MTAT.03.286: Design and Analysis of Algorithms **University of Tartu**

Tutorial 14

Instructor: Vitaly Skachek; Assistants: Behzad Abdolmaleki, Oliver-Matis Lill, Eldho Thomas.

Bin packing[1]

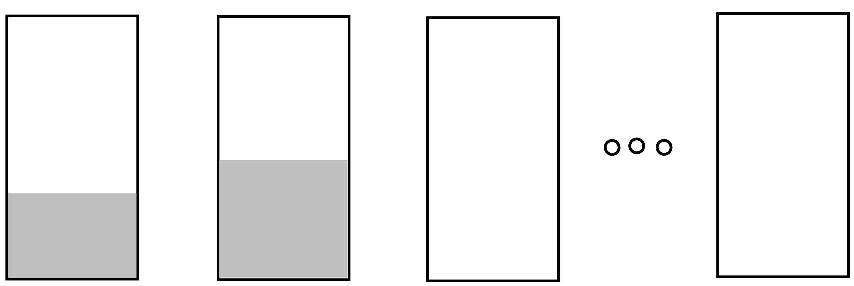
Problem. Consider n items with sizes $a_1, a_2, \cdots, a_n$, all $a_i \in (0, 1]$. Find a packing of these items into unit-sized bins that minimizes the number of bins used in the packing.

Algorithm First-Fit

The proposed algorithm is greedy-type algorithm.

1. Consider elements in an arbitrary order.
2. Let $B_1, B_2, \cdots, B_k$ be the list of partially packed bins.
3. Try to put the next item, a_i, in one of these bins, in that order.
4. If a_i does not fit into any of these bins, open a new bin B_{k+1}, and put a_i there.

Observation: if the algorithm uses m bins, then at least $m - 1$ bins are more than half full.



Indeed, if there are two half-empty bins, then why did we use the second bin, if the first bin is still half-empty?

We obtain that

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

Since $\sum_{i=1}^{n} a_i \leq \mathsf{OPT}$, we have that

$$m - 1 < 2 \cdot \mathsf{OPT} \text{ or, equivalently, } m \leq 2 \cdot \mathsf{OPT} .$$

[1]Additional reading: Section 9 in the book of V.V. Vazirani "Approximation Algorithms".

Here, the approximation factor of the proposed algorithm is 2.

Next, we will show a better approximation algorithm, which is PTAS.

Theorem 1 *For any ε, $0 < \varepsilon \le \frac{1}{2}$, there is an algorithm that runs in time polynomial in n, and finds a packing using at most $(1+2\varepsilon) \cdot \mathsf{OPT} + 1$ bins.*

In order to prove this theorem, we will need the following two lemmas.

Lemma 2 *Let $k \in \mathbb{N}$ be a fixed nonnegative integer. Consider a restriction of the bin packing problem to instances in which each item is of size at least $\varepsilon > 0$, ε is a constant, and the number of distinct item sizes is K. There is a polynomial-time exact (non-approximation) algorithm that optimally solves this problem in time polynomial in n.*

Proof. The number of possible items in a bin is bounded from above by $M \triangleq \lfloor 1/\varepsilon \rfloor$. Therefore, the number of different types of packings in one bin is bounded from above by $R = \binom{M+K}{M}$ (for a packing, we choose up to M sizes out of K possible sizes, possibly with repetitions).

The total number of bins is at most n. We can assume that the bins are identical. Then, imagine that each bin "selects" a type of packing out of R types (possibly with repetitions). That gives us that the maximal possible number of packings of n elements is $\binom{n+R}{R}$. The optimal packing can be chosen by going through all possible packing and choosing one that uses the minimal number of bins. That gives us that the exact solution can be found with complexity:

$$\binom{n+R}{R} = O((n+R)^R) ,$$

which is polynomial in n (under assumption that K and ε are constants). $\square$

Note: the proposed exact algorithm has complexity exponential in $1/\varepsilon$, and therefore the resulting algorithm is not FPTAS.

