Bin Packing Problems

BIN-PACKING PROBLEM
There are n items given of sizes a_1, ..., a_n. How many bins of size 1 do we need to pack these n items?

Clearly, we can assume that $a_i \leq 1$ for all $i = 1, ..., n$ since otherwise the problem is not feasible. We get this problem from the cutting-stock problem when $L = 1$, $n = m$, $b_i = 1$ and $a_i = \ell_i$ for all $i = 1, ..., m$.

Theorem 1 *There is no ρ-factor approximation algorithm for the bin-packing problem for $\rho < \frac{3}{2}$, unless $P = NP$.*

Proof. Given integers c_i, $i = 1, ..., n$, let $C = \sum_{i=1}^{n} c_i$ and let $a_i = \frac{2c_i}{C}$ for $i = 1, ..., n$. Then, these n items can be packed into 2 bins if and only if there is a subset $S \subseteq \{1, ..., n\}$ such that $\sum_{j \in S} c_j = \sum_{j \notin S} c_j$, which is a known NP-complete decision problem (called partition). $\square$

There are several simple approximation algorithms in the literature for bin-packing.

NEXT-FIT

Input: Parameters, n and a_i, $i = 1, ..., n$.

Initialize: Set $k = 1$ and $B_k = 0$.

Main Loop: **For** $i = 1, ..., n$ **do**

> **If** $B_k + a_i > 1$ **then** set $k = k + 1$ and $B_k = a_i$
>
> **otherwise** set $B_k = B_k + a_i$.

Output: k.

Theorem 2 *Given an instance $I = \{a_1, a_2, ..., a_n\}$ of bin-packing, let us denote by $NF(I)$ the number of bins used by* Next-Fit*. Then we have*

$$NF(I) \leq 2\left\lceil \sum_{i=1}^{n} a_i \right\rceil - 1 \leq 2OPT(I) - 1,$$

where $OPT(I)$ denotes the minimum number of bins needed to pack I.

Proof. The algorithm stops in $O(n)$ steps, after executing **Main Loop**, and each step can be implemented in $O(1)$ time. Moreover, to run Next-Fit one does not need to know in advance the a_i values, they can be read one-by-one in the course of the algorithm (so called *on-line algorithm*).

To see the correctness, let us denote by i_j the index of the last item packed into the j-th bin by Next-Fit, for $j = 1, ..., k = NF(I)$, and let $i_0 = 0$. (That is, items $1, ..., i_1$ are packed into bin #1, ..., and $i_k = n$.) Then we must have [WHY?]

$$\sum_{i=i_{2j-2}+1}^{i_{2j}} a_i > 1$$

for $j = 1, ..., \lfloor \frac{k}{2} \rfloor$. Summing up these inequalities, we get

$$\frac{k-1}{2} \leq \left\lfloor \frac{k}{2} \right\rfloor \leq \left\lceil \sum_{i=1}^{n} a_i \right\rceil - 1 \leq OPT(I) - 1.$$

□

Theorem 3 *If for an instance I we have a constant $\epsilon > 0$ such that $a_i \leq \epsilon$ for $i = 1, ..., n$, then*

$$NF(I) \leq \left\lceil \frac{\sum_{i=1}^{n} a_i}{1-\epsilon} \right\rceil \leq \left\lceil \frac{OPT(I)}{1-\epsilon} \right\rceil.$$

Proof. We have for $j = 1, ..., NF(I) - 1$ that

$$\sum_{i=i_{j-1}+1}^{i_j} a_i > 1 - \epsilon$$

and thus

$$OPT(I) \geq \sum_{i=1}^{n} a_i > \sum_{j=1}^{NF(I)-1} \sum_{i=i_{j-1}+1}^{i_j} a_i > (NF(I)-1)(1-\epsilon).$$

$\square$

An improved version utilizes a larger buffer area, in which several bins can be kept open at a time.

