

Packing Groups of Items into Multiple Knapsacks

Lin Chen¹ and Guochuan Zhang^{*2}

- 1 Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary
chenlin198662@gmail.com
- 2 Zhejiang University, College of Computer Science, Hangzhou, China
zgc@zju.edu.cn

Abstract

We consider a natural generalization of the classical multiple knapsack problem in which instead of packing single items we are packing groups of items. In this problem, we have multiple knapsacks and a set of items which are partitioned into groups. Each item has an individual weight, while the profit is associated with groups rather than items. The profit of a group can be attained if and only if every item of this group is packed. Such a general model finds applications in various practical problems, e.g., delivering bundles of goods. The tractability of this problem relies heavily on how large a group could be. Deciding if a group of items of total weight 2 could be packed into two knapsacks of unit capacity is already *NP*-hard and it thus rules out a constant-approximation algorithm for this problem in general. We then focus on the parameterized version where the total weight of items in each group is bounded by a factor δ of the total capacity of all knapsacks. Both approximation and inapproximability results with respect to δ are derived. We also show that, depending on whether the number of knapsacks is a constant or part of the input, the approximation ratio for the problem, as a function on δ , changes substantially, which has a clear difference from the classical multiple knapsack problem.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics

Keywords and phrases approximation algorithms, lower bound, multiple knapsack, bin packing

Digital Object Identifier 10.4230/LIPIcs.STACS.2016.28

1 Introduction

The classical multiple knapsack problem aims at a most profitable subset of given items which admits a feasible packing on a given set of knapsacks. In this setting, if an item is packed, its profit is counted into the objective value. In this paper, we investigate a scenario in which items appear in groups, and the items in a group share a single profit. In other words, one can get the profit if and only if all items in the group are packed (can be placed into different knapsacks). It is obviously a natural generalization of the classical model where each group consists of exactly one item. More precisely, the problem of packing groups of items into multiple knapsacks (*GMKP*) is defined as follows. There are N disjoint sets (groups) of items $S_i = \{J_j^i \mid 1 \leq j \leq n_i\}$ where J_j^i is the j -th item of the i -th set. Each item has a weight $w(J_j^i) = w_j^i$. There are m identical knapsacks (bins), each having a capacity of B . There is a profit p_i for each set S_i , which could be achieved only if every item of the set is packed. The goal is to pack items into knapsacks such that the total profit is maximized.

* Research supported in part by NSFC (11271325).

By scaling, we assume that the capacity of each knapsack is 1 and $w_j^i \in [0, 1]$. We define the weight of S_i as $w_i = w(S_i) = \sum_{j \in S_i} w_j^i$, and its density ratio as $p_i/w(S_i)$. Throughout the paper, “bins” and “knapsacks” are used interchangeably.

Although most of the times it is reasonable to assume that every single item has an individual profit, as we do in the classical (multiple) knapsack problem, it does happen in many cases that the profit can only be defined for a group of items, instead of each of them. Consider several people going for hiking together. All the necessities for building a tent, like the poles, ropes, sticks and the tent itself, have one uniform value which could be achieved only when each of them is carried. Another example would be the delivery of huge equipments, which could be split into smaller parts and carried by multiple trucks. However, no part of one equipment has an individual value and it only makes sense to carry all the parts. All of these natural applications motivate us to study the *GMKP* problem.

In general, *GMKP* does not admit any constant ratio approximation algorithm as it is easy to see that deciding whether a single group of items (with the profit of 1) could be packed into $m = 2$ bins is exactly the Partition problem and is NP-complete. However, the intractability of the problem follows from the fact that a single group may have a weight as large as the total capacity of all the knapsacks (bins), which is often not the case in practice. For example, all parts of one huge equipment may exceed the capacity of one truck, however, compared with the total capacity of all the trucks owned by the delivery company, it is usually small. Hence, we put additionally the constraint that $w(S_i) \leq \delta m$ for all i and discuss the approximability of the problem with respect to the parameter $\delta \in (0, 1]$.

Throughout this paper, we say that an algorithm has an approximation factor c if it always produces a feasible packing with total profit at least c times the optimal value. Clearly $c < 1$. For the sake of conventional convenience, if the factor c is arbitrarily small, we say such an algorithm does not have a constant ratio.

Related Work

We first provide a brief overview on the classical multiple knapsack problem (*MKP*). In *MKP*, every item j has a weight w_j and profit p_j , and every knapsack (bin) i has an individual capacity of B_i . The goal is to pack items into knapsacks such that the total profit is maximized. In 1999, Kellerer [11] provided a PTAS (Polynomial Time Approximation Scheme) for the special case of the multiple knapsack problem where all knapsacks have the uniform capacity, i.e., $B_i = B$. Later on, Chekuri and Khanna [2] gave a PTAS for the general multiple knapsack problem where each B_i can be different. This PTAS was later improved by Jansen [6] [7] to an EPTAS (Efficient Polynomial Time Approximation Scheme) of a running time $2^{O(\log^4(1/\epsilon)/\epsilon)} + n^{O(1)}$. On the other hand, Jansen et al. [9] also showed that unless the Exponential Time Hypothesis fails, there is no approximation scheme which has a running time of $2^{o(1/\epsilon)} + n^{O(1)}$ for the multiple knapsack problem even if there are only two knapsacks (of the unit capacity). Thus, allowing the number of knapsacks m to be part of the input as well as allowing each knapsack to have a distinct capacity does not essentially make the problem harder in the sense that the $2^{O(\log^4(1/\epsilon)/\epsilon)} + n^{O(1)}$ time EPTAS for the general *MKP* is almost the best possible even for the special case that $m = 2$. However, things are substantially different for *GMKP* where the profit is associated with groups instead of items. We show in this paper that if m is a constant, *GMKP* admits a constant-factor approximation algorithm as long as $\delta < 1$. If m is part of the input, *GMKP* admits a constant-factor approximation algorithm only if $\delta \leq 2/3$. Furthermore, if we allow knapsacks to have distinct capacities, then even if there are only two kinds of knapsacks, say, m_1 knapsacks of capacity c_1 and m_2 knapsacks of capacity c_2 , then for any $\delta \in (0, 1)$ the

■ **Table 1** Overview of the results.

m	δ	Upper Bound	Lower Bound
constant	$(0, 1)$	$1 - f(\delta) + \epsilon$	$1 - f(\delta) - \epsilon$
input	$(2/3, 1)$	ϵ	0
	$(1/3, 2/3]$	$1/2 + \epsilon$	$1/2 - \epsilon$
	$(1/4, 1/3]$	$1 - 3\delta/(2 + 3\delta) + \epsilon$	$1/2 - \epsilon$
	$(0, 1/4]$	$1 - 3\delta/(2 + 3\delta) + \epsilon$	$1 - 2\delta - \epsilon$

constraint $w(S_i) \leq \delta(m_1c_1 + m_2c_2)$ is no longer capable of guaranteeing a constant-factor approximation (See the full version of the paper). Hence, unlike *MKP*, the parameter m , as well as the capacities of knapsacks, influences *GMKP* substantially. We hope the study in this line will help reveal the impact of these parameters.

