An EPTAS for Cardinality Constrained Multiple Knapsack via Iterative Randomized Rounding

Ilan Doron-Arad
Computer Science Department, Technion, Haifa, Israel

Ariel Kulik
Computer Science Department, Technion, Haifa, Israel

Hadas Shachnai
Computer Science Department, Technion, Haifa, Israel

Abstract

In [Math. Oper. Res., 2011], Fleischer et al. introduced a powerful technique for solving the generic class of *separable assignment problems* (SAP), in which a set of items of given values and weights needs to be packed into a set of bins subject to separable assignment constraints, so as to maximize the total value. The approach of Fleischer at al. relies on solving a configuration LP and sampling a configuration for each bin independently based on the LP solution. While there is a SAP variant for which this approach yields the best possible approximation ratio, for various special cases, there are discrepancies between the approximation ratios obtained using the above approach and the state-of-the-art approximations. This raises the following natural question: Can we do better by *iteratively* solving the configuration LP and sampling a few bins at a time?

To assess the potential of the iterative approach we consider a specific SAP variant as a case-study, Uniform Cardinality Constrained Multiple Knapsack, for which we answer this question affirmatively. The input is a set of items, each has a value and a weight, and a set of uniform capacity bins. The goal is to assign a subset of the items of maximum total value to the bins such that (*i*) the capacity of any bin is not exceeded, and (*ii*) the number of items assigned to each bin satisfies a given *cardinality* constraint. While the technique of Fleischer et al. yields a $\left(1-\frac{1}{e}\right)$-approximation for the problem, we show that iterative randomized rounding leads to *efficient polynomial time approximation scheme (EPTAS)*, thus essentially resolving the complexity status of the problem. Our analysis of iterative randomized rounding may be useful for solving other SAP variants.

2012 ACM Subject Classification Theory of computation

Keywords and phrases multiple knapsack, cardinality constraint, EPTAS, iterative randomized rounding

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.27

Category APPROX

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2308.12622` [9]

Funding *Ariel Kulik*: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 852780-ERC (SUBMODULAR)

1 Introduction

We consider problems in the class of maximizing assignment problems with packing constraints, also known as separable assignment problems (SAP). A general problem in this class is defined by a set of bins and a set of items to be packed in the bins. There is a *value* v_{ij} (also called *profit* sometimes) associated with assigning item i to bin j. We are also given a separate packing constraint for each bin j. The goal is to find an assignment of a subset of the items to the bins which maximizes the total value accrued. This class includes several well studied problems such as the generalized assignment problem (GAP).

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024).
Editors: Amit Kumar and Noga Ron-Zewi; Article No. 27; pp. 27:1–27:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In [12], Fleischer at al. introduced a powerful technique for solving SAP and its variants. The technique relies on first solving a configuration linear programming (configuration-LP) relaxation of the problem. Subsequently, configurations (i.e., feasible subsets of items for a single bin) are sampled independently according to a distribution specified by the LP solution to obtain an integral solution for the given instance. For many SAP variants, such as GAP, the approximation guarantee of the resulting algorithm is $(1 - {}^{1}/_{e})$.

Intuitively, we can do better using the following iterative randomized rounding approach: Iteratively solve a configuration LP relaxation of the problem for the remaining items and bins and sample a *few* configurations based on the distribution specified by the LP solution, until all bins are used. We note that if the LP is solved only once and all of the bins are packed based on the solution, then we have exactly the algorithm of Fleischer et al. [12]. This raises the following question:

Can iterative randomized rounding improve the approximation ratio of [12]?

As shown in [12], under standard complexity assumptions, there is a SAP variant for which their approximation ratio of $(1 - {}^{1}/_{e})$ is the best possible. However, for various special cases (such as MULTIPLE KNAPSACK and GAP), there are discrepancies between the approximation guarantee obtained using the algorithm of [12] and the state-of-the-art approximations. This indicates that the iterative approach may potentially lead to improved approximation for some variants (compared to [12]). Hence, to assess the potential of the iterative approach we focus as a case study on one interesting SAP variant, namely, UNIFORM CARDINALITY CONSTRAINED MULTIPLE KNAPSACK (CMK). Specifically, we show that iterative randomized rounding is superior to the technique of [12] and use it to essentially resolve the complexity status of this problem.

An input for CMK consists of a set of items, each has a value and a weight, and a set of uniform capacity bins. The goal is to assign a subset of the items of maximum total value to the bins such that (*i*) the total weight of items in each bin does not exceed its capacity, and (*ii*) the number of items assigned to each bin satisfies a given cardinality constraint.[1] CMK has real-world applications in cloud computing, as well as in manufacturing systems and radio networks (see the full version of the paper [9]).

1.1 Related Work

Iterative randomized rounding of configuration-LPs has been recently used for obtaining the current state-of-the-art approximation for *vector bin packing* in [17]. In this problem, the goal is to pack a set of items, each given by a d-dimensional size vector for some $d > 1$, in a minimum number of d-dimensional bins, where a subset of items fits in a bin if it adheres to the capacity constraints in all dimensions. We are not aware of an application of iterative randomized rounding of configuration-LPs for *maximization* problems.

The MULTIPLE KNAPSACK WITH UNIFORM CAPACITIES (UMK) problem is the special case of CMK with no cardinality constraint, or equivalently, where the cardinality constraint is larger than the total number of items. In the more general MULTIPLE KNAPSACK (MK) problem, the capacity of the bins may be arbitrary. In terms of approximation algorithms, UMK and MK are well understood. A *polynomial time approximation scheme* PTAS for

[1] See a more formal definition in Section 1.3.

UMK was given by Kellerer [16]. Later, Chekuri and Khanna [5] developed the first PTAS for MK and ruled out the existence of a *fully* PTAS (FPTAS), already for UMK with only two bins. Jansen designed more involved *efficient* PTAS (EPTAS) for MK [13, 14], thus resolving the complexity status of the problem.[2]

For CMK, a randomized $(1 - 1/e)$-approximation follows from the previously mentioned results of Fleischer et al. [12] for SAP. More specifically, the authors present a randomized algorithm for SAP whose approximation guarantee is $((1 - 1/e) \cdot \beta)$, where β is the best approximation ratio for the single bin subproblem.[3] A slightly more efficient approximation ratio for CMK follows from a recent result of Cohen et al. [7] who give a randomized $(1 - \ln(2)/2 - \varepsilon) \approx 0.653$-approximation for *uniform 2-dimensional vector multiple knapsack*. In this problem, the cardinality constraint of CMK is generalized to a second knapsack constraint.

We note that a PTAS for CMK can be obtained using ideas of [5]. We outline the main steps. First, item values are discretized into $O(\log n)$ value classes, where n is the number of items. Then, enumeration is used to roughly determine the number of items taken from each value class. Clearly, one should take from each value class the items with smallest weights, leading to a reduced problem of packing sufficient items from each value class in the m given (uniform) bins. Packing these items in $(1 + \varepsilon) \cdot m + O(1)$ uniform bins can be done using an asymptotic FPTAS for bin packing with cardinality constraint [11] (or more generally, for bin packing with a partition matroid [8]). Finally, the algorithm keeps the m bins with highest values. We note that the running time of the enumeration step is very high. This leaves open the question whether CMK admits an EPTAS.

1.2 Our Results

Our main contribution is in showing that iterative randomized rounding can substantially improve the approximation guarantee of the configuration-LP rounding approach of [12]. The analysis is based on concentration bounds; thus, our iterative algorithm is applied to a slightly restricted subclass of CMK instances in which the value of each configuration is relatively small, and the number of bins is large w.r.t. the given error parameter. Recall that even for two bins the problem does not admit an FPTAS [5]. Hence, we do not expect the iterative approach to work for a small number of bins. More specifically, given an error parameter $\varepsilon \in (0, 0.1)$ we say that a CMK instance $\mathcal{I}$ is ε-*simple* if (i) every feasible subset of items C which can be packed in a single bin has value at most $\varepsilon^{30} \cdot \mathrm{OPT}(\mathcal{I})$, (ii) $m > \exp(\exp(\varepsilon^{-30}))$, and (iii) $\varepsilon \cdot m \in \mathbb{N}$, where m is the number of bins and $\mathrm{OPT}(\mathcal{I})$ is the optimum value of $\mathcal{I}$.[4] For clarity, we first state our algorithmic result for the subclass of ε-simple CMK instances (see Section 3 for more details).

▶ **Theorem 1.** *For every $\varepsilon \in (0, 0.1)$ and an ε-simple* CMK *instance $\mathcal{I}$, iterative randomized rounding (see Algorithm 1) returns a $(1 - \varepsilon)$-approximation for $\mathcal{I}$ in time $\left(\frac{|\mathcal{I}|}{\varepsilon}\right)^{O(1)}$, where $|\mathcal{I}|$ is the encoding size of $\mathcal{I}$.*

An in depth look into the algorithm of Fleischer et al. [12] reveals that the approximation ratio of their algorithm on ε-simple instance is not better than $\left(1 - \frac{1}{e}\right)$, indicating the improved ratio in Theorem 1 stems from the use of the iterative approach.

