Improved Approximation for Vector Bin Packing

Nikhil Bansal* Marek Eliáš* Arindam Khan†

Abstract

We study the d-dimensional vector bin packing problem, a well-studied generalization of bin packing arising in resource allocation and scheduling problems. Here we are given a set of d-dimensional vectors $v_1, \ldots, v_n$ in $[0,1]^d$, and the goal is to pack them into the least number of bins so that for each bin B, the sum of the vectors in it is at most 1 in every dimension, i.e., $||\sum_{v_i \in B} v_i||_\infty \leq 1$. For the 2-dimensional case we give an asymptotic approximation guarantee of $1 + \ln(1.5) + \epsilon \approx (1.405 + \epsilon)$, improving upon the previous bound of $1 + \ln 2 + \epsilon \approx (1.693 + \epsilon)$. We also give an almost tight $(1.5 + \epsilon)$ absolute approximation guarantee, improving upon the previous bound of 2 [23]. For the d-dimensional case, we get a $1.5 + \ln(\frac{d+1}{2}) + \epsilon \approx 0.807 + \ln(d+1) + \epsilon$ guarantee, improving upon the previous $(1 + \ln d + \epsilon)$ guarantee [2]. Here $(1 + \ln d)$ was a natural barrier as rounding-based algorithms can not achieve better than d approximation. We get around this by exploiting various structural properties of (near)-optimal packings, and using multi-objective multi-budget matching based techniques and expanding the Round & Approx framework to go beyond rounding-based algorithms. Along the way we also prove several results that could be of independent interest.

1 Introduction

We consider the *d-dimensional vector bin packing* problem (d-VP) defined as follows: Given a collection of d-dimensional vectors $v_1, \ldots, v_n \in [0,1]^d$, partition them into the minimum number of feasible sets $B_1, \ldots, B_m$, where a set B_j is feasible if the sum of the vectors contained in it does not exceed 1 in any coordinate, i.e., $\|\sum_{v \in B_j} v\|_\infty \leq 1$. Usually, the vectors $v_1, \ldots, v_n$ are referred to as items and the sets $B_1, \ldots, B_m$ as bins.

d-VP is a natural generalization of the classical bin packing problem (which corresponds to $d = 1$) to the multiple resource setting. If we view the bins as servers with d different resources (processing power, memory, bandwidth etc.) and each item as a job requiring some specified amount of each resource, then d-VP corresponds to the problem of scheduling the jobs feasibly on the fewest number of servers. Starting from the classical work of Garey et al. [15], d-VP and its variants[1] such as the makespan minimization version (aka vector scheduling) have been studied extensively (e.g. [13, 8, 23, 34, 2]) both in the offline and online settings. Even the two-dimensional case has received a lot of attention motivated by novel applications in layout design, logistics, loading and scheduling [32, 33, 7, 17, 31, 30]. In recent years, d-VP has been revisited in connection with virtual machine placement in cloud computing [27, 4, 1, 26, 6, 20].

In this paper, we focus on the offline problem. Already for bin packing an easy reduction from *Partition* shows that it is NP-hard to determine whether the items can be packed into two bins or not, and hence it cannot be approximated better than 3/2. So, as is usual for bin-packing, we say that an algorithm A has *asymptotic* approximation ratio ρ if $\mathsf{A}(I) \leq \rho \cdot \mathsf{Opt}(I) + o(\mathsf{Opt}(I))$ for any instance I. Here, $\mathsf{Opt}(I)$ denotes the optimum value for I. If no $o(\mathsf{Opt}(I))$ term is allowed, we call it an *absolute* approximation ratio. If, for any given constant $\epsilon > 0$, there is an algorithm with $\rho = (1 + \epsilon)$, we say that the problem admits an asymptotic polynomial-time approximation scheme (APTAS).

Previous Results: For classical bin packing, de la Vega and Lueker [10] gave the first APTAS based on an elegant *linear grouping* technique. This result was improved by Karmarkar and Karp [22] based on iterated rounding techniques. The current best result is due to Hoberg and Rothvoss [19] based on connections to discrepancy [12, 29] and achieves a logarithmic additive error.

However, d-VP behaves quite differently for $d > 1$. Already for $d = 2$, Woeginger [34] showed that no APTAS exists unless P = NP (however, the lower bound on the ratio is quite small, less than 1.01). This is still the best known lower bound for any constant d. Interestingly, much stronger lower bounds are known

*Eindhoven University of Technology, Netherlands. Email: n.bansal@tue.nl, m.elias@tue.nl. Supported by NWO grant 639.022.211 and ERC consolidator grant 617951.

†Georgia Institute of Technology, Atlanta, GA, USA. Email: akhan67@gatech.edu. Supported by NSF EAGER award grants CCF-1415496 and CCF-1415498.

[1] There is another multi-dimensional generalization of bin packing called d-dimensional Geometric Bin Packing (GBP), where rectangular items are packed into unit-sized cubes [3]. However, we will not consider this here.

when d is part of the input. Based on a reduction from graph coloring [18], Chekuri and Khanna [8] showed that no $d^{1/2-\epsilon}$ approximation is possible for any $\epsilon > 0$, unless NP = ZPP. This can be improved to $d^{1-\epsilon}$ (this simple reduction[2] was pointed to us by Jan Vondrák). Throughout this paper we focus on the case when d is a fixed constant and not part of the input.

On the algorithmic side, a $d+\epsilon$ asymptotic approximation follows easily [10] from results for bin packing: Partition the items into d classes based on which coordinate of the item has the highest value, and apply the APTAS for one-dimensional case to each of these d classes separately. Interestingly, there is a very natural barrier for improving upon the ratio of d. In particular, any rounding-based algorithm, i.e., that rounds the sizes of large items into an $O_d(1)$ number of types (a widely used technique in bin packing and resource allocation problems), cannot give a better than d approximation (see Theorem 2.2). Thus any non-trivial algorithm must implicitly or explicitly consider the original (unrounded) sizes of items while packing them.

The first major result was obtained by Chekuri and Khanna [8] who gave an $O(\log d)$ asymptotic approximation for the problem. This was improved to $(1 + \ln d + \epsilon)$, without the big-O, by Bansal, Caprara, and Sviridenko, based on the so-called Round & Approx framework (R&A) [2]. This gives a bound of about 1.693 and 2.099 for $d = 2$ and $d = 3$, which are presumably the cases of most practical interest. The main idea of the R&A method (details in Section 2) is the following: if there is a "suitable" ρ approximation algorithm for a packing problem, then one obtains a $(1+\ln \rho+\epsilon)$ approximation. In the original approach of [2], the definition of a suitable algorithm involved a certain technical notion of subset-obliviousness. Later, Bansal and Khan [3] simplified and extended the applicability of this approach (in the context of geometric bin-packing) by showing that any rounding based algorithm is subset-oblivious. As rounding-based algorithms cannot beat d and extending R&A beyond such algorithms was unclear, this was a major barrier to improving the ratio of $1 + \ln(d)$.

Given the large gap between the known lower and upper bounds for the problem, it is of great interest to improve these bounds at least for small values of d.

Our Results and Techniques. The main result of this paper is the following theorem:

THEOREM 1.1. *For any small constant $\epsilon > 0$, there is a polynomial time algorithm with an asymptotic approximation ratio of $(1 + \ln(1.5) + \epsilon) \approx (1.405 + \epsilon)$ for 2-D vector packing.*

This gives a substantial improvement upon the current ≈ 1.693 bound for 2-D vector packing, but, more importantly, it overcomes the barrier of $1 + \ln d$ mentioned above.

Theorem 1.1 is based on two (perhaps surprising) ideas. First we give a $(1.5+\epsilon)$ asymptotic approximation for any $\epsilon > 0$, without the R&A framework. To do this, we show that there exists a "well-structured" 1.5-approximate solution, and then search (approximately) over the space of such solutions. However, as this structured solution (necessarily) uses unrounded item sizes, it is unclear how to search over the space of such solutions efficiently. So a key idea is to define this structure carefully based on matchings, and use an elegant recent algorithm for the multiobjective-multibudget matching problem by Chekuri, Vondrák, and Zenklusen [9]. As we show, this allows us to both use unrounded sizes and yet enumerate the space of solutions like in rounding-based algorithms. A more detailed overview can be found in Section 3.

The second step is to apply the R&A framework to the above algorithm, however, there are some difficulties. Firstly, the algorithm is not rounding-based. Secondly, even proving subset obliviousness for rounding based algorithms for d-VP is more involved than for geometric bin-packing. Roughly, in the geometric version, items with even a single small coordinate have small area, which makes it easier to handle, while in d-VP such skewed items can cause problems. To get around these issues, we use additional technical observations about the structure of d-VP.

Another consequence of the these techniques is the following tight (absolute) approximation guarantee[3], improving upon the guarantee of 2 by Kellerer and Kotov [23].

THEOREM 1.2. *For any small constant $\epsilon > 0$, there is a polynomial time algorithm with an absolute approximation ratio of $(1.5 + \epsilon)$ for 2-D vector packing.*

We extend the approach for $d = 2$ to give a $(d+1)/2+\epsilon$ approximation (for $d = 2$, this is precisely the 3/2 bound mentioned above). Then we show how to incorporate it into R&A. However, applying the R&A framework is more challenging here and instead of the

[2]Let G be a graph on n vertices. In the d-VP instance, there will be $d = n$ dimensions and n items, one for each vertex. For each vertex i, we create an item i that has size 1 in coordinate i, size $1/n$ in coordinate j for each neighbor j of i, and size 0 in all other coordinates. It is easy to verify that a set of items S can be packed in a bin if and only if S is an independent set in G.

[3]Recall that 3/2 is tight even for 1-VP via the Partition problem, and hence for 2-VP. So even though 1-VP and 2-VP have very different asymptotic approximability, they have very similar absolute approximability.

ideal $1+\ln((d+1)/2) \approx 0.307+\ln d$ we get the following $(\ln(d+1)+0.807)$-approximation:

THEOREM 1.3. *For any small constant $\epsilon > 0$, there is a polynomial time algorithm with an asymptotic approximation ratio of $(1.5 + \ln \frac{d+1}{2} + \epsilon) \approx \ln(d+1) + 0.807 + \epsilon$ for d-VP.*

This improves over the previous best approximation for all $d > 4$. Moreover, note that our bound of $(\frac{d+1}{2} + \epsilon)$-approximation also gives a $2+\epsilon$ approximation for $d = 3$, improving upon the previous bounds of $1 + \ln 3 + \epsilon \approx 2.099 + \epsilon$.

Along the way, we also prove several additional results which could be of independent interest. For example, in Section 5 we obtain several results related to resource augmented packing which has been studied for other variants of bin packing [21, 5]. We specifically show the following.

THEOREM 1.4. *There is a polynomial time algorithm for packing vectors into at most $(1 + 2\epsilon)\mathsf{Opt} + 1$ bins with ϵ resource augmentation in $(d-1)$ dimensions (i.e., bins have length $(1+\epsilon)$ in $(d-1)$ dimensions and 1 in the other dimension), where Opt denotes the minimum number of unit bins needed to pack these items.*

Organization of the paper. In Section 2, we state basic definitions and explain the main tools which are used throughout the paper. Section 3 describes the main strategy and intuition behind our results, whose actual exposition starts in Section 4. There we prove Theorem 1.4 by describing the proposed APTAS algorithm with resource augmentation. In Section 5, we present a preliminary approximation algorithm for d-VP with approximation ratio of $(d+1)/2$, which we later incorporate into R&A framework in Section 6, getting our final results for 2-dimensional case in Subsection 6.1 (Theorem 1.1) and d-dimensional case in Subsection 6.2 (Theorem 1.3). The algorithm with absolute approximation guarantee of 3/2 proposed in Theorem 1.2 is presented at the end of Section 5.

2 Preliminaries

Let us first fix some notation and definitions. We denote $[n] := \{1, 2, \ldots, n\}$, for $n \in \mathbb{N}$. An instance I of d-VP is specified by n items $\{v_1, v_2, \ldots, v_n\}$, where $v_i \in [0,1]^d$ for each $i \in [n]$. For $\ell \in [d]$, v_i^ℓ denotes ℓ-th coordinate of item v_i. We will use the terms dimension and coordinate interchangeably. Let β be a fixed parameter. We call an item *small*, if all its coordinates are smaller than β, otherwise it is *large*. A bin B has *slack* δ in dimension ℓ if $\sum_{v \in B} v^\ell \le (1-\delta)$. For a set S of items, let σ_S be the vector denoting the coordinate-wise sum of all vectors in S, i.e., $\sigma_S = \sum_{v_j \in S} v_j$. We denote $\sigma_S \le v$ if $\sigma_S^\ell \le v^\ell$ for all dimensions $\ell \in [d]$. Let L_B be the set of large items in a bin B. We call a *configuration of big items* in B to be the vector $\sum_{v_i \in L_B} v_i$. A *configuration of small items* is the remaining space in the bin, i.e., vector $\mathbb{1} - \sum_{v_i \in L_B} v_i$.

Rounding specification and Realizability: Consider a partition $R_1 \cup \cdots \cup R_k$ of I into k classes, and a function $R\colon I \to [0,1]^d$ which maps all items $v \in R_i$ to some item $\tilde{v}_i$. We call the instance $\tilde{I} := \{R(v) \mid v \in I\}$ a *rounding* of I to k item types. Sometimes, in our algorithms we do not know which items will be rounded in what way. We have classes $W_1, \ldots, W_k \subseteq I$, not necessarily disjoint, and for each class W_i there will be an item $\tilde{v}_i$ and a number w_i specified, meaning that exactly w_i items from W_i are supposed to be rounded to item $\tilde{v}_i$. We call this a *rounding specification*. We say, that a rounding specification is realizable for instance I, if there is a rounding $\tilde{I}$ and a function $R\colon I \to \tilde{I}$ that satisfies the requirements of the rounding specification. This can be checked, for example, by solving a suitable flow problem. Being able to guess the right rounding specification and test its realizability will be crucial in Section 5.

