Approximate Solutions to Bin Packing Problems

Edward G. Coffman, Jr. * János Csirik † Gerhard J. Woeginger ‡

March 15, 1999

1 Introduction

The following abstract packing problem arises in a wide variety of contexts in the real world. A list $L = \langle a_1, a_2, \ldots, a_n \rangle$ of *items* must be packed into, i.e., partitioned among, a minimum-cardinality set of *bins* $B_1, B_2, \ldots$ subject to the constraint that the the set of items in any bin fits within that bin's capacity. To give just a few of the applications modeled by bin packing, we mention storage allocation for computer networks, assigning advertisements to newspaper columns, packing fruit slices into tins, assigning commercials to station breaks on television, copying a collection of files to several floppy disks, packing trucks with a given weight limit, and the cutting-stock problems of various industries like those producing lumber and cable.

Let A denote an arbitrary packing algorithm and let OPT denote an algorithm that produces optimal packings. Let $A(L)$ and $\mathrm{OPT}(L)$ denote the numbers of bins used by algorithms A and OPT. To measure worst-case behavior, the *asymptotic worst-case ratio* R_A^∞ of algorithm A is defined by

$$R_A^\infty := \limsup_{m \to \infty} R_A^m,$$

where

$$R_A^m := \sup_L \{A(L)/\mathrm{OPT}(L) \mid \mathrm{OPT}(L) = m\}.$$

This ratio, which we shall just call the asymptotic ratio, is the performance metric of choice for bin packing algorithms; the standard objectives of bin packing problems are algorithms with small asymptotic ratios. Although we focus on these ratios, comments on *absolute* ratios, $R_A := \sup_L \{A(L)/\mathrm{OPT}(L)\}$, will sometimes be appropriate. Certain classes of algorithms have the undesirable property that a small asymptotic ratio is obtained at the expense of a large absolute ratio.

*egc@bell-labs.com. Bell Labs - Lucent Technologies, 700 Mountain Ave., Murray Hill, NJ 07974-0636, U.S.A.

†csirik@inf.u-szeged.hu. Department of Computer Science, University of Szeged, Aradi vértanúk tere 1, H–6720 Szeged, Hungary. Supported by the Project 20u2 of the Austro-Hungarian Action Fund.

‡gwoegi@opt.math.tu-graz.ac.at. Institut für Mathematik B, TU Graz, Steyrergasse 30, A-8010 Graz, Austria. Supported by the START program Y43-MAT of the Austrian Ministry of Science and by the Project 20u2 of the Austro-Hungarian Action Fund.

In some applications, an *a priori* upper bound α $(0 < \alpha \leq 1)$ on the size of all items is known. To gauge the performance of approximation algorithms on lists of such items, the *parametric asymptotic ratio* $R_A^\infty(\alpha)$ is introduced as

$$R_A^\infty(\alpha) := \limsup_{m \to \infty} R_A^m(\alpha),$$

where

$$R_A^m(\alpha) := \sup_L \{A(L)/\mathrm{OPT}(L) \mid \mathrm{OPT}(L) = m \text{ and all } a_i \in L \text{ are } \leq \alpha\}.$$

The problem defined so far is quite general. Implicitly, we have allowed the bins B_i to have varying capacities and both bins and items to be multidimensional. The bulk of this paper, not to mention the bin packing literature, is confined to the one dimensional problem with equal bin capacities, which we normalize to 1 for convenience; in this case, any set of items whose total size is at most 1 fits into a bin. Correspondingly, items are always assumed to be numbers drawn from an interval $[0, \alpha]$ for some $\alpha \leq 1$. We keep with these classical assumptions throughout the next section; the third and last section will cover more general versions of bin packing. We note once and for all that the versions of bin packing of interest to us here are NP-hard, except where stated otherwise.

Many extensions and variants have been instrumental in maintaining the interest in the general area of bin packing. We mention a few basic illustrations here; others can be found in the more comprehensive treatment by Coffman, Galambos, Martello, and Vigo [12]. Three principal parameters of bin packing problems are the number of items, the bin capacity, and the number of bins. Choosing the number of bins as the objective function gives the classical problem, but choosing one of the two others as the objective function also gives interesting problems. Multiprocessor scheduling actually predates bin packing and is the problem with a given number m of bins and the objective of finding the minimum common capacity such that all items of the given list can be packed into m bins. The problem falls more properly within scheduling theory and is surveyed in that context in the survey article by Lawler, Lenstra, Shmoys, and Rinnooy Kan [40]. The second problem is that of finding a maximum cardinality subset of the given list of items which can be packed into a given number of bins with a given capacity (see e.g. Coffman, Leung, and Ting [16] and Coffman and Leung [15]). The dual of packing is covering, where the item sizes in a bin must sum to *at least* 1. As a final variant we mention *dual bin packing*: maximize the number of unit capacity bins covered by a given list of items (Assmann, Johnson, Kleitman, and Leung [1]).