---

[2] We give formal definitions of approximation schemes in Section 2.

[3] The paper [12] shows that the existence of an FPTAS for the single bin subproblem, as in the case of 0/1-knapsack with cardinality constraint, implies a $(1 - 1/e)$-approximation for the corresponding variant of SAP.

[4] In our discussion of ε-simple instances, we did not attempt to optimize the constants.

We give a simple reduction showing that our algorithm for ε-simple instances yields a randomized EPTAS for general CMK instances.

This essentially resolves the complexity status of CMK, since an FPTAS is ruled out [5].

▶ **Theorem 2.** *There is a randomized* EPTAS *for* CMK*.*

For the proof of Theorem 2 see Section 3.

1.3 Technical Overview

In the following, we outline our algorithmic approach and its analysis. For clarity, we focus in this section on high-level ideas and omit some technical details to improve clarity. We start with a more formal definition of CMK. An instance of CMK consists of a set of items I, a weight function $\mathbf{w} : I \to [0, 1]$, a value function $\mathbf{v} : I \to \mathbb{R}_{\geq 0}$, a number of bins $m \in \mathbb{N}_{>0}$, and a cardinality constraint $k \in \mathbb{N}_{>0}$. A *solution* is a tuple $(C_1, \ldots, C_m)$ such that for all $j \in \{1, \ldots, m\}$ it holds that $C_j \subseteq I$, $|C_j| \leq k$ and $\mathbf{w}(C_j) = \sum_{i \in C_j} \mathbf{w}(i) \leq 1$. The *value* of the solution $(C_1, \ldots, C_m)$ is $\sum_{i \in S} \mathbf{v}(i)$ where $S = \bigcup_{j=1}^{m} C_j$. The goal is to find a solution of maximum value. Note that we allow the sets $C_1, \ldots, C_m$ to intersect, but if an item appears in multiple sets its value is counted only once.

The Algorithm

Our algorithm applies an iterative randomized rounding approach based on a configuration-LP. The use of such linear program dates back to the work of Karmarkar and Karp on bin packing [15], and such linear programs are commonly used in approximation algorithms for resource allocation problems (e.g., [3, 1, 12, 13, 17, 7]).

We use a *configuration polytope* $P(\ell) \subseteq [0, 1]^I$, where $\bar{y} \in P(\ell)$ can be intuitively interpreted as "there is a way to fractionally pack the items into ℓ bins such that each item $i \in I$ is packed $\bar{y}_i$ times". The algorithm takes as an input a value $\varepsilon > 0$ which serves as a discretization factor and determines the approximation ratio. Our iterative approach uses $\frac{1}{\varepsilon}$ iterations, and each iteration packs $\varepsilon \cdot m$ of the remaining bins. Therefore, at the beginning of the j-th iterations, $(j - 1) \cdot \varepsilon \cdot m$ bins were packed (in previous iterations) and $(1 - (j - 1) \cdot \varepsilon) \cdot m$ bins are still empty. We use S_j to denote the set of items that were not packed by the end of the j-th iteration (and thus are still available for packing). The main steps of the algorithm are as follows.

1. Initialize $S_0 \leftarrow I$ to be all the items.
2. For j from 1 to $\frac{1}{\varepsilon}$ do:
   a. Solve the linear program

$$\begin{aligned} \max \quad & \sum_{i \in I} \bar{y}_i \cdot \mathbf{v}(i) \\ \text{s.t.} \quad & \bar{y} \in P\left(m \cdot (1 - (j - 1) \cdot \varepsilon)\right) \\ & \bar{y}_i = 0 \qquad\qquad \forall i \in I \setminus S_{j-1} \end{aligned} \tag{1}$$

   That is, we want to obtain a maximum value using $m \cdot (1 - (j - 1) \cdot \varepsilon)$ bins and only items in S_{j-1}. Let $\bar{y}^j$ be the solution found.

   b. Sample $\varepsilon \cdot m$ bins according to the solution $\bar{y}^j$ (defined more formally in later); update S_j to be S_{j-1} minus all the items packed in the current iteration.
3. Return the collection of m packed bins.

The linear program in (1) can be approximated efficiently, but cannot be solved exactly in polynomial time. For the purpose of this technical overview, we assume it can be solved exactly. In Item 2b we use the randomized rounding technique for configuration LPs of Fleischer et al. [12]. The same randomized rounding technique is commonly used by other algorithms (e.g., [3, 1, 17, 2]). We give the full details on the sampling process in Section 3.

High Level Analysis

Let Q_j be the set of items packed in the j-th iteration (that is, $Q_j = S_j \setminus S_{j-1}$). In the j-th iteration, the linear program uses $m \cdot (1 - (j-1) \cdot \varepsilon)$ bins and attains value of $\sum_{i \in I} \bar{y}^j_i \cdot \mathbf{v}(i)$, with an average value of $\frac{\sum_{i \in I} \bar{y}^j_i \cdot \mathbf{v}(i)}{m \cdot (1-(j-1)\cdot\varepsilon)}$ per bin. As the number of bins sampled in each iteration is *small*, it can be shown that with high probability the average value per bin in the packing generated by the randomized rounding is roughly the same as the average value in the fractional solution. That is,

$$\frac{\sum_{i \in Q_j} \mathbf{v}(i)}{\varepsilon \cdot m} \approx \frac{\sum_{i \in I} \bar{y}^j_i \cdot \mathbf{v}(i)}{m \cdot (1 - (j-1) \cdot \varepsilon)},$$

or equivalently,

$$\sum_{i \in Q_j} \mathbf{v}(i) \approx \varepsilon \cdot \frac{\sum_{i \in I} \bar{y}^j_i \cdot \mathbf{v}(i)}{1 - (j-1) \cdot \varepsilon}.$$

Observe the left hand term is the value attained from items packed in the j-th iteration. In each iteration of the algorithm the distribution by which the bins are sampled is updated, so the algorithm does not pack items already packed in previous iterations (by the constraints $\bar{y}_i = 0$ for $i \in I \setminus S_{j-1}$ in (1)). Thus, we have that $Q_1, \ldots, Q_{\varepsilon^{-1}}$ are disjoint. It follows that the value of the solution returned by the algorithm is

$$\mathbf{v}(I \setminus S_{\varepsilon^{-1}}) = \sum_{j=1}^{\varepsilon^{-1}} \sum_{i \in Q_j} \mathbf{v}(i) \approx \varepsilon \cdot \sum_{j=1}^{\varepsilon^{-1}} \frac{\sum_{i \in I} \bar{y}^j_i \cdot \mathbf{v}(i)}{1 - (j-1) \cdot \varepsilon} \tag{2}$$

Ideally, we would like the average value per bin to be (at least) $\frac{\text{OPT}}{m}$ in each of the solutions $\bar{y}^j$, where OPT is the value of the optimal solution of the instance. That is, the average value per bin in each of the iterations remains the average value per bin in the optimum. As $\bar{y}^j$ conceptually uses $m \cdot (1 - (j-1)\varepsilon)$ bins, this implies that

$$\sum_{i \in I} \bar{y}^j_i \cdot \mathbf{v}(i) \gtrsim \frac{\text{OPT}}{m} \cdot m \cdot (1 - (j-1) \cdot \varepsilon) = \text{OPT} \cdot (1 - (j-1) \cdot \varepsilon), \tag{3}$$

for every $j \in [\varepsilon^{-1}]$. If we assume (3) holds and plug it into (2), we get that the value of the solution returned by the algorithm is

$$\mathbf{v}(I \setminus S_{\varepsilon^{-1}}) \approx \varepsilon \cdot \sum_{j=1}^{\varepsilon^{-1}} \frac{\sum_{i \in I} \bar{y}^j_i \cdot \mathbf{v}(i)}{1 - (j-1) \cdot \varepsilon} \gtrsim \varepsilon \cdot \sum_{j=1}^{\varepsilon^{-1}} \frac{\text{OPT} \cdot (1 - (j-1) \cdot \varepsilon)}{1 - (j-1) \cdot \varepsilon} = \text{OPT}.$$

That is, the algorithm returns a solution of value close to OPT (not strictly better naturally), assuming (3) holds. This leaves us with the goal of showing that (3) holds with high probability.