Now we describe some standard tools and ideas that are used repeatedly in the following sections.

Linear grouping: This is a simple but powerful idea introduced by [10]. Let $I = \{v_1, \ldots, v_n\}$, where $v_1 \ge v_2 \ge \cdots \ge v_n$, be a (one-dimensional) bin packing instance with large items and k be a fixed parameter. Split I into groups, each containing k consecutive items (except possibly for the last group). Construct a rounding $\tilde{I}$ of I by rounding all items in a group W_i to the size of largest item $(v_{(i-1)k+1})$ in W_i. We denote $\tilde{W}_i$ the rounded items of W_i. This rounding satisfies $\mathsf{Opt}(\tilde{I}) \le \mathsf{Opt}(I) + k$, as in the optimal packing of I, the items of W_i can be replaced by items from $\tilde{W}_{i+1}$ (since they are smaller) and the k items of $\tilde{W}_1$ can be packed into k additional bins.

Multi-objective/multi-budget (MOMB) matching problem: Here we are given a graph $G := (V, E)$, k linear functions $f_1, f_2, \ldots, f_k : E \to \mathbb{R}_+$ (called demands), ℓ linear functions $g_1, g_2, \ldots, g_\ell : E \to \mathbb{R}_+$ (called budgets), and the goal is to find a matching $\mathcal{M}$ satisfying $f_i(\mathcal{M}) \ge D_i$ for all $i \in [k]$ and $g_i(\mathcal{M}) \le B_i$ for all $i \in \ell$. Chekuri et al. [9] showed the following elegant result:

THEOREM 2.1. (SEE [9]) *For any constant $\gamma > 0$ and any constant number of demands and budgets $k+\ell$, there is a polynomial time algorithm that finds a matching S, such that*

- *Each linear budget constraint is satisfied:* $g_i(S) \leq B_i$.
- *Each linear demand is nearly satisfied:* $f_i(S) \geq (1-\gamma)D_i$.

Or else, returns a certificate that the instance is not feasible with demands D_i and budget B_i.

This algorithm plays an important role in our results, and we describe its use in more detail later.

Configuration LP: This is a class of strong LP relaxations that is widely used for bin packing type problems. There is a variable for each feasible way of packing a bin (called a *bin configuration*). This allows the packing problem to be cast as a set covering problem, where each item in the instance I must be covered by some configuration. Let $\mathcal{C}$ denote the set of all valid configurations for the instance I. The configuration LP is defined as:

(2.1)
$$\min \Big\{ \sum_{C \in \mathcal{C}} x_C : \sum_{C \ni v} x_C \geq 1 \quad \forall v \in I, x_C \geq 0 \quad \forall C \in \mathcal{C} \Big\}$$

As the size of $\mathcal{C}$ can possibly be exponential in the size of I, one typically considers the dual of this LP:

(2.2)
$$\max \Big\{ \sum_{v \in I} w_v : \sum_{v \in C} w_v \leq 1 \quad \forall C \in \mathcal{C}, w_v \geq 0 \quad \forall v \in I \Big\}$$

The separation problem for the dual is the following knapsack problem: Given set of weights w_i, is there a feasible configuration with total weight of items more than 1? For d-VP, the dual separation problem is the d-dimensional vector knapsack problem which could be solved to within $(1+\epsilon)$ accuracy [14]. This gives an algorithm solving (2.1), by the well-known connection between separation and optimization [16]. Note that the factor $1+\epsilon$ appears in the objective function of (2.1) only and not in the sizes of the items. In particular, the configurations produced use the unrounded item sizes.

R&A Framework and Rounding Based Algorithms: Here, we give a high-level idea of the R&A Framework in [2]. We use it in Section 6 and give a more detailed description there. Let I be an instance and A be a suitable ρ-approximation algorithm for d-VP. We proceed roughly as follows:

1. Solve configuration LP (2.1), let x^* be the (near)-optimal solution and $z^* := \sum_{C \in \mathcal{C}} x^*_C$.
2. Repeat $\lceil (\ln \rho) z^* \rceil$ times: select a random configuration C from $\mathcal{C}$, with probability x^*_C / z^*.
3. Pack the remaining set of unpacked items S in additional bins using A.

Note, that an item $i \in I$ lies in S with probability about $1/\rho$. Therefore, if we could claim that $\mathsf{Opt}(S) \leq (\frac{1}{\rho}) \cdot \mathsf{Opt}(I)$, we get

$$\lceil (\ln \rho) z^* \rceil + \mathsf{A}(S) \leq \lceil (\ln \rho) z^* \rceil + \rho \cdot \mathsf{Opt}(S),$$

what is at most $(1 + \ln \rho)\mathsf{Opt}(I)$, since $z^* \leq \mathsf{Opt}(I)$. However, it is unclear how to relate $\mathsf{Opt}(S)$ and $\mathsf{Opt}(I)$. The idea in [2] was to show, that if A is *subset-oblivious*, then for a random subset S, value of $\mathsf{A}(S)$ is essentially the same as $(1/\rho)\mathsf{A}(I)$, and therefore $\mathsf{A}(S) \leq \mathsf{Opt}(I)$. Usually, simple rounding-based algorithms can be shown to be subset-oblivious. Unfortunately, such algorithms are not very useful for d-VP as we show below.

Limitations of rounding-based algorithms: Here we show a lower bound for all rounding-based algorithms. As rounding-based, we consider all algorithms which round the coordinates of items up.

THEOREM 2.2. *Any algorithm that rounds up large coordinates of items to $O_d(1)$ types can not achieve better than d approximation for d-VP.*

Proof. Let A be an algorithm that rounds large coordinates of items to r (a constant) number of types. We describe an instance for which the approximation ratio of A is at least $d(1-\frac{1}{t})$, for arbitrary $t \in \mathbb{N}$: Let $m := t \cdot r$, and $\gamma > 0$ be a suitably small constant. We denote $v_{i,\ell}$ a vector having the ℓ-th coordinate $v^\ell_{i,\ell}$ equal to $(1-\epsilon_i)$ and all other coordinates equal to $\epsilon_i/(d-1)$, where $\epsilon_i := \frac{\gamma}{d^i}$. Instance $I := \{v_{i,\ell} \mid i \in [m], \ell \in [d]\}$ has an optimal packing into m bins $B_i := \{v_{i,\ell} | \ell \in [d]\}$ for $i \in [m]$. Note, that this is the unique optimal packing, since $v_{i,\ell}$ and $v_{i',\ell'}$ cannot be packed together for $i \neq i'$.

Now, let the rounding algorithm A round the large coordinates of items to r types $(1-\delta_1) < \cdots < (1-\delta_r)$. We denote $J := \{i_h | (1-\epsilon_{i_h}) \leq (1-\delta_h) < (1-\epsilon_{i_h+1})\}$. Only items $v_{i,\ell}$ for $i \in J$ could possibly retain their own sizes after rounding, all other items will have their large coordinates rounded up to $(1-\epsilon'_i)$, where $\epsilon'_i \leq \epsilon_i/d$.

Rounding just increases sizes of items, so the rounded items from $v_{i,\ell}$ and $v_{i',\ell'}$ still cannot be packed together for $i \neq i'$. Moreover, we claim that, for $i \notin J$, even $v_{i,\ell}$ and $v_{i,\ell'}$ cannot be packed together after the rounding. Let us consider the sum of ℓ-th coordinates of $v_{i,\ell}$ and $v_{i,\ell'}$: we know that $v^\ell_{i,\ell} + v^\ell_{i,\ell'} = (1-\epsilon_i) + \frac{\epsilon_i}{d-1}$. However, since $i \notin J$, it becomes at least $(1 - \epsilon_i/d) + \frac{\epsilon_i}{d-1} > 1$ after the rounding, and thereby they cannot be packed together in one bin.

Therefore, the algorithm A cannot find a packing of I into less than $d \cdot \big|[m] \setminus J\big| = d(m-r)$ bins and the approximation ratio is at least $\frac{d(m-r)}{m} = d(1-\frac{1}{t})$, since

$m = tr$. We can construct such an instance of arbitrary size by taking multiple copies of I. □

3 Overview and Roadmap

Before proving our results, we first give some intuition behind the main ideas and techniques. The starting point is the following simple observation. Suppose there is an optimal packing $\mathcal{P}$ of I where each bin in $B \in \mathcal{P}$ has some fixed slack δ in each dimension. Then one can get an optimal packing easily using the following resource augmentation result:

THEOREM 3.1. (CHEKURI, KHANNA [8]) *If a d-VP instance can be packed in m bins, then for any $\delta > 0$, a packing in m bins of size $(1+\delta, \ldots, 1+\delta)$ can be found in time poly$(n, f(\delta))$ for some function f.*

However this is too good to hope for, and there can be a large gap between packings with and without slack. For example in 2-dimensional case, if all items are $(0.5, 0.5)$, then $\mathsf{Opt}(I) = m/2$, while any packing with slack needs m bins. However, note that this instance, or any instance where each bin has at most 2 items, can be easily solved by matching. For d-dimensional case, we can consider the instance from the proof of Theorem 2.2, or even simpler one can be constructed: We take $I = \{v_{i,\ell} | i \in [m], \ell \in [d]\}$, where $v_{i,\ell}$ has coordinate ℓ equal to $1-\delta$ and the other coordinates equal to $\frac{\delta}{d-1}$. Then, clearly, $\mathsf{Opt} = m$, while $\mathsf{Opt}_{(1-\delta)} = md$.

On the other hand, we can claim the following:

LEMMA 3.1. (STRUCTURAL LEMMA FOR d-VP) *Fix any $\delta < 1/5$. For any packing $\mathcal{P}$ using m bins, there exists a packing $\mathcal{P}'$ into at most $\lceil \frac{d+1}{2} m \rceil$ bins, such that for each bin B in $\mathcal{P}'$: (i) either B contains at most 2 items, or (ii) at least $d-1$ of the dimensions in B has slack at least δ.*

We call such a packing $\mathcal{P}'$ a *structured* packing, this lemma is proved in Section 5.1.

Finding structured packings: Our goal is to find the structured packing $\mathcal{P}'$ efficiently. To handle bins of type (ii), we show a variant of Theorem 3.1 that requires resource augmentation only in $d-1$ dimensions (instead of d). This is Theorem 1.4 and is proved in Section 4. Now, if we knew which items were packed in the *matching* bins (of type (i) above, i.e., bins that contain at most two items), then an APTAS for $\mathcal{P}'$ would follow by applying Theorem 1.4 on the remaining items. However, it is unclear how to guess the items in the matching bins efficiently, as their sizes are not rounded.

To get around this, we flip the idea on its head. We observe that Theorem 1.4 for packing of bins with slack is based on rounding item sizes, and hence only requires knowledge of how many items of each size type (according to its internal rounding) are present in the instance. In fact, it works just with rounding specifications instead of the real instance. So we guess the right rounding specification, i.e. how the item classes look like and how many items belong to each of them and, in addition, numbers of items from each class which are to be packed in the matching bins. This leads precisely to the multi-objective budgeted matching problem [9]. In particular, we consider a graph with a vertex for each item and an edge between two items if they can be packed together. For each of the $O(1)$ types of items we specify how many items from this class are to be matched, and then apply Theorem 2.1 to find a matching with the guessed quota of items from each class. This way we can separate items which are to be packed into matching bins and use Theorem 1.4 to pack the rest. This procedure gives 3/2-asymptotic approximation for $d = 2$ and $(d+1)/2$-approximation for general d, and is described in Section 5.

Applying the *R&A* framework: We apply the R&A framework in different ways depending on whether $d = 2$ or $d > 2$. For $d = 2$, we first find a packing into matching bins, and then we apply R&A on remaining items. Roughly speaking, this works because the remaining items are packed using the APTAS which is rounding based. The proof has some additional technical difficulties compared to the previous result on 2-dimensional geometric bin packing [3], due to skewed items that are large in one dimension and small in another. This is described in Section 6.1.

To achieve our result for $d > 2$, we use a different kind of structured packing. Instead of matching bins we consider *compact* bins containing at most d items. There is no analogous results to Theorem 2.1 for multi-objective budgeted d-dimensional matching. However, we can procede in the following way. First we apply random sampling (part of R&A) to the whole instance. This step decreases the numbers of items belonging to compact bins to roughly one or two and that allows us to use Theorem 2.1 to pack them. Then we apply APTAS of Theorem 1.4 to the remaining items. The details are more complicated and we refer the reader to Section 6.2.

4 Vector Packing with Resource Augmentation

In this section we consider packing when resource augmentation is allowed in $(d-1)$-dimensions. We call these dimensions to be *augmentable*, and the only other dimension, which we are not allowed to augment, we call *non-augmentable*. Without loss of generality, we assume the last dimension to be the non-augmentable dimension. In this section we provide an APTAS algo-

rithm for this variant of vector packing and prove Theorem 1.4. Due to space limitation, we omit some of the proofs in this section. For the detailed proofs, we refer the readers to [24].

