Strip Packing vs. Bin Packing

Xin Han[1]   Kazuo Iwama[1]   Deshi Ye[2] Guochuan Zhang[3]

[1] School of Informatics, Kyoto University, Kyoto 606-8501, Japan

{hanxin, iwama}@kuis.kyoto-u.ac.jp

[2] Department of Computer Science, The University of Hong Kong, Hong Kong

yedeshi@cs.hku.hk

[3] Department of Mathematics, Zhejiang University, China

zgc@zju.edu.cn

Abstract

In this paper we establish a general algorithmic framework between bin packing and strip packing, with which we achieve the same asymptotic bounds by applying bin packing algorithms to strip packing. More precisely we obtain the following results: (1) Any offline bin packing algorithm can be applied to strip packing maintaining the same asymptotic worst-case ratio. Thus using FFD (MFFD) as a subroutine, we get a practical (simple and fast) algorithm for strip packing with an upper bound 11/9 (71/60). A simple AFPTAS for strip packing immediately follows. (2) A class of Harmonic-based algorithms for bin packing can be applied to online strip packing maintaining the same asymptotic competitive ratio. It implies online strip packing admits an upper bound of 1.58889 on the asymptotic competitive ratio, which is very close to the lower bound 1.5401 and significantly improves the previously best bound of 1.6910 and affirmatively answers an open question posed [5].

1 Introduction

In strip packing a set of rectangles with widths and heights both bounded by 1, is packed into a strip with width 1 and infinite height. Rectangles must be packed such that no two rectangles overlap with each other and the sides of the rectangles are parallel to the strip sides. Rotations are not allowed. The objective is to minimize the height of the strip to pack all the given rectangles. If we know all rectangles before constructing a packing, then this problem is *offline*. In contrast in *online* strip packing rectangles are coming one by one and a placement decision for the current rectangle must be done before the next rectangle appears. Once a rectangle is packed it is never moved again.

It is well known that strip packing is a generalization of bin packing. Namely if we restrict all input rectangles to be of the same height, then strip packing is equivalent to bin packing. Thus any negative results for bin packing still hold for strip packing. More precisely, strip packing is NP-hard in the strong sense and the lower bound 1.5401 [15] is valid for online strip packing.

Previous results.   For the offline version Coffman et al. [4] presented algorithms NFDH (Next Fit Decreasing Height) and FFDH (First Fit Decreasing Height), and showed that the respective asymptotic worst-case ratios are 2 and 1.7. Golan [6] and Baker et al. [2] improved it to 4/3 and 5/4, respectively. Using linear programming and random techniques, an asymptotic fully polynomial time approximation schemes (AFPTAS) was given by Kenyon and Rémila [9]. In the online version Baker and Schwarz [3] introduced an online strip packing algorithm called a shelf algorithm. A shelf is a rectangular part of the strip with width one and height at most

one so that (i) every rectangle is either completely inside or completely outside of the shelf and (ii) every vertical line through the shelf intersects at most one rectangle. Shelf packing is an elegant idea to exploit bin packing algorithms. By employing bin packing algorithms *Next Fit* and *First Fit* Baker and Schwarz [3] obtained the asymptotic competitive ratios of 2 and 1.7, respectively. This idea was extended to the Harmonic shelf algorithm by Csirik and Woeginger [5], obtaining an asymptotic competitive ratio of $h_\infty \approx 1.6910$. Moreover it was shown that h_∞ is the best upper bound a shelf algorithm can achieve, no matter what online bin packing algorithm is used. Note that there were already several algorithms for online bin packing that have asymptotic competitive ratios better than h_∞ in late 80s and early 90s [10, 11, 12, 16]. Naturally an open question was posed in [5] for finding better online strip packing algorithms that are not based on the shelf concept.

The core of shelf packing is reducing the two-dimensional problem to the one-dimensional problem. Basically shelf algorithms consist of two steps. The first one is *shelf design* which only takes the heights of rectangles into account. One shelf can be regarded as a bin with a specific height. The second step is *packing into a shelf*, where rectangles with similar heights are packed into the same shelves. This step is done by employing some bin packing algorithms that pack the rectangles with a total width bounded by one into a shelf. Clearly, to maintain the quality of bin packing algorithms in shelf packing we must improve the first step. Along this line we make the following contributions.

Our contributions. We propose a batch packing strategy and establish a general algorithmic framework between bin packing and strip packing. It is shown that any offline bin packing algorithm can be used for offline strip packing maintaining the asymptotic worst-case ratio. As an example, the well known bin packing algorithm FFD can approximate strip packing with an asymptotic worst-case ratio of 11/9. A simple AFPTAS can easily be derived from [8].

