Annales Univ. Sci. Budapest., Sect. Comp. 46 (2017) 215–222

ON OPTIMAL UNIT FRACTION BIN PACKING

Zsolt Németh (Budapest, Hungary)

Dedicated to the memory of Antal Iványi

Communicated by Ferenc Schipp

(Received June 17, 2017; accepted July 20, 2017)

Abstract. In this paper, we consider a variant of the classical bin packing problem, called unit fraction bin packing (UFBP), where all item sizes are unit fractions. It turns out that the number of bins used by an optimal packing in UFBP is very close to the sum of item sizes. We investigate the relations between these two values, and give a polynomial time optimal algorithm for some cases.

1. Introduction

In the classical bin packing problem, given a sequence $a := (a)_{i=1}^{L}$ of length $L \in \mathbb{N}^+ := \{1, 2, \ldots\}$ of item sizes $a_1, a_2, \ldots, a_L$, where $0 < a_i \leq 1$ for all $i = 1, 2, \ldots, n$. The goal is to pack these items in unit sized *bins* using as few as possible such that the total size of items packed in one bin does not exceed 1. Regarding computational complexity, finding the necessary, but also sufficient (referred to as *optimal* later) number of bins is NP-hard, and the decision problem – deciding if the objects will fit into a specified number of bins – is NP-complete [7].

The problem is closely related to some operational research problems (e.g. cutting stock and knapsack problems), and also to many scheduling problems

Key words and phrases: Bin packing, unit fraction bin packing, optimal algorithm, first fit.
2010 Mathematics Subject Classification: 68Q17, 68R05.

in information theory, like multiprocessor scheduling [4] and virtual machine memory scheduling [9].

We note that in many cases, instead of finding an optimal solution, simple heuristic algorithms like *first fit* and *best fit* are applied to either the original or the sorted (*decreasing*) sequence of object sizes. The performance of these methods compared to an optimal one is widely investigated, see e.g. [10], [2] and [3].

A variant of the classical problem is the *unit fraction bin packing* problem (or UFBP), in which all item sizes are unit fractions, i.e. of the form $\frac{1}{k}$ for some $k \in \mathbb{N}^+$. It turns out that this case is connected to the so called *windows scheduling* problem [1], [5]. While the complexity class of finding an optimal packing is unknown, we will see that in most comparisons the UFBF problem behaves more nicely than the classical problem.

Let us introduce the notation

$$S(a) := \sum_{i=1}^{L} a_i$$

for the total size of objects, and $H(a) := \lceil S(a) \rceil$. For an algorithm A, we will denote the number of bins used by the algorithm for a sequence a by $A(a)$. Specially, the number of bins used by an optimal algorithm for a is $OPT(a)$. It is clear that $H(a) \leq OPT(a)$ for arbitrary a.

It is known [6] that for the bin packing problem, $OPT(a) \leq 2H(a) - 1$ for any a, and there exists a sequence for which the equation holds. So the difference between the total size of objects and the total volume of used bins can be quite large. On the other hand, for UFBP the inequality $OPT(a) \leq H(a)+1$ was proved by Bar-Noy et al. [1]. In the same paper, they also showed that some well-known heuristic approaches have much better asymptotical properties for UFBP. In fact, for the so-called *first fit decreasing* (FFD) algorithm, we have $OPT(a) \leq FFD(a) \leq H(a) + 1$. Note that FFD is not an optimal algorithm, e.g. letting $a = \langle \frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \rangle$, one can see that an optimal packing uses 2 bins while FFD uses 3.

Now it is clear that for any sequence a, $OPT(a)$ is either $H(a)$ or $H(a)+1$. In this paper, we investigate conditions under which $OPT(a) = H(a)$ holds. We will show that if the difference of $S(a)$ and $H(a)$ surpasses a certain threshold, this equation holds regardless of a. In these cases, a simple polynomial time optimal packing algorithm can be given.

2. Preliminary results

First, we observe that the theoretical bound given in [1], i.e. for any sequence a we have $OPT(a) \leq H(a) + 1$, is sharp.

