INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

An Approximation Scheme for a One-Dimensional Bin Packing Problem with Shelf Divisions

E. C. Xavier *F. K. Miyazawa*

Technical Report - IC-04-008 - Relatório Técnico

August - 2004 - Agosto

An Approximation Scheme for a One-Dimensional Bin Packing Problem with Shelf Divisions *

E. C. Xavier F. K. Miyazawa *

Abstract

Given bins of size B, non-negative values d and Δ, and a list L of items, each item $e \in L$ with size s_e and class c_e, we define a shelf as a subset of items packed inside a bin with total items size at most Δ such that all items in this shelf have the same class. Two subsequent shelves must be separated by a shelf divisor of size d. The size of a shelf is the total size of its items plus the size of the shelf divisor. The Class Constrained Shelf Bin Packing Problem (CCSBP) consists to pack the items of L into the minimum number of bins, such that, the items are divided into shelves and the total size of the shelves in a bin is at most B. We present an asymptotic approximation scheme for the CCSBP problem where the number of different classes is bounded by a constant C. To our knowledge, this is the first approximation result where shelves of non-null size are used in packing problems.

Key Words: Approximation algorithms, bin packing, shelf packing.

1 Introduction

In this paper we present an approximation scheme for a 1-D bin packing problem where items are divided by shelves. We first define this problem formally.

An instance $I = (L, s, c, d, \Delta, B)$ for the CCSBP problem is a list of items L, where s_e for each item $e \in S$, is the size of the item, c_e is the class of the item, d is the size of the shelf divisor, Δ is the maximum size of a shelf and B is the size of the bins. We denote by $n = |I| = |L|$ the number of items. Without loss of generality we consider $s_e \leq \Delta$ for each $e \in L$. Given a set of items $L' \subseteq L$ we denote by $s(L') = \sum_{e \in L'} s_e$. We denote by $[n]$ the set $1, \ldots, n$.

A shelf packing $\mathcal{P}$ of an instance I for the CCSBP problem is a set of bins $\mathcal{P} = \{P_1, \ldots, P_k\}$, where the items packed in a bin $P_i \in \mathcal{P}$ are partitioned into shelves $\{N_1^i, \ldots, N_{q_i}^i\}$ such that for each shelf N_j^i we have that $s(N_j^i) \leq \Delta$, all items in N_j^i are of the same class and $\sum_{j=1}^{q_i}(s(N_j^i) + d) \leq B$. We denote by $|\mathcal{P}|$ the number of bins used in this packing and $ns(P_i)$ the number of shelves used in a bin P_i of $\mathcal{P}$.

The Class Constrained Shelf Bin Packing Problem (CCSBP) consists to find a shelf packing of the items of L into the minimum number of bins. This problem is NP-hard

*University of Campinas, 13081-970 Campinas, SP. Research supported in part by CNPq — Conselho Nacional de Desenvolvimento Científico e Tecnológico, MCT/CNPq under PRONEX program (Proc. 664107/97-4) and CNPq (Proc. 470608/01–3, 464114/00–4, 478818/03–3, 300301/98–7).

since it is a generalization of the bin packing problem. We note that the term shelf is used under another context in the literature for the 2-D Strip packing problem. In this case, packings are two staged packings divided into levels.

There are many practical applications for the CCSBP problem even when there is only one class of items. For example, when the items to be packed must be separated by non-null shelf divisors (inside a bin) and each shelf have a limited capacity. The CCSBP problem is also adequate when some items cannot be stored in a same shelf (like foods and chemical products). In most of the cases, the sizes of the shelf divisions have non-negligible width. Although these problems are very common in practice, to our knowledge this paper is the first to present approximation results for them.

An interesting application for the CCSBP problem was introduced by Ferreira et al. [3] in the iron and steel industry, where a raw material roll must be cut into final rolls grouped by certain properties after two cutting phases.

Given an algorithm A_ε, for some $\varepsilon > 0$, and an instance I for some problem P we denote by $A_\varepsilon(I)$ the value of the solution returned by algorithm A_ε when executed on instance I and by $\mathrm{OPT}(I)$ the value of an optimal solution for this instance. We say that A_ε, for $\varepsilon > 0$, is an asymptotic polynomial time approximation scheme (APTAS) for the problem CCSBP if there exists constants t and K such that for any instance I we have $A_\varepsilon(I) \leq (1 + t\varepsilon)\mathrm{OPT}(I) + K$.

