_{K+1}$, i.e., it converges to $T_\infty \approx 1.691$ for large values of K.*

2.2 Average-Case Analysis

We study the average-case performance of the HM algorithm, assuming item sizes are distributed uniformly in the interval $(0, 1]$. Like most related work, we make use of the results related to the *up-right matching* problem. An instance of this problem includes n points generated uniformly and independently at random in a unit-square in the plane. Each point receives a $\oplus$ or $\ominus$ label with equal probability. The goal is to find a maximum matching of $\oplus$ points with $\ominus$ points so that in each pair of matched points the $\oplus$ point appears above and to the right of the $\ominus$ point. Let U_n denote the number of unmatched points in an optimal up-right matching of n points. For the expected size of U_n, it is known that $E[U_n] = \Theta(\sqrt{n} \lg^{3/4} n)$ [21,16,19,8]. Given an instance of bin packing defined by a sequence σ, one can make an instance of up-right matching as follows [12]. Each item x of size $s(x)$ in σ is plotted as a point in the unit square. The vertical coordinate of the point corresponds to the index of x in σ (scaled to fit in the square). If x is smaller than $1/2$, the point is labelled as $\oplus$ and its horizontal coordinate will be $1 - 2s(x)$ where $s(x)$ is the size of x; otherwise, the point will be $\ominus$ and its horizontal coordinate will be $2s(x) - 1$. A solution to the up-right matching instance gives a packing of σ in which the items associated with a pair of matched points are placed in the same bin. Note that the sum of the sizes of these two items is no more than the bin capacity. Also, in such a solution, each bin contains at most two items.

For our purposes, we study σ_t as a subsequence of σ which only includes items which belong to the same class in the HM algorithm. The items in σ_t are generated uniformly at random from $(\frac{1}{t+1}, \frac{1}{t}] \cup (\frac{t-1}{t}, \frac{t}{t+1}]$ where t is a positive integer. Since the two intervals have the same length, as we will describe, the items can be plotted in a similar manner on the unit square. Any bin packing algorithm which closes a bin after placing a small item can be used for the up-right matching problem. Each edge in the matching instance corresponds to a bin which includes one small and one large item. Recall that the algorithm Matching Best Fit (MBF) is similar to Best Fit except that it closes a bin as soon as it receives an item with size smaller than or equal to 1/2. So, MBF can be applied for the up-right matching problem. Indeed, it creates an optimal up-right matching, i.e., if we apply MBF on a sequence σ_t which is randomly generated from $(0, 1]$, the number of unmatched points will be $\Theta(\sqrt{n_t} \lg^{3/4} n_t)$, where n_t is the length of σ_t [21]. We show the same result holds for the bin packing sequences in which items are taken uniformly at random from $(\frac{1}{t+1}, \frac{1}{t}] \cup (\frac{t-1}{t}, \frac{t}{t+1}]$.

Lemma 3. *For a sequence σ_t of length n_t in which item sizes are selected uniformly at random from $(\frac{1}{t+1}, \frac{1}{t}] \cup (\frac{t-1}{t}, \frac{t}{t+1}]$, we have $E[\text{MBF}(\sigma_t)] = n_t/2 + \Theta(\sqrt{n_t} \lg^{3/4} n_t)$.*

Proof. Define an instance of up-right matching as follows. Let x, with size $s(x)$, be the i-th item of σ_t $(1 \leq i \leq n_t)$. If x is small, plot a point with $\oplus$ label at position $(1 - (s(x) \times t(t+1) - t), i/n_t)$; otherwise, plot a point with $\ominus$ label at position $(s(x) \times t(t+1) - (t^2 - 1), i/n_t)$. This way, the points will be bounded in a unit square. Since item sizes are generated uniformly at random from the two intervals and the sizes of the intervals are the same, the point locations and labels are assigned uniformly and independently at random. Hence, the number of unmatched points in the up-right matching solution by MBF is expected to be $\Theta(\sqrt{n_t} \lg^{3/4} n_t)$. The unmatched points are associated with the items in σ_t which are packed as a single item in their bins by MBF. Let sg denote the number of such items. We have $E[sg] = \Theta(\sqrt{n_t} \lg^{3/4} n_t)$. Except these sg items, other items are packed with exactly one other item in the same bin. So we have $\text{MBF}(\sigma_t) - sg = n_t/2$ which implies $E[\text{MBF}(\sigma_t)] = n_t/2 + E[sg] = n_t/2 + \Theta(\sqrt{n_t} \lg^{3/4} n_t)$. □

