SIGACT News Online Algorithms Column 26: Bin packing in multiple dimensions

Rob van Stee

University of Leicester

United Kingdom

I mentioned in a previous column [22] that the best known lower bound for the two-dimensional online bin packing problem is 1.907 by Blitz, van Vliet, Woeginger [2], which is an unpublished (and now lost [24]) manuscript. I have realized since then that even the penultimate result, 1.856, was published only in André van Vliet's Ph.D. thesis [23] and is not readily available. It therefore seems like a good idea to describe his methods here, though not in full detail. This will include a discussion of the three-dimensional case. I will also survey other results in multidimensional packing, that were left out in my previous column.

I would like to invite more contributions to this column, be it surveys, conference reports, or technical articles related to online algorithms and competitive analysis. If you are considering becoming a guest writer, don't hesitate to mail me at `rvs4@le.ac.uk`.

1 Two-dimensional online bin packing

Recall that two-dimensional bin packing is the problem of packing rectangular items into square bins of size 1 without overlap. The goal is to minimize the total number of bins used and we typically consider the asymptotic performance ratio, meaning that we are only interested in large inputs. (Put another way, we allow an additive constant to the competitive ratio.)

A main problem in deriving lower bounds for more than one dimension is that it becomes nontrivial to determine which sets of items can or cannot be packed into a bin. To show that a set of items can be packed, it suffices to give a packing for it (which might not be easy to find, however). On the other hand, there is no known efficient way of showing that a particular set of items *cannot* be packed into a bin.

Recall that the best known lower bound for one-dimensional bin packing at the time André wrote his PhD thesis uses the following item sizes.

$$\frac{1}{2}+\varepsilon, \frac{1}{3}+\varepsilon, \frac{1}{7}+\varepsilon, \frac{1}{43}+\varepsilon, \frac{1}{1807}+\varepsilon, \ldots \tag{1}$$

The results for two dimensions uses items for which both the heights and the widths are elements of this sequence.

The first input uses only the three largest elements for all sizes. Table 1 shows the sizes of the various items. Here ε and δ are sufficiently small numbers. The δ_i values are chosen in such a way that a set consisting of one item of every column can be packed together into a single bin.

Each item will arrive n times (as a group), starting with n copies of item 1. The last row of the table shows the optimal cost for packing the input up to and including the items of the present column, divided by n. The specific choices of the δ_i values further guarantee that various combinations of items cannot be packed side by side into a bin.

The next step is now to determine the value of the lower bound implied by this construction. As with all these constructions, this is done using a linear program. The linear program has a variable for each pattern (a pattern is a multiset of items that can be packed into a single bin). The objective is to minimize the total number of bins, i.e., the total number of times that any pattern is used. The constraints specify that each item must be packed, by counting how often each item occurs in each pattern and adding them all up.

item	1	2	3	4	5	6	7	8	9
height	$\frac{1}{7}+\varepsilon$	$\frac{1}{7}+\varepsilon$	$\frac{1}{7}+\varepsilon$	$\frac{1}{3}+\varepsilon$	$\frac{1}{3}+\varepsilon$	$\frac{1}{3}+\varepsilon$	$\frac{1}{2}+\varepsilon$	$\frac{1}{2}+\varepsilon$	$\frac{1}{2}+\varepsilon$
width	$\frac{1}{7}+\delta_1$	$\frac{1}{3}+\delta_2$	$\frac{1}{2}+\delta_3$	$\frac{1}{7}+\delta_4$	$\frac{1}{3}+\delta_5$	$\frac{1}{2}+\delta_6$	$\frac{1}{7}+\delta_7$	$\frac{1}{3}+\delta_8$	$\frac{1}{2}+\delta_9$
δ_i	$\frac{1}{42}-200\delta$	100δ	100δ	$\frac{1}{42}-20\delta$	10δ	10δ	$\frac{1}{42}-2\delta$	δ	δ
OPT$/n$	$\frac{1}{36}$	$\frac{1}{12}$	$\frac{1}{6}$	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{7}{12}$	$\frac{3}{4}$	1

Table 1: A lower bound input for two-dimensional bin packing

Van Vliet avoids a total enumeration of all the patterns by using a column generation approach. He starts with an arbitrary initial set of columns (packing patterns) that allow a feasible solution and then adds new columns if that is profitable. Having found an optimal solution to the linear program with a given set of columns, the dual variables are then used to search for new columns with negative reduced costs. As long as such columns are found, they are added to the linear program and the problem is reoptimized.

This approach works well here because the final values of the dual variables have a special structure which allows us to search for packing patterns with negative reduced cost very efficiently.