We further prove that a class of online bin packing algorithm based on Super Harmonic algorithm [13] can be used in online strip packing maintaining the same asymptotic competitive ratio. This result implies that the known Harmonic based bin packing algorithms [10, 11, 12, 13] can be converted into online strip packing algorithms without changing their asymptotic competitive ratios (better than h_∞), and thus affirmatively answers the open question in [5]. Note that the current champion algorithm for online bin packing is Harmonic++ by Seiden [13], which has an asymptotic competitive ratio of 1.58889. Hence strip packing admits an online algorithm with the same upper bound of 1.58889.

Main ideas. Recall that strip packing becomes bin packing if all rectangles have the same height. It motivates us to construct new rectangles with the same height by bundling a subset of given items. More precisely, in the offline case, we pack in batch the rectangles with similar width into rectangular bins of pre-specified height of c, where $c > 1$ is a sufficiently large constant. Then we obtain a set of new rectangles (rectangular bins) of the same height. The next step is to use bin packing algorithms on the new set. In the on-line case the strategy is slightly different. We divide the rectangles into two groups according to their widths, to which we apply the above batching strategy and the standard shelf algorithms respectively.

Asymptotic worst-case (competitive) ratio. To evaluate an approximation (online) algorithms for strip packing and bin packing we use the standard measure defined as follows.

Given an input list L and an approximation (online) algorithm A, we denote by $OPT(L)$ and $A(L)$, respectively, the height of the strip used by an optimal (offline) algorithm and the height used by (online) algorithm A for packing list L.

The *asymptotic worst-case (competitive) ratio* R_A^∞ of algorithm A is defined by

$$R_A^\infty = \lim_{n\to\infty} \sup_L \{A(L)/OPT(L) | OPT(L) = n\}.$$

2 The offline problem

Given a rectangle R, throughout the paper, we use $w(R)$ and $h(R)$ to denote its width and height, respectively.

Fractional strip packing. A fractional strip packing of L is a packing of any list L' obtained from L by subdividing some of its rectangles by horizontal cuts: each rectangle (w, h) is replaced by a sequence $(w, h_1), (w, h_2), ..., (w, h_k)$ of rectangles such that $h = \sum_{i=1}^k h_i$.

Homogenous lists. Let L and L' be two lists where any rectangle of L and L' takes a width from q distinct numbers $w_1 > w_2 > \cdots > w_q$. List L is *r-homogenous* to L' where $r \geq 1$ if

$$\sum_{w(R')=w_i, R'\in L'} h(R') \leq \sum_{w(R)=w_i, R\in L} h(R) \leq r \cdot \sum_{w(R')=w_i, R'\in L'} h(R').$$

The following lemma is an implicit byproduct of the APTAS for strip packing given by Kenyon and Rémila [9].

Lemma 1 *For each strip packing instance I and $\epsilon > 0$, we have $OPT(I) \leq (1+\epsilon)OPT_{FSP}(I) + O(\epsilon^{-2})$, where $OPT_{FSP}(I)$ is the optimal value of fractional strip packing for instance I.*

The next lemma shows a useful property of *homogenous* lists.

Lemma 2 *Given two lists L and L', if L is r-homogenous to L', we have $OPT_{FSP}(L') \leq OPT_{FSP}(L) \leq r \cdot OPT_{FSP}(L')$.*

Proof. If $r = 1$, it is easy to see that any fractional strip packing of L is a fractional packing of L' and vice versa. The conclusion thus follows immediately.

Now we consider the case that $r > 1$. By adding some rectangles to L' we can get a new list L'_1 which is 1-homogenous to L. We have

$$OPT_{FSP}(L') \leq OPT_{FSP}(L'_1) = OPT_{FSP}(L).$$

On the other hand we obtain another list L'_2 by prolonging in height all rectangles of L', i.e., if $(w, h) \in L'$, then $(w, rh) \in L'_2$. Clearly

$$OPT_{FSP}(L'_2) \leq r \cdot OPT_{FSP}(L').$$

Moreover, $OPT_{FSP}(L) \leq OPT_{FSP}(L'_2)$. The lemma holds. □

Theorem 1 *Given two lists L and L', if L is r-homogenous to L', then for any $\epsilon > 0$*

$$OPT(L) \leq r(1+\epsilon)OPT(L') + O(\epsilon^{-2}).$$

Proof. By Lemma 1,

$$OPT(L) \leq (1+\epsilon)OPT_{FSP}(L) + O(\epsilon^{-2}).$$

By Lemma 2,

$$OPT_{FSP}(L) \leq r \cdot OPT_{FSP}(L').$$

Moreover $OPT_{FSP}(L') \leq OPT(L')$. Hence we have this theorem. □

In the following we are ready to present our approach for offline strip packing. Given an input list $L = \{R_1, \ldots, R_n\}$ such that $w_1 \geq w_2 \geq \cdots \geq w_n$, where $R_i = (w_i, h_i)$, and a constant $c > 1$, we construct an offline algorithm $B\&P_A$ using some bin packing algorithm A as a subroutine. Basically the strategy consists of two stages.