Our problem is closely related to the all-or-nothing generalized assignment problem (*AGAP*) [1]. The *AGAP* problem also asks for a most profitable packing of n groups of items into m identical knapsacks, where the profit of a group is defined to be the total profit of items in the group, and is achieved only if every item of this group is packed. The major difference between *AGAP* and our *GMKP* problem is that *AGAP* further requires that every knapsack could accommodate at most one item from each group. This additional constraint allows *AGAP* to admit an $O(1)$ -approximation algorithm, while *GMKP* does not admit any constant approximation algorithm in general.

Our problem is also closely related to the bin packing problem (*BPP*) in which every item has a weight and the goal is to pack all the items into the smallest number of bins. In *GMKP*, if we know which groups are selected by the optimum solution, we get a bin packing problem as we need to pack the items of the selected groups into a fixed number of bins. It is proved in [8] that the problem is $W[1]$ -hard parameterized by the number of bins m even with unary encoding. This result directly implies the $W[1]$ -hardness of our problem.

Our Contribution

We give a thorough study on the approximability of *GMKP* with respect to the parameter δ . From now on we will use $GMKP(\delta)$ to specify the parameter. The reader may refer to Table 1 for an overview, where each lower bound means there exists an algorithm achieving a profit at least a certain fraction of the optimum, and each upper bound means that there does not exist a polynomial time algorithm achieving a profit of such a fraction of the optimum under $P \neq NP$. Here $f(\delta) = 1/(1/\delta + 1)$ if $1/\delta$ is an integer and could be divided by m , and $f(\delta) = 1/\lceil 1/\delta \rceil$ otherwise, and $\epsilon > 0$ is an arbitrarily small constant.

The main contribution of this paper is to give a full characterization of the approximability of $GMKP(\delta)$, and distinguish $GMKP(\delta)$ with m being a constant from $GMKP(\delta)$ with m being part of the input based on such a characterization. Our results imply that, if m is a constant, then $GMKP(\delta)$ could be approximated to a factor of roughly $1 - \delta$, hence it admits a constant-ratio approximation algorithm as long as $\delta < 1$. However, if m is part of the input, $GMKP(\delta)$ does not admit any constant-ratio approximation algorithm when $\delta > 2/3$ (assuming $P \neq NP$), and admits a $(1/2 - \epsilon)$ -approximation algorithm as long as $\delta \leq 2/3$. Furthermore, when δ is sufficiently small (e.g., $\delta \in (0, 1/4]$), the approximation ratio lies within $[1 - 2\delta - \epsilon, 1 - 3\delta/(2 + 3\delta) + \epsilon]$, which has a clear difference from the ratio of $1 - \delta$ for the case that m is a constant.

To achieve our results, we study $OPT(m)$ as a function of m , where $OPT(m)$ is the optimum profit by using m bins. By modifying the classical dynamic programming algorithm

for MKP [2], we show that a profit of $(1 - \epsilon)OPT(m)$ could be achieved by using $(1 + \epsilon)m$ bins even for the $GMKP(\delta)$ problem. Hence, we could derive in polynomial time a feasible solution of profit $(1 - \epsilon)OPT((1 - \epsilon)m)$. A crucial observation leading to a PTAS for MKP is that $OPT(m)$ is somehow “continuous” in the sense that $OPT((1 - \epsilon)m) \geq (1 - \epsilon)OPT(m)$. However, it is no longer true for $GMKP(\delta)$. Indeed, if m is part of the input, we prove that by assuming $P \neq NP$, for any $\epsilon > 0$ and $c < 3/2$, the inequality $OPT((1 - \epsilon)m) \geq (1 - c\delta)OPT(m)$ does not hold, which implies a jump on the optimum. We also show that $OPT((1 - \epsilon)m) \geq (1 - 2\delta - O(\epsilon))OPT(m)$, which implies a $(1 - 2\delta - \epsilon)$ -approximation algorithm. To prove such a bound, we will use the configuration LP for bin packing problem introduced in [4] and apply discrepancy analysis to estimate how the deletion of certain items influences the whole packing.

2 Packing into a Constant Number of Bins

We give almost tight approximation algorithms for $GMKP(\delta)$ when m is a constant. We start with the upper bound, as is shown by the following theorem.

► **Theorem 1.** *Assuming $P \neq NP$, there is no $(1 - f(\delta) + \epsilon)$ -approximation algorithm for the group packing problem $GMKP(\delta)$ for any constant m , where $f(\delta) = 1/(1/\delta + 1)$ if $1/\delta$ is an integer and could be divided by m , and $f(\delta) = 1/\lceil 1/\delta \rceil$ otherwise.*

The approximability of $GMKP(\delta)$ relies on the function f which has a jump when $1/\delta$ is an integer and could be divided by m . This is due to the hardness result of the following *Repartition* problem.

Repartition- (x, m) . Given x sets of integers S_1, S_2, \dots, S_x where $S_i = \{b_j^i \in Z^+ \mid 1 \leq j \leq n_i, n_i \in Z^+\}$, $\sum_{j \in S_i} b_j^i = B$ for every i and $m \mid Bx$, the problem asks whether there exists a repartition of all the integers b_j^i into m disjoint sets such that the integers in each set sum up to exactly Bx/m .

► **Lemma 2.** *Repartition- (x, m) is NP-complete for any x and m such that x could not be divided by m , and is polynomially solvable otherwise.*

It is easy to see that Partition is actually a special case of the Repartition problem by taking $x = 1$ and $m = 2$.

We complement Theorem 1 by giving an algorithm with the approximation ratio that almost matches the bound.

► **Theorem 3.** *There is a $(1 - f(\delta) - \epsilon)$ -approximation algorithm for the group packing problem $GMKP(\delta)$ if m is a constant, where $f(\delta) = 1/(1/\delta + 1)$ if $1/\delta$ is an integer and could be divided by m , and $f(\delta) = 1/\lceil 1/\delta \rceil$ otherwise.*

To prove Theorem 3, we need the following lemma.

► **Lemma 4.** *If there exists a feasible solution Sol of profit ϕ for the group packing problem $GMKP(\delta)$ when m is a constant, and the total weight of all the items in the solution is at most $(1 - \epsilon)m$, then there exists a polynomial time algorithm which returns a feasible solution of profit at least ϕ .*

The proof of Lemma 4 is a combination of guessing out big items (with weight larger than ϵ^2), and greedily selecting and packing small items (with weight no more than ϵ^2). The fact that the total weight of items in Sol is no more than $(1 - \epsilon)m$ ensures that there is

enough room (the amount of ϵm) to offset the errors caused by the possible wrongly selection and packing of small items.