THEOREM 1.4 (RESTATED). *Let $\epsilon > 0$ be a constant and I be an instance of d-VP having an optimal packing into m unit-size bins. Then there is a polynomial time algorithm for packing vectors of I into at most $(1+2\epsilon)m+1$ bins with ϵ resource augmentation in $(d-1)$ dimensions (i.e., bins have size $(1+\epsilon)$ in the first $(d-1)$ dimensions and size 1 in the last one).*

Here is the overview of the algorithm. Given ϵ and a guess of the optimal value m, we describe a procedure that either returns a feasible packing into $(1+2\epsilon)m+1$ bins with ϵ resource augmentation in $(d-1)$ dimensions, or shows that the guess is incorrect. First, we choose $\beta = \epsilon/2d$ and classify items into *large* and *small*. Recall, that we call an item large, if it is $\geq \beta$ in at least one coordinate, otherwise it is small. Then we round the large items into a constant number of classes and pack them using dynamic programming. This part contributes a major slice of the overall time complexity. Afterwards we replace the rounded items by the original ones and the final step is to pack the small items using linear programming into the residual space of the bins and possibly into some additional bins if needed.

4.1 Rounding of large items. We apply different rounding to augmentable and non-augmentable coordinates. Coordinates $1, \ldots, d-1$ (those with the resource augmentation allowed) are rounded to the multiples of α, where $\alpha = \frac{\epsilon^2}{2d^2}$, and the d-th coordinate is rounded using linear grouping.

Rounding of augmentable coordinates: We create an instance $\widehat{Q}$ rounded in the first $d-1$ coordinates by replacing each large item p_i of I with an item $\widehat{q}_i$ as follows:

$$\widehat{q}_i^{\ell} := \begin{cases} \lceil p_i^{\ell}/\alpha \rceil \alpha & \text{if } \ell \in \{1, \ldots, (d-1)\}, \\ p_i^{\ell} & \text{if } \ell = d. \end{cases}$$

We classify the original instance I into classes $\{W^u \mid u \in \{1, \ldots, \lceil \frac{1}{\alpha} \rceil\}^{d-1}\}$ where $W^u := \{p_i \mid \widehat{q}_i^{\ell} = u^{\ell} \cdot \alpha \ \forall \ell \in [d-1]\}$, creating $r_A := (\lceil \frac{1}{\alpha} \rceil)^{d-1}$ classes altogether.

Rounding of the non-augmentable coordinate: Let $\lambda := \frac{\epsilon\beta}{2d}$ be a constant. We apply rounding of the last coordinate using linear grouping on each W^u separately, splitting it into $a := \lceil \frac{1}{\lambda} \rceil$ groups in the following way: Let $p_1, \ldots, p_{k_u}$, where $k_u := |W^u|$, be the items of W^u sorted in nonincreasing order according to their last coordinate. For each $j = 1, \ldots, a-1$ we define class $W^{u,j}$ having $b := \lceil \lambda k_u \rceil$ items as follows: $W^{u,j} := \{p_{(j-1)b+1}, \ldots, p_{jb}\}$. The last class $W^{u,a} := \{p_{(a-1)b+1}, \ldots, p_{k_u}\}$ might contain less than b items. The first resp. the largest item in each group we call *round vector*. To get the final rounded instance Q we replace each vector $p_i \in W^{u,j}$ by q_i, where

$$q_i^{\ell} := \widehat{q}_i^{\ell} \quad \text{for } \ell \in [d-1],$$
$$q_i^{d} := \max\{p^d \mid p \in W^{u,j}\}.$$

Thus we round-up the dth dimension to the dth coordinate of the *round vector* of the group and the other coordinates are rounded to multiples of α.

After the rounding, the items of I are classified into $r_L := \lceil 1/\lambda \rceil \cdot r_A = \lceil \frac{1}{\lambda} \rceil \cdot (\lceil \frac{1}{\alpha} \rceil)^{(d-1)}$ item classes. Thus any *configuration* of rounded items in a bin can be described as tuple $(k_1, k_2, \ldots, k_{r_L})$ where k_i indicates the number of vectors of the i'th class. As there can be at most d/β large items in a bin, there are at most $(d/\beta)^{r_L}$ possible bin *configurations*. We show that this rounding procedure fulfills the following lemma:

LEMMA 4.1. *Let I be an instance of d-dimensional vector packing and $\mathcal{P} = \{I_1, \ldots, I_m\}$ be a packing of I into m unit bins. Let Q be the rounding of large items of I using the described procedure. Then there is a packing of $Q' \subseteq Q$ into m bins of type $(1+\epsilon, 1+\epsilon, \ldots, 1+\epsilon, 1)$ such that:*

1. *Number of leftover items is small: $|Q \setminus Q'| \leq \frac{\epsilon m}{2}$*
2. *For $i \in [m]$, the configuration of small items in Q_i is the same or larger than the configuration of small items in I_i in all dimensions.*

Proof. Let $W^{u,j}$ be the classes the rounding as described above, and denote $p_{u,j} \in W^{u,j}$ the round vector of the class $W^{u,j}$. We construct a packing of rounded instance as follows: First, we remove all small items from the packing. Then, for each u, we remove items $W^{u,1} \setminus \{p_{u,1}\}$ from the packing $\mathcal{P}$ and put roundings of items in $W^{u,2} \setminus \{p_{u,2}\}$ in their places. In general, we place roundings of items $W^{u,j+1} \setminus \{p_{u,j+1}\}$ in place of items $W^{u,j} \setminus \{p_{u,j}\}$ for each u, leaving items $\bigcup_u (W^{u,1} \setminus \{p_{u,1}\})$ unpacked. The round vectors $p_{u,j}$ we replace by their roundings $q_{u,j}$ in their original places. Let us denote $Q_1, \ldots, Q_m$ the content of the packing bins after this replacement procedure.

Now, we show how many items remain unpacked. The size of $\bigcup_u (W^{u,1} \setminus \{p_{u,1}\})$ is

$$\sum_u (|W^{u,1}| - 1) \leq \sum_u (\lceil \lambda |W^u| \rceil - 1) \leq \sum_u \lambda |W^u|,$$

i.e. it is at most $\lambda|Q|$, since $|Q| = \sum_u |W^u|$. As each large item is bigger than β in at least one coordinate,

there cannot be more than $\frac{d}{\beta}$ large items in any bin I_i. Thereby, for $\lambda = \frac{\epsilon}{2d}\beta$, at most $\lambda\frac{d}{\beta}m \leq \frac{\epsilon m}{2}$ large items were left unpacked.

Now we prove part (2), which also implies packability of Q_i into augmented bins. The last dimension is the easiest. Let L_i be the set of large items in the bin I_i and recall that the small configuration is $\mathbb{1} - \sum_{p\in L_i} p$. We have replaced each large item of bin I_i by an item which is same or smaller in the last coordinate. Therefore, for the small configuration in dimension d, we have $1 - \sum_{q\in Q_i} q^d \geq 1 - \sum_{p\in L_i} p^d$. Let us now consider dimension $\ell \in [d-1]$. As we know, there are at most $\frac{d}{\beta}$ in I_i and each of them increased in rounding by at most α. Therefore the total increase in coordinate ℓ in Q_i can be at most $\frac{d}{\beta}\alpha = \epsilon$, as $\alpha = \frac{\epsilon}{d}\beta$. And since this dimension is augmented, the small configuration in dimension ℓ is at least $(1+\epsilon) - \sum_{q\in Q_i} q^\ell \geq 1 - \sum_{p\in L_i} p^\ell$. □

4.2 Packing of large items. There is a constant number of item types r_L in Q, thus there are only $r \leq (\frac{d}{\beta})^{r_L}$, number of possible configurations of a single bin. We call $M := (m_1, \ldots, m_r)$ a bin configuration, where m_i specifies number of bins in the packing of type i. Since $m \leq n$, this gives us $O(m^r)$, a polynomial number of configurations of large items for m bins. If we know the right bin configuration, we pack Q using the following lemma:

LEMMA 4.2. ([8, LEMMA 2.3]) *Let $M = (m_1, \ldots m_r)$ be a bin configuration. There is an algorithm with running time $O\big((\frac{dn}{\beta})^{r_L}\cdot m\big)$ to decide if there is a packing of Q that respects M.*

LEMMA 4.3. *Let $I_1, \ldots, I_m$ be some optimal packing of I. In polynomial time, we can find a packing of Q into bins $Q_1, \ldots, Q_m$ plus at most $\lfloor \frac{\epsilon m}{2} \rfloor$ additional bins, such that small configuration of each bin Q_i is same or larger than the bin I_i in every dimension for $i \in [m]$.*

Proof. Lemma 4.1 says that there is a packing of Q' respecting such a bin configuration which has small configuration of the first m bins at least the same as $\mathcal{P}$. Since there are $O(m^r)$ possible bin configurations, we can try all of them and find the packing of Q' using Lemma 4.2. Since $|Q \setminus Q'| \leq \lfloor \frac{\epsilon m}{2} \rfloor$, we can pack them separately into at most $\lfloor \frac{\epsilon m}{2} \rfloor$ additional bins. □

4.3 Packing of small items. To pack the small vectors, we use an assignment LP as in [8]. However in [8], as resource augmentation was allowed in all dimensions, small vectors could have been packed without using any additional bins. In our case we need extra bins.

LEMMA 4.4. *Let $I_1, \ldots, I_m$ be some packing of I and $Q_1, \ldots, Q_m$ be a packing of Q' into m bins of type $(1+\epsilon, 1+\epsilon, \ldots, 1+\epsilon, 1)$ such that the small configuration of each bin Q_i is the same or larger than the bin I_i in every dimension. Then we can pack all the small items of I into residual space of bins $Q_1, \ldots, Q_m$ and at most $\left\lceil \frac{m}{2} \cdot (\epsilon + \frac{\epsilon^2}{d}) \right\rceil$ additional bins.*

Proof. Let us denote S to be the set of all small items in I and for each bin B_j and each coordinate $\ell \in [d]$ we denote $b_j^\ell := \sum_{q\in Q_i} q^\ell$. We formulate the following assignment LP with variables x_{ij} saying whether item $p_i \in S$ should be packed into bin B_j.

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \forall p_i \in S$$

$$\sum_{i=1}^{|S|} x_{ij} p_i^\ell \leq (1+\epsilon) - b_j^\ell \quad \forall j \in [m], \ell \in [d-1]$$

$$\sum_{i=1}^{|S|} x_{ij} p_i^\ell \leq 1 - b_j^\ell \quad \forall j \in [m], \ell \in [d]$$

$$x_{ij} \geq 0 \quad \forall i, j$$

This LP is feasible, since there was a feasible packing into a same or even smaller space of those bins. Let us consider a basic solution x to this LP. The integrally assigned items we can pack directly, the others we pack into additional bins. Let us denote F the set of items assigned fractionally. We claim that $|F| \leq dm$, then we need at most $\lceil \frac{dm}{\lfloor \frac{1}{\beta} \rfloor} \rceil \leq \lceil \frac{dm}{\lfloor \frac{2d}{\epsilon} \rfloor} \rceil \leq \lceil \frac{m}{2}(\epsilon + \frac{\epsilon^2}{d}) \rceil$ bins to pack them, since each item is smaller than β in all coordinates. This follows from polyhedral theory: x is a basic solution and therefore satisfies at least $m|S|$ constraints with equality. There are $dm+|S|$ non-trivial constraints, thus at most $dm + |S|$ variables can be strictly positive. As each item is assigned to at least one bin, number of items assigned fractionally to more bins, i.e. the size of F, can be at most dm. □

4.4 Algorithm. In Algorithm 1, a summary of the algorithm is given.

Proof. [of Theorem 1.4] If there is a packing of I into m unit bins, from Lemma 4.1 we know that there is a packing of rounded instance Q of large items of I into $m' := m + \lfloor \frac{\epsilon m}{2} \rfloor$ unit bins having still enough space for the small items S. Moreover, Lemmas 4.3 and 4.4 assure us that we can indeed find the packing of $Q \cup S$ into $m'' := m + \lfloor \frac{\epsilon m}{2} \rfloor + \lceil \frac{m}{2} \cdot (\epsilon + \epsilon^2/d) \rceil \leq m + m\cdot(\epsilon + \frac{\epsilon^2}{2d}) + 1$ augmented bins in polynomial time. Therefore, if this procedure fails, we know that our guess of m was too small. We can find the right guess using binary search between 1 and n in $\log n$ attempts.

Guess $m := \mathsf{Opt}(I)$
A. Rounding:
Create rounding Q of large items of I
B. Packing of large items.
B1. Guess bin configuration of $m + \lceil \frac{\epsilon m}{2} \rceil$ bins of size $(1+\epsilon, \ldots, 1+\epsilon, 1)$
B2. Pack items in Q into configuration M, or return to Guessing phase
C. Packing of small items.
Pack small items using assignment LP, or return to Guessing phase.
Replace items in Q by original ones

Algorithm 1: d-VP WITH RESOURCE AUGMENTATION IN $(d-1)$ DIMENSIONS

Since we were rounding up, we can replace all rounded items q_i in the packing of $Q \cup S$ by corresponding unrounded p_i, getting a feasible packing of I into the same number of augmented bins. Note that until now we did not use any exact information about the large items of I during the packing. Actually, in this moment we can pack even similar large items from a completely different instance instead, we just need it to have the same rounding specification as I. We use this observation in the following section. □

5 Finding a well-structured approximation solution for vector packing

Now we describe our preliminary approximation algorithm, which is incorporated into R&A framework in Section 6. Here is the main theorem of this section:

THEOREM 5.1. *Given any constant ϵ, such that $0 < \epsilon < \frac{1}{56d^2}$, and a d-VP instance I, there is a polynomial time algorithm to pack I into at most $\lceil (\frac{d+1}{2})\mathsf{Opt}(I) \rceil \cdot (1+2\epsilon) + 1$ unit bins.*

This already gives $(1.5+\epsilon)$ and $(2+\epsilon)$ asymptotic approximations for 2-VP and 3-VP improving upon the current best guarantees of $(1.693+\epsilon)$ and $(2.099+\epsilon)$ respectively.