Stage 1 - *Batching.* Pack $R_1, \ldots, R_i$ by NF algorithm in the vertical direction into a slip $S_1 = (w_1, c)$, where $\sum_{j=1}^{i} h_j \leq c < \sum_{j=1}^{i+1} h_j$ and pack $R_{i+1}, \ldots, R_k$ into a slip $S_2 = (w_{i+1}, c)$, and so on, until all items are packed, shown as Figure 1. (Note that except for the last slip, all slips have the packed heights at least $(c-1)$.)

Stage 2 - *Packing.* Except for the last slip, pack all slips into the strip by algorithm A, since all slips have the same heights c. Then append the last slip on the top of the strip.
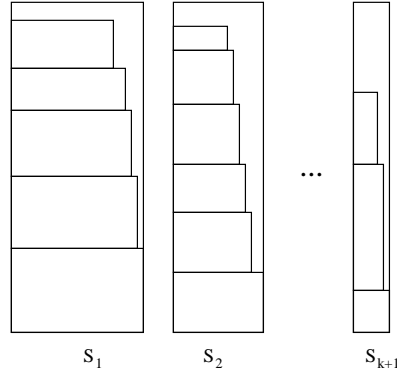

Figure 1: Packing rectangles into slips

We present the main result for the offline case. In terms of the asymptotic worst case ratio, strip packing is essentially the same as bin packing.

Theorem 2 *The asymptotic worst-case ratio $R_{B\&P_A}^{\infty} = R_A^{\infty}$ for any bin packing algorithm A.*

Proof. Assume that $R_A^{\infty} = \alpha$. After the first stage of algorithm $B\&P_A$, we get a series of slips $S_1, \ldots, S_k, S_{k+1}$, shown as Figure 1. We then round up every item (w_j, h_j) in slip S_i to $(w(S_i), h_j)$ and obtain a new list $\bar{L}$, where $w(S_i)$ is the width of slip S_i. On the other hand, we obtain another list $\underline{L}$ by rounding down every item (w_j, h_j) in slip S_i to $(w(S_{i+1}), h_j)$ (here we set $w(S_{k+2}) = 0$). We have

$$OPT(\underline{L}) \leq OPT(L) \leq OPT(\bar{L}) \qquad (1)$$

Denote two sets $L_1 = \{S_1, \ldots, S_k\}$ and $L_2 = \{S_2, \ldots, S_k\}$. Then

$$OPT(L_2) \leq OPT(L_1) \leq OPT(L_2) + c. \qquad (2)$$

We can treat S_i as a one-dimensional item ignoring its height since $h(S_i) = c$ for $i = 1, 2, \ldots, k$. Let $I(L_1)$ be the corresponding item set for bin packing induced from the list L_1, i.e, $I(L_1) = \{w(S_1), w(S_2), \ldots, w(S_k)\}$. And $OPT(I(L_1))$ is the minimum number of bins used to pack $I(L_1)$. It follows that $OPT(L_1) = c \cdot OPT(I(L_1))$.

Note that L_2 is $c/(c-1)$-homogenous to $\underline{L}$, by Theorem 1, we have

$$OPT(L_2) \leq \frac{c}{c-1}(1+\epsilon)OPT(\underline{L}) + O(c + \epsilon^{-2}). \qquad (3)$$

Now we turn to algorithm $B\&P_A$. After Stage 1 the list L becomes $L_1 \cup \{S_{k+1}\}$. At Stage 2 we deal with a bin packing problem: pack $k+1$ items with size of $w(S_i)$ into the minimum number

of bins. The bin packing algorithm A is applied to $I(L_1)$ while S_{k+1} occupies a bin itself. Thus $B\&P_A(L) \le c \cdot A(I(L_1)) + c$. Since $R_A^\infty = \alpha$, we have $A(I(L_1)) \le \alpha OPT(I(L_1)) + O(1)$. Then

$$B\&P_A(L) \le c \cdot A(I(L_1)) + c \le \alpha \cdot c \cdot OPT(I(L_1)) + O(c) = \alpha \cdot OPT(L_1) + O(c).$$

Combining with (2),(3), (1), we have

$$
\begin{aligned}
B\&P_A(L) &\le \alpha OPT(L_2) + O(c) & (4) \\
&\le \frac{\alpha c}{(c-1)}(1+\epsilon)OPT(\underline{L}) + O(\epsilon^{-2} + c) & (5) \\
&\le \frac{\alpha c}{(c-1)}(1+\epsilon)OPT(L) + O(\epsilon^{-2} + c). & (6)
\end{aligned}
$$

As c goes to infinite, this theorem follows. $\qquad\square$

By Theorem 2, any offline bin packing algorithm can be transformed into an offline strip packing algorithm without changing the asymptotic worst case ratio. If the well known algorithm FFD ([1] [7][17]) is used in our approach, then we get a simple and fast algorithm $B\&P_{FFD}$ for strip packing and have the following result from Theorem 2.