With Lemma 4, Theorem 3 is proved via selecting out appropriate sets whose total weight is at least ϵm and the total profit is at most $f(\delta)OPT$.

Proof of Theorem 3. According to Lemma 4 (for ease of calculation we substitute ϵ by ϵ^2 in the lemma), if the optimum solution of $GMKP(\delta)$ has a total weight at most $m(1 - \epsilon^2)$ then the theorem is already proved. Otherwise it suffices to prove that given the optimum solution, say, Sol , we can always delete some sets such that the total weight of these sets is at least $\epsilon^2 m$, and the total profit is at most $f(\delta)OPT$. In Sol if there exists a single set of weight at least $\epsilon^2 m$ and total profit at most $f(\delta)OPT$ we are done. Otherwise every set of weight at least $\epsilon^2 m$ has a profit strictly larger than $f(\delta)OPT$. We call such sets as critical sets and there are at most $1/f(\delta) - 1$ critical sets (recall that $1/f(\delta)$ is an integer).

Suppose $1/\delta$ is not an integer dividable by m , then there are at most $1/f(\delta) - 1 = \lceil 1/\delta \rceil - 1$ critical sets, with total weight at most $[1 - \delta(\lceil 1/\delta \rceil - 1)]m$. Since δ is a constant, it is always possible to choose sufficiently small ϵ such that $1 - \delta(\lceil 1/\delta \rceil - 1) \geq 2\epsilon$. As the total weight of all the sets is larger than $(1 - \epsilon^2)m$, we know that in addition to critical sets there are also other sets in Sol , and the total weight of these non-critical sets is at least $(1 - \epsilon^2)m - \delta(\lceil 1/\delta \rceil - 1)m \geq \epsilon m$. Notice that the total profit of non-critical sets is at most OPT , hence the average ratio of these sets is upper bounded by $OPT/(\epsilon m)$. Hence, by selecting least profitable (in terms of ratios, i.e., $p_i/w(S_i)$) non-critical sets such that their total weight is in $(\epsilon^2 m, 2\epsilon^2 m]$, we know their total profit is at most $2\epsilon^2 m \cdot OPT/(\epsilon m) \leq 2\epsilon OPT$. Overall, we find sets with total weight at least $\epsilon^2 m$ and total profit at most $2\epsilon OPT \leq f(\delta)OPT$, which proves the theorem.

Suppose $1/\delta = \lambda$ is an integer dividable by m , then $f(\delta) = 1/(1 + \lambda)$. Recall that every set of weight at least $\epsilon^2 m$ has a profit strictly larger than $OPT/(1 + \lambda)$. Consider the optimum solution. If there exist λ sets of items such that their total profit is at least $\lambda/(1 + \lambda) \cdot OPT$, then we can guess out these λ sets. As λ could be divided by m , we put items of λ/m sets into one bin, and their total weight is at most $\delta m \cdot \lambda/m = 1$. Hence we derive a feasible packing with profit at least $\lambda/(1 + \lambda) \cdot OPT = (1 - f(\delta))OPT$. Otherwise, any λ sets in the optimum solution have a total profit less than $\lambda/(1 + \lambda) \cdot OPT$, specifically, the λ sets of the largest weight also have a total profit less than $\lambda/(1 + \lambda) \cdot OPT$. Hence, among the λ sets of the largest weight, the one of the smallest profit has a profit at most $OPT/(1 + \lambda)$, implying that it is not critical, hence has a weight at most $\epsilon^2 m$. Thus, there are at most $\lambda - 1$ critical sets in the optimum, and their total weight is at most $\delta m \cdot (\lambda - 1) = (1 - \delta)m$. Given that the total weight of all the sets is at least $m(1 - \epsilon^2)$, we know that the non-critical sets have a total weight at least $(\delta - \epsilon^2)m$, and total profit at most OPT . Hence, by selecting out least profitable (in terms of ratios, i.e., $p_i/w(S_i)$) non-critical sets such that their total weight is in $(\epsilon^2 m, 2\epsilon^2 m]$, we know their total profit is at most $2\epsilon^2 m \cdot OPT/(\delta m - \epsilon^2 m) \leq 2\epsilon OPT$ (by taking ϵ sufficiently small such that $\delta > 2\epsilon$). According to Lemma 4 the theorem is proved. ◀

3 Packing into an Arbitrary Number of Bins

Extending the PTAS [2] for the multiple knapsack problem, we have the following.

► **Theorem 5.** *There exists a dynamic programming algorithm for $GMKP(\delta)$ which returns a solution of profit $OPT(m)/(1 + \epsilon)$ by using $m(1 + \epsilon)$ bins.*

Notice that a feasible solution uses m bins, thus the above theorem actually ensures a feasible solution with profit at least $OPT(m(1 - \epsilon))/(1 + \epsilon)$. In the classical multiple knapsack problem, the profit is associated with items, hence for each bin in the solution of $OPT(m)$ we can calculate the total profit of items packed into this bin. If we delete ϵm bins with the least total profit of items, we obtain a solution of profit at least $(1 - \epsilon)OPT(m)$ with $m(1 - \epsilon)$ bins, implying that $OPT(m(1 - \epsilon)) \geq (1 - \epsilon)OPT(m)$, and a PTAS follows directly from Theorem 5. However, this inequality is no longer true when the profit is associated with groups instead of items. We will discuss in the following the approximability of $GMKP(\delta)$ with respect to the value of δ .

Throughout this section we let $OPT = OPT(m)$ for simplicity. We assume ϵ to be an arbitrary small fractional value such that $1/\epsilon$ is an integer, and m to be sufficiently large such that $m\epsilon$ is always an integer.

3.1 $\delta > 2/3$

► **Theorem 6.** *Assuming $P \neq NP$, there is no constant ratio approximation algorithm for the group packing problem $GMKP(\delta)$ when $\delta > 2/3$.*

Consider the Bin Packing Problem which asks whether a set of items of weights a_1, a_2, \dots, a_n could be packed into m bins of capacity 1. We denote by $BPP(\delta)$ if $\sum a_j \leq \delta m$. Theorem 6 follows directly from the following lemma.

► **Lemma 7.** *$BPP(\delta)$ is strongly NP-complete for $\delta > 2/3$.*

Proof. We reduce from 3-Partition. In the 3-Partition problem, we are given a set of $3k$ positive integers $\{b_1, b_2, \dots, b_{3k}\}$ such that $\sum b_j = kB$. The problem asks whether there exists a partition of the integers into k disjoint subsets U_1, U_2, \dots, U_k such that for every i , $|U_i| = 3$ and $\sum_{b_j \in U_i} b_j = B$.