Results: In this paper we present an asymptotic approximation scheme for the problem CCSBP when the number of different classes is bounded by a constant C. To our knowledge, this is the first approximation result where shelves of non-null width are used in packing problems.

Related Work: The bin packing is a well studied problem. Fernandez de la Vega and Lueker [2] presented an asymptotic PTAS for the bin packing problem. In [1] Dawande et al. presented approximation schemes for a class constrained version of the bin packing where bins can have different sizes and each bin can pack items of at most K different classes. In [4], Schachnai and Tamir presented a dual polynomial time approximation scheme for the class-constrained bin-packing (CCBP) problem. In the CCBP problem, the set of items must be packed into the minimum number of bins with capacity 1, each bin containing items of at most K classes. All these previous works also consider that the number of different classes is bounded by a constant.

2 An APTAS for the Class Constrained Shelf Bin Packing Problem

In this section we introduce an asymptotic approximation scheme for the CCSBP problem when the number of different classes is bounded by a constant C.

Without loss of generality, we assume that all bins have capacity 1 and each class belongs to the set $[C]$. The algorithm is presented in Figure 1 and is denoted by $\mathrm{ASBP}_\varepsilon$. It consider two cases: When $\varepsilon \geq d + \Delta$, for which it uses an algorithm denoted by $\mathrm{ASBP}'_\varepsilon$ and the opposite case, when an algorithm denoted by $\mathrm{ASBP}''_\varepsilon$ is used. Note that the algorithm $\mathrm{ASBP}''_\varepsilon$ rescale the instance so that its shelf capacity is 1.

Algorithm $\text{ASBP}_\varepsilon(L, s, c, d, \Delta, B)$
Input: List of items L, each item $e \in L$ with size s_e and class c_e, maximum capacity of a shelf Δ, shelf divisors of size d, bins of capacity $B = 1$.
Output: Shelf packing $\mathcal{P}$ of L.
Subroutines: Algorithms ASBP'_ε and $\text{ASBP}''_\varepsilon$.

```
If ε ≥ d + Δ then
    P ← ASBP'_ε(L, s, c, d, Δ, B)
else
    Scale the sizes d, Δ, B and s_e, for each e ∈ L, proportionally so that Δ = 1.
    // The condition to enter in this case is now equivalent to ε ≤ (d + Δ)/B.
    P ← ASBP''_ε(L, s, c, d, Δ, B).
Return P.
```

Figure 1: Algorithm ASBP_ε.

3 The Algorithm ASBP'_ε

We first present the algorithm ASBP'_ε and show that it is an APTAS for its corresponding case. In the next section, we show the same result for the algorithm $\text{ASBP}''_\varepsilon$. The algorithm ASBP'_ε uses two subroutines. One subroutine is an APTAS for the one dimensional bin packing problem presented by Fernandez de la Vega and Lueker [2]. Vazirani [5] presented a version of this algorithm, which we denote by FL_ε where the following theorem is valid.

Theorem 3.1 *For any $\varepsilon > 0$, there exists a polynomial time algorithm FL_ε to pack a list of items L, each item $e \in L$ with size $s_e \in [0, \Delta]$, into bins of capacity Δ such that $\text{FL}_\varepsilon(L) \leq (1+\varepsilon)\,\text{OPT}_\Delta(L)+1$, where $\text{OPT}_\Delta(L)$ is the minimum number of bins of capacity Δ to pack L.*

Another algorithm for the one-dimensional bin packing problem used as subroutine is the algorithm First-Fit (FF). This algorithm can be described as follows: Given a list of items L to be packed into bins, it packs the items in the order given by L. It tries to pack each new item into one of the existing bins, considering the order they were generated. If it is not possible to pack an item in any of the existing bins, the algorithm packs it into a new bin.

The algorithm ASBP'_ε is presented in Figure 2 and consists in: Given an instance I, the algorithm ASBP'_ε first packs all items of the instance into bins of size Δ using the algorithm FL_ε. The algorithm ASBP'_ε considers each one of these bins of size Δ as a shelf, where the size of a shelf is its total items size plus the size d of a shelf divisor. The algorithm ASBP'_ε packs these shelves into bins of size 1 using the algorithm FF.

The following lemma is valid for the algorithm ASBP'_ε.

Lemma 3.2 *The algorithm ASBP'_ε, is an APTAS for the* CCSBP *problem when the given instance I is such that $B = 1$ and $\varepsilon \geq d + \Delta$.*

Algorithm $\text{ASBP}'_\varepsilon(L, s, c, \Delta, d, B)$

Input: List of items L, each item $e \in L$ with size s_e and class c_e, maximum capacity of a shelf Δ, shelf divisors of size d, bins of capacity $B = 1$ and $\varepsilon \geq d + \Delta$.