Recall that ROM is a subroutine of HM. The main difference between ROM and MBF is in placing small items without companions. For those, ROM applies the NF strategy while MBF opens a new bin for each item. Clearly, ROM has an advantage.

Lemma 4. *For any instance σ of the bin packing problem, the number of bins used by* ROM *to pack σ is no more than that of* MBF*.*

To prove the main result, we also need to show that MBF is monotone:

Lemma 5. *Removing an item does not increase the number of bins used by* MBF*.*

Provided with the above lemmas, we prove the main result of this section.

Theorem 2. *For packing a sequence σ of length n in which item sizes are selected uniformly at random from $(0, 1]$, the expected wasted space of* HM *is $\Theta(\sqrt{n} \lg^{3/4} n)$.*

Proof. Let σ^- be a copy of σ in which the items which are placed in mature bins are removed. Let $\sigma_1^-, \ldots, \sigma_K^-$ be the subsequences of σ^- formed by items belonging to different classes of HM. We have:

$$\text{HM}(\sigma) = \sum_{t=1}^{K} \text{ROM}(\sigma_t^-) \leq \sum_{t=1}^{K} \text{MBF}(\sigma_t^-) \leq \sum_{t=1}^{K} \text{MBF}(\sigma_t)$$

The inequalities come from Lemmas 4 and 5, respectively. By Lemma 3, we have:

$$E[\text{HM}(\sigma)] \leq \sum_{t=1}^{K} \left(n_t/2 + \Theta(\sqrt{n_t} \lg^{3/4} n_t) \right) = \frac{n}{2} + \Theta(\sqrt{n} \lg^{3/4} n)$$

The last equation holds since K is a constant. The expected value of $s(\sigma)$, the total size of items in σ, is $n/2$. Consequently, for the expected waste of HM , we have the following equality which completes the proof:

$$E[\text{HM}(\sigma) - s(\sigma)] = n/2 + \Theta(\sqrt{n} \lg^{3/4} n) - n/2 = \Theta(\sqrt{n} \lg^{3/4} n) \quad \square$$

3 Refined Harmonic Match

In this section, we introduce a slightly more complicated algorithm, called Refined Harmonic Match (RHM), which has a better competitive ratio than BF and HM while performing as well as them on average. Similar to HM, RHM classifies items based on their sizes. The classes defined for RHM are the same as those of HM_K with $K = 19$. The items which belong to class $t \geq 2$ are treated using the HM strategy. Namely, a set of mature bins are maintained. If an item fits in mature bins, it is placed there using the BF strategy; otherwise, it is placed together with similar items of its class using the ROM strategy. At the same time, the bins closed by the ROM strategy are declared as being mature. The only difference between HM and RHM in packing items of class 1, i.e., items in the range $(1/3, 2/3]$. RHM divides these items into four groups $a = (1/3, 37/96]$, $b = (37/96, 1/2]$, $c = (1/2, 59/96]$, and $d = (59/96, 2/3]$. To handle the sequences which result in the lower bound of T_∞ for competitive ratios of HA and HM, RHM designates a fraction of bins opened by items of type a to host the future c items. Note that the total size of a c item and an a item is no more than 1.