2 The classical problem

A bin packing algorithm is called *on-line* if it packs every item a_i solely on the basis of the sizes of the items a_j, $1 \leq j \leq i$, i.e., without any information on subsequent items. The decisions of an on-line algorithm are irrevocable; packed items cannot be repacked at later times. A bin packing algorithm that can use full knowledge of all items in packing L is called *off-line*. Below, we cover on-line algorithms first and then discuss a number of off-line algorithms.

2.1 On-line algorithms

2.1.1 Simple algorithms

The simplest approximation algorithm for bin packing is probably the NEXT FIT (NF) algorithm: NF makes a single scan through the list L and shortsightedly packs the items one after the other into a unique, *active* bin. In case an item does not fit into the active bin, the bin is closed (never to be used again), and an empty bin is opened and becomes the new active bin. The running time of NF is linear in the number of items packed. It is easy to see that NF has an asymptotic ratio $R^{\infty}_{NF} = 2$: The list $L = \langle \frac{1}{2}, \varepsilon, \frac{1}{2}, \varepsilon, \frac{1}{2}, \varepsilon, \ldots \rangle$ that consists of n pairs $\frac{1}{2}$, ε where $0 < \varepsilon < \frac{1}{n}$ shows that the asymptotic ratio cannot be better than 2. Moreover, the total contents assigned to two consecutive active bins is always at least 1, and this yields an upper bound of 2 on the asymptotic ratio. Johnson [33, 34] gave a complete analysis of the parametric asymptotic ratio of NF; he showed that for $1/2 \le \alpha \le 1$, $R^{\infty}_{NF}(\alpha) = 2$ and for $\alpha < 1/2$, $R^{\infty}_{NF}(\alpha) = 1/(1-\alpha)$.

An obvious drawback of NF is that it never uses the empty space in closed bins. A simple modification gives the FIRST FIT (FF) algorithm: FF also makes a single scan through the list L, but it never closes an active bin. When packing a new item, FF puts it into the lowest indexed bin into which it will fit. A new bin is started only if the item does not fit into any non-empty bin. The running time of FF is $O(n \log n)$ and thus greater than the $O(n)$ running time of NF. However, FF has a much better asymptotic ratio. Johnson et al [35] proved the following: Let m be a positive integer with $1/(m+1) < \alpha \le 1/m$. Then for $m = 1$, $R^{\infty}_{FF}(\alpha) = 17/10$ holds and for $m \ge 2$, $R^{\infty}_{FF}(\alpha) = (m+1)/m$ holds.

Other classical on-line algorithms are BEST FIT (BF), WORST FIT (WF), and ALMOST WORST FIT (AWF). BF behaves like FF, except that it puts the next item into the bin into which it will fit with the smallest gap left over; ties are broken arbitrarily. WF puts the next item into a nonempty bin with the largest gap, starting a new bin only if this largest gap is not big enough. AWF tries first to put the next item into a nonempty bin with the second largest gap; if the item does not fit there then AWF behaves like WF. All three variants belong to the class of so-called ANY FIT (AF) algorithms: An AF algorithm scans once through the list L while packing the items. It *never* puts an item a_i into an empty bin, unless the item does not fit into any partially filled bin. Similarly, an ALMOST ANY FIT (AAF) algorithm is an AF algorithm that never puts an item into a partially filled bin with the lowest level, unless there is more than one bin having this level or unless the bin of lowest level is the only one that has enough room. Johnson [34] proved a beautiful and surprising result: An AF algorithm can never have an asymptotic ratio better than FF, and all AAF algorithms have the same asymptotic ratio as FF; these statements hold even for the parametric ratios. As an easy consequence we get that the AAF algorithms BF and AWF have asymptotic ratios 17/10. The asymptotic ratio of the AF algorithm WF, however, is 2, the same as NF.

2.1.2 Bounded space on-line algorithms

An on-line bin packing algorithm is said to use *k-bounded-space* if for each new item, the choice of bins into which it may be packed is restricted to a set of k or fewer active bins. A bin becomes active when it receives its first item; once a bin is closed, it can never become active

k	AFF_k	ABF_k	ABB_k	H_k	SH_k	Champion
2	2.00000	1.85000	1.70000	2.00000	2.00000	ABB
3	1.85000	1.80000	1.70000	1.75000	1.75000	ABB
4	1.80000	1.77500	1.70000	1.71429	1.72222	ABB
5	1.77500	1.76000	1.70000	1.70000	1.70000	ABB, H, SH
6	1.76000	1.75000	1.70000	1.70000	1.69444	SH
7	1.75000	1.74286	1.70000	1.69444	1.69388	SH
8	1.74286	1.73750	1.70000	1.69388	1.69106	SH
9	1.73750	1.73333	1.70000	1.69345	1.69104	SH
∞	1.70000	1.70000	1.70000	1.69103	1.69103	H, SH

Table 1: Asymptotic ratios for bounded-space bin packing algorithms, rounded to five decimal places.

again. NEXT FIT uses 1-bounded-space, whereas FF, WF and AWF all use unbounded space. There are four very natural bounded-space bin packing algorithms that are defined via simple *packing* rules for items and simple *closing* rules for bins: A new item can always be packed into the lowest indexed bin (as in FF) or into the bin with the smallest remaining gap (as in BF). If the new item does not fit into any active bin, some active bin has to be closed; in this case one can always choose the lowest indexed bin (the FIRST bin) or the fullest bin (the BEST bin). The corresponding four algorithms are called AFF_k, AFB_k, ABB_k, and ABF_k: Here A stands for Algorithm, the second letter denotes the packing rule (Best fit or First fit), the third letter denotes the closing rule (Best bin or First bin), and k is the upper bound on the number of active bins.

