INSTITUT FÜR INFORMATIK

A fast approximation scheme for the multiple knapsack problem

Klaus Jansen

Bericht Nr. 0916
February 2010

CHRISTIAN-ALBRECHTS-UNIVERSITÄT
ZU KIEL

A fast approximation scheme for the multiple knapsack problem *

Klaus Jansen
Institut für Informatik, Universität zu Kiel
Christian-Albrechts-Platz 4, 24098 Kiel, Germany.
E-mail: kj@informatik.uni-kiel.de

February 11, 2010

Abstract

In this paper we propose an improved efficient approximation scheme for the multiple knapsack problem (MKP). Given a set $\mathcal{A}$ of n items and set $\mathcal{B}$ of m bins with possibly different capacities, the goal is to find a subset $S \subseteq \mathcal{A}$ of maximum total profit that can be packed into $\mathcal{B}$ without exceeding the capacities of the bins. Kellerer gave a polynomial time approximation scheme (PTAS) for MKP with identical capacities and Chekuri and Khanna presented a PTAS for MKP with arbitrary capacities with running time $n^{O(1/\epsilon^8 \log(1/\epsilon))}$. Recently we found an efficient polynomial time approximation scheme (EPTAS) for MKP with running time $2^{O(1/\epsilon^5 \log(1/\epsilon))} poly(n)$. Here we present an improved EPTAS with running time $2^{O(1/\epsilon \log(1/\epsilon)^4)} + poly(n)$. If the modified round-up property for bin packing with different sizes is true, the running time can be improved to $2^{O(1/\epsilon \log(1/\epsilon)^2)} + poly(n)$.

1 Introduction

The knapsack problem is a fundamental problem in combinatorial optimization [11, 17, 20]. One interesting generalization is the multiple knapsack problem (MKP), in which we are given a set $\mathcal{A}$ of n items and a set $\mathcal{B}$ of m bins or knapsacks. Each item $a \in \mathcal{A}$ has a size $size(a) \in \mathbb{Q}^+$ and a profit $profit(a) \in \mathbb{Q}^+$ and each bin $b \in \mathcal{B}$ has a capacity or size $c(b) \in \mathbb{Q}^+$. The goal of MKP is to find a subset $S \subseteq \mathcal{A}$ that can be packed into B without exceeding the capacities of the bins and that has maximum total profit $profit(S) = \sum_{a \in S} profit(a)$. The maximum total profit among all feasible subsets $S \subseteq \mathcal{A}$ that can be packed into $\mathcal{B}$ is denoted by $OPT(\mathcal{A}, \mathcal{B})$. MKP has many applications in computer science, operations research, and related disciplines; see also the books by Martello and Toth [18] and by Kellerer, Pferschy, and Pisinger [15]. An interesting application arises in scheduling jobs on identical processors where some machines are non-available during fixed time periods or where some high priority jobs are preassigned to processors [5, 21].

A maximization problem X admits a polynomial-time approximation scheme (PTAS), if there is a family of algorithms $\{A_\epsilon \mid \epsilon > 0\}$ such that for any $\epsilon > 0$ and any instance I of X, A_ϵ produces a $(1-\epsilon)$ - approximate solution in time $|I|^{f(1/\epsilon)}$ for some function f. If ϵ is very small then the value

*Supported in part by EU project AEOLUS, Algorithmic principles for building efficient overlay computers, contract number 015964.

$|I|^{f(1/\epsilon)}$ can be very large. Two important restricted classes of approximation schemes were defined to reduce the running time. An efficient polynomial-time approximation scheme (EPTAS) is a PTAS with running time of the form $f(1/\epsilon)poly(|I|)$, while a fully time polynomial time approximation scheme (FPTAS) runs in time $poly(|I|, 1/\epsilon)$.

Results. The decision version of MKP is to determine whether there is a feasible packing with profit at least p; this is a generalization of the classical bin packing problem and, therefore, strongly NP-complete. In contrast to the classical knapsack problem, MKP even with two bins with the same capacity does not have an FPTAS unless P=NP [2, 4]. Kellerer [14] gave a polynomial-time approximation scheme (PTAS) for MKP with identical capacities. This result has been generalized by Chekuri and Khanna [4] who gave a PTAS for MKP with general capacities. The running time of their PTAS is $n^{O(1/\epsilon^8 \log(1/\epsilon))}$. Chekuri and Khanna [4] posed the question of whether there is a PTAS with an improved running time and conjectured that an efficient polynomial time approximation scheme (EPTAS) with running time $f(1/\epsilon)poly(n)$ for some function f might be possible. Fellows [8] considered it as a significant open problem to determine whether MKP admits a fixed parameter tractable (FPT) algorithm or it is W[1]-hard. Notice that if the standard parametrization of an optimization problem is W[1]-hard, then the optimization problem does not have an EPTAS (unless FPT=W[1]) [1, 3]. For a survey on approximation algorithms and parameterized complexity we refer to [19]. Recently we [12] found an EPTAS for MKP with running time $2^{O(1/\epsilon^5 \log(1/\epsilon))}poly(n)$ answering the open question posed by Chekuri and Khanna in the affirmative. In this paper we prove the following main result:

Theorem 1.1 *There is an efficient polynomial-time approximation scheme (EPTAS) for the multiple knapsack problem with running time* $2^{O(1/\epsilon \log(1/\epsilon)^4)} + poly(n)$.

Interestingly, if the modified round-up conjecture for the bin packing problem with different bin sizes is true (showing that the integrality gap is bounded by a constant C), similar to the modified round-up conjecture for the classical bin packing problem by Scheithauer and Terno [22], then we can reduce the above running time to $2^{O(1/\epsilon \log(1/\epsilon)^2)} + poly(n)$. Notice that the term $poly(n) \leq poly(n, 1/\epsilon)$ mainly depends on the approximate solution of a linear program relaxation of MKP [10, 12]. Furthermore, as a by-product we obtain an interesting result for the bin packing problem. If the minimum number of bins $OPT(I) \leq poly(1/\delta)$ for a constant $\delta > 0$, then there is an approximation algorithm for bin packing which computes a packing using $OPT(I) + 1$ bins and runs in time $2^{O(1/\delta \log(1/\delta)^2)} + n$.

Techniques. In contrast to the previous approaches by Chekuri and Khanna [4] and by Kellerer [14], we use a linear program relaxation for MKP. This allows us to select fractional pieces of items and to distribute them among different bin groups. We used this idea to obtain the first EPTAS for MKP, but still with a large running time $2^{O(1/\epsilon^5 \log(1/\epsilon))}poly(n)$. In contrast to our first approach, we do not round the bin sizes in our new algorithm. To reduce the running time above we propose several interesting new techniques. The first technique is a rounding method for the LP relaxation of MKP via a network flow problem with integral capacities. For each bin group B_ℓ with $M = 1/\delta \log(1/\delta)^2$ bins (in general with different bin capacities), we generate a set of rectangles for the fractional selected items and build stacks St_ℓ for the wide rectangles using some ideas by Kenyon and Remila [16] for 2-dimensional strip packing. To generate integral capacities in the flow problem, we add a few dummy rectangles of total constant height $< 1/\delta^2$ to each stack St_ℓ. After this step we round the widths of the rectangles to a constant number of values and obtain an integral value for the total height of rectangles for each rounded width. This implies also a smaller gap between the fractional strip packing values of the set of rounded and original rectangles. The narrow rectangles are divided into a polynomial number of groups according to their widths. Here we use intervals of sizes with geometrically decreasing values $\delta c^{(\ell)}_{max}, \delta c^{(\ell)}_{max}/(1+\delta), \ldots, \delta c^{(\ell)}_{max}/(1+\delta)^k$, where $c^{(\ell)}_{max}$ is the largest bin capacity in group B_ℓ. This enables us to generate integral capacities in the flow problem also for the narrow rectangles.

For the rounding phase we also generalize a classical result for bin packing by Karmarkar and Karp [13] and Shmonin [23] to bin packing with different bin sizes: we prove that the additive integrality gap between the corresponding ILP and LP formulations for instances with d different item sizes and different bin sizes is bounded by $O(\log(d)^2)$. For MKP with at least $M = \lceil 1/\delta \log(1/\delta)^2 \rceil$ bins with the same largest capacity (where $\delta = \Theta(\epsilon)$) we obtain in this way an approximation scheme that computes a solution with profit at least $(1-\epsilon)OPT(\mathcal{A}, \mathcal{B})$. The running time of the approximation scheme is $poly(n, 1/\epsilon)$. Another technique improves the running time of the EPTAS for MKP with a constant number $\gamma \in [1/\delta, poly(1/\delta)]$ of bins to $2^{O(\gamma \log(1/\epsilon)^2)}n$: In the first phase we guess and pack high profit items into $\mathcal{B}$. Then based on ideas for bin packing by Karmarkar and Karp [13] we round the sizes of the medium size items with medium profit items. Furthermore, we show that the rounded items can be packed into $\mathcal{B}$ plus one additional bin b whose size is a small fraction of the remaining space in the bins after packing the high profit items. To eliminate the additional bin, we apply a shifting strategy on $\mathcal{B}$ and b. This is possible, since $\mathcal{B}$ contains at least $1/\delta$ pieces (where $1/\delta$ is integral) that lie completely inside of the bins each with size equal to the size of the additional bin b.

For general instances of MKP we combine the new techniques with a modified LP relaxation. Here we split the set of all bins into two groups: $\mathcal{B}_2$ with a constant number $\lceil 1/\delta \log(1/\delta)^2 \rceil$ of bins with the largest capacities and $\mathcal{B}_1$ consisting of all remaining bins. Then, similar to the approach for a constant number of bins, we guess the high and medium profit items for $\mathcal{B}_2$. Based on ideas for bin packing by Karmarkar and Karp [13] we again round the sizes of medium size items with medium profit. Furthermore, we show that these rounded items can be packed into $\mathcal{B}_2$ plus one additional bin b whose size is a small fraction of the remaining space in the bins after packing the high profit items. By an interesting exchange argument we can suppose that $\mathcal{B}_2$ contains the larger original medium sizes for each subinterval of rounded items. Using this argument we are able to guess the rounded medium sizes and to select the corresponding items for $\mathcal{B}_2$. To eliminate the additional bin, we apply here a shifting strategy on $\mathcal{B}_2$ and b. Finally we introduce a modified linear program relaxation to select the other items for the bin groups and generalize the rounding strategy also for the solution of this linear program.

Organization of the paper. In Section 2 we focus on instances of MKP having many bins with the largest capacity. Here we introduce the new rounding strategy for the solution of the linear program relaxation. We present an improved approximation algorithm for MKP where the bin group with the largest capacity has at least $\lceil 1/\delta \log(1/\delta)^2 \rceil$ bins. Next we give an approximation algorithm for bin packing instances with a constant number of bins. Then in Section 3 we present an improved approximation scheme for instances of MKP with a constant number of bins. Finally, we study the general case of MKP in Section 4. Here we combine the techniques of the previous sections (via a modified linear program relaxation and a generalized rounding strategy).

2 Instances of MKP with Many Bins with the Largest Capacity

Let $c_1 \leq \ldots \leq c_m$ be the capacities of the m bins in an instance of MKP. In this section we consider the case where there are at least $\lceil 1/\delta \log(1/\delta)^2 \rceil$ bins with the largest capacity c_m. The value δ depends on ϵ and will be specified later. For simplicity we suppose also that $1/\delta$ is integral. In the following we give an outline of our algorithm for this case:

(1) Solve an LP-relaxation of MKP approximately, split the bins into blocks of size close to

$\lceil 1/\delta \log(1/\delta)^2 \rceil$ and construct a 2D strip packing instance for each block.

(2) Round the corresponding rectangles block-wise and select items via a network flow computation.

(3) Use a bin packing algorithm with different bin sizes to pack the selected items into slightly more than $\lceil 1/\delta \log(1/\delta)^2 \rceil$ bins for each block.

(4) Apply a shifting strategy to select a subset of items that fits into the bins for each block.

For each bin b_k with capacity c_k, let $C_1^{(k)}, \ldots, C_{H_k}^{(k)}$ be the set of all configurations for the bin b_k; where a configuration is a subset $S \subseteq \mathcal{A}$ of items with total size $\sum_{a \in S} size(a) \leq c_k$. The main idea of the relaxation is to use a fractional variable $x_i \in [0, 1]$ for each item a_i (which selects a piece of each item) and to distribute the corresponding piece as smaller fractional pieces among the configurations for different bin capacities. The variable $y_j^{(k)}$ in the LP denotes the length of configuration $C_j^{(k)}$ (similar to a configuration variable in bin packing [9]). As relaxation we use the following LP for MKP instances; see also [12].

$$
\begin{array}{lcll}
\max \sum_{i=1}^{n} profit(a_i) x_i & & & \\
\sum_{k=1}^{m} \sum_{j: a_i \in C_j^{(k)}} y_j^{(k)} & = & x_i & \text{for } i = 1, \ldots, n, \\
\sum_{j=1}^{H_k} y_j^{(k)} & \leq & 1 & \text{for } k = 1, \ldots, m, \\
y_j^{(k)} & \geq & 0 & \text{for } j = 1, \ldots, H_k \text{ and } k = 1, \ldots, m, \\
x_i & \in & [0, 1] & \text{for } i = 1, \ldots, n.
\end{array}
$$

The LP above is a relaxation of MKP; the fractional selected items can be interpreted as rectangles and can be fractionally packed into rectangular regions of width c_k and height 1. We can compute an approximate solution $(\bar{x}, \bar{y})$ of the LP relaxation where $\sum_{j=1}^{H_k} \bar{y}_j^{(k)} \leq (1 + 2\alpha)$ for $k = 1, \ldots, m$ and whose objective value is at least $(1 - 3\alpha) OPT(LP) \geq (1 - 3\alpha) OPT(\mathcal{A}, \mathcal{B})$ where $\alpha = O(\epsilon)$ [12]. This solution which may violate some of the constraints slightly can be transformed into another solution $(\tilde{x}, \tilde{y})$ with objective value at least $(1 - 5\alpha) OPT(LP) \geq (1 - 5\alpha) OPT(\mathcal{A}, \mathcal{B})$ without violating the constraints above. For this step we simply scale the values $\tilde{y}_j^{(k)} = \bar{y}_j^{(k)} / (1 + 2\alpha)$ and $\tilde{x}_i = \bar{x}_i / (1 + 2\alpha)$.

2.1 New rounding strategy

In this subsection we propose a new rounding technique in order to select the items more efficiently. In the following we build t blocks B_ℓ with $M = \lceil 1/\delta \log(1/\delta)^2 \rceil$ bins each; with the exception of the block B_1 which might have fewer than M bins (see also Figure 1). We have $t = \lceil \frac{m}{M} \rceil$ bin groups where B_t contains the M largest bins $b_{m-M+1}, \ldots, b_m$, B_{t-1} the next M largest bins $b_{m-2M+1}, \ldots, b_{m-M}$ etc. By this construction the first group B_1 consists of either $\{b_1, \ldots, b_M\}$ if m is dividable by M or $\{b_1, \ldots, b_{m - \lfloor \frac{m}{M} \rfloor M}\}$ otherwise. We denote with $c_1^{(\ell)} \leq \ldots \leq c_M^{(\ell)}$ the capacities of the bins in block B_ℓ. Furthermore, let $c_{max}^{(\ell)} = c_M^{(\ell)}$ the maximum capacity in block B_ℓ. Notice that B_t contains M bins with the same capacity (using our assumption above).

Now consider an item a_i with $\tilde{x}_i > 0$. We denote with $z_i^{(\ell)} = \sum_{b_k \in B_\ell} \sum_{j: a_i \in C_j^{(k)}} \tilde{y}_j^{(k)}$ the fraction of item a_i assigned to block B_ℓ for $\ell = 1, \ldots, t$. For each block B_ℓ, all large pieces with $size(a_i) > \delta c_{max}^{(\ell)}$ can be interpreted as wide rectangles of the form $(size(a_i), z_i^{(\ell)})$ with width $size(a_i) \leq c_{max}^{(\ell)}$ and height $z_i^{(\ell)} \leq 1$. Next we stack all these rectangles ordered by their widths. We obtain a stack St_ℓ of height H_ℓ (see Figure 2). Now we add to the stack St_ℓ a set X_ℓ of dummy rectangles of width $\delta^2 c_{max}^{(\ell)}$ and height 1 (with the exception of one rectangle with height at most 1) until the modified stack $\overline{St}_\ell$ has

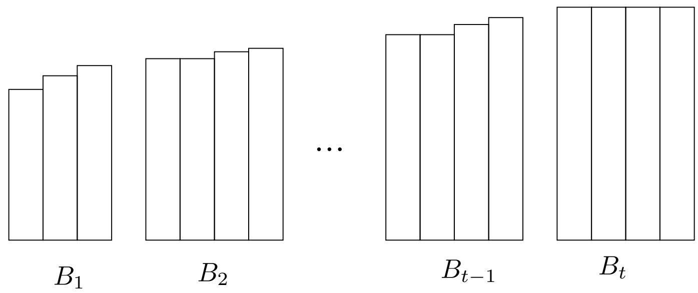

Figure 1: Building the blocks B_ℓ of bins with $\ell = 1, \ldots, t$.

total height $\overline{H}_\ell = d_\ell/\delta^2$ where $d_\ell \in \mathbb{Z}^+$ and $(d_\ell - 1)/\delta^2 < H_\ell \leq \bar{H}_\ell$. Notice that the total height of the rectangles in X_ℓ is at most $1/\delta^2$. Let $L^{(\ell)}_{wide}$ and $\bar{L}^{(\ell)}_{wide}$ be the sets of all rectangles on stack St_ℓ and $\overline{St}_\ell$, respectively.

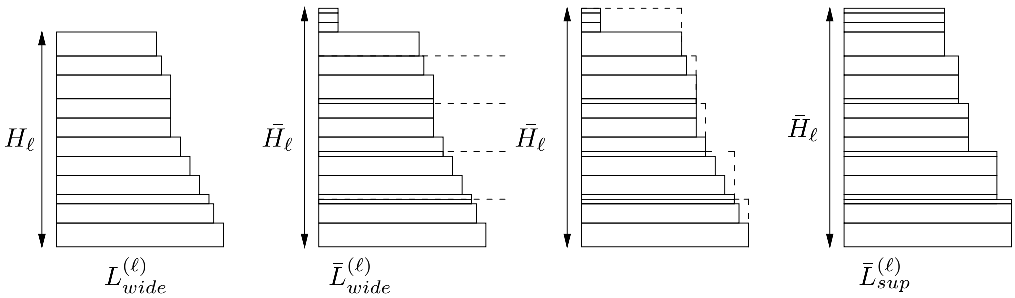

Figure 2: The construction of the stacks for $L^{(\ell)}_{wide}$, $\bar{L}^{(\ell)}_{wide}$ and $\bar{L}^{(\ell)}_{sup}$

Now we split the stack $\overline{St}_\ell$ into $1/\delta^2$ groups of height $\delta^2\bar{H}_\ell = d_\ell$. If a piece lies in two groups of the stack $\overline{St}_\ell$ (more than two groups is not possible, since the height of each rectangle is at most $1 \leq d_\ell$), then we split the rectangle into two rectangles that fit into their groups completely. Finally, we round up each rectangle in group j on stack $\overline{St}_\ell$ to the maximal width in group j. Let $\bar{L}^{(\ell)}_{sup}$ be the set of rectangles obtained after the rounding (see also Figure 2 for the construction of the stacks and sets of rectangles). Next we compare the minimum fractional strip packing height for the instances $L^{(\ell)}_{wide}$ and $\bar{L}^{(\ell)}_{sup}$ into a strip with different horizontal layers. Let $c_1 \leq \ldots \leq c_M$ be the widths of the M horizontal layers. The first $M-1$ layers have height 1 and layer M has unbounded height. The widths $c_1, \ldots, c_M$ are the bin capacities $c^{(\ell)}_1, \ldots, c^{(\ell)}_M$ in block B_ℓ. For a set L of N rectangles of the form $r_i = (w_i, h_i)$ with heights $h_i \leq 1$, let $R^{(k)}$ be a set of rectangles (called a configuration) that fits into a horizontal layer of width c_k; i.e. $\sum_{r_i \in R^{(k)}} w_i \leq c_k$. Let $R^{(k)}_1, \ldots, R^{(k)}_{H_k}$ be the set of configurations for width c_k. Use variables $v^{(k)}_1, \ldots, v^{(k)}_{H_k}$ to denote the heights of the configurations. The linear program $LP(L, B_\ell)$ for an instance L with N rectangles and block B_ℓ has the following form:

$$\begin{array}{lcll}
\min \sum_{j=1}^{H_M} v_j^{(M)} & & & \\
\sum_{k=1}^{M} \sum_{j: r_i \in R_j^{(k)}} v_j^{(k)} & = & h_i & \text{for } i = 1, \ldots, N, \\
\sum_{j=1}^{H_k} v_j^{(k)} & \leq & 1 & \text{for } k = 1, \ldots, M-1, \\
v_j^{(k)} & \geq & 0 & \text{for } j = 1, \ldots, H_k \text{ and } k = 1, \ldots, M.
\end{array}$$

For each instance L, let $LIN(L, B_\ell)$ be the value of the linear program above where the widths $c_1, \ldots, c_M$ are the capacities of the bins in block B_ℓ. This value is the minimum height of a fractional strip packing into a strip consisting of M horizontal layers of widths $c_1 \leq \ldots \leq c_M$. Notice that we count in the objective function of the LP above only the packing into the widest layer of width c_M. Let $AREA(L)$ be the total area of all rectangles in L. Since $L^{(\ell)}_{wide}$ fits fractionally into the M bins, $LIN(L^{(\ell)}_{wide}, B_\ell) \leq 1$.