Theorem 2.1. *There exists a sequence a of item sizes, such that $OPT(a) = H(a) + 1$.*

Proof. Let $a = \langle \frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5} \rangle$. For this sequence we have $H(a) = 2$. The object of size $\frac{1}{2}$ can be packed either with an $\frac{1}{3}$ or two $\frac{1}{5}$-s. In both cases the sum of the remaining sizes is greater than 1, so one additional bin is not sufficient. Consequently $OPT(a) = 3$. ■

We note that in this counterexample, the sum of sizes $S(a) = \frac{59}{30}$ is relatively close to $H(a) = 2$.

Next, we prove that if the difference of $S(a)$ and $H(a)$ is large enough, $H(a) = OPT(a)$ must hold.

Theorem 2.2. *For any sequence a, if $S(a) \leq H(a) - \frac{1}{2}$, then $OPT(a) = H(a)$.*

Proof. Indirectly, suppose that there exists at least one sequence which is a counterexample to our claim. Consider the ones having the least possible $H(a)$ value, and among these having the minimal number of items. Denote such a sequence by a. Obviously $1 \notin a$, or the example is not minimal.

Let $m = \min a$ be the smallest object size in a. Now if we remove an item of size m from the sequence, then the resulting sequence b can be packed into at most $H(a)$ bins. Indeed, either $H(b) = H(a) - 1$ holds for b, so its elements fit into $H(a)$ bins, or $H(b) = H(a)$ and $S(b) \leq H(b) - \frac{1}{2}$, so b fits into $H(b) = H(a)$ bins by the minimality of a. Now the removed item m cannot be fitted into any of these bins, otherwise a is not a counterexample. Hence, the sum of item sizes must be greater than $1 - m$ in each bin.

Consequently we have

$$H(a) \cdot (1 - m) + m < S(a) \leq H(a) - \frac{1}{2} \quad \Leftrightarrow \quad m > \frac{1/2}{H(a) - 1}. \tag{2.1}$$

Since m is the minimal item size in a, the rightmost fraction actually estimates every element of the sequence – they are unit fractions with denominator less than $\frac{H(a)-1}{1/2}$.

Using this bound and the minimality of a, we will deduce an upper bound on $S(a)$. Notice that, since a has minimal number of items, there cannot be more than one item of size $\frac{1}{2k}$, $k \in \mathbb{N}^+$: otherwise just replace two of these by a

single $\frac{1}{k}$, and we have a smaller counterexample. Similarly, a can have at most $2k$ items of $\frac{1}{2k+1}$ size for any $k \in \mathbb{N}^+$, otherwise remove $2k+1$ of those and we have a counterexample for smaller total size. Consequently

$$S(a) < \frac{1}{2} + \sum_{k=2}^{H(a)-1} \left(\frac{2k-2}{2k-1} + \frac{1}{2k} \right) \leq \frac{1}{2} + \sum_{k=2}^{H(a)-1} 1 = H(a) - \frac{3}{2}. \tag{2.2}$$

But, by definition, $S(a) > H(a) - 1$, contradicting the above inequality. ■

In the next section we are going to tighten this result, eventually obtaining

Theorem 2.3. *If $S(a) \leq H(a) - \frac{10}{31}$ holds for a, then $OPT(a) = H(a)$.*

3. Further improvements

In this section, we are looking for ways to improve the previous result. It is clear that there exists $c_0 > 0$, such that the statement

Proposition 3.1. *If $S(a) \leq H(a) - c$ holds for a, then $OPT(a) = H(a)$.*

holds for every $c > c_0$, and for $c < c_0$ there are counterexamples. Note that Theorem 2.1 and 2.2 implies that c_0 is between $\frac{1}{30}$ and $\frac{1}{2}$. Better upper bounds for c_0 follow in this section.

For a given value of c, we may prove this the same way as Theorem 2.2 if we meet certain requirements. Notice that the proof has two main ingredients, depending on c. First, in order to have a minimal counterexample a, the item sizes must be bounded: similarly to (2.1), for the minimal element m now we have

$$H(a) \cdot (1-m) + m < S(a) \leq H(a) - c \quad \Leftrightarrow \quad m > \frac{c}{H(a)-1}. \tag{3.1}$$

So the denominators in a must be smaller than $\frac{H(a)-1}{c}$.

The other main ingredient was the upper estimation for $S(a)$, based on the minimal element. Our initial approach to obtain (2.2) can be improved for a better result: for any $k \in \mathbb{N}^+$, a can have at most $d(k) - 1$ items of size $\frac{1}{k}$, where $d(k)$ denotes the smallest divisor of k greater than 1. Otherwise, we could replace $d(k)$ pieces of $\frac{1}{k}$ by a single $\frac{d(k)}{k}$ (unit fraction) item.