- Algorithm AFF_k. This algorithm was defined under the name NEXT-k FIT by Johnson [34] in 1974. Provably tight bounds were not in hand until almost 20 years later. Csirik and Imreh [21] constructed a sequence of worst-case examples that show $R^\infty_{AFF_k} \geq \frac{17}{10} + \frac{3}{10(k-1)}$, and Mao [50] proved a matching upper bound on $R^\infty_{AFF_k}$.
- Algorithm ABF_k. Mao [49] proved that $R^\infty_{ABF_k} = \frac{17}{10} + \frac{3}{10k}$.
- Algorithm AFB_k. Zhang [62] adapted the analysis of Mao [50] to this algorithm and proved that $R^\infty_{AFB_k} = \frac{17}{10} + \frac{3}{10(k-1)}$.
- Algorithm ABB_k. This algorithm was investigated by Csirik and Johnson [22]. In comparison to the other three algorithms, ABB_k uses the better packing and closing rules and it has the better asymptotic ratio, $R^\infty_{ABB_k} = 17/10$ holds for any $k \geq 2$.

So, except for small k, the asymptotic ratios of all four algorithms are around 17/10; since they are of the ANY FIT type, they of course cannot outperform FF.

Lee and Lee [41] introduced a new class of bounded space algorithms using bin reservation techniques. Their algorithm HARMONIC is based on a partition of the interval $(0,1]$ into k

subintervals, where the partitioning points are $1/2, 1/3, \ldots, 1/k$; to each of these subintervals there corresponds a single active bin, and only items belonging to this subinterval are packed into this bin. If a new item arrives that does not fit into its corresponding active bin, the bin is closed and a new bin is activated. In [41] it is proved that $R^{\infty}_{H_k}$ tends to the number $h_\infty \approx 1.69103$. This number is defined by $h_\infty = \sum_{i=1}^{\infty} \frac{1}{t_i - 1}$ where $t_1 = 2$ and $t_{i+1} = t_i(t_i - 1) + 1$ for $i \geq 1$; it is a prominent number in bin packing. Woeginger [58] introduced the SIMPLIFIED HARMONIC (SH_k) algorithm, a modification of HARMONIC with a different interval structure and with a slightly better asymptotic ratio for small values of k.

A summary of the asymptotic ratios of the bounded-space algorithms for some small values of k is given in Table 1. The asymptotic ratios of all five algorithms always remain above h_∞. In fact, Lee and Lee [41] showed that a bounded-space algorithm cannot have an asymptotic ratio better than h_∞.

2.1.3 Better algorithms

The first on-line algorithm for bin packing with $R^{\infty}_{A} < h_\infty$ was Yao's [60] REVISED FF (RFF). Both the definition and the analysis of RFF are fairly involved; it is something of a cross between FF and HARMONIC with the asymptotic ratio $R^{\infty}_{RFF} = 5/3$. All other known on-line algorithms that beat this h_∞ bound are variants of HARMONIC with a special treatment of the large items $\geq 1/3$. Lee and Lee [41] described the REFINED HARMONIC (RH) algorithm, which was based on H_{20}, and proved the asymptotic ratio $R^{\infty}_{RH} = 373/228 \approx 1.639$. Ramanan, Brown, Lee and Lee [52] introduced the MODIFIED HARMONIC (MH) with $R^{\infty}_{MH} \approx 1.616$. Finally, in 1991 Richey [54] introduced the HARMONIC+1 algorithm, our current champion. The definition of HARMONIC+1 uses a partition of $[0, 1]$ into more than 70 intervals. Its design and analysis were performed with the help of linear programming. Its asymptotic ratio is less than 1.5888.

Finally, we discuss lower bounds on the performance of *any* on-line bin packing algorithm. Roughly speaking, a lower bound argument can proceed as follows. Suppose an on-line algorithm A is confronted with a huge set of tiny items. If A packs these tiny items very tightly, it will not be able to find an efficient packing for the larger items that might arrive later; if such items actually do arrive, A is going to lose. On the other hand, if A leaves lots of room for large items while packing the tiny items, the large items might not arrive; in that case, A is again going to lose. Yao [60] formulated this idea mathematically; he used $1/7 + \varepsilon$ as the size of the tiny items, and $1/3 + \varepsilon$ and $1/2 + \varepsilon$ as sizes of large items. He proved that, with certain lists of such items, the asymptotic ratio of every on-line bin packing algorithm A must satisfy $R^{\infty}_{A} \geq 1.5$. Brown [7] and Liang [47] independently generalized this lower bound to 1.53635. Ten years later van Vliet [56, 57] found an elegant linear programming formulation for the Brown/Liang construction. Van Vliet gave an exact analysis and increased the lower bound to $R^{\infty}_{A} \geq 1.5401$. This is currently the best lower bound known for on-line bin packing.