Corollary 1 *Given constants $\epsilon > 0$ and $c > 1$, for any strip packing instance L, $B\&P_{FFD}(L) \le \frac{11c}{9(c-1)}(1+\epsilon)OPT(L) + O(\epsilon^{-2} + c)$, where $c \le \epsilon OPT(L)$.*

3 The online problem

In this section we consider online strip packing. In the online case we are not able to sort the rectangles in advance because of no information on future items. Due to this point we cannot reach a complete matching between bin packing algorithms and strip packing algorithms generated from the former. However we can deal with a class H of Super Harmonic algorithms [13](to be given in the appendix), which includes all known online bin packing algorithms based on Harmonic. Such an algorithm can be used in online strip packing without changing its asymptotic worst-case ratio.

A general algorithm of Super Harmonic algorithms has the following characteristics.

- Items are classified into $k+1$ groups by their sizes, where k is a constant integer.

- Those items in the same group are packed by the same manner.

Let A be any algorithm of Super Harmonic algorithm. Our approach $G\&P_A$ is presented below.

Grouping: A rectangle is *wide* if its width is at least ϵ; otherwise it is *narrow*, where $\epsilon > 0$ is a given small number. We further classify *wide* rectangles into k classes, where k is a constant, as Algorithm A does. Let $1 = t_1 > t_2 > \cdots > t_k > t_{k+1} = \epsilon$. Denote I_j to be the interval $(t_{j+1}, t_j]$ for $j = 1, ..., k$. A rectangle is of type-i if its width $w \in I_i$.

Packing narrow rectangles: Apply the standard shelf algorithm NF_r [3] to *narrow* rectangles $R = (w, h)$, where $0 < r < 1$ is a parameter. Round h to r^s if $r^{s+1} < h \le r^s$. If R cannot be packed into the current open shelf with height of r^s, then close the current one and open a new one with height r^s and pack R into it, otherwise just pack R into the current one by NF.

Packing wide rectangles: We pack *wide* rectangles into bins of $(1, c)$, where $c = o(OPT(L)) > 1$ is a constant. Similarly as the offline case we batch the items of the same type and pack them

into a slip. Here we specify the width of the slip by values t_i for $i < k + 1$ and name a slip (t_i, c) of type-i. Suppose that the incoming rectangle R is of type i ($w \in (t_{i+1}, t_i]$). If there is a slip of type-i with a packed height less than $c - 1$, then pack R into it by algorithm NF in the vertical direction. Otherwise create a new empty slip of type-i with size (t_i, c) and place R into the new slip by NF algorithm in the vertical direction. As soon as a slip is created, view it as one dimensional item and pack it by algorithm A into a bin of $(1, c)$. Figure 2(b) shows an illustration.
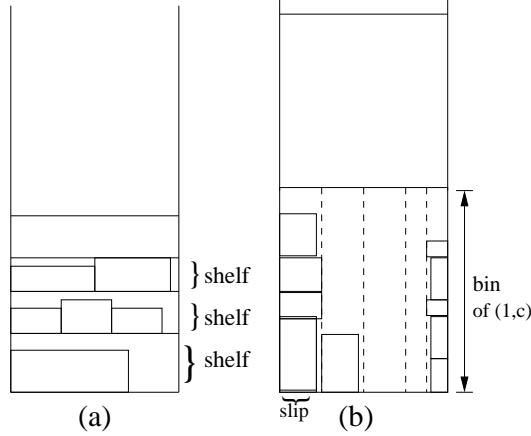

Figure 2: Shelf packing vs our packing

The weighting function technique introduced by Ullman [14] has been widely used in performance analysis of bin packing algorithms [5][10][13]. Roughly speaking, the weight of an item indicates the maximum portion of a bin that the item occupies. Then, Seiden generalized the idea of weighting function and proposed a weighting system which can be used to analyze Harmonic, Refined Harmonic, Modified Harmonic, Modified Harmonic 2, Harmonic+1 and Harmonic++. The following analysis of $G\&P_A$ is based on the weighting system proposed by Seiden [13].