Lemma 2.1

$$\begin{array}{rcl} LIN(\bar{L}^{(\ell)}_{sup}, B_\ell) & \leq & \delta M + 3, \\ AREA(\bar{L}^{(\ell)}_{sup}) & \leq & (1+\delta) AREA(L^{(\ell)}_{wide}) + 2c^{(\ell)}_{max}. \end{array}$$

Proof: The area $AREA(X_\ell)$ can be bounded by $(\bar{H}_\ell - H_\ell)\delta^2 c^{(\ell)}_{max} \leq (1/\delta^2)\delta^2 c^{(\ell)}_{max} = c^{(\ell)}_{max}$. This implies that $AREA(\bar{L}^{(\ell)}_{wide}) \leq AREA(L^{(\ell)}_{wide}) + c^{(\ell)}_{max}$ and $LIN(\bar{L}^{(\ell)}_{wide}, B_\ell) \leq LIN(L^{(\ell)}_{wide}, B_\ell) + 1 \leq 2$ (using an extra layer of height 1 for the additional rectangles in X_ℓ). Using the rounding of Kenyon and Remila [16] we obtain $LIN(\bar{L}^{(\ell)}_{sup}, B_\ell) \leq LIN(\bar{L}^{(\ell)}_{wide}, B_\ell) + \delta^2 \bar{H}_\ell$. Now $\bar{H}_\ell \leq H_\ell + 1/\delta^2$ and $H_\ell \delta c^{(\ell)}_{max} \leq AREA(L^{(\ell)}_{wide}) \leq M c^{(\ell)}_{max}$. Therefore, $\delta^2 H_\ell \leq \delta M$. These inequalities together imply $LIN(\bar{L}^{(\ell)}_{sup}, B_\ell) \leq LIN(L^{(\ell)}_{wide}, B_\ell) + \delta^2 H_\ell + 2 \leq \delta M + 3$. For the area bound again using [16], $AREA(\bar{L}^{(\ell)}_{sup}) \leq AREA(\bar{L}^{(\ell)}_{wide}) + \delta^2 \bar{H}_\ell c^{(\ell)}_{max} \leq AREA(\bar{L}^{(\ell)}_{wide}) + \delta^2(H_\ell + 1/\delta^2) c^{(\ell)}_{max} \leq AREA(L^{(\ell)}_{wide}) + 2c^{(\ell)}_{max} + \delta^2 H_\ell c^{(\ell)}_{max}$. Finally, using $\delta H_\ell c^{(\ell)}_{max} \leq AREA(L^{(\ell)}_{wide})$ we get $AREA(\bar{L}^{(\ell)}_{sup}) \leq (1+\delta) AREA(L^{(\ell)}_{wide}) + 2c^{(\ell)}_{max}$. □

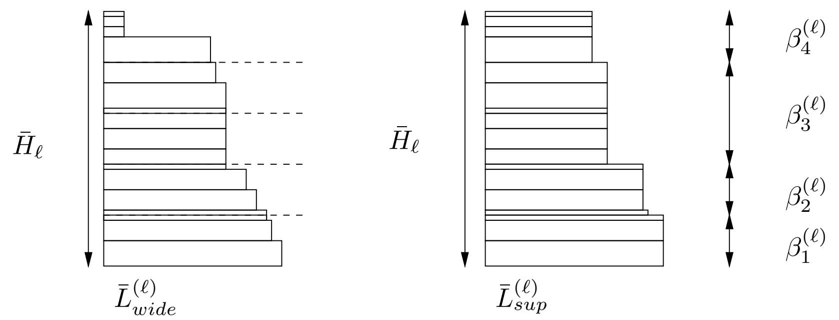

Figure 3: The heights $\beta^{(\ell)}_j$ for the groups j in $\bar{L}^{(\ell)}_{sup}$

$\bar{L}^{(\ell)}_{sup}$ consists of a set of *wide rectangles* with a constant number $a(\ell) \leq 1/\delta^2$ of different widths $w^{(\ell)}_1 > \ldots > w^{(\ell)}_{a(\ell)}$, where $w^{(\ell)}_{a(\ell)} > \delta c^{(\ell)}_{max}$, $w^{(\ell)}_1 \leq c^{(\ell)}_{max}$. For each width $w^{(\ell)}_j$, let $\beta^{(\ell)}_j$ be the total height of rectangles in $\bar{L}^{(\ell)}_{sup}$ with width $w^{(\ell)}_j$. We obtain a stack with $a(\ell)$ groups with different rectangle widths. Using the fact that the height of the stack $\overline{St}_\ell$ is equal to d_ℓ/δ^2 with $d_\ell \in \mathbb{N}_0$, the height of group j in stack $\overline{St}_\ell$ corresponding to $\bar{L}^{(\ell)}_{sup}$ is equal $\beta^{(\ell)}_j \in \mathbb{Z}^+$ (see Figure 3).

Let $L^{(\ell)}_{narrow}$ be the set of *narrow rectangles* $(size(a_i), z^{(\ell)}_i)$ with $size(a_i) \leq \delta c^{(\ell)}_{max}$ allocated to bins in block B_ℓ. The total area of these rectangles is denoted with $AREA(L^{(\ell)}_{narrow}) = \sum_{i: size(a_i) \leq \delta c^{(\ell)}_{max}} z^{(\ell)}_i size(a_i)$. We divide this area into smaller areas as follows. For each interval $int_{\ell,k} = (\frac{\delta}{(1+\delta)^k} c^{(\ell)}_{max}, \frac{\delta}{(1+\delta)^{k-1}} c^{(\ell)}_{max}]$ with $k \in \mathbb{N}$, let $Area(\ell, int_{\ell,k}) = \sum_{i: size(a_i) \in int_{\ell,k}} z^{(\ell)}_i size(a_i)$ be the total area of pieces of items a_i with $size(a_i) \in int_{\ell,k}$ allocated to bins in block B_ℓ. Using the smallest original item size in interval $int_{\ell,k}$, the maximum number of possible items in $int_{\ell,k}$ with this total area is $\eta_{\ell,k} = \left\lceil \frac{Area(\ell, int_{\ell,k})}{\frac{\delta}{(1+\delta)^k} c^{(\ell)}_{max}} \right\rceil$. In our algorithm, it is necessary to take all intervals $int_{\ell,k}$ for which $\frac{\delta}{(1+\delta)^k} c_{max}^{(\ell)} \geq \frac{1}{2n} c_{max}^{(\ell)}$. Let $index(\ell)$ be the largest index k such that this inequality is satisfied. The inequality is also equivalent to $2\delta n \geq (1+\delta)^k$ or $\log(2\delta n) \geq k \log(1+\delta)$. This implies an upper bound $k \leq \frac{\log(2\delta n)}{\log(1+\delta)}$. Therefore, we set $index(\ell) = \lfloor \frac{\log(2\delta n)}{\log(1+\delta)} \rfloor + 1$ and obtain that $index(\ell) = O([\log(\epsilon) + \log(n)]/\log(1+\epsilon))$ using $\delta = \theta(\epsilon)$. For items with $size(a_i) \leq \frac{1}{2n} c_{max}^{(\ell)}$ we use an additional interval $int_{\ell,index(\ell)+1} = (-\infty, \delta/(1+\delta)^{index(\ell)}]$ and set $\eta_{\ell,index(\ell)+1} = n$. Notice that this construction for the narrow items implies that the number of intervals is bounded by a polynomial in n and $1/\epsilon$.

Lemma 2.2 *If $A^{(\ell)} \subseteq \mathcal{A}$ is a set of small items with $|\{a_i \in A^{(\ell)} | size(a_i) \in int_{\ell,k}\}| \leq \eta_{\ell,k}$ for each k, then $Area(A^{(\ell)}) \leq (1+\delta) Area(L_{narrow}^{(\ell)}) + (3/2+\delta) c_{max}^{(\ell)}$.*

Proof: Using the bound above, the total area of $\{a_i \in A^{(\ell)} | size(a_i) \in int_{\ell,k}, k \leq index(\ell)\}$ can be bounded by

$$\eta_{\ell,k} \frac{\delta}{(1+\delta)^{k-1}} c_{max}^{(\ell)} \leq \left(\frac{Area(\ell, int_{\ell,k})}{\frac{\delta}{(1+\delta)^k} c_{max}^{(\ell)}} + 1\right) \frac{\delta}{(1+\delta)^{k-1}} c_{max}^{(\ell)} \leq (1+\delta) Area(\ell, int_{\ell,k}) + \frac{\delta}{(1+\delta)^{k-1}} c_{max}^{(\ell)}.$$

For items with $size(a_i) \leq 1/(2n) c_{max}^{(\ell)}$, the total area is bounded by $c_{max}^{(\ell)}/2$. Summing up the total area over all intervals and using the geometric sum $\sum_{k \geq 0} 1/(1+\delta)^k = (1+\delta)/\delta$, we obtain $Area(A^{(\ell)}) \leq (1+\delta) \sum_{k \geq 1} Area(\ell, int_{\ell,k}) + c_{max}^{(\ell)} \sum_{k \geq 1} \frac{\delta}{(1+\delta)^{k-1}} + c_{max}^{(\ell)}/2 \leq (1+\delta) Area(L_{narrow}^{(\ell)}) + (3/2+\delta) c_{max}^{(\ell)}$. □

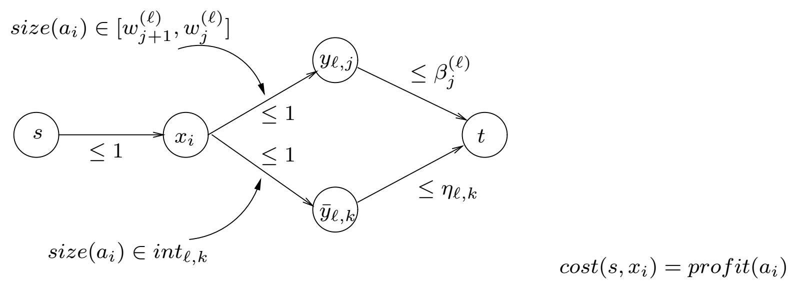

Figure 4: Flow network to select the items

Now we set up a flow network $G = (N, E)$ of the following form (see also Figure 4 for an illustration). The vertex set consists of a source s and sink t, a node x_i for each item a_i and several nodes for each block B_ℓ with (at most) M bins. For each block, we have a node $y_{\ell,j}$ for each rounded wide width $w_j^{(\ell)}$ where $j = 1, \ldots, a(\ell)$ and a node $\bar{y}_{\ell,k}$ for each interval $int_{\ell,k}$ where $k = 1, \ldots, index(\ell) + 1$. The edge set E is defined by

$$\{(s, x_i) | i = 1, \ldots, n\} \cup \{(x_i, y_{\ell,j}) | w_{j+1}^{(\ell)} \leq size(a_i) \leq w_j^{(\ell)}\} \cup \{(x_i, \bar{y}_{\ell,k}) | size(a_i) \in int_{\ell,k}\} \cup$$
$$\{(y_{\ell,j}, t) | \ell = 1, \ldots, t \text{ and } j = 1, \ldots, a(\ell)\} \cup \{(\bar{y}_{\ell,k}, t) | \ell = 1, \ldots, t \text{ and } k = 1, \ldots, index(\ell) + 1\}$$

All edges have lower capacities 0. The upper capacities of the edges (s, x_i), $(x_i, y_{\ell,j})$ and $(x_i, \bar{y}_{\ell,k})$ are 1. In addition the capacity of the edge $(y_{\ell,j}, t)$ is $\beta_j^{(\ell)}$ and the capacity of the edge $(\bar{y}_{\ell,k}, t)$ is $\eta_{\ell,k}$. Notice that all capacities are integral and that the number of vertices in G is bounded by a polynomial in n and $1/\epsilon$ using $\delta = O(\epsilon)$. Furthermore, we have cost values for each edge: for each edge (s, x_i) the cost value $c(s, x_i) = profit(a_i)$ and for each other edge the cost value is 0.

Lemma 2.3 *There is a fractional flow in the network $G = (N, E)$ with profit at least $(1-5\alpha)OPT(LP)$.*

Proof: The LP solution $(\tilde{x}, \tilde{y})$ has objective value at least $\sum_i \tilde{x}_i \mathit{profit}(a_i) \geq (1-5\alpha)OPT(LP)$. We can define a feasible flow in the network G as follows. For each arc $(s, x_i) \in E$, we set $f((s, x_i)) = \tilde{x}_i \in [0, 1]$. Suppose that a piece of a rectangle corresponding to item a_i with $\mathit{size}(a_i) \in (w_{j+1}^{(\ell)}, w_j^{(\ell)}]$ lies on stack St_ℓ, i.e. $z_i^{(\ell)} > 0$ and $\mathit{size}(a_i) > \delta c_{max}^{(\ell)}$. Notice that items of original size $w_j^{(\ell)}$ might be rounded up to the next larger value $w_{j-1}^{(\ell)}$. Then either the entire rectangle $(\mathit{size}(a_i), z_i^{(\ell)})$ lies in group $j-1$ or j of the stack corresponding to $\bar{L}_{sup}^{(\ell)}$ or the rectangle is divided into two pieces $(\mathit{size}(a_i), z_i^{(\ell)}(0))$ and $(\mathit{size}(a_i), z_i^{(\ell)}(1))$ with $z_i^{(\ell)} = z_i^{(\ell)}(0) + z_i^{(\ell)}(1)$ which lie in two consecutive groups $j-1, j \in \{1, \ldots, a(\ell)\}$. In the first case we set $f((x_i, y_{\ell,j-1})) = z_i^{(\ell)} \in [0,1]$ or $f((x_i, y_{\ell,j})) = z_i^{(\ell)}$ and in the second case we use $f((x_i, y_{\ell,j-1})) = z_i^{(\ell)}(0) \in [0, 1]$ and $f((x_i, y_{\ell,j})) = z_i^{(\ell)}(1) \in [0, 1]$. Note that the flow value $\sum_i f((x_i, y_{\ell,j})) = f((y_{\ell,j}, t)) \leq \beta_j^{(\ell)}$ using the properties of the stack.

If a piece of a rectangle corresponding to item a_i is narrow (i.e. $z_i^{(\ell)} > 0$ and $\mathit{size}(a_i) \leq \delta c_{max}^{(\ell)}$) and $\mathit{size}(a_i) \in \mathit{int}_{\ell,k}$, then we define $f((x_i, \bar{y}_{\ell,k})) = z_i^{(\ell)} \in [0, 1]$. For $k \leq \mathit{index}(\ell)$ we can show that

$$f((\bar{y}_{\ell,k}, t)) = \sum_i f((x_i, \bar{y}_{\ell,k})) = \sum_{i:\mathit{size}(a_i) \in \mathit{int}_{\ell,k}} z_i^{(\ell)} \leq \eta_{\ell,k}.$$

Notice that

$$\sum_{i:\mathit{size}(a_i) \in \mathit{int}_{\ell,k}} z_i^{(\ell)} \frac{\delta c_{max}^{(\ell)}}{(1+\delta)^k} \leq \mathit{Area}(\ell, \mathit{int}_{\ell,k}) \leq \sum_{i:\mathit{size}(a_i) \in \mathit{int}_{\ell,k}} z_i^{(\ell)} \frac{\delta c_{max}^{(\ell)}}{(1+\delta)^{k-1}}.$$

Therefore, $\sum_{i:\mathit{size}(a_i) \in \mathit{int}_{\ell,k}} z_i^{(\ell)} \leq \frac{\mathit{Area}(\ell, \mathit{int}_{\ell,k})}{\frac{\delta}{(1+\delta)^k} c_{max}^{(\ell)}} \leq \eta_{\ell,k}$. For $k = \mathit{index}(\ell) + 1$, we have

$$f((\bar{y}_{\ell,k}, t)) = \sum_{i:\mathit{size}(a_i) \in \mathit{int}_{\ell,k}} f((x_i, \bar{y}_{\ell,k})) \leq \sum_i z_i^{(\ell)} \leq n = \eta_{\ell,k}.$$

This shows that the flow f is feasible and the cost of f is equal to $\sum_{(s,x_i) \in E} f((s, x_i))c(s, x_i) = \sum_i \tilde{x}_i \mathit{profit}(a_i) \geq (1-5\alpha)OPT(LP)$. □

Taking negative profit values, there is a minimum cost flow in the network with cost $\leq -(1 - 5\alpha)OPT(LP)$. Since we have a totally unimodular constraint matrix, each basic solution in the linear program corresponding to the flow problem is integral. Therefore, there is a minimum cost integral flow (among all flow values) in the network with the same cost. By computing the minimum cost flow for each integral flow value $v = 1, \ldots, n$ and taking the best solution, we obtain an integral flow $g : E \to \mathbb{N}$ in the network with profit $\geq (1-5\alpha)OPT(LP)$. This integral flow gives us a subset of selected items $A_{select} = \{a_i | g((s, x_i)) = 1\}$ with profit close to the optimum profit. For each block B_ℓ with M bins, let $A_{wide}^{(\ell)} = \{a_i | \exists j \in \{1, \ldots, a(\ell)\}$ with $g(x_i, y_{\ell,j}) = 1\}$ and $A_{narrow}^{(\ell)} = \{a_i | \exists k \in \{1, \ldots, \mathit{index}(\ell) + 1\}$ with $g(x_i, \bar{y}_{\ell,k}) = 1\}$ be the set of wide and narrow items for the block B_ℓ, respectively. Let $\bar{A}_{wide}^{(\ell)}$ and $\bar{A}_{narrow}^{(\ell)}$ be corresponding sets with rectangles of width equal to the $\mathit{size}(a)$ and height 1. We obtain the following result.

Lemma 2.4 *The algorithm above computes sets $A_{wide}^{(\ell)}, A_{narrow}^{(\ell)}$ of items for each block B_ℓ for $\ell = 1, \ldots, t$ with profit$(\bigcup_\ell A_{wide}^{(\ell)} \cup A_{narrow}^{(\ell)}) \geq (1-5\alpha)OPT(LP)$ such that the following properties are satisfied:*

- $|\{a_i \in A_{wide}^{(\ell)} | g(x_i, y_{\ell,j}) = 1\}| \leq \beta_j^{(\ell)}$ *for each $j = 1, \ldots, a(\ell)$ and $\ell = 1, \ldots, t$.*

- $|\{a_i \in A^{(\ell)}_{narrow} | g(x_i, \bar{y}_{\ell,k}) = 1\}| \leq \eta_{\ell,k}$ *for each* $j = 1, \dots, index(\ell) + 1$ *and* $\ell = 1, \dots, t$.