First-Fit
Input: Parameters, n and a_i, $i = 1, ..., n$.
Initialize: Set $B_1 = B_2 = ... = 0$.
Main Loop: **For** $i = 1, ..., n$ **do**
Let h be the smallest index for which $B_h + a_i \leq 1$.
Set set $B_h = B_h + a_i$.
Output: $k = \min\{h \mid B_h > 0\}$.

Theorem 4 (Garey, Graham, Johnson and Yao (1976)) First-Fit *runs in $O(n^2)$ time and for every instance I of bin-packing provides a solution of value $FF(I)$ for which*

$$FF(I) \leq \left\lceil \frac{17}{10} OPT(I) \right\rceil.$$

A further improvement can be achieved by giving up the on-line nature of the procedure.

First-Fit Decreasing
Input: Parameters, n and a_i, $i = 1, ..., n$.
Initialize: Sort the items such that $a_1 \geq a_2 \geq \cdots \geq a_n$.
Main Loop: Apply First-Fit for this sorted instance.

Theorem 5 (Yue (1990)) First-Fit Decreasing *runs in* $O(n^2)$ *time and for every instance* I *of bin-packing provides a solution of value* $FFD(I)$ *for which*

$$FFD(I) \leq \frac{11}{9} OPT(I) + 1.$$

WATCH OUT!! $\frac{11}{9} < \frac{3}{2}$!! WHY IS THIS NOT PROVING THAT P=NP???

Let us note that First-Fit and Next-Fit are *on-line* algorithms, in other words the assignment of an item to a bin is produced at the moment of reading the item. The best on-line algorithm for bin-packing is by Yao (1980), producing asymptotically a $\frac{5}{3}$-approximation (i.e., when OPT(I) is large enough, then $YAO(I) \leq \frac{5}{3} OPT(I)$). On the other hand, Vliet (1992) proved that no on-line algorithm can provide asymptotically better than 1.54-approximation.

Solving approximately the knapsack problem in the two stage cutting stock algorithm, the result of Fernandez de la Vega and Lueker can be extended to provide a fully polynomial asymptotic approximation scheme for the bin packing problem (see Karmarkar and Karp (1982)).

An Asymptotic Approximation Scheme for Cutting-Stock

Let us return to the more general cutting-stock problem, and for the sake of simplicity, let us assume that $L = 1$, and all order lengths ℓ_j, $j = 1, ..., m$ are between 0 and 1.

Theorem 6 (Fernandez de la Vega and Lueker (1981)) *Let us consider an instance* I *of the cutting stock problem (with* $L = 1$*), and let* x *be a feasible solution to the linear programming formulation which has at most* m *nonzero components (e.g., an arbitrary basic feasible solution). Then an integer feasible solution* $\hat{x}$ *can be found on* $O(|I|)$ *time, for which*

$$\sum_{c \in \mathcal{C}} \hat{x}_c \leq \sum_{c \in \mathcal{C}} x_c + \frac{m+1}{2}.$$

Proof. Let us consider a smaller instance I' obtained from I by changing the b_i, $i = 1, ..., m$ parameters to

$$b'_i = max\{0, b_i - \sum_{c \in \mathcal{C}} \lfloor x_c \rfloor c_i\} \quad \text{for} \quad i = 1, ..., m.$$

Introducing $S = \{i \mid b'_i > 0\}$, we can obviously write

$$\sum_{i \in S} b'_i \ell_i = \sum_{i \in S} \ell_i \left(b_i - \sum_{c \in \mathcal{C}} \lfloor x_c \rfloor c_i \right)$$

and thus

$$\sum_{i=1}^{m} b'_i \ell_i = \sum_{i \in S} b'_i \ell_i \leq \sum_{c \in \mathcal{C}} (x_c - \lfloor x_c \rfloor) \left(\sum_{i \in S} c_i \ell_i \right) \leq \sum_{c \in \mathcal{C}} x_c - \sum_{c \in \mathcal{C}} \lfloor x_c \rfloor \tag{1}$$

is implied by the fact that $\sum_{i \in S} c_i \ell_i \leq \sum_{i=1}^{m} c_i \ell_i \leq 1$ for every cutting pattern $c \in \mathcal{C}$.