Let $\epsilon > 0$ be an arbitrarily small positive number with $1/\epsilon$ being an integer. Given a 3-Partition instance, we let $b'_i = b_i + B/\epsilon$ and $B' = (1 + 3/\epsilon)B$. We construct an instance of $BPP(\delta)$ with $\delta = 2/3 + O(\epsilon)$ in the following way.

There are $3k$ key items of weights $a_i = b'_i/B'$ for $1 \leq i \leq 3k$. There are $2k/\epsilon$ dummy items, each of weight $(B + B/\epsilon)/B'$. There are $m = k + k/\epsilon$ bins, each of capacity 1. Hence the total weight of items is $(k\epsilon + 5k + 2k/\epsilon)/(\epsilon + 3) \leq (2/3 + O(\epsilon))(k + k/\epsilon)$, i.e., it is a feasible instance of $BPP(\delta)$ for $\delta = 2/3 + O(\epsilon)$.

Suppose the 3-Partition problem admits a feasible solution. Then the bin packing problem also admits a feasible solution by packing all the key items into k bins, and all the dummy items into k/ϵ bins.

Suppose the bin packing problem admits a feasible solution. It is easy to verify that there are three possibilities with respect to the items packed into a single bin. A bin contains only key items, and there are at most three of them, or it contains only dummy items, and there are at most two of them, or it contains one key item and one dummy item. Let x, y, z denote the number of bins with the above-mentioned three kinds of “configuration”, respectively. We have the following constraints,

$$3x + z \geq 3k, \quad 2y + z \geq 2k/\epsilon, \quad x + y + z = m = k + k/\epsilon.$$

Let $z = k + k/\epsilon - x - y$ and plug it back into the first two inequalities, simple calculations show that $x \geq k$ and $y \geq k/\epsilon$. Given that $x, y, z \geq 0$, it follows directly that $x = k$ and $y = k/\epsilon$. Hence, all the key items are packed into k bins, implying a solution to the 3-Partition problem. ◀

3.2 $1/3 < \delta \leq 2/3$

► **Theorem 8.** *Assuming $P \neq NP$, for any $\epsilon > 0$ there is no $(1/2 + \epsilon)$ -approximation algorithm for $GMKP(\delta)$ when $\delta > 1/3$.*

Proof. Recall that the proof of Lemma 7 shows that it is strongly NP-hard to decide whether a group of items of total weight $(2/3 + O(\epsilon))m$ could be packed into m bins. To modify it into a feasible instance of $GMKP(\delta)$ for $1/3 < \delta \leq 2/3$, we divide these items into two groups with roughly the same total weight via a simple greedy algorithm, i.e., we open two groups A and B which are initially empty, and each time we add one item into the group with a smaller total weight of items. By doing so items could be divided such that the difference of the total weight between two groups is at most the weight of the largest item, which is $O(\epsilon m)$. Hence, $w(A), w(B) \leq (1/3 + O(\epsilon))m$. Let the profit of either group be 1. If there exists a $(1/2 + \epsilon)$ -approximation algorithm, then it returns a solution with profit strictly larger than 1 if the two groups of items can both be packed into m bins, and returns a solution with profit at most 1 otherwise. Hence, we can use the approximation algorithm to decide whether all the items could be packed into m bins, which is a contradiction to Lemma 7. ◀

We complement Theorem 8 by providing a $(1/2 - \epsilon)$ -approximation algorithm for $GMKP(\delta)$ when $\delta \leq 2/3$. To achieve this, we first consider $BPP(\delta)$.

► **Lemma 9.** *$BPP(\delta)$ is polynomial-time solvable when $\delta \leq 2/3$.*

The Lemma actually falls as corollary of the following observation for the Longest Processing Time (LPT) algorithm for the *Machine Scheduling* problem. In the machine scheduling problem, given is a set of jobs, each of processing time p_j , and the goal is to assign these jobs onto parallel machines such that the completion time of the job that completes last is minimized. LPT is the algorithm that orders jobs in non-increasing order of their processing times, and always assigns a job to the machine with the least load.

► **Lemma 10** ([5]). *If every job has a processing time larger than $OPT/3$ where OPT is the optimum makespan, then LPT produces an optimal schedule.*

Proof of Lemma 9. Suppose the optimum uses m bins. We show that FFD (First Fit Decreasing) [10] uses no more than m bins. FFD is the algorithm that assigns items into bins in the following way; it sorts items by weight from the largest to the smallest and sorts bins in an arbitrary way. Then it packs each item into the first bin that still has the enough remaining capacity to accommodate it. Consider all the items larger than $1/3$. FFD packs them into no more than m bins via Lemma 10. For the remaining items, if FFD opens an $(m + 1)$ -st bin for some item j , then at this time all the m bins are filled up to at least $2/3$, hence the total weight of the items, except item j , is at least $2/3m$, which is a contradiction. ◀

Notice that the proof of the above lemma also shows that items of total weight W can always be packed into $\lceil 3/2 \cdot W \rceil$ bins, if every item has a weight no more than $1/2$. To see why, consider items of weight larger than $1/3$. FFD can always pack two of them into one bin. Thus in the solution returned by FFD, except for one bin, every bin is filled up to at least $2/3$. This observation leads to the following lemma.

► **Lemma 11.** *A set of items can always be packed into $|S_{>1/2}| + \lceil 3/2 \cdot W_{\leq 1/2} \rceil$ bins, where $S_{>1/2}$ is the set of items whose weight is strictly larger than $1/2$, and $W_{\leq 1/2}$ is the total weight of the remaining items.*

Now we are ready to prove the following theorem.

► **Theorem 12.** *There exists a $(1/2 - \epsilon)$ -approximation algorithm for $GMKP(\delta)$ when $\delta \leq 2/3$.*

The proof idea is to show that, all the sets selected by the optimum solution of $GMKP(\delta)$ could be divided into two groups such that either group could be packed into αm bins with some constant $\alpha < 1$. If the above claim is true, then $OPT/2$ could be achieved by using at most αm bins, and we could apply Theorem 5 to derive a feasible solution of profit at least $(1/2 - \epsilon)OPT$.

Proof. Consider an optimum solution. A set is called huge if its weight is at least ϵm . There are at most $1/\epsilon$ huge sets in the optimum solution and we can guess them (by enumeration). Suppose we guess the correct sets and let them be S_1 to S_h . We partition them into two groups such that either group has a total weight at most $2/3 \cdot m$. This could be achieved via a simple greedy strategy, i.e., we treat each set S_i as a job of processing time $w(S_i)$ and apply LPT (longest processing time first) to schedule them on two identical machines. The makespan of the solution returned is either $\delta m \leq 2/3 \cdot m$ if there are only one or two jobs, or at most $1/2(\sum_{i=1}^h w(S_i) - w(S_j)) + w(S_j) \leq 1/2 \cdot m + 1/2 \cdot 1/3m \leq 2/3 \cdot m$ where S_j is the job that finishes last and hence of weight at most $m/3$. Let A and B denote the two groups returned by the above procedure. Let C be the group of remaining sets in the optimum solution, then each set of C has a weight at most ϵm .