Linear Structures and Equation (3)

To show that (3) holds we define a random vector $\bar{\gamma}^j \in [0,1]^I$ for every $j \in [\varepsilon^{-1}]$. We use $\bar{\gamma}^j$ to lower bound the value of the configuration-LP. We show that with high probability (i) the value of $\bar{\gamma}^j$ (that is, $\sum_{i\in I} \bar{\gamma}^j_i \cdot \mathbf{v}(i)$) is $\approx (1-(j-1)\varepsilon) \cdot \text{OPT}$ and (ii) $\bar{\gamma}^j \in P((1+\delta)\cdot m_j)$ where $m_j = (1-(j-1)\cdot\varepsilon)\cdot m$ is the number of remaining bins at the beginning of the j-th iteration and $\delta > 0$ is small. Once properties (i) and (ii) are shown, it follows that $\frac{\bar{\gamma}^j}{1+\delta}$ is a solution of high value for the linear program in the j-th iteration, and (3) immediately follows as the algorithm finds an optimal solution in every iteration. Property (i) is shown using a simple calculation of the expected value of the vector $\bar{\gamma}^j$ followed by an application of a concentration bound which shows that with high probability the value of $\bar{\gamma}^j$ does not deviate afar from its expected value. Showing property (ii) is more challenging.

The polytope $P(\ell)$ can be represented via a finite set of linear constraints $\mathcal{S} \subseteq \mathbb{R}^I_{\geq 0}$ by $P(\ell) = \{\bar{y} \in [0,1]^I \,|\, \forall \bar{u} \in \mathcal{S} :\ \bar{u}\cdot\bar{y} \leq \ell\}$ (the set $\mathcal{S}$ is the same for every ℓ). While $\mathcal{S}$ is finite, its size is non-polynomial in the input instance. A naive approach to show that $\bar{\gamma}^j \in P((1+\delta)m_j)$ is to consider each constraint $\bar{u} \in \mathcal{S}$ separately, and apply concentration bounds to show $\bar{y}\cdot\bar{u} \lesssim m_j$ with high probability. Subsequently, the union bound can be used to lower bound the probability that $\bar{y}\cdot\bar{u} \lesssim m_j$ for every $\bar{u} \in \mathcal{S}$ simultaneously. However, due to the large number of vectors in $\mathcal{S}$, a direct application of the union bound does not lead to such useful lower bound.

We use a *linear structure* to overcome the above challenge. The linear structure provides an approximate representation of the configuration polytope using a small number of constraints (that is, the number of constraints only depends on ε). As the number of constraints is reduced, we can now apply the above logic successfully − use a concentration bound to show that each constraint of the linear structure holds independently with high probability, and then use the union bound to show that all the constraints hold simultaneously with high probability. By the properties of the linear structure, once we show that all constraints hold, we are guaranteed that $\bar{\gamma}^j \in P((1+\delta)\cdot m_j)$, as stated in (ii).

The concept of linear structure was introduced in [17]. It is essentially a non-constructive version of the *subset oblivious* algorithms used by the Round&Approx framework of [3]. We construct the linear structure for CMK based on ideas from [17, 1]. The structure leverages the relatively simple structure of the cardinality constraint.

Technical Contribution

In this paper, we present the first use of an iterative randomized rounding approach of a configuration-LP for a maximization problem. As such, the paper provides the basic foundations required for the analysis of iterative randomized rounding for maximization problems. Iterative randomized rounding of a configuration-LP has been recently used for bin packing problems in [17]. Indeed, in some places the analysis only requires simple adaptations of ideas from [17]. In other parts, the adaptation is more challenging.

These challenges arise mainly due to the fact that while in bin packing all the remaining items must be fully packed by the configuration-LP, in maximization problems the remaining items may be partially selected or not selected at all by the configuration-LP. Thus, the probability of an item to be packed after j iterations may take different values for different items. In contrast, this probability is the same for all items in the case of bin packing. Similarly, while in the case of bin packing all items must be packed by the configuraiton-LP in every iteration, in maximization problem there is a degree of freedom in the selection of items to be packed. This, in turn, led to a different approach for the use of the linear structure.

We note that while the paper [7] deals with a generalization of CMK and uses several similar concepts (configuration LP, sampling, subset oblivious algorithms), the algorithm in [7] does not use an iterative approach. It relies on two separate stages: the first uses a randomized rounding of a configuration-LP that is solved once, and the second stage uses a combinatorial algorithm. We believe the analysis of the iterative randomized rounding algorithm presented in this paper will be useful in showing iterative randomized rounding yields an improved approximation for the Uniform 2-dimensional Vector Multiple Knapsack (2d-UMK) problem considered in [7]. The main challenge in applying our analysis to 2d-UMK is that our analysis relies on a robust linear structure, which is unlikely to exist for 2d-UMK (as that would lead to a PTAS, contradicting the hardness results in [7]). This can potentially be bypassed with the use of an analog of the linear structure that holds for 2d-UMK and adaption of the analysis to this potential structure.

1.4 Organization

In Section 2 we give some definitions and notation. Section 3 presents our main algorithm and an outline of its analysis. In Section 4 we give the detailed analysis (proofs of Lemmas 6, 7, 10 and 11). The proofs of Lemma 9 and Lemma 4 are given in the full version of the paper [9].

2 Preliminaries

We start with some definitions and notation. Let $\text{OPT}(I)$ be the value of an optimal solution for an instance I of a maximization problem Π. For $\alpha \in (0,1]$, a solution x for the instance I is an α-approximate solution if its value is at least $\alpha \cdot \text{OPT}(I)$. For $\alpha \in (0,1]$, we say that $\mathcal{A}$ is an α-approximation algorithm for Π if for any instance I of Π, $\mathcal{A}$ outputs an α-approximate solution for I. An algorithm $\mathcal{A}$ is a *randomized* α-approximation for Π if for any instance I of Π it always returns a solution for I, and the solution is an α-approximate solution with probability at least $\frac{1}{2}$. A *polynomial-time approximation scheme* (PTAS) for a maximization problem Π is a family of algorithms $(\mathcal{A}_\varepsilon)_{\varepsilon>0}$ such that for any $\varepsilon > 0$, $\mathcal{A}_\varepsilon$ is a polynomial-time $(1-\varepsilon)$-approximation algorithm for Π. As the running time of a PTAS may be impractically high, two restrictive classes of PTAS have been proposed in the literature: $(\mathcal{A}_\varepsilon)_{\varepsilon>0}$ is an *efficient* PTAS (EPTAS) if the running time of $\mathcal{A}_\varepsilon$ is of the form $f\left(\frac{1}{\varepsilon}\right) \cdot n^{O(1)}$, where f is an arbitrary function, and n is the bit-length encoding size of the input instance; $(\mathcal{A}_\varepsilon)_{\varepsilon>0}$ is a *fully* PTAS (FPTAS) if the running time of $\mathcal{A}_\varepsilon$ is bounded by $\left(\frac{n}{\varepsilon}\right)^{O(1)}$. Given a boolean expression $\mathcal{D}$, we define $\mathbb{1}_{\mathcal{D}} \in \{0,1\}$ such that $\mathbb{1}_{\mathcal{D}} = 1$ if $\mathcal{D}$ is true and $\mathbb{1}_{\mathcal{D}} = 0$ otherwise.

We give an alternative definition of our problem that will be used in the technical sections. An instance of CMK is a tuple $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$, where I is a set of items, $\mathbf{w} : I \to [0,1]$ is the weight function, $\mathbf{v} : I \to \mathbb{R}_{\geq 0}$ is the value function, $m \in \mathbb{N}_{>0}$ is the number of bins, and $k \in \mathbb{N}_{>0}$ is the cardinality constraint. A *configuration* of the instance $\mathcal{I}$ is $C \subseteq I$ such that $|C| \leq k$ and $\mathbf{w}(C) = \sum_{i \in C} \mathbf{w}(i) \leq 1$. Let $\mathcal{C}_\mathcal{I}$ be the set of all configurations of $\mathcal{I}$, and $\mathcal{C}_\mathcal{I}(i) = \{C \in \mathcal{C} \mid i \in C\}$ the set of all configurations which contain $i \in I$. When clear from the context, we simply use $\mathcal{C} = \mathcal{C}_\mathcal{I}$ and $\mathcal{C}(i) = \mathcal{C}_\mathcal{I}(i)$.

A *solution* of $\mathcal{I}$ is a tuple of m configurations $S = (C_1, \ldots, C_m) \in \mathcal{C}^m$. The value of the solution $S = (C_1, \ldots, C_m)$ is $\mathbf{v}(S) = \mathbf{v}\left(\bigcup_{b \in [m]} C_b\right)$ (generally, for any set $B \subseteq A$ and a function $f : A \to \mathbb{R}_{\geq 0}$, we use $f(B) = \sum_{b \in B} f(b)$). The objective is to find a solution of maximum value. Let $\text{OPT}(\mathcal{I})$ be the optimal solution value for the instance $\mathcal{I}$, and $|\mathcal{I}|$ the encoding size of $\mathcal{I}$. W.l.o.g., we consider a tuple with fewer than m configurations to be a solution. In this case, for some $r \leq m$, the tuple $(C_1, \ldots, C_r) \in \mathcal{C}^r$ is equivalent to the solution $(C_1, \ldots, C_r, \emptyset, \ldots, \emptyset) \in \mathcal{C}^m$.