In what follows, we introduce an online algorithm called Refined Relaxed Online Match (RRM) as a subroutine of RHM that is specifically used for placing items of class 1. At each step of the algorithm, when two items of class 1 are placed in the same bin, that bin is declared to be mature and will be added to the set of mature bins maintained by the HM algorithm that packs items of other classes. RRM uses the following strategy to place an item x of class 1 ($x \in (1/3, 2/3]$). If x is a d-item, RRM opens a new bin for x. If x is a c item, the algorithms checks whether there are bins with an a item designated to be paired with a c item. In case there are, x is placed in a bin with an a item using the BF strategy; otherwise, a new bin is opened for x. For a and b items (small items of class 1), RRM uses the BF strategy to select a bin with enough space which includes a single large item (if there is such a bin). This is particularly important to guarantee a good average-case behavior. If x is a b item, the algorithms checks the bin with the highest level in which x fits; if such a bin includes a c or a b item, x is placed there. Otherwise (when there is no selected bin or when it has an a item), a new bin is opened for x. If x is an a item, the algorithm uses the BF strategy to place it into a bin with a d or c item. If no suitable bin exist, x is placed in a bin with a single a item (there is at most one such bin). If there is no such bin, a new bin is opened for x.

When a new bin is opened for an a-item, the bin will be marked to either include a c item or another a item in the future. We define A-bins as those which include two a items or a single a item designated to be paired with another a item, and define C-bins as those which include either a c item together with an a or a b item or a single a item designated to be paired with a c item in the future. RHM tries to maintain the number of A-bins as close to three times the number of C-bins as possible. Namely, when a bin is opened for an a item, if the number of A-bins is less than 3 times of C-bins, the bin is declared as an A-bin to host another a item later; otherwise, the open bin is declared as a C-bin to host a c item. This way, the number of A-bins is close to (but no more than) 3 times that of C-bins.

3.1 Worst-Case Analysis

In this section, we prove an upper bound of 1.636 for the competitive ratio of RHM. Since RHM applies HM for placing items of class $t \geq 2$, by Lemma 2, the number of bins opened by RHM for these items is no more than that of Harmonic. An analysis of the number of bins opened by the Harmonic algorithm gives the following lemma.

Lemma 6. *For the number of bins used by* RHM *to pack a sequence σ we have*

$$\mathrm{RHM}(\sigma) \leq \mathrm{RRM}(\sigma_{cl_1}) + n_X + \sum_{t=2}^{18} \lfloor \tfrac{n_t}{t+1} \rfloor + 20W'/19 + 20$$

in which σ_{cl_1} is the subsequence formed by items of class 1, n_X is the number of large items in classes other than class 1, n_t is the number of small items in class t, and W' is the total size of small items in class 19 (the last class).

Using the above lemma, we prove the following theorem.

Theorem 3. *The competitive ratio of* RHM *is at most* $373/228 < 1.636$.

To prove the theorem, in the packing of RRM for items of class 1, we define a_1-bins as those which only include one a-item designated to be paired with a c-item. We consider the following two cases and prove the theorem for each case separately.

– Case 1: There is at least one a_1-bin in the final packing.
– Case 2: There is no a_1-bin in the final packing.

Let $n_\tau(\tau \in \{a,b,c,d\})$ denote the number of items of class q in the input sequence. In both cases, we formulate the number of bins opened by RRM as a function of the number of items in each group (i.e., as a function of n_a, n_b, n_c, and n_d). By definition of RRM, no c-bin and a_1-bin can exist at the same time. So, in Case 1, there is no c-bin in the packing. We can bound the number of C-bins by proving the inequality $3N_\mathrm{C} \leq N_\mathrm{A} + 3$ where N_C and N_A respectively denote the number of C-bins and A-bins. Using the definition of A-bins and C-bins, we show the number of bins opened by RRM is at most $n_d + 4n_a/7 + 4n_b/7 + 1$. Plugging this to Lemma 6 and applying a straightforward weighting function similar to that of Lee and Lee [14] completes the proof. In Case 2, we note that $N_\mathrm{A} \leq 3N_\mathrm{C}$ and use it to show the number of bins opened by RRM is at most $n_d + n_c + n_b/2 + 3n_a/7 + 2$. Applying another weighting function completes the proof. The details will appear in the long version of the paper.

3.2 Average-Case Analysis

We show that the average-case performance of RHM is as good as BF and HM. Except the following lemma, other aspects of the proof are similar to those in Section 2.2.