Note that groups A and B are known via guessing (enumeration), while the group C is unknown. Furthermore, the total weight of items in group A (or B) is at most $2/3 \cdot m$. Thus according to Lemma 9, all the items of A (or B) could be packed into m bins. If the total profit of sets in A (or B) is at least $OPT/2$, the theorem is proved.

Otherwise, we prove the theorem using Theorem 5. Consider items of weight larger than $1/2$. Let z_A , z_B and z_C be the number of such items in groups A , B and C respectively. Let W_A , W_B and W_C be the total weight of remaining items in groups A , B and C . We have the following inequalities.

$$\begin{aligned} z_A + z_B + z_C &\leq m \\ 1/2 \cdot (z_A + z_B + z_C) + W_A + W_B + W_C &\leq m \end{aligned}$$

According to the above two inequalities, we have

$$z_A + z_B + z_C + 3/2(W_A + W_B + W_C) \leq 7/4 \cdot m.$$

According to Lemma 11, to pack items of group A or group B we need at most $z_A + \lceil 3/2 \cdot W_A \rceil \leq z_A + 3/2 \cdot W_A + 1$ or $z_B + 3/2 \cdot W_B + 1$ bins, respectively. There are two possibilities.

Case 1. Either $z_A + 3/2 \cdot W_A + 1$ or $z_B + 3/2 \cdot W_B + 1$ is very large, i.e., at least $7/8 \cdot m$. Assume w.l.o.g that $z_A + 3/2 \cdot W_A + 1 \geq 7/8 \cdot m$. Recall that the profit of A is less than $OPT/2$, hence the profit of $B \cup C$ is at least $OPT/2$. Notice that $z_A + 3/2 \cdot W_A + 1 \geq 7/8 \cdot m$ implies that $z_B + z_C + 3/2 \cdot (W_B + W_C) + 1 \leq 7/8 \cdot m + 2 \leq (7/8 + \epsilon)m$, hence the sets in $B \cup C$ can be packed into $(7/8 + \epsilon)m$ bins via Lemma 11, which implies that $OPT(m(1-\epsilon)) \geq OPT((7/8 + \epsilon)m) \geq 1/2OPT$. Using Theorem 5 we know that the dynamic programming algorithm will return a feasible solution with profit at least $(1/2 - \epsilon)OPT$.

Case 2. $z_A + 3/2 \cdot W_A + 1 \leq 7/8 \cdot m$ and $z_B + 3/2 \cdot W_B + 1 \leq 7/8 \cdot m$. We claim that C could be partitioned into C_1 and C_2 such that $A' = A \cup C_1$, $B' = B \cup C_2$, $z_{A'} + 3/2 \cdot W_{A'} + 1 \leq (7/8 + 2\epsilon)m$ and $z_{B'} + 3/2 \cdot W_{B'} + 1 \leq (7/8 + 2\epsilon)m$ (Here $z_{A'}$, $z_{B'}$, $W_{A'}$ and $W_{B'}$ are defined analogously as before). If the claim is true, then either A' or B' has a profit at least $OPT/2$, implying that $OPT(m(1 - \epsilon)) \geq OPT((7/8 + 2\epsilon)m) \geq 1/2 OPT$, and Theorem 12 is proved. To see why the claim holds, we consider the sets in C and let them be S_1 to S_h . We let $z_C(S_i)$ be the number of items with weight larger than $1/2$ in S_i , and $W_C(S_i)$ be the total weight of remaining items in S_i . As group C consists of sets whose weight is at most ϵm , we have $1/2 \cdot z_C(S_i) + W_C(S_i) \leq \epsilon m$ for $1 \leq i \leq h$. To show the partition of C we again view each set S_i as a job of processing time $z_C(S_i) + 3/2 \cdot W_C(S_i) \leq 2\epsilon m$. We shall schedule these jobs onto two identical machines with the initial load of $z_A + 3/2 \cdot W_A + 1 \leq 7/8 \cdot m$ and $z_B + 3/2 \cdot W_B + 1 \leq 7/8 \cdot m$, respectively. Applying List-Scheduling, we claim that after all the jobs are scheduled, the makespan is at most $(7/8 + 2\epsilon)m$ since otherwise, the job that finishes last must be some job S_i , and thus the load of either machine is strictly larger than $7/8 \cdot m$, which contradicts the fact that $z_A + z_B + z_C + 3/2(W_A + W_B + W_C) = z_A + z_B + 3/2(W_A + W_B) + \sum_i (z_C(S_i) + 3/2 W_C(S_i)) \leq 7/4 \cdot m$. Taking the sets scheduled on two machines as A' and B' , we obtain the desired partition. \blacktriangleleft

3.3 $\delta \leq 1/3$

With a similar proof as for Theorem 8, we have the following lower bound.

► **Theorem 13.** *Assuming $P \neq NP$, there is no $(1 - 3\delta/(2 + 3\delta) + O(\epsilon))$ -approximation algorithm for $GMKP(\delta)$ for any $\epsilon > 0$ when $\delta \leq 1/3$.*

We complement Theorem 13 with the following theorem.

► **Theorem 14.** *Given an arbitrary $\epsilon > 0$, there exists a $(1 - 2\delta - O(\epsilon))$ -approximation algorithm for $GMKP(\delta)$ when $\delta \leq 1/3$.*

By Theorem 5, it suffices to prove $OPT((1 - \epsilon)m) \geq (1 - 2\delta - O(\epsilon))OPT(m)$, as is shown by the following Lemma 15.

We remark that, although intuitively one might expect to show that $OPT(m(1 - \epsilon)) \geq (1 - O(\epsilon))OPT(m)$, or at least $OPT((1 - \epsilon)m) \geq (1 - \delta - O(\epsilon))OPT(m)$ for sufficiently small δ , Theorem 13 already implies that $OPT((1 - \epsilon)m) \geq (1 - c\delta)OPT(m)$ does not hold in general for $c < 3/2$.

► **Lemma 15.** *$OPT((1 - \Theta(\epsilon^2))m) \geq (1 - 2\delta - O(\epsilon))OPT(m)$ for $m \geq 20/\epsilon^3$.*

We remark that the above lemma is actually true for any $\delta \in (0, 1]$: for $\delta > 1/2$ it is trivially true, while for $\delta \in (1/4, 1/2]$ although a $(1 - 2\delta - O(\epsilon))$ -approximation algorithm follows, yet the $(1/2 - \epsilon)$ -approximation algorithm presented in the previous subsection performs better.