The algorithm proceeds as follows: Instead of optimal packing which is hard to find, it is trying to find the smallest *structured packing*, which we can find much easier. Now let us define *structured packing*, where δ is a suitably small constant whose value will be specified later.

DEFINITION 5.1. *A packing $\mathcal{P}$ is* structured *if each bin $B \in \mathcal{P}$ satisfies one of the following:*

- *B consists of precisely one single large item v. We denote $\mathcal{B}_S$ the collection of such bins.*
- *B consists of precisely two large items v_i, v_j. We denote $\mathcal{B}_T$ the collection of such bins.*
- *B has δ-slack in at least $d-1$ dimensions. The collection of bins with δ-slack in dimensions $[d]\setminus\{\ell\}$ is denoted by $\mathcal{B}_\ell$. If B has δ-slack in all dimensions we assign it (arbitrarily) to $\mathcal{B}_1$.*

Lemma 3.1 assures us about the existence of small structured packings and its proof is contained in Subsection 5.1. Together with the following lemma it already implies Theorem 5.1.

LEMMA 5.1. *Let $\epsilon > 0$ be a constant and I be an instance of d-dimensional vector packing having a structured packing into m' bins, then there is a polynomial time algorithm that pack items in I into at most $(1+2\epsilon)m' + 1$ bins.*

Note, that the imprecision term is the same as in Theorem 1.4 and, indeed, is coming from the APTAS algorithm of the previous section which is used to pack bins of type $\mathcal{B}_\ell$, for $\ell \in [d]$. Therefore, we set δ in the definition of structured packing to $\frac{\epsilon}{1+\epsilon}$. This way, if we have a bin $B \in \mathcal{B}_\ell$ and we pack it using APTAS with resource augmentation by ϵ from Section 4, we get a feasible packing, since it is easy to check that $(1-\delta)\cdot(1+\epsilon) = 1$, for $\delta = \frac{\epsilon}{1+\epsilon}$. Proof of Lemma 5.1 including the main algorithm of this section is in Subsection 5.2. Subsection 5.3 contains the algorithm proving our result on absolute approximation ratio for 2-VP, Theorem 1.2.

5.1 Existence of small structured packing

The goal of this subsection is to prove the following theorem:

LEMMA 3.1 (RESTATED). *Let I be an instance of d-dimensional vector packing. Any optimal packing of I into m bins can be transformed to a packing into $\lceil \frac{d+1}{2} m \rceil$ bins of types $\mathcal{B}_S, \mathcal{B}_T, \mathcal{B}_1, \ldots, \mathcal{B}_d$.*

Case $d = 1$ is trivial. Thus we start by considering $d = 2$, which is not very difficult. For $d > 2$, the proof is more complicated and we split the proof into cases when d is odd and when d is even.

LEMMA 5.2. *Let $\delta < 1/5$, and let B be a bin of 2-VP that is not structured. Then at least one of the following holds:*

1. *There is a large item $p \in B$ such that $p \le (1/2, 1/2)$ and p is $> \delta$ in at least one coordinate.*
2. *There is a subset R of items in B such that $\sum_{p\in R} p \le (2\delta, 2\delta)$ and either:*
   (i) $B \setminus R$ has δ-slack in some dimension, or
   (ii) $|B \setminus R| \le 2$.

Proof. Let T denote the set of items $v \in B$ such that $v \leq (\delta, \delta)$. As there can be at most two items in a bin with some coordinate strictly $> 1/2$ (at most one item for each coordinate), if there is no p satisfying the first requirement, we know that $|B \setminus T| \leq 2$.

If $(\sum_{v \in T} v) \leq (\delta, \delta)$, we set $R := T$ and $|B \setminus R| \leq 2$, as desired. Otherwise, $(\sum_{v \in T} v)$ is greater than δ in at least one dimension. So we greedily add items of T to R until $B \setminus R$ contains δ slack in at least one dimension. Since T only contains items $< (\delta, \delta)$, the items of R can not sum to more than $\delta + \delta \leq 2\delta$ in any coordinate. □

The proof of Lemma 3.1 for $d = 2$ follows easily from the following statement:

LEMMA 5.3. *For any two unstructured bins B_1 and B_2 in I, their items can be repacked into three structured bins.*

Proof. As $\delta < 1/5$, we have $4\delta < (1 - \delta)$ and $1/2 + \delta < (1 - \delta)$. From each B_1 and B_2, let us remove either the large item p or a set R (as in Lemma 5.2), and pack them into a new bin B'. We first note that if either p_i or R_i is removed from some bin B_i, then B_i becomes structured (it either has δ-slack in some dimension, or at most 2 items). Second, if the repacked items are p_1 and p_2, then B' is structured as it has exactly 2 large items. Otherwise, if at least one item is of type R_i, then B' has δ-slack in both dimensions as both $1/2 + 2\delta \leq 1 - \delta$ and $2\delta + 2\delta \leq 1 - \delta$. □

The idea of the proof for $d > 2$ is the following: we continue removing large items from B each creating slack in one or more dimensions, until slack in at least $(d - 1)$ dimensions is created. However, it can happen that there are no more large items left in B and there is still not enough slack in some dimensions. This case might lead to the following problem. If we select the set R of items to be removed from B greedily, at some point it can happen that the sum of the items in R is already over $(1 - \delta)$ in some of the dimensions, while still much less than δ in the others. However, using a more sophisticated LP-based argument we can find a set R which will create slack in required dimensions and still be relatively small in all dimensions. Let κ be a constant, dependent only on d, whose value we will choose later. Now let us prove the following lemma:

LEMMA 5.4. *Let B be a bin and C be the set of dimensions where the sum of items in B is more than $(1 - \delta)$. If all items in B are smaller than δ in all coordinates of C, then we can find $R \subseteq B$ such that the sum of items in R is at least δ in coordinates of C, but at most $\kappa\delta$ in all coordinates, for some constant $\kappa > 0$.*

$$
\begin{aligned}
&\min \sum_{p_i \in B'} x_i \\
&\sum_{p_i \in B'} x_i p_i^\ell \geq \delta \quad \forall\, \ell \in C, \\
&\sum_{p_i \in B'} x_i p_i^\ell \leq 2\delta \quad \forall\, \ell \in [d], \\
&0 \leq x_i^\ell \leq 1 \quad \forall\, i \in B', \ell \in [d].
\end{aligned} \tag{5.3}
$$

Proof. We restrict our attention to the items of $B' \subseteq B$ which are smaller than $3\delta d$ in all dimensions. In fact, this way we do not loose much volume in coordinates of C, since there can be at most $d \cdot \frac{1}{3\delta d}$ items in $B \setminus B'$ and all of them are at most δ in all coordinates in C. Therefore, $\sum_{p \in B'} p^\ell \geq (1 - \delta) - \delta \frac{d}{3\delta d} \geq 1 - \delta - 1/3$, for $\ell \in C$. We formulate LP (5.3) for items in B'. To show the feasibility of this LP, let us consider the solution $x' = (2\delta, \ldots, 2\delta)$: For coordinate $\ell \in C$ we have $\sum_{p_i \in B'} p_i^\ell x_i \geq (1 - \delta - 1/3)2\delta = 2\delta - 2\delta^2 - 2\delta/3 \geq \delta$, for $\delta < 1/6$. On the other hand, for any coordinate $\ell \in [d]$, $\sum_{p_i \in B'} p_i^\ell x_i \leq 2\delta$, since $\sum_{p_i \in B'} p_i^\ell \leq 1$.

Let us fix a basic optimal solution x to (5.3) and set $R := \{p_i | \, x_i > 0\}$. We claim that the sum $\sum_{p_i \in S} p_i$ is at most $\kappa\delta$ in each dimension for some constant κ dependent just on d. Let F denote the set of fractional variables x_i. We know that each basic solution to the LP (5.3) fulfils at least $|B'|$ constraints with equality. Apart from constraints $0 \leq x_i^\ell \leq 1$, which would make the variable integral if fulfilled with equality, there are at most $(|C| + d) \leq 2d$ nontrivial constraints and they allow us $|F| \leq 2d$. Since each item is at most $3\delta d$ in all coordinates, we get $\sum_{p_i \in S} p_i \leq 2\delta + 2d \cdot 3\delta d \leq 7\delta d^2$ as $d \geq 2$. Hence, the claim follows by taking $\kappa := 7d^2$. □

Now we are ready to proof Lemma 3.1 for d odd. This is the key lemma used in the proof.

LEMMA 5.5. *If d is odd then at least one of the following holds: either*

(i) there is $V \subseteq B$, such that $|V| \leq d - 1$ and $B \setminus V$ has slack in at least $d - 1$ dimensions, or

(ii) there are $V, R \subseteq B$, such that $|V| \leq d - 2$, $\sum_{p \in R} p \leq \kappa\delta$, and $B \setminus (V \cup R)$ has slack in all dimensions.

Proof. Let C be the set of coordinates in which B contains slack less than δ. In each step we select $v \in B$ with the largest coordinate when restricted to dimensions currently in C. If v is at least δ in that coordinate, we add it to V. Otherwise if v is $< \delta$ in that coordinate, or if $|C|$ is already ≤ 1, we stop.

If $|C| \leq 1$, we are in case (i) and we have finished, since each item in V removed at least one dimension

from C and thereby $|V| \leq d-1$. If not, we have $|V| \leq d-2$ and we know that all items in $B \setminus V$ have all coordinates in C smaller than δ. Thanks to Lemma 5.4 we can find $R \subseteq B \setminus V$ satisfying (ii). □

Proof. [of Lemma 3.1 for d odd] Let $B_1, \ldots, B_m$ are bins in optimal packing of I. For each $i \in [m]$, we claim that we can remove some items from B_i and pack them separately into at most $(d-1)/2$ additional structured bins. To this end, let us apply Lemma 5.5 to bin B_i. In case (i) we can just pack items in V in pairs into at most $\frac{d-1}{2}$ bins creating bins of type $\mathcal{B}_T$ and eventually $\mathcal{B}_S$. In case (ii) we consider two subcases. If $|V| \leq d-3$, we pack R into a separate bin (it has slack in all dimensions) and pack V in pairs into at most $\frac{d-3}{2}$ bins of type $\mathcal{B}_T$ or $\mathcal{B}_S$, thereby using at most $\frac{d-3}{2}+1 = \frac{d-1}{2}$ additional structured bins. If $|V| = d-2$, we claim that there is an item $v \in V$ which is greater than $1/2$ in at most one coordinate. This is clearly true, since items in V create slack in $d-2$ dimensions and there cannot be two items bigger than $1/2$ in the same coordinate. We pack v together with R into the same bin (contains slack in at least d-1 dimensions) and we pack the rest of V in pairs into $\frac{d-3}{2}$ additional bins. In both cases (i) and (ii) we are left with $B_i \setminus V$ and $B_i \setminus (V \cup R)$ respectively, with slack in at least $d-1$ dimensions what completes the proof. □

In the case when d is even, we have to proceed more carefully. From each bin B_i we remove some items which can be packable into $\frac{d-2}{2}$ structured bins plus we are allowed a residue set R_i (similarly to case $d=2$ in Lemma 5.2) roughly smaller than $1/2$ in all coordinates, that can be packed together with any residue R_j of some other bin into a structured bin.

LEMMA 5.6. *If d is even then at least one of the following holds:*

(i) There are $V, R \subseteq B$, such that $|V| \leq d-2$, for R (possibly empty) we have $\sum_{p \in R} p \leq \kappa\delta$, and $B \setminus (V \cup R)$ has slack in at least $(d-1)$ dimensions.

(ii) There is $V \subseteq B$, such that $|V| = d-2$ and $B \setminus V$ has slack in exactly $d-2$ dimensions.

Proof. We proceed in a similar way to Lemma 5.5. We maintain C as a set of dimensions in which there is still no δ slack yet and in each step we add such item to V which is largest in coordinates of C. If we manage to create slack in at least $d-2$ coordinates this way and using at most $d-2$ items, requirement (ii) is satisfied. Otherwise we use Lemma 5.4 to find R which creates slack in all the other dimensions. □

Proof. [of Lemma 3.1, for the case when $d(>2)$ is even] Let $B_1, \ldots, B_m$ are bins in optimal packing of I. For each $i \in [m]$, we split content of B_i into $\frac{d}{2}$ structured-packing bins plus one residue set R_i containing either a single item p of size $\leq 1/2$ in all coordinates or several small items summing up to at most $(1/2-\delta)$ in at least $(d-1)$ dimensions and at most $1/2$ in the other one. Afterwards, we can pack residues R_i and R_j from two different bins together creating either a bin with slack in at least $(d-1)$ dimensions or a bin of type $\mathcal{B}_T$. This way we create a structured packing of I into at most $\frac{d}{2}m + \lceil \frac{m}{2} \rceil = \lceil \frac{d+1}{2} m \rceil$ bins.

Now, let us show that for each $i \in [m]$, we can decompose B_i in the desired way. Let us apply Lemma 5.6 to B_i. In case (i) we are done: we can pack V in pairs into at most $\frac{d-2}{2}$ bins of type $\mathcal{B}_T$ or $\mathcal{B}_S$ and choose $R_i := R$. Remaining $B \setminus (V \cup R)$ has slack in at least $(d-1)$ dimensions, and thus can be converted to a structured packing.