Our main algorithm, given in Section 3, is applied to a restricted subclass of *simple* instances. We now give a more formal definition for this subclass of instances.

▶ **Definition 3.** *Let $\varepsilon \in (0, 0.1)$, We say that a CMK instance $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$ is* ε-simple *if the following conditions hold.*
- *For every $C \in \mathcal{C}$, we have that $\mathbf{v}(C) \leq \varepsilon^{30} \cdot \mathrm{OPT}(\mathcal{I})$.*
- $m > \exp(\exp(\varepsilon^{-30}))$
- $\varepsilon \cdot m \in \mathbb{N}$.

We give a reduction showing that our algorithm for ε-simple instances yields a randomized EPTAS for general CMK instances.[5] This is formalized in the next lemma (we give the proof in [9]).

▶ **Lemma 4.** *Given $\varepsilon \in (0, 0.1)$ such that $\varepsilon^{-\frac{1}{2}} \in \mathbb{N}$, let $\mathcal{A}$ be a randomized algorithm which returns a $(1 - \varepsilon)$-approximate solution for any ε-simple CMK instance $\mathcal{I}$ in time $\left(\frac{|\mathcal{I}|}{\varepsilon}\right)^{O(1)}$. Then, there is a randomized* EPTAS *for CMK.*

Theorem 2 follows from Theorem 1 and Lemma 4.

3 The Algorithm

In this section, we formally present our iterative randomized rounding algorithm for ε-simple CMK instances. The algorithm relies on a linear programming (LP) relaxation of CMK that we formalize through the notion of fractional solutions.

A *fractional solution* for an instance $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$ is a vector $\bar{x} \in \mathbb{R}^{\mathcal{C}}_{\geq 0}$; the value $\bar{x}_C$ represents a fractional selection of the configuration C for the solution. The *coverage* of $\bar{x}$ is the vector $\texttt{cover}(\bar{x}) \in \mathbb{R}^{I}_{\geq 0}$ defined by

$$\forall i \in I: \quad \texttt{cover}_i(\bar{x}) \;=\; (\texttt{cover}(\bar{x}))_i \;=\; \sum_{C \in \mathcal{C}(i)} \bar{x}_C.$$

The vector $\bar{x}$ is *feasible* if $\texttt{cover}(\bar{x}) \in [0,1]^I$. The size of $\bar{x}$ is $\|\bar{x}\| = \sum_{C \in \mathcal{C}} \bar{x}_C$ (throughout this paper, for every vector $\bar{z} \in \mathbb{R}^n$ we use $\|\bar{z}\| = \sum_{i=1}^{n} |\bar{z}_i|$). The *value* of $\bar{y} \in [0,1]^I$ is $\mathbf{v}(\bar{y}) = \sum_{i \in I} \bar{y}_i \cdot \mathbf{v}(i)$. The *value* of $\bar{x}$ is the value of the cover of $\bar{x}$, that is, $\mathbf{v}(\bar{x}) = \mathbf{v}(\texttt{cover}(\bar{x}))$. For $\ell \in \mathbb{N}_{>0}$, let $[\ell] = \{1, \ldots, \ell\}$.

A solution $S = (C_1, \ldots, C_m)$ for $\mathcal{I}$, where $C_1, \ldots, C_m$ are disjoint and non-empty, can be encoded as a feasible fractional solution $\bar{x} \in \{0,1\}^{\mathcal{C}}$ defined by $\bar{x}_{C_b} = 1$ for every $b \in [m]$, and $\bar{x}_C = 0$ for every other configuration. It is easy to verify that $\|\bar{x}\| = m$, $\texttt{cover}_i(\bar{x}) = 1$ for every $i \in S$, $\texttt{cover}_i(\bar{x}) = 0$ for every $i \in I \setminus S$, and $\mathbf{v}(\bar{x}) = \mathbf{v}(S)$.

We use fractional solutions to define a linear program (LP). Let K be a set and $\bar{\gamma} \in \mathbb{R}^K$. The *support* of $\bar{\gamma}$ is $\mathrm{supp}(\bar{\gamma}) = \{i \in K \mid \bar{\gamma}_i \neq 0\}$. Let $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$ be a CMK instance. For every set $S \subseteq I$ of remaining items and $\ell \in \mathbb{N}$ remaining bins, we define the configuration LP of S and ℓ by

$$\texttt{LP}(S,\ell): \quad \begin{array}{ll} \max & \mathbf{v}(\bar{x}) \\ \text{s.t.} & \bar{x} \text{ is a feasible fractional solution for } \mathcal{I} \\ & \mathrm{supp}(\bar{x}) \subseteq 2^S \\ & \|\bar{x}\| = \ell \end{array}$$

[5] In our discussion of ε-simple instances, we did not attempt to optimize the constants.

That is, in $\texttt{LP}(S,\ell)$ exactly ℓ configurations are selected[6], and these configurations contain only items in S. We can formally define the configuration polytope $P(\ell)$ discussed in Section 1.3 via fractional solutions by

$$P(\ell) = \{\texttt{cover}(\bar{x}) \mid \bar{x} \text{ is a feasible fractional solution for } \mathcal{I} \text{ and } \|\bar{x}\| \leq \ell\}. \tag{4}$$

It can be shown that $\texttt{LP}(S_j, (1-(j-1)\cdot\varepsilon)\cdot m)$ is equivalent to the linear program in (1).

A generalization of $\texttt{LP}(S,\ell)$ for the *separable assignment problem* (SAP) was considered in [12]. Given $p_i \geq 0$ for every $i \in I$, the paper [12] shows that linear programs such as $\texttt{LP}(S,\ell)$ admit an FPTAS whenever the *single bin problem* – of finding $C \in \mathcal{C}$ such that $\sum_{i\in I} p_i$ is maximized – admits an FPTAS. As the single bin case of CMK has an FPTAS (e.g., [4, 18, 10]), we get the following.

▶ **Lemma 5.** *There is an algorithm which given a CMK instance $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$, $S \subseteq I$, $\ell \in \mathbb{N}$ and $\varepsilon > 0$, finds a $(1-\varepsilon)$-approximate solution for $\texttt{LP}(S,\ell)$ in time $\left(\frac{|\mathcal{I}|}{\varepsilon}\right)^{O(1)}$.*

Given a fractional solution $\bar{x}$ such that $\|\bar{x}\| \neq 0$, we say that a random configuration $R \in \mathcal{C}$ is *distributed by* $\bar{x}$, and write $R \sim \bar{x}$, if $\Pr(R = C) = \frac{\bar{x}_C}{\|\bar{x}\|}$ for all $C \in \mathcal{C}$.

The pseudocode of our algorithm for CMK is given in Algorithm 1. In each iteration $1 \leq j \leq \varepsilon^{-1}$, the algorithm uses the solution $\bar{x}^j$ for $\texttt{LP}(S_{j-1}, m_j)$ to sample $\varepsilon \cdot m$ configurations, where S_{j-1} is the set of items remaining after iteration $(j-1)$, and m_j is the number of remaining (unassigned) bins.

Algorithm 1 Iterative Randomized Rounding.

input : Error parameter $\varepsilon \in (0, 0.1)$, $\varepsilon^{-\frac{1}{2}} \in \mathbb{N}$, and an ε-simple CMK instance $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$

output : A solution for the instance

Initialize $S_0 \leftarrow I$

for $j = 1, \ldots, \varepsilon^{-1}$ **do**

  Find a $(1-\varepsilon)$-approximate solution $\bar{x}^j$ for $\texttt{LP}(S_{j-1}, m_j)$, where $m_j = m\,(1-(j-1)\cdot\varepsilon)$.

  Sample independently $q = \varepsilon \cdot m$ configurations $R^j_1, \ldots, R^j_q \sim \bar{x}^j$.

  Update $S_j = S_{j-1} \setminus \left(\bigcup_{b=1}^{q} R^j_b\right)$.

Return as solution $\left(R^j_b\right)_{1\leq j\leq \varepsilon^{-1},\ 1\leq b\leq q}$

Consider the execution of Algorithm 1 with the input $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$ and $\varepsilon \in (0, 0.1)$ such that $\varepsilon^{-\frac{1}{2}} \in \mathbb{N}$. The notations we use below, such as $\bar{x}^j$, S_j, and R^j_b, refer to the variables used throughout the execution of the algorithm. Clearly, Algorithm 1 returns a solution for $\mathcal{I}$. Furthermore, by Lemma 5, the running time of the algorithm is polynomial in $\mathcal{I}$ and ε^{-1}. Let $V = \mathbf{v}\left(\bigcup_{j=1}^{\varepsilon^{-1}} \bigcup_{b=1}^{q} R^j_b\right) = \mathbf{v}\left(I \setminus S_{\varepsilon^{-1}}\right)$ be the value of the returned solution.