Weighting Systems: Let $\mathbb{R}$ and $\mathbb{N}$ be the sets of real numbers and nonnegative integers, respectively. A *weighting system* for algorithm A is a tuple $(\mathbb{R}^m, \mathbf{w}_A, \xi_A)$. $\mathbb{R}^m$ is a vector space over the real numbers with dimension m. The function $\mathbf{w}_A : (0, 1] \mapsto \mathbb{R}^m$ is called the *weighting function*. The function $\xi_A : \mathbb{R}^m \mapsto \mathbb{R}$ is called the *consolidation function*. Seiden defined a $2K + 1$ dimensional weighting system for Super Harmonic, where K is a parameter of Super Harmonic algorithm. Real numbers $\alpha_i, \beta_i, \gamma_i, \epsilon$ and functions $\phi(i), \varphi(i)$ are defined in Super Harmonic algorithm. The unit basis vectors of the weighting system are denoted by

$$\mathbf{b}_0, \mathbf{b}_1, ..., \mathbf{b}_K, \mathbf{r}_1, ..., \mathbf{r}_K.$$

The weighting function is

$$\mathbf{w}_A(x) = \begin{cases} (1 - \alpha_i)\frac{\mathbf{b}_{\phi(i)}}{\beta_i} + \alpha_i \frac{\mathbf{r}_{\varphi(i)}}{\gamma_i} & \text{if } x \in I_i \text{ with } i \leq k, \\ x\frac{\mathbf{b}_0}{1-\epsilon} & \text{if } x \in I_{k+1}. \end{cases}$$

The consolidation function is

$$\xi_A(\mathbf{x}) = \mathbf{x} \cdot \mathbf{b}_0 + \max_{1 \leq j \leq K+1} \min \left\{ \sum_{i=j}^{K} \mathbf{x} \cdot \mathbf{r}_i + \sum_{i=1}^{K} \mathbf{x} \cdot \mathbf{b}_i, \sum_{i=1}^{K} \mathbf{x} \cdot \mathbf{r}_i + \sum_{i=1}^{j-1} \mathbf{x} \cdot \mathbf{b}_i \right\}.$$

Lemma 3 *[13] For all sequences of bin packing $\delta = (p_1, ..., p_n)$,*

$$cost_A(\delta) \leq \xi_A\Big(\sum_{i=1}^{n} \mathbf{w}_A(p_i)\Big) + O(1).$$

This means that the cost of algorithm A is bounded by the total weight of the items.

We can obtain a similar result with Lemma 3 by defining our weighting function as follows,

$$\mathbf{w}_A(P) = y \cdot \mathbf{w}_A(x),$$

where P is a rectangle of size (x, y).

Lemma 4 *For any sequence of rectangles $L = (P_1, ..., P_n)$, the cost by $G\&P_A$ is*

$$cost_A(L) \leq \max\{\frac{c}{c-1}, \frac{1}{r}\}\xi_A\Big(\sum_{i=1}^{n} \mathbf{w}_A(P_i)\Big) + O(1).$$

Since the proof is similar with the one in [13], we give it in the appendix.

For bin packing, a *pattern* is a tuple $q = \langle q_1, ..., q_k \rangle$ over $\mathbb{N}$ such that

$$\sum_{i=1}^{k} q_i t_{i+1} < 1,$$

where q_i is the number of items of type i contained in the bin. Intuitively, a pattern describes the contents of a bin. The weight of pattern q is

$$\mathbf{w}_A(q) = \mathbf{w}_A\Big(1 - \sum_{i=1}^{k} q_i t_{i+1}\Big) + \sum_{i=1}^{k} q_i \mathbf{w}_A(t_i).$$

Define $\mathcal{Q}$ to be the set of all patterns q. Note that $\mathcal{Q}$ is necessarily finite.

A *distribution* is a function $\chi : \mathcal{Q} \mapsto \mathbb{N}_{\geq 0}$ such that

$$\sum_{q \in \mathcal{Q}} \chi(q) = 1.$$

Given an instance of bin packing δ, Super Harmonic uses $cost(\delta)\chi(q)$ bins containing items as described by the pattern q.

Lemma 5 *[13] For any distribution χ, if we set A as Harmonic++ then*

$$\xi_A\Big(\sum_{q \in \mathcal{Q}} \chi(q)\mathbf{w}_A(q)\Big) \leq 1.58889.$$

Theorem 3 *If we set algorithm A to Harmonic++, then the asymptotic competitive ratio of algorithm $G\&P_A$ is 1.58889, where c is a constant..*
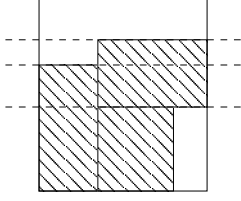
Figure 3: Cutting an optimal packing into layers

Proof. Given an optimal packing for L, we cut the optimal packing into layers such that all rectangles in each layer have the same height, shown as in Fig. 3. (Here the rectangle may be a part of the original one.)

