

Bin Packaging

pp. 74-78

Vazirani, Vijay V., (©2001) Approximation algorithms, Berlin: Springer

Staff and students of London School of Economics and Political Science (LSE) are reminded that copyright subsists in this extract and the work from which it was taken. This Digital Copy has been made under the terms of a CLA licence which allows you to:

- access and download a copy;
- print out a copy;

Please note that this material is for use ONLY by students registered on the course of study as stated in the section below. All other staff and students are only entitled to browse the material and should not download and/or print out a copy.

This Digital Copy and any digital or printed copy supplied to or made by you under the terms of this Licence are for use in connection with this Course of Study. You may retain such copies after the end of the course, but strictly for your own personal use.

All copies (including electronic copies) shall include this Copyright Notice and shall be destroyed and/or deleted if and when required by London School of Economics and Political Science (LSE).

Except as provided for by copyright law, no further copying, storage or distribution (including by e-mail) is permitted without the consent of the copyright holder.

The author (which term includes artists and other visual creators) has moral rights in the work and neither staff nor students may cause, or permit, the distortion, mutilation or other modification of the work, or any other derogatory treatment of it, which would be prejudicial to the honour or reputation of the author.

Course of Study: MA421 - Advanced Algorithms

Title: Approximation algorithms

Name of Author: Vazirani, Vijay V.

Name of Publisher: Springer

9 Bin Packing

Consider the following problem.

Problem 9.1 (Bin packing) Given n items with sizes $a_1, \dots, a_n \in (0, 1]$, find a packing in unit-sized bins that minimizes the number of bins used.

This problem finds many industrial applications. For instance, in the stock-cutting problem, bins correspond to a standard length of paper and items correspond to specified lengths that need to be cut.

It is easy to obtain a factor 2 approximation algorithm for this problem. For instance, let us consider the algorithm called First-Fit. This algorithm considers items in an arbitrary order. In the i th step, it has a list of partially packed bins, say B_1, \dots, B_k . It attempts to put the next item, a_i , in one of these bins, in this order. If a_i does not fit into any of these bins, it opens a new bin B_{k+1} , and puts a_i in it. If the algorithm uses m bins, then at least $m - 1$ bins are more than half full. Therefore,

$$\sum_{i=1}^n a_i > \frac{m-1}{2}.$$

Since the sum of the item sizes is a lower bound on OPT, $m - 1 < 2\text{OPT}$, i.e., $m \leq 2\text{OPT}$ (see Notes for a better analysis). On the negative side:

Theorem 9.2 For any $\varepsilon > 0$, there is no approximation algorithm having a guarantee of $3/2 - \varepsilon$ for the bin packing problem, assuming $\mathbf{P} \neq \mathbf{NP}$.

Proof: If there were such an algorithm, then we show how to solve the NP-hard problem of deciding if there is a way to partition n nonnegative numbers a_1, \dots, a_n into two sets, each adding up to $\frac{1}{2} \sum_i a_i$. Clearly, the answer to this question is ‘yes’ iff the n items can be packed in 2 bins of size $\frac{1}{2} \sum_i a_i$. If the answer is ‘yes’ the $3/2 - \varepsilon$ factor algorithm will have to give an optimal packing, and thereby solve the partitioning problem. \square

9.1 An asymptotic PTAS

Notice that the argument in Theorem 9.2 uses very special instances: those for which OPT is a small number, such as 2 or 3, even though the number

of items is unbounded. What can we say about ‘typical’ instances, those for which OPT increases with n ?

Theorem 9.3 For any ε , $0 < \varepsilon \leq 1/2$, there is an algorithm \mathcal{A}_ε that runs in time polynomial in n and finds a packing using at most $(1 + 2\varepsilon)\text{OPT} + 1$ bins.

The sequence of algorithms, \mathcal{A}_ε , form an *asymptotic polynomial time approximation scheme* for bin packing, since for each $\varepsilon > 0 \exists N > 0$, and a polynomial time algorithm in this sequence, say \mathcal{B} , such that \mathcal{B} has an approximation guarantee of $1 + \varepsilon$ for all instances having $\text{OPT} \geq N$. However, Theorem 9.3 should not be considered a practical solution to the bin packing problem, since the running times of the algorithms \mathcal{A}_ε are very high.