So we have

$$S(a) \leq \sum_{k=2}^{\frac{1}{m}} \frac{d(k)-1}{k},$$

and now we are able to finish the proof if c is such a value that

$$\sum_{k=2}^{\frac{1}{m}} \frac{d(k)-1}{k} \leq H(a)-1$$

holds for every a and m satisfying (3.1), thus contradicting $S(a) > H(a)-1$. We are going to find the minimal c value for this.

Let us introduce the notation

$$G(n) := \sum_{k=2}^{n} \frac{d(k)-1}{k}.$$

Denote by $n(H)$ the largest integer n such that $G(n) < H-1$. For any minimal counterexample sequence a with minimal value at most $\frac{1}{n}$,

$$\frac{1}{n} > \frac{c}{H(a)-1}$$

must hold. Letting $\frac{1}{n+1} = \frac{c}{H(a)-1}$ gives $c = \frac{H(a)-1}{n+1}$, a sufficient c value for the given $H(a)$.

Now to obtain a value of c satisfying these conditions for every $H(a)$, we should choose it to be the supremum of the expression $\frac{H-1}{n(H)+1}$ over all $H = H(a) \geq 2$.

Lemma 3.1.

$$\sup_{2 \leq H} \frac{H-1}{n(H)+1} = \frac{3}{7}.$$

Proof. We show that if H is large enough, the argument tends to zero. Denoting the greatest divisor of k other than itself by $e(k)$, it is clear that $e(k) = 1$ for prime k, and $e(k) \geq \sqrt{k}$ otherwise. So, denoting the number of primes less than or equal to n by $\pi(n)$,

$$G(n) = \sum_{k=2}^{n} \frac{d(k)-1}{k} < \sum_{k=2}^{n} \frac{1}{e(k)} < \pi(n) + \sum_{k=2}^{n} \frac{1}{\sqrt{k}},$$

and it can be proved by induction that

$$\sum_{k=2}^{n} \frac{1}{\sqrt{n}} \leq 2\sqrt{n}.$$

It is known [8] for the number of primes function that

$$\pi(n) < 1.25506 \, \frac{n}{\ln n},$$

so we have

$$G(n) < 1.25506\frac{n}{\ln n} + 2\sqrt{n}.$$

By the definition of $n(H)$ we have $G(n(H)) < H - 1 \leq G(n(H)+1)$. Consequently

$$\frac{H-1}{n(H)+1} \leq \frac{G(n(H)+1)}{n(H)+1} < \frac{1.25506}{\ln n(H)} + \frac{2}{\sqrt{n(H)}}. \tag{3.2}$$

The right hand side is strictly decreasing, tending to 0 as $H \to +\infty$ (so $n(H) \to$ $\to +\infty$).

To find the supremum, we used a computer program to evaluate $\frac{H-1}{n(H)+1}$ starting at $H = 2$. We keep track of the greatest encountered value, and stop when the upper bound given by (3.2) falls below this value. At this point we have the supremum, which is $\frac{3}{7}$ in our case. ■

Corollary 3.1. *If $S(a) \leq H(a) - \frac{3}{7}$ holds for a, then $OPT(a) = H(a)$.*

In fact, one may further enhance this result by noting that in a minimal counterexample, there cannot be items of sizes $\frac{1}{3}$ and $\frac{1}{6}$ at the same time, because then we could replace them by a single $\frac{1}{2}$. More generally, we can see that a minimal counterexample cannot have a subsequence of items with the total size being a unit fraction itself.

Another possible improvement can be made by observing that for smaller values of $H(a)$, there are very few candidates for a minimal counterexamples, so one may generate them and find the optimal packing for these cases.

Without further discussion, we state that after brute-forcing the optimal packing for the possible sequences with $H(a) \leq 10$, and a few additional tricks to avoid some unit fraction sums, we have

Proposition 3.2. *If $S(a) \leq H(a) - \frac{10}{31}$ holds for a, then $OPT(a) = H(a)$.*

4. An optimal algorithm for $OPT(a) = H(a)$

In this section, we give a simple, polynomial time optimal algorithm for sequences a satisfying $S(a) \leq H(a) - \frac{3}{7}$. Let us consider the algorithm making the following two steps.