Furthermore, $\bar{A}^{(\ell)}_{wide} \leq \bar{L}^{(\ell)}_{sup}$*,* $LIN(\bar{A}^{(\ell)}_{wide}, B_\ell) \leq \delta M + 3$*,* $AREA(\bar{A}^{(\ell)}_{wide}) \leq (1+\delta) AREA(L^{(\ell)}_{wide}) + 2c^{(\ell)}_{max}$*, and* $Area(\bar{A}^{(\ell)}_{narrow}) \leq (1+\delta) Area(L^{(\ell)}_{narrow}) + (3/2+\delta) c^{(\ell)}_{max}$*.*

2.2 Bin packing and shifting strategy

It remains to prove that we can pack the item set $A^{(\ell)}_{wide} \cup A^{(\ell)}_{narrow}$ into slightly more than M bins for each $\ell = 1, \dots, t$. In the following we prove an upper bound for the number of bins used for $A^{(\ell)}_{wide} \cup A^{(\ell)}_{narrow}$. Let $OPT_{ILP}(S, B_\ell)$ be the minimum value of an integral solution for $LP(S, B_\ell)$ for a subset $S \subseteq \mathcal{A}$ of items. This value corresponds to the number of bins of capacity $c^{(\ell)}_{max}$ used for S, where the $M-1$ bins of capacities $c^{(\ell)}_1, \dots, c^{(\ell)}_{M-1}$ are not counted, but can be used as additional space.

To prove the upper bound, we first study a natural generalization of bin packing with d different item sizes $s_1 > \dots > s_d$. In our problem, the bins in general have also different sizes. An instance I consists here of n_i items of size s_i, for $i = 1, \dots, d$ and M bins $b_1, \dots, b_M$ with sizes $c_1 \leq \dots \leq c_M$. A configuration $K^{(k)}$ is a multiset $\{a_1 : s_1, \dots, a_d : s_d\}$ such that $\sum_{j=1}^{d} a_j s_j \leq c_k$ (i.e. the items fit into a bin of size c_k). Again, let $K^{(k)}_1, \dots, K^{(k)}_{H_k}$ be the sequence of all configurations to bin size c_k. We use a variable $v^{(k)}_j$ to indicate the fractional number of bins with configuration $K^{(k)}_j$. Furthermore, let $a(K^{(k)}_j, s_i)$ denote how often size s_i occurs in configuration $K^{(k)}_j$. The LP (and also the ILP) for an instance $I = (\mathcal{A}, \mathcal{B})$ with $\mathcal{A} = \{n_1 : s_1, \dots, n_d : s_d\}$ and $\mathcal{B} = \{b_1, \dots, b_m\}$ with item sizes $s_1, \dots, s_d$ and bin capacities $c_1, \dots, c_M$ can be rewritten as:

$$\begin{array}{lcll}
\min \sum_{j=1}^{H_M} v^{(M)}_j & & & \\
\sum_{k=1}^{M} \sum_{K^{(k)}_j} a(K^{(k)}_j, s_i) v^{(k)}_j & = & n_i & \text{for } \; i = 1, \dots, d, \\
\sum_{j=1}^{H_k} v^{(k)}_j & = & 1 & \text{for } \; k = 1, \dots, M-1, \\
v^{(k)}_j & \geq & 0 & \text{for } \; j = 1, \dots, H_k \;\text{ and }\; k = 1, \dots, M.
\end{array}$$

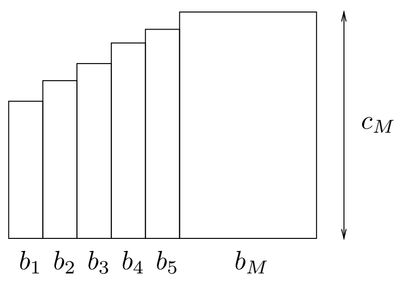

Figure 5: An instance for our bin packing problem with different capacities.

Notice that $v^{(k)}_j \leq 1$ for $k = 1, \dots, M-1$ and that $v^{(M)}_j$ are unbounded. In the objective function we only count the number of bins with largest capacity c_M (the structure of the bins where we use an arbitrary number of extra bins of capacity c_M is given in Figure 5). For a given instance I, let $LIN(I)$ be the minimum value of the linear program and let $OPT(I)$ be the minimum value of the corresponding integer program where all values $v^{(k)}_j$ are integral. Furthermore, note that some of the configurations are not maximal with respect to the capacities of the bins (and maybe empty sets). The number of variables in the LP and ILP is exponentially large, but the number of equalities is only $M-1+d$. Therefore, each basic solution has at most $M-1+d$ variables $v_j^{(k)}$ with value larger than 0. We can prove the following result:

Lemma 2.5 *The number of bins with fractional variables $v_j^{(k)} \in (0,1)$ for $k \leq M-1$ plus the number of fractional variables $v_j^{(M)} \notin \mathbb{N}$ is at most d.*

Proof: Let x be the number of bins with index at most $M-1$ with a fractional variable $v_j^{(k)}$ and let y be the number of fractional variables $v_j^{(M)}$ for bins with capacity c_M. Since $\sum_{j=1}^{H_k} v_j^{(k)} = 1$ for $k = 1, \ldots, M-1$, each of first $M-1$ bins with a fractional variable $v_j^{(k)}$ has at least two fractional variables. Therefore, the total number of fractional variables is at least $2x + (M-1-x) + y = M-1+x+y$. If $x+y \geq d+1$, then we get at least $M+d$ fractional variables. But this cannot happen since as described above the maximum number of positive variables is at most $M-1+d$. □

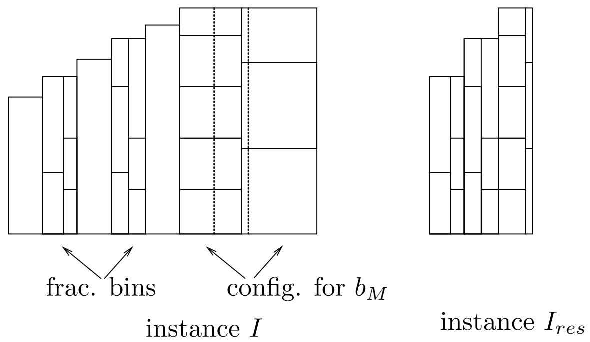

Figure 6: The instances I and I_{res} for a given solution of the linear program.

The proof also shows that the number of bins with fractional variables $v_j^{(k)} \in (0,1)$ for $k \leq M-1$ plus the number of variables $v_j^{(M)} > 0$ is at most d. Now consider an optimum basic solution (v) of the LP for instance I. Removing the integral values from the solution $\bar{v}_j^{(k)} = v_j^{(k)} - \lfloor v_j^{(k)} \rfloor$ and the bins $b_k \in \{b_1, \ldots, b_{M-1}\}$ with only an integral variable $v_j^{(k)} = 1$ generates a fractional solution $(\bar{v})$ for a residual instance I_{res}. The instance I_{res} has a reduced number of bins bounded by d or $d+1$. Notice that we obtain $d+1$ bins, only if the values $v_j^{(M)} = 0$ for all configurations for bin b_M. This implies that in this case instance I does not use bin b_M and that $LIN(I) = 0$. The reduction is illustrated in Figure 6 where we remove the integral values for the original instance I and obtain the residual instance I_{res} with smaller total size. Using the Lemma above and $\bar{v}_j^{(k)} < 1$, the total size $Area(I_{res}) \leq dc_{max}$. To see this, notice that for each bin b_k in I_{res} we get a size of at most $c_k \leq c_{max}$ and for each fractional variable corresponding to b_M we get at most c_{max}. Since the number of bins plus the number of fractional variables is at most d, $Area(I_{res}) \leq dc_{max}$. Now we can prove an upper bound for the integrality gap for bin packing with different bin sizes similar to a result for the classical bin packing problem [13, 23].

Theorem 2.1 *Let I be an instance of the bin packing problem with capacities $c_1, \ldots, c_M$ and d different item sizes. Then $OPT(I) \leq LIN(I) + C\log(d)^2$, where C is a constant independent of I.*

Proof: First notice the following Lemma comparing the original and residual instance:

Lemma 2.6 *Let I be an instance of the bin packing problem with different sizes $c_1, \ldots, c_M$, and let I_{res} be its residual instance. Then, $OPT(I) - LIN(I) \leq OPT(I_{res}) - LIN(I_{res})$.*

Proof: Notice that

$$LIN(I) = \sum_j v_j^{(M)} = \sum_j \lfloor v_j^{(M)} \rfloor + \sum_j (v_j^{(M)} - \lfloor v_j^{(M)} \rfloor) = \sum_j \lfloor v_j^{(M)} \rfloor + LIN(I_{res}).$$

Let A be the constraint matrix for instance I and $b = (n_1, \ldots, n_d, 1, \ldots, 1)^T \in \mathbb{R}^{d+M-1}$ the right hand side of the system $Av = b$ of equalities. Furthermore, let b_{res} be the residual vector with $b_{res} = (n'_1, \ldots, n'_d, y_1, \ldots, y_{M-1})$, where n'_i is the reduced number of items in I_{res} of size s_i, $i = 1, \ldots, d$ and $y_j \in \{0, 1\}$ depending on whether the bin b_j is covered by the integral part of the solution. Then, $b = Av = A(\lfloor v \rfloor) + A(v - \lfloor v \rfloor) = A(\lfloor v \rfloor) + b_{res}$. This implies that $v - \lfloor v \rfloor$ is feasible with respect to A and b_{res}. In other words, a feasible solution for I_{res} gives also a feasible solution for I by adding the integral part $\lfloor v \rfloor$. This gives the objective value $LIN(I) = \sum_j \lfloor v_j^{(M)} \rfloor + LIN(I_{res})$. On the other hand, $OPT(I) \leq \sum_j \lfloor v_j^{(M)} \rfloor + OPT(I_{res})$. Therefore, we obtain $OPT(I) - LIN(I) \leq OPT(I_{res}) - LIN(I_{res})$. □

The lemma implies that the integrality gap is attained for a residual instance. Using a similar argument as in Lemma 2.5, the number d' of bins with fractional variables $v_j^{(k)} \in (0, 1)$ for $k \leq M - 1$ plus the objective value $LIN(I_{res})$ (counting the fractional packing into bins of size c_M) is bounded by d. This implies also that $d' + \lceil LIN(I_{res}) \rceil \leq d$. Since the total area $Area(I_{res}) \leq dc_{max}$, we further can eliminate the smaller sizes $\leq (1/d)c_{max}$.

Lemma 2.7 *Let I_{res} be a residual instance of the bin packing problem with different bin sizes bounded by c_{max}, and let I_{red} be the instance obtained from I_{res} by eliminating all items of size bounded by $(1/d)c_{max}$. Then $OPT(I_{res}) - \lceil LIN(I_{res}) \rceil \leq \max\{OPT(I_{red}) - \lceil LIN(I_{red}) \rceil, 1\}$.*

Proof: Taking an optimum packing for I_{red}, we use the first fit (FF) algorithm to pack the remaining items of size bounded by $(1/d)c_{max}$. If FF does not use a new bin, then $OPT(I_{red}) = OPT(I_{res})$. Using $LIN(I_{res}) \geq LIN(I_{red})$, the inequality above follows directly.

Otherwise we obtain a packing, where each bin except the last, is filled up to the capacity of at least $c_i - (1/d)c_{max}$. In our residual instance, there are at most d' bins with index at most $M - 1$ plus a number of bins with capacity c_M. Suppose that $OPT(I_{res}) > \lceil LIN(I_{res}) \rceil + 1$ (otherwise $OPT(I_{res}) - \lceil LIN(I_{res}) \rceil \leq 1$ and we are done). Notice that this assumption implies that $OPT(I_{res}) \geq 2$. Furthermore, all d' bins with index at most $M - 1$ and $\lceil LIN(I_{res}) \rceil$ bins with larger capacities c_M are filled up to at most $(1/d)c_{max}$. This holds also for the solution generated by FF. Since $d' + \lceil LIN(I_{res}) \rceil \leq d$ and I_{res} fits fractionally into $d' + \lceil LIN(I_{res}) \rceil$ bins, all items in I_{res} with the exception of a set of items with total area $d \cdot (1/d)c_{max} = c_{max}$ are packed into the first $d' + \lceil LIN(I_{res}) \rceil$ bins. Since the remaining set must fit in one bin of size c_{max}, FF uses at most $\lceil LIN(I_{res}) \rceil + 1$ bins of size c_{max}. Therefore, $OPT(I_{res}) \leq \lceil LIN(I_{res}) \rceil + 1$. □

Now we can prove Theorem 2.1 by induction on d. For $d = 1$, a greedy solution that packs as many items of size s_1 into the bins gives an optimum solution for the LP and ILP and, therefore, $OPT(I) = LIN(I)$. Suppose that $d > 1$ and consider a residual instance I_{res} of our bin packing problem. Furthermore, we can eliminate all items of size at most $(1/d)c_{max}$ and get either 2 or $OPT(I_{red}) - LIN(I_{red}) + 1$ as integrality gap. Consider now I_{red} and reduce the number of item sizes to $d/2$. Using our result for I_{res}, the total area $Area(I_{red}) \leq Area(I_{res}) \leq dc_{max}$. For simplicity suppose that $c_{max} = 1$. Then, I_{red} contains only items of size at least $1/d$ and area $Area(I_{res}) \leq d$.

We order the items according to their sizes $a_1 \geq \ldots \geq a_n$ and split the item set into different groups. For the first group G_1 we take the largest k_1 items with sizes $a_1, \ldots, a_{k_1}$ such that $\sum_{i=1}^{k_1} a_i > 2$ and $\sum_{i=1}^{k_1-1} a_i \leq 2$. Then, we define the second group G_2 in the same way, with items $a_{k_1+1}, \ldots, a_{k_1+k_2}$, until their total size is also larger than 2. We repeat this process until all items are considered and obtain sets $G_1, \ldots, G_K$ with $k_i = |G_i|$ items for $i = 1, \ldots, K$. The number K of groups is at most $\lfloor Area(I_{red})/2 \rfloor + 1 \leq \lfloor d/2 \rfloor + 1$. Since all items have size $> 1/d$, $k_i \leq 2d$ for $i = 1, \ldots, K$. Using the ordering of the items, we obtain $k_1 \leq \ldots \leq k_{K-1}$.

Now, let G'_i be the set of the largest k_{i-1} items from G_i for $i = 2, \ldots, K-1$. Then, $G_{i-1} \geq G'_i$ for $i = 2, \ldots, K-1$. Furthermore, let H_i be the set of items obtained from G'_i by rounding up the sizes to the largest size in G'_i, for $i = 2, \ldots, K-1$. Then, $G'_i \leq H_i \leq G_{i-1}$, for $i = 2, \ldots, K-1$ where $\leq$ is a partial order on bin packing instances with the interpretation that $I_A \leq I_B$ if there exists a one-to-one function $f : I_A \rightarrow I_B$ such that $size(x) \leq size(f(x))$ for each item $x \in I_A$. We use now as new instance $J = \bigcup_{i=2}^{K-1} H_i$ with $K - 2 \leq \lfloor d/2 \rfloor - 1$ item sizes and a remaining item set $J' = G_1 \cup G_K \cup \bigcup_{i=2}^{K-1}(G_i \setminus G'_i)$. Below we will bound the total size of J'. But first notice that $J \leq I_{red} \leq J \cup J'$. This implies for the LP and ILP values for bin packing with different sizes:

$$\begin{array}{lclcl} OPT(J) & \leq & OPT(I_{red}) & \leq & OPT(J \cup J'), \\ LIN(J) & \leq & LIN(I_{red}) & \leq & LIN(J \cup J'). \end{array}$$

This implies that the integrality gap $OPT(I_{red}) - LIN(I_{red})$ can be bounded by $OPT(J \cup J') - LIN(J) \leq OPT(J) - LP(J) + Bin(J')$ (where $Bin(J')$ denotes the number of additional bins of maximum size $c_{max} = 1$ for J'). In the following we give a bound for the number $Bin(J') \leq 2Area(J') + 1$. The total area $Area(J') \leq Area(G_1 \cup G_K) + \sum_{i=2}^{K-1} Area(G_i \setminus G'_i) \leq 6 + \sum_{i=2}^{K-1} Area(G_i \setminus G'_i)$. Since $k_i - 1$ items in G_i have total size at most 2, $Area(G_i \setminus G'_i) \leq 2\frac{k_i - k_{i-1}}{k_i - 1}$. This implies $size(G') \leq 6 + 2\sum_{i=2}^{K-1} \frac{k_i - k_{i-1}}{k_i - 1} \leq 6 + 2\sum_{i=2}^{K-1} \sum_{j=k_{i-1}}^{k_i - 1} \frac{1}{j} = 6 + 2\sum_{j=k_1}^{k_{K-1}-1} \frac{1}{j} \leq 6 + 2\ln(k_{K-1}) \leq 6 + 2\ln(2d)$. Therefore, the number $Bin(J')$ of bins for J' is at most $13 + 4\ln(2d) \leq C' \ln(d) \leq C' \log(d)$ for a constant C'. We set $C = C' + 1$. For $d = 2, 3$, the number $K - 2$ of item sizes is $\lfloor d/2 \rfloor - 1 = 0$. In this case, $J = \emptyset$ and $LIN(I_{red}) \geq 0$. This implies that $OPT(I) - LIN(I) \leq \max\{2, OPT(I_{red}) - LIN(I_{red}) + 1\} \leq Bin(J') + 1 \leq C' \log(d) + 1 \leq (C' + 1)\log(d)^2 = C \log(d)^2$ for $d = 2, 3$.

By induction for $d \geq 4$ we have $OPT(J) - LIN(J) \leq C \log(\lfloor d/2 \rfloor - 1)^2$ (using that J has at most $\lfloor d/2 \rfloor - 1$ item sizes). Therefore, $OPT(I) - LIN(I) \leq \max\{2, C \log(\lfloor d/2 \rfloor - 1)^2 + C \log(d) + 1\}$. The second term can be bounded by $C \log(d)^2 - C \log(d) + C + 1 \leq C \log(d)^2$ (using that $C + 1 \leq 4C \leq C \log(d)^2$ for $d \geq 4$ and $C \geq 1$). □

Notice that the Theorem 2.1 also gives an algorithm for the bin packing problem with different bin capacities that uses only $\bar{C} \log(d)^2$ additional bins (where $\bar{C}$ is a constant that is slightly larger than C). It is a generalization of the algorithm by Karmarkar and Karp for the classical bin packing problem that computes in each recursive step an approximate solution of the linear program with value $LIN(I) + 1$. We apply the Theorem above to the rounded wide items selected for block B_ℓ. Notice that the number of sizes d of large items is bounded by $1/\delta^2$ for each block B_ℓ.

Lemma 2.8

$$OPT_{ILP}(A^{(\ell)}_{wide} \cup A^{(\ell)}_{narrow}, B_\ell) \leq C' \log(1/\delta)^2$$

where C' is a constant.