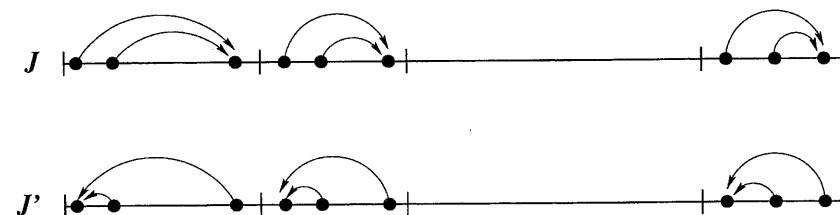
We will prove Theorem 9.3 in three steps.

Lemma 9.4 Let $\varepsilon > 0$ be fixed, and let K be a fixed nonnegative integer. Consider the restriction of the bin packing problem to instances in which each item is of size at least ε and the number of distinct item sizes is K . There is a polynomial time algorithm that optimally solves this restricted problem.

Proof: The number of items in a bin is bounded by $\lceil 1/\varepsilon \rceil$. Denote this by M . Therefore, the number of different bin types is bounded by $R = \binom{M+K}{M}$ (see Exercise 9.4), which is a (large!) constant. Clearly, the total number of bins used is at most n . Therefore, the number of possible feasible packings is bounded by $P = \binom{n+R}{R}$, which is polynomial in n (see Exercise 9.4). Enumerating them and picking the best packing gives the optimal answer. \square

Lemma 9.5 Let $\varepsilon > 0$ be fixed. Consider the restriction of the bin packing problem to instances in which each item is of size at least ε . There is a polynomial time approximation algorithm that solves this restricted problem within a factor of $(1 + \varepsilon)$.

Proof: Let I denote the given instance. Sort the n items by increasing size, and partition them into $K = \lceil 1/\varepsilon^2 \rceil$ groups each having at most $Q = \lfloor n\varepsilon^2 \rfloor$ items. Notice that two groups may contain items of the same size.



Construct instance J by rounding up the size of each item to the size of the largest item in its group. Instance J has at most K different item sizes.

Therefore, by Lemma 9.4, we can find an optimal packing for J . Clearly, this will also be a valid packing for the original item sizes. We show below that $\text{OPT}(J) \leq (1 + \varepsilon)\text{OPT}(I)$, thereby proving the lemma.

The following clever argument accomplishes this. Let us construct another instance, say J' , by rounding down the size of each item to that of the smallest item in its group. Clearly $\text{OPT}(J') \leq \text{OPT}(I)$. The crucial observation is that a packing for instance J' yields a packing for all but the largest Q items of instance J (Exercise 9.6 asks for a formal proof). Therefore,

$$\text{OPT}(J) \leq \text{OPT}(J') + Q \leq \text{OPT}(I) + Q.$$

Since each item in I has size at least ε , $\text{OPT}(I) \geq n\varepsilon$. Therefore, $Q = \lfloor n\varepsilon^2 \rfloor \leq \varepsilon\text{OPT}(I)$. Hence, $\text{OPT}(J) \leq (1 + \varepsilon)\text{OPT}(I)$. \square

Proof of Theorem 9.3: Let I denote the given instance, and I' denote the instance obtained by discarding items of size $< \varepsilon$ from I . By Lemma 9.5, we can find a packing for I' using at most $(1 + \varepsilon)\text{OPT}(I')$ bins. Next, we start packing the small items (of size $< \varepsilon$) in a First-Fit manner in the bins opened for packing I' . Additional bins are opened if an item does not fit into any of the already open bins.

