# Modular and Submodular Optimization with Multiple Knapsack Constraints via Fractional Grouping

**Yaron Fairstein** ✉
Computer Science Department, Technion, Haifa, Israel

**Ariel Kulik** ✉
Computer Science Department, Technion, Haifa, Israel

**Hadas Shachnai** ✉
Computer Science Department, Technion, Haifa, Israel

**Abstract**

A multiple knapsack constraint over a set of items is defined by a set of bins of arbitrary capacities, and a weight for each of the items. An assignment for the constraint is an allocation of subsets of items to the bins which adheres to bin capacities. In this paper we present a unified algorithm that yields efficient approximations for a wide class of submodular and modular optimization problems involving multiple knapsack constraints. One notable example is a *polynomial time approximation scheme* for Multiple-Choice Multiple Knapsack, improving upon the best known ratio of 2. Another example is Non-monotone Submodular Multiple Knapsack, for which we obtain a $(0.385 - \varepsilon)$-approximation, matching the best known ratio for a single knapsack constraint. The robustness of our algorithm is achieved by applying a novel *fractional* variant of the classical linear grouping technique, which is of independent interest.

**2012 ACM Subject Classification** Theory of computation → Packing and covering problems

**Keywords and phrases** Sumodular Optimization, Multiple Knapsack, Randomized Rounding, Linear Grouping, Multiple Choice Multiple Knapsack

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2021.41

**Related Version** *Full Version*: https://arxiv.org/abs/2007.10470

## 1 Introduction

The Knapsack problem is one of the most studied problems in mathematical programming and combinatorial optimization, with applications ranging from power management and production planning, to blockchain storage allocation and key generation in cryptosystems [31, 26, 38, 41]. In a more general form, knapsack problems require assigning items of various sizes (weights) to a set of bins (knapsacks) of bounded capacities. The bin capacities then constitute the hard constraint for the problem. Formally, a *multiple knapsack constraint* (MKC) over a set of items is defined by a collection of bins of varying capacities and a non-negative weight for each item. A feasible solution for the constraint is an assignment of subsets of items to the bins, such that the total weight of items assigned to each bin does not exceed its capacity. This constraint plays a central role in the classic Multiple Knapsack problem [8, 23, 24]. The input is an MKC and each item also has a profit. The objective is to find a feasible solution for the MKC such that the total profit of assigned items is maximized.

Multiple Knapsack can be viewed as a maximization variant of the Bin Packing problem [25, 13]. In Bin Packing we are given a set of items, each associated with non-negative weight. We need to pack the items into a minimum number of identical (unit-size) bins.


29th Annual European Symposium on Algorithms (ESA 2021).
Editors: Petra Mutzel, Rasmus Pagh, and Grzegorz Herman; Article No. 41; pp. 41:1–41:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A prominent technique for approximating Bin Packing is *grouping*, which decreases the number of distinct weights in the input instance. Informally, a subset of items is partitioned into groups $G_1, \ldots, G_\tau$, and all the items within a group are treated as if they have the *same* weight (e.g., [13, 25]). By properly forming the groups, the increase in the number of bins required for packing the instance can be bounded. Classic grouping techniques require knowledge of the items to be packed, and thus cannot be easily applied in the context of maximization problems, and specifically for a multiple knapsack constraint.

The main technical contribution of this paper is the introduction of *fractional grouping*, a variant of linear grouping which can be applied to multiple knapsack constraints. Fractional Grouping partitions the items into groups using an easy to obtain fractional solution, bypassing the requirement to know the items in the solution.

Fractional Grouping proved to be a robust technique for maximization problems. We use the technique to obtain, among others, a *polynomial-time approximation scheme (PTAS)* for the Multiple-Choice Multiple Knapsack Problem, a $(0.385 - \varepsilon)$-approximation for non-monotone submodular maximization with a multiple knapsack constraint, and a $(1 - e^{-1} - o(1))$-approximation for the Monotone Submodular Multiple Knapsack Problem with Uniform Capacities.

## 1.1 Problem Definition

We first define formally key components of the problem studied in this paper.

A *multiple knapsack constraint* (MKC) over a set $I$ of items, denoted by $\mathcal{K} = (w, B, W)$, is defined by a weight function $w : I \to \mathbb{R}_{\geq 0}$, a set of bins $B$ and bin capacities given by $W : B \to \mathbb{R}_{\geq 0}$. An *assignment* for the constraint is a function $A : B \to 2^I$ which assigns a subset of items to each bin. An assignment $A$ is *feasible* if $\sum_{i \in A(b)} w(i) \leq W(b)$ for all $b \in B$. We say that $A$ is an *assignment of* $S$ if $S = \bigcup_{b \in B} A(b)$.

A set function $f : 2^I \to \mathbb{R}$ is *submodular* if for any $S \subseteq T \subseteq I$ and $i \in I \setminus T$ it holds that $f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T)$.[1] Submodular functions naturally arise in numerous settings. While many submodular functions, such as coverage [19] and matroid rank function [6], are monotone, i.e., for any $S \subseteq T \subseteq I$, $f(S) \leq f(T)$, this is not always the case (cut functions [18] are a classic example). A special case of submodular functions is *modular* (or, *linear*) functions in which, for any $S \subseteq T \subseteq I$ and $i \in I \setminus T$, we have $f(S \cup \{i\}) - f(S) = f(T \cup \{i\}) - f(T)$.

For a constant $d \in \mathbb{N}$, the problem of *Submodular Maximization with d-Multiple Knapsack Constraints (d-MKCP)* is defined as follows. The input is $\mathcal{T} = \left(I, (\mathcal{K}_t)_{t=1}^{d}, \mathcal{I}, f\right)$, where $I$ is a set of items, $\mathcal{K}_t$, $1 \leq t \leq d$ are $d$ MKCs over $I$, $\mathcal{I} \subseteq 2^I$ and $f : 2^I \to \mathbb{R}_{\geq 0}$ is a non-negative submodular function. $\mathcal{I}$ is an additional constraint which can be one of the following: $(i)$ $\mathcal{I} = 2^I$, i.e., any subset of items can be selected. $(ii)$ $\mathcal{I}$ is the independent set of a matroid,[2] or $(iii)$ $\mathcal{I}$ is the intersection of independent sets of two matroids, or $(iv)$ $\mathcal{I}$ is a matching.[3] A solution for the instance is $S \in \mathcal{I}$ and $(A_t)_{t=1}^{d}$, where $A_t$ is a feasible assignment of $S$ w.r.t $\mathcal{K}_t$ for $1 \leq t \leq d$. The value of the solution is $f(S)$, and the objective is to find a solution of maximal value.