2.2 Off-line algorithms

2.2.1 Simple algorithms

The FF and BF algorithms do not work well on lists ordered by increasing item size. Intuitively, the large items at the end of the list should be combined with the small items at the beginning of the list, whereas FF and BF produce very dense packings for the small items and very poor packings for the large items. Hence, it is natural to design off-line algorithms that first sort the list by decreasing size, and afterwards behave like one of the simple on-line algorithms. This yields FIRST FIT DECREASING (FFD), BEST FIT DECREASING (BFD) and NEXT FIT DECREASING (NFD). Not surprisingly, the performance of NFD is not very strong: Baker and Coffman [3] proved that $R^{\infty}_{NFD} = h_{\infty}$ holds. On the other hand, the improvement of FFD and BFD over the on-line packing algorithms is dramatic: Johnson [33] showed that $R^{\infty}_{FFD} = R^{\infty}_{BFD} = 11/9 \approx 1.22$. The proof of this result is notorious: more than 70 pages of tedious case analysis. In fact Johnson proved the inequality $FFD(L) \leq (11/9)\text{OPT}(L) + 4$. The additive constant of 4 in this inequality was later reduced to 3 by Baker [2] and then to 1 by Yue [61]; the proofs in [2, 61] are simpler than Johnson's but are based on similar ideas. but slightly simpler.

2.2.2 Better algorithms

After Johnson's 11/9 theorem in the early 70's, quite a few years passed before a polynomial-time algorithm beating the 11/9 bound was published. Yao [60] showed that the bound was beaten by his REVISED FFD (RFFD) algorithm that behaves as FFD but has a special treatment for the items of size $> 1/3$. The running time deteriorates to $O(n^{10} \log n)$, and the asymptotic ratio is improved only very slightly to $11/9 - 10^{-7}$; but the result shows that there is nothing magic about the 11/9 bound.

Fernandez de la Vega and Lueker [30] constructed a polynomial time approximation scheme (PTAS, for short) for bin packing. In other words, they proved that for $\varepsilon > 0$ there exists an approximation algorithm A_ε that has a running time polynomial in the size of the input list L (but exponential in $1/\varepsilon$), and that computes a packing with $A_\varepsilon(L) \leq (1+\varepsilon)\text{OPT}(L) + \text{const}(\varepsilon)$, where $\text{const}(\varepsilon) \to \infty$ as $\varepsilon \to 0$. This result was considerably strengthened by Karmarkar and Karp [36] who designed another PTAS that guarantees $A_\varepsilon(L) \leq \text{OPT}(L) + \log^2(\text{OPT}(L))$. The algorithm in [36] uses a lot of tricks: elimination of small items, rounding techniques, linear programming formulations, the ellipsoid method, and so on. The running time of the algorithm is roughly $O(n^9)$. In this area, the development stops with Karmarkar and Karp, although many questions remain open. Is it possible to guarantee that $A_\varepsilon(L) \leq \text{OPT}(L) + \log(\text{OPT}(L))$ in polynomial time (i.e., replace the $\log^2$ in [36] by a simple log)? We feel that the answer to this question should be YES. Is it possible to guarantee that $A_\varepsilon(L) \leq \text{OPT}(L) + 1$ in polynomial time? The answer to this question might indeed be YES! (and it seems to be completely out of reach of our current techniques).

3 Generalizations

We now relax certain of the assumptions of the classical problem.

3.1 Variable sized bin packing

In variable sized bin packing, the items are packed into bins of several different types $B_1, \ldots, B_r$ with sizes $1 = s(B_1) > s(B_2) > \ldots > s(B_r)$. There is an infinite supply of bins of each type. The goal is to pack the items $a_i \in [0,1]$ into a set of bins with *smallest total size* (observe that the special case of a single bin type B_1 of size $s(B_1) = 1$ is the classical one dimensional bin packing problem). For a list L of items and an approximation algorithm A, denote by $s(A, L)$ the total size of bins used by algorithm A. Denote by $s(\mathrm{OPT}, L)$ the total size of bins used in an optimal packing. The quality of algorithm A is measured by

$$R_A^{var} = \lim_{k\to\infty} \sup\{s(A,L)/s(\mathrm{OPT},L) \mid \mathrm{OPT}(L) \geq k\}.$$

In the on-line version of variable sized bin packing, every time a new bin is opened, the online algorithm decides which bin type to use next.

Friesen and Langston [32] give three simple approximation algorithms for variable sized bin packing with the asymptotic ratios 2, 3/2, and 4/3. Only the first of these three algorithms is on-line. This algorithm always chooses the largest bin size when a new bin is opened, and otherwise behaves just like NF. Not surprisingly, it has the same asymptotic ratio as NF. It can be shown that *any* on-line algorithm that always chooses the largest possible bin size has an asymptotic ratio of at least 2, and also *any* on-line algorithm that always chooses the smallest available bin size has an asymptotic ratio of at least 2. Kinnersley and Langston [38] design a hybrid strategy called FFf. FFf uses the packing strategy of FF. The bin type selection is based on a so-called *filling factor* f with $1/2 \leq f \leq 1$. Suppose that FFf must start a new bin as item a_i arrives. If $a_i \leq 1/2$, then FFf starts a new bin of size 1. If $a_i > 1/2$, then FFf chooses the smallest bin size in the interval $[a_i, a_i/f]$; in case no such bin size exists, FFf chooses size 1. The asymptotic ratio of FFf is at most $1.5 + f/2$, and there exist values of f (e.g. $f = 3/5$) for which this bound is tight. Zhang [63] shows that the asymptotic ratio of FF$\frac{1}{2}$ equals 17/10, thus matching the guarantee of FF.