Proof: Using our selection algorithm, $\bar{A}^{(\ell)}_{wide} \leq \bar{L}^{(\ell)}_{sup}$. Rounding up all items in $A^{(\ell)}_{wide}$ according to values of the stack for $\bar{L}^{(\ell)}_{sup}$ generates a subset of items $\tilde{A}^{(\ell)}_{wide}$ with $d = 1/\delta^2$ item sizes. Using Theorem 2.1, $OPT_{ILP}(A^{(\ell)}_{wide}, B_\ell) \leq OPT_{ILP}(\tilde{A}^{(\ell)}_{wide}, B_\ell) \leq LIN(\tilde{A}^{(\ell)}_{wide}, B_\ell) + C(\log d)^2$ where C is a constant independent of the instance. In our algorithm, $d = 1/\delta^2$ in each block B_ℓ and $LIN(\tilde{A}^{(\ell)}_{wide}, B_\ell) \leq LIN(\bar{L}^{(\ell)}_{sup}, B_\ell) \leq \delta M + 3$. This implies that $OPT_{ILP}(A^{(\ell)}_{wide}, B_\ell) \leq \delta M + 3 + C(\log(1/\delta^2))^2 \leq C' \log(1/\delta)^2$ for a constant C'. Since $Area(L^{(\ell)}_{narrow} \cup L^{(\ell)}_{wide}) \leq \sum_{i=1}^{M} c^{(\ell)}_i$, we obtain $AREA(\bar{A}^{(\ell)}_{wide} \cup \bar{A}^{(\ell)}_{narrow}) \leq (1+\delta)Area(L^{(\ell)}_{narrow} \cup L^{(\ell)}_{wide}) + 4c^{(\ell)}_{max} \leq \sum_{i=1}^{M-1} c^{(\ell)}_i + \delta \sum_{i=1}^{M-1} c^{(\ell)}_i + (5+\delta)c^{(\ell)}_{max} \leq$

$\sum_{i=1}^{M-1} c_i^{(\ell)} + [5 + \log(1/\delta)^2 + \delta]c_{max}^{(\ell)}$. Using this area bound, all items fit fractionally into the $M-1$ bins of sizes $c_1^{(\ell)}, \ldots, c_{M-1}^{(\ell)}$ and at most $\lceil \log(1/\delta)^2 + \delta \rceil + 5$ bins of size $c_{max}^{(\ell)}$.

Now, there are two cases depending on whether the greedy algorithm for the narrow items uses extra bins or not. Let X be the number of bins generated by the greedy algorithm (starting with the optimum solution for the wide items).

Case 1: $X \leq OPT_{ILP}(A_{wide}^{(\ell)}, B_\ell)$. Here we do not use any further bins.

Case 2: $X > OPT_{ILP}(A_{wide}^{(\ell)}, B_\ell)$. Consider a packing into the first $M' = [M + \lceil \log(1/\delta)^2 + \delta \rceil + 4]$ bins. If $X \leq M'$, then we obtain also a bound for the number of bins. If X is larger than M', then the first M' bins are full except for an amount of at most $\delta c_{max}^{(\ell)}$. Since all items fit fractionally into the M' bins, an area of at most $\delta M' c_{max}$ is not covered. For this area $\delta M' c_{max}^{(\ell)}$, we need at most $y = \lceil \delta/(1-\delta) M' \rceil$ additional bins filled to at least $(1-\delta)c_{max}^{(\ell)}$. The number y can be bounded by $\lceil 2\log(1/\delta)^2 \rceil$ for $\delta \leq 1/4$. In this case $3\lceil \log(1/\delta)^2 + \delta \rceil + 4 \leq C' \log(1/\delta)^2$ additional bins of size $c_{max}^{(\ell)}$ are sufficient to pack all items (using $\delta \leq 1/4$ and $C' \geq 5$).

Therefore, $OPT_{ILP}(A_{wide}^{(\ell)} \cup A_{narrow}^{(\ell)}, B_\ell)$ is bounded by $C' \log(1/\delta)^2$. □

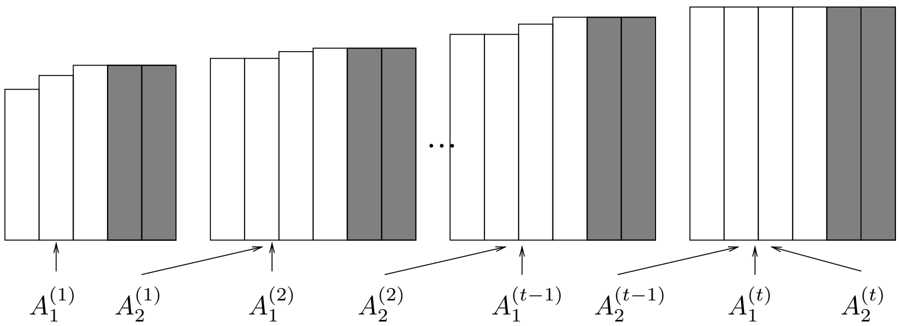

Figure 7: The item sets $A_1^{(\ell)}, A_2^{(\ell)}$ are placed via shifting into the blocks B_ℓ for $\ell = 1, \ldots, t$.

In our algorithm we simply use $\delta = \alpha$ (where α is the accuracy in the solution of the LP). Notice that the above algorithm for bin packing with different bin capacities generates also a packing for $A_{wide}^{(\ell)} \cup A_{narrow}^{(\ell)}$ into $M + \lfloor \bar{C}' \log(1/\delta)^2 \rfloor$ bins or $M' + \lfloor \bar{C}' \log(1/\delta)^2 \rfloor$ bins for B_1 with $M' \leq M$ bins where $\bar{C}'$ is a constant. A subset $A_1^{(\ell)}$ of items fits into the block B_ℓ and the remaining set $A_2^{(\ell)} = A_{wide}^{(\ell)} \cup A_{narrow}^{(\ell)} \setminus A_1^{(\ell)}$ fits into $\lfloor \bar{C}' \log(1/\delta)^2 \rfloor$ bins of size $c_{max}^{(\ell)}$. Using the shifting technique (see Figure 7) we can prove the following result.

Lemma 2.9 *For each $\ell = 1, \ldots, t-2$, we can select a subset $X_{\ell+1} \subseteq A_2^{(\ell)} \cup A_1^{(\ell+1)}$ with profit at least $(1 - \bar{C}'\delta)profit(A_2^{(\ell)} \cup A_1^{(\ell+1)})$ that can be packed into block $B_{\ell+1}$.*

Proof: The total number of bins to pack $A_2^{(\ell)} \cup A_1^{(\ell+1)}$ is $\lfloor \bar{C}' \log(1/\delta)^2 \rfloor + \lceil 1/\delta \log(1/\delta)^2 \rceil$. Removing $x = \lfloor \bar{C}' \log(1/\delta)^2 \rfloor$ bins generates a solution for $B_{\ell+1}$, since $A_2^{(\ell)}$ is packed into bins of smaller capacities. We split now the bins into groups with x bins where the last group has less than or equal to x bins. The number of groups with exactly x bins is at least $\lfloor \frac{\lfloor \bar{C}' \log(1/\delta)^2 \rfloor + \lceil 1/\delta \log(1/\delta)^2 \rceil}{\lfloor \bar{C}' \log(1/\delta)^2 \rfloor} \rfloor \geq \lfloor \frac{1}{\delta \bar{C}'} \rfloor + 1$. One of these groups has profit at most $\frac{1}{\lfloor \frac{1}{\delta \bar{C}'} + 1 \rfloor} profit(A_1^{(\ell)} \cup A_2^{(\ell+1)})$. Using $\frac{1}{\lfloor a/b \rfloor + 1} \leq b/a$ for $a, b > 0$, the profit loss is at most $\delta \bar{C}' profit(A_1^{(\ell)} \cup A_2^{(\ell+1)})$. Removing this group of bins and the items gives an item set $X_{\ell+1}$ with profit at least $(1 - \bar{C}'\delta)profit(A_1^{(\ell)} \cup A_2^{(\ell+1)})$. □

For the last block B_t we use the property that all bins in this group have the same capacity. Therefore we can select a subset $X_t \subseteq A_2^{(t-1)} \cup A_1^{(t)} \cup A_2^{(t)}$ with profit at least $(1 - 2\bar{C}'\delta)profit(A_2^{(t-1)} \cup$

$A_1^{(t)} \cup A_2^{(t)}$) that can be packed into block B_t. For the first block B_1 we simply take $X_1 = A_1^{(1)}$. Notice that our solution satisfies $\sum_\ell \mathit{profit}(A^{(\ell)}_{wide} \cup A^{(\ell)}_{narrow}) \geq (1 - 5\delta) OPT(\mathcal{A}, \mathcal{B})$ using $\alpha = \delta$. Lemma 2.3 above implies that $\mathit{profit}(\bigcup X_\ell) \geq (1 - (5 + 2\bar{C}')\delta) OPT(\mathcal{A}, \mathcal{B})$. Using $\delta \leq \frac{1}{5+2\bar{C}'}\epsilon$ we obtain a solution with profit at least $(1-\epsilon) OPT(\mathcal{A}, \mathcal{B})$. In our algorithm we set $\delta = 1/\lceil \frac{5+2\bar{C}'}{\epsilon} \rceil$ (to get $\delta \leq \frac{1}{5+2\bar{C}'}\epsilon$ and $1/\delta$ is integral).

Theorem 2.2 *For any $\epsilon \leq 1$ and $\delta = 1/\lceil \frac{5+2\bar{C}'}{\epsilon} \rceil$, our algorithm generates a feasible solution with profit at least $(1-\epsilon) OPT(\mathcal{A}, \mathcal{B})$ for MKP instances with $M \geq \lceil 1/\delta \log(1/\delta)^2 \rceil$ bins with the same largest capacity. The running time is polynomial in n and $1/\epsilon$.*

2.3 Bin packing with small objective value $OPT(I)$

In this subsection we describe an approximation algorithm for bin packing with a small number $OPT(I) \leq \gamma$ of bins, where γ is a constant which is bounded by a polynomial in $1/\delta$. In the following we suppose first that $\gamma \leq 1/\delta^3$ and that $1/\delta$ is integral. In addition we give an AEPTAS for bin packing that uses $\max\{OPT(I) + 1, (1+\epsilon) OPT(I)\}$ bins and runs in time $2^{O(1/\epsilon \log(1/\epsilon)^4)} + poly(n, 1/\epsilon)$.

First we guess the optimum value $k \in \{1, \ldots, \gamma\}$. Our algorithm below tries to compute a packing into k or $k+1$ bins. Using binary search with $O(\log(\gamma))$ iterations we are able to compute a packing into $OPT(I) + 1$ bins. Let us divide the instance I now into three groups:

$$\begin{aligned} I_{large} &= \{a_i \in I | \mathit{size}(a_i) > \tfrac{1}{2K} \tfrac{1}{\log(1/\delta)}\}, \\ I_{medium} &= \{a_i \in I | \mathit{size}(a_i) \in (\delta^4, \tfrac{1}{2K} \tfrac{1}{\log(1/\delta)}]\}, \\ I_{small} &= \{a_i \in I | \mathit{size}(a_i) \leq \delta^4\}, \end{aligned}$$

where K is a constant specified later. In the first phase of our algorithm we consider the large items. Since each bin has at most $\lfloor 2K \log(1/\delta) \rfloor$ large items and $OPT(I) \leq \gamma$, the total number of all large items in I is at most $\gamma \lfloor 2K \log(1/\delta) \rfloor$. Suppose that $I_{large} = \{a_1, \ldots, a_\ell\}$ where $\ell \leq \gamma \lfloor 2K \log(1/\delta) \rfloor$. We can assign large items to bins via a mapping $f : \{1, \ldots, \ell\} \to \{1, \ldots, k\}$ where $k \leq \gamma$ is the guessed value above. A mapping f is feasible, if and only if $\sum_{i: f(i) = j} \mathit{size}(a_i) \leq 1$ for all $j = 1, \ldots, k$. The total number of feasible mappings or assignments of large items to bins is at most $(\gamma)^{\gamma \lfloor 2K \log(1/\delta) \rfloor} = 2^{O(\gamma \log(\gamma) \log(1/\delta))}$. For $\gamma = poly(1/\delta)$, we have $2^{O(\gamma \log(1/\delta)^2)}$ assignments. Each feasible mapping f generates a pre-assignment $\mathit{preass}(b_j) \in [0, 1]$ for the bins $b_j \in \{b_1, \ldots, b_k\}$; i.e. $\mathit{preass}(b_j) = \sum_{i: f(i) = j} \mathit{size}(a_i) \leq 1$. Notice that one of the $2^{O(\gamma \log(\gamma) \log(1/\delta))}$ mappings corresponds to a packing of the large items in an optimum solution (if there exists a packing of I into k bins).

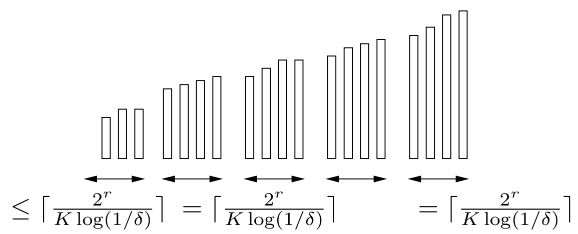

Figure 8: The instance I_r for interval $(2^{-(r+1)}, 2^{-r}]$.

In the second phase we use a geometric rounding for the medium items. This method was introduced by Karmarkar and Karp [13] for the original bin packing problem and all large items. Here, let

I_r be the set of all items from I_{medium} whose sizes lie in $(2^{-(r+1)}, 2^{-r}]$ where $2^r > K\log(1/\delta)$ or equivalently $\frac{1}{K}\frac{1}{\log(1/\delta)} > \frac{1}{2^r}$ (see Figure 8 for an example). Let $r(0)$ be the smallest integer r such that $2^r > K\log(1/\delta)$. Then, $2^{r(0)-1} \leq K\log(1/\delta)$ and $2^{r(0)} > K\log(1/\delta)$. This implies that the interval with the smallest index contains items of size in $(1/2^{r(0)+1}, 1/2^{r(0)}]$ and that $\frac{1}{2K}\frac{1}{\log(1/\delta)} \in (1/2^{r(0)+1}, 1/2^{r(0)}]$. For each $r \geq r(0)$ let J_r and J'_r be the instances obtained by applying linear grouping with parameter or group size $g = \lceil \frac{2^r}{K\log(1/\delta)} \rceil$ to I_r. To do this we divide each instance I_r into groups $G_{r,1}, G_{r,2}, \ldots, G_{r,q_r}$ such that $G_{r,1}$ contains the g largest items in I_r, $G_{r,2}$ contains the next g largest items and so on. Each group of items is rounded up to the largest size within the group (see Figure 9 for an illustration of the linear grouping). Let $G'_{r,i}$ be the multi-set of items obtained by rounding the size of each item in $G_{r,i}$. Then, $J_r = \bigcup_{i\geq 2} G'_{r,i}$ and $J'_r = G'_{r,1}$.

Furthermore, let $J = \bigcup J_r$ and $J' = \bigcup J'_r$. Then, $J_r \leq I_r \leq J_r \cup J'_r$ where $\leq$ is the partial order on bin packing instances as described in the proof of Theorem 2.1. Furthermore, J'_r consists of one group of items with the largest medium items in $(2^{-(r+1)}, 2^{-r}]$. The cardinality of each group (with exception of maybe the smallest group in I_r) is equal to $\lceil \frac{2^r}{K\log(1/\delta)} \rceil$.

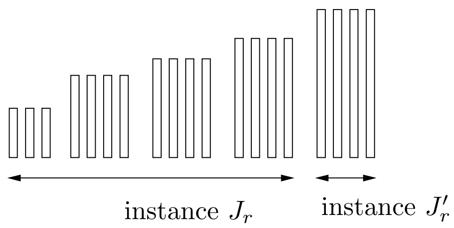

Figure 9: The rounded instance J_r and J'_r for interval $(2^{-(r+1)}, 2^{-r}]$.

Lemma 2.10 *For $\delta \leq 1/2$ and $K = 8$, we have size$(J') \leq 7/8$.*

Proof: Each non-empty set J'_r contains at most $\lceil \frac{2^r}{K\log(1/\delta)} \rceil$ items each of size at most $1/2^r$. Hence $size(J'_r) \leq (\frac{2^r}{K\log(1/\delta)}+1)\frac{1}{2^r} = \frac{1}{K\log(1/\delta)}+\frac{1}{2^r}$. This implies that the total size $size(J') = \sum_{r\geq r(0)} size(J'_r) \leq \sum_{r\geq r(0)}(\frac{1}{K\log(1/\delta)}+\frac{1}{2^r})$. Let $r(1)$ be the index with $\delta^4 \in (2^{-(r(1)+1)}, 2^{-r(1)}]$. Then, the number of indices $r \in \{r(0), \ldots, r(1)\}$ is equal to the number of intervals $(2^{-(r+1)}, 2^{-r}]$ which may contain a medium item. Since $\frac{1}{K\log(1/\delta)} > 1/2^{r(0)}$, $\delta \leq 1/2$ and $K \geq 2$, we have $r(0) \geq 2$. Therefore, the number of such intervals $r(1) - r(0) + 1$ can be bounded by $\lceil \log(1/\delta^4) \rceil \leq 4\log(1/\delta) + 1$. Using $\frac{1}{2^{r(0)}} < \frac{1}{K\log(1/\delta)}$ and $\sum_{r=0}^{\infty} 1/2^r \leq 2$, the total size $size(J')$ is at most $(4\log(1/\delta)+1)\frac{1}{K\log(1/\delta)} + \sum_{r\geq r(0)} \frac{1}{2^r} < \frac{4\log(1/\delta)+3}{K\log(1/\delta)}$. Therefore, for $\delta \leq 1/2$ and $K = 8$ we obtain $\log(1/\delta) \geq 1$ and $size(J') < \frac{4\log(1/\delta)+3}{8\log(1/\delta)} \leq \frac{7\log(1/\delta)}{8\log(1/\delta)} = 7/8$. □

The lemma above implies that $OPT(J') = 1$, since these items fit into one bin. Since $\bigcup_{r\geq r(0)} J_r \leq I_{medium} \leq \bigcup_{r\geq r(0)}(J_r \cup J'_r)$ and $J' = \bigcup_{r\geq r(0)} J'_r$, we obtain:

Lemma 2.11

$$OPT(I_{large} \cup \bigcup_{r\geq r(0)} J_r \cup I_{small}) \leq OPT(I_{large} \cup I_{medium} \cup I_{small}) \leq OPT(I_{large} \cup \bigcup_{r\geq r(0)} J_r \cup I_{small}) + 1.$$

Lemma 2.12 *The number of rounded sizes for medium items is at most $O(\gamma \log(1/\delta))$.*

Proof: Let $n(I_r)$ be the number of medium items in I_r, and let $m(I_r)$ be the number of groups (or rounded sizes) generated by the linear grouping for I_r. Then, $size(I_r) \geq \frac{1}{2^{r+1}} n(I_r) \geq \frac{1}{2^{r+1}}[(m(I_r) - 1)\left\lceil \frac{2^r}{K\log(1/\delta)} \right\rceil]$. Notice that one group may have less than $\lceil \frac{2^r}{K\log(1/\delta)} \rceil$ items. This implies that $m(I_r) - 1 \leq \frac{2^{r+1} size(I_r)}{\lceil \frac{2^r}{K\log(1/\delta)} \rceil}$. Using $\lceil \frac{2^r}{K\log(1/\delta)} \rceil \geq \frac{2^r}{K\log(1/\delta)}$, $m(I_r) \leq 2K\log(1/\delta) size(I_r) + 1$. Since the number of intervals for the medium items is at most $4\log(1/\delta) + 1$, the total number of rounded medium sizes $\sum_{r \geq r(0)} m(I_r) \leq \sum_{r \geq r(0)} (2K\log(1/\delta) size(I_r) + 1) \leq 2K\log(1/\delta) \sum_{r \geq r(0)} size(I_r) + 4\log(1/\delta) + 1$. Since all medium items fit into γ bins, $size(I_{medium}) = \sum_{r \geq r(0)} size(I_r) \leq \gamma$ and $\sum_{r \geq r(0)} m(I_r) \leq O(\gamma\log(1/\delta))$. This implies that we have at most $O(\gamma\log(1/\delta))$ rounded medium item sizes. □

For $OPT(I) \leq 1/\delta$, the number of rounded medium item sizes is at most $O(1/\delta\log(1/\delta))$.