We assume the function $f$ is given via a value oracle. We further assume that the input indicates the type of constraint that $\mathcal{I}$ represents. Finally, $\mathcal{I}$ is given via a membership oracle, and if $\mathcal{I}$ is a matroid intersection, a membership oracle is given for each matroid.

[1] Alternatively, for every $S, T \subseteq I$: $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$.
[2] A formal definition for matroid can be found in [34].
[3] $\mathcal{I}$ is a matching if there is a graph $G = (V, I)$, and $S \in \mathcal{I}$ *iff* $S$ is a matching in $G$.

**Table 1** Results of Theorem 1 for $d$-MKCP.

| Type of Additional Constraint | Modular Maximization | Monotone Submodular Max. | Non-Monotone Sub. Max |
|---|---|---|---|
| No additional constraint | PTAS | $1 - e^{-1} - \varepsilon$ | $0.385 - \varepsilon$ |
| Matroid constraint | PTAS | $1 - e^{-1} - \varepsilon$ | – |
| 2 matroids or a matching | PTAS | – | – |

We refer to the special case in which $f$ is monotone (modular) as monotone (modular) $d$-MKCP. Also, we use non-monotone $d$-MKCP when referring to general $d$-MKCP instances. Similarly, we refer to the special case in which $\mathcal{I}$ is an independent set of a matroid (intersection of independent sets of two matroids or a matching) as $d$-MKCP with a matroid (matroid intersection or matching) constraint. If $\mathcal{I} = 2^I$ we refer to the problem as $d$-MKCP with no additional constraint. Thus, for example, in instances of modular 1-MKCP with a matroid constraint the function $f$ is modular and $\mathcal{I}$ is an independent set of a matroid.

Instances of $d$-MKCP naturally arise in various settings (see a detailed example in the full version of this paper [16]).

## 1.2 Our Results

Our main results are summarized in the next theorem (see also Table 1).

▶ **Theorem 1.** *For any fixed $d \in \mathbb{N}_+$ and $\varepsilon > 0$, there is*

1. *A randomized PTAS for modular d-MKCP ($(1 - \varepsilon)$-approximation). The same holds for this problem with a matroid constraint, matroid intersection constraint, or a matching constraint.*
2. *A polynomial-time random $(1 - e^{-1} - \varepsilon)$-approximation for monotone d-MKCP with a matroid constraint.*
3. *A polynomial-time random $(0.385 - \varepsilon)$-approximation for non-monotone d-MKCP with no additional constraint.*

All of the results are obtained using a single algorithm (Algorithm 2). The general algorithmic result encapsulates several important special cases. The Multiple-Choice Multiple Knapsack Problem is a variant of the Multiple Knapsack Problem in which the items are partitioned into classes $C_1, \ldots, C_k$, and at most one item can be selected from each class. Formally, Multiple-Choice Multiple Knapsack is the special case of modular 1-MKCP where $\mathcal{I}$ describes a partition matroid.[4] The problem has natural applications in network optimization [12, 37]. The best known approximation ratio for the problem is 2 due to [12]. This approximation ratio is improved by Theorem 1, as stated in the following.

▶ **Corollary 2.** *There is a randomized PTAS for the Multiple-Choice Multiple Knapsack Problem.*

While the Multiple Knapsack Problem and the Monotone Submodular Multiple Knapsack Problem are well understood [8, 23, 24, 15, 35], no results were previously known for the Non-Monotone Submodular Multiple Knapsack Problem, the special case of non-montone 1-MKCP with no additional constraint. A constant approximation ratio for the problem is obtained as a special case of Theorem 1.

[4] That is, $\mathcal{I} = \{S \subseteq I \mid \forall 1 \leq j \leq k : \ |S \cap C_j| \leq 1\}$ where $C_1, \ldots, C_k$ is a partition of $I$.

▶ **Corollary 3.** *For any $\varepsilon > 0$ there is a polynomial time random $(0.385 - \varepsilon)$-approximation for the Non-Monotone Submodular Multiple Knapsack Problem.*

A PTAS for Multistage Multiple Knapsack, a multistage version of the Multiple Knapsack Problem, can be obtained via a reduction to modular $d$-MKCP with a matroid constraint.[5] Here, to obtain a $(1 - O(\varepsilon))$-approximation for the multistage problem, the reduction solves instances of modular $\Theta\left(\frac{1}{\varepsilon}\right)$-MKCP with a matroid constraint (see [14] for details). Beyond the rich set of applications, our ability to derive such a general result is an evidence for the robustness of fractional grouping, the main technical contribution of this paper.

Our result for modular $d$-MKCP, for $d \geq 2$, generalizes the PTAS for the classic $d$-dimensional Knapsack problem ($\mathcal{I} = 2^I$ and $|B_t| = 1$ for any $1 \leq t \leq d$). Furthermore, a PTAS is the best we can expect as there is no *efficient PTAS (EPTAS)* already for $d$-dimensional Knapsack, unless W[1] = FPT [28]. While there is a well-known PTAS for Multiple Knapsack [8], existing techniques do not readily enable handling additional constraints, such as a matroid constraint.

The approximation ratio obtained for monotone $d$-MKCP is nearly optimal, as for any $\varepsilon > 0$ there is no $(1 - e^{-1} + \varepsilon)$-approximation for monotone submodular maximization with a cardinality constraint in the oracle model [32]. The approximation ratio is also tight under $P \neq NP$ due to the special case of coverage functions [19]. Previous works [15, 35] obtained the same approximation ratio for the Monotone Submodular Multiple Knapsack Problem (i.e, monotone 1-MKCP). However, as in the modular case, existing techniques are limited to handling a single MKC (with no other constraints).

In the non-montone case, the approximation ratio is in fact $(c - \varepsilon)$ for any $\varepsilon > 0$, where $c > 0.385$ is the ratio derived in [4]. This approximation ratio matches the current best known ratio for non-monotone submodular maximization with a single knapsack constraint [4]. A 0.491 hardness of approximation bound for non-monotone $d$-MKCP follows from [22].