Now, we show this cutting does not change the total weight. Given a rectangle $R = (x, y)$, if we cut it into $P_1, ..., P_m$ such that $P_i = (x, y_i)$ and $y = \sum_i y_i$, then

$$\mathbf{w}_A(R) = y\mathbf{w}_A(x) = \sum_i y_i \mathbf{w}_A(x) = \sum_i \mathbf{w}_A(P_i).$$

Let L' be the list induced from L by the above cutting. Then

$$\xi_A\Big(\sum_{R \in L} \mathbf{w}_A(R)\Big) = \xi_A\Big(\sum_{R \in L'} \mathbf{w}_A(R)\Big). \tag{7}$$

It is not difficult to see each layer corresponds to a *pattern* of bin packing. Let h_q is the total height of the pattern q. So,

$$OPT(L) = \sum_{q \in \mathcal{Q}} h_q = \sum_{q \in \mathcal{Q}} OPT(L)\chi(q),$$

where $\mathcal{Q}$ is the set of all pattern and $\chi(q)$ is one distribution of $\mathcal{Q}$. Then

$$\xi_A\Big(\sum_{R \in L'} \mathbf{w}_A(R)\Big) \leq \xi_A\Big(\sum_{q \in \mathcal{Q}} h_q \mathbf{w}_A(q)\Big) = OPT(L)\xi_A\Big(\sum_{q \in \mathcal{Q}} \chi(q)\mathbf{w}_A(q)\Big).$$

If we set algorithm A to Harmonic++, then by lemma 5, $\xi_A\Big(\sum_{q \in \mathcal{Q}} \chi(q)\mathbf{w}_A(q)\Big) \leq 1.58889$, then by (7), $\xi_A(\sum_{R \in L} \mathbf{w}_A(R)) \leq 1.58889 OPT(L)$. By lemma 4 and when r goes to 1 and c goes to ∞, the asymptotic competitive ratio of algorithm $G\&P_A$ is 1.58889. □

4   Concluding Remarks

Although strip packing is a generalization of the one dimensional bin packing problem, we show from the point of algorithmic view that it is essentially the same as bin packing. In terms of asymptotic performance we give a universal method to apply the algorithmic results for bin packing to strip packing maintaining the solution quality. However our approach cannot be applied to strip packing in terms of absolute performance. Note that algorithm FFD has an absolute worst-case ratio of $3/2$ which is the best possible unless $P = NP$. It is challenging to prove or disprove the existence of a $3/2$-approximation algorithm for offline strip packing.

References

[1] B.S. Baker, A new proof for the first-fit decreasing bin-packing algorithm. *J. Algorithms* 6, 49-70, 1985.

[2] B.S. Baker, D.J. Brown, and H.P. Katseff, A 5/4 algorithm for two-dimensional packing. *J. Algorithms* 2, 348-368, 1981.

[3] B.S. Baker and J.S. Schwarz, Shelf algorithms for two-dimensional packing problems, *SIAM J. Comput.* 12, 508-525, 1983.

[4] E.G. Coffman, M.R. Garey, D.S. Johnson, and R.E. Tarjan, Performance bounds for level oriented two dimensional packing algorithms, *SIAM J. Comput.* 9, 808-826, 1980.

[5] J. Csirik and G.J. Woeginger, Shelf algorithm for on-line strip packing, *Information Processing Letters* 63, 171-175, 1997.

[6] I. Golan, Performance bounds for orthogonal, oriented two-dimensional packing algorithms, *SIAM J. Comput.* 10, 571-582, 1981.

[7] D.S. Johnson, Near-optimal bin-packing algorithms, *doctoral thesis, M.I.T., Cambridge, Mass.*, 1973.

[8] N. Karmarkar and R.M. Karp, An efficient approximation scheme for the one-dimensional bin-packing problem, In *Proc. 23rd Annual IEEE Symp. Found. Comput. Sci.*, 312-320, 1982.

[9] C. Kenyon and E.Rémila, A near-optimal solution to a two-dimensional cutting stock problem, *Mathematics of Operations Research* 25, 645-656, 2000.

[10] C.C. Lee and D.T. Lee, A simple on-line bin-packing algorihtm, *J. ACM* 32, 562-572, 1985.

[11] P.V. Ramanan, D.J. Brown, C.C. Lee, and D. T. Lee, On-line bin packing in linear Time, *J. Algorithms* 10, 305-326, 1989.

[12] M.B. Richey, Improved bounds for harmonic-based bin packing algorithms, *Discrete Appl. Math.* 34, 203-227, 1991.

[13] S.S. Seiden, On the online bin packing problem, *J. ACM* 49, 640-671, 2002.

[14] J.D. Ullman, The performance of a memory allocation algorithm. *Tech. Rep. 100, Princeton University, Princeton, N.J.,Oct.*, 1971.

[15] A. van Vliet, An improved lower bound for on-line bin packing algorithms, *Inform. Process. Lett.* 43, 277-284,1992.

[16] A.C.-C. Yao, New Algorithms for Bin Packing, *J. ACM* 27, 207-227, 1980.

[17] M. Yue, A simple proof of the inequality FFD(L) $\leq$ 11/9OPT(L) +1, $\forall L$ for the FFD bin-packing algorithm, *Acta mathematicae applicatae sinica* 7, 321-331, 1991.