Let us next consider two solutions for I'. The first one, y^1 is defined by $y^1_c = \lceil x_c \rceil - \lfloor x_c \rfloor$ for $c \in \mathcal{C}$, while the second one we obtain by Next-Fit applied to the bin-packing problem I' consisting of $n = \sum_{i=1}^{m} b'_i$ items, b'_1 of length ℓ_1, b'_2 of length ℓ_2, etc.

Then, since x has at most m non-zeros, we have

$$\sum_{c \in \mathcal{C}} y^1_c \leq m,$$

while for the second solution we have by Theorem 2

$$\sum_{c \in \mathcal{C}} y^2_c \leq 2 \left\lceil \sum_{i=1}^{m} b'_i \ell_i \right\rceil - 1 \leq 2 \sum_{i=1}^{m} b'_i \ell_i + 1.$$

Thus, denoting by z the better one of these two, we get that z is a solution for I' with

$$\sum_{c \in \mathcal{C}} z_c \ \leq \ \min \left\{ \sum_{c \in \mathcal{C}} y^1_c, \sum_{c \in \mathcal{C}} y^2_c \right\} \leq \frac{\sum_{c \in \mathcal{C}} y^1_c \ + \ \sum_{c \in \mathcal{C}} y^2_c}{2} \ \leq \ \sum_{i=1}^{m} b'_i \ell_i + \frac{m+1}{2}.$$

Then, $\hat{x} = \lfloor x \rfloor + z$ is a solution for which by (1) we have

$$\sum_{c \in \mathcal{C}} \hat{x}_c \leq \sum_{c \in \mathcal{C}} \lfloor x_c \rfloor + \sum_{i=1}^{m} b'_i \ell_i + \frac{m+1}{2} \leq \sum_{c \in \mathcal{C}} x_c + \frac{m+1}{2}.$$

$\square$

Corollary 1 (Fernandez de la Vega and Lueker (1981)) *Let m and $\epsilon > 0$ be fixed constants, and consider an instance I of cutting-stock with m different order lengths, none of which is shorter than ϵ (we still assume $L = 1$). Then, we can find a solution $\hat{x}$ in $O(|I|)$ time for which*

$$\sum_{c \in \mathcal{C}} \hat{x}_c \leq OPT(I) + \frac{m+1}{2}.$$

Proof. Let us note that under the conditions stated, we have $|\mathcal{C}| \leq (m+1)^{1/\epsilon}$ and thus both the number of constraints (m) and the number of variables ($|\mathcal{C}|$) in the linear programming problem is a constant. Thus, an optimal integral solution x^* can be found in constant time (e.g., with the simplex algorithm). Since we have $\sum_{c \in \mathcal{C}} x^*_c \leq OPT(I)$ the claim follows by Theorem 6. $\square$

From this, by using a similar idea we considered for binary knapsack problems, namely by grouping the items which have "similar" lengths, they obtained the following asymptotic approximation result:

Theorem 7 (Fernandez de la Vega and Lueker (1981)) *For every $\frac{1}{2} \geq \epsilon > 0$ it is possible to obtain in $O(n\frac{1}{\epsilon^2})$ time (plus the time needed to solve the corresponding linear programming problem) a feasible integral solution $\widehat{x}$ for which*

$$\sum_{c \in \mathcal{C}} \widehat{x}_c \leq (1+\epsilon) Z_{IP} + \frac{1}{\epsilon^2}.$$

$\square$

Karmarkar and Karp (1982) improved on the above by solving the linear program "only approximately" by using a variant of the ellipsoid algorithm, and obtained a fully polynomial asymptotic approximation scheme.