Output: Shelf packing $\mathcal{P}$ of L.

Subroutines: Algorithms FL_ε and FF.

1. Let L_c be the items of class c in L.

2. For each class $c \in [C]$ let $\mathcal{P}^c_\Delta$ be the packing of L_c obtained by the algorithm FL_ε using bins of capacity Δ.

3. Let $\mathcal{P}_\Delta$ be the union of the packings $\mathcal{P}^c_\Delta$, for each $c \in [C]$.

4. Consider each bin $D \in \mathcal{P}_\Delta$ as a shelf with size $\sum_{e \in D} s_e + d$.

5. Let S be the set of shelves obtained from $\mathcal{P}_\Delta$.

6. Let $\mathcal{P}$ be the packing of the shelves in S into unit bins using the algorithm FF.

7. Return $\mathcal{P}$.

Figure 2: Algorithm ASBP'_ε where $\varepsilon \geq d + \Delta$.

Proof. In step 2, the algorithm obtains a packing $\mathcal{P}^c_\Delta$ of items of class c in L (items in L_c) into bins of capacity Δ using the algorithm FL_ε. By Theorem 3.1, we have

$$|\mathcal{P}^c_\Delta| \leq (1 + \varepsilon)\text{OPT}_\Delta(L_c) + 1. \tag{1}$$

The algorithm then considers each bin in $\mathcal{P}_\Delta$ as a shelf and obtains a shelf packing $\mathcal{P}$ using the algorithm FF to pack these shelves into unit bins. Since $\varepsilon \geq d + \Delta$, all bins of $\mathcal{P}$, except perhaps the last, must be filled by at least $1 - \varepsilon$. So,

$$\begin{aligned}
(\text{ASBP}'_\varepsilon(L) - 1)(1 - \varepsilon) &\leq s(L) + |\mathcal{P}_\Delta| d \\
&\leq s(L) + ((1 + \varepsilon)\sum_{c=1}^{C} \text{OPT}_\Delta(L_c) + C)d \\
&= s(L) + d\sum_{c=1}^{C} \text{OPT}_\Delta(L_c) + d\varepsilon \sum_{c=1}^{C} \text{OPT}_\Delta(L_c) + dC + dC\varepsilon \quad (2) \\
&\leq \text{OPT}(L) + \varepsilon\text{OPT}(L) + 2C \\
&= (1 + \varepsilon)\text{OPT}(L) + 2C,
\end{aligned}$$

where inequality (2) is valid since $\sum_{c=1}^{C} \text{OPT}_\Delta(L_c) \leq ns(\text{OPT}(L))$ and then

$$s(L) + d\sum_{c=1}^{C} \text{OPT}_\Delta(L_c) \leq \text{OPT}(L).$$

Also notice that $d < 1$. Therefore, for any $0 < \varepsilon < 1/3$ we have

$$\begin{aligned}
\text{ASBP}'_\varepsilon(L) &\leq \frac{1 + \varepsilon}{1 - \varepsilon}\text{OPT}(L) + \frac{2C}{1 - \varepsilon} + 1 \\
&\leq (1 + 3\varepsilon)\text{OPT}(L) + 4C + 1.
\end{aligned}$$

□

4 The Algorithm $\mathrm{ASBP}''_\varepsilon$

Now, suppose the algorithm $\mathrm{ASBP}_\varepsilon$ obtains a shelf packing with the algorithm $\mathrm{ASBP}''_\varepsilon$. Throughout this section, we consider that s_e, d, Δ and B is the rescaled instance, with $\Delta = 1$. Note that, the equivalent condition to enter in this case is

$$\varepsilon < \frac{d+\Delta}{B} = \frac{d+1}{B} \leq 1. \tag{3}$$

Notice that the maximum number of shelves completely filled packed in a bin is at most $\left\lceil \frac{B}{d+\Delta} \right\rceil$ which from (3) is at most $\frac{1}{\varepsilon}+1$. Observe that if there is any bin with more shelves than $\frac{2}{\varepsilon}+2$ of a same class, it has at least two shelves of this class with total size at most Δ. In this case, these two shelves can be combined into only one shelf. Without loss of generality we consider that each bin, in a solution for the CCSBP problem, contains at most $\frac{2}{\varepsilon}+2$ shelves of a same class.