Appendix

Super Harmonic Algorithm

In Super Harmonic [13] algorithm, items are classified into $k+1$ classes, where $k = 70$. Let $t_1 = 1 > t_2 > ... > t_k > t_{k+1} = \epsilon > t_{k+2} = 0$ be real numbers. The interval I_j is defined to be $(t_{j+1}, t_j]$ for $j = 1, ..., k$. And an item with size x has type-i if $x \in I_i$.

Parameters in Harmonic algorithm: Each type-i item is assigned a color, *red* or *blue*, $i \leq k$.

The algorithm uses two sets of counters, $e_1, ..., e_k$ and $s_i, ..., s_k$, all of which are initially zero. The total number of type-i items is denoted by s_i, while the number of type-i *red* items is denoted by e_i. For $1 \leq i \leq k$, during packing process, for type-i items, the balance between s_i and e_i is kept, i.e., $e_i = \lfloor \alpha_i s_i \rfloor$, where $\alpha_1, ..., \alpha_k \in [0,1]$ are constants.

$\delta_i = 1 - t_i \beta_i$ is the left space when a bin is filled with β_i type-i items. If possible, the left space is used for *red* items. $D = \{\Delta_1, ..., \Delta_K\}$ is the set of spaces into which red items can be packed, and $0 = \Delta_0 < \Delta_1 < \cdots < \Delta_K < 1/2$; where $K \leq k$. $\Delta_{\phi(i)}$ is the space used to hold *red* items in a bin which holds β_i *blue* items of type-i, where function ϕ is defined as $\{1,...,k\} \mapsto \{0,...,K\}$. And ϕ satisfies $\Delta_{\phi(i)} \leq \delta_i$. $\phi(i) = 0$ indicates that no red items are accepted. Define $\gamma_i = 0$ if $t_i > \Delta_K$, otherwise $\gamma_i = max\{1, \lfloor \Delta_1 / t_i \rfloor\}$. In the case that $\Delta_K \geq t_i > \Delta_1$, we set $\gamma_i = 1$. Again, this seems to be the best choice from a worst case perspective. Define

$$\varphi(i) = \min\{j | t_i \leq \Delta_j, 1 \leq j \leq K\}.$$

Intuitively, $\varphi(i)$ is the index of the smallest space in D into which a red item of type i can be placed.

Naming bins: Bins are named as follows:

$$\{i | \phi_i = 0, 1 \leq i \leq k, \}$$

$$\{(i, ?) | \phi_i \neq 0, 1 \leq i \leq k, \}$$

$$\{(?, j) | \alpha_j \neq 0, 1 \leq j \leq k, \}$$

$$\{(i, j) | \phi_i \neq 0, \alpha_j \neq 0, \gamma_j t_j \leq \Delta_{\phi(i)}, 1 \leq i, j \leq k\}.$$

Group (i) contains bins that hold only *blue* items of type-i. Group (i, j) contains bins that contain *blue* items of type-i and *red* items of type-j. *Blue* group $(i, ?)$ and *red* group $(?, j)$ are indeterministic bins, in which they *currently* contain only *blue* items of type-i or *red* items of type-j respectively. During packing, *red* items or *blue* items will be packed if necessary, i.e., indeterministic bins will be changed into (i, j).

Super Harmonic

1. For each item p: $i \leftarrow$ type of p,

    (a) if $i = k + 1$ then using NF algorithm,

    (b) else $s_i \leftarrow s_i + 1$; if $e_i < \lfloor \alpha_i s_i \rfloor$ then $e_i \leftarrow e_i + 1$; { color p red }

        i. If there is a bin in group $(?, i)$ with fewer than γ_i type-i items, then place p in it. Else if, for any j, there is a bin in group (j, i) with fewer than γ_i type-i items then place p in it.

        ii. Else if there is some bin in group $(j, ?)$ such that $\Delta_{\phi(j)} \geq \gamma_i t_i$, then pack p in it and change the bin into (j, i).

        iii. Otherwise, open a bin $(?, i)$, pack p in it.

10

(c) else {color p blue}:

    i. if $\phi_i = 0$ then if there is a bin in group i with fewer than β_i items then pack p in it, else open a new group i bin, then pack p in it.

    ii. Else:

        A. if, for any j, there is a bin in group (i,j) or $(i,?)$ with fewer than β_i type-i items, then pack p in it.

        B. Else if there is a bin in group $(?,j)$ such that $\Delta_{\phi(i)} \geq \gamma_j t_j$ then pack p in it, and change the group of this bin into (i,j).

        C. Otherwise, open a new bin $(i,?)$ and pack p in it.

Lemma 6 *If the total area of* narrow *rectangles is S then the cost for* narrow *rectangles by $G\&P_A$ is at most $\frac{S}{r(1-\epsilon)} + O(1)$.*

Proof. Note that every narrow rectangle has its width at most ϵ. Given a close shelf with height h, the total area of rectangles in it is larger than $r \cdot h(1 - \epsilon)$. If the total cost of all close shelves is H_1, then $S > r \cdot H_1(1 - \epsilon)$. On the other hand, at any time in the strip packing maintained by algorithm $G\&P_A$, the total cost of all open shelves (in each of which the total width of rectangles packed is less than $1 - \epsilon$) is less than $\sum_{i=0}^{\infty} r^i = 1/(1 - r)$ $(0 < r < 1)$.