Our technique can be cast also as a variant of *contention resolution scheme* [11]. The scheme can be used to derive approximation algorithms for special cases of $d$-MKCP which are not considered in Theorem 1. Such a scheme can be found in an earlier version of this paper [17].[6]

The *Monotone Submodular Multiple Knapsack Problem with Uniform Capacities* (USMKP) is the special case of $d$-MKCP in which $\mathcal{I} = 2^I$, $d = 1$, $f$ is monotone, and furthermore, all the bins in the MKC have the same capacity. That is, $\mathcal{K}_1 = (w, B, W)$ and $W(b_1) = W(b_2)$ for any $b_1, b_2 \in B$. This restricted variant of $d$-MKCP commonly arises in real-life applications (e.g., in file assignment to several identical storage devices). The best known approximation ratio for USMKP is $(1 - e^{-1} - \varepsilon)$ for any fixed $\varepsilon > 0$ [15, 35]. Another contribution of this paper is an improvement of this ratio.

▶ **Theorem 4.** *There is a polynomial-time random $(1 - e^{-1} - O((\log |B|)^{-\frac{1}{4}}))$-approximation for the Monotone Submodular Multiple Knapsack Problem with Uniform Capacities.*

## 1.3 Related Work

In the classic Multiple Knapsack problem, the goal is to maximize a modular set function subject to a single multiple knapsack constraint. A PTAS for the problem was first presented by Chekuri and Khanna [8]. The authors also ruled out the existence of a *fully polynomial time approximation scheme (FPTAS)*. An EPTAS was later developed by Jansen [23, 24].

[5] See, e.g., [2] for the Multistage Knapsack model.
[6] We were unable to obtain tight approximation ratios for the studied problems using this approach.

In the Bin Packing problem, we are given a set $I$ of items, a weight function $w : I \rightarrow \mathbb{R}_{\geq 0}$ and a capacity $W > 0$. The objective is to partition the set $I$ into a minimal number of sets $S_1, \ldots, S_m$ (i.e., find a *packing*) such that $\sum_{i \in S_b} w(i) \leq W$ for all $1 \leq b \leq m$. In [25] the authors presented a polynomial-time algorithm which returns a packing using $\text{OPT} + O(\log^2 \text{OPT})$ bins, where OPT is the number of bins in a minimal packing. The result was later improved by Rothvoß [33].

Research work on monotone submodular maximization dates back to the late 1970's. In [32] Nemhauser and Wolsey presented a greedy-based tight $(1 - e^{-1})$-approximation for maximizing a monotone submodular function subject to a cardinality constraint, along with a matching lower bound in the oracle model. The greedy algorithm of [32] was later generalized to monotone submodular maximization subject to a knapsack constraint [27, 36].

A major breakthrough in the field of submodular optimization resulted from the introduction of algorithms for optimizing the *multilinear extension* of a submodular function ([6, 30, 7, 40, 20, 5]). For $\bar{x} \in [0,1]^I$, we say that a random set $S \subseteq I$ is distributed by $\bar{x}$ (i.e., $S \sim \bar{x}$) if $\Pr(i \in S) = \bar{x}_i$, and the events $(i \in S)_{i \in I}$ are independent. Given a function $f : 2^I \rightarrow \mathbb{R}_{\geq 0}$, its *multilinear extension* is $F : [0,1]^I \rightarrow \mathbb{R}_{\geq 0}$ defined as $F(\bar{x}) = \mathbb{E}_{S \sim \bar{x}}[f(S)]$.

The input for the problem of optimizing the multilinear relaxation is an oracle for a submodular function $f : 2^I \rightarrow \mathbb{R}_{\geq 0}$ and a downward closed solvable polytope $P$.[7] The objective is to find $\bar{x} \in P$ such that $F(\bar{x})$ is maximized, where $F$ is the multilinear extension of $f$. The problem admits a random $(1 - e^{-1} - o(1))$-approximation in the monotone case and a random $(0.385 + \delta)$-approximation in the non-monotone case (for some small constant $\delta > 0$) due to [7] and [4].

Several techniques were developed for *rounding* a (fractional) solution for the multilinear optimization problem to an integral solution. These include Pipage Rounding [1], Randomized Swap Rrounding [9], and Contention Resolution Schemes [11]. These techniques led to the state of art results for many problems (e.g., [29, 7, 1, 9]).

A random $(1 - e^{-1} - \varepsilon)$-approximation for the Monotone Submodular Multiple Knapsack problem was presented in [15]. The technique in [15] modifies the objective function and its domain. This modification does not preserve submodularity of a non-montone function and the combinatorial properties of additional constraints. Thus, it does not generalize to $d$-MKCP.

A deterministic $(1-e^{-1}-\varepsilon)$-approximation for Monotone Submodular Multiple Knapsack was later obtained by Sun et al. [35]. Their algorithm relies on a variant of the greedy algorithm of [36] which cannot be extended to the non-monotone case, or easily adapted to handle more than a single MKC.

## 1.4 Technical Overview

In the following we describe the technical problem solved by fractional grouping and give some insight to the way we solve this problem. For simplicity, we focus on the special case of 1-MKCP, in which the number of bins is large and all bins have unit capacity. Let $(I, (w, B, W), 2^I, f)$ be a 1-MCKP instance where $W(b) = 1$ for all $b \in B$. Also, assume that no two items have the same weight. Let $S^*$ and $A^*$ be an optimal solution for the instance.

[7] A polytope $P \in [0,1]^I$ is *downward closed* if for any $\bar{x} \in P$ and $\bar{y} \in [0,1]^I$ such that $\bar{y} \leq \bar{x}$ (that is, $\bar{y}_i \leq \bar{x}_i$ for every $i \in I$) it holds that $\bar{y} \in P$. A polytope $P \in [0,1]^I$ is *solvable* if, for any $\bar{\lambda} \in \mathbb{R}^I$, a point $\bar{x} \in P$ such that $\bar{\lambda} \cdot \bar{x} = \max_{\bar{y} \in P} \bar{\lambda} \cdot \bar{y}$ can be computed in polynomial time, where $\bar{\lambda} \cdot \bar{x}$ is the dot product of $\bar{\lambda}$ and $\bar{x}$.