Main Lemmas

In the following, we describe the main lemmas we prove in order to lower bound the value of V. The proofs of the lemmas are given in Section 4 and the full version of the paper [9].

[6] Note that $\bar{x}_\emptyset$ may be greater than 1.

A simple calculation shows that the expected value of $\mathbf{v}(R_b^j)$, given all the samples up to (and including) iteration $(j-1)$, is $\frac{\mathbf{v}(\bar{x}^j)}{m\cdot(1-(j-1)\cdot\varepsilon)}$. To compute the expected value of $\mathbf{v}\left(\bigcup_{b=1}^{q} R_b^j\right)$, we need to take into consideration events in which an item $i\in I$ appears in several configurations among $R_1^j,\ldots,R_q^j$. In Section 4.2 we show that, since only a small number of configurations are sampled in each iteration (in comparison to the overall remaining number of bins), such events have small effect on the expected value (with the exception of the last $\varepsilon^{-\frac{1}{2}}$ iterations). This observation is coupled with a concentration bound to prove the next lemma.

▶ **Lemma 6.** *With probability at least* $1-\exp\left(-\varepsilon^{-8}\right)$*, it holds that*

$$V \;=\; \mathbf{v}(I\setminus S_{\varepsilon^{-1}}) \;\geq\; \sum_{j=1}^{\varepsilon^{-1}-\varepsilon^{-\frac{1}{2}}} \mathbf{v}(\bar{x}^j)\cdot\frac{\left(\varepsilon-\varepsilon^{\frac{3}{2}}\right)}{1-(j-1)\varepsilon} - \varepsilon^9\cdot\mathrm{OPT}(\mathcal{I}).$$

Lemma 6 is the formal statement of (2). Lemma 6 essentially reduces the problem of deriving a lower bound for V to obtaining a lower bound on $\mathbf{v}(\bar{x}^j)$.

To obtain a lower bound for $\mathbf{v}(\bar{x}^j)$ we use the following steps. We define random vectors $\bar{\gamma}^j\in[0,1]^I$ for every $j\in[\varepsilon^{-1}]$ such that $\mathbf{v}(\bar{\gamma}^j)$ is high, and there is $\bar{z}^j$ such that $\texttt{cover}(\bar{z}^j)=\bar{\gamma}^{j-1}$ and $\|\bar{z}^j\|\approx m_j$. We scale down $\bar{z}^j$ to obtain a solution for $\texttt{LP}(S_{j-1},m_j)$ of value $\approx\mathbf{v}(\bar{\gamma}^{j-1})$, and consequently get a lower bound for $\mathbf{v}(\bar{x}^j)$. We use a *linear structure* defined below, to show the existence of $\bar{z}^j$. We further use auxiliary random vectors $\bar{\lambda}^j$ to define $\bar{\gamma}^j$.

Let $(\Omega,\mathcal{F},\Pr)$ be the probability space defined by the execution of the algorithm. Define the σ-algebras $\mathcal{F}_0=\{\emptyset,\Omega\}$ and $\mathcal{F}_j=\sigma\left(\{R_b^{j'}\mid 1\leq j'\leq j,\ 1\leq b\leq q\}\right)$. That is, $\mathcal{F}_j$ describes events which only depend on the outcomes of the random sampling up to (and including) the j-th iteration of the algorithm. We follow the standard definition of conditional probabilities and expectations given σ-algebras (see, e.g., [6]).

Fix an optimal solution $(C_1^*,\ldots,C_m^*)$ for the instance and let $S^*=\bigcup_{j=1}^{m}C_j^*$ be the set of items in this solution. Also, given a set $S\subseteq I$ denote by $\mathbb{1}_S$ the vector $\bar{z}\in\{0,1\}^I$ satisfying $\bar{z}_i=1$ for $i\in S$, and $\bar{z}_i=0$ otherwise.

We define $\bar{\gamma}^j$ and $\bar{\lambda}^j$ inductively using S^*. Define $\bar{\gamma}^0=\mathbb{1}_{S^*}$, that is $\bar{\gamma}_i^0=1$ for every $i\in S^*$ and $\bar{\gamma}_i^0=0$ for every $i\in I\setminus S^*$. For every $j\in[\varepsilon^{-1}-1]$ define $\bar{\lambda}^j\in\mathbb{R}_{\geq 0}^I$ by

$$\bar{\lambda}_i^j = \frac{1-j\cdot\varepsilon}{1-(j-1)\varepsilon}\cdot\frac{1}{\Pr(i\in S_j\mid\mathcal{F}_{j-1})}\cdot\bar{\gamma}_i^{j-1} \tag{5}$$

for all $i\in S_{j-1}$ and $\bar{\lambda}_i^j=0$ for $i\notin S_{j-1}$. Intuitively, the expression $\Pr(i\in S_j\mid\mathcal{F}_{j-1})$ in (5) is the probability that item i will still be available for packing after the j-th iteration, where the probability is calculated at the end of the iteration $j-1$. Also, for every $j\in[\varepsilon^{-1}-1]$ define $\bar{\gamma}^j\in\mathbb{R}_{\geq 0}^I$ by

$$\bar{\gamma}_i^j = \mathbb{1}_{i\in S_j}\cdot\bar{\lambda}_i^j \qquad \forall i\in I. \tag{6}$$

Observe that $\bar{\lambda}^j$ is $\mathcal{F}_{j-1}$-measurable random variable whereas $\bar{\gamma}^j$ is $\mathcal{F}_j$-measurable. Intuitively, this means that the value of $\bar{\lambda}^j$ is known by the end of the $(j-1)$-th iteration, while the value of $\bar{\gamma}^j$ is only known by the end of the j-th iteration.

The lower bound on $\mathbf{v}(\bar{\gamma}^{j-1})$ relies on a simple calculation of expectations followed by a concentration bound. By induction it can be shown that $\mathbb{E}[\bar{\gamma}_i^{j-1}]=(1-(j-1)\varepsilon)\cdot\mathbb{1}_{i\in S^*}$, and therefore,

$$\mathbb{E}[\mathbf{v}(\bar{\gamma}^{j-1})]=(1-(j-1)\varepsilon)\cdot\mathbf{v}(S^*)=(1-(j-1)\varepsilon)\cdot\mathrm{OPT}(\mathcal{I}).$$

We use concentration bounds to show that indeed $\mathbf{v}(\bar{\gamma}^{j-1})$ does not deviate from its expected value.

▶ **Lemma 7.** *With probability at least* $1 - \exp(-\varepsilon^{-20})$*, it holds that*

$$\forall j \in [\varepsilon^{-1}]: \quad \mathbf{v}(\bar{\gamma}^{j-1}) \geq (1 - \varepsilon(j-1)) \cdot \mathrm{OPT}(\mathcal{I}) - \varepsilon^3 \cdot \mathrm{OPT}(\mathcal{I}).$$

We give the proof of Lemma 7 in the full version of the paper [9].

Our next challenge is to show that there is a solution for $\texttt{LP}(S_{j-1}, m_j)$ whose cover is roughly $\bar{\gamma}^{j-1}$, which can be alternatively stated as $\bar{\gamma}^{j-1} \in P(\ell)$ where $\ell \approx m_j$, and $P(\ell)$ is as defined in (4). To this end, we introduce a *linear structure* for CMK. The main idea in linear structures is that they allow us to determine that $\bar{\gamma}^j \in P(\ell)$ by checking if $\bar{\gamma}^j$ satisfies a small number of linear inequalities.

Given a vector $\bar{u} \in \mathbb{R}^I_{\geq 0}$ which defines an inequality in the linear structure, we use concentration bounds to show that $\bar{\gamma}^j \cdot \bar{u} \leq \mathbb{E}[\bar{\gamma}^j \cdot \bar{u}] + \xi$, where ξ is an error terms. The concentration bounds we use only provide useful guarantees if the error term ξ is of order of the maximum sum of entries in $\bar{u}$ w.r.t. a single configuration, that is, $\texttt{tol}(\bar{u}) = \max\left\{\sum_{i \in C} \bar{u}_i \,\middle|\, C \in \mathcal{C}\right\}$. We refer to the value $\texttt{tol}(\bar{u})$ as the *tolerance* of $\bar{u}$. We consequently require the linear structure to be robust to additive errors of order of the tolerance. Also, we say that $S \subseteq I$ *can be packed into* $\ell \in \mathbb{N}$ *bins* if there are ℓ configurations $C_1, \ldots, C_\ell \in \mathcal{C}$ such that $\bigcup_{b=1}^{\ell} C_b = S$.