1) First, we reduce the input sequence a to a new one, which is a candidate for being a counterexample in the $c = \frac{3}{7}$ case. Such a sequence contains a unit fraction $\frac{1}{k}$, $k \in \mathbb{N}^+$ at most $d(k) - 1$ times.

For this, we rearrange the items in increasing order, and, starting from the smallest, we count the ones having the same size. If at any point the number of $\frac{1}{k}$ items reaches $d(k)$, we remove these $d(k)$ items and insert a single $\frac{d(k)}{k}$ item (indicating that those $d(k)$ items should be packed together), keeping the increasing ordering. When we reach the end of the sequence, we pack every item of size 1 to separate bins, removing them from the sequence.

This way, since we never insert a smaller item then the ones currently being removed, we only have to process the sequence a single time to reduce the amount of $\frac{1}{k}$ items to at most $d(k) - 1$. Denote this new sequence by b. Since we only removed items of size 1, $S(b) \leq H(b) - \frac{3}{7}$ must hold.

2) By the proof of Corollary 3.1, we know that items of sizes $> \frac{3/7}{H(b)-1}$ have total size less than $H(b) - 1$, meaning that *first fit decreasing* (FFD) will pack them into at most $H(b)$ bins (see Introduction).

Now the smaller items can be packed into any of these $H(b)$ bins where they fit. Indeed, if there is one of size m' which cannot be packed, it implies that for the total size

$$H(b)(1 - m') + m' < S(b) \leq H(b) - \frac{3}{7},$$

consequently $m' > \frac{3/7}{H(b)-1}$, a contradiction.

So we proved that the sequence b can be packed into at most $H(b)$ bins using the FFD algorithm, obtaining an optimal packing for the original sequence a.

Using a similar idea, an optimal algorithm may be constructed for the case $S(a) \leq H(a) - \frac{10}{31}$.

References

[1] **Bar-Noy, A., R.E. Ladner and T. Tamir,** Windows Scheduling as a Restricted Version of Bin Packing, *ACM Transactions on Algorithms,* **3(3)** (2007), Article No. 28.

[2] **Dósa, Gy.,** The tight bound of First Fit Decreasing bin-packing algorithm is $FFD(I) \leq (11/9)OPT(I) + 6/9$, in: *Chen, Bo; Paterson, Mike; Zhang, Guochuan: Combinatorics, Algorithms, Probabilistic and Experimental Methodologies,* Springer Berlin Heidelberg, 2007, pp. 1-11.

[3] **Dósa, Gy. and J. Sgall,** First Fit bin packing: A tight analysis, in: *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013),* Dagstuhl, Germany, 2013, pp. 538-549.

[4] **Garey, M.R. and D.S. Johnson,** *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W. H. Freeman and Company, 1979, p. 238.

[5] **Han X., C. Peng, D. Ye, D. Zhang and Y. Lan,** Dynamic bin packing with unit fraction items revisited, *Information Processing Letters,* **110(23)** (2010), 1049–1054.

[6] **Iványi, A.,** Performance bounds for simple bin packing algorithms, *Ann. Univ. Sci. Budapest. Sect. Comput.,* **5** (1984), 77–82.

[7] **Korte, B. and J. Vygen,** Bin-Packing, in: *Combinatorial Optimization: Theory and Algorithms. Algorithms and Combinatorics Vol. 21,* Springer Berlin Heidelberg, 2006, pp. 426-441.

[8] **Rosser, J.B. and L. Schoenfeld,** Approximate formulas for some functions of prime numbers, *Illinois J. Math.,* **6** (1962), 64-94.

[9] **Sindelar, M., R. Sitaraman and P. Shenoy,** Sharing-Aware Algorithms for Virtual Machine Colocation, if: *Proceedings of 23rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), San Jose, CA, June 2011*, ACM, 2011, pp. 367-378.

[10] **Yue, M.,** A simple proof of the inequality $FFD(L) \leq 11/9\ OPT(L) + 1$, $\forall L$ for the FFD bin-packing algorithm, *Acta Mathematicae Applicatae Sinica,* **7(4)** (1991), 321-331.

Zs. Németh
Department of Numerical Analysis
Eötvös Loránd University
H–1117 Budapest
Pázmány Péter sétány 1/C
Hungary
birka0@gmail.com