Fix an arbitrary small $\mu > 0$ such that $\mu^{-2} \in \mathbb{N}$. We say that an item $i \in I$ is *heavy* if $w(i) > \mu$; otherwise, $i$ is *light*. Let $H \subseteq I$ denote the heavy items. We can apply linear grouping [13] to the heavy items in $S^*$. That is, let $h^* = |S^* \cap H|$ be the number of heavy items in $S^*$, and partition $S^* \cap H$ to $\mu^{-2}$ groups of cardinality $\mu^2 \cdot h^*$, assuming the items are sorted in decreasing order by weights (for simplicity, assume $h^* \geq \mu^{-2}$ and $\mu^2 \cdot h^* \in \mathbb{N}$). Specifically, $S^* \cap H = G_1^* \cup \ldots \cup G_{\mu^{-2}}^*$, where $|G_k^*| = \mu^2 \cdot h^*$ for all $1 \leq k \leq \mu^{-2}$ and for any $i_1 \in G_{k_1}^*$, $i_2 \in G_{k_2}^*$ where $k_1 < k_2$ we have that $w(i_1) > w(i_2)$. Also, for any $1 \leq k \leq \mu^{-2}$ let $q_k$, the $k$-th pivot, be the item of highest weight in $G_k^*$.

We use the pivots to generate a new collection of groups $G_1, \ldots, G_{\mu^{-2}}$ where $G_k = \{i \in H \mid w(q_{k+1}) < w(i) \leq w(q_k)\}$ for $1 \leq k < \mu^{-2}$, and $G_{\mu^{-2}} = \{i \in H \mid w(i) \leq w(q_{\mu^{-2}})\}$. Clearly, $G_k^* \subseteq G_k$ for any $1 \leq k \leq \mu^{-2}$. Let $X = \{i \in H \mid w(i) > w(q_1)\}$ be the set of largest items in $H$.

A standard *shifting* argument can be used to show that any set $S \subseteq I \setminus X$, such that $w(S) \leq |B|$ and $|S \cap G_k| \leq \mu^2 \cdot h^*$ for all $1 \leq k \leq \mu^{-2}$, can be packed into $(1 + 2\mu)|B| + 1$ bins as follows.[8] The items in $S \cap G_k$ can be packed in place of the items in $G_{k-1}^*$ in $A^*$, each of the items in $S \cap G_1$ can be packed in a separate bin (observe that $|S \cap G_1| \leq \mu^2 \cdot h^* \leq \mu|B|$ as packing of $h^*$ heavy items requires at least $h^* \cdot \mu$ bins). Finally, First-Fit can be used to pack the light items in $S$.

Now, assume we know $q_1, \ldots, q_{\mu^{-2}}$ and $h^*$; thus, the sets $G_1, \ldots, G_{\mu^{-2}}$ and $X$ can be constructed. Consider the following optimization problem: find $S \subseteq I \setminus X$ such that $w(S) \leq |B|$, $|S \cap G_k| \leq \mu^2 \cdot h^*$ for all $1 \leq k \leq \mu^{-2}$, and $f(S)$ is maximal. The problem is an instance of non-monotone submodular maximization with a $(1 + \mu^{-2})$-dimensional knapsack constraint, for which there is a $(0.385 - \varepsilon)$-approximation algorithm [29, 4]. The algorithm can be used to find $S \subseteq I \setminus X$ which satisfies the above constraints and $f(S) \geq (0.385 - \varepsilon) \cdot f(S^*)$, as $S^*$ is a feasible solution for the problem. Subsequently, $S$ can be packed into bins using a standard bin packing algorithm. This will lead to a packing of $S$ into roughly $(1 + 2\mu)|B| + O(\log^2 |B|)$ bins. By removing the bins of least value (along with their items), and using the assumption that $|B|$ is sufficiently large, we can obtain a set $S'$ and an assignment of $S'$ into $B$ such that $f(S)$ is arbitrarily close to $0.385 \cdot f(S^*)$.

Indeed, we do not know the values of $q_1, \ldots, q_{\mu^{-2}}$ and $h^*$. This prevents us from using the above approach. However, as in [3], we can overcome this difficulty through exhaustive enumeration. Each of $q_1, \ldots, q_{\mu^{-2}}$ and $h^*$ takes one of $|I|$ possible values. Thus, by iterating over all $|I|^{1+\mu^{-2}}$ possible values for $q_1, \ldots, q_{\mu^{-2}}$ and $h^*$, and solving the above problem for each, we can find a solution of value at least $0.385 \cdot f(S^*)$.

While this approach is useful for our restricted class of instances, due to the use of exhaustive enumeration it does not scale to general instances, where bin capacities may be arbitrary. Known techniques ([15]) can be used to reduce the number of unique bin capacities in a general MKC to be logarithmic in $|B|$. As enumeration is required for each unique capacity, this results in $|I|^{\Theta(\log |B|)}$ iterations, which is non-polynomial.

Fractional Grouping overcomes this hurdle by using a polytope $P \subseteq [0,1]^I$ to represent an MKC. A grouping $G_1^{\bar{y}}, \ldots, G_\tau^{\bar{y}}$ with $\tau \leq \mu^{-2} + 1$ is derived from a vector $\bar{y} \in P$. The polytope $P$ bears some similarity to configuration linear programs used in previous works ([24, 21, 3]). While $P$ is not solvable, it satisfies an approximate version of solvability which suffices for our needs.

Fractional grouping satisfies the main properties of the grouping defined for $S^*$. Each of the groups contains roughly the same number of fractionally selected items. That is, $\sum_{i \in G_k^{\bar{y}}} \bar{y}_i \approx \mu^2 |B|$ for all $1 \leq k \leq \tau$. Furthermore, we show that if $\bar{y}$ is strictly contained in

[8] For a set $S \subseteq I$ we denote $w(S) = \sum_{i \in S} w(i)$.

$P$ then any subset $S \subseteq I$ satisfying $(i)$ $|S \cap G_k| \leq \mu|B|$ for all $1 \leq k \leq \tau$, and $(ii)$ $w(S \setminus H)$ is sufficiently small, can be packed into strictly less than $|B|$ bins (see the details in Section 2). The existence of a packing for $S$ relies on a shifting argument similar to the one used above. In this case, however, the structure of the polytope $P$ replaces the role of $S^*$ in our discussion.

This suggests the following algorithm. Use the algorithm of [4] to find $\bar{y} \in P$ such that $F(\bar{y}) \geq (0.385 - \varepsilon)f(S^*)$, and sample a random set $R \sim (1-\delta)^2\bar{y}$. By the above property, $R$ can be packed into strictly less than $|B|$ bins with high probability, as $\mathbb{E}\left[|R \cap G_k|\right] \ll \mu|B|$. Thus, $R$ can be packed into $B$ using a bin packing algorithm. Standard tools (specifically, the FKG inequality as used in [11]) can also be used to show that $\mathbb{E}[f(R)]$ is arbitrarily close to $F(\bar{y})$. Hence, we can obtain an approximation ratio arbitrarily close to $(0.385 - \varepsilon)$ while avoiding enumeration.