We first describe the subroutines used by the algorithm $\mathrm{ASBP}''_\varepsilon$.

Algorithm $\mathrm{A_{ALL}}$**:** This is an algorithm used as subroutine by $\mathrm{ASBP}''_\varepsilon$ to generate all possible packings with at most $\frac{2}{\varepsilon}+2$ shelves of a same class, when the size of each item is bounded below by a constant and the number of distinct sizes is also bounded by a constant. The algorithm may generate empty shelves (used latter to pack small items). The following lemma guarantee the existence of such an algorithm.

Lemma 4.1 *Given an instance $I = (L, s, c, d, \Delta, B)$, with $\Delta = 1$, where the number of distinct items sizes is a constant t, the number of different classes is bounded by a constant C and each item $e \in L$ has size $s_e \geq \varepsilon^2$ then there exists a polynomial time algorithm that generates all possible shelf packings of L with at most $\frac{2}{\varepsilon}+2$ shelves of a same class in each bin.*

Proof. The number of items in a shelf is bounded by $p = 1/\varepsilon^2$. Given a class, the number of different shelves for it is bounded by $r' = \binom{p+t+1}{p}$ and so, the number of different shelves is bounded by $r = Cr'$. Since the number of shelves in a bin is bounded by $q = C(\frac{2}{\varepsilon}+2)$, the number of different bins with is bounded by $u = \binom{q+r}{q}$. All the values p, q, r and u depends only on ε and C.

Therefore, the number of all feasible packings is bounded by $\binom{n+u}{n}$, which is polynomial in n. □

Algorithm $\mathrm{A_{LR}}$**:** Another subroutine used by $\mathrm{ASBP}''_\varepsilon$ is an algorithm that we denote by $\mathrm{A_{LR}}$. This algorithm uses the linear rounding technique, presented by Fernandez de la Vega and Lueker [2], and consider only items with size at least ε^2. The algorithm $\mathrm{A_{LR}}$ returns a pair $(\mathcal{P}_1, \mathbb{P})$, where $\mathcal{P}_1$ is a packing for a list of very big items and $\mathbb{P}$ is a set of packings for the remaining items.

For the use of the linear rounding technique, we use the following notation: Given two lists of items X and Y, let $X_1, \ldots, X_C$ and $Y_1, \ldots, Y_C$ be the partition of X and Y respectively in classes, where X_c and Y_c have only items of class c for each $c \in [C]$. We write $X \preceq Y$ if there is an injection $f_c : X_c \to Y_c$ for each $c \in [C]$ such that $s(e) \leq s(f(e))$ for all $e \in X_c$.

Algorithm $\mathrm{A_R}$**:** Given two lists X and Y such that $X \preceq Y$ then there exists an algorithm, denoted by $\mathrm{A_R}$ (Replace), that given a packing $\mathcal{P}_Y$, it obtains a packing $\mathcal{P}_X$ for X such that $|\mathcal{P}_X| = |\mathcal{P}_Y|$ as the next lemma guarantees. This algorithm is used by the algorithm $\mathrm{A_{LR}}$.

Lemma 4.2 *If X and Y are two lists with $X \preceq Y$, then* $\mathrm{OPT}(X) \leq \mathrm{OPT}(Y)$. *Moreover, if $\mathcal{P}_Y$ is a shelf packing of Y then there exists a polynomial time algorithm* $\mathrm{A_R}$ *that given $\mathcal{P}_Y$ obtains a shelf packing $\mathcal{P}_X$ of X such that $|\mathcal{P}_X| = |\mathcal{P}_Y|$.*

Proof. The algorithm $\mathrm{A_R}$ sorts the lists X_c and Y_c for each $c \in [C]$ in non-increasing order of items size and then replace in this order, each item of Y_c in the packing $\mathcal{P}_Y$ by an item of X_c. The possible remaining items of Y_c are removed. □

For any instance X, denote by $\overline{X}$ the instance with precisely $|X|$ items with size equal to the size of the smallest item in X. Clearly, $\overline{X} \preceq X$.

Algorithm SFF**:** The packing $\mathcal{P}_1$ is obtained by an algorithm denoted by SFF (Shelf First Fit). This algorithm is an adaptation of the algorithm FF for the problem CCSBP. It packs the items in the order of the input list L. The algorithm SFF packs the next item $e \in L$ in the first shelf of class c_e (in the order they where created) of a bin that have sufficient space to accommodate it. If there is no such shelf, the algorithm packs the item in a new shelf. This new shelf is packed into the first possible bin. If necessary, it packs the shelf into a new bin.