So the total cost for narrow items is at most $\frac{S}{r(1-\epsilon)} + O(1)$. $\qquad\square$

Lemma 7 *$G\&P_A$ algorithm maintains the following invariants.*
i) at most one bin has fewer than β_i slips in any group $(i,?)$ or (i).
ii) at most one bin has fewer than γ_i slips in any group $(?,i)$.
iii) at most three bins have fewer than β_i slips or fewer than γ_i slips in any group (i,j).
iiii) at most k bins have a slip with the total packed height less than $c - 1$.

Proof. Since i), ii), iii) are direct from Lemma 2.2 [13], we just prove the claim in iiii). Totally, there are k kinds of slips and we maintains that any time, for each kind of a slip, there is at most one slip with the total packed height less $c - 1$. So, iiii) holds. $\qquad\square$

In Super Harmonic algorithm, if we define the *class* of a red item of type i to be $\varphi(i)$ and the class of a blue item of type i to be $\phi(i)$. Let B_i and R_i be the number of bins containing blue items of class i and red items of class i, respectively.

Lemma 8 *[13] In Super Harmonic algorithm, the total number of bins for red and blue items is at most*

$$B_0 + \max_{1 \leq j \leq K+1} \min \left\{ \sum_{i=j}^{K} R_i + \sum_{i=1}^{K} B_i, \sum_{i=1}^{K} R_i + \sum_{i=1}^{j-1} B_i \right\} + O(1).$$

Proof of Lemma 4

Proof. Let B_i and R_i be the number of bins containing blue slips of class i and red slips of class i, respectively. Let D be the total area of narrow rectangles. By lemmas 6, 7, 8, the total cost $cost_A(L)$ is at most

$$\frac{D}{r(1 - \epsilon)} + c \cdot \left(B_0 + \max_{1 \leq j \leq K+1} \min \left\{ \sum_{i=j}^{K} R_i + \sum_{i=1}^{K} B_i, \sum_{i=1}^{K} R_i + \sum_{i=1}^{j-1} B_i \right\} \right) + O(1)$$

To complete the proof, we show that this is at $\max\{\frac{1}{r}, \frac{c}{c-1}\}\xi_A(\mathbf{x}) + O(1)$, where $\mathbf{x} = \sum_{i=1}^{n} w_A(R_i)$. Consider D first,

$$\frac{D}{1-\epsilon} = \mathbf{b_0} \cdot \sum_{x \in I_{k+1}} y\mathbf{w}_A(x).$$

Given a close slip P with width x, i.e, the packed height in it is at least $c-1$,

$$\sum_{R \in P} \mathbf{w}_A(R) \geq (c-1)\mathbf{w}_A(x).$$

Let l_j be the number of type j slips with packed heights at least c-1 and $l_m = \sum_{j=0}^{k} l_j$ be the total number of these slips. Let x_h be the width of the h-th slip. Then

$$\begin{aligned}
\mathbf{b_i} \cdot \sum_{j=1}^{n} \mathbf{w}_A(R_j) &\geq \mathbf{b_i} \cdot \sum_{h=1}^{l_m} (c-1)\mathbf{w}_A(x_h) \\
&= \mathbf{b_i} \cdot \sum_{x_h \in I_j, \phi(j)=i} (c-1)\mathbf{w}_A(x_h) \\
&= (c-1) \sum_{0 \leq j \leq k, \phi(j)=i} \frac{(1-\alpha_j)l_j}{\beta_j}
\end{aligned}$$

Consider the cost for packing blue slips, say cB_i, by lemma 7,

$$cB_i = c \sum_{0 \leq j \leq k, \phi(j)=i} \frac{(1-\alpha_j)l_j}{\beta_j} + O(1) \leq \frac{c}{c-1} \times \mathbf{b_i} \cdot \sum_{j=1}^{n} \mathbf{w}_A(R_j) + O(1)$$

So, in the same way, we have

$$cR_i = c \sum_{1 \leq j \leq k, \varphi(j)=i} \frac{\alpha_j l_j}{\beta_j} + O(1) \leq \frac{c}{c-1} \times \mathbf{r_i} \cdot \sum_{j=1}^{n} \mathbf{w}_A(R_j) + O(1)$$

Hence, we have $cost_A(L) \leq \max\{\frac{c}{c-1}, \frac{1}{r}\}\xi_A\left(\sum_{i=1}^{n} \mathbf{w}_A(R_i)\right) + O(1)$. $\qquad \square$