This construction is then extended to five layers of three items, eventually proving a lower bound of

$$\frac{15043464}{8124301} \approx 1.851.$$

The most difficult part of the proof is to check the feasibility of a dual solution, proving that the primal solution found is indeed optimal. This is done by finding the optimal solutions for certain sublists of the input.

Van Vliet notes that adding more items to each layer (using widths of the form $1/43 + \delta_j$ and $1/1807 + \delta_j$) is not effective, and that the implied lower bound even decreases. The problem is that these added items are quite small (in at least one dimension) and can be added for free to many packing patterns that we already have. Moreover, to make place for these additional items, the items of width $1/7 + \delta_j$ must be made slightly smaller and can therefore more easily be put side-by-side with items from previous levels.

Indeed, it seems to me that an improvement might be achieved by giving the items in a slightly different order instead of giving them level by level. A more effective order might be

$$a_1, a_2, a_4, a_5, a_3, a_6, a_7, a_8, a_9.$$

Moreover, I believe that it is not a good idea to go back to smaller items after giving larger ones, so I would furthermore suggest to combine the items a_3 and a_6 into one item, as well as the items a_7 and a_8. This means that the area of the items is now monotonically increasing as the sequence proceeds. This may very well be the way in which the unpublished lower bound of 1.907 was proved.

Of course, a relatively large gap to the current best upper bound of 2.55 by Han et al. [15] remains. It would be interesting to see how much of this is due to using Harmonic-type algorithms in both dimensions. Naturally, it is easily possible that we could improve the upper bound by *not* considering the dimensions completely separately, as all known algorithms do. However, it is quite hard to see how the Harmonic framework could be avoided. I conjecture that the true competitive ratio for this problem is greater than 2, however.

1.1 Rotations

An interesting variation of this problem is the case where items may be rotated over 90 degrees. Fujita and Hada [11] were the first to consider this problem and presented two algorithms for it. Epstein [8] showed that their first algorithm has a competitive ratio of at most 2.63889, and gave an improved algorithm with competitive ratio 2.54679. Her algorithm is bounded space and she furthermore shows that any bounded space algorithm must have competitive ratio at least 2.53536, so the remaining gap is very small. Recall that a bounded space algorithm has only a constant number of open bins at any time.

item	$4j-3$	$4j-2$	$4j-1$	$4j$
width	$\frac{1}{2}-(2k-2j+2)\varepsilon$	$\frac{1}{2}+(2k-2j+2)\varepsilon$	$\frac{1}{2}-(2k-2j+1)\varepsilon$	$\frac{1}{2}+(2k-2j+1)\varepsilon$
height	$\frac{1}{2}-(k-j+1)\varepsilon$	$\frac{1}{2}-(k-j+1)\varepsilon$	$\frac{1}{2}+(k-j+1)\varepsilon$	$\frac{1}{2}+(k-j+1)\varepsilon$
δ_i	$1/t_{k-j+1}+\varepsilon$			

Table 2: A part of a lower bound input for three-dimensional bin packing

Epstein [8] also presented a 2.45-competitive algorithm using unbounded space. It seems that there does not exist any lower bound for this problem, apart from the ones implied by lower bounds for the square packing problem. No online algorithm for square packing can be better than 1.6406-competitive [10], whereas the best known algorithm has a competitive ratio of 2.1187 [17]. It is somewhat surprising that the best known upper bound for the case of rotations is so much higher than that for squares. Intuitively, it would seem that having the option of rotating items should help significantly. However, of course it needs to be remembered that the optimal solution that we compare to also has the option to rotate items.

2 Three-dimensional online bin packing

For three dimensions, in order to keep the calculations manageable, van Vliet used the numbers in (1) only in one dimension, and made the sizes of the items close to 1/2 in the other two dimensions.

He considered the constructions DkW2H2, $2 \le k \le 5$, where k denotes the number of elements of the series (1) that are used for defining the depth of the items. The sizes of the items $a_1, \ldots, a_{4k}$ are given in Table 2. In this way, a lower bound of

$$\frac{1378064}{674429} \approx 2.043$$

is proved. Van Vliet conjectures that it should be possible to significantly increase this bound by using the series (1) in all three dimensions. The generation of the feasible packing patterns will obviously be a limiting factor in this approach, but some clever enumeration techniques might help to solve these problems. Han et al. [15] mention that their result implies a 4.3198-competitive algorithm for three dimensions, so the gap here is still quite large.

Also in this case, analogously to my suggestion for the two-dimensional case, it might be a good idea to ensure that the volumes of the items are roughly increasing over time, instead of reverting to using items with smaller volumes.