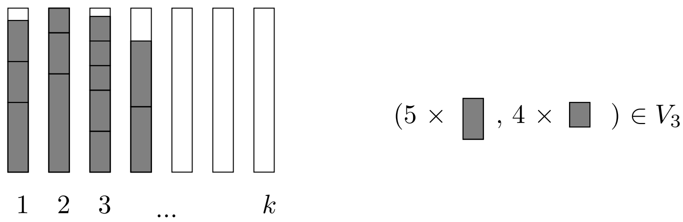

Figure 10: The dynamic program for rounded medium items $J = \cup_j J_r$

Now we describe the third phase of our algorithm. The rounded medium item sizes lie in the interval $[\delta^4, \frac{1}{2K\log(1/\delta)}]$ and there at most $R \leq O(\gamma\log(1/\delta))$ many different rounded item sizes. For each $j = 1, \ldots, R$ let k_j be the number of items for each rounded item size x_j. Since $x_j \geq \delta^4$ and $OPT(I) \leq \gamma$, $k_j \leq \gamma/\delta^4$ for each item size x_j. To describe a packing for one bin b we use a mapping $p : \{1, \ldots, R\} \rightarrow \{1, \ldots, 1/\delta^4\}$ where $p(j)$ gives the number of items of size x_j in b. A mapping p is feasible, if and only if $\sum_j p(j)x_j + preass(b) \leq 1$ where $preass(b)$ is the total size of large items assigned to b in the first phase of the algorithm. The total number of feasible mappings for one bin is at most $(1/\delta^4)^{O(\gamma\log(1/\delta))} = 2^{O(\gamma\log(1/\delta)^2)}$. Using a dynamic program we go over the bins from b_1 up to b_k where $k \leq \gamma$ is the guessed number of bins. For each $A = 1, \ldots, k$, we compute a set V_A of vectors $(a_1, \ldots, a_R)$ where a_j gives the number of items of size x_j used for the bins $b_1, \ldots, b_A$ (see also Figure 10). The cardinality of each set V_A is at most $(\gamma/\delta^4)^{O(\gamma\log(1/\delta))} = 2^{O(\gamma\log(\gamma/\delta)\log(1/\delta))}$. For $\gamma \leq 1/\delta$, the cardinality is bounded by $2^{O(1/\delta\log(1/\delta)^2)}$. The update step from one bin to the next can be implemented in time $2^{O(\gamma\log(\gamma/\delta)\log(1/\delta))} \cdot 2^{O(\gamma\log(1/\delta)^2)} \cdot poly(\gamma, 1/\delta) \leq 2^{O(\gamma\log(\gamma/\delta)\log(1/\delta))}$. Again for $\gamma \leq 1/\delta$, we obtain a running time $2^{O(1/\delta\log(1/\delta)^2)}$.

If there is a solution for our bin packing instance I into k bins, then the set V_k contains a vector $(n_1, \ldots, n_R)$ that corresponds to the number of rounded medium item sizes in $\bigcup_{r \geq r(0)} J_r$. Notice that the other set $\bigcup_{r \geq r(0)} J'_r$ will be placed into an additional bin b_{k+1}. We can also compute a packing of the medium items into the bins as follows. First, we compute all vector sets V_A for $A = 1, \ldots, k$. If for two vectors $a = (a_1, \ldots, a_k) \in V_A$ and $a' = (a'_1, \ldots, a'_k) \in V_{A+1}$ the medium items given by the difference $a' - a$ and the preassigned large items fit into bin b_{A+1}, we store the corresponding pair (a, a') in a set S_{A+1}. By using a directed acyclic graph $D = (V, E)$ with vertex set $V = \{[a, A] | a \in V_A, A = 1, \ldots, k\}$ and $E = \{([a, A], [a', A+1]) | (a, a') \in S_{A+1}, A = 1, \ldots, k-1\}$, we may compute a feasible packing of large and medium rounded items into the bins $b_1, \ldots, b_k$. This can be done via depth first search starting with the vector $(n_1, \ldots, n_R) \in V_k$ that corresponds to the number of rounded medium item sizes. The algorithm to compute the directed acyclic graph and the backtracking algorithm can be implemented in time $2^{O(\gamma\log(\gamma/\delta)\log(1/\delta))}$.

In the last phase of our algorithm we add the small items via a greedy algorithm to the bins. Consider a process which starts with a given packing of large and medium items into the bins $b_1, \ldots, b_{k+1}$. We insert a small item into the first bin in which the item fits, and open a new bin if necessary.

Lemma 2.13 *If $OPT(I) = k \leq 1/\delta^3$, then our algorithm packs all items into at most $k+1$ bins.*

Proof: Assume by contradiction that we use more than $k+1$ bins for the small items. In this case, the total size of the items packed into the bins is larger than $(k+1)(1-\delta^4) = k+1-\delta^4(k+1)$. Since $\delta^4(k+1) \leq \delta^4(1/\delta^3+1) = \delta + \delta^4 < 1$ for each $\delta \leq 1/2$, the total size of the original items and, hence, $OPT(I)$ is larger than k. □

The algorithm for bin packing works as follows. Given an instance I with n items $a_1, \ldots, a_n$ and $size(a_i) \in (0,1]$ for $i = 1, \ldots, n$ and a constant upper bound $OPT(I) \leq \gamma$, we test whether I can be packed into $k \in \{1, \ldots, \gamma\}$ bins where the smallest number k of bins where I can be packed into is obtained via binary search. We suppose that $\gamma \leq 1/\delta^3$ where $1/\delta$ is an integral constant and $size(I) \leq k$ (as otherwise $OPT(I) > k$). To test whether I can be packed into k bins we use the following algorithm.

(1) Set $K = 8$ and divide the instance I into three groups I_{large}, I_{medium}, and I_{small}.

(2) Assign the large items to the k bins considering all different feasible pre-assignments.

(3) Use geometric rounding on the sizes of the medium items; for each interval $(2^{-(r+1)}, 2^{-r}]$ apply linear grouping with group size $\lceil \frac{2^r}{K \log(1/\delta)} \rceil$ to the item set I_r and compute rounded item sets J_r and J'_r.

(4) For each pre-assignment apply the dynamic program to assign the medium items in J_r to the bins $b_1, \ldots, b_k$, and place the set $\bigcup_j J'_r$ into an additional bin b_{k+1} (if possible).

(5) If we are able to pack all large and medium items in the $k+1$ bins for at least one pre-assignment then assign the small items via a greedy algorithm also to the bins $b_1, \ldots, b_{k+1}$ and obtain $OPT(I) \leq k+1$; otherwise $OPT(I) > k$.

We can generalize the algorithm also to larger number of bins where γ is bounded by a polynomial function $poly(1/\delta)$. For $\gamma \leq 1/\delta^\ell$ with constant ℓ, we simply use $\delta^{\ell+1}$ as upper bound for the small items. Then the number of intervals with medium items is bounded by $(\ell+1)\log(1/\delta)+1$. This implies that the number of rounded medium sizes is at most $O(\gamma \log(1/\delta) + (\ell+1)\log(1/\delta)) = O(\gamma \log(1/\delta))$. The total number k_j of items with rounded medium size x_j is at most $\gamma/\delta^{\ell+1}$ and the number of feasible assignments to one bin is bounded by $(1/\delta^{\ell+1})^{O(\gamma \log(1/\delta))} = 2^{O(\gamma \log(1/\delta)^2)}$. Therefore, the running time of the dynamic program can be bounded as above. Furthermore, all items including the small ones can be packed into $k+1$ bins for $OPT(I) \leq k$ similarly as discussed above.

Theorem 2.3 *For $OPT(I) \leq \gamma \leq poly(1/\delta)$ and $\delta \leq 1/2$, there is an approximation algorithm for bin packing which computes a packing of the items in I into at most $OPT(I)+1$ bins and runs in time $2^{O(\gamma \log(\gamma/\delta) \log(1/\delta))} + n$.*

For $OPT(I) \leq \gamma \leq 1/\epsilon$, the running time of the algorithm above is bounded by $2^{O(1/\epsilon \log(1/\epsilon)^2)} + n$. Note that we can generalize the algorithm above also for bin packing with different sizes. With the same running time we obtain a packing into at most $OPT(I)+1$ bins where the additional bin has size $c_{max} = \max_i c(b_i)$.

Notice the approximation algorithms by Karmarkar and Karp [13] generate solutions for bin packing with value $(1+\epsilon)OPT(I) + \frac{1}{2\epsilon^2} + 3$ and value $OPT(I) + O(\log(OPT(I))^2)$, but run in time polynomial in n and $1/\epsilon$. On the other hand, the additive constants are larger in these algorithms. For $OPT(I) \geq 8/\epsilon^3$, the algorithm by Karmarkar and Karp [13] with accuracy $\epsilon/2$ generates a packing with at most $(1+\epsilon/2)OPT(I) + \frac{2}{\epsilon^2} + 3 \leq (1+\epsilon)OPT(I)$ for each $\epsilon \leq 2\sqrt{1/6}$. Our above algorithm for $OPT(I) \leq 8/\epsilon^3$ computes a solution with $OPT(I)+1$ bins for each $\epsilon \leq 1/16$. Furthermore, for $OPT(I) \in [C/\epsilon \log(8/\epsilon^3)^2, 8/\epsilon^3]$, the algorithm by Karmarkar and Karp [13] generates also a packing into $OPT(I) + C\log(OPT(I))^2 \leq (1+\epsilon)OPT(I)$ bins. This follows from $C\log(OPT(I))^2 \leq C\log(8/\epsilon^3)^2 \leq \epsilon OPT(I)$. Our algorithm above for $OPT(I) \leq O(1/\epsilon \log(1/\epsilon)^2)$ runs in time $2^{O(1/\epsilon \log(1/\epsilon)^4)} + n$. This discussion implies the following result.

Theorem 2.4 *There is an approximation algorithm for bin packing which computes a packing of the items in I into at most* $\max\{OPT(I)+1, (1+\epsilon)OPT(I)\}$ *bins and runs in time* $2^{O(1/\epsilon \log(1/\epsilon)^4)} + poly(n, 1/\epsilon)$.

3 Instances of MKP with a Constant Number of Bins

Recently, we have proposed an approximation algorithm for MKP for an instance $(\mathcal{A}, \mathcal{B})$ with γ bins that produces a packing into γ bins with profit at least $(1-\epsilon)OPT(\mathcal{A}, \mathcal{B})$ and has a running time $2^{O((\gamma/\epsilon)\log(\gamma/\epsilon))}n$ [12]. For $\gamma \leq 1/\epsilon \log(1/\epsilon)^2$, this gives a running time $2^{O(1/\epsilon^2 \log(1/\epsilon)^3)}n$. In this section, we suppose that $\gamma \leq \lceil 1/\delta \log(1/\delta)^2 \rceil$ where $\delta = \Theta(\epsilon)$ is a constant that depends on ϵ and $1/\delta$ is integral (δ will be specified later). We show how to improve the running time above to $2^{O(1/\epsilon \log(1/\epsilon)^4)}n$. Let $APP(\mathcal{A}, \mathcal{B})$ be an approximate value for MKP obtained by the greedy algorithm with accuracy $\epsilon'/2$ [4] where $\epsilon' \leq 1$ is specified later; i.e. $APP(\mathcal{A}, \mathcal{B}) \geq (1/2 - \epsilon'/4)OPT(\mathcal{A}, \mathcal{B})$. Notice that $2(1+\epsilon')APP(\mathcal{A}, \mathcal{B}) \geq OPT(\mathcal{A}, \mathcal{B})$. In the following we give an outline of our algorithm for a constant number of bins.

(1) Reduce the number of high and medium profit items, guess high profit items and medium profit items of large size for $\mathcal{B}$ and guess an assignment for these items (corresponding to an optimum solution).

(2) Guess rounded medium sizes for medium profit items and number of items with rounded sizes (corresponding to an optimum solution).

(3) Select medium profit items corresponding to the guesses above and pack these items into $\mathcal{B}$ plus one additional bin of small size.

(4) Select small profit items for the remaining space via a fractional knapsack algorithm, distribute the selected items to the bins, and eliminate items which are not completely inside a bin.

(5) Apply a shifting argument to eliminate the additional bin.

The main difficulty lies in the last step, if there are only few bins in the instance. To apply the shifting argument, the maximum medium size has to be defined depending on the remaining total capacity after assigning the high profit items to the bins.

3.1 Largest bins and high profit items

In the first phase we place items with high profit, larger than $2\rho(1+\epsilon')APP(\mathcal{A},\mathcal{B}) \geq \rho OPT(\mathcal{A},\mathcal{B})$ into the γ bins in $\mathcal{B}$, where $\rho = \Theta(\delta)$ is also a constant specified later. Notice that an optimum solution can have at most $1/\rho = O(1/\delta)$ items with high profit. Using the following Lemma we can reduce the number of high profit items in our instance.

Lemma 3.1 *[12] There is a set CA_h of items of high profit in I with $|CA_h| \leq O(1/\delta^2 \log(1/\delta^2)) =$ poly$(1/\delta)$ such that an optimum solution which selects only high profit items from CA_h has profit at least $(1-3\epsilon')$ $OPT(\mathcal{A},\mathcal{B})$ for $\epsilon' \leq 1/2$.*

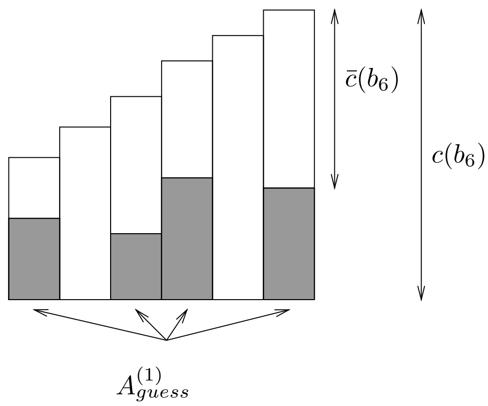

Figure 11: The pre-assignment of high profit items $A^{(1)}_{guess}$ into the bins and modified capacities $\bar{c}(b_i)$.

Since there are at most $1/\rho$ items with high profit in any optimum solution and using the above Lemma, we can guess the high profit items out of the candidate set CA_h. The number of choices is at most $(|CA_h|+1)^{1/\rho} = 2^{O(1/\delta \log(1/\delta))}$ using $|CA_h|+1 \leq poly(1/\delta)$. For each feasible choice, we try to pack the chosen candidates into the $\gamma \leq \lceil 1/\delta \log(1/\delta)^2 \rceil$ bins. This can be done via an assignment from candidates to bins. The number of these assignments is at most $(\gamma)^{1/\rho} = 2^{O(1/\delta \log(1/\delta))}$. An assignment is feasible if the assigned candidates fit into the corresponding bins. The total number of guesses in this phase is bounded by $2^{O(1/\delta \log(1/\delta))}$. Let $A^{(1)}_{guess}$ be the chosen candidates and let $Area_{Rem} = \sum_{i=1}^{\gamma} c(b_i) - size(A^{(1)}_{guess})$ be the remaining space. Let $\bar{c}(b_i)$ be the capacity of b_i after the placement of the high profit items. For an illustration of a pre-assignment of high profit items into few bins and modified bin capacities we refer to Figure 11. Suppose for simplicity that the largest capacity $\bar{c}(b_i)$ among the bins is 1; i.e. $\max_i \bar{c}(b_i) = 1$ (otherwise we scale the sizes of the items). This implies that $Area_{Rem} \leq \gamma \leq \lceil 1/\delta \log(1/\delta)^2 \rceil$.

3.2 Medium profit items

In the second phase we consider items with medium profit $profit(a_i) \in [2(\rho/\gamma)(1+\epsilon')APP(\mathcal{A},\mathcal{B}), 2\rho(1+\epsilon')APP(\mathcal{A},\mathcal{B}))$. Since $2(\rho/\gamma)(1+\epsilon)APP(\mathcal{A},\mathcal{B}) \geq (\rho/\gamma)OPT(\mathcal{A},\mathcal{B})$, any feasible solution can have at most $\gamma/\rho = \Theta(\gamma/\delta)$ many items with medium or high profit using $1/\rho = \Theta(1/\delta)$. Using the same arguments as for the high profit items we obtain:

Lemma 3.2 *[12] There is a set CA_m of items in I with medium profit with $|CA_m| \leq O(\gamma/\delta^2 \log(\gamma/\delta))$ such that the profit loss of an optimum solution which selects only medium profit items from CA_m is at most $3\epsilon' OPT(\mathcal{A},\mathcal{B})$ for $\epsilon' \leq 1/2$.*

In our case $|CA_m| \leq O(1/\delta^3 \log(1/\delta)^3)$. Note that any solution can have at most $\gamma/\rho = \Theta(\gamma/\delta)$ medium profit items. In the next phase of the algorithm, we assign medium profit items to the bins. Depending on the sizes of these items we do the following steps.

Step A: Consider the medium profit items with large size $size(a_i) \in (\frac{\delta Area_{Rem}}{2K\log(1/\delta)^3}, 1]$ (where K is a constant specified later). Then, there are at most $\lfloor 2K/\delta \log(1/\delta)^3 \rfloor$ many items of this form in the bins. Next, we guess the medium profit items of large size for the γ bins. This can be done via a guessing step to select the candidates. Afterwards, we assign the chosen candidates to the bins (if possible). Let $A^{(2)}_{guess}$ be the chosen candidate set. The number of choices and assignments using $K = O(1)$ is bounded by $(|CA_m|+1)^{\lfloor 2K/\delta \log(1/\delta)^3 \rfloor} = 2^{O(1/\delta \log(1/\delta)^4)}$ and $(\gamma)^{\lfloor 2K/\delta \log(1/\delta)^3 \rfloor} = 2^{O(1/\delta \log(1/\delta)^4)}$, respectively.

Step B: Now we consider medium profit items with size smaller than or equal to $\frac{\delta Area_{Rem}}{2K\log(1/\delta)^3}$. The main idea is to round the medium sizes $size(a_i) \in [\delta^6 Area_{Rem}, \frac{\delta Area_{Rem}}{2K\log(1/\delta)^3}]$ corresponding to the optimum solution using linear grouping over sizes $(2^{-(r+1)}, 2^{-r}]$ and to reduce the number of different medium sizes in the instance (similar to bin packing).

Let I_r be the set of all medium items whose sizes lie in $(2^{-(r+1)}, 2^{-r}]$ where $2^r > \frac{K\log(1/\delta)^3}{\delta Area_{Rem}}$. For each r let J_r and J'_r be the instances obtained by applying linear grouping with group size $g = \lceil \frac{2^r \delta Area_{Rem}}{K\log(1/\delta)^3} \rceil$. To apply linear grouping divide each set I_r into groups $G_{r,1}, \ldots, G_{r,q_r}$ such that $G_{r,1}$ contains the g largest items in I_r, $G_{r,2}$ contains the next g largest items and so on. Each group is rounded up to the largest size within the group. Let $G'_{r,i}$ be the multi-set of items obtained by rounding the size of each item in $G_{r,i}$. Then, $J_r = \bigcup_{i\geq 2} G'_{r,i}$ and $J'_r = G'_{r,1}$.