The algorithm $\mathrm{A_{LR}}$ is presented in Figure 3. It consists in the following: Let $G_1, \ldots, G_C$ the partition of the input list G into classes $1, \ldots, C$ and let $n_c = |G_c|$ for each class c. The algorithm $\mathrm{A_{LR}}$ partition each list G_c into groups $G_c^1, G_c^2, \ldots, G_c^{k_c}$. Let $G^1 = \cup_{c=1}^{C} G_c^1$. The algorithm generates a packing $\mathcal{P}_1$ of G^1 using $O(\varepsilon)\mathrm{OPT}(L) + 1$ bins and a set $\mathbb{P}$ with polynomial number of packings for the items in $G \setminus G^1$. The packing $\mathcal{P}_1$ is generated by the algorithm SFF and the set of packings $\mathbb{P}$ is generated using the algorithms $\mathrm{A_{ALL}}$ and $\mathrm{A_R}$.

The following lemma is valid for the packing $\mathcal{P}_1$.

Lemma 4.3 *The packing $\mathcal{P}_1$ for the items in G^1 is such that $|\mathcal{P}_1| \leq 4\varepsilon\,\mathrm{OPT}(L) + 1$.*

Proof. First, consider the total items size packed in a bin T of some shelf packing. If s_T is the total items size of T then s_T is at most $\lceil B/(d+\Delta) \rceil \Delta$. Therefore, any optimum solution must satisfy

$$\mathrm{OPT}(L) \geq \frac{s(L)}{\lceil B/(d+\Delta) \rceil \Delta} = \frac{s(L)}{\lceil B/(d+1) \rceil} \geq \frac{1}{2}\,\frac{s(L)}{B/(d+1)}. \tag{4}$$

Notice that $\sum_{c=1}^{C} n_c = n$. The algorithm SFF packs at least $\lfloor B/(d+\Delta) \rfloor$ shelves in each bin, each shelf with at least one item. This means that each bin has at least $\lfloor B/(d+\Delta) \rfloor$ items, except perhaps the last, each item with size at least ε^2 and at most 1. Since the

ALGORITHM $A_{LR}(G)$

Input: List G with n items, each item $e \in G$ with size $s_e > \varepsilon^2$; maximum capacity of a shelf $\Delta = 1$; shelf divisors of size d and bins of capacity B.

Output: A pair $(\mathcal{P}_1, \mathbb{P})$, where $\mathcal{P}_1$ is a packing and $\mathbb{P}$ is a set of packings, where $\mathcal{P}_1 \cup \mathcal{P}'$ is a packing of G for each $\mathcal{P}' \in \mathbb{P}$.

Subroutines: Algorithms A_{ALL}, SFF and A_R.

1. Partition G into lists G_c for each class $c = 1, \ldots, C$.

1. Partition each set G_c into $q_c = \lfloor n_c \varepsilon^3 \rfloor$ groups $G_c^1, G_c^2, \ldots, G_c^{k_c}$, such that

$$G_c^1 \succeq G_c^2 \succeq \cdots \succeq G_c^{k_c},$$
$$\text{where } |G_c^j| = q_c \text{ for all } j = 1, \ldots, k_c - 1,$$
$$\text{and } |G_c^{k_c}| \le q_c.$$

2. Let $G^1 = \cup_{c=1}^{C} G_c^1$.

3. Let $\mathcal{P}_1$ be a packing of G^1 obtained by the algorithm SFF.

4. Let $\mathbb{Q}$ be the set of all possible packings obtained with the algorithm A_{ALL} over the list $(\overline{G_1^1} \| \ldots \| \overline{G_1^{k_1-1}} \| \ldots \| \overline{G_C^1} \| \ldots \| \overline{G_C^{k_C-1}})$.

5. Let $\mathbb{P}$ be the set of packings obtained with the algorithm A_R over each pair $(\mathcal{Q}, G_1^2 \| \ldots \| G_1^{k_1} \| \ldots \| G_C^2 \| \ldots \| G_C^{k_C})$, where $\mathcal{Q} \in \mathbb{Q}$.