Now we consider the case (ii). Note, that here we have a nice correspondence between the dimensions of $[d] \setminus C$ and the items in V. We know, that each item in V is responsible for creating slack in one particular dimension, so we can write $V = \{v_\ell |\ \ell \in [d] \setminus C\}$, where $B \setminus \{v_\ell\}$ has slack in dimension ℓ. We will use this property later. We apply Lemma 5.2 to the two dimensions in C. Here are the possible outputs of this Lemma:

(1) There is $p \in B_i \setminus V$ which is at least δ in some coordinate of C and is at most $1/2$ in both.

(2) There is $R \subseteq B_i \setminus V$ which sums to at most 2δ in coordinates of C, and either

  (i) $B_i \setminus (V \cup R)$ has slack in at least $(d-1)$ dimensions, or

  (ii) for $Q := B_i \setminus (V \cup R)$ we have $|Q| \leq 2$ and each item in Q is larger than $1/2$ in some coordinate of C.

In case (1) p is in fact $\leq 1/2$ in all coordinates, because from dimensions $[d] \setminus C$ the larges items are already contained in V and therefore p cannot be larger than $1/2$ there. We set $R_i := \{p\}$ and pack V in pairs into at most $\frac{d-2}{2}$ additional bins and we are done.

Now we consider case (2). Let us denote $C_R \subseteq [d] \setminus C$ to be the set of dimensions where R sums to more than $1/2 - \delta$.

In case $(2i)$, if $|C_R| \leq 1$, then we can pack V in pairs and set $R_i := R$, since the sum of items in R is at most 2δ in coordinates of C. Otherwise if $|C_R| > 1$, let $V' := \{v_i \in V |\ i \notin C_R\}$. We know that $|V'| \leq d-4$, since $|C_R| \geq 2$. Then $B \setminus (V' \cup R)$ has slack in at least $(d-1)$ dimensions. We pack $B \setminus (V' \cup R)$ into a structured bin. Similarly we can pack R into another structured bin as it has slack in all dimensions. For items in V' we can pack them in pairs into at most $\frac{d-4}{2}$ bins of type $\mathcal{B}_S$ and $\mathcal{B}_T$, setting $R_i := \emptyset$.

In case $(2ii)$, if $|C_R| \leq 1$, we can set $R_i := R$ and pack V in pairs. Otherwise if $|C_R| > 1$, let $V_R := \{v_i \in V |\ i \in C_R\}$ (note that $|V_R| \geq 2$), and $V' := (V \setminus V_R) \cup Q$, $|V'| \leq d-2$. We select $v', v'' \in V'$ arbitrarily. Since R and V_R contain slack in all dimensions, $R \cup \{v'\}$ and $V_R \cup \{v''\}$ contain slack in at least $(d-1)$ dimensions and we can pack tham into slack bins, and $V'' := V' \setminus \{v', v''\}$ into at most $\frac{d-4}{2}$ bins of type $\mathcal{B}_T$ and $\mathcal{B}_S$, since $|V''| = |V'| - 2 = d-4$. □

5.2 Finding the best structured packing. Let us assume, that we are given a partition of I into subinstances I_S, I_T and $I_1, \ldots, I_d$ consisting of items which are packed into m_S bins of type $\mathcal{B}_S$, m_T bins of type $\mathcal{B}_T$, and m_ℓ bins of type $\mathcal{B}_\ell$, for $\ell \in [d]$ respectively in some optimal structured packing. Then we can easily pack items of I_S, each item separately, into exactly m_S bins, I_T we can pack using matching into exactly m_T bins, and I_ℓ we can pack into $(1 + 2\epsilon)m_\ell + 1$ bins using the Algorithm 1 from the previous section. Together with Lemma 3.1 which proves the existence of small structured packings this would give us the approximation bound promised by Theorem 5.1. Note, that we just need to have a realizable rounding specification for the Algorithm 1 to work (see proof of Theorem 1.4). We cannot start with separating I_S and I_T directly, we need to make sure, that the leftover items will have the same rounding specification as in the optimal structured packing. Therefore we proceed as follows: We guess the rounding classes for the d instances of Algorithm 1 and classify all large items to the rounding classes. Afterwards, we guess how many items from which rounding class should be separated to I_S and I_T. Having right guesses, we are able to pack I_S and I_T, and the leftover items will have the right rounding specification.

To describe the algorithm, we use similar notations as in Section 4. When we use the Theorem 1.4, we choose the same ϵ as the one in the statement of Theorem 5.1, see the discussion in the beginning of Section 5. Please recall the following constants depending on ϵ from the rounding procedure in Section 4: r_A is the number of rounding classes W^u of augmentable coordinates, each of them contains $\lceil \frac{1}{\lambda} \rceil$ classes of linear grouping, making $r_L = r_A \cdot \lceil \frac{1}{\lambda} \rceil$ classes $W^{u,j}$ altogether.

5.2.1 Preprocessing stage: In the beginning, we separate the set S of small items of I. This can be done in linear time. We denote $I' := I \setminus S$.

5.2.2 Guessing stage: In this stage we guess all parameters which need to be guessed. After all parameters are assigned, we proceed to the next stage, the actual packing algorithm, to determine whether there exists some packing with respect to the current guess of the parameter. If no, we return here to try another guess. This stage could be also called the stage of enumeration, since in the most of the guessed parameters we might have to enumerate all possible values in the worst case.

Guessing bin numbers: First we guess m', the number of bins in the smallest structured packing of I. We can do it using binary search between 1 and n, since, we can claim that m' is greater than our current guess if the Packing stage fails (we prove this claim later). After guessing m', we guess its partition into bins of types $\mathcal{B}_S, \mathcal{B}_T$ and $\mathcal{B}_1, \ldots, \mathcal{B}_d$, i.e. numbers $m_S, m_T, m_1, \ldots, m_d$. They all are in a range from 1 to m', so we guess $d+2$ numbers, each of them for sure in range 1 to n. So, even if we have to try all the possibilities, it is always polynomial number of attempts.

Guessing rounding specification: For each $\ell \in [d]$, we define the rounding classes W^u_ℓ according to the first part of the rounding procedure in Section 4 (rounding of augmentable coordinates). For each ℓ, we have r_A (a constant) number of classes. This part is easy and deterministic. However, each class W^u_ℓ should be subdivided into $\lceil \frac{1}{\lambda} \rceil$ classes according to part Rounding of the non-augmentable coordinate in Section 4. In Section 4, this part was deterministic too, it consisted of ordering all elements of the class and selecting $\lceil \frac{1}{\lambda} \rceil$ round vectors. However, we do not know which items belong to instance I_ℓ and will be rounded according to W^u_ℓ. Therefore the round vectors need to be guessed among all items of size corresponding to the class W^u_ℓ. Although many of these choices might not be valid, it does not hurt us, since we claim that the number of all possibilities is polynomial: Clearly, none of the classes W^u_ℓ contains more than n items. So there are at most $d \cdot r_A \cdot n^{(\lceil \frac{1}{\lambda} \rceil)}$ possible choices of round vectors. This way we can prepare $d \cdot r_L$ rounding classes $W^{u,j}_\ell$.

For each $W^{u,j}_\ell$, let $w^{u,j}_\ell$ be the number of items which are supposed to belong to this rounding class in the packing using APTAS in Section 4. Now for each item in bins of type $\mathcal{B}_S$ and $\mathcal{B}_T$, we also assign them to a rounded size class whose size is larger than the size of the item. For each class $W^{u,j}_\ell$, let $s^{u,j}_\ell, t^{u,j}_\ell$ be the number of items assigned to class $W^{u,j}_\ell$ which should be packed into bins of type $\mathcal{B}_S$ and $\mathcal{B}_T$ respectively. Now for each class we guess the number $h^{u,j}_\ell = s^{u,j}_\ell + t^{u,j}_\ell + w^{u,j}_\ell$.

Verifying realizability: It can happen, that preceding guesse of rounding classes and numbers $h^{u,j}_\ell$ might not be realizable for the instance I', e.g., for some class $W^{u,j}_\ell$, there can be less than $h^{u,j}_\ell$ items in I of the corresponding size. To find out whether there exists an assignment of items in I' into the rounding classes re-

specting numbers $h_\ell^{u,j}$ we use a technique from [28]. We specify the following flow network $G := (V, E)$:

- we have vertices s and t for source and sink,
- a vertex for each item and for each class $W_\ell^{u,i}$,
- edge of capacity 1 from s to each item,
- for each item, we have an edge of capacity 1 from the item to all d possible rounding classes which the item could belong to,
- we add an edge of capacity $h_\ell^{u,j}$ from each class $W_\ell^{u,j}$ to t.

Using the algorithm of Dinic [11] we find a maximum integral flow from s to t in time $O(|E| \cdot |V|^2) \leq O(n + nd + dr_L) \cdot (n + dr_L)^2 = O_{d,\epsilon}(n^3)$. If it does not saturate some edge outgoing from s or incoming to t, this means that we either cannot assign some item to any class, or some class cannot have the required number of items assigned. In either case, we know that no valid assignment exists and the rounding specification is not realizable and we need to go back to the guessing stage and make another guess. Otherwise we can proceed to the following step.

Guessing $w_\ell^{u,j}, s_\ell^{u,j}$ and $t_\ell^{u,j}$: Now for each class $W_\ell^{u,j}$, we guess numbers $w_\ell^{u,j}$. Actually, for a fixed u and ℓ, they will be always the same except for the last one, so we need to guess just two values $w_\ell^{u,1}$ and $w_\ell^{u,\lceil \frac{1}{\lambda} \rceil}$. Note, that the choice of these numbers finishes the preparation of the rounding specification for the d instances for packing by APTAS with resource augmentation.

Also for each class $W_\ell^{u,j}$, we guess numbers $s_\ell^{u,j}, t_\ell^{u,j}$ determining the number of items assigned to class $W_\ell^{u,j}$ which should be packed into bins of type $\mathcal{B}_S$ and $\mathcal{B}_T$ respectively. For r_L classes we guess two numbers which are without any doubts in range between 1 and n, therefore there are at most n^{2r_L} possible guesses.

LEMMA 5.7. *For any instance I there is a right choice of parameters guessed in guessing stage. All parameters in the guessing phase can be guessed in polynomial time.*

Proof. We defined instance of d-VP as a set of vectors $v_i \in [0,1]^d$. Therefore any instance has some packing (e.g. packing each item separately) and, thanks to Lemma 3.1, a structured packing. For the structured packing we can generate rounding classes according to the rounding procedure in Section 4 and set numbers $w_\ell^{u,j}, s_\ell^{u,j}$, and $t_\ell^{u,j}$ according to the placement of items of the rounding class in the packing.

All steps in the guessing stage (as noted at the end of each part) can be done in polynomial time, and thereby the whole stage takes polynomial time. □

5.2.3 Packing stage: Now we describe how to pack the items according to the guesses from the guessing stage.

Packing of I_T: To find a packing of I_T into bins of type $\mathcal{B}_T$, we construct a graph (I', E), where $(p_i, p_j) \in E$ if p_i and p_j fit together into a bin. When creating G, we consider the original unrounded sizes of items, although they are already assigned to some rounding class. We want to find a matching such that from each class $W_\ell^{u,j}$, exactly $t_\ell^{u,j}$ items are matched. We formulate the following LP:

$$\sum x_{uv} \leq 1 \quad \text{for all vertices } v \in V,$$
$$\sum_{u \in W_\ell^{u,j}} x_{uv} = t_\ell^{u,j} \quad \text{for all classes } W_\ell^{u,j},$$
$$0 \leq x_e \leq 1 \quad \text{for all edge } e \in E.$$

Note that this is a matching LP with r_L (a constant) number of additional linear constraints. We can solve this LP in polynomial time, see Theorem 2.1 using an algorithm of Chekuri, Vondrák, and Zenklusen [9]. This algorithm either returns a matching which covers between $t_\ell^{u,j}$ and $(1-\gamma)t_\ell^{u,j}$ items from each class $W_\ell^{u,j}$ or it returns a certificate that this LP is infeasible (See Theorem 2.1). We choose parameter $\gamma := \epsilon$. If the LP is infeasible, then either the rounding classes or the numbers $t_\ell^{u,j}$ are incorrect and we should try another guess. Otherwise, we pack the matched items, each pair into a single bin, and we are left with at most $\epsilon t_\ell^{u,j}$ residual unmatched items in each class, those we select arbitrarily. Note, that $\sum t_\ell^{u,j} = 2m_T$ (otherwise guesses are invalid), and therefore there are at most $2\epsilon m_T$ residual items. We can pack each of them in a separate bin, using in this stage $(1-\epsilon)m_T + 2\epsilon m_T \leq (1+\epsilon)m_T$ bins in total.

Packing of I_S: From each class $W_\ell^{u,j}$ we select arbitrary $s_\ell^{u,j}$ items not belonging to I_T and pack them into separate m_S bins.

Packing of $I_1, \ldots, I_d$ by APTAS: Please note, that we still do not know the instances I_ℓ, for $\ell \in [d]$. We know just their rounding specifications (we have guessed them). However, using rounding specification we can reconstruct roundings of their large items. For each $\ell \in [d]$, we proceed as follows: Let Q_ℓ be rounding of large items from I_ℓ. We assume that I_ℓ has a packing into bins $I_\ell^1, \ldots, I_\ell^{m_\ell}$ of type $\mathcal{B}_\ell$. Recall, that these bins have slack δ in all dimensions except for ℓ. Therefore, (after swapping coordinates ℓ and d if needed) we can use Lemma 4.3 to pack Q_ℓ into unit bins $Q_\ell^1, \ldots, Q_\ell^{m_\ell}$ and $\lceil \frac{\epsilon m_\ell}{2} \rceil$ additional (unit) bins. This is thanks to our choice of δ: bins with slack δ in $d-1$ dimensions become unit bins after resource augmentation. Pleas note, that there is a guessing of the bin configuration hidden in

Lemma 4.3.