We give a brief introduction to the proof. Consider the solution with the profit of $OPT(m)$. In order to prove the inequality, among the sets of items packed in this solution, we need to select some sets such that their total profit is small (at most $(2\delta + O(\epsilon))OPT(m)$), and the deletion of them saves many bins (at least $\Omega(\epsilon^2 m)$ bins). Obviously these sets could not be the sets that consist of items that are very small. To see why, imagine that in $OPT(m)$ each bin is filled up by a huge item of size larger than $1/2$ and a bunch of small items, then even if we delete all the small items the number of bins required for the remaining huge items is still m . Hence, we should better delete sets that contain many big items. To show that such a deletion, combined with the repacking of remaining items could eventually save a significant

number of bins, we will iteratively modify the instance and then apply the discrepancy theory to the Gilmore Gomory LP relaxation [4] for the modified instance. The idea of applying discrepancy theory to Bin Packing is also used in [3] to derive the relationship between Bin Packing and the three-permutation-problem.

Proof. We assume that sets packed in $OPT(m)$ are S_1 to S_h . We further assume that $\sum_i w(S_i) > (1/2 - \epsilon)m$ since otherwise $OPT((1 - \epsilon)m) = OPT(m)$. To see why, suppose $\sum_i w(S_i) \leq (1/2 - \epsilon)m$. We let $S_{1/2}$ be the set of items in S_1 to S_h whose weight is larger than $1/2$, and $W_{\leq 1/2}$ be the total weight of remaining items. Then $1/2|S_{>1/2}| + W_{\leq 1/2} \leq \sum_i w(S_i) \leq (1/2 - \epsilon)m$, whereas $|S_{>1/2}| + \lceil 3/2W_{\leq 1/2} \rceil \leq \sum_i w(S_i) \leq 2(1/2 - \epsilon)m + 1 \leq (1 - \epsilon)m$. According to Lemma 11 all the sets could be packed into $(1 - \epsilon)m$ bins, hence $OPT((1 - \epsilon)m) = OPT(m)$.

From now on we will abuse the notation w_i a bit to also denote item i , and we may also abuse the notation $OPT(m)$ to denote the solution that achieves the profit. Let w_1 to w_n be all the items of S_1 to S_h such that $w_1 \geq w_2 \geq \dots \geq w_n$. Let γ be the least index such that $w_1 + w_2 + \dots + w_\gamma > (1/2 - \epsilon)m$. Obviously w_1 to w_γ should belong to at least $\lceil (1/2 - \epsilon)/\delta \rceil$ different sets among S_1 to S_h . For simplicity let these sets be S_1 to S_ℓ with $\ell \geq \lceil (1/2 - \epsilon)/\delta \rceil$.

Let $S^\gamma = \{w_1, w_2, \dots, w_\gamma\}$, $S_i^\gamma = S_i \cap S^\gamma$, $w(S_i^\gamma) = \sum_{j \in S_i^\gamma} w_j$, $p(S_i^\gamma) = p_i$, $\rho(S_i^\gamma) = p_i/w(S_i^\gamma)$. We assume w.l.o.g that $\rho(S_1^\gamma) \geq \rho(S_2^\gamma) \geq \dots \geq \rho(S_\ell^\gamma)$.

Consider the following knapsack problem. We take each S_i^γ as a single item. Then these ℓ items can be packed into a knapsack of capacity $\sum_{i=1}^\ell w(S_i^\gamma) \geq (1/2 - \epsilon)m$ with the total profit of $\sum_{i=1}^\ell p_i \leq OPT(m)$. Recall that $w(S_i^\gamma) \leq \delta m$. We let $\ell' \leq \ell$ be the least index such that $w(S_{\ell'}^\gamma) + w(S_{\ell'+1}^\gamma) + \dots + w(S_\ell^\gamma) \in (8\epsilon m, (8\epsilon + \delta)m]$. Furthermore, since $S_{\ell'}^\gamma$ to S_ℓ^γ are the least profitable items (in terms of ratios), we know that

$$\frac{\sum_{i=\ell'}^\ell p_i}{\sum_{i=\ell'}^\ell w(S_i^\gamma)} \leq \frac{\sum_{i=1}^\ell p_i}{\sum_{i=1}^\ell w(S_i^\gamma)} \leq \frac{\sum_{i=1}^\ell p_i}{(1/2 - \epsilon)m},$$

$$\sum_{i=\ell'}^\ell p_i \leq (8\epsilon + \delta)m \cdot \frac{\sum_{i=1}^\ell p_i}{(1/2 - \epsilon)m} \leq (2\delta + O(\epsilon)) \sum_{i=1}^\ell p_i \leq (2\delta + O(\epsilon))OPT(m).$$

Suppose we delete sets $S_{\ell'}$ to S_ℓ from the optimum solution $OPT(m)$. The total profit of the remaining sets is at least $(1 - 2\delta - O(\epsilon))OPT(m)$, and in the following we show that to pack all the items of the remaining sets, $(1 - \Theta(\epsilon^2))m$ bins suffice, which proves the lemma.

Notice that directly deleting items of sets $S_{\ell'}$ to S_ℓ from the solution of $OPT(m)$ leaves some empty space in the m bins, and we aim to somehow merge these spaces to create $\Theta(\epsilon^2 m)$ empty bins. Instead of iteratively moving items, we will use a ‘‘global approach’’ by applying the discrepancy theory to the configuration LP for the bin packing problem.

Consider the instance of packing items w_1, w_2, \dots, w_n . For any set of items X , we denote by $\sigma(X)$ the minimum number of bins needed to pack them. Let $S = \{w_1, w_2, \dots, w_n\}$, $S' = \cup_{i=\ell'}^\ell S_i^\gamma \subseteq S^\gamma$. It is easy to see that $\sigma(S) \leq m$, $w(S') \in (8\epsilon m, (8\epsilon + \delta)m]$. To prove the lemma, it suffices to prove Claim 1.

► **Claim 1.** $\sigma(S \setminus S') \leq (1 - \Theta(\epsilon^2))m$.

Consider w_γ . We claim that, if $w_\gamma \leq 2\epsilon$, then $\sigma(S \setminus S') \leq (1 - \epsilon)m \leq (1 - O(\epsilon^2))m$. To see why, recall the definition of γ , we have $w_1 + w_2 + \dots + w_{\gamma-1} \leq (1/2 - \epsilon)m$, implying that these items could be packed into $(1 - \epsilon)m$ bins. We now delete items of $S' \subseteq \{w_1, w_2, \dots, w_\gamma\}$ from this solution, and then pack items w_γ to w_n via First-Fit. We claim that, we do not

need to open new bins. Suppose the claim is not true, then among these $(1 - \epsilon)m$ bins at least $(1 - \epsilon)m - 1$ bins are filled up to at least $1 - 2\epsilon$. Hence, $((1 - \epsilon)m - 1)(1 - 2\epsilon) \leq w(S \setminus S') \leq (1 - 8\epsilon)m$, which is a contradiction.