Lemma 3.3 *Let $Opt_{medium} \cup Opt_{small}$ be a set of (medium profit) items with medium and small size that are packed into $\mathcal{B}$. Then the rounded medium items and the small items fit into $\mathcal{B}$ plus one additional bin of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ for $K = 56$ and $\delta \leq 1/10$. Furthermore, the number of rounded medium sizes is bounded by $O(1/\delta \log(1/\delta)^3)$ and a packing of these items into the bins can be computed in time $2^{O(1/\delta \log(1/\delta)^4)}$.*

Proof: Let $J = \bigcup J_r$ and $J' = \bigcup J'_r$ be the set of rounded items for the medium items in $I_r \subseteq Opt_{medium}$. Then, $J_r \leq I_r \leq J_r \cup J'_r$ where $\leq$ is the partial order on bin packing instances. Using the rounding we get $\bigcup J_r \leq \bigcup I_r \leq \bigcup (J_r \cup J'_r)$. Therefore the items in J_r fit into $\mathcal{B}$. Each non-empty set I_r contains at most $\lceil \frac{2^r \delta Area_{Rem}}{K\log(1/\delta)^3} \rceil$ items each of size at most $1/2^r$. Hence, $size(J'_r) \leq \frac{\delta Area_{Rem}}{K\log(1/\delta)^3} + \frac{1}{2^r}$. This implies that $size(J') \leq \sum_{r\geq r(0)} (\frac{\delta Area_{Rem}}{K\log(1/\delta)^3} + \frac{1}{2^r})$, where $r(0)$ is the smallest r such that $2^r > \frac{K\log(1/\delta)^3}{\delta Area_{Rem}}$. Notice that $\frac{\delta Area_{Rem}}{2K\log(1/\delta)^3} \in (\frac{1}{2^{r(0)+1}}, \frac{1}{2^{r(0)}}]$.

Since the number of sets J'_r with at least one medium item is bounded by $\lceil \log(1/\delta^5) \rceil + 1 \leq \lfloor 6\log(1/\delta) \rfloor$ for $\delta \leq 1/4$, the total size $size(J') \leq \frac{6\log(1/\delta)\delta Area_{Rem}}{K\log(1/\delta)^3} + \sum_{r\geq r(0)} \frac{1}{2^r} \leq \frac{7\delta Area_{Rem}}{K\log(1/\delta)^2} \leq \frac{\delta Area_{Rem}}{8\log(1/\delta)^2}$ for $K = 56$. The total number of medium profit items in any feasible solution is at most $\frac{\gamma}{\rho} \leq \frac{\log(1/\delta)^2}{\delta^2}$ using $\delta \leq \rho$ and $\gamma \leq 1/\delta \log(1/\delta)^2$. Therefore, all medium profit items of size $\leq \delta^6 Area_{Rem}$ in the optimum solution have total size at most $\frac{\delta^6 \log(1/\delta)^2 Area_{Rem}}{\delta^2} \leq \delta^4 \log(1/\delta)^2 Area_{Rem} \leq \frac{\delta Area_{Rem}}{8\log(1/\delta)^2}$ for $\delta^3 \leq \frac{1}{8\log(1/\delta)^4}$ or $\delta \leq 1/10$. For $\delta < 1/10$, all of them together with the medium size items in J' fit into the additional bin of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$.

To prove the bound for the number of rounded sizes, compare the number $n(I_r)$ of medium items in I_r with the number $m(I_r)$ of groups or rounded sizes generated by the linear grouping for I_r. Then, $size(I_r) \geq \frac{1}{2^{r+1}}(m(I_r)-1)\lceil \frac{2^r \delta Area_{Rem}}{K\log(1/\delta)^3} \rceil$. This implies that $m(I_r) \leq \lfloor \frac{2K\log(1/\delta)^3 size(I_r)}{\delta Area_{Rem}} + 1 \rfloor$. Since the number of intervals for the medium items is at most $\lfloor 6\log(1/\delta) \rfloor$, the total number of rounded medium sizes $\sum_r m(I_r) \leq \lfloor 2K\log(1/\delta)^3 \sum_r \frac{size(I_r)}{\delta Area_{Rem}} + 6\log(1/\delta) \rfloor$. Since all medium items in $\bigcup I_r$ fit into the bins in $\mathcal{B}$, $\sum_r size(I_r) \leq Area_{Rem}$. This implies that we have at most $\lfloor (2K+6)/\delta \log(1/\delta)^3 \rfloor$ different medium sizes.

In the next step we describe how the rounded medium profit items can be packed into γ bins. Notice that the capacities $\bar{c}(b_i)$ of the bins are different in general and that $\sum_i \bar{c}(b_i) = Area_{Rem}$. We use sets V_A of vectors $(a_1, \ldots, a_{med})$ to store how many rounded medium items are packed into the first A bins. Each solution contains at most γ/ρ many items with medium profit. Therefore, each bin contains at most $\gamma/\rho = \Theta(\gamma/\delta)$ items. Furthermore, note that the high profit items are preassigned in the first phase of our algorithm and are not counted in the capacities $\bar{c}(b_i)$. On the other hand, the medium profit items with large size have to be counted in this packing phase. A packing of medium size items into one bin can be described by a mapping $p : \{1, \ldots, med\} \rightarrow \{0, 1, \ldots, \gamma/\rho\}$ where $p(j)$ gives the number of items with the j.th rounded size. A mapping p for bin b_i is feasible if the corresponding rounded items plus the preassigned large items (with medium profit) fit into the bin b_i. The number of feasible mappings p is at most $(\gamma/\delta+1)^{med} = 2^{O(1/\delta \log(1/\delta)^4)}$ using $\gamma \leq \lceil 1/\delta \log(1/\delta)^2 \rceil$ and $med \leq O(1/\delta \log(1/\delta)^3)$. Using a dynamic program similar to bin packing, we go over the bins $b_1, \ldots, b_\gamma$ and compute the sets V_A of vectors for $A = 1, \ldots, \gamma$. The cardinality of each set V_A is bounded by $2^{O(1/\delta \log(1/\delta)^4)}$. Notice that it is not necessary to pack the largest groups of items among the intervals $(2^{-(r+1)}, 2^{-r}]$ into the γ bins. Here we have only to test whether the total size of these rounded items and the items with small size $\leq \delta^6 Area_{Rem}$ is at most $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$; if the total size is too large then we discard the corresponding guess. The running time of the procedure including the time to compute the packing is again bounded by $2^{O(1/\delta \log(1/\delta)^4)}$. □

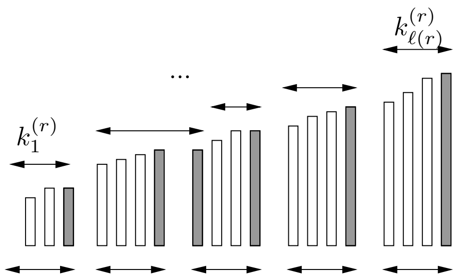

Figure 12: Guessing the rounded medium sizes $b_i^{(r)}$ and the numbers $k_i^{(r)}$ of items with size in $(b_{i-1}^{(r)}, b_i^{(r)}]$.

Guessing the rounded medium sizes. In our algorithm, we now guess the rounded medium sizes of the medium profit items that are placed in B (see Figure 12 for an illustration; the guessed sizes are indicated by dark shaded rectangles). The number *med* of medium sizes is bounded by $\lfloor C/\delta \log(1/\delta)^3 \rfloor$ where $C \leq 2K+6$ is a constant. Let *Opt* be an optimum set of items and $Opt_{medium} \subseteq Opt$ be a subset of medium profit items with medium size placed into $\mathcal{B}$.

Lemma 3.4 *Our algorithm guesses the rounded medium sizes $b_1^{(r)} < \ldots < b_{\ell(r)}^{(r)}$ of Opt_{medium} within each interval $(2^{-(r+1)}, 2^{-r}]$ and the numbers $k_i^{(r)}$ of items of size within $(b_{i-1}^{(r)}, b_i^{(r)}]$ which are placed into $\mathcal{B}$. The number of different guesses is bounded by $2^{O(1/\delta \log(1/\delta)^4)}$.*

Proof: By guessing *med* items in CA_m with medium size, we obtain for each interval $(2^{-(r+1)}, 2^{-r}]$ a sequence $s_1^{(r)} \leq \ldots \leq s_{\ell'(r)}^{(r)}$ of the largest rounded sizes among the groups. The number of choices is $2^{O(1/\delta \log(1/\delta)^4)}$. The number of items in each group (with exception of the group with the smallest items) is exactly $\lceil \frac{2^r \delta Area_{Rem}}{K \log(1/\delta)^3} \rceil$ (the value used in the bin packing approach). We also guess the cardinality x_r of the group with the smallest items for each interval. Notice that each $x_r \le 1/\delta^5$. Let $b_1^{(r)} < \ldots < b_{\ell(r)}^{(r)}$ be the subsequence of $s_1^{(r)}, \ldots, s_{\ell'(r)}^{(r)}$ with different sizes. Notice that some of the $s_i^{(r)}$ values might be equal, since the optimum solution of MKP could have a large number of items of the same size. Suppose that $b_0^{(r)} = 1/2^{r+1}$ for $r < r(1)$, $b_0^{(r(1))} = \delta^6 Area_{Rem}$ (the smallest medium size), $b_{\ell(r)+1}^{(r)} = 1/2^r$ for $r > r(0)$ and $b_{\ell(r(0))+1}^{(r(0))} = \frac{\delta Area_{Rem}}{2K\log(1/\delta)^3}$ (the largest medium size). For each interval $(b_{i-1}^{(r)}, b_i^{(r)}]$ with $i = 1, \ldots, \ell(r)+1$ we can compute (using the sequence $(s^{(r)})$ and the cardinalities of the groups) the total number $\bar{k}_i^{(r)}$ of medium items with medium profit and rounded value $b_i^{(r)}$ (see Figure 12).

In addition we can guess for each rounded medium size $b_{i-1}^{(r)}$ the number of items in $\mathcal{B}$ of this size that are rounded up to the next higher number $b_i^{(r)}$. For each rounded medium size $b_{i-1}^{(r)}$ there are at most $\lceil \frac{2^r \delta Area_{Rem}}{K\log(1/\delta)^3} \rceil - 1 \le O(1/\delta^5)$ items that are rounded up. Therefore, the number of guesses is at most $(\frac{1}{\delta^5})^{med} = 2^{O(1/\delta \log(1/\delta)^4)}$ using $med \le O(1/\delta \log(1/\delta)^3)$. Using this information we can compute the total number $k_i^{(r)}$ of items for $\mathcal{B}$ with size in each interval $(b_{i-1}^{(r)}, b_i^{(r)}]$ (see also Figure 12 for an illustration). □

If our instance does not have at least $k_i^{(r)}$ items with medium profit and size in $(b_{i-1}^{(r)}, b_i^{(r)}]$, then we discard the corresponding guess. Notice that each number $k_i^{(r)} \le \gamma/\rho$, since there are at most γ/ρ medium profit items in the optimum solution. Therefore, if $k_i^{(r)} > \gamma/\rho$ we also discard the choice. Notice that the algorithm above only selects the structure of the medium profit items, but not the items themselves.

3.3 Selection of medium and small profit items

First we select the medium profit items in accordance to the structure guessed above. For the items a_i with size $size(a_i) \in [\frac{\delta Area_{Rem}}{2K\log(1/\delta)^3}, 1]$ we have chosen a subset $A_{guess}^{(2)}$ to be packed in the γ bins. Using our guessing step, we should take $k_i^{(r)}$ medium profit items with size in $(b_{i-1}^{(r)}, b_i^{(r)}]$ for the γ bins. A greedy algorithm that takes $k_i^{(r)}$ items (that are in CA_m and have size in $(b_{i-1}^{(r)}, b_i^{(r)}]$) ordered by their profits generates the best possible solution. Let $\bar{K}_{medium}$ be the computed set of medium profit items of medium size. Furthermore, our algorithm selects the subset $\bar{K}_{small}$ of all small items with medium profit. If the total size of $\bar{K}_{small}$ is larger than $\frac{\delta Area_{Rem}}{8\log(1/\delta)^2}$, then there is a packing of a subset of these items with total size $\frac{\delta Area_{Rem}}{8\log(1/\delta)^2} + \delta^6 Area_{Rem}$ into one bin of capacity $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ with profit larger than $OPT(\mathcal{A}, \mathcal{B})$. But this is not possible, since one of the $\gamma \le 1/\delta \log(1/\delta)^2$ bins has capacity at least $\frac{\delta Area_{Rem}}{\log(1/\delta)^2}$. Let $Opt_{medium} \cup Opt_{small} \subseteq Opt$ be the subset of medium profit items (of medium and small size) placed into $\mathcal{B}$ (corresponding to the optimum solution with numbers $k_i^{(r)}$ for the subintervals $(b_{i-1}^{(r)}, b_i^{(r)}]$).

Lemma 3.5 *For the guess corresponding to the optimum numbers, the set $A_{guess}^{(3)} = \bar{K}_{medium} \cup \bar{K}_{small}$ can be packed into $\mathcal{B}$ plus one bin of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ and $profit(\bar{K}_{medium} \cup \bar{K}_{small}) \ge profit(Opt_{medium} \cup Opt_{small})$.*

Proof: For the guess corresponding to the optimum solution, our algorithm computes a subset $\bar{K}_{medium}$ of medium size items that can be packed into $\mathcal{B}$ plus one additional bin; the set K'_{medium} with the largest $\lceil \frac{2^r \delta Area_{Rem}}{K\log(1/\delta)^3} \rceil$ items among the intervals fits into the additional bin and the set $K_{medium} = \bar{K}_{medium} \setminus K'_{medium}$ fits into $\mathcal{B}$. Notice that all medium profit items in our instance of small size $\le \delta^6 Area_{Rem}$ have total size $\frac{\delta Area_{Rem}}{8\log(1/\delta)^2}$ for $\delta \le 1/10$ and can be placed into the additional bin. For $\delta < 1/10$ the subset $\bar{K}_{small}$ together with the set K'_{medium} fits into the additional bin as before. □

To choose the remaining items of small profit $\leq 2\rho/\gamma(1+\epsilon')APP(\mathcal{A},\mathcal{B})$, we use a classical fractional knapsack algorithm. We simply take all remaining items with small profit and set the capacity of the knapsack to $cap - size(A^{(2)}_{guess} \cup J_{medium})$ where $cap = \sum_{i=1}^{\gamma} \bar{c}(b_i)$ and J_{medium} is defined as below. For the chosen medium items packed into $\mathcal{B}$ we use the lower bound $size(J_{medium})$, where J_{medium} is a multiset with $k_i^{(r)}$ items with rounded size $b_i^{(r)}$ among all subintervals — without taking the largest group for each interval $(2^{-(r+1)}, 2^{-r}]$. Comparing with the set Opt_{medium} of medium sized items with medium profit packed into $\mathcal{B}$, the rounded size $size(J_{medium}) \leq size(Opt_{medium})$ for the corresponding guess. Moreover, our algorithm computes a subset $\bar{K}_{medium}$ for the guess above such that $size(Opt_{medium}) \leq size(\bar{K}_{medium})$. On the other hand, our algorithm packs only the subset K'_{medium} without the largest $\lceil \frac{2^r \delta Area_{Rem}}{K \log(1/\delta)^3} \rceil$ items among the intervals into $\mathcal{B}$ and $size(K'_{medium}) \leq size(J_{medium})$. Therefore, we can set the capacity $cap = \sum_{i=1}^{\gamma} \bar{c}(b_i) - size(A^{(2)}_{guess} \cup J_{medium})$. Let A_S be the computed set of items with at most one fractional item. Suppose for simplicity that the last item in A_S is fractional (if there is any). Afterwards we distribute the computed set A_S to the bins. By this process, we have to split at most $\gamma - 1$ items (one for each bin b_i with $i < \gamma$)) and have at most one fractional item in the last bin b_γ. Let $Split_S \subset A_S$ be the items which are not completely assigned to a bin. The total profit of $Split_S$ is at most $2\rho(1+\epsilon')OPT(\mathcal{A},\mathcal{B})$. Our algorithm generates in this way a packing for $\bigcup_{i=1}^{3} A^{(i)}_{guess} \cup (A_S \setminus Split_S)$, but uses an additional bin of small size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$. We consider the bins after the first pre-assignment phase of high profit items. The capacities of the bins are $\bar{c}(b_1), \ldots, \bar{c}(b_\gamma)$ and $max_i \bar{c}(b_i) = 1$.

Lemma 3.6 *The total area of the bins with small capacities* $\bar{c}(b_i) \leq \frac{\delta Area_{Rem}}{2\log(1/\delta)^2}$ *is at most* $(1/2)Area_{Rem}(1 + \delta/\log(1/\delta)^2)$. *Furthermore, there are at least* $1/\delta$ *pieces with size* $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ *that lie inside the bins for* $\delta \leq 1/4$.

Proof: Since we have at most $\lceil 1/\delta \log(1/\delta)^2 \rceil$ bins, the total area of bins with small capacities is at most $\lceil 1/\delta \log(1/\delta)^2 \rceil \frac{\delta Area_{Rem}}{2\log(1/\delta)^2} \leq (1/2)Area_{Rem}(1 + \delta/\log(1/\delta)^2)$. This implies that the total area of larger bins with capacities $\bar{c}(b_i) > \frac{\delta Area_{Rem}}{2\log(1/\delta)^2}$ is at least $(1/2)Area_{Rem}(1 - \delta/\log(1/\delta)^2)$. Splitting the total remaining area of large bins into pieces of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$, gives us at least $\left\lfloor \frac{Area_{Rem}(\frac{1}{2} - \frac{\delta}{2\log(1/\delta)^2})}{\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}} \right\rfloor = \lfloor 2/\delta \log(1/\delta)^2 - 2 \rfloor$ many pieces.

Next take each bin of large size $\bar{c}(b_i) > \frac{\delta Area_{Rem}}{2\log(1/\delta)^2}$ separately and split this bin into pieces of size exactly $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ with exception of maybe one piece of size smaller than $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$. Then, take all smaller pieces (one per bin) and split the remaining area again in pieces. The number of pieces of the second group is smaller than the number of bins; otherwise each bin contains an additional complete piece of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ and this contradicts to our construction. This implies that the number of these pieces is also at most $\lfloor 1/\delta \log(1/\delta)^2 \rfloor$.

Therefore, the first group contains at least $\lfloor 2/\delta \log(1/\delta)^2 \rfloor - \lfloor 1/\delta \log(1/\delta)^2 \rfloor - 2$ pieces. Since $\lfloor 2x \rfloor \geq 2\lfloor x \rfloor$, this number of pieces in the first group is at least $\lfloor 1/\delta \log(1/\delta)^2 \rfloor - 2 \geq 1/\delta$. The last inequality holds (using that $1/\delta$ is integral), whenever $1/\delta \log(1/\delta)^2 - 2 \geq 1/\delta$. For $\delta \leq 1/4$, $1/\delta \log(1/\delta)^2 - 2 \geq 4/\delta - 2 \geq 1/\delta$. By construction, the pieces of the first group lie completely inside the bins. □

The main idea now is to exchange one piece of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ with the additional bin that contains J' and the small items of medium profit (see Figure 13 for an illustration of the exchange step). The fractional profit of one piece is at most $\delta OPT(\mathcal{A},\mathcal{B})$. Removing all items that lie inside of the piece plus at most two items that lie partially in the piece generates a gap of size at least $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ inside of one bin. The profit loss is at most $\delta OPT(\mathcal{A},\mathcal{B})$ plus the profit of two medium profit items; in total at

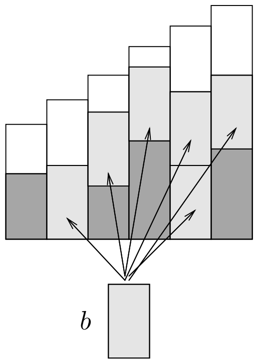

Figure 13: Shifting argument to eliminate the additional bin b.

most $(4\rho(1+\epsilon')+\delta)OPT(\mathcal{A},\mathcal{B})$. Finally, we take among all feasible guesses a solution with highest profit. In total we generate a solution of MKP with profit loss at most $(6(1+\epsilon')\rho+6\epsilon'+\delta)OPT(\mathcal{A},\mathcal{B}) \leq 14\delta OPT(\mathcal{A},\mathcal{B}) \leq \epsilon OPT(\mathcal{A},\mathcal{B})$ for $\rho = \epsilon' = \delta \leq \epsilon/14$ and $\epsilon \leq 1$. We choose $\delta = 1/\lceil 14/\epsilon \rceil$. We obtain the following result:

Theorem 3.1 *For each $\epsilon \leq 1$, there is an algorithm for MKP instances with $\gamma \leq \lceil 1/\delta \log(1/\delta)^2 \rceil$ bins that computes a solution with profit at least $(1-\epsilon)OPT(\mathcal{A},\mathcal{B})$ and has a running time $2^{O(1/\epsilon \log(1/\epsilon)^4)} \cdot n$.*

Using a modified analysis, the running time can be bounded also by $2^{O(\gamma \log(1/\epsilon)^2)} \cdot n$ for $\gamma \leq poly(1/\delta)$ and $\gamma \geq 1/\delta$. For $\gamma \leq 1/\delta$, we obtain a modified algorithm with running time $2^{O(1/\delta \log(1/\delta)^2)} \cdot n$. Notice that $\gamma \leq \lceil 1/\epsilon \log(1/\epsilon)^2 \rceil \leq \lceil 1/\delta \log(1/\delta)^2 \rceil$ and that the algorithm generates in this case a solution of MKP with value $(1-\epsilon)OPT(\mathcal{A},\mathcal{B})$ and the running time above. In the next section we show how to obtain an EPTAS for general instances with running time $2^{O(1/\epsilon \log(1/\epsilon)^4)} poly(n)$ by combining the ideas in Section 2 and 3.

