Algorithm Design (XVII)

Approximation Algorithms II

Guoqiang Li

School of Software, Shanghai Jiao Tong University

Bin Packing

Bin Packing: Problem Statement

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

An 2-approximation Algorithm

First-Fit Algorithm:

- Consider items in arbitrary order.
- In the i-th step, it has a list of partially packed bins, say $B_1, \ldots, B_k$.
- 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.

Analysis

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, $m - 1 < 2 \cdot \mathrm{OPT}$, i.e., $m \leq 2 \cdot \mathrm{OPT}$.

A Hardness Result

Theorem

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

Proof.

If there were such an algorithm, then the NPC problem of deciding if there is a way to partition n nonnegative numbers $a_1, \ldots, a_n$ into two sets, each adding up to $1/2 \sum_i a_i$.

The answer to this question is "yes" iff the n items can be packed in 2 bins of size $1/2 \sum_i a_i$.

If the answer is "yes" the $3/2 - \epsilon$ factor algorithm will have to give an optimal packing.

APTAS

Definition

An asymptotic polynomial-time approximation scheme (APTAS) is a family of algorithm $\{A_\epsilon\}$ along with a constant c where there is an algorithm A_ϵ for each $\epsilon > 0$ such that A_ϵ returns a solution of value at most $(1+\epsilon)\mathrm{OPT} + c$ for minimization problems.

For any ϵ, $0 < \epsilon \leq 1/2$, there is an algorithm A_ϵ that runs in time polynomial in n and finds a packing using at most $(1 + 2\epsilon)\mathrm{OPT} + 1$ bins.

We will introduce the algorithm in three steps.

Instances with Large Items

Lemma

Let $\epsilon > 0$ 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 $\lfloor 1/\epsilon \rfloor$. Denote this by M. Therefore, the number of different bin types is bounded by

$$R = \binom{M+K}{M}$$

which is a large constant.

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.

Enumerating them and picking the best packing gives the optimal answer.

k Composition of M

$$x_1 + x_2 + \ldots + x_k = M$$

- k composition of M: $x_i \geq 1$

$$\binom{M-1}{k-1}$$

- weak k composition of M: $x_i \geq 0$

$$\binom{M+k-1}{k-1}$$

Removing the Restriction of K

Lemma

Let $\epsilon > 0$ 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 $(1 + \epsilon)$.

Removing the Restriction of K

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

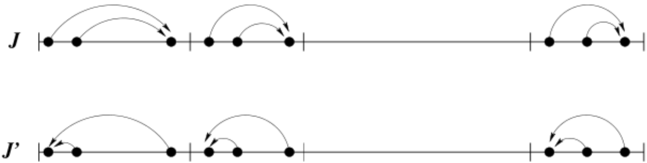
Removing the Restriction of K

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.

Then we can find an optimal packing for J, this will also be a valid packing for the original item size.

We will show that

$$\mathrm{OPT}(J) \leq (1+\epsilon)\mathrm{OPT}(I)$$



Proof

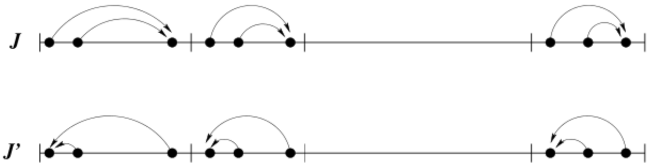
Let us construct another instance, say J' , by rounding down the size of each item to that of the smallest item in its group.

Clearly $\mathrm{OPT}(J') \leq \mathrm{OPT}(I)$.

The crucial observation is that a packing for instance J yields a packing for all but the largest Q items of instance J. Therefore,

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

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



Now we present the APTAS algorithm for Bin-Packing.

- Let I denote the given instance, and I' denote the instance obtained by discarding items of size $< \epsilon$ from I.
- By previous lemma, we can find a packing for I' using at most $(1+\epsilon)\mathrm{OPT}(I')$ bins.
- Next, we start packing the small items (of size $< \epsilon$) 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.

Analysis

If no additional bins are needed, then we have a packing in $(1+\epsilon)\mathrm{OPT}(I') \leq (1+\epsilon)\mathrm{OPT}(I)$ 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 $1-\epsilon$.

Therefore, the sum of the item sizes in I is at least $(M-1)(1-\epsilon)$. Since this is a lower bound on OPT, we get

Analysis

$$M \leq \frac{\mathrm{OPT}}{(1-\epsilon)} + 1 \leq (1+2\epsilon)\mathrm{OPT} + 1$$

where we have used the assumption that $\epsilon \leq 1/2$.

Hence, for each value of ϵ, $0 < \epsilon \leq 1/2$, we have a polynomial time algorithm achieving a guarantee of $(1+2\epsilon)\mathrm{OPT}+1$.

Summary of Algorithm

Algorithm A_ϵ is summarized below.

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