From now on we assume $w_\gamma > 2\epsilon$. In this case we do not prove Claim 1 directly. In the following, we will iteratively give Claim 2 to Claim 5 and show that, for $1 \leq i \leq 4$, Claim $i + 1$ implies Claim i . We then prove Claim 5 at the end, which suffices to show the truth of Claim 1, and consequently the lemma.

Consider small items whose weight is at most ϵ . We modify small items in the following way. We iteratively agglomerate small items into a big item of weight $[\epsilon, 2\epsilon)$. At last there may still be some small items left with total weight less than ϵ , and we simply agglomerate them into a single item. Let S^\sharp be the sets of modified items, then it is easy to see that except at most one item, each item in S^\sharp has a weight at least ϵ , and $w(S) = w(S^\sharp)$. Furthermore, if we order items of S^\sharp in non-increasing order of their weight, the first γ items would still be w_1 to w_γ . Hence, $S' \subseteq S^\gamma \subseteq S^\sharp$. As the modification procedure only agglomerate items, to prove Claim 1, it suffices to prove the following Claim 2.

► **Claim 2.** $\sigma(S^\sharp \setminus S') \leq (1 - \Theta(\epsilon^2))m$.

Notice that $w(S') \in (8\epsilon m, (8\epsilon + \delta)m]$ and the weight of each item is at most $1 \leq \epsilon m$. We can easily split S' into S'_1 and S'_2 such that $w(S'_1) \in [4\epsilon m, 5\epsilon m)$ and $w(S'_2) \geq 3\epsilon m$. As items are agglomerated, it is no longer true that $\sigma(S^\sharp) \leq m$. However, we claim that, $\sigma(S^\sharp \setminus S'_1) \leq m$. To see why, consider the solution of $\sigma(S) \leq m$. We take out all the small items together with items of S'_1 . Now we add back the agglomerated items via First-Fit. We claim that, we do not need to open new bins since otherwise, at least m bins are filled up to at least $1 - 2\epsilon$, implying that $w(S^\sharp \setminus S'_1) \geq (1 - 2\epsilon)m$, which is a contradiction as $w(S^\sharp \setminus S'_1) = w(S) - w(S'_1) \leq (1 - 4\epsilon)m$.

Let $S^\sharp = S^\gamma \cup S_\alpha$. Claim 2 is equivalent to $\sigma(S^\sharp \setminus S') = \sigma((S^\gamma \setminus (S'_1 \cup S'_2)) \cup S_\alpha) \leq (1 - \Theta(\epsilon^2))m$. By re-indexing items we assume that $S^\gamma \setminus S'_1 = \{w_1, w_2, \dots, w_{\gamma'}\}$ for some $\gamma' < \gamma$, and $S_\alpha = \{w_{\gamma'+1}, w_{\gamma'+2}, \dots, w_{n'}\}$ where $w_{\gamma'+1} \geq w_{\gamma'+2} \geq \dots \geq w_{n'}$. As $w(S'_2) \geq 3\epsilon m$, S'_2 consists at least $3\epsilon m$ items of $S^\gamma \setminus S'_1$. Instead of deleting items of S'_2 , we consider the instance of deleting $3\epsilon m$ largest items from S_α , i.e., deleting $\hat{S}_\alpha = \{w_{\gamma'+1}, \dots, w_{\gamma'+3\epsilon m}\}$ (if $n' \leq \gamma' + 3\epsilon m$ then $\hat{S}_\alpha = S_\alpha$). Compare $(S^\gamma \setminus (S'_1 \cup S'_2)) \cup S_\alpha$ with $(S^\gamma \setminus S'_1) \cup (S_\alpha \setminus \hat{S}_\alpha)$. Since there are at least $3\epsilon m$ items in S'_2 , each being larger than (or equal to) any item in \hat{S}_α , we know there exists an injection such that each item in $(S^\gamma \setminus (S'_1 \cup S'_2)) \cup S_\alpha$ could be mapped to a larger or equal item in $(S^\gamma \setminus S'_1) \cup (S_\alpha \setminus \hat{S}_\alpha)$. Hence, to prove Claim 2, it suffices to prove the following Claim 3.

► **Claim 3.** $\sigma((S^\gamma \setminus S'_1) \cup (S_\alpha \setminus \hat{S}_\alpha)) \leq (1 - \Theta(\epsilon^2))m$.

Recall that $\sigma((S^\gamma \setminus S'_1) \cup S_\alpha) \leq m$. Consider a feasible solution of packing items of $(S^\gamma \setminus S'_1) \cup S_\alpha$ into m bins (empty bins are allowed). We say a bin is critical if items from $S^\gamma \setminus S'_1$ occupy the space of at most $1/2$, and non-critical otherwise. Hence, there are at most $(1 - \epsilon)m$ non-critical bins since otherwise the total weight of items from $S^\gamma \setminus S'_1$ is larger than $(1 - \epsilon)m/2 \geq (1/2 - \epsilon)m \geq w(S^\gamma)$, which is a contradiction.

Let $\beta \geq \epsilon m$ be the number of critical bins. Let S^c be the set of items packed in critical bins, $S_\alpha^c = S^c \cap S_\alpha$ and $\tau = |S_\alpha^c|$. For simplicity let $w'_1 \geq w'_2 \geq \dots \geq w'_\tau$ be all the items of S_α^c . Let $\hat{S}_\alpha^c = \{w'_1, \dots, w'_{3\epsilon m}\}$ be the largest $3\epsilon m$ items in $S_\alpha^c \subseteq S_\alpha$. Compare \hat{S}_α and \hat{S}_α^c , i.e., the largest $3\epsilon m$ items in S_α and the largest $3\epsilon m$ items in $S_\alpha^c \subseteq S_\alpha$. Obviously there is an injection which maps each item in \hat{S}_α^c to a larger or equal item in \hat{S}_α . Hence to prove Claim 3 it suffices to prove the following Claim 4.

► **Claim 4.** $\sigma((S^\gamma \setminus S'_1) \cup (S_\alpha \setminus \hat{S}_\alpha^c)) \leq (1 - \Theta(\epsilon^2))m$.

A critical configuration is a configuration for items of S_α^c , represented by a column vector $\nu = (b_1, b_2, \dots, b_\tau)^T$, where $b_i \in \{0, 1\}$ denoting whether $w'_i \in S_\alpha^c$ is packed. Obviously the packing of each critical bin could be represented by some items of $S^\gamma \setminus S'_1$ together with a critical configuration, and there are β critical configurations corresponding to the β critical bins. Let $B = (\nu_1, \nu_2, \dots, \nu_\beta) \in \{0, 1\}^{\tau \times \beta}$ be the matrix of these β configurations. Then obviously $Be_\beta = e_\tau$ where e_k is a column vector with k components, each being 1.