4 General Instances of MKP

In this section we study general instances of MKP. Let $\delta > 0$ be a constant such that $1/\delta$ is integral (δ will be specified later). If the number m of bins is smaller than the constant $\lceil 1/\delta (\log 1/\delta)^2 \rceil$, then we can use the approach in Section 3 and obtain a parameterized approximation scheme that generates a solution of MKP with profit at least $(1-\epsilon)OPT(\mathcal{A},\mathcal{B})$ for $\delta \leq 1/\lceil 14/\epsilon \rceil$. Suppose from now that the number m of bins is larger than $\lceil 1/\delta (\log 1/\delta)^2 \rceil$. In the following we give an outline of our algorithm for an instance with a large number of bins:

(1) Split the set $\mathcal{B}$ bins into blocks B_ℓ of size close to $\lceil 1/\delta \log(1/\delta)^2 \rceil$ and reduce the high and medium profit items in our instance.

(2) Guess high profit and medium profit items for block B_{t+1} with the largest bin sizes (corresponding to an optimum solution).

(3) Solve a modified LP relaxation of MKP to select the other items for $\mathcal{B}$ (including small profit items for block B_{t+1}).

(4) Construct 2D strip packing instances for each block B_ℓ of bins, round the corresponding rectangles and select items via a network flow computation.

(5) Use a bin packing algorithm to pack items into slightly more than $\lceil 1/\delta \log(1/\delta)^2 \rceil$ bins and select items via a shifting argument for the first t blocks.

(6) For block B_{t+1} pack the high and medium profit items into the bins plus one additional bin of small size.

(7) Use a second shifting argument to reduce the number of selected items of small profit for B_{t+1}, distribute these items to the bins and remove the fractional items.

(8) Finally use a third shifting argument to eliminate the additional bin.

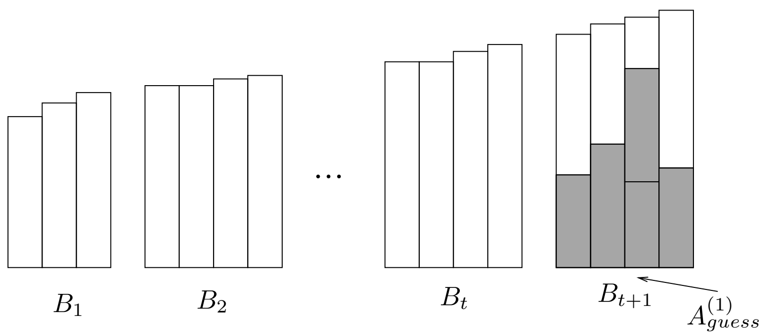

Figure 14: Block structure for a general instance of MKP with guessed set $A^{(1)}_{guess}$ for B_{t+1} .

Let us order the bins corresponding to their capacities $c(b_1) \leq \ldots \leq c(b_m)$. Now we build $t+1 = \lceil \frac{m}{\lceil 1/\delta(\log 1/\delta)^2 \rceil} \rceil$ blocks B_ℓ with $M = \lceil 1/\delta(\log 1/\delta)^2 \rceil$ bins; with exception of the block B_1 (that contains the bins with the smallest capacities) with maybe less than M bins (see Figure 14 for an illustration). We denote with $c^{(\ell)}_1, \ldots, c^{(\ell)}_M$ the capacities of the bins in B_ℓ for $\ell = 1, \ldots, t+1$.

In the next phase we select high and medium profit items for block B_{t+1} with the largest bin capacities. To do this we first compute the candidate set CA_h of high profit items and the candidate set CA_m of medium profit items. The cardinalities of the sets are: $|CA_h| \leq O(1/\delta^2 \log(1/\delta))$ and $|CA_m| \leq O(1/\delta^3 \log(1/\delta)^3)$ (see also Section 3.1 and 3.2). By rounding the corresponding profits we lose at most $6\epsilon' OPT(\mathcal{A}, \mathcal{B})$, but can suppose that our approximate solution contains only high and medium profit items from the set $CA_m \cup CA_h$. Now we guess the high profit items for B_{t+1} that contains only $M = \lceil 1/\delta(\log 1/\delta)^2 \rceil$ bins and guess a packing for them into the bins (similar to Section 3). The total number of guesses in this phase is bounded by $2^{O(1/\delta \log(1/\delta))}$. Let $A^{(1)}_{guess}$ be the chosen items and let $Area_{Rem} = \sum_{b_i \in B_{t+1}} c(b_i) - size(A^{(1)}_{guess})$ be the remaining space. Furthermore, $\bar{c}(b_i)$ denotes the remaining capacity of bin $b_i \in B_{t+1}$ after the packing of $A^{(1)}_{guess}$. A feasible pre-assignment of a set $A^{(1)}_{guess}$ into B_{t+1} is indicated in Figure 14.

Suppose for simplicity again that the largest remaining capacity is $\max_{b_i \in B_{t+1}} \bar{c}(b_i) = 1$ (otherwise scale all capacities and item sizes). This implies $Area_{Rem} \leq \lceil 1/\delta(\log 1/\delta)^2 \rceil$. For the medium profit items with large size $\in (\frac{\delta Area_{Rem}}{2K \log(1/\delta)^3}, 1]$ we use a guessing step to select a subset $A^{(2)}_{guess}$ and to assign $A^{(2)}_{guess}$ to the bins in B_{t+1} (if possible). For each feasible guess above, we now guess the rounded medium sizes for B_{t+1}, generate subintervals $(b^{(r)}_{i-1}, b^{(r)}_i]$ of medium sizes and numbers $k^{(r)}_i$ of medium items per subinterval for B_{t+1} (similar to Section 3). For the selection step of the medium profit items of medium sizes we use the following result.

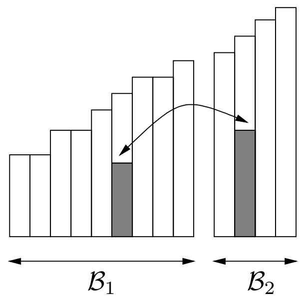

Figure 15: Exchange medium items for each subinterval $(b_{i-1}^{(r)}, b_i^{(r)}]$ between $\mathcal{B}_1$ and $\mathcal{B}_2$.

Lemma 4.1 *Suppose that there is a packing of a subset $S \subseteq \mathcal{A}$ of items into all bins $\bigcup_{\ell=1}^{t+1} B_\ell$. Then we can generate a modified packing of S into two bin groups $\mathcal{B}_1 = \bigcup_{\ell=1}^{t} B_\ell$ and $\mathcal{B}_2 = B_{t+1}$ plus one additional bin of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$ such that $\mathcal{B}_2$ contains the larger medium sizes for each subinterval $(b_{i-1}^{(r)}, b_i^{(r)}]$ where $\delta^6 Area_{Rem} \leq \min_r b_0^{(r)}$ and $\max_{r,i} b_i^{(r)} \leq \frac{\delta Area_{Rem}}{2K\log(1/\delta)^3}$.*

Proof: Suppose that there is a packing of S into $\mathcal{B}_1 \cup \mathcal{B}_2$. Let us exchange items a_j in the packing with size $size(a_j) \in (b_{i-1}^{(r)}, b_i^{(r)}]$ between $\mathcal{B}_1$ and $\mathcal{B}_2$ such that $\mathcal{B}_1$ contains the smaller sizes for each subinterval (see Figure 15 for an illustration). Clearly, the smaller sizes fit into $\mathcal{B}_1 = \bigcup_{\ell=1}^{t} B_\ell$. On the other hand, $\mathcal{B}_2 = B_{t+1}$ gets now larger sizes for each subinterval $(b_{i-1}^{(r)}, b_i^{(r)}]$ that in general do not fit in the corresponding bins. But we obtain the same number $k_i^{(r)}$ of items for each subinterval. Using the rounding, all items in the subinterval are rounded up to $b_i^{(r)}$. Using the arguments in the proof of Lemma 3.3, the rounded sizes fit into $\mathcal{B}_2$ plus one additional bin of small size. □

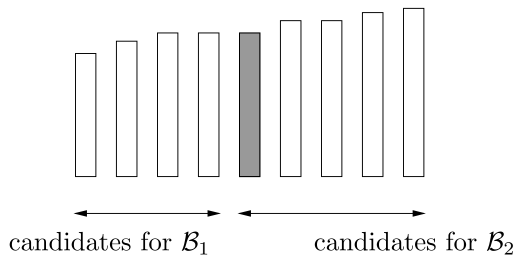

Figure 16: Guessing the index of the smallest item in $\mathcal{B}_2 = B_{t+1}$ for each subinterval $(b_{i-1}^{(r)}, b_i^{(r)}]$.

We use this exchange step above for all medium items with medium profit. Sort all medium items of the set CA_m in non-decreasing order of their sizes and guess the smallest index of a medium profit item that is packed into $\mathcal{B}_2 = B_{t+1}$ for each subinterval $(b_{i-1}^{(r)}, b_i^{(r)}]$ (see Figure 16 where the index of the item with smallest size for $\mathcal{B}_2$ is indicated by a dark shaded rectangle). Using the modification above this implies that $\mathcal{B}_2$ contains only items with indices larger than or equal to the guessed index for each subinterval. Since we have a constant number of candidates in each subinterval (bounded by a polynomial in $1/\delta$) and the number of subintervals is $O(1/\delta \log(1/\delta)^3)$, we can guess all these indices in time $(|CA_m|)^{O(1/\delta\log(1/\delta)^3)} = 2^{O(1/\delta\log(1/\delta)^4)}$. In each subinterval we choose $k_i^{(r)}$ items in CA_m with index larger than or equal to the guessed index. A greedy algorithm that takes $k_i^{(r)}$ feasible items ordered by their profits generates the best solution for the guessed index. Let $\bar{K}_{medium}$ be the selected set for the guess above. Similar to Section 3, the set $\bar{K}_{small}$ of all medium profit items with small size $\leq \delta^6 Area_{Rem}$ in our instance has total size $\frac{\delta Area_{Rem}}{8\log(1/\delta)^2}$ and can be placed into the additional bin of small size. Let $A^{(3)}_{guess} = \bar{K}_{medium} \cap \bar{K}_{small}$ be the selected set of medium profit items of medium and small size for $\mathcal{B}_2 = B_{t+1}$.

Let $A^{(r)}_i$ be the set of medium profit items in the subinterval $(b^{(r)}_{i-1}, b^{(r)}_i]$ with index smaller than the guessed index. Furthermore, let $\bar{K}_{large}$ be the set of all medium profit items with large size. Notice that $\mathcal{A}_1 = \bigcup_{i,r} A^{(r)}_i \cup (CA_h \setminus A^{(1)}_{guess}) \cup (\bar{K}_{large} \setminus A^{(2)}_{guess})$ consists of items with medium and large profit that may be selected for $\mathcal{B}_1 = \bigcup_{\ell=1}^t B_\ell$. Furthermore, the set $\mathcal{A}_2$ consists of items with small profit $\leq (\rho/\gamma)2(1+\epsilon')APP(\mathcal{A},\mathcal{B})$ that may be selected for $\mathcal{B}_1 \cup \mathcal{B}_2$ where $\gamma = |B_{t+1}| = \lceil 1/\delta \log(1/\delta)^2 \rceil$. On the other hand, $\bar{A}^{(r)}_i$ denotes the set of medium profit items with index larger than or equal to the guessed index for subinterval $(b^{(r)}_{i-1}, b^{(r)}_i]$. Then, $F = \bigcup_{i,r} \bar{A}_{i,r} \cup \bar{K}_{small} \cup \bigcup_{k=1}^2 A^{(k)}_{guess}$ is the forbidden set of medium profit items for $\mathcal{B}_1 = \bigcup_{\ell=1}^t B_\ell$.

Modified linear program relaxation. In the next phase we select the other items in $\mathcal{A}_1 \cup \mathcal{A}_2$ via a modified LP. Let $cap = \sum_{b_k \in B_{t+1}} \bar{c}(b_k) - size(A^{(2)}_{guess}) - size(J_{medium})$ be the remaining total capacity in the largest bins, where J_{medium} is the total size of all rounded medium size items with medium profit for B_{t+1} without the largest group for each interval $(2^{-(r+1)}, 2^{-r}]$ (see also Section 3.3). For simplicity suppose that the first $n' \leq n$ items have small profit and fit into the knapsack of size cap; i.e. $\{a_1, \ldots, a_{n'}\} \subset \mathcal{A}_2$. Next suppose that the next $n'' - n'$ items have either a small profit and size larger than cap or have medium or high profit; i.e. $\{a_{n'+1}, \ldots, a_{n''}\} = \mathcal{A}_1 \cup (\mathcal{A}_2 \setminus \{a_1, \ldots, a_{n'}\})$. For each bin b_k with capacity c_k, let $C^{(k)}_1, \ldots, C^{(k)}_{H_k}$ be the set of all configurations for the bin b_k; where a configuration is a subset $S \subseteq \mathcal{A}$ of items with total size $\sum_{a \in S} size(a) \leq c_k$. The main idea of the relaxation is to use a fractional variable $x_i \in [0,1]$ for each item a_i (which selects a piece of each item) and to distribute the corresponding piece as smaller fractional pieces among the configurations for different bin capacities. The variable $y^{(k)}_j$ in the LP denotes the length of configuration $C^{(k)}_j$ in bin b_k (similar to a configuration variable in bin packing [9]). For each of the first n' items with small profit, we use also a variable z_i to indicate a fractional piece selected for the bins in B_{t+1}. The modified linear program has the following form:

$$\begin{array}{lcll}
\max \sum_{i=1}^{n''} profit(a_i) x_i & & & \\
\sum_{k=1}^{m-M} \sum_{j: a_i \in C^{(k)}_j} y^{(k)}_j + z_i & = & x_i & \text{for } i = 1, \ldots, n' \\
\sum_{k=1}^{m-M} \sum_{j: a_i \in C^{(k)}_j} y^{(k)}_j & = & x_i & \text{for } i = n'+1, \ldots, n'' \\
\sum_{j=1}^{H_k} y^{(k)}_j & \leq & 1 & \text{for } k = 1, \ldots, m-M \\
\sum_{i=1}^{n'} size(a_i) z_i & \leq & cap & \\
y^{(k)}_j & \geq & 0 & \text{for } j = 1, \ldots, H_k \text{ and } k = 1, \ldots, m-M \\
z_i & \in & [0,1] & \text{for } i = 1, \ldots, n' \\
x_i & \in & [0,1] & \text{for } i = 1, \ldots, n''
\end{array}$$

Lemma 4.2 *The modified linear program LP is a relaxation of the MKP instance $(\mathcal{A},\mathcal{B})$ where $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$ after selecting the items of $\bigcup_{k=1}^3 A^{(k)}_{guess}$ for $\mathcal{B}_2 = B_{t+1}$ and the additional bin; i.e. the objective value of the LP is at least the maximum profit of a subset of $\mathcal{A} \setminus F$ packed together with $\cup_{k=1}^3 A^{(k)}_{guess}$ into $\mathcal{B}$ where we allow to pack further high and medium profit items into $\mathcal{B}_1$ and small profit items into $\mathcal{B}_1 \cup \mathcal{B}_2$.*

We suppose that all additional items with small profit that may be placed into $\mathcal{B}_2 = B_{t+1}$ have size at most $\delta\, cap$. We need this property in the rounding strategy later. Notice that there are at most $1/\delta$ small profit items of larger size in the bins. The total profit of these items can be bounded by $(1/\delta)2(\rho/\gamma)(1+\epsilon')APP(\mathcal{A},\mathcal{B})$. Using $\gamma = \lceil 1/\delta(\log 1/\delta)^2 \rceil$ and $APP(\mathcal{A},\mathcal{B}) \le OPT(\mathcal{A},\mathcal{B})$, the assumption above implies a profit loss of at most $2\rho(1+\epsilon')OPT(\mathcal{A},\mathcal{B})$. In the LP we simply remove a z_i variable, if the size of the corresponding item is too large. This implies for the modified linear program LP' that $OPT(LP') \ge OPT(LP) - 2\rho(1+\epsilon')OPT(\mathcal{A},\mathcal{B})$.

Lemma 4.3 *[12] We can compute an approximate solution $(\bar{x},\bar{y},\bar{z})$ of the modified LP' in time $poly(n,1/\alpha)$ where $\sum_{j=1}^{H_k} \bar{y}_j^{(k)} \le (1+2\alpha)$, $\sum_{i=1}^{n'} size(a_i)\bar{z}_i \le cap(1+2\alpha)$, and whose objective value is at least $(1-3\alpha)OPT(LP')$.*

The solution of the linear program LP' can be transformed in another solution $(\tilde{x},\tilde{y},\tilde{z})$ with objective value at least $(1-5\alpha)OPT(LP')$ without violating the constraints above. Here again we simply scale the values $\tilde{y}_j^{(k)} = \bar{y}_j^{(k)}/(1+2\alpha)$, $\tilde{z}_i = \bar{z}_i/(1+2\alpha)$ and $\tilde{x}_i = \bar{x}_i/(1+2\alpha)$.

Now we describe how to round the $(\tilde{x},\tilde{y},\tilde{z})$ solution of the modified LP. First we generate t stacks and rounded sets of rectangles $L_{wide}^{(\ell)}$ for the blocks B_ℓ, $\ell = 1,\ldots,t$ (see also Section 2). Here $L_{wide}^{(\ell)}$ consists of wide rectangles $(size(a_i), z_i^{(\ell)})$ where $size(a_i) > \delta c_{max}^{(\ell)}$ and $z_i^{(\ell)} = \sum_{b_k \in Bl_\ell} \sum_{j: a_i \in C_j^{(k)}} \tilde{y}_j^{(k)}$. In the same way as in Section 2 we add dummy rectangles to the stacks and generate the rounded wide rectangles $\bar{L}_{sup}^{(\ell)}$ for $\ell = 1,\ldots,t$. Moreover we define the set $L_{narrow}^{(\ell)}$ of narrow rectangles $(size(a_i), z_i^{(\ell)})$ with $size(a_i) \le \delta c_{max}^{(\ell)}$ and split these rectangles into groups depending on intervals $int_{\ell,k}$. Furthermore, we define the numbers $\eta_{\ell,k}$ for $\ell = 1,\ldots,t$. Then we build one additional stack for the small profit items placed into Bl_{t+1}. Sort the small profit items $a_1,\ldots,a_{n'}$ in non-decreasing order of their sizes and put the items with sizes in $[\delta^2 cap, \delta cap]$ as rectangles $(size(a_i), z_i)$ with width $size(a_i)$ and height z_i in the order above on a stack. This generates a stack St_{t+1} of total height H_{t+1}. Let $L_{wide}^{(t+1)}$ be the corresponding set of rectangles. After adding few dummy rectangles with width zero and total height at most $1/\delta^2$, we obtain a modified stack $\overline{St}_{t+1}$ with height $\bar{H}_{t+1} \le H_{t+1} + 1/\delta^2$, where $\bar{H}_{t+1} = a_{t+1}/\delta^2$ with $a_{t+1} \in \mathbb{Z}^+$ and $(a_{t+1}-1)/\delta^2 < H_{t+1}$. Then we split $\overline{St}_{t+1}$ into $1/\delta^2$ groups of height a_{t+1}. Splitting a rectangle into two pieces if necessary and rounding up the width of each rectangle in group j of $\overline{St}_{t+1}$ to the widest width in group j generates a set $\bar{L}_{sup}^{(t+1)}$ of rectangles with widths in $[\delta^2 cap, \delta cap]$.