3 Dynamic bin packing

A multidimensional version of a dynamic bin packing model, which was introduced by Coffman et al. [3] for the one-dimensional case, was studied by Epstein and Levy [9]. This is an online model where items do not only arrive but may also leave. Each event is an arrival or a departure of an item. Durations are not known in advance, i.e., an algorithm is notified about the time that an items leaves only upon its departure. An algorithm may re-arrange the locations inside bins, but the items may not migrate between bins.

In two dimensions, they designed a 4.25-approximation algorithm for dynamical packing of squares, and provided a lower bound of 2.2307 on the performance ratio. For rectangles the upper and lower bounds are 8.5754 and 3.7 respectively. For three-dimensional cubes they presented an algorithm which is a 5.37037-approximation, and a lower bound of 2.117. For three-dimensional boxes, they supplied a 35.346-approximation algorithm and a lower bound of 4.85383. For higher dimensions, they define and analyze the algorithm NFDH for the offline box packing problem. This algorithm was studied before for rectangle packing (two-dimensional only) [4], and for square and cube packing for any dimension [20, 18], but not for box packing. For d-dimensional boxes they provided an upper bound of $2 \cdot 3.5^d$ and a lower bound of $d+1$. Note that, as already mentioned elsewhere, the best upper bound known for the regular offline multi-dimensional box packing problem is exponential as well. For d-dimensional cubes they provided an upper bound of

$$O\left(\frac{d}{\ln d}\right)$$

and a lower bound of 2.

One older paper by Coffman and Gilbert [5] studies a related problem. In this problem, squares of a bounded size, which arrive and leave at various times, must be kept in a single bin. The paper gives lower bounds on the size of such a bin, so that all squares can fit. It is not allowed to re-arrange the locations in the bin.

4 Higher dimensions

In three-dimensional strip packing, the input consists of a list of three-dimensional boxes that need to be packed into a strip with square base of size 1 and unbounded height. As in two-dimensional strip packing, the goal is to minimize the height used.

The only result I am aware of that is explicitly for three-dimensional strips is an upper bound of 2.89 achieved already back in 1992 by Li and Cheng [19]. However, it looks like the upper bound for two-dimensional bin packing of 2.55 by Han et al. [15] (as well as previous upper bounds [6, 21] that were below 2.89) should carry over to this problem in a straightforward way, similar to the way that one-dimensional bin packing algorithms can be used for two-dimensional strip packing while maintaining the same competitive ratio, as was shown by Han et al. [16]. The idea is to classify items by height and pack items of different height classes separately. This only loses an (additional) additive constant compared to the bin packing algorithm that is used.

The algorithm by Csirik and van Vliet [6] deserves separate attention, because it in fact works for any dimension d and achieves a competitive ratio of 1.691^d. To the best of my knowledge, this remains the only result for dimensions greater than 3. The best known lower bound is less than 3 for all d, so we in fact have an exponential gap between the upper and the lower bound here. It is very hard to see where the true answer might lie. For the similar problem of vector packing, this issue was recently resolved, as we will see next.

5 Vector packing

Another way to generalize bin packing to more dimensions is to consider vector (bin) packing. In this problem, a set of vectors need to be packed into multidimensional bins in such a way that the vector sum of vectors assigned to any bin does not exceed the all-one vector. Put another way, for each bin and each dimension, the sum of the coordinates of all vectors assigned to this bin must be at most 1. It has been known for a long time that First Fit achieves a competitive ratio of $d + 0.7$—this was shown by Garey et al. [13] already back in 1976. For a long time however, the best known lower bound for this problem was constant and at most 2, by Galambos et al. [12].

Motivated by this, Epstein [7] considered variable-sized vector packing. Here the algorithm is given a collection $\mathcal{B}$ of bin capacity profiles (d-dimensional vectors) which includes the all-one vector. She showed that for certain choices of $\mathcal{B}$, it is possible to prove linear lower bounds, whereas for other choices, a competitive ratio of $1+\varepsilon$ can be achieved. She conjectured that in the standard case described above, the competitive ratio is super-constant, but sublinear.

Recently, Azar et al. [1] showed that the ratio must be very close to linear by giving a lower bound of

$$\Omega\left(d^{1-\varepsilon}\right)$$

for any $\varepsilon > 0$. More generally, for the case where the bin size is B in every dimension, they give a lower bound of

$$\Omega\left(d^{1/B-\varepsilon}\right)$$

for any integer $B \geq 1$. These lower bounds are information-theoretic and hence apply also to exponential-time algorithms. The authors also give an upper bound of

$$O\left(d^{1/(B-1)} \log d^{B/(B-1)}\right)$$

(so the denominator of the exponent is off by 1). The authors note that for $B \geq \log d$ the bound becomes

$$O(\log d).$$

For the interesting special case of $\{0, 1\}$ vectors, the lower bound is

$$\Omega\left(d^{1/(B+1)} - \varepsilon\right)$$

and the upper bound is

$$O\left(d^{1/B} \log d^{(B+1)/B}\right).$$

The lower bound is proved using a lower bound strategy for online graph coloring from a 1992 paper by Halldorsson and Szegedy [14]. In that problem, a sequence of nodes arrives online together with edges to previously arrived neighbors, and the adversary is transparent: it reveals the color it gives to a node v_i immediately after the online algorithm makes its choice. This feature is actually important for the arguments in Azar et al. [1].