We now use the idea of [3] to re-write the equation $Be_\beta = e_\tau$. Let matrix A be defined as $A_i = \sum_{j=1}^i B_j$ where A_i (B_j , resp.) denotes the i -th (j -th, resp.) row of the matrix A (B , resp.), i.e., A_{ij} denotes the total number of items w'_1 to w'_i in the j -th configuration ν_j . Then from $Be_\beta = e_\tau$ we derive $Ae_\beta = (1, 2, \dots, \tau)^T$. Furthermore, since each items, except the smallest one, is of weight at least ϵ , each configuration consists at most $1/\epsilon$ items. As each column of A is monotone, A is a monotone matrix with each entry $A_{ij} \in \{0, 1, 2, \dots, 1/\epsilon\}$. Hence, A is a $1/\epsilon$ -monotone matrix. Let A' be the matrix of attaching $A_{\tau+1} = (1/\epsilon, 1/\epsilon, \dots, 1/\epsilon)^T$ as the new last row of A , then A' is also monotone with

$$A'e_\beta = (1, 2, \dots, \tau, \beta/\epsilon)^T.$$

► **Claim 5.** There exists a 0-1 vector $x = (x_1, x_2, \dots, x_\beta)^T$ such that $A'x = (\psi, (\beta - \Delta)/\epsilon)^T$, where $\psi = Ax$ is a vector with i -th component $\psi_i \geq \max\{0, i - 3\epsilon m\}$ for $1 \leq i \leq \tau$, and $\Delta = \Omega(\epsilon^2 m)$.

We prove Claim 5 implies Claim 4 by showing that items of $(S^\gamma \setminus S'_1) \cup (S_\alpha \setminus \hat{S}_\alpha^c)$ could be packed into $\beta - \Delta/2$ bins. Indeed, $Ax = \psi$ means by using $x_i \in \{0, 1\}$ copies of the configuration ν_i (i.e., the i -th column of B), we can pack a subset $S^* \subseteq S_\alpha^c = \{w'_1, \dots, w'_\tau\}$ of items such that $|S^* \cap \{w'_1, \dots, w'_i\}| = \psi_i$. As $\psi_i \geq \max\{0, i - 3\epsilon m\}$, using these configurations we are able to pack items of $S_\alpha^c \setminus \hat{S}_\alpha^c$. Furthermore, $\sum x_i = \beta - \Delta$ means that in total we save Δ critical configurations, by removing which we get Δ bins which are at most half full since in a critical bin, items that are not in the critical configuration have a total weight at most $1/2$. Hence, we can merge items of two such bins into one bin, i.e., we can save $\Delta/2$ bins and Claim 4 follows if $\Delta = \Omega(\epsilon^2 m)$.

We have shown so far that Claim $i + 1$ implies Claim i for $1 \leq i \leq 4$, hence Claim 1, and consequently the lemma, will follow from the truth of Claim 5. We now prove Claim 5. We show there exists such an integer solution x . Consider the fractional solution $y = \theta e_\beta$ where $\theta = 1 - \epsilon m/\tau$. Obviously $A'y = (\theta, 2\theta, \dots, \tau\theta, \beta\theta/\epsilon)^T$. According to the discrepancy theory [12] there exists an integer solution $x \in \{0, 1\}^\beta$ such that

$$\|A'x - A'y\|_\infty \leq \text{lin}disc(A').$$

It is shown in [3] that for k -monotone $m \times n$ matrices the linear discrepancy is bounded by $5k \log_2(2 \min\{m, n\})$, hence we have

$$\text{lin}disc(A') \leq 5/\epsilon \cdot \log_2(2 \min\{\tau + 1, \beta\}) \leq 5/\epsilon \cdot \log_2(2m/\epsilon),$$

i.e., $\|A'x - A'y\|_\infty \leq 5/\epsilon \cdot \log_2(2m/\epsilon) = d$. Let $A'x = (\psi, \omega)$, then $\psi_i \geq \max\{0, i\theta - d\}$. For $i \geq 3\epsilon m$, $m \geq 20/\epsilon^3$, we have

$$i\theta - d \geq i(1 - \epsilon m/\tau) - 5/\epsilon \cdot \log_2(2m/\epsilon) \geq i - \epsilon m - 5/\epsilon \cdot \log_2(2m/\epsilon) \geq i - 3\epsilon m.$$

For ω , we have $\omega \leq \beta\theta/\epsilon + d$. Since each item, except for the smallest one, has a weight at least ϵ , we have $\tau \leq \beta/\epsilon + 1$, and thus $\theta \leq 1 - \epsilon m/(\beta/\epsilon + 1) \leq 1 - \epsilon^2 m/(2\beta)$. Recall that

$\beta \geq \epsilon m$, for $m \geq 20/\epsilon^3$ we have

$$\omega \leq \beta\theta/\epsilon + d \leq \beta/\epsilon - \epsilon m/2 + 5/\epsilon \cdot \log_2(2m/\epsilon) \leq \beta/\epsilon - \epsilon m/4.$$

Thus, $\Delta \geq \epsilon^2 m/4 = \Omega(\epsilon^2 m)$ as we desired. ◀

References

- 1 R. Adany, M. Feldman, E. Haramaty, R. Khandekar, B. Schieber, R. Schwartz, H. Shachnai, and T. Tamir. All-or-nothing generalized assignment with application to scheduling advertising campaigns. In *Proc. of IPCO 2013*, pages 13–24, 2013.
- 2 C. Chekuri and S. Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 35(3):713–728, 2006.
- 3 F. Eisenbrand, D. Pálvölgyi, and T. Rothvoss. Bin packing via discrepancy of permutations. *ACM Trans. Algorithms*, 9(3):39–49, 2013.
- 4 P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9:39–49, 1961.
- 5 R. L. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. Appl. Math.*, 17(2):416–429, 1969.
- 6 K. Jansen. Parameterized approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 39(4):1392–1412, 2009.
- 7 K. Jansen. A fast approximation scheme for the multiple knapsack problem. In *Proc. of SOFSEM'12*, pages 313–324, 2012.
- 8 K. Jansen, S. Kratsch, D. Marx, and I. Schlotter. Bin packing with fixed number of bins revisited. *J. Comput. Syst. Sci.*, 79(1):39–49, 2013.
- 9 K. Jansen, F. Land, and K. Land. Bounding the running time of algorithms for scheduling and packing problems. In *Proc. of WADS'13*, pages 313–324, 2013.
- 10 D. S. Johnson. *Near-optimal bin-packing algorithms*. Doctoral Thesis. MIT Press, 1973.
- 11 H. Kellerer. A polynomial time approximation scheme for the multiple knapsack problem. In *Proc. of APPROX'99*, pages 51–62, 1999.
- 12 J. Matousek. *Geometric discrepancy*. Springer-Verlag, 1999.