Csirik [20] designed an on-line algorithm for variable sized bin packing that is based on the HARMONIC algorithm. For every item size a_i, the algorithm computes a corresponding bin type B_j. All items that are assigned to the same bin type B_j are then packed by a HARMONIC algorithm into bins of size $s(B_j)$. For any collection of bin types, the asymptotic ratio of this algorithm can be made arbitrarily close to $h_\infty \approx 1.691$. The algorithm performs even better for special collections of bin types; e.g. for two bin types with sizes $s(B_1) = 1$ and $s(B_2) = 0.7$, the asymptotic ratio is 1.4. These constitute the strongest results currently known for the on-line version of variable sized bin packing.

For the off-line version of variable sized bin packing, Murgolo [51] constructs a polynomial time approximation scheme. This scheme is based on and extends the techniques of Karmarkar and Karp [36].

3.2 Packings in Higher Dimensions

There are several generalizations of bin packing to higher dimensions; we will first discuss packings in which items are vectors and then discuss packings of rectangles in two dimensional arenas.

3.2.1 Vector Packing

In vector packing, instead of an item being a single number, it is a d-dimensional vector $a_i = (v_1(a_i), \ldots, v_d(a_i))$, where $0 \le v_j(a_i) \le 1$ holds for $1 \le j \le d$. The goal is to pack all items into the minimum number of bins in such a way that, in every bin, the sum of all vectors is at most one simultaneously in every coordinate. The vector packing problem arises as a crucial subproblem in scheduling with resource constraints (cf. Garey, Graham, Johnson and Yao [29]).

Kou and Markowsky [39] call an approximation algorithm A for d-dimensional vector packing *reasonable*, if A never yields packings in which the contents of two non-empty bins could be combined into a single bin (in one dimension, this obviously corresponds to ANY FIT algorithms). Kou and Markowsky show that any reasonable vector packing algorithm A obeys the bound $R_A^\infty \le d+1$. Since the obvious generalization of FF to d-dimensional vector packing is reasonable, d-dimensional FF has an asymptotic ratio of at most $d+1$. Garey, Graham, Johnson and Yao [29] performed an exact analysis of d-dimensional FF and showed that it has a slightly better asymptotic ratio of $d + 7/10$. This currently provides the best asymptotic ratio known for d-dimensional on-line vector packing. Galambos, Kellerer and Woeginger [27] and Blitz, van Vliet and Woeginger [6] discuss lower bounds for on-line vector packing. Their lower bounds are rather weak and tend to 2 as d goes to infinity. For $d = 2$ dimensions, the best known lower bound is 1.6712.

For off-line algorithms, the method of Fernandez de la Vega and Lueker [30] yields asymptotic ratios of $d + \varepsilon$ in d dimensions, where ε can be made arbitarily close to 0. Yao [60] proved that in a reasonable (but somewhat restricted) model of computation, any off-line approximation algorithm with $o(n \log n)$ running time must have an asymptotic ratio $R_A^\infty \ge d$. Chekuri and Khanna [9] give polynomial time approximation algorithms for d-dimensional vector packing whose asymptotic ratios grow with $O(\ln d)$. Their approach is based on a linear programming relaxation of this problem. Clearly, their algorithm does not fall into Yao's [60] restricted model of computation. It is an open problem whether $R_A^\infty <$const is possible for some polynomial time approximation algorithm A in any dimension $d \ge 2$ where the value of const is independent of d. In fact, it would even be interesting to obtain asymptotic ratios growing like $O(\ln \ln d)$. Woeginger [59] (see also Chekuri and Khanna [9]) observed that unless $P = NP$, there cannot exist a polynomial time approximation scheme for 2-dimensional vector packing.

3.2.2 Packing in Strips and Two Dimensional Bins

Strip packing originated in the work of Baker, Coffman, and Rivest [5]. In this extension to two dimensions, the items are rectangles and the goal is to pack them into a unit-width semi-infinite strip so as to minimize the total length of the strip spanned by the packing. Packed rectangles can not overlap each other or the boundaries of the strip. We consider the strip to be vertical and require that rectangles be packed orthogonally in their given orientations, i.e., their edges must be parallel to the bottom or sides of the strip and no rotations are allowed. If we place boundaries at the integers and require that no rectangle overlap any such boundary, then we have the two-dimensional packing set-up of Chung, Garey, and Johnson [10], where the objective is to minimize the number of bins (unit squares) needed, i.e., the nearest integer no smaller than the height of the corresponding strip packing.