The upper bound works by first assigning the items to a set of virtually enlarged bins, and then assigning vectors from each virtual bin to real bins of size B. This is done first using a simple randomized algorithm which is then derandomized.

References

[1] Yossi Azar, Ilan Reuven Cohen, Seny Kamara, and F. Bruce Shepherd. Tight bounds for online vector bin packing. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 961–970. ACM, 2013.

[2] David Blitz, Andre van Vliet, and Gerhard J. Woeginger. Lower bounds on the asymptotic worst-case ratio of online bin packing algorithms. Unpublished manuscript, 1996.

[3] Edward G. Coffman, Michael R. Garey, and David S. Johnson. Dynamic bin packing. *SIAM J. Comput.*, 12:227–258, 1983.

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

[5] Edward G. Coffman and E. N. Gilbert. Dynamic, First Fit packings in two- or more dimensions. *Information and Control*, 61(1):1–14, 1984.

[6] Janos Csirik and Andre van Vliet. An online algorithm for multidimensional bin packing. *Oper. Res. Lett.*, 13:149–158, 1993.

[7] Leah Epstein. Two dimensional packing: the power of rotation. In *Proc. of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS'2003)*, pages 398–407, 2003.

[8] Leah Epstein. Two-dimensional online bin packing with rotation. *Theor. Comput. Sci.*, 411(31-33):2899–2911, 2010.

[9] Leah Epstein and Meital Levy. Dynamic multi-dimensional bin packing. *J. Discrete Algorithms*, 8(4):356–372, 2010.

[10] Leah Epstein and Rob van Stee. Online square and cube packing. *Acta Informatica*, 41(9):595–606, 2005.

[11] Satoshi Fujita and Takeshi Hada. Two-dimensional on-line bin packing problem with rotatable items. *Theoretical Computer Science*, 289(2):939–952, 2002.

[12] Gabor Galambos, H. Kellerer, and Gerhard J. Woeginger. A lower bound for online vector packing algorithms. *Acta Cybernet.*, 11:23–34, 1993.

[13] Michael R. Garey, Ronald L. Graham, David S. Johnson, and A. C. C. Yao. Resource constrained scheduling as generalized bin packing. *J. Combin. Theory Ser. A*, 21:257–298, 1976.

[14] M. M. Halldórsson and M. Szegedy. Lower bounds on on-line graph coloring. In *Proc. 3rd Symp. on Discrete Algorithms (SODA)*, pages 211–216. ACM/SIAM, 1992.

[15] Xin Han, Francis Y. L. Chin, Hing-Fung Ting, Guochuan Zhang, and Yong Zhang. A new upper bound 2.5545 on 2d online bin packing. *ACM Transactions on Algorithms*, 7(4):50, 2011.

[16] Xin Han, Kazuo Iwama, Deshi Ye, and Guochuan Zhang. Strip packing vs. bin packing. In Ming-Yang Kao and Xiang-Yang Li, editors, *AAIM*, volume 4508 of *Lecture Notes in Computer Science*, pages 358–367. Springer, 2007.

[17] Xin Han, Deshi Ye, and Yong Zhou. A note on online hypercube packing. *Central European Journal of Operations Research*, 18:221–239, 2010.

[18] Yoshiharu Kohayakawa, Flavio K. Miyazawa, Prabhakar Raghavan, and Yoshiko Wakabayashi. Multidimensional cube packing. 40(3):173–187, 2004.

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

[20] A. Meir and L. Moser. On packing of squares and cubes. *J. Combin. Theory*, 5:126–134, 1968.

[21] Steve S. Seiden and Rob van Stee. New bounds for multi-dimensional packing. *Algorithmica*, 36(3):261–293, 2003.

[22] Rob van Stee. SIGACT news online algorithms column 20: the power of harmony. *SIGACT News*, 43(2):127–136, 2012.

[23] André van Vliet. *Lower and upper bounds for online bin packing and scheduling heuristics.* PhD thesis, Erasmus University, Rotterdam, The Netherlands, 1995.

[24] Gerhard J. Woeginger. Personal communication, 2011.