This core idea of fractional grouping for bins of uniform capacities can be scaled to obtain Theorem 1. This scaling involves use of existing techniques for submodoular optimization ([15, 9, 10, 7, 4]), along with a novel *block association* technique we apply to handle MKCs with arbitrary bin capacities.

### Organization

We present the fractional grouping technique in Section 2. Our algorithms for uniform bin capacities and the general case are given in Section 3 and 4, respectively. Due to space constraints, the block association technique and some proofs are omitted. Those appear in the full version [16].

## 2 Fractional Grouping

Given an MKC $(w, B, W)$ over $I$, a subset of bins $K \subseteq B$ is a *block* if all the bins in $K$ have the same capacity. Denote by $W^*_K$ the capacities of the bins in block $K$, then $W^*_K = W(b)$ for any $b \in K$.

We first define a polytope $P_K$ which represents the block $K \subseteq B$ of an MKC $(w, B, W)$ over $I$. To simplify the presentation, we assume the MKC $(w, B, W)$ and $K$ are fixed throughout this section. W.l.o.g., assume that $I = \{1, 2, \ldots, n\}$ and $w(1) \geq w(2) \geq \ldots \geq w(n)$. A $K$-*configuration* is a subset $C \subseteq I$ of items which fits into a single bin of block $K$, i.e., $w(C) \leq W^*_K$. We use $\mathcal{C}_K$ to denote the set of all $K$-configurations. Formally, $\mathcal{C}_K = \{C \subseteq I \mid w(C) \leq W^*_K\}$.

▶ **Definition 5.** *The* extended block polytope of $K$ *is*

$$P^e_K = \left\{ \bar{y} \in [0,1]^I, \bar{z} \in [0,1]^{\mathcal{C}_K} \;\middle|\; \begin{array}{lrcl} & \sum_{C \in \mathcal{C}_K} \bar{z}_C & \leq & |K| \\ \forall i \in I: & \bar{y}_i & \leq & \sum_{C \in \mathcal{C}_K \text{ s.t. } i \in C} \bar{z}_C \end{array} \right\} \tag{1}$$

The first constraint in (1) bounds the number of selected configurations by the number of bins. The second constraint requires that each selected item is (fractionally) covered by a corresponding set of configurations. It is easy to verify that, for any $(\bar{y}, \bar{z}) \in P^e_K$, it holds that $\sum_{i \in I} w(i) \cdot \bar{y}_i \leq |K| \cdot W^*_K$.

▶ **Definition 6.** *The* block polytope of $K$ *is*

$$P_K = \left\{ \bar{y} \in [0,1]^I \mid \exists \bar{z} \in [0,1]^{\mathcal{C}_K} : (\bar{y}, \bar{z}) \in P^e_K \right\}. \tag{2}$$

While $P_K^e$ and $P_K$ are defined using an exponential number of variables (as $\bar{z} \in [0,1]^{\mathcal{C}_K}$ and $\mathcal{C}_K$ is exponential), it follows from standard arguments (see, e.g., [21, 25]) that, for any $\bar{c} \in \mathbb{R}^I$, $\max_{\bar{y} \in P_K} \bar{c} \cdot \bar{y}$ can be approximated.

▶ **Lemma 7.** *There is a* fully polynomial-time approximation scheme (FPTAS) *for the problem of finding $\bar{y} \in P_K$ such that $\bar{c} \cdot \bar{y}$ is maximized, given an MKC $(w, B, W)$, a block $K \subseteq B$ and a vector $\bar{c} \in \mathbb{R}^I$, where $P_K$ is the block polytope of $K$.*

A formal proof for Lemma 7 if given in [16]. We say that $A : K \to 2^I$ is a *feasible assignment for* $K$ if $w(A(b)) \leq W_K^*$ for any $b \in K$. Also, we use $\mathbb{1}_S = \bar{x} \in \{0,1\}^I$, where $\bar{x}_i = 1$ if $i \in S$ and $\bar{x}_i = 0$ if $i \in I \setminus S$. The next lemma implies that the definition of $P_K^e$ is sound for the problem.

▶ **Lemma 8.** *Let $A$ be a feasible assignment for $K$ and $S = \bigcup_{b \in K} A(b)$. Then $\mathbb{1}_S \in P_K$.*

The lemma is easily proved, by setting $\bar{z}_C = 1$ if $A(b) = C$ for some $b \in B$, and $\bar{z}_C = 0$ otherwise. We say an item $i \in I$ is *$\mu$-heavy* for $\mu > 0$ (w.r.t $K$) if $W_K^* \geq w(i) > \mu \cdot W_K^*$; $i \in I$ is *$\mu$-light* if $w(i) \leq \mu W_K^*$. Denote by $H_{K,\mu}$ and $L_{K,\mu}$ the sets of $\mu$-heavy items and $\mu$-light items, respectively.

Given a vector $\bar{y} \in P_K$, we now describe the partition of $\mu$-heavy items into groups $G_1, \ldots, G_\tau$, for some $\tau \leq \mu^{-2} + 1$. Starting with $k = 1$ and $G_k = \emptyset$, add items from $H_{K,\mu}$ to the current group $G_k$ until $\sum_{i \in G_k} \bar{y}_i \geq \mu|K|$. Once the constraint is met, mark the index of the last item in $G_k$ as $q_k$, the *$\mu$-pivot* of $G_k$, close $G_k$ and open a new group, $G_{k+1}$. Each of the groups $G_1, \ldots, G_{\tau-1}$ represents a fractional selection of $\approx \mu|K|$ heavy items of $\bar{y}$. The last group, $G_\tau$, contains the remaining items in $H_{K,\mu}$, for which the $\mu$-pivot is $q_{max}$ (last item in $H_{K,\mu}$). We now define formally the partition process.