Lemma 3 *Let $0 < \varepsilon \le \frac{1}{2}$ be fixed. Consider a restriction of the bin packing problem to instances in which each element is of size at least ε. There exists an efficient approximation algorithm that solves this problem within an approximation factor $1+\varepsilon$.*

Proof. Let $\mathcal{I}$ denotes the given instance of the problem. Sort n items by increasing order of their sizes. Partition them into $K = \lceil \frac{1}{\varepsilon^2} \rceil$ groups, each group having at most $\mathcal{Q} = \lfloor n\varepsilon^2 \rfloor$ elements (add virtual items of size 0 if needed).

Construct an instance $\mathcal{J}$ by rounding the size of each item up to the size of the largest item in its group. Instance $\mathcal{J}$ has at most K different item sizes. By Lemma 2, we can find an optimal packing for $\mathcal{J}$. It will also be a valid packing for the original sizes. We show below that

$$\mathsf{OPT}(\mathcal{J}) \le (1+\varepsilon) \cdot \mathsf{OPT}(\mathcal{I}) .$$

Let us construct an instance $\mathcal{J}'$ by rounding down the size of each item to that of the smallest item in the group. Then,

$$\mathsf{OPT}(\mathcal{J}') \le \mathsf{OPT}(\mathcal{I}) .$$

The packing for $\mathcal{J}'$ yields a packing for all but the largest $\mathcal{Q}$ items in $\mathcal{J}$. Why?

Therefore,

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

Size of each element in $\mathcal{I}$ has size of at least ε, $\mathsf{OPT}(\mathcal{I}) \geq n\varepsilon$. Therefore, $\mathcal{Q} = \lfloor n\varepsilon^2 \rfloor \leq \varepsilon \cdot \mathsf{OPT}(\mathcal{I})$. Hence,

$$\mathsf{OPT}(\mathcal{J}) \leq (1+\varepsilon) \cdot \mathsf{OPT}(\mathcal{I}) .$$

□

Proof of Theorem 1. Let $\mathcal{I}$ denote the given instance, and $\mathcal{I}'$ denote the instance obtained by discarding items of size $< \varepsilon$ from $\mathcal{I}$. By Lemma 3, we can find packing for $\mathcal{I}'$ by using $(1+\varepsilon)\cdot\mathsf{OPT}(\mathcal{I}')$ bins.

Next, we pack the small items (those of size $< \varepsilon$) in a First-Fit manner in the bins opened for packing $\mathcal{I}'$. If no additional bins are needed, then we have a packing in

$$\leq (1+\varepsilon) \cdot \mathsf{OPT}(\mathcal{I}') \leq (1+\varepsilon) \cdot \mathsf{OPT}(\mathcal{I})$$

bins. Otherwise (additional bins are needed), let M be the total number of bins used. Clearly, all but the last bin must be full up to the extent of at least $1-\varepsilon$ (otherwise, we can add additional size $< \varepsilon$ elements). Therefore, the sum of sizes of elements in these bins $\geq (M-1)(1-\varepsilon)$. Since $\mathsf{OPT}(\mathcal{I}) \geq (M-1)(1-\varepsilon)$, we have

$$M \leq \frac{\mathsf{OPT}(\mathcal{I})}{1-\varepsilon} + 1 \leq (1+2\varepsilon) \cdot \mathsf{OPT}(\mathcal{I}) + 1 ,$$

where we used the assumption $\varepsilon \leq 1/2$. [2] □

Conclusion: for any value of ε, $0 < \varepsilon \leq \frac{1}{2}$, we have a polynomial-time in n algorithm for bin packing that produces a solution with the number of bins $\leq (1+2\varepsilon) \cdot \mathsf{OPT}(\mathcal{I}) + 1$.

Summary of the algorithm

1. Remove items of size ε.
2. Round up the sizes to obtain a constant number K of item sizes.
3. Find an optimal packing.
4. Use this packing for the original sizes.
5. Pack the elements of size $< \varepsilon$ by using First-Fit.

[2] Indeed, for $\varepsilon \leq 1/2$, $(1-\varepsilon)(1+2\varepsilon) = 1 + \varepsilon - 2\varepsilon^2 \geq 1$, and therefore $\frac{1}{1-\varepsilon} \leq 1 + 2\varepsilon$.