6. Return $(\mathcal{P}^1, \mathbb{P})$.

Figure 3: Algorithm to obtain packings for items with size at least ε^2.

group G_1 has at most $n\varepsilon^3$ items, the number of bins in the shelf packing $\mathcal{P}_1$ can be bounded as follows.

$$\begin{aligned} |\mathcal{P}_1| &\le \left\lceil \frac{n\varepsilon^3}{\lfloor B/(d+\Delta) \rfloor} \right\rceil = \left\lceil \frac{n\varepsilon^3}{\lfloor B/(d+1) \rfloor} \right\rceil \\ &\le 2\frac{n\varepsilon^3}{B/(d+1)} + 1 \le 2\frac{\varepsilon\, s(L)}{B/(d+1)} + 1 \\ &\le 4\varepsilon\, \mathrm{OPT}(L) + 1, \end{aligned} \tag{5}$$

where the inequality (5) is valid from (4). ∎

Algorithm SMALL**:** The algorithm $\mathrm{ASBP}''_\varepsilon$ also uses another subroutine denoted by SMALL to pack small items (size less than ε^2) into a given packing. Let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be a shelf packing of a list of items L and suppose we have to pack a set S of small items, with size at most ε^2, into $\mathcal{P}$. The packing of the small items is obtained from a solution of a linear program. Let $N_1^{ic}, \ldots, N_{n_{ic}}^{ic}$ be the shelves of class c in the bin P_i of the packing $\mathcal{P}$. For each shelf N_j^{ic}, define a non-negative variable x_j^{ic}. The variable x_j^{ic} indicates the total size of small items of class c that is to be packed in the shelf N_j^{ic}. Consider the following linear program denoted by LPS:

$$
\begin{aligned}
& \max \sum_{i=1}^{k}\sum_{c=1}^{C}\sum_{j=1}^{n_{ic}} x_j^{ic} \\
s(N_j^{ic}) + x_j^{ic} & \leq \Delta && \forall\ i \in [k],\ c \in [C],\ j \in [n_{ic}], && (1) \\
\sum_{c=1}^{C}\sum_{j=1}^{n_{ic}} (s(N_j^{ic}) + x_j^{ic} + d) & \leq B && \forall\ i \in [k], && (2) \quad \text{(LPS)} \\
\sum_{i=1}^{t}\sum_{j=1}^{n_{ic}} x_j^{ic} & \leq s(S_c) && \forall\ c \in [C], && (3) \\
x_j^{ic} & \geq 0 && \forall\ i \in [k],\ c \in [C],\ j \in [n_{ic}] && (4)
\end{aligned}
$$

where S_c is the set of small items of class c in S.

The constraint (1) guarantees that the amount of space used in each shelf is at most Δ and constraint (2) guarantees that the amount of space used in each bin is at most B. The constraint (3) guarantees that variables x_j^{ic} is not greater than the total size of small items.

Given a packing $\mathcal{P}$, and a set S of small items, the algorithm SMALL first solves the linear program LPS, and then packs small items in the following way: For each variable x_j^{ic} it packs, while possible, the small items of class c into the shelf N_j^{ic}, so that the total size of the packed small items is at most x_j^{ic}. The possible remaining small items are grouped by classes and packed using the algorithm SFF into new bins.

The following lemma is valid for the algorithm SMALL.

Lemma 4.4 *Let $\mathcal{P}$ be a shelf packing of a list of items L, where each bin of $\mathcal{P}$ have at most $\frac{2}{\varepsilon} + 2$ shelves of a same class, G the set of items in L with size at least ε^2 and S the set $L \setminus G$. Let G' be a list of items with $G' \preceq G$ and $\hat{\mathcal{P}}$ a packing of the items $G' \cup S$ obtained from $\mathcal{P}$ as follows:*

1. *Let $\mathcal{P}_1$ the packing obtained from $\mathcal{P}$ removing the items of S.*
2. *Let $\mathcal{P}_2$ the packing of G' using the algorithm $\mathrm{A_R}$ over the pair $(\mathcal{P}_1, G')$.*
3. *Let $\hat{\mathcal{P}}$ the packing obtained applying the algorithm* SMALL *over the pair $(\mathcal{P}_2, S)$.*

Then, we have $|\hat{\mathcal{P}}| \leq (1 + 8C\varepsilon)|\mathcal{P}| + C + 1$.

Proof. Notice that $|\mathcal{P}_2| = |\mathcal{P}|$ and for each shelf N_j in a bin of $\mathcal{P}$, its corresponding shelf N'_j in $\mathcal{P}_2$ is such that $s(N'_j) \leq s(N_j)$. If $|\hat{\mathcal{P}}| = |\mathcal{P}|$ then the lemma follows. So suppose the algorithm SMALL uses additional bins to pack the items of S.