▶ **Definition 8** (Linear Structure)**.** *Let* $(I, \mathbf{w}, \mathbf{v}, m, k)$ *be a CMK instance and* $\delta > 0$ *a parameter. Also, consider a subset* $S \subseteq I$ *such that* S *can be packed in* $\ell \in \mathbb{N}$ *bins. A* δ-linear structure *of* S *is a set of vectors* $\mathcal{L} \subseteq \mathbb{R}^I_{\geq 0}$ *which satisfy the following property.*

- *Let* $\bar{y} \in ([0,1] \cap \mathbb{Q})^I$*,* $0 < \alpha < 1$ *and* $t > 0$*, such that*
  1. $\mathrm{supp}(\bar{y}) \subseteq S$
  2. $\forall \bar{u} \in \mathcal{L}: \quad \bar{u} \cdot \bar{y} \leq \alpha \cdot \bar{u} \cdot \mathbb{1}_S + t \cdot \texttt{tol}(\bar{u})$

  Then, there is a fractional solution $\bar{x}$ *whose cover is* $\bar{y}$ *and* $\|\bar{x}\| \leq \alpha \cdot \ell + 20\delta\ell + (t+1) \cdot \exp(\delta^{-5})$*.*

The size *of the structure* $\mathcal{L}$ *is* $|\mathcal{L}|$*.*

Alternatively, a δ-linear structure guarantees for S that for every $\bar{y} \in [0,1]^I$ with rational entries, $0 < \alpha < 1$ and $t > 0$, if $\mathrm{supp}(\bar{y}) \subseteq S$ and $\bar{y}$ satisfies $|\mathcal{L}|$ linear inequalities, then $\bar{y} \in P(\alpha \cdot \ell + 20\delta\ell + (t+1) \cdot \exp(\delta^{-5}))$.

In [9] we prove the next result.

▶ **Lemma 9.** *Given* $\delta > 0$*, let* $I = (I, \mathbf{w}, \mathbf{v}, m, k)$ *be a CMK instance, and* $S \subseteq I$ *a subset which can be packed into* $\ell > \exp(\delta^{-5})$ *bins. Then there is a* δ*-linear structure* $\mathcal{L}$ *of* S *of size at most* $\exp\left(\delta^{-4}\right)$*.*

The above lemma is an adaptation of a construction of [17] used to solve the vector bin packing problem, in which there are additional requirements for the packing of S. Our adaptation leverages the relative simplicity of a cardinality constraint to omit these additional requirements.

We use Lemma 9 to show the existence of an ε^2-linear structure of S^*, where S^* is the set of items in an optimal solution. We use the linear structure to show the existence of a fractional solution $\bar{z}^j$ such that $\texttt{cover}(\bar{z}^j) = \bar{\gamma}^{j-1}$ and $\|\bar{z}^j\| \approx (1 - (j-1)\varepsilon)m$ for every $j \in [\varepsilon^1]$. A simple scaling is then used to construct a solution for $\texttt{LP}(S_{j-1}, m_j)$ and establish the following lower bound on $\mathbf{v}(\bar{x}^j)$.

▶ **Lemma 10.** *With probability at least* $1 - \exp(-\varepsilon^{-20})$*, it holds that*

$$\forall j \in [\varepsilon^{-1}]: \quad \mathbf{v}(\bar{x}^j) \geq (1 - \varepsilon) \cdot \left(1 - \frac{30 \cdot \varepsilon^2}{1 - (j-1)\varepsilon}\right) \cdot \mathbf{v}(\bar{\gamma}^{j-1}).$$

We give the proof of Lemma 10 in [9]. Together, Lemma 10 and Lemma 7 essentially give the formal proof of (3). Finally, using Lemmas 6, 7, and 10, we obtain the next result, whose proof is given in [9].

▶ **Lemma 11.** *With probability at least* $1-\exp(-\varepsilon^{-5})$*, it holds that* $V \geq (1-60\sqrt{\varepsilon})\cdot \mathrm{OPT}(\mathcal{I})$.

Theorem 1 follows directly from Lemma 11.

4 The Analysis

Consider an execution of Algorithm 1 with the input $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$ and $\varepsilon > 0$. We use the notation and definitions as given in Section 3. Also, let $\bar{y}^j = \texttt{cover}(\bar{x}^j)$ be the coverage of $\bar{x}^j$. Observe $\bar{x}^j$ and $\bar{y}^j$ are $\mathcal{F}_{j-1}$-measurable. That is, their values are determined by the outcomes of the samples up to (and including) the $j-1$ iteration. As in Section 3 we let $(C^*_1, \ldots, C^*_m)$ be an optimal solution for the instance $\mathcal{I}$. We define $S^* = \bigcup_{b=1}^{m} C^*_b$ and $\mathrm{OPT} = \mathbf{v}(S^*) = \mathrm{OPT}(\mathcal{I})$.

4.1 Concentration Bounds

Before we give the proofs of Lemmas 6, 7, 10, and 11, we need to introduce some concentration bounds for *self-bounding functions*.

▶ **Definition 12.** *A non-negative function* $f : \mathcal{X}^n \to \mathbb{R}_{\geq 0}$ *is called self-bounding if there exist* n *functions* $f_1, \ldots, f_n : \mathcal{X}^{n-1} \to \mathcal{R}$ *such that for all* $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$,

$$0 \leq f(x) - f_t(x^{(t)}) \leq 1, \qquad \text{and}$$

$$\sum_{t=1}^{n} \left(f(x) - f_t(x^{(t)}) \right) \leq f(x),$$

where $x^{(t)} = (x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_n) \in \mathcal{X}^{n-1}$ *is obtained by dropping the* t*-th component of* x.

We rely on the following concentration bound due to Boucheron, Lugosi and Massart [3].

▶ **Lemma 13.** *Let* $f : \mathcal{X}^n \to \mathbb{R}_{\geq 0}$ *be a self-bounding function and let* $X_1, \ldots, X_n \in \mathcal{X}$ *be independent random variables. Define* $Z = f(X_1, \ldots, X_n)$*. Then the following holds:*

1. $\Pr\left(Z \geq \mathbb{E}[Z] + t\right) \leq \exp\left(-\frac{t^2}{2\cdot\mathbb{E}[Z] + \frac{t}{3}}\right)$, *for every* $t \geq 0$.
2. $\Pr\left(Z \leq \mathbb{E}[Z] - t\right) \leq \exp\left(-\frac{t^2}{2\cdot\mathbb{E}[Z]}\right)$, *for every* $t > 0$.

The setting considered in [3] can be trivially extended to a setting in which the random variable are conditionally independent on a σ-algebra $\mathcal{G}$ (see [6] for the definition of conditional independence) and the function f itself is a $\mathcal{G}$-measurable random function. This is formally stated in the next lemma.

▶ **Lemma 14.** *Let* $(\Omega, \mathcal{F}, \Pr)$ *be a finite probability space and let* $\mathcal{G} \subseteq \mathcal{F}$ *be a* σ*-algebra. Let* D *be a finite set of self-bounding function from* χ^ℓ *to* $\mathbb{R}_{\geq 0}$ *and let* $f \in D$ *be a* $\mathcal{G}$*-measurable random function. Also, let* $X_1, \ldots, X_\ell \in \chi$ *be random variables which are conditionally independent given* $\mathcal{G}$*. Define* $Z = f(X_1, \ldots, X_n)$*. Then the following holds:*

1. $\Pr\left(Z \geq \mathbb{E}[Z \mid \mathcal{G}] + t \mid \mathcal{G}\right) \leq \exp\left(-\frac{t^2}{2\cdot\mathbb{E}[Z \mid \mathcal{G}] + \frac{t}{3}}\right)$, *for every* $t \geq 0$.
2. $\Pr\left(Z \leq \mathbb{E}[Z \mid \mathcal{G}] - t \mid \mathcal{G}\right) \leq \exp\left(-\frac{t^2}{2\cdot\mathbb{E}[Z \mid \mathcal{G}]}\right)$ *for every* $t \geq 0$.

The generalization in Lemma 14 is required since the variables $R^j_1, \ldots, R^j_q$ are dependent for $q > 1$ while being conditionally independent given the variables $R^{j'}_b$ for every $j' < j$ and $b \in [q]$. The following construction for self-bounding function was shown in [7].

▶ **Lemma 15.** *Let $\mathcal{I} = (I, \mathbf{w}, \mathbf{v}, m, k)$ be a CMK instance, and $h : I \to \mathbb{R}_{\geq 0}$. For some $\ell \in \mathbb{N}_{>0}$ define $f : \mathcal{C}^\ell \to \mathbb{R}_{\geq 0}$ by $f(C_1, \ldots, C_\ell) = \frac{h(\bigcup_{i \in [\ell]} C_i)}{\eta}$ where $\eta \geq \max_{C \in \mathcal{C}} h(C)$. Then f is self-bounding.*

4.2 The proof of Lemma 6

The first step towards the proof of Lemma 6 is to show a lower bound on the probability of an item to appear in one of the sampled configurations $R_1^j, \ldots, R_q^j$ in terms of $\bar{y}_i^j$.