Lemma 7. *For any instance σ of the bin packing problem in which items are in the range $(1/3, 2/3]$, the number of bins used by* RRM *to pack σ is no more than that of Matching Best Fit (*MBF*).*

The key observation in the proof is that RRM uses the BF strategy to place a small item x in a bin which includes a large item. Note that small items are a and b items in the RRM algorithm. Only if such a bin does not exist, RRM deviates from the BF strategy (this is the main difference between RRM and Refined Harmonic of [14]). Given Lemma 7, a similar argument as the proof of Theorem 2 results in the following theorem.

Theorem 4. *For a sequence σ of length n in which item sizes are selected uniformly at random from $(0,1]$, the expected wasted space of* RHM *is $\Theta(\sqrt{n}\lg^{3/4} n)$.*

4 Experimental Evaluation

The results of the previous sections indicate that HM and RHM have similar average-case performance as BF if item sizes are taken uniformly at random from the range $(0,1]$. In this section, we expand the range of distributions beyond this distribution to further observe the performance of these algorithms. For that, we considered uniform distribution with different ranges for items sizes (ranges $(0,1/2]$ and $(0,1/10]$), as well as Normal and Weibull distributions with different parameters. We also considered uniform instances in which items are sorted in decreasing order of their sizes. The details about these distributions can be found in the long version of the paper. For all distributions, we computed the average number of bins used by different algorithms for packing 1000 sequences of length 100,000. For algorithms that classify items by their sizes, the number of classes K is set to 20.

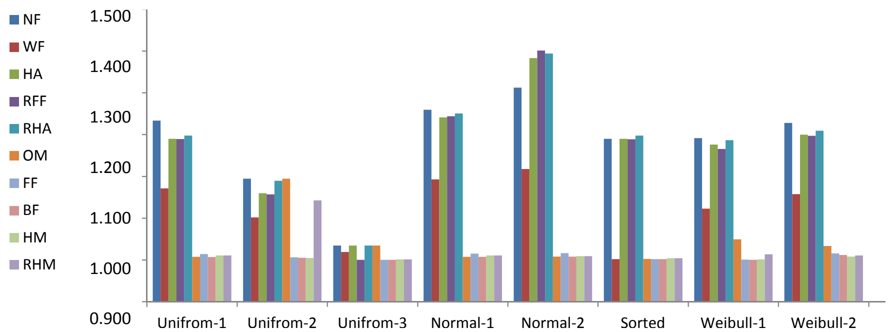

Fig. 1: The bar chart for the experimental average ratios of online bin packing algorithms. To make the results more visible, the vertical scale is changed to start at 0.9.

We compute the *experimental average ratio* of an algorithm as the ratio between the observed expected number of bins used by the algorithm and that of OPT. We estimate the number of bins opened by OPT to be the total size of items. Figure 1 shows the bar chart for experimental average ratio of different online algorithms. It can be seen that HM and RHM, along with BF and FF, have a significant advantage over other algorithms.

A difference between the packings of HM and RHM occurs when a number of small items of the first class (items of type a in RHM) appear before any large item of the same class (an item of type c). In these cases, RHM reserves some bins for subsequent large items (by declaring the bins to be C-bins). For symmetric distributions, where items of sizes x and $1-x$ appear with the same probability, it is unlikely that many small items appear before the next large item. Consequently, the average number of bins used by HM and RHM are the same. On the other hand, for asymmetric sequences where small items are more likely to appear, e.g., Uniform-2 with item sizes in the range $(0, 1/2]$, HM has a visible advantage over RHM. In these sequences, there is no reason to reserve bins for the large items since they are unlikely to appear.

5 Remarks

HM and RHM can be seen as variants of Harmonic and Refined Harmonic algorithms in which small and large items are carefully matched in order to improve the average-case performance. We believe that the same approach can be applied to improve the average performance of other Super Harmonic algorithms and in particular that of Harmonic++. Given the complicated nature of these algorithms, modifying them involves a detailed analysis which we leave as a future work.

It is possible to study the performance of bin packing algorithms using the relative worst order analysis [3]. Under this measure, when all items are larger than $\frac{1}{K+1}$, Harmonic with parameter K is strictly better than FF and BF by a factor of 6/5 [3]. Applying Lemma 2, when all items are larger than $\frac{1}{K+2}$, Harmonic Match with parameter K is strictly better than FF and BF. This provides another theoretical evidence for the advantage of Harmonic Match over BF and FF.