The *bottom-left* algorithm of Baker, Coffman, and Rivest [5] and the *level-oriented* (or *shelf*) algorithm by Coffman, Garey, Johnson, and Tarjan [11] were fundamental to packing in two dimensions. In a bottom-left packing, each successive item is placed as near the bottom of the strip as possible and then as far left at that height as possible. If the rectangles are packed in decreasing-width order, an asymptotic ratio of 2 is achieved. As the name suggests, in level-oriented packings, each rectangle is placed on one of a sequence of levels. The bottom-most level is the bottom of the strip; a higher level is a horizontal boundary running through the tallest rectangle in the next lower level. Thus, levels correspond to (horizontal) bins in the one-dimensional problem. The algorithms NF, FF, and BF can be adapted in the obvious way to level packings. Prior to applying any of these algorithms, the list may be ordered by decreasing height or width. To cite a typical result (see Coffman, Garey, Johnson, and Tarjan [11]), we mention that FF with rectangles in decreasing height order has the 17/10 asymptotic ratio of FF in the one dimensional case. Baker, Brown, and Katseff [4] extended the basic ideas and designed an algorithm with an asymptotic ratio of 5/4. As a final note, we mention that a polynomial-time approximation scheme for a (somewhat restricted) special case of strip packing was devised by Fernandez de la Vega and Zissimopoulos [31]. Finally, Kenyon and Remila [37] succeeded in constructing an approximation scheme for the general strip packing problem.

Shelves are preset levels and were introduced as the basis of on-line level-oriented algorithms by Baker and Schwarz [8]. As an example, assume that 1 is an a priori bound on rectangle heights and suppose that the only shelf heights possible are r^k, $k \geq 0$ for some given $r \in (0,1)$. Rectangles with heights in $(r^{k+1}, r^k]$ must be placed on shelves of height r^k, starting a new such shelf on top of the current packing whenever necessary according to the given packing algorithm. A principal result of Baker and Schwarz [8] is that the FF shelf algorithm has an asymptotic ratio of $1.7/r$, but an absolute ratio that grows like $1/(1-r)$. In an analogous way, Csirik and Woeginger [25] transform the HARMONIC algorithm into the corresponding HARMONIC shelf algorithm. The asymptotic ratio of the HARMONIC shelf algorithm can be made arbitrarily close to 1.691. Achieving an asymptotic ratio better than 1.691 for on-line strip packing is probably difficult; in [25] it is argued that no shelf algorithm can do this, in which case a completely different approach will be necessary.

The algorithmic techniques of one dimensional and strip packing are easily applied to two dimensional bin packing. Combining FFD bin packing with FFDH strip packing yields an asymptotic ratio known to be in the interval $[2.022, 2.125]$; cf. Chung, Garey, and Johnson [10].

Shelf algorithms were adapted for on-line two dimensional bin packing by Coppersmith and Raghavan [19]. Variants and extensions of their algorithms were introduced by Csirik, Frenk, and Labbé [23] and by Li and Cheng [42]; in the latter paper, an algorithm was presented with the provably tight asymptotic ratio of $2\frac{39}{40}$. The HARMONIC algorithm (see Section 2.1.2) was extended to the two dimensional problem by Li and Cheng [44]; their algorithm has an asymptotic ratio of $(1.691\ldots)^2$, the square of the corresponding result in one dimension. Csirik and van Vliet [24] subsequently described an improvement that substantially reduces the absolute ratio of the Li-Cheng algorithm while leaving the asymptotic ratio unchanged.

Lower bounds for on-line two dimensional bin packing were taken up by Galambos [26].

He proved a lower bound of 1.6. A later collaboration with van Vliet [28] yielded 1.808. The Ph.D. thesis of van Vliet [57] contains 1.851. The best bound currently known is $R_A^\infty \geq 1.907$ (Blitz, van Vliet and Woeginger [6]).

4 Final Remarks

While worst-case analysis of bin packing remains quite active, it must be noted that, in the last 15 years or so, average-case analysis has received at least as much attention. The monograph by Coffman and Lueker [17] on the subject describes basic results and analytic techniques. A more recent survey of average-case results for one-dimensional bin packing can be found in [13]. To illustrate typical average-case estimates, suppose the n items of L are independent, uniform random draws from $[0, 1]$ and consider the wasted space W in the classical NF, FF, and BF packings (i.e., the difference in the number of bins used and the total size of the items). Then we have $\mathrm{E}[W_n^{NF}] \sim n/6$, $(n \to \infty)$ (Coffman, So, Hofri, and Yao [18]), $\mathrm{E}[W_n^{FF}] = \Theta(n^{2/3})$ (Shor [55], Coffman, Johnson, Shor, and Weber [14]), and $\mathrm{E}[W_n^{BF}] = \Theta(\sqrt{n}\log^{3/4} n)$ (Leighton and Shor [46], Shor [55], Rhee and Talagrand [53]). These results can be compared to $\mathrm{E}[W_n^{OPT}] = \sqrt{n}$ (Lueker [48]).