We know that S is the set of small items of the instance $\bigcup_\ell I_\ell$ which has a packing $\{I_\ell^i |\ i \in [m_\ell], \ell \in [d]\}$. Moreover, the small configuration of Q_ℓ^i is same or larger than small configuration of I_ℓ^i in all dimensions for each i and ℓ, as claimed by Lemma 4.3. Therefore we can use Lemma 4.4 to pack S into residual space of bins Q_ℓ^i and at most $\left\lceil \frac{\sum m_\ell}{2} \cdot (\epsilon + \frac{\epsilon^2}{d}) \right\rceil$ additional bins. If the packing procedure failed, we have either guessed wrong bin configuration when packing some Q_ℓ, or, if we have already tried all of them, we have wrong guesses of the rounding specification or numbers m_ℓ. Afterwards, we replace items of Q_ℓ by unrounded items according to the assignment found in realizability verification step.

Proof. [Proof of Lemma 5.1] The guessing stage takes polynomial time by Lemma 5.7. For a correct guess, I_T items are packed into $(1+\epsilon)m_T$ bins, I_S items are packed into exactly m_S bins. Remaining I_ℓ type items are packed in $\sum_\ell (m_\ell + \lfloor \frac{\epsilon m_\ell}{2} \rfloor)$ bins and small items into additional $\sum_\ell \frac{m_\ell}{2} \cdot (\epsilon + \frac{\epsilon^2}{d}) + 1$ bins. This makes at most $(1+\epsilon+\frac{\epsilon^2}{2d}) \sum_\ell m_\ell$. Puting this together, we use at most

$$m_S + (1+\epsilon)m_T + (1+\epsilon+\frac{\epsilon^2}{2d}) \sum_\ell m_\ell \leq (1+\epsilon+\frac{\epsilon^2}{2d})m'$$

bins, as $m' = m_S + m_T + \sum_\ell m_\ell$. □

1. Separate set S of small items

Guessing stage:

2. Guess $m', m_S, m_T, m_1, \ldots, m_d$
3. For each $\ell \in [d]$: create classes $W_\ell^{u,j}$ and guess a number $(w_\ell^{u,j} + t_\ell^{u,j} + s_\ell^{u,j})$ for each class
4. Check validity of the guesses using flow-net
5. Guess numbers $w_\ell^{u,j}, t_\ell^{u,j}, s_\ell^{u,j}$

Packing stage:

6. Find multiobjective matching and pack I_T into bins $\mathcal{B}_T$
7. Select I_S arbitrarily according to numbers $s_\ell^{u,j}$ and pack it into bins $\mathcal{B}_S$
8. For each $\ell \in [d]$: Pack I_ℓ into $\mathcal{B}_\ell$ using Algorithm 1
9. If any step fails, then go to the next guess in Guessing stage.
10. **return** packing $\mathcal{B}_S, \mathcal{B}_T, \mathcal{B}_1, \ldots, \mathcal{B}_d$

Algorithm 2: d-VP WITH $(\frac{(d+1)}{2} + \epsilon)$-ASYMPTOTIC APPROXIMATION

Proof. [of Theorem 5.1] From Lemma 3.1, there exists a packing of I into $m' := \lceil \frac{d+1}{2} \mathsf{Opt}(I) \rceil$ bins of type $\mathcal{B}_S, \mathcal{B}_T, \mathcal{B}_1, \ldots, \mathcal{B}_d$. Thanks to Lemma 5.1, we can find a packing of I in polynomial time into $(1+\epsilon+\frac{\epsilon^2}{2d})m'$ bins, what is at most $\lceil \frac{d+1}{2} \mathsf{Opt}(I) \rceil \cdot (1+\epsilon+\frac{\epsilon^2}{2d}) + 1$. □

5.3 Tight absolute approximation of $(3/2+\epsilon)$ for 2-VP

We give an algorithm with absolute approximation ratio of $(1.5+\epsilon)$ for 2-VP. Note that even for 1-D bin packing the lower bound on absolute approximation is 3/2. So even though 1-D BP and 2-VP are quite distinct in asymptotic approximability, they are arbitrarily close in the absolute approximability setting.

THEOREM 1.2 (RESTATED). *For any constant $\epsilon > 0$, there is a polynomial time algorithm with an absolute approximation ratio of $(1.5+\epsilon)$ for 2-VP.*

Proof. First, note that the problem is trivial if $\mathsf{Opt} = 1$. Let $\epsilon' = \epsilon/2$. By Theorem 5.1, we can get a packing into $(3/2+\epsilon')\mathsf{Opt} + c_2$ bins in polynomial time. There are two cases.

Case A. $\mathsf{Opt} > c_2/\epsilon$: In this case $(3/2+\epsilon')\mathsf{Opt} + c_2 \leq 3\mathsf{Opt}/2 + (2\epsilon')\mathsf{Opt} \leq (1.5+\epsilon)\mathsf{Opt}$.

Case B. $\mathsf{Opt} \leq c_2/\epsilon'$: Let $c_1 = c_2/\epsilon'$. As $\mathsf{Opt} \leq c_1$, there are at most $c_1 d/\beta$ large items, and thus there are only a constant number of large configurations. So in polynomial time, we can find the large configurations in optimal packing and small vectors can be added using assignment LP except at most dc_1 number of small items. We can pack these small items into one additional bin by choosing β small enough such that $\beta d c_1 \leq 1$. This gives a solution with value at most $\mathsf{Opt} + 1 \leq (3/2)\mathsf{Opt}$. □

6 Improved Approximation using R&A Framework

In this section, we combine the ideas from the previous section with the R&A framework to obtain improved approximation guarantees. We describe the 2-dimensional case in section 6.1 which is simpler, and then consider the d-dimensional case in section 6.2. As our algorithms do not round all the big items, we will apply the R&A framework in a somewhat non-standard way.

We begin by describing the implication of the R&A framework that we need, as stated in Theorem 6.1.

Let I be a d-VP instance, and let z^* denote the optimum value of the configuration LP for I. Recall, that in the R&A framework, we pick a parameter $\rho > 1$ and sample $(\ln \rho)z^*$ configurations from the LP solution, and then pack the residual items J separately.

Let $I' \subset I$ be a subset of items (in our applications I' will consist of small items and big items that are

rounded in some good packing). Let $\tilde{I}$ be the rounding of items in I' such that all big items are rounded to t_1 (constant) types, and suppose that there is some packing $\mathcal{P}$ of $\tilde{I}$ into m bins. Then we have the following:

THEOREM 6.1. *If we apply the R&A framework to I with parameter ρ and $J' = J \cap I'$ is the residual set of elements in I', then for any small constant $\epsilon > 0$, with high probability, there is a rounded packing $\mathcal{P}_{J'}$ of J' into $\frac{(1+\epsilon')m}{\rho} + c$ bins where c only depends on ϵ and ρ.*

Before proving the result, we need the following concentration inequality [25].

LEMMA 6.1. *(Independent Bounded Difference Inequality) [25] Let $X = (X_1, \ldots, X_n)$ be a family of independent random variables, with $X_j \in A_j$ for $j = 1, \ldots, n$, and $f : \prod_{j=1}^{n} A_j \to \mathbb{R}$ be a function such that $f(x) - f(x') \leq c_j$, whenever the vectors x and x' differ only in the j-th coordinate. Let $\mathbb{E}[f(X)]$ be the expected value of the random variable $f(X)$. Then for any $t \geq 0$,*

$$\mathbb{P}[f(X) - \mathbb{E}(f(X)) \geq t] \leq \exp(-2t^2 / \sum_{i=1}^{n} c_j^2).$$

Proof. (Theorem 6.1). The idea is simple but slightly technical. Consider some optimum rounded packing $\mathcal{P}$ of $\tilde{I}$ into m bins. We first write an auxiliary configuration LP (on $\tilde{I}$) that approximates the value of m to within an $O(1)$ additive error.

As there are t_1 (constant) types of big items and there can be at most d/β big items in a configuration, there are $(d/\beta)^{t_1}$ types of big configurations and thus $(d/\beta)^{t_1}$ types of small configurations. Let t_2 ($\leq (d/\beta)^{t_1}$) be the number of types of small configurations in $\mathcal{P}$. Let $\mathcal{R}_j$ be the set of big items of j'th type for $j \in [t_1]$ and $\mathcal{S}_j$ be the set of small items belonging to j'th type of small configurations for $j \in [t_2]$ in $\mathcal{P}$ Let $|\mathcal{R}_j| = r_j$ for $j \in [t_1]$ and let s_j be the number of bins with small configuration of type j for $j \in [t_2]$. Let h_j^C be the number of items of type $\mathcal{R}_j$ in configuration C and $g_j^C = 1$ if configuration C has small configuration of type j. Consider the following auxiliary $LP(\tilde{I})$ where $\mathcal{C}$ is the set of configurations of $\tilde{I}$:

$$\begin{aligned} \min & \sum_{C \in \mathcal{C}} x_C \\ & \sum_{C \in \mathcal{C}} h_j^C x_C \geq r_j \quad \forall\, j \in [t_1], \\ & \sum_{C \in \mathcal{C}} g_j^C x_C \geq s_j \quad \forall\, j \in [t_2], \\ & x_C \geq 0 \quad \forall\, C \in \mathcal{C}. \end{aligned} \tag{6.4}$$

Note that in $LP(\tilde{I})$ we essentially consider small configurations of each bin as a single item. Let $t = t_1 + t_2$. As the LP has t (which is a constant) number of constraints and the optimal integral solution of $LP(\tilde{I})$ has value m,

$$m \leq LP(\tilde{I}) + t. \tag{6.5}$$

Let $\epsilon_1 > 0$ be a small constant that we will choose later. Now define another instance J'' such that it contains $r_j(1+\epsilon_1)/\rho$ big items of type $j \in [t_1]$ and $s_{j'}(1+\epsilon_1)/\rho$ items of type $j' \in [t_2]$ such that the size of each item in type j' is equal to the size of j''th type of small configuration. So J'' is a shrunk down version of $\tilde{I}$ where each small configuration is replaced by a single item of that same size.

Now consider the following $LP(J'')$:

$$\begin{aligned} \min & \sum_{C \in \mathcal{C}} x_C \\ & \sum_{C \in \mathcal{C}} h_j^C x_C \geq \frac{r_j(1+\epsilon_1)}{\rho} \quad \forall\, j \in [t_1], \\ & \sum_{C \in \mathcal{C}} g_j^C x_C \geq \frac{s_j(1+\epsilon_1)}{\rho} \quad \forall\, j \in [t_2], \\ & x_C \geq 0 \quad \forall\, C \in \mathcal{C}. \end{aligned} \tag{6.6}$$

As the right hand side of constraints in $LP(\tilde{I})$ and $LP(J'')$ differ by a factor of $\frac{(1+\epsilon_1)}{\rho}$, we get:

$$\frac{(1+\epsilon_1)}{\rho} LP(\tilde{I}) = LP(J''). \tag{6.7}$$

Let $\mathsf{Opt}(J'')$ be the optimal integral solution to $LP(J'')$. Let J denote the set of residual items after applying $R\&A$ to I. Suppose we apply the same rounding as in $\tilde{I}$ to the items in J. Let us denote this instance to be $\tilde{J}$. We will show that the items in $\tilde{J}$ can be packed in $(1+\epsilon_2)\mathsf{Opt}(J'')$ bins, for a small positive constant ϵ_2 that we will choose later.

LEMMA 6.2. $\mathsf{Opt}(\tilde{J}) \leq (1+\epsilon_2)\mathsf{Opt}(J'')$.

Proof. As J is the set of uncovered residual items after $(\ln \rho)z^*$ iterations of randomized rounding. For any item $i \in I$,

$$\begin{aligned} \mathbb{P}(i \in J) &\leq (1 - \sum_{C \ni i} x_C^*/z^*)^{(\ln \rho) z^*} \\ &\leq (1 - 1/z^*)^{(\ln \rho) z^*} \leq 1/\rho. \end{aligned} \tag{6.8}$$

Here we use that $\sum_{C \ni i} x_C^* \geq 1$ for all $i \in I$.

So $\mathbb{E}[|\mathcal{R}_i \cap J|] \leq |\mathcal{R}_i|/\rho$ for all $i \in [t_1]$. Now we show concentration around the mean using Lemma 6.1.

Consider the function $f_i^B(x) = \mathcal{R}_i \cap J$, i.e., the number of items of type $\mathcal{R}_i$ in J. This is a function

of $X = (X_1, \ldots, X_{\lceil(\ln \rho)z^*\rceil})$, i.e., $\lceil(\ln \rho)z^*\rceil$ independent random variables (selected configurations in randomized rounding phase of R & A framework) where X_i corresponds to the configuration selected in ith iteration. Changing the value of any X_i may lead to the selection of a different configuration C' in place of configuration C in that iteration. So if vectors x and x' differ only in j'th coordinate,

$$f_i^B(x) - f_i^B(x') \quad \leq \quad \max\{|\mathcal{R}_i \cap C|, |\mathcal{R}_i \cap C'|\} \leq 1/\beta,$$

as there can be at most β (big) items of type $\mathcal{R}_i$ in a bin.