References

1. Balogh, J., Békési, J., Galambos, G.: New lower bounds for certain classes of bin packing algorithms. Theoret. Comput. Sci. 440–441, 1–13 (2012)
2. Bentley, J.L., Johnson, D.S., Leighton, F.T., McGeoch, C.C., McGeoch, L.A.: Some unexpected expected behavior results for bin packing. In: Proc. 16th Symp. on Theory of Computing (STOC). pp. 279–288 (1984)
3. Boyar, J., Favrholdt, L.M.: The relative worst order ratio for online algorithms. ACM Trans. Algorithms 3(2) (2007)
4. Coffman, E.G., Garey, M.R., Johnson, D.S.: Approximation algorithms for bin packing: A survey. In: Approximation algorithms for NP-hard Problems (1997)
5. Coffman, E.G., Hofri, M., So, K., Yao, A.C.C.: A stochastic model of bin packing. Inform. and Control 44, 105–115 (1980)
6. Coffman, E.G., Johnson, D.S., Shor, P.W., Weber, R.R.: Bin packing with discrete item sizes, part II: tight bounds on First Fit. Random Struct. Algorithms 10(1-2), 69–101 (1997)
7. Coffman, E.G., Lueker, G.S.: Probabilistic analysis of Packing and Partitioning Algorithms. John Wiley, New York (1991)
8. Coffman, E.G., Shor, P.W.: A simple proof of the $O(\sqrt{n \log^{3/4} n})$ up-right matching bound. SIAM J. Discrete Math. 4, 48–57 (1991)
9. Csirik, J., Galambos, G.: An $O(n)$ bin-packing algorithm for uniformly distributed data. Computing 36(4), 313–319 (1986)
10. Gu, X., Chen, G., Xu, Y.: Deep performance analysis of refined harmonic bin packing algorithm. J. Comput. Sci. Tech. 17, 213–218 (2002)
11. Johnson, D.S.: Near-optimal bin packing algorithms. Ph.D. thesis, MIT (1973)
12. Karp, R.M., Luby, M., Marchetti-Spaccamela, A.: Probabilistic analysis of multidimensional binpacking problems. In: Proc. 16th Symp. on Theory of Computing (STOC). pp. 289–298 (1984)
13. Kousiouris, G.: Minimizing the effect of dos attacks on elastic cloud-based applications. In: Proc. International Conference on Cloud Computing and Services Science, 2014. pp. 622–628 (2014)
14. Lee, C.C., Lee, D.T.: A simple online bin packing algorithm. J. ACM 32, 562–572 (1985)
15. Lee, C.C., Lee, D.T.: Robust online bin packing algorithms. Tech. rep., Northwestern University (1987)
16. Leighton, F.T., Shor, P.: Tight bounds for minimax grid matching with applications to the average case analysis of algorithms. Combinatorica 9, 161–187 (1989)
17. Ramanan, P., Tsuga, K.: Average-case analysis of the modified harmonic algorithm. Algorithmica 4, 519–533 (1989)
18. Ramanan., P.V., Brown, D.J., Lee, C.C., Lee, D.T.: On-line bin packing in linear time. J. Algorithms 10, 305–326 (1989)
19. Rhee, W.T., Talagrand, M.: Exact bounds for the stochastic upward matching problem. Trans. AMS 307(1), 109–125 (1988)
20. Seiden, S.S.: On the online bin packing problem. J. ACM 49, 640–671 (2002)
21. Shor, P.W.: The average-case analysis of some online algorithms for bin packing. Combinatorica 6, 179–200 (1986)
22. Shor, P.W.: How to pack better than Best-Fit: Tight bounds for average-case on-line bin packing. In: Proc. 32nd Symp. on Foundations of Computer Science (FOCS). pp. 752–759 (1991)
23. Yao, A.C.C.: New algorithms for bin packing. J. ACM 27, 207–227 (1980)