▶ **Definition 9.** *Let $\bar{y} \in P_K$ and $\mu \in \left(0, \frac{1}{2}\right]$. Also, let $q_0 \in \{0, 1, \ldots, n\}$ and $q_{\max} \in I$ such that $H_{K,\mu} = \{i \in I \mid q_0 < i \leq q_{\max}\}$. The $\mu$-pivots of $\bar{y}$, given by $q_1, \ldots, q_\tau$, are defined inductively, i.e.,*

$$q_k = \min \left\{ s \in H_{K,\mu} \;\middle|\; \sum_{i=q_{k-1}+1}^{s} \bar{y}_i \geq \mu \cdot |K| \right\}.$$

*If the set over which the minimum is taken is empty, let $\tau = k$ and $q_\tau = q_{\max}$. The $\mu$-*grouping *of $\bar{y}$ consists of the sets $G_1, \ldots, G_\tau$, where $G_k = \{i \in H_{K,\mu} \mid q_{k-1} < i \leq q_k\}$ for $1 \leq k \leq \tau$.*

Note that in the above definition it may be that $q_0 \neq 0$ as there may be items $i \in I$ for which $w(i) > W_K^*$. Given a polytope $P$ and $\delta \in \mathbb{R}$, we use the notation $\delta P = \{\delta \bar{x} \mid \bar{x} \in P\}$. The main properties of fractional grouping are summarized in the next lemma.

▶ **Lemma 10** (Fractional Grouping). *For any $\bar{y} \in P_K$ and $0 < \mu < \frac{1}{2}$ there is a polynomial time algorithm which computes a partition $G_1, \ldots, G_\tau$ of $H_{K,\mu}$ with $\tau \leq \mu^{-2} + 1$ for which the following hold:*

1. *$\sum_{i \in G_k} \bar{y}_i \leq \mu \cdot |K| + 1$ for any $1 \leq k \leq \tau$.*
2. *Let $S \subseteq H_{K,\mu} \cup L_{K,\mu}$ such that $|S \cap G_k| \leq \mu|K|$ for every $1 \leq k \leq \tau$, and $w(S \cap L_{K,\mu}) \leq \sum_{i \in L_{K,\mu}} \bar{y}_i \cdot w(i) + \lambda \cdot W_K^*$ for some $\lambda \geq 0$. Also, assume $\bar{y} \in (1-\delta)P_K$ for some $\delta \geq 0$. Then $S$ can be packed into $(1 - \delta + 3\mu)|K| + 4 \cdot 4^{\mu^{-2}} + 2\lambda$ bins of capacity $W_K^*$.*

*We refer to $G_1, \ldots, G_\tau$ as* the $\mu$-grouping *of $\bar{y}$.*

**Proof.** It follows from Definition 9 that $G_1, \ldots, G_\tau$ can be computed in polynomial time. Also, $\sum_{i \in G_\tau} \bar{y}_i < \mu \cdot |K|$ and

$$\forall 1 \leq k < \tau : \qquad \mu \cdot |K| \leq \sum_{i \in G_k} \bar{y}_i \leq \mu \cdot |K| + 1. \tag{3}$$

Furthermore, $\tau \leq \mu^{-2} + 1$. Thus, it remains to show Property 2 in the lemma.

Define the *type* of a configuration $C \in \mathcal{C}_K$, denoted by type$(C)$, as the vector $T \in \mathbb{N}^\tau$ with $T_k = |C \cap G_k|$. Let $\mathcal{T} = \{\text{type}(C) \mid C \in \mathcal{C}_K\}$ be the set of all types. Given a type $T \in \mathcal{T}$, consider a set of items $Q \subseteq H_{K,\mu} \setminus G_1$, such that $|Q \cap G_k| \leq T_{k-1}$ for any $2 \leq k \leq \tau$, then $w(Q) \leq W_K^*$. This is true since we assume the items in $H_{K,\mu}$ are sorted in non-increasing order by weights. We use this key property to construct a packing for $S$.

We note that $\sum_{k=1}^{\tau} |C \cap G_k| < \mu^{-1}$ for any $C \in \mathcal{C}_K$ (otherwise $w(C) > W_K^*$, as $G_k \subseteq H_{K,\mu}$). It follows that $|\mathcal{T}| \leq 4^{\mu^{-2}}$. Indeed, the number of types is bounded by the number of different non-negative integer $\tau$-tuples whose sum is at most $\mu^{-1}$.

By Definition 5, there exists $\bar{z} \in [0,1]^{\mathcal{C}_K}$ such that $(\bar{y}, \bar{z}) \in (1-\delta)P_K^e$. For $T \in \mathcal{T}$, let $\eta(T) = \sum_{C \in \mathcal{C}_K \text{ s.t. type}(C) = T} \bar{z}_C$. Then, for any $1 \leq k \leq \tau - 1$, we have

$$\mu|K| \leq \sum_{i \in G_k} \bar{y}_i \leq \sum_{i \in G_k} \sum_{C \in \mathcal{C}_k \text{ s.t. } i \in C} \bar{z}_C = \sum_{C \in \mathcal{C}_K} |G_k \cap C| \bar{z}_C = \sum_{T \in \mathcal{T}} T_k \cdot \eta(T) \tag{4}$$

The first inequality follows from (3). The second inequality follows from (1). The two equalities follow by rearranging the terms.

Using $\bar{z}$ (through the values of $\eta(T)$) we define an assignment of $S \cap (G_2 \cup \ldots \cup G_\tau)$ to $\eta = \sum_{T \in \mathcal{T}} \lceil \eta(T) \rceil$ bins. We initialize $\eta$ sets (bins) $A_1, \ldots, A_\eta = \emptyset$ and associate a type with each set $A_b$, such that there are $\lceil \eta(T) \rceil$ sets associated with the type $T \in \mathcal{T}$, using a function $R$. That is, let $R : \{1, 2, \ldots, \eta\} \to \mathcal{T}$ such that $|R^{-1}(T)| = \lceil \eta(T) \rceil$. We assign the items in $S \cap (G_2 \cup \ldots \cup G_\tau)$ to $A_1, \ldots, A_\eta$ while ensuring that $|A_b \cap G_k| \leq R(b)_{k-1}$ for any $1 \leq b \leq \eta$ and $2 \leq k \leq \tau$. In other words, the number of items assigned to $A_b$ from $G_k$ is at most the number of items from $G_{k-1}$ in the configuration type $T$ assigned to bin $b$ by $R$. The assignment is obtained as follows. For every $2 \leq k \leq \tau$, iterate over the items $i \in S \cap G_k$, find $1 \leq b \leq \eta$ such that $|A_b \cap G_k| < R(b)_{k-1}$ and set $A_b \leftarrow A_b \cup \{i\}$. It follows from (4) and the conditions of the lemma that such $b$ will always be found.