Therefore, by Lemma 6.1,

$$\mathbb{P}[f_i^B(X) - \mathbb{E}(f_i^B(X)) \geq \gamma z^*] \leq \exp\left(\frac{-2(\gamma z^*)^2}{\lceil(\ln \rho)z^*\rceil/\beta^2}\right).$$

Thus in the asymptotic case (when z^* is sufficiently large, i.e. at least $\frac{\ln \rho \cdot t_1 t_2}{\gamma^2 \beta^2}$) we can take union bound over all t_1 cases and with high probability for all large item classes, $f_i^B(X) - \mathbb{E}(f_i^B(X)) < \gamma z^*$. In particular, we set $\gamma = \frac{\epsilon_3^2}{t_1 t_2 \rho}$ for some small positive constant ϵ_3 to be choosen later. As in the packing LP for J'', we have $\frac{(1+\epsilon_1)}{\rho}|\mathcal{R}_i|$ big items of type $\mathcal{R}_i$, we can pack $\mathbb{E}(f_i^B(X)) \leq |\mathcal{R}_i|/\rho$ items in $\tilde{J}$ in the slots of same type items in J''. And all the remaining large items are packed into at most $t_1\gamma z^* \leq \frac{\epsilon_3^2 z^*}{t_2 \rho} \leq \frac{\epsilon_3^2 \mathsf{Opt}(J')}{t_2}$ in separate extra bins.

We now consider the small items. Consider a small configuration of type $j := (h_1, \ldots, h_d)$. Let $\mathcal{S}_j$ be the set of items in all small configurations of type j and s_j be the number of small configurations of type j. Let function $f_{\mathcal{S}_j}^k$ be $\sum_{v \in \mathcal{S}_j \cap J} v^k . \frac{1}{h_k}$, i.e., the length of items in $\mathcal{S}_j \cap J$ in k'th dimension scaled by a factor $\frac{1}{h_k}$. Intuitively the scaling makes each dimension of $\mathcal{S}_j$ to be one. This is again function of $\lceil(\ln \rho)z^*\rceil$ independent random variables (selected configurations). Thus as above, whenever the vectors x and x' differ only in one coordinate, we get:

$$\begin{aligned} & f_{\mathcal{S}_j}^k(x) - f_{\mathcal{S}_j}^k(x') \\ \leq \quad & \max\{\sum_{v \in \mathcal{S}_j \cap C} v^k . \frac{1}{h_k}, \sum_{v \in \mathcal{S}_j \cap C'} v^k . \frac{1}{h_k}\} \leq \frac{1}{h_k} \cdot h_k \leq 1. \end{aligned}$$

Thus by Lemma 6.1,

$$\mathbb{P}[f_{\mathcal{S}_j}^k(X) - \mathbb{E}(f_{\mathcal{S}_j}^k(X)) \geq \gamma z^*] \leq \exp\left(\frac{-2(\gamma z^*)^2}{\lceil(\ln \rho)z^*\rceil)}\right).$$

Now if $s_j < \frac{\epsilon_3 z^*}{t_1 t_2}$, one can pack these small items in all t_2 classes of small configurations into at most $\epsilon_3 z^*/t_1$ additional bins. Otherwise if $s_j \geq \frac{\epsilon_3 z^*}{t_1 t_2}$, then with high probability, $f_{\mathcal{S}_j}^k(X) \leq (\frac{s_j}{\rho} + \gamma z^*) \leq (1+\epsilon_3) \cdot \frac{s_j}{\rho}$. Thus, all small items in $\mathcal{S}_j \cap J$ can be packed into the corresponding small configurations of $\frac{(1+\epsilon_3)}{\rho} s_j$ bins in J'' fractionally by assigning each item to each bin with $\frac{\rho}{(1+\epsilon_3)s_j}$ fraction.

Using an assignment LP we can get an integral packing of all items into corresponding small configurations of $\frac{(1+\epsilon_3)}{\rho} s_j$ bins except possibly up to $\lceil d \cdot \frac{(1+\epsilon_3)}{\rho} s_j \rceil$ items. However as the original sizes of these items are $\leq \beta$ in all dimensions, we can pack them in $O(d\beta \frac{(1+\epsilon_3)}{\rho} s_j)$ additional bins.

So asymptotically, the number of bins $\mathsf{Opt}(\tilde{J})$ in the optimal packing of $(\tilde{J})$ is:

$$\begin{aligned} & (1+\epsilon_3)\mathsf{Opt}(J'') + \frac{\epsilon_3 \mathsf{Opt}(J'')}{t_1} + \\ & \lceil (d\beta + \beta^2 d) \frac{(1+\epsilon_3)}{\rho} \mathsf{Opt}(J'') \rceil + \frac{\epsilon_3^2 \mathsf{Opt}(J'')}{t_2} \end{aligned}$$

which is at most $(1+\epsilon_2)\mathsf{Opt}(J'')$ for a suitable ϵ_2. □

So asymptotically when z^* is sufficiently large,

$$\begin{aligned} \mathsf{Opt}(J) \quad \leq \quad & \mathsf{Opt}(\tilde{J}) \\ \leq \quad & (1+\epsilon_2)\mathsf{Opt}(J'') \qquad \text{(Lemma 6.2)} \\ \leq \quad & (1+\epsilon_2)(LP(J'') + t) \\ \leq \quad & \frac{(1+\epsilon_1)(1+\epsilon_2)LP(\tilde{I})}{\rho} + O(1) \quad \text{(Lemma 6.7)} \\ \leq \quad & (1+\epsilon')LP(\tilde{I})/\rho + O(1) \\ \leq \quad & (1+\epsilon')\mathsf{Opt}(\tilde{I})/\rho + c. \end{aligned}$$

This completes the proof. □

6.1 Approximation Algorithm for 2-D vector packing: Now we show a $(1 + \ln(3/2))$-asymptotic approximation algorithm for 2-VP. The main idea is the following: Recall that in the 3/2-asymptotic approximation, we cannot round all the big items. So, we will apply Theorem 6.1 suitably with I' as those big items that are rounded, plus small times (i.e. $I \setminus I'$ consists of arguments that will be unrounded, and placed using matching). We then show how to analyze the performance of this algorithm.

Consider some optimum packing of I. Call a bin B to be *compact*, if it has a subset of items K with $|K| \leq 2$ and $(\sum_{v \in K} v) \geq (1-\delta, 1-\delta)$. We call other remaining bins to be *noncompact*. We will show that R&A can be applied to non-compact bins successfully. Let m_C and m_N denote the number of compact and non-compact bins respectively and let $m = m_C + m_N$. The algorithm is described in the figure (Algorithm 3) and works as follows.

A. Guessing stage:

A1. Guess $\mathsf{Opt}(I), m_C$, and m_N, and take $\epsilon = \frac{\epsilon'}{6}$,

A2. For each $\ell \in [d]$:

Create classes $W_\ell^{u,j}$ and guess corresponding round vectors and $w_\ell^{u,j}, c_\ell^{u,j}$, the number of items in compact and noncompact bins in each size classes,

B. Packing stage:

B1. Use MOMB matching on the original item sizes to pack $c_\ell^{u,j}$ items from class $W_\ell^{u,j}$ into $(1+\delta)m_C$ bins,

B2. Apply R&A to the configuration LP with $\rho = 3/2$; B3. Apply Theorem 6.1 to pack the remaining items S using algorithm 2 into $m_N + 4\epsilon m + c + 3$ bins, B4. If packing in Step B1 or B3 fails, go to next guess.

Algorithm 3: $(1.405+\epsilon')$-APPROXIMATION FOR 2-VP

Separating compact bins: We separate pairs of large items belonging to the compact bins using an idea similar to the preceding section. First, we guess rounding classes $W_\ell^{u,j}$ for rounding of the non-compact bins together with the numbers $c_\ell^{u,j}, w_\ell^{u,j}$ of items from each class which are to be packed into compact bins and noncompact bins, respectively. The graph $G = (I, E)$ is a little bit different: we add edge between v and v', if they form a compact bin together. Then we find a MOMB matching in G satisfying guessed requirements $c_\ell^{u,j}$, and pack the matched items into roughly m_C separate bins.

Single large items bigger than $(1-\delta)$ in both coordinates can be separated easily in linear time. As shown in Lemma 6.4 below, the volume of the small items in compact bins is negligible, so they cause no harm to the R&A part (they can be packed into at most $2\delta m_C \le 2\delta m$ additional bins). Therefore, we do not need to separate them.

Packing non-compact bins using R&A: We apply R&A to the configuration LP of the original instance with $\rho = 3/2$ (we could also define a configuration LP on the instance obtained by removing the large items in compact bins, but it does not help in the analysis). This uses at most $\lceil m \cdot \ln \rho \rceil$ bins in this step. Let S denote the set of residual items (minus the large compact items already matched above). Now, we use an important property of non-compact bins, that is proved formally in Lemma 6.3: the items from m_N non-compact bins can be rounded to constant number of types and repacked into $\frac{3}{2}m_N$ bins Therefore, applying Theorem 6.1, $\mathsf{Opt}(S) \le m_N/\rho \le \frac{2}{3}m_N$. Then we pack S using the $\frac{3}{2}$-approximation algorithm into roughly m_N bins. Altogether, we used roughly $m_C + \lceil \ln \rho \rceil m + m_N$ bins.

We now prove the two lemmas mentioned above.

LEMMA 6.3. *All items in m_N non-compact bins can be packed into $\lceil 3m_N/2 \rceil (1+2\epsilon)$ bins where large items are rounded to a constant number of types.*

Proof. From structural properties in Lemma 5.2, for non-compact bins either there is a large item $p \in B$ such that $p \le (1/2, 1/2)$ and p is $> \delta$ in at least one coordinate, or there is a subset R of items in B such that $\sum_{p \in R} p \le (2\delta, 2\delta)$ and $B \setminus R$ has δ-slack in some dimension. Thus the removal of item(s) p or R from such bin B creates δ slack in B in at least one dimension. So from the proof of Lemma 5.3, any two such bins can be repacked into 3 bins such that either it has slack δ in one dimension or it contains two p type items. Now if the bins have δ-slack in at least one dimension, using Algorithm 1 we can get a rounded packing loosing only a factor $(1 + \epsilon + \epsilon^2/d) \le (1+2\epsilon)$ for small ϵ. On the other hand, for bins containing two p type items, one can just round p type items into (1/2,1/2). So for m_N non-compact bins we can produce a rounded-packing in $\lceil 3m_N/2 \rceil (1+2\epsilon)$ rounded bins. □

LEMMA 6.4. *All items in m_C compact bins can be packed into $(1+2\delta)m_C + 1$ bins where m_C bins contain single or two items and remaining $2\delta m_C + 1$ bins contain only small items.*

Proof. Given m_C compact bins, we can remove the items $\le (\delta, \delta)$ in the compact bins and greedily pack them into additional $\lceil \frac{m_C}{\lfloor 1/\delta \rfloor} \rceil \le 2\delta m_C + 1$ extra bins for small values of δ. □

6.1.1 Analysis of the Algorithm

THEOREM 1.1 (RESTATED). *For any small constant $\epsilon' > 0$, the algorithm above (Algorithm 3) has an asymptotic approximation ratio of $(1 + \ln(1.5) + \epsilon') \approx (1.405 + \epsilon')$ for 2-D vector packing.*

Proof. The binary search to find m ($:= \mathsf{Opt}(I)$) takes $O(\log n)$ time. For each guess there are $O(n^2)$ guesses for m_C, m_N. Also as there are total t_L (constant) number of round vectors, they can be guessed in $O(n^{t_L})$ time. Thus in polynomial time we can guess the suitable values for the *guessing stage* in the algorithm.

Now there are three steps in the *packing stage* of the algorithm. In the first step we pack $(1+\delta)m_C$ number of bins using multi-objective multi-budget matching on the original item sizes. In the randomized rounded step of R&A with $\rho = 3/2$ we use at most $\lceil (\ln \rho) z^* \rceil \le \ln \rho \cdot m + 1$

bins. In the last step we pack the remaining items in S into $m_N + 4\epsilon m + c_m + 3$ number of bins. So, for any feasible solution the total number of bins needed $(1+\delta)m_C + (\ln \rho)m + m_N + 4\epsilon m + O(1)$ which is atmost $(1 + \epsilon' + \ln \rho)(m_C + m_N) + O(1)$ bins, where $\epsilon' = 6\epsilon$. This gives $(1 + \ln(\frac{3}{2}) + \epsilon')$ asymptotic approximation if the algorithm returns a feasible solution.

To complete the proof we need to show that the algorithm always returns a feasible packing for a correct guess.

In the *packing using matching* step, we create an edge between two nodes v_i and v_j only if they form a compact bin together, i.e., $v_i + v_j \geq (1-\delta, 1-\delta)$. Note that we can always separate very large items $v_i \geq (1-\delta, 1-\delta)$ and pack them separately. In this step the items we pack in m_C bins might not be the same items in the compact bins in the optimal solution. However the packed items in these m_C bins are very similar to those items in the compact bins. We pack c_ℓ^{ij} items from W_ℓ^{ij} using MOMB matching into $(1+\delta)m_C$ bins as in Section 5 with $\nu = \delta$. So, for a correct guess at the end of *packing using matching*, each class W_ℓ^{ij} has only $\approx (c_\ell^{ij} + w_\ell^{ij}) - c_\ell^{ij} = w_\ell^{ij}$ items left.

So after packing using matching, from Lemma 6.3 and 6.4 remaining items have a rounded packing in By Theorem 6.1, S can be packed into at most $((3m_N/2)(1+2\epsilon)/\rho + 2\delta m_C + 2)(1+\epsilon) \leq m_N + 4\epsilon(m_N + m_C) + c + 3$ bins. This concludes the proof. □

6.2 Improved Approximation for d-Dimensional Vector Packing: For d-VP, we adopt a somewhat different approach. The main problem is that there can be d items in a bin such that if their sizes are rounded up, then we need $\Omega(d)$ many bins to pack them. So we cannot afford to round such items, and they must be packed together using some matching. However, there are no good results for multiobjective d-dimensional matching. So we work somewhat indirectly to use the result for (standard) matching.