The set $\bar{L}_{sup}^{(\ell)}$ consists of a set of rectangles with at most $1/\delta^2$ (a constant number) different widths $w_1^{(\ell)} > \ldots > w_{a(\ell)}^{(\ell)}$ for each $\ell = 1,\ldots,t+1$ where $a(\ell) \le 1/\delta^2$. For each width $w_j^{(\ell)}$, let $\beta_j^{(\ell)}$ be the total height of rectangles in $\bar{L}_{sup}^{(\ell)}$ with rounded width $w_j^{(\ell)}$ for $\ell = 1,\ldots,t+1$. Since the additional stack $\overline{St}_{t+1}$ has also height a_{t+1}/δ^2 where $a_{t+1} \in \mathbb{Z}^+$, the numbers $\beta_j^{(t+1)}$ are also integral for each group j.

Lemma 4.4

$$Area(\bar{L}_{sup}^{(t+1)}) \quad \le \quad (1+\delta)Area(L_{wide}^{(t+1)}) + \delta cap$$

Proof: The rounding of the stack $\overline{St}_{t+1}$ implies the inequality $Area(\bar{L}_{sup}^{(t+1)}) \le Area(L_{wide}^{(t+1)}) + \delta^2 \bar{H}_{t+1}\delta cap$, where $\bar{H}_{t+1}$ is the height of the stack and δcap is the maximum width of a rectangle. Notice that $\bar{H}_{t+1} \le H_{t+1} + 1/\delta^2$ and $\delta^2 \bar{H}_{t+1} \le \delta^2 H_{t+1} + 1$. Furthermore, the area $Area(L_{wide}^{(t+1)})$ is at least $H_{t+1}\delta^2 cap$. Therefore, $Area(\bar{L}_{sup}^{(t+1)}) \le Area(L_{wide}^{(t+1)}) + (\delta^2 H_{t+1} + 1)\delta cap \le (1+\delta)Area(L_{wide}^{(t+1)}) + \delta cap$. □

Notice that $Area(L_{wide}^{(t+1)})$ is bounded by cap. For the small profit items with size $< \delta^2 cap$ we form intervals $int_{t+1,k} = (\frac{\delta^2 cap}{(1+\delta)^k}, \frac{\delta^2 cap}{(1+\delta)^{k-1}}]$ for $k \in \mathbb{N}$. The total area of all small profit pieces allocated to $\mathcal{B}_2 = B_{t+1}$ with size in $int_{t+1,k}$ is given by $Area(t+1, int_{t+1,k}) = \sum_{i: size(a_i) \in int_{t+1,k}, profit(a_i) \text{small}} z_i size(a_i)$. As capacity for the flow problem we use $\eta_{t+1,k} = \left\lceil \frac{Area(t+1, int_{t+1,k})}{\frac{\delta^2 cap}{(1+\delta)^k}} \right\rceil$. Notice that the total area of $\eta_{t+1,k}$ items of sizes in $int_{t+1,k}$ is at most $\eta_{t+1,k}\frac{\delta^2 cap}{(1+\delta)^{k-1}} \le (1+\delta)Area(t+1, int_{t+1,k}) + \frac{\delta^2 cap}{(1+\delta)^{k-1}}$. Summing over all intervals $int_{t+1,k}$, the total area is at most $(1+\delta)Area(L_{narrow}^{(t+1)}) + (1+\delta)\delta cap$ (see also Lemma 2.2) where $L_{narrow}^{(t+1)}$ is the set of rectangles $(size(a_i), z_i)$ over all small profit items a_i with $size(a_i) < \delta^2 cap$. For items with very small size $size(a_i) \le (1/n)\delta cap$, the total size is at most δcap. Define $index(t+1)$ as in Section 2 and use for $int_{t+1,index(t+1)+1}$ the capacity $\eta_{t+1,index(t+1)+1} = n$. In total we generate a flow network for $t+1$ blocks and compute an integral solution of the following form:

Lemma 4.5 *The algorithm (via the solution of the flow network) computes sets $A_{wide}^{(\ell)}, A_{narrow}^{(\ell)}$ of items for each block B_ℓ ($\ell = 1, \ldots, t+1$) with profit$(\bigcup_\ell A_{wide}^{(\ell)} \cup A_{narrow}^{(\ell)}) \ge (1-5\alpha)OPT(LP')$ such that the following properties are satisfied:*

- $|\{a_i \in A_{wide}^{(\ell)} | g(x_i, y_{\ell,j}) = 1\}| \le \beta_j^{(\ell)}$ *for each* $j = 1, \ldots, a(\ell)$ *and* $\ell = 1, \ldots, t+1$.
- $|\{a_i \in A_{narrow}^{(\ell)} | g(x_i, \bar{y}_{\ell,k}) = 1\}| \le \eta_{\ell,k}$ *for each* $j = 1, \ldots, index(\ell)+1$ *and* $\ell = 1, \ldots, t+1$.

In addition, $LIN(A_{wide}^{(\ell)}, B_\ell) \le \delta M + 3$, $Area(A_{wide}^{(\ell)}) \le (1+\delta)Area(L_{wide}^{(\ell)}) + 2c_{max}^{(\ell)}$, and $Area(A_{narrow}^{(\ell)}) \le (1+\delta)Area(L_{narrow}^{(\ell)}) + (3/2+\delta)c_{max}^{(\ell)}$ for $\ell = 1, \ldots, t$. Furthermore, $Area(A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)}) \le (1+\delta)Area(L_{wide}^{(t+1)} \cup L_{narrow}^{(t+1)}) + 4\delta cap$.

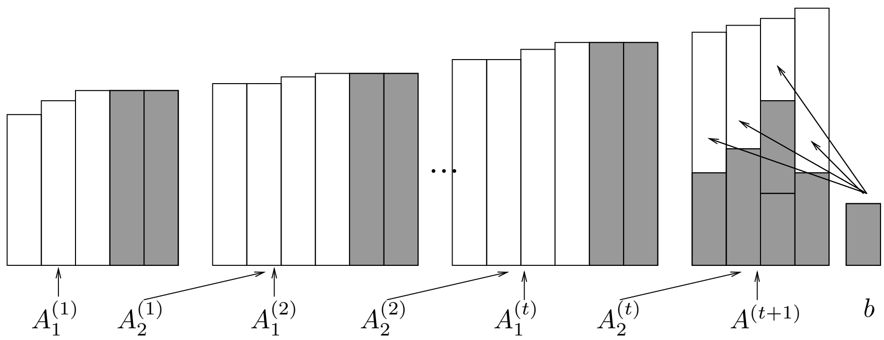

Figure 17: Shifting strategy applied to a general instance of MKP.

In the following we show that most of the selected items can be placed into the bins. Using the algorithm in Section 2 for $\alpha = \delta$, we generate a packing for $A_{wide}^{(\ell)} \cup A_{narrow}^{(\ell)}$ into $M + \lfloor C' \log(1/\delta)^2 \rfloor$ bins for $\ell = 1, \ldots, t$: a subset $A_1^{(\ell)}$ fits into the block B_ℓ and the remaining set $A_2^{(\ell)}$ fits into $\lfloor C' \log(1/\delta)^2 \rfloor$ bins of size $c_{max}^{(\ell)}$ (see Figure 17 for an illustration). Via the shifting argument we select a subset $X_{\ell+1} \subset A_2^{(\ell)} \cup A_1^{(\ell+1)}$ with profit at least $(1 - \bar{C}'\delta)$*profit*$(A_2^{(\ell)} \cup A_1^{(\ell+1)})$ that can be packed into block $B_{\ell+1}$ for $\ell = 1, \ldots, t-1$ (see Lemma 2.9). For the first block B_1 we take $X_1 = A_1^{(1)}$. For the last block B_{t+1} with selected set $A^{(t+1)} = A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)}$ the situation is a bit more complicated. Suppose from now that $1/\delta$ is integral and a multiple of 5.

Lemma 4.6 *We can compute a subset $X'_{t+1} \subseteq A^{(t+1)} = A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)}$ with*

$$profit(X'_{t+1}) \ge (1-5\delta)profit(A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)}) - 4\delta\rho(1+\epsilon')OPT(\mathcal{A}, \mathcal{B})$$

such that $AREA(X'_{t+1}) \le cap$.

Proof: Notice that $Area(A_{wide}^{(t+1)} \cup A_{narrow}^{t+1}) \leq (1+\delta)Area(L_{wide}^{(t+1)} \cup L_{narrow}^{(t+1)}) + 4\delta cap \leq cap + 5\delta cap$. Pack all these items into a bin of capacity $cap + 5\delta cap$ and split the bin into pieces of size $5\delta cap$. The number of pieces is $\frac{cap+5\delta cap}{5\delta cap} = 1 + \frac{1}{5\delta}$; an integral number using the assumption above. Then, there is at least one piece with profit at most $\frac{1}{1+\frac{1}{5\delta}} \mathit{profit}(A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)}) \leq 5\delta\, \mathit{profit}(A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)})$. Removing the items that lie inside this piece of the bin plus two items of small profit that lie partially in the bin gives a subset X'_{t+1} with $\mathit{profit}(X'_{t+1}) \geq (1-5\delta)\mathit{profit}(A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)}) - 4\delta\rho(1+\epsilon')OPT(\mathcal{A},\mathcal{B})$. □

In the next phase we distribute the small profit items in X'_{t+1} to the bins in $\mathcal{B}_2 = B_{t+1}$ (via a greedy algorithm similar to section 3.3). Again we have to eliminate some fractional items and lose a profit of at most $2\rho(1+\epsilon')OPT(\mathcal{A},\mathcal{B})$. Including the 2 additional items above, the profit of the placed items X_{t+1} is at least $(1-5\delta)\mathit{profit}(A_{wide}^{(t+1)} \cup A_{narrow}^{(t+1)}) - 4\rho OPT(\mathcal{A},\mathcal{B})$ (using $\epsilon'/2 + \delta + \delta\epsilon' \leq 1/2$). This implies that $profit(\bigcup_{\ell=1}^{t+1} X_\ell \cup A_2^{(t)}) \geq (1 - \max\{10, 5+\bar{C}'\}\delta)OPT(LP') - 4\rho OPT(\mathcal{A},\mathcal{B})$.

Using our assumption about the modified LP' we have $OPT(LP') \geq OPT(LP) - 2\rho(1+\epsilon')OPT(\mathcal{A},\mathcal{B})$. In the guessing step for the medium and high profit items for $\mathcal{B}_2$, we lose at most $6\epsilon' OPT(\mathcal{A},\mathcal{B})$ via the rounding of the profits. Let us take an optimum solution for instance $(\mathcal{A},\mathcal{B})$ with rounded high and medium profits. Then, our algorithm guesses a subset of items $\bigcup_{k=1}^{3} A_{guess}^{(k)}$ corresponding to the optimum solution such that $OPT(LP) + \mathit{profit}(\bigcup_k A_{guess}^{(k)}) \geq OPT(\mathcal{A},\mathcal{B}) - 6\epsilon' OPT(\mathcal{A},\mathcal{B})$.

Our algorithm computes for the set $\bigcup_{\ell=1}^{t+1} X_\ell \cup A_2^{(t)}$ and the guessed sets $\bigcup_k A_{guess}^{(k)}$ a packing into $\mathcal{B}_1 \cup \mathcal{B}_2$ plus $\lfloor \bar{C}' \log(1/\delta)^2 \rfloor$ bins of size $c_{max}(\mathcal{B}_1)$ and one additional bin of size $\frac{\delta Area_{Rem}}{4\log(1/\delta)^2}$. Using the shifting technique we remove the additional bin and the $\lfloor \bar{C}' \log(1/\delta)^2 \rfloor$ bins (see also Figure 17 for an illustration) and lose profit at most $(4\rho(1+\epsilon')+\delta)OPT(\mathcal{A},\mathcal{B}) + \bar{C}'\delta OPT(\mathcal{A},\mathcal{B})$. For the calculation of the profit loss we refer to Lemma 2.9 and Section 3 (the shifting argument to eliminate the additional bin b). Therefore, the profit of the selected set is at least

$$\begin{array}{ll}
(1 - \max\{10, 5+\bar{C}'\}\delta)OPT(LP') + \mathit{profit}(\bigcup_k A_{guess}^{(k)}) - (4\rho(2+\epsilon') + \delta + \bar{C}'\delta)OPT(\mathcal{A},\mathcal{B}) & \geq \\
(1 - \max\{10, 5+\bar{C}'\}\delta)OPT(LP) + \mathit{profit}(\bigcup_k A_{guess}^{(k)}) - (6\rho(1+\epsilon') + 4\rho + \delta + \bar{C}'\delta)OPT(\mathcal{A},\mathcal{B}) & \geq \\
(1 - \max\{10, 5+\bar{C}'\}\delta)OPT(\mathcal{A},\mathcal{B}) - (6\epsilon' + 10\rho + \rho + \delta + \bar{C}'\delta)OPT(\mathcal{A},\mathcal{B}) & = \\
(1 - \max\{28+\bar{C}', 23+2\bar{C}'\}\delta)OPT(\mathcal{A},\mathcal{B}) &
\end{array}$$

(using $\epsilon' \leq 1/6$ and $\delta = \rho = \epsilon'$). For $\delta \leq \epsilon/\max\{28+\bar{C}', 23+2\bar{C}'\}$, the profit of our solution is at least $(1-\epsilon)OPT(\mathcal{A},\mathcal{B})$. In our algorithm we set $\delta = 1/(5\lceil \max\{28+\bar{C}', 23+2\bar{C}'\}/5\epsilon \rceil)$ and obtain the property that $1/\delta$ is integral and a multiple of 5. The running time of our algorithm is dominated by the guessing steps for $\mathcal{B}_2$ and can be bounded by $2^{O(1/\delta \log(1/\delta)^4)} poly(n)$. By considering two cases with $poly(n) \leq 2^{O(1/\delta \log(1/\delta)^4)}$ and $2^{O(1/\delta \log(1/\delta)^4)} \leq poly(n)$, the running time also can be bounded by $2^{O(1/\delta \log(1/\delta)^4)} + poly(n)$. This nice argument was used also by Downey et al. [7] for FPT algorithms.

5 Remark

If the modified round-up conjecture for our bin packing problem with different bin sizes $OPT_{ILP}(\mathcal{A},\mathcal{B}) \leq LIN(\mathcal{A},\mathcal{B}) + C$ is true (where $(\mathcal{A},\mathcal{B})$ is an instance with item set $\mathcal{A}$ and bin set $\mathcal{B}$ and where C is a constant) similar to the modified round-up conjecture for the classical bin packing problem by Scheithauer and Terno [22], then we can modify our algorithm and the rounding scheme. This bound implies that we have to eliminate only a constant number of bins in the shifting phase instead of $\bar{C}' \log(1/\delta)^2$ many bins. Therefore, we can modify our construction such that $\mathcal{B}_1$ contains groups with $1/\delta$ bins and $\mathcal{B}_2$ consists of one group with $1/\delta$ bins. Then, the guessing steps for $\mathcal{B}_2 = B_{t+1}$ can be implemented in time $2^{O(1/\delta \log(1/\delta)^2)}$ and the running time of the entire algorithm can be bounded by $2^{O(1/\delta \log(1/\delta)^2)} + poly(n)$.

Acknowledgments. The author thanks Mike Fellows and Roberto Solis-Oba for many helpful discussions.

References

[1] C. Bazgan: Schémas d'approximation et complexité paramétrée, *Technical Report*, Université Paris Sud, 1995.

[2] A. Caprara, H. Kellerer, and U. Pferschy: The multiple subset sum problem, *SIAM Journal of Optimization*, 11 (2000), 308-319.

[3] M. Cesati and L. Trevisan: On the efficiency of polynomial time approximation schemes, *Information Processing Letters*, 64 (1997), 165-171.

[4] C. Chekuri and S. Khanna, A PTAS for the multiple knapsack problem, *Proceedings of the 17.th ACM-SIAM Symposium on Discrete Algorithms*, SODA 2000, 213-222 and *SIAM Journal on Computing*, 35 (2006), 713-728.

[5] F. Diedrich and K. Jansen: Improved approximation algorithms for scheduling with fixed jobs, *Proceedings of the 20.th ACM-SIAM Symposium on Discrete Algorithms*, SODA 2009, 675-684.

[6] R. Downey: Parametrized complexity for the skeptic, *Proceedings of the 18.th IEEE Conference on Computational Complexity*, CCC 2003, 147-169.

[7] R. Downey, M. Fellows, and U. Stege: Parameterized complexity: a framework for systematically confronting computational intractability, in: *Contemporary Trends in Discrete Mathematics*, R. Graham, J. Kratochví, J. Nesetril and F. Roberts (eds.), Proceedings DIMACS-DIMATIA Workshop, Prague, 1997, ACM-DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 49 (1999), 49-99.

[8] M.R. Fellows: Blow-ups, win/win's, and crown rules: some new directions in FPT, *Proceedings of the 29.th Workshop on Graph Theoretical Concepts in Computer Science*, WG 2003, LNCS 2880, 1-12.

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

[10] M.D. Grigoriadis, L.G. Khachiyan, L. Porkolab and J. Villavicencio: Approximate max-min resource sharing for structured concave optimization, *SIAM Journal on Optimization*, 41 (2001), 1081-1091.

[11] O.H. Ibarra and C.E. Kim: Fast approximation algorithms for the knapsack and sum of subset problems, *Journal of the ACM*, 22 (1975), 463-468.

[12] K. Jansen: Parameterized approximation scheme for the multiple knapsack problem, *Proceedings of the 20.th ACM-SIAM Symposium on Discrete Algorithms*, SODA 2009, 665-674 and *SIAM Journal on Computing*, to appear.

[13] N. Karmarkar and R.M. Karp: An efficient approximation scheme for the one-dimensional bin-packing problem, *Proceedings of the 23.th IEEE Symposium on Foundations of Computer Science*, FOCS 1982, 312-320.

[14] H. Kellerer: A polynomial time approximation scheme for the multiple knapsack problem, *Proceedings of the 2.nd Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, APPROX 1999, LNCS 1671, 51-62.

[15] H. Kellerer, U. Pferschy, and D. Pisinger: *Knapsack Problems*, Springer Verlag, Berlin, 2004.

[16] C. Kenyon and E. Remila: Approximate strip packing, *Mathematics of Operations Research*, 25 (2000), 645-656.

[17] E.L. Lawler: Fast approximation algorithms for knapsack problems, *Mathematics of Operations Research*, 4 (1979), 339-356.

[18] S. Martello and P. Toth: *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley and Sons, New York, 1990.

[19] D. Marx: Parametrized complexity and approximation algorithms, *The Computer Journal*, 51 (2008), 60-78.

[20] S. Sahni: Approximate algorithms for the 0/1 knapsack problem, *Journal of the ACM* 22 (1975), 115-124.

[21] M. Scharbrodt, A. Steger, and H. Weisser: Approximability of scheduling with fixed jobs, *Proceedings of the 10.th ACM-SIAM Symposium on Discrete Algorithms*, SODA 1999, 961-962.

[22] G. Scheithauer and J. Terno: Theoretical investigations on the modified integer round-up property for the one-dimensional cutting stock problem, *European Journal of Operational Research* 20 (1997), 93-100.

[23] G. Shmonin: Parameterised integer programming, integer cones, and related problems, PhD thesis, Univ. Paderborn, 2007.