Our discussion of generalizations can be continued to problems of three or more dimensions. In the large, the algorithmic extensions from one to two dimensions can be essentially repeated as extensions from d to $d+1$ dimensions for $d \geq 2$; as one might expect, asymptotic ratios tend to worsen as d increases. We refer the reader to Coppersmith and Raghavan [19], Galambos and van Vliet [28] and Li and Cheng [42, 43, 45].

References

[1] S. B. Assmann, D. S. Johnson, D. J. Kleitman, and J. Y-T. Leung. On a dual version of the one-dimensional bin packing problem. *J. Algorithms*, 5(4):502–525, 1984.

[2] B. S. Baker. A new proof for the First-Fit-Decreasing bin-packing algorithm. *J. Algorithms*, 6:49–70, 1985.

[3] B. S. Baker and E. G. Coffman. A tight asymptotic bound for Next-Fit-Decreasing bin-packing. *SIAM J. Alg. Disc. Meth.*, 2:147–152, 1981.

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

[5] B. S. Baker, E. G. Coffman, Jr., and R. L. Rivest. Orthogonal packings in two dimensions. *SIAM J. Comput.*, 9:846–855, 1980.

[6] D. Blitz, A. Van Vliet, and G. J. Woeginger. Lower bounds on the asymptotic worst-case ratio of on-line bin packing algorithms. Unpublished manuscript, 1996.

[7] D. J. Brown. A lower bound for on-line one-dimensional bin packing algorithms. Technical Report R-864, Coordinated Sci. Lab., Urbana, Illinois, 1979.

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

[9] C. Chekuri and S. Khanna. On multi-dimensional packing problems. in *Proceedings of the tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 185–194, 1999.

[10] F. R. K. Chung, M. R. Garey, and D. S. Johnson. On packing two-dimensional bins. *SIAM J. Alg. Disc. Meth.*, 3:66–76, 1982.

[11] E. G. Coffman, Jr., 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.

[12] E. G. Coffman, Jr., G. Galambos, S. Martello, and D. Vigo. Bin packing approximation algorithms: Combinatorial Analysis, in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos (Eds.). Kluwer Academic Publishers, 1998.

[13] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Bin Packing Approximation Algorithms: A Survey. In *Approximation Algorithms for NP-Hard Problems*, D. Hochbaum (ed.). PWS Publishing Co. (1996), Boston, MA.

[14] E. G. Coffman, Jr., D. S. Johnson, P. W. Shor, and R. R. Weber. First-Fit Bin Packing: A Tight Bound on Expected Wasted Space. *Probab. Th. and Related Fields*, **10**(1997), 69-101.

[15] E. G. Coffman, Jr. and J. Y. Leung. Combinatorial analysis of an efficient algorithm for processor and storage allocation. *SIAM J. Comput.*, 8:202–217, 1979.

[16] E. G. Coffman, Jr., J. Y. Leung, and D. W. Ting. Bin packing: maximizing the number of pieces packed. *Acta Informatica*, 9:263–271, 1977.

[17] E. G. Coffman, Jr. and G. S. Lueker. *Probabilistic Analysis of Packing and Partitioning Algorithms*. Wiley, New York, 1991.

[18] E. G. Coffman, Jr., K. So, M. Hofri, and A. C. Yao. A stochastic model of bin-packing. *Inf. and Cont.*, 44:105–115, February 1980.

[19] D. Coppersmith and P. Raghavan. Multidimensional on-line bin packing: Algorithms and worst-case analysis. *Oper. Res. Lett.*, 8:17–20, 1989.

[20] J. Csirik. An on-line algorithm for variable-sized bin packing. *Acta Informatica*, 26:697–709, 1989.

[21] J. Csirik and B. Imreh. On the worst-case performance of the NkF bin packing heuristic. *Acta Cybernetica*, 9:89–105, 1989.

[22] J. Csirik and D. S. Johnson. Bounded space on-line bin packing: best is better than first. In *Proc. 2nd ACM-SIAM Symp. Discrete Algorithms*, 309–319, 1991.

[23] J. Csirik, J. B. G. Frenk, and M. Labbé. Two dimensional rectangle packing: On line methods and results. ARIDAM V, Rutgers University, 1990.

[24] J. Csirik and A. van Vliet. An on-line algorithm for multidimensional bin packing. *Oper. Res. Lett.*, 13:149–158, 1993.

[25] J. Csirik and G. J. Woeginger. Shelf algorithms for on-line strip packing. *Inform. Process. Lett.*, 63:171–175, 1997.

[26] G. Galambos. A 1.6 lower-bound for the two-dimensional on-line rectangle bin-packing. *Acta Cybernetica*, 10:21–24, 1991.

[27] G. Galambos, H. Kellerer, and G. J. Woeginger. A lower bound for on-line vector packing algorithms. *Acta Cybernetica*, 11:23–34, 1994.

[28] G. Galambos and A. van Vliet. Lower bounds for 1-, 2- and 3-dimensional on-line bin packing algorithms. *Computing*, 52:281–297, 1994.

[29] M. R. Garey, R. L. Graham, D. S. Johnson, and A. C. C. Yao. Resource constrained scheduling as generalized bin packing. *J. Comb. Th. Ser. A.*, 21:257–298, 1976.