Upon completion of the process, we have that $S \cap (G_2 \cup \ldots \cup G_\tau) = A_1 \cup \ldots \cup A_\eta$. Furthermore, for every $1 \leq b \leq \eta$, there are $C \in \mathcal{C}_K$ and $T \in \mathcal{T}$ such that type$(C) = T = R(b)$. Since $A_b \subseteq G_2 \cup \ldots \cup G_\tau$, we have

$$w(A_b) = \sum_{k=2}^{\tau} w(A_b \cap G_k) \leq \sum_{k=2}^{\tau} T_{k-1} \cdot w(q_{k-1}) = \sum_{k=2}^{\tau} |C \cap G_{k-1}| \cdot w(q_{k-1}) \leq \sum_{i \in C} w(i) \leq W_K^*.$$

The first inequality holds since $w(q_{k-1}) \geq w(i)$ for every $i \in G_k$, and the second holds since $w(q_{k-1}) \leq w(i)$ for every $i \in G_{k-1}$. By similar arguments, for every $2 \leq k \leq \tau$, we have

$$w(S \cap G_k) \leq |S \cap G_k| \cdot w(q_{k-1}) \leq \mu|K| \cdot w(q_{k-1}) \leq \sum_{i \in G_{k-1}} \bar{y}_i \cdot w(q_{k-1}) \leq \sum_{i \in G_{k-1}} \bar{y}_i \cdot w(i). \tag{5}$$

The third inequality is due to (3). Using (5) and the conditions in the lemma,

$$\begin{aligned} w(S \setminus G_1) = w(S \cap L_{K,\mu}) + \sum_{k=2}^{\tau} w(S \cap G_k) \ &\leq \sum_{i \in L_{K,\mu}} \bar{y}_i w(i) + \lambda W_K^* + \sum_{k=1}^{\tau-1} \sum_{i \in G_k} \bar{y}_i w(i) \\ &\leq \sum_{i \in I} \bar{y}_i \cdot w(i) + \lambda W_K^* \leq (1-\delta) W_K^* \cdot |K| + \lambda W_K^*. \end{aligned} \tag{6}$$

We use First-Fit (see, e.g., Chapter 9 in [39]) to add the items in $S \cap L_{K,\mu}$ to the sets (=bins) $A_1, \ldots, A_\eta$ while maintaining the capacity constraint, $w(A_b) \leq W^*_K$. First-Fit iterates over the items $i \in S \cap L_{K,\mu}$ and searches for a minimal $b$ such that $w(A_b \cup \{i\}) \leq W^*_K$. If such $b$ exists, First-Fit updates $A_b \leftarrow A_b \cup \{i\}$; otherwise, it adds a new bin with $i$ as its content. Let $\eta'$ be the number of bins by the end of the process. As $w(i) \leq \mu W^*_K$ for $i \in S \cap L_{K,\mu}$, and due to (6), it holds that $\eta' \leq \max\{\eta, (|K|(1-\delta)+\lambda)(1+2\mu)+1\}$. Finally,

$$\eta = \sum_{T \in \mathcal{T}} \lceil \eta(T) \rceil \leq |\mathcal{T}| + \sum_{T \in \mathcal{T}} \eta(T) \leq 4^{\mu^{-2}} + \sum_{C \in \mathcal{C}_K} \bar{z}_C \leq 4^{\mu^{-2}} + (1-\delta)|K|.$$

Thus, there is a packing of $S \setminus G_1$ into at most $(1-\delta)|K| + 4^{\mu^{-2}} + 1 + 2\mu|K| + 2\lambda$ bins of capacity $W^*_K$. Since $|S \cap G_1| \leq \mu|K|$, each of the items in $S \cap G_1$ can be packed into a bin of its own. This yields a packing using at most $(1-\delta+3\mu)|K| + 4 \cdot 4^{\mu^{-2}} + 2\lambda$ bins. ◀

## 3 Uniform Capacities

In this section we apply fractional grouping (as stated in Lemma 10) to solve the Monotone Submodular Multiple Knapsack Problem with Uniform Capacities (USMKP). An instance of the problem consists of an MKC $(w, B, W)$ over a set $I$ of items, such that $W^*_B = W(b)$ for all $b \in B$, and a submodular function $f : 2^I \to \mathbb{R}_{\geq 0}$. For simplicity, we associate a solution for the problem with a feasible assignment $A : B \to 2^I$. Then, the set of assigned items is given by $S = \bigcup_{b \in B} A(b)$.

Our algorithm for USMKP instances applies *Pipage Rounding* [1, 6]. The input for a Pipage Rounding step is a (fractional) solution $\bar{x} \in [0,1]^I$, and two items $i_1, i_2 \in I$ with costs $c_1, c_2 \geq 0$. The Pipage Rounding step returns a new random solution $\bar{x}' \in [0,1]^I$ such that $\bar{x}'_i = \bar{x}_i$ for $i \in I \setminus \{i_1, i_2\}$, $\bar{x}_{i_1} \cdot c_1 + \bar{x}_{i_2} \cdot c_2 = \bar{x}'_{i_1} \cdot c_1 + \bar{x}'_{i_2} \cdot c_2$, and either $\bar{x}'_{i_1} \in \{0,1\}$ or $\bar{x}'_{i_2} \in \{0,1\}$. Furthermore, for any submodular function $f : 2^I \to \mathbb{R}_{\geq 0}$ it holds that $\mathbb{E}[F(\bar{x}')] \geq F(\bar{x})$, where $F$ is the multilinear exetension of $f$. Algorithm 1 calls a subroutine $\mathsf{Pipage}(\bar{x}, f, G, \bar{c})$, which can be implemented by an iterative application of Pipage Rounding steps as long as $\bar{x}$ contains two fractional entries, and randomly sampling the last remaining fractional entry. The properties of Pipage are summarized in the next result.