Given a bin E, denote by $ns(E)$ the number of shelves in E, $ns_c(E)$ the number of shelves of class c in E, $ss(E)$ the total size of small items in E and $ss_c(E)$ the total size of small items of class c in E.

Consider the linear program LPS. An optimum solution for LPS leads to an optimal fractional packing $\mathcal{P}^*$ of the small items such that $|\mathcal{P}^*| = |\mathcal{P}|$. Consider a bin P_i^* of $\mathcal{P}^*$ and $\hat{P}_i$ the corresponding bin in $\hat{\mathcal{P}}$. We first prove that the following inequality is valid,

$$ss(P_i^*) - ss(\hat{P}_i) \leq 4C\varepsilon. \qquad (6)$$

To prove (6), notice that $ns_c(P_i^*) = ns_c(\hat{P}_i)$ for each class c. Given a shelf N_j^{ic} and the corresponding variable x_j^{ic}, the algorithm SMALL packs a set of items T_j^{ic} in N_j^{ic} such that $x_j^{ic} - s(T_j^{ic}) \leq \varepsilon^2$ since a small item has size at most ε^2. Since each bin in $\mathcal{P}^*$ has at most $\frac{2}{\varepsilon} + 2$ shelves of a same class we have for each class $c \in [C]$

$$\begin{aligned} ss_c(\hat{P}_i) &\geq \sum_{j=1}^{n_{ic}} (x_j^{ic} - \varepsilon^2) = ss_c(P_i^*) - ns_c\varepsilon^2 \\ &\geq ss_c(P_i^*) - (\frac{2}{\varepsilon} + 2)\varepsilon^2 \geq ss_c(P_i^*) - 4\varepsilon. \end{aligned}$$

Since the above inequalities are valid for each class we can conclude the proof of (6). From (6), we know that the total size of small items packed in additional bins by SMALL with the algorithm SFF, is at most $4C\varepsilon|\mathcal{P}^*| = 4C\varepsilon|\mathcal{P}|$. Denote by $\hat{\mathcal{Q}}$ the set of additional bins. The number of shelves with total items size at least $\Delta - \varepsilon^2$ in $\hat{\mathcal{Q}}$ is at most $\left\lceil \frac{4C\varepsilon|\mathcal{P}|}{1-\varepsilon^2} \right\rceil \leq 8C\varepsilon|\mathcal{P}| + 1$. Since each additional bin has at least one shelf with items size at least $\Delta - \varepsilon^2$, except perhaps in C bins, the number of bins in $\hat{\mathcal{Q}}$ is at most $8C\varepsilon|\mathcal{P}| + C + 1$. Therefore, the number of bins in $\hat{\mathcal{P}}$ is at most $(1 + 8C\varepsilon)|\mathcal{P}| + C + 1$. ◻

In Figure 4 we present the algorithm ASBP$''_\varepsilon$. The algorithm first obtain a pair $(\mathcal{P}_1, \mathbb{P})$ where $\mathcal{P}_1 \cup \mathcal{P}'$ is a packing of the items with size at least ε^2, for any $\mathcal{P}' \in \mathbb{P}$. For each packing $\mathcal{P}_1 \cup \mathcal{P}'$, $\mathcal{P}' \in \mathbb{P}$, the algorithm ASBP$''_\varepsilon$ uses the algorithm SMALL to pack the items with size less than ε^2 into the packing $\mathcal{P}'$. The algorithm returns a packing with the smallest number of bins.

ALGORITHM ASBP$''_\varepsilon(L, s, c, d, \Delta, B)$

Input: List of items L, each item $e \in L$ with size s_e and class c_e, maximum capacity of a shelf $\Delta = 1$, shelf divisors of size d, bins of capacity B and $\varepsilon \leq (d + \Delta)/B$.

Output: Shelf packing $\mathcal{P}$ of L.

Subroutines: Algorithms A_{LR} and SMALL.

1. Let G the items $e \in L$ with size $s_e \geq \varepsilon^2$ and S the set $L \setminus G$.
2. Let $(\mathcal{P}_1, \mathbb{P})$ be a pair obtained from the algorithm A_{LR} applied over the list G.
3. For each $\mathcal{Q} \in \mathbb{P}$ do
4. let $\hat{\mathcal{Q}}$ the packing obtained using the algorithm SMALL to pack S into $\mathcal{Q}$.
5. Let $\mathcal{P}$ be a packing $\mathcal{P}_1 \cup \hat{\mathcal{Q}}$ where $\mathcal{Q} \in \mathbb{P}$ and $|\hat{\mathcal{Q}}|$ is minimum.
6. Return $\mathcal{P}$.