The starting point are the following two structural properties for d-VP.

LEMMA 6.5. *If there exists a feasible packing $\mathcal{P}$ of μn_1 items into n_1 bins, a packing into $\frac{(\mu+1)}{2} \cdot n_1$ bins can be found in polynomial time.*

Proof. It suffices to show a packing into $\frac{(\mu+1)}{2} \cdot n_1$ bins where each bin contains one or two items. This packing can then be found using maximum matching. Let x be the number of bins in $\mathcal{P}$ containing an odd number of items. Pack one item from each such bin into a separate bin by itself. Each bin now has an even number of items, which can be packed in $(\mu n_1 - x)/2$ bins using matching, giving a packing with $x + (\mu n_1 - x)/2 \leq (\mu n_1 + n_1)/2$ bins. □

LEMMA 6.6. *For any $\epsilon > 0$, if there is a feasible packing of a d-VP instance I into m bins, then there is a packing into at most $(2+\epsilon)m + 1$ bins such that m bins contain at most $(d-1)$ items, and all other bins have $O(1)$ types of large items and small configurations.*

Proof. Let $B_1, \ldots, B_m$ be a feasible packing of I. Each bin B_i, for $i \in 1, \ldots, m$, we apply an argument similar to the one in the proof of Lemma 5.5. Let $\delta = \epsilon/(14d^2)$.

Consider bin B_i. Let C be the list of coordinates where B_i has slack less than δ. First, we remove large items from B_i iteratively such that each removal creates slack in at least one coordinate of C. Call these removed items V_i. If $|C| \leq 1$, we are done with bin B_i.

However, if $|C| > 1$ and there are no large items whose removal would create slack in some coordinate of C, then all items currently in B_i must be smaller than δ in all coordinates in C. In this case, we use Lemma 5.4 to find a set R_i such that $\sum_{p \in R_i} p \leq \kappa\delta$ in all dimensions and $B_i \setminus (V_i \cup R_i)$ contains slack in all dimensions.

We pack V_i in a separate bin B_i', and clearly this bin contains at most $d-1$ items. The items in R_i for $i = 1, \ldots, m$, can be packed greedily into at most $2m\kappa\delta = \epsilon m$ bins in total such that each bin still has δ slack in each dimension.

As all bins except the B_i' for $i = 1, \ldots, m$, have slack in at least $d-1$ dimensions, the items in these bins can be rounded as discussed in previous sections. □

We now use these results to give our algorithm (Algorithm 4).

Let $m := \mathsf{Opt}(I)$. Consider the packing in $(2+\epsilon)m + 1$ bins in Lemma 6.6. Let I_D be the set of items in those m bins containing at most $(d-1)$ items. Let $I_N = I \setminus I_D$, are the items in the remaining rounded bins.

We apply $R\&A$ to I with $\rho = (d+1)/2$ and let J be the residual set of items. The algorithm will pack (after suitable guessing) the items in $I_D \cap J$ using Lemma 6.5 and MOMB matching. As we will see later this will use about $3m/2$ bins.

The items in $I_N \cap J$ are be packed into nearly $m(1+\epsilon)/\rho$ bins using Theorem 6.1, we show that remaining items have a rounded packing into nearly $\frac{m}{\rho} + O(1)$ bins.

6.2.1 Analysis

THEOREM 1.3 (RESTATED). *For any constant ϵ with $\epsilon < \frac{1}{3d^2}$, there is a poly-time algorithm (Algorithm 4)*

A. Guessing stage:
A1. Guess $m := \mathsf{Opt}(I)$ and create rounding classes $W_\ell^{u,j}$.
B. Packing stage:
B1. Choose $\rho = \frac{(d+1)}{2}$ and solve configuration LP and apply randomized rounding for $\lceil LP(I) \cdot \ln(\rho) \rceil$ iterations,
B2. Let J be the set of items left at the end of randomized rounding. Guess numbers $\mu_\ell^{ij} = |(I_D \cap J) \cap W_\ell^{ij}|, \nu_\ell^{ij} = |(I_N \cap J) \cap W_\ell^{ij}|$,
B3. Use MOMB matching to pack $\mu_\ell^{u,j}$ items from classes $W_\ell^{u,j}$ into $(1+3\epsilon)(d-1)m/(2\rho) + (1+\epsilon)m/2 + O(1)$ bins,
B4. Pack remaining items using Algorithm 2 in $(1+10\epsilon)m/\rho + O(1)$ bins,
B5. If packing in Step B3 or B4 fails, go to next guess.

Algorithm 4: $(\ln(d+1) + 1.5 - \ln 2 + O(\epsilon))$-APPROXIMATION FOR d-VP.

with an asymptotic approximation ratio of $(1.5 + \ln(d+1) - \ln 2 + \epsilon) \approx \ln(d+1) + 0.807 + \epsilon$ *for d-VP.*

Proof. Let $m = \mathsf{Opt}(I)$.

The $R\&A$ step uses up to $\ln \rho \cdot m + 1$ bins. We count the number of bins used in other steps.

By (6.8), for any item $i \in I$, $\mathbb{P}(i \in J) = 1/\rho$, and hence $\mathbb{E}[|I_D \cap S|] = (d-1)m/\rho$. Using Lemma 6.1 as in Theorem 6.1, we have that with high probability,

$$|I_D \cap S| \leq \frac{(1+\epsilon)\mathbb{E}[|I_D \cap S|]}{\rho} + O(1).$$

By Lemma 6.5 (and multi-objective multi-budget matching) these items from $I_D \cap S$ can be packed into

$$\left(\frac{(d-1+\rho)}{2\rho}\right) m + O(\epsilon m)$$

bins.

Finally, as there is a rounded-packing of items in I_N into $m(1+\epsilon)$ bins, by Theorem 6.1, there is a packing of $I_N \cap S$ into $\frac{m(1+\epsilon)(1+\epsilon)}{\rho} + O(1)$ bins.

Putting this all together, the number of bins needed is at most

$$\begin{aligned} & m \cdot \ln(\rho) + (1+O(\epsilon))m\left(\frac{(d-1+\rho)}{2\rho} + \frac{1}{\rho}\right) + O(1) \\ \leq \quad & (1+O(\epsilon))m \cdot \left(\ln(\rho) + 1/2 + \frac{(d+1)}{2\rho}\right) \end{aligned}$$

The choice of $\rho = (d+1)/2$ optimizes the expression above, to give an $\ln(d+1) + 3/2 - \ln 2$ asymptotic approximation. □

We note that all algorithms in the paper can be derandomized using the potential function based standard arguments in [2].

The algorithm above improves the previous $1 + \ln d$ approximation for all $d > 4$. For $d = 2$ and 3, the algorithms in the previous sections give an approximation of $(1.405 + \epsilon)$ and $(2 + \epsilon)$ which also improves previously known bounds for these cases.

7 Conclusion

Given our current knowledge, it is possible that for each constant d there exists a (say) 2 approximation algorithm for d-VP running in time $n^{f(d)}$ for some arbitrary function f. Resolving this is an extremely interesting question. Our bounds can probably be improved further for small values of d by proving and using multi-objective multi-budget variant of d-dimensional matching for $d > 2$. However, this approach is unlikely to give a $(1-\delta)\ln d$ approximation for large d, given the $\Omega(d/\ln d)$ lower bound for d-dimensional matching.

Acknowledgments. We thank Prasad Tetali for helpful discussions.

References

[1] Yossi Azar, Ilan R. Cohen, Seny Kamara, and Bruce Shepherd, *Tight bounds for online vector bin packing*, in STOC, 2013, pp. 961–970.
[2] Nikhil Bansal, Alberto Caprara, and Maxim Sviridenko, *Improved approximation algorithms for multidimensional bin packing problems*, in FOCS, 2006, pp. 697–708.
[3] Nikhil Bansal and Arindam Khan, *Improved approximation algorithm for two-dimensional bin packing*, in SODA, 2014, pp. 13–25.
[4] Nikhil Bansal, Kang-Won Lee, Viswanath Nagarajan, and Murtaza Zafer, *Minimum congestion mapping in a cloud*, in PODC, 2011, pp. 267–276.
[5] Nikhil Bansal and Maxim Sviridenko, *Two-dimensional bin packing with one-dimensional resource augmentation*, Discrete Optimization, 4 (2007), pp. 143–153.
[6] Nikhil Bansal, Tjark Vredeveld, and Ruben van der Zwaan, *Approximating vector scheduling: Almost matching upper and lower bounds*, in LATIN, 2014, pp. 47–59.
[7] Alberto Caprara and Paolo Toth, *Lower bounds and algorithms for the 2-dimensional vector packing problem*, Discrete Applied Mathematics, 111 (2001), pp. 231–262.
[8] Chandra Chekuri and Sanjeev Khanna, *On multidimensional packing problems*, SIAM J. Comput., 33 (2004), pp. 837–851.
[9] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen, *Multi-budgeted matchings and matroid in-*

tersection via dependent rounding, in SODA, 2011, pp. 1080–1097.

[10] Wenceslas Fernandez de la Vega and George S. Lueker, *Bin packing can be solved within 1+epsilon in linear time*, Combinatorica, 1 (1981), pp. 349–355.

[11] E. A. Dinic, *An algorithm for solution of a problem of maximum flow in a network with power estimation*, Soviet Mathematics Doklady, 11 (1970), p. 12771280.

[12] Friedrich Eisenbrand, Dömötör Pálvölgyi, and Thomas Rothvoss, *Bin packing via discrepancy of permutations*, ACM Transactions on Algorithms, 9 (2013), p. 24.

[13] Hans Frenk, János Csirik, Martine Labbé, and Shuzhong Zhang, *On the multidimensional vector bin packing*, University of Szeged. Acta Cybernetica, (1990), pp. 361–369.

[14] Alan M. Frieze and M. R. B. Clarke, *Approximation algorithms for the m-dimensional 0–1 knapsack problem: Worst-case and probabilistic analyses*, European Journal of Operational Research, 15 (1984), pp. 100–109.

[15] Michael R. Garey, Ronald L. Graham, David S. Johnson, and Andrew C. Yao, *Resource constrained scheduling as generalized bin packing*, Journal of Combinatorial Theory, Series A, 21 (1976), pp. 257–298.

[16] Martin Grötschel, László Lovász, and Alexander Schrijver, *Geometric algorithm and combinatorial optimization*, Algorithms and Combinatorics: Study and Research Texts, Springer-Verlag, Berlin, (1988).

[17] Bernard T. Han, George Diehr, and Jack S. Cook, *Multiple-type, two-dimensional bin packing problems: Applications and algorithms*, Annals of Operations Research, 50 (1994), pp. 239–261.

[18] Johan Håstad, *Clique is hard to approximate within* $n^{1-\epsilon}$, in FOCS, 1996, pp. 627–636.

[19] Rebecca Hoberg and Thomas Rothvoss, *A logarithmic additive integrality gap for bin packing*, CoRR, abs/1503.08796 (2015).

[20] Sungjin Im, Nathaniel Kell, Janardhan Kulkarni, and Debmalya Panigrahi, *Tight bounds for online vector scheduling*, in FOCS (to appear), 2015.

[21] Klaus Jansen and Roberto Solis-Oba, *Rectangle packing with one-dimensional resource augmentation*, Discrete Optimization, 6 (2009), pp. 310–323.

[22] Narendra Karmarkar and Richard M. Karp, *An efficient approximation scheme for the one-dimensional bin-packing problem*, in FOCS, 1982, pp. 312–320.

[23] Hans Kellerer and Vladimir Kotov, *An approximation algorithm with absolute worst-case performance ratio 2 for two-dimensional vector packing*, Operations Research Letters, 31 (2003), pp. 35–41.

[24] Arindam Khan, *Approximation algorithms for multi-dimensional bin packing*, PhD Thesis, Georgia Institute of Technology, Atlanta, USA, 2015.

[25] Colin McDiarmid, *Concentration*, in Probabilistic methods for algorithmic discrete mathematics, Springer, 1998, pp. 195–248.

[26] Adam Meyerson, Alan Roytman, and Brian Tagiku, *Online multidimensional load balancing*, in APPROX, 2013, pp. 287–302.

[27] Rina Panigrahy, Kunal Talwar, Lincoln Uyeda, and Udi Wieder, *Heuristics for vector bin packing*. `http://research.microsoft.com`, 2011.

[28] Lars Dennis Prädel, *Approximation Algorithms for Geometric Packing Problems*, PhD thesis, Kiel, Christian-Albrechts-Universität, 2012.

[29] Thomas Rothvoss, *Approximating bin packing within O(log OPT * log log OPT) bins*, in FOCS, 2013, pp. 20–29.

[30] S. C. Sarin and W. E. Wilhelm, *Prototype models for two-dimensional layout design of robot systems*, IIE transactions, 16 (1984), pp. 206–215.

[31] Hadas Shachnai and Tami Tamir, *Approximation schemes for generalized two-dimensional vector packing with application to data placement*, Journal of Discrete Algorithms, 10 (2012), pp. 35–48.

[32] Frits C. R. Spieksma, *A branch-and-bound algorithm for the two-dimensional vector packing problem*, Computers & operations research, 21 (1994), pp. 19–25.

[33] D. Vercruyssen and H. Muller, *Simulation in production*, A report of the University of Gent, Belgium, (1987).

[34] Gerhard J. Woeginger, *There is no asymptotic ptas for two-dimensional vector packing*, Information Processing Letters, 64 (1997), pp. 293–297.