If no additional bins are needed, then we have a packing in $(1 + \varepsilon)\text{OPT}(I')$ $\leq (1 + \varepsilon)\text{OPT}(I)$ bins. In the second case, let M be the total number of bins used. Clearly, all but the last bin must be full to the extent of at least $1 - \varepsilon$. Therefore, the sum of the item sizes in I is at least $(M - 1)(1 - \varepsilon)$. Since this is a lower bound on OPT , we get

$$M \leq \frac{\text{OPT}}{(1 - \varepsilon)} + 1 \leq (1 + 2\varepsilon)\text{OPT} + 1,$$

where we have used the assumption that $\varepsilon \leq 1/2$. Hence, for each value of ε , $0 < \varepsilon \leq 1/2$, we have a polynomial time algorithm achieving a guarantee of $(1 + 2\varepsilon)\text{OPT} + 1$. \square

Algorithm \mathcal{A}_ε is summarized below.

Algorithm 9.6 (Algorithm \mathcal{A}_ε for bin packing)

1. Remove items of size $< \varepsilon$.
2. Round to obtain constant number of item sizes (Lemma 9.5).
3. Find optimal packing (Lemma 9.4).
4. Use this packing for original item sizes.
5. Pack items of size $< \varepsilon$ using First-Fit.

9.2 Exercises

9.1 Give an example on which First-Fit does at least as bad as $5/3 \cdot \text{OPT}$.

9.2 (Johnson [156]) Consider a more restricted algorithm than First-Fit, called Next-Fit, which tries to pack the next item only in the most recently started bin. If it does not fit, it is packed in a new bin. Show that this algorithm also achieves factor 2. Give a factor 2 tight example.

9.3 (C. Kenyon) Say that a bin packing algorithm is *monotonic* if the number of bins it uses for packing a subset of the items is at most the number of bins it uses for packing all n items. Show that whereas Next-Fit is monotonic, First-Fit is not.

9.4 Prove the bounds on R and P stated in Lemma 9.4.

Hint: Use the fact that the number of ways of throwing n identical balls into k distinct bins is $\binom{n+k-1}{n}$.

9.5 Consider an alternative way of establishing Lemma 9.5. All items having sizes in the interval $(\varepsilon(1 + \varepsilon)^r, \varepsilon(1 + \varepsilon)^{r+1}]$ are rounded up to $\min(\varepsilon(1 + \varepsilon)^{r+1}, 1)$, for $r \geq 0$. Clearly, this yields a constant number of item sizes. Does the rest of the proof go through?

Hint: Consider the situation that there are lots of items of size $1/2$, and $1/2 \neq \varepsilon(1 + \varepsilon)^r$ for any $r \geq 0$.

9.6 Prove the following statement made in Lemma 9.5, “A packing for instance J' yields a packing for all but the largest Q items of instance J .”

Hint: Throw away the Q largest items of J and the Q smallest items of J' , and establish a domination.

9.7 Use the fact that integer programming with a fixed number of variables is in **P** to give an alternative proof of Lemma 9.4. (Because of the exorbitant running time of the integer programming algorithm, this variant is also impractical.)

9.8 Show that if there is an algorithm for bin packing having a guarantee of $\text{OPT}(I) + \log^2(\text{OPT}(I))$, then there is a fully polynomial approximation scheme for this problem.

9.9 (C. Kenyon) Consider the following problem.

Problem 9.7 (Bin covering) Given n items with sizes $a_1, \dots, a_n \in (0, 1]$, maximize the number of bins opened so that each bin has items summing to at least 1.

Give an asymptotic PTAS for this problem when restricted to instances in which item sizes are bounded below by c , for a fixed constant $c > 0$.

Hint: The main idea of Algorithm 9.6 applies to this problem as well.

9.3 Notes

The first nontrivial bin packing result, showing that First-Fit requires at most $(17/10)\text{OPT} + 3$ bins, was due to Ullman [256]. The asymptotic PTAS is due to Fernandez de la Vega and Lueker [92]. An improved algorithm, having a guarantee of $\text{OPT}(I) + \log^2(\text{OPT}(I))$ was given by Karmarkar and Karp [170]. For further results, see the survey of Coffman, Garey, and Johnson [52]. The result cited in Exercise 9.7, showing that integer programming with a fixed number of variables is in \mathbf{P} , is due to Lenstra [192]. Bin packing has also been extensively studied in the on-line model. For these and other on-line algorithms see Borodin and El-Yaniv [32].