Figure 4: Algorithm ASBP$''_\varepsilon$.

The following lemma concludes the analysis of the algorithm ASBP$''_\varepsilon$.

Lemma 4.5 *The algorithm* ASBP$''_\varepsilon$, *is an APTAS for the* CCSBP *problem when the given instance I is such that $\Delta = 1$, $\varepsilon \leq (d+\Delta)/B$ and the number of different classes is bounded by some constant C.*

Proof. Given an instance $I = (L, s, c, d, \Delta, B)$, with $\Delta = 1$, let G be the set of items in L with size at least ε^2 and S the set $L \setminus G$. The items in G are packed by the algorithm A_{LR}. It first partitions G in lists G_c for each class c and then it partitions each list G_c into groups $G_c^1 \succeq G_c^2 \succeq \ldots \succeq G_c^{k_c}$. From Lemma 4.3 the following inequality is valid for the list $G^1 = \cup_{c=1}^{C} G_c^1$.

$$|\mathcal{P}_1| \leq 4\varepsilon \, \mathrm{OPT}(L) + 1. \tag{7}$$

The packing of the items in $G_1^2 \| \ldots \| G_1^{k_1} \| \ldots \| G_C^2 \| \ldots \| G_C^{k_C}$ is obtained from the set of all possible packings of $\overline{G_1^1} \| \ldots \| \overline{G_1^{k_1-1}} \| \ldots \| \overline{G_C^1} \| \ldots \| \overline{G_C^{k_C-1}}$. Notice that

$$\overline{G_1^1} \| \ldots \| \overline{G_1^{k_1-1}} \| \ldots \| \overline{G_C^1} \| \ldots \| \overline{G_C^{k_C-1}} \succeq G_1^2 \| \ldots \| G_1^{k_1} \| \ldots \| G_C^2 \| \ldots \| G_C^{k_C}.$$

Let $\mathcal{O}$ be an optimum shelf packing of I, $\mathcal{O}_1$ the packing obtained from $\mathcal{O}$ without the items of S but with the possible empty shelves and $\mathcal{O}_2$ the packing of $\mathcal{O}_1$ rounding down each item size to the corresponding item in $\overline{G_1^1} \| \ldots \| \overline{G_1^{k_1-1}} \|, \ldots, \| \overline{G_C^1} \| \ldots \| \overline{G_C^{k_C-1}}$. Clearly, $\mathcal{O}_2 \in \mathbb{P}$. Let $\hat{\mathcal{O}}$ be a packing obtained from the algorithm A_R over the pair $(\mathcal{O}_2, G_1^2 \| \ldots \| G_1^{k_1} \|, \ldots, \| G_C^2 \| \ldots \| G_C^{k_C})$. If $\mathcal{Q}$ is a packing obtained applying the algorithm SMALL over the pair $(\hat{\mathcal{O}}, S)$, we have from Lemma 4.4.

$$\mathcal{Q} \leq (1 + 8C\varepsilon)|\mathcal{O}| + C + 1 = (1 + 8C\varepsilon)\mathrm{OPT}(L) + C + 1 \tag{8}$$

Since the algorithm $\mathrm{ASBP}''_\varepsilon$ obtains a packing $\mathcal{P}$ that uses at most the number of bins in $\mathcal{P}_1 \cup \mathcal{Q}$, the theorem follows from the inequalities (7) and (8). ◻

From lemmas 3.2 and 4.5, the following theorem is valid.

Theorem 4.6 *The algorithm* $\mathrm{ASBP}_\varepsilon$ *is an APTAS for the* CCSBP *problem.*

References

[1] M. Dawande, J. Kalagnanam, and J. Sethuranam. Variable sized bin packing with color constraints. *Eletronic Notes in Discrete Mathematics*, 7:4, 2001.

[2] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1 + \epsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[3] J. S. Ferreira, M. A. Neves, and P. Fonseca e Castro. A two-phase roll cutting problem. *European J. Operational Research*, 44:185–196, 1990.

[4] H. Shachnai and T. Tamir. Polynomial time approximation schemes for class-constrained packing problems. *Journal of Scheduling*, 4(6):313–338, 2001.

[5] V. Vazirani. *Approximation Algorithms.* Springer-Verlag, 2001.