[30] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1+\varepsilon$ in linear time. *Combinatorica*, 1:349–355, 1981.

[31] W. Fernandez de la Vega and V. Zissimopoulos. An approximation scheme for strip-packing of rectangles with bounded dimensions. Technical Report, LRI. University of Paris XI, France, 1995.

[32] D. K. Friesen and M. A. Langston. Variable sized bin packing. *SIAM J. Comput.*, 15:222–230, 1986.

[33] D. S. Johnson. *Near-optimal bin packing algorithms.* PhD thesis, MIT, Cambridge, MA, 1973.

[34] D. S. Johnson. Fast algorithms for bin packing. *J. Comput. System Sci.*, 8:272–314, 1974.

[35] 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. Comput.*, 3:256–278, 1974.

[36] N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin packing problem. In *Proc. 23rd Ann. Symp. on Foundations of Computer Science*, 289–298, 1984.

[37] C. Kenyon and E. Remila. Approximate strip-packing. to appear in *Math. Oper. Res.*.

[38] N. G. Kinnersley and M. A. Langston. On-line variable-sized bin packing. *Discr. Appl. Math.*, 22:143–148, 1988.

[39] L. T. Kou and G. Markowsky. Multidimensional bin packing algorithms. *IBM J. Research and Development*, 21:443–448, 1977.

[40] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. in *Handbook of Operations Research and Management Science Vol. 4, Logistics of Production and Inventory*, S. C. Graves, A. H. G. Rinnooy Kan, and P. H. Zipkin, editors, North Holland, 445–522, 1993.

[41] C. C. Lee and D. T. Lee. A simple on-line bin packing algorithm. *J. Assoc. Comput. Mach.*, 32:562–572, 1985.

[42] K. Li and K.-H. Cheng. Generalized First-Fit algorithms in two and three dimensions. *Int. J. Foundations Comp. Sci.*, 1(2):131–150, 1990.

[43] K. Li and K. H. Cheng. On three dimensional packing. *SIAM J. Comput.*, 19:847–867, 1990.

[44] K. Li and K.-H. Cheng. A generalized HARMONIC algorithm for on-line multidimensional bin packing. Technical Report TR UH-Cs-90-2, University of Houston, January, 1990.

[45] K. Li and K. H. Cheng. Heuristic algorithms for on-line packing in three dimensions. *J. Algorithms*, 13:589–605, 1992.

[46] T. Leighton and P. Shor. Tight bounds for minimax grid matching with applications to the average case analysis of algorithms. *Combinatorica*, 9(2):161–187, 1989.

[47] F. M. Liang. A lower bound for on-line bin packing. *Inform. Process. Lett.*, 10:76–79, 1980.

[48] G. S. Lueker. An average-case analysis of bin packing with uniformly distributed item sizes. Technical Report 181, University of California at Irvine, Department of Information and Computer Science, 1982.

[49] W. Mao. Best-k-Fit bin packing. *Computing*, 50:265–270, 1993.

[50] W. Mao. Tight worst-case performance bounds for Next-k-Fit bin packing. *SIAM J. Comput.*, 22:46–56, 1993.

[51] F. D. Murgolo, An efficient approximation scheme for variable-sized bin packing. *SIAM J. Comput.*, 16:149–161, 1987.

[52] P. Ramanan, D. J. Brown, C. C. Lee, and D. T. Lee. On-line bin packing in linear time. *J. Algorithms*, 10:305–326, 1989.

[53] W. T. Rhee and M. Talagrand. Exact bounds for the stochastic upward matching problem. *Trans. Amer. Math. Soc.*, 307(1):109–125, May 1988.

[54] M. B. Richey. Improved bounds for harmonic-based bin packing algorithms. *Discr. Appl. Math.*, 34:203–227, 1991.

[55] P. W. Shor. The average-case analysis of some on-line algorithms for bin packing. *Combinatorica*, 6(2):179–200, 1986.

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

[57] A. Van Vliet. *Lower and upper bounds for on-line bin packing and scheduling heuristics.* PhD thesis, Erasmus University, Rotterdam, The Netherlands, 1995.

[58] G. J. Woeginger. Improved space for bounded-space on-line bin packing. *SIAM J. Discr. Math.*, 6:575–581, 1993.

[59] G. J. Woeginger. There is no asymptotic PTAS for two-dimensional vector packing. *Inform. Process. Lett.*, 64:293–297, 1997.

[60] A. C. C. Yao. New algorithms for bin packing. *J. Assoc. Comput. Mach.*, 27:207–227, 1980.

[61] M. Yue. A simple proof of the inequality $FFD(L) \leq \frac{11}{9}\mathrm{OPT}(L) + 1\ \forall L$, for the FFD bin-packing algorithm. *Acta Math. App. Sinica*, 7:321–331, 1991.

[62] G. Zhang. A tight worst-case performance bound for AFB-k. Technical Report 015, Institute of Applied Mathematics, Academia Sinica, Beijing, China, 1994.

[63] G. Zhang. Worst-case analysis of the FFH algorithm for on-line variable-sized bin packing. *Computing*, 56:165–172, 1996.