▶ **Lemma 16.** *For every $i \in I$ and $j \in \left[\varepsilon^{-1}\right]$ it holds that $\Pr\left(i \in S_{j-1} \setminus S_j \mid \mathcal{F}_{j-1}\right) \geq 1 - \exp\left(-\varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon}\right)$.*

Proof. By a simple calculation,

$$\begin{aligned}
\Pr\left(i \in S_{j-1} \setminus S_j \mid \mathcal{F}_{j-1}\right) &= \Pr\left(i \in \bigcup_{b=1}^{q} R_b^j \,\middle|\, \mathcal{F}_{j-1}\right) \\
&= 1 - \Pr\left(i \notin \bigcup_{b=1}^{q} R_b^j \,\middle|\, \mathcal{F}_{j-1}\right) \\
&= 1 - \prod_{b=1}^{q} \Pr\left(i \notin R_b^j \,\middle|\, \mathcal{F}_{j-1}\right) \qquad (7)\\
&= 1 - \prod_{b=1}^{q} \left(1 - \Pr\left(i \in R_b^j \,\middle|\, \mathcal{F}_{j-1}\right)\right) \\
&= 1 - \prod_{b=1}^{q} \left(1 - \Pr\left(R_b^j \in \mathcal{C}(i) \,\middle|\, \mathcal{F}_{j-1}\right)\right).
\end{aligned}$$

The third equality holds as $R_1^j, \ldots, R_q^j$ are conditionally independent given $\mathcal{F}_{j-1}$. Therefore, by (7) and since the configurations are distributed by $\bar{x}^j$ we have

$$\begin{aligned}
\Pr\left(i \in S_{j-1} \setminus S_j \mid \mathcal{F}_{j-1}\right) &= 1 - \prod_{b=1}^{q} \left(1 - \Pr\left(R_b^j \in \mathcal{C}(i) \,\middle|\, \mathcal{F}_{j-1}\right)\right) \\
&= 1 - \left(1 - \frac{\sum_{C \in \mathcal{C}(i)} \bar{x}_C^j}{\|\bar{x}^j\|}\right)^q \\
&= 1 - \left(1 - \frac{\bar{y}_i^j}{m \cdot (1 - (j-1) \cdot \varepsilon)}\right)^{\varepsilon \cdot m} \\
&= 1 - \left(\left(1 - \frac{\bar{y}_i^j}{m \cdot (1 - (j-1) \cdot \varepsilon)}\right)^{\frac{m \cdot (1-(j-1)\cdot\varepsilon)}{\bar{y}_i^j}}\right)^{\frac{\varepsilon \cdot \bar{y}_i^j}{(1-(j-1)\cdot\varepsilon)}} \\
&\geq 1 - \left(e^{-1}\right)^{\frac{\varepsilon \cdot \bar{y}_i^j}{(1-(j-1)\cdot\varepsilon)}} \\
&= 1 - \exp\left(-\frac{\varepsilon \cdot \bar{y}_i^j}{(1 - (j-1) \cdot \varepsilon)}\right).
\end{aligned}$$

The inequality holds since $(1 - \frac{1}{x})^x \leq \frac{1}{e}$ for all $x \geq 1$. ◀

The next lemma uses Lemma 16 to lower bound the total value of sampled configurations in the j-th iteration.

▶ **Lemma 17.** *For all $j \in \left[\varepsilon^{-1} - \varepsilon^{-\frac{1}{2}}\right]$ it holds that*

$$\mathbb{E}\left[\mathbf{v}(S_{j-1} \setminus S_j) \mid \mathcal{F}_{j-1}\right] \geq \mathbf{v}(\bar{x}^j) \cdot \left(\varepsilon - \varepsilon^{\frac{3}{2}}\right) \frac{1}{1-(j-1)\varepsilon}.$$

Proof. By Lemma 16 we get

$$\begin{aligned}
\mathbb{E}\left[\mathbf{v}(S_{j-1} \setminus S_j) \mid \mathcal{F}_{j-1}\right] &= \sum_{i \in I} \mathbf{v}(i) \cdot \Pr\left(i \in S_{j-1} \setminus S_j \mid \mathcal{F}_{j-1}\right) \\
&\geq \sum_{i \in I} \mathbf{v}(i) \cdot \left(1 - \exp\left(-\varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon}\right)\right) \\
&\geq \sum_{i \in I} \mathbf{v}(i) \cdot \left(\varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon} - \left(\varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon}\right)^2\right) \qquad (8) \\
&= \sum_{i \in I} \mathbf{v}(i) \cdot \left(\varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon} \cdot \left(1 - \varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon}\right)\right).
\end{aligned}$$

The second inequality follows from $1 - \exp(-x) \geq x - x^2$ for all $x \geq 0$. By (8) we have

$$\begin{aligned}
\mathbb{E}\left[\mathbf{v}(S_{j-1} \setminus S_j) \mid \mathcal{F}_{j-1}\right] &\geq \sum_{i \in I} \mathbf{v}(i) \cdot \left(\varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon} \cdot \left(1 - \varepsilon \cdot \frac{1}{1-(\varepsilon^{-1} - \varepsilon^{-\frac{1}{2}} - 1)\varepsilon}\right)\right) \\
&= \sum_{i \in I} \mathbf{v}(i) \cdot \left(\varepsilon \cdot \frac{\bar{y}_i^j}{1-(j-1)\varepsilon} \cdot \left(1 - \frac{\varepsilon}{\varepsilon + \varepsilon^{\frac{1}{2}}}\right)\right) \\
&= \frac{1}{1-(j-1)\varepsilon} \cdot \left(\varepsilon - \frac{\varepsilon^2}{\varepsilon + \varepsilon^{\frac{1}{2}}}\right) \cdot \sum_{i \in I} \mathbf{v}(i) \cdot \bar{y}_i^j \\
&= \frac{1}{1-(j-1)\varepsilon} \cdot \left(\varepsilon - \frac{1}{\varepsilon^{-1} + \varepsilon^{-\frac{3}{2}}}\right) \cdot \mathbf{v}(\bar{x}^j) \\
&\geq \mathbf{v}(\bar{x}^j) \cdot \left(\varepsilon - \varepsilon^{\frac{3}{2}}\right) \frac{1}{1-(j-1)\varepsilon}.
\end{aligned}$$

The first inequality holds since $j \leq \varepsilon^{-1} - \varepsilon^{-\frac{1}{2}}$ and since $\bar{x}^j$ is a feasible solution for $\texttt{LP}(S, m_j)$; thus, $\bar{y}^j \in [0,1]^I$. ◀

We can also use Lemma 14 to show that the value of the configurations sampled in the j-th iteration does not deviate significantly from its expected value.

▶ **Lemma 18.** *For all $j \in [\varepsilon^{-1}]$ it holds that*

$$\Pr\left(\mathbf{v}(S_{j-1} \setminus S_j) \leq \mathbb{E}\left[\mathbf{v}\left(S_{j-1} \setminus S_j\right) \mid \mathcal{F}_{j-1}\right] - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I})\right) \leq \exp\left(-\varepsilon^{-9}\right).$$

Proof. Recall that $q = \varepsilon \cdot m$. Define a function $f : \mathcal{C}^q \to \mathbb{R}_{\geq 0}$ by $f(X) = \frac{\mathbf{v}(S)}{\varepsilon^{30} \cdot \mathrm{OPT}(\mathcal{I})}$ for all $X = (C_1, \ldots, C_q) \in \mathcal{C}^q$. Since $\mathcal{I}$ is ε-simple it holds that $\mathbf{v}(C) \leq \varepsilon^{30} \cdot \mathrm{OPT}$ for all $C \in \mathcal{C}$, thus, by Lemma 15 it follows that f is a self-bounding function. Therefore,

$$\begin{aligned}
&\Pr\left(\mathbf{v}(S_{j-1} \setminus S_j) \leq \mathbb{E}\left[\mathbf{v}\left(S_{j-1} \setminus S_j\right) \;\middle|\; \mathcal{F}_{j-1}\right] - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I}) \;\middle|\; \mathcal{F}_{j-1}\right) \\
=\ &\Pr\left(f\left(R_1^j, \ldots, R_q^j\right) \leq \mathbb{E}\left[f\left(R_1^j, \ldots, R_q^j\right) \;\middle|\; \mathcal{F}_{j-1}\right] - \varepsilon^{-20} \;\middle|\; \mathcal{F}_{j-1}\right) \\
\leq\ &\exp\left(-\frac{\varepsilon^{-40}}{2 \cdot \mathbb{E}\left[f\left(R_1^j, \ldots, R_q^j\right) \;\middle|\; \mathcal{F}_{j-1}\right]}\right).
\end{aligned} \tag{9}$$

The inequality holds by Lemma 14. In addition, since $R_1^j, \ldots, R_q^j$ is a solution for $\mathcal{I}$, it also holds that

$$\mathbb{E}\left[f\left(R_1^j, \ldots, R_q^j\right) \;\middle|\; \mathcal{F}_{j-1}\right] \leq \frac{\mathrm{OPT}(\mathcal{I})}{\varepsilon^{30} \cdot \mathrm{OPT}(\mathcal{I})} = \varepsilon^{-30}. \tag{10}$$

Hence, by the above

$$\begin{aligned}
&\Pr\left(\mathbf{v}(S_{j-1} \setminus S_j) \leq \mathbb{E}\left[\mathbf{v}\left(S_{j-1} \setminus S_j\right) \;\middle|\; \mathcal{F}_{j-1}\right] - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I}) \;\middle|\; \mathcal{F}_{j-1}\right) \\
\leq\ &\exp\left(-\frac{\varepsilon^{-40}}{2 \cdot \mathbb{E}\left[f\left(R_1^j, \ldots, R_q^j\right) \;\middle|\; \mathcal{F}_{j-1}\right]}\right) \\
\leq\ &\exp\left(-\frac{\varepsilon^{-40}}{2 \cdot \varepsilon^{-30}}\right) \\
=\ &\exp\left(-\frac{\varepsilon^{-10}}{2}\right) \\
\leq\ &\exp\left(-\varepsilon^{-9}\right).
\end{aligned} \tag{11}$$

The first inequality holds by (9). The second inequality follows from (10). For the last inequality, recall that $\varepsilon < 0.1$. Therefore, by (11) it holds that (unconditionally on $\mathcal{F}_{j-1}$),

$$\Pr\left(\mathbf{v}(S_{j-1} \setminus S_j) \leq \mathbb{E}\left[\mathbf{v}\left(S_{j-1} \setminus S_j\right) \;\middle|\; \mathcal{F}_{j-1}\right] - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I})\right) \leq \exp\left(-\varepsilon^{-9}\right). \qquad \blacktriangleleft$$

The proof of Lemma 6 follows from Lemma 17 and Lemma 18.

▶ **Lemma 6.** *With probability at least* $1 - \exp\left(-\varepsilon^{-8}\right)$*, it holds that*

$$V \;=\; \mathbf{v}(I \setminus S_{\varepsilon^{-1}}) \;\geq\; \sum_{j=1}^{\varepsilon^{-1} - \varepsilon^{-\frac{1}{2}}} \mathbf{v}(\bar{x}^j) \cdot \frac{\left(\varepsilon - \varepsilon^{\frac{3}{2}}\right)}{1 - (j-1)\varepsilon} - \varepsilon^9 \cdot \mathrm{OPT}(\mathcal{I}).$$

Proof.

$$\begin{aligned}
&\Pr\left(\mathbf{v}(I \setminus S_{\varepsilon^{-1}}) \geq \sum_{j=1}^{\varepsilon^{-1}-\varepsilon^{-\frac{1}{2}}} \mathbf{v}(\bar{x}^j) \cdot \frac{\left(\varepsilon - \varepsilon^{\frac{3}{2}}\right)}{1-(j-1)\varepsilon} - \varepsilon^9 \cdot \mathrm{OPT}(\mathcal{I})\right)\\
\geq\ &\Pr\left(\bigwedge_{j \in \left[\varepsilon^{-1}-\varepsilon^{-\frac{1}{2}}\right]} \left(\mathbf{v}(S_{j-1} \setminus S_j) \geq \mathbf{v}(\bar{x}^j) \cdot \frac{\left(\varepsilon - \varepsilon^{\frac{3}{2}}\right)}{1-(j-1)\varepsilon} - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I})\right)\right)\\
\geq\ &\Pr\left(\bigwedge_{j \in \left[\varepsilon^{-1}-\varepsilon^{-\frac{1}{2}}\right]} \left(\mathbf{v}(S_{j-1} \setminus S_j) \geq \mathbb{E}\left[\mathbf{v}\left(S_{j-1} \setminus S_j\right) \mid \mathcal{F}_{j-1}\right] - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I})\right)\right).\\
&\geq 1 - \Pr\left(\bigvee_{j \in \left[\varepsilon^{-1}-\varepsilon^{-\frac{1}{2}}\right]} \left(\mathbf{v}(S_{j-1} \setminus S_j) < \mathbb{E}\left[\mathbf{v}\left(S_{j-1} \setminus S_j\right) \mid \mathcal{F}_{j-1}\right] - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I})\right)\right).
\end{aligned} \tag{12}$$

The first inequality holds because if all $\varepsilon^{-1} - \varepsilon^{-\frac{1}{2}}$ events in the second expression occur, then so is the event in the first expression. The second inequality holds by Lemma 17. By (12) and the union bound

$$\begin{aligned}
&\Pr\left(\mathbf{v}(I \setminus S_{\varepsilon^{-1}}) \geq \sum_{j=1}^{\varepsilon^{-1}-\varepsilon^{-\frac{1}{2}}} \mathbf{v}(\bar{x}^j) \cdot \frac{\left(\varepsilon - \varepsilon^{\frac{3}{2}}\right)}{1-(j-1)\varepsilon} - \varepsilon^9 \cdot \mathrm{OPT}(\mathcal{I})\right)\\
\geq\ & 1 - \sum_{j \in \left[\varepsilon^{-1}-\varepsilon^{-\frac{1}{2}}\right]} \Pr\left(\mathbf{v}(S_{j-1} \setminus S_j) \geq \mathbb{E}\left[\mathbf{v}\left(S_{j-1} \setminus S_j\right) \mid \mathcal{F}_{j-1}\right] - \varepsilon^{10} \cdot \mathrm{OPT}(\mathcal{I})\right)\\
\geq\ & 1 - \varepsilon^{-1} \cdot \exp\left(-\varepsilon^{-9}\right)\\
\geq\ & 1 - \exp(-\varepsilon^{-8}).
\end{aligned}$$

The second inequality holds Lemma 18. For the last inequality, recall that $\varepsilon < 0.1$. ◀

The remaining proofs are given in the full version of the paper [9].

References

1 Nikhil Bansal, Marek Eliás, and Arindam Khan. Improved approximation for vector bin packing. In *Proc. SODA*, pages 1561–1579. SIAM, 2016.

2 Nikhil Bansal and Arindam Khan. Improved approximation algorithm for two-dimensional bin packing. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on discrete algorithms*, pages 13–25. SIAM, 2014.

3 Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. A sharp concentration inequality with applications. *Random Structures & Algorithms*, 16(3):277–292, 2000.

4 Alberto Caprara, Hans Kellerer, Ulrich Pferschy, and David Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *Eur. J. Oper. Res.*, 123(2):333–345, 2000.

5 Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, 35(3):713–728, 2005.

6 Yuan Shih Chow and Henry Teicher. *Probability theory: independence, interchangeability, martingales*. Springer Science & Business Media, 1997.

7 Tomer Cohen, Ariel Kulik, and Hadas Shachnai. Improved approximation for two-dimensional vector multiple knapsack. In *34th International Symposium on Algorithms and Computation, ISAAC*, pages 20:1–20:17, 2023.

8 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An afptas for bin packing with partition matroid via a new method for lp rounding. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2023.

9 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An EPTAS for cardinality constrained multiple knapsack via iterative randomized rounding. *arXiv preprint*, 2023. `arXiv:2308.12622`.

10 Ilan Doron-Arad, Ariel Kulik, and Hadas Shachnai. An fptas for budgeted laminar matroid independent set. *Operations Research Letters*, 51(6):632–637, 2023.

11 Leah Epstein and Asaf Levin. Afptas results for common variants of bin packing: A new method for handling the small items. *SIAM Journal on Optimization*, 20(6):3121–3145, 2010.

12 Lisa Fleischer, Michel X Goemans, Vahab S Mirrokni, and Maxim Sviridenko. Tight approximation algorithms for maximum separable assignment problems. *Math. Oper. Res.*, 36(3):416–431, 2011.

13 Klaus Jansen. Parameterized approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, 39(4):1392–1412, 2010.

14 Klaus Jansen. A fast approximation scheme for the multiple knapsack problem. In *Proc. SOFSEM*, pages 313–324, 2012.

15 Narendra Karmarkar and Richard M Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 312–320. IEEE, 1982.

16 Hans Kellerer. A polynomial time approximation scheme for the multiple knapsack problem. In *RANDOM-APPROX*, pages 51–62. Springer, 1999.

17 Ariel Kulik, Matthias Mnich, and Hadas Shachnai. Improved approximations for vector bin packing via iterative randomized rounding. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1366–1376, 2023.

18 Wenxin Li, Joohyun Lee, and Ness Shroff. A faster fptas for knapsack problem with cardinality constraint. *Discrete Applied Mathematics*, 315:71–85, 2022.