▶ **Lemma 11.** *There is a polynomial time procedure $\mathsf{Pipage}(\bar{x}, f, G, \bar{c})$ for which the following holds. Given $\bar{x} \in [0,1]^I$, a submodular function $f : 2^I \to \mathbb{R}_{\geq 0}$, a subset of items $G \subseteq I$ and a cost vector for the items $\bar{c} \in \mathbb{R}^G_{\geq 0}$, the procedure returns a random vector $\bar{x}' \in [0,1]^I$ such that $\mathbb{E}[F(\bar{x}')] \geq F(\bar{x})$, $\bar{x}'_i \in \{0,1\}$ for $i \in G$, $\bar{x}'_i = \bar{x}_i$ for all $i \in I \setminus G$, and there is $i^* \in G$ such that $\sum_{i \in G} \bar{x}'_i \cdot c_i \leq c_{i^*} + \sum_{i \in G} \bar{x}_i \cdot c_i$.*

To solve USMKP instances, our algorithm initially finds $\bar{y} \in P_B$, where $P_B$ is the block polytope of $B$ (note that $B$ is a block in this case), for which $F(\bar{y})$ is *large* ($F$ is the multilinear extension of the value function $f$). The algorithm chooses a small value for $\mu$ and uses $G_1, \ldots, G_\tau$, the $\mu$-grouping of $(1-4\mu)\bar{y}$, to guide the rounding process. Pipage rounding is used to convert $(1-4\mu) \cdot \bar{y}$ to $S \subseteq I$ while preserving the number of selected items from each group as $\approx \mu|B|$, and the total weight of items selected from $L_{B,\mu}$ (i.e., $\mu$-light items) as $\approx (1-4\mu) \cdot \sum_{i \in L_{B,\mu}} \bar{y}_i \cdot w(i)$. An approximation algorithm for bin packing is then used to find a packing of $S$ to the bins. Lemma 10 ensures the resulting packing uses at most $|B|$ bins for sufficiently large $B$. In case the packing requires more than $|B|$ bins we simply assume the algorithm returns an empty solution. We give the pseudocode in Algorithm 1.

▶ **Lemma 12.** *Algorithm 1 yields a $\left(1 - e^{-1} - O\left((\log|B|)^{-\frac{1}{4}}\right)\right)$-approximation for USMKP.*

**Algorithm 1** Submodular Multiple Knapsack with Uniform Capacities.

**Input:** An MKC $(w, B, W)$ over $I$ with uniform capacities. A submodular function $f : 2^I \to \mathbb{R}_{\geq 0}$.

1 Find an approximate solution $\bar{y} \in P_B$ for $\max_{\bar{y} \in P_B} F(\bar{y})$, where $P_B$ is the block polytope of $B$, and $F$ is the multilinear extension of $f$.

2 Choose $\mu = \min \left\{ (\log |B|)^{-\frac{1}{4}}, \frac{1}{2} \right\}$.

3 Set $\bar{y}^0 \leftarrow (1 - 4\mu)\bar{y}$. and let $G_1, \ldots, G_\tau$ be the $\mu$-grouping of $\bar{y}^0$.

4 **for** $k = 1, 2, \ldots, \tau$ **do** $\bar{y}^k \leftarrow \mathsf{Pipage}\left(\bar{y}^{k-1}, f, G_k, \bar{1}\right)$.

5 $\bar{y}' = \mathsf{Pipage}\left(\bar{y}^\tau, f, L_{B,\mu}, (w(i))_{i \in L_{B,\mu}}\right)$.

6 Let $S = \{ i \in I \mid \bar{y}'_i = 1 \}$.

7 Pack the items in $S$ into $B$ using a bin packing algorithm. Return the resulting assignment.

**Proof.** Let $A^*$ be an optimal solution for the input instance, and $\text{OPT} = f\left(\bigcup_{b \in B} A^*(b)\right)$ its value. By Lemma 8, $\mathbb{1}_{\bigcup_{b \in B} A^*(b)} \in P_B$. Let $c = 1 - e^{-1}$. By using the algorithm of [7] we have that $F(\bar{y}) \geq \left(c - \frac{1}{|I|}\right) \cdot \text{OPT}$ ($\bar{y}$ is defined in Step 1 of Algorithm 1). The algorithm of [7] is used with the FPTAS of Lemma 7 as an oracle for solving linear optimization problems over $P_B$. We note that a $\left(c - \frac{1}{|I|}\right)$-approximate solution can be obtained even when the algorithm is only given an FPTAS (and not an exact solver) for linear optimization problems over the polytope.

Since the multilinear extension has negative second derivatives [7], it follows that $F(\bar{y}^0) \geq (1 - 4\mu) \cdot \left(c - \frac{1}{|I|}\right) \cdot \text{OPT}$. Now, consider the vector $\bar{y}'$ output in Step 5 of the algorithm. By Lemma 11, it follows that $\mathbb{E}\left[F(\bar{y}')\right] \geq F(\bar{y}^0) \geq (1-4\mu) \cdot \left(c - \frac{1}{|I|}\right) \cdot \text{OPT}$, and $\bar{y}' \in \{0,1\}^I$ (note that $\bar{y}'_i = \bar{y}_i = 0$ for any $i$ with $w(i) > W^*_B$ due to (1)). Thus, for the set $S$ defined in Step 6 of the algorithm, we have $\mathbb{E}\left[f(S)\right] \geq (1 - 4\mu) \cdot \left(c - \frac{1}{|I|}\right) \cdot \text{OPT} \geq \left(c - O\left((\log |B|)^{-\frac{1}{4}}\right)\right) \cdot \text{OPT}$ (observe we may assume w.l.o.g that $|I| \geq |B|$).

To complete the proof, it remains to show that the bin packing algorithm in Step 7 packs all items in $S$ into the bins $B$. By Lemma 11, for any $1 \leq k \leq \tau$, it holds that $|S \cap G_k| = \sum_{i \in G_k} \bar{y}'_i \leq 1 + \sum_{i \in G_k} \bar{y}^0_i \leq \mu \cdot |B| + 2$ (the last inequality follows from Lemma 10). Similarly, there is $i^* \in L_{B,\mu}$ such that

$$w(S \cap L_{B,\mu}) = \sum_{i \in L_{B,\mu}} \bar{y}'_i \cdot w(i) \leq w(i^*) + \sum_{i \in L_{B,\mu}} \bar{y}^0_i \cdot w(i) \leq \mu \cdot W^*_B + \sum_{i \in L_{B,\mu}} \bar{y}^0_i \cdot w(i).$$

To meet the conditions of Lemma 10, we need to remove (up to) two items from each group, i.e., $S \cap G_k$, for $1 \leq k \leq \tau$. Let $R \subseteq S$ be a minimal subset such that $|(S \setminus R) \cap G_k| \leq \mu |B|$ for all $1 \leq k \leq \tau$. By the above we have that $|R| \leq 2 \cdot \tau \leq 2 \cdot (\mu^{-2} + 1)$. Therefore, $S \setminus R$ satisfies the conditions of Lemma 10. Hence, by taking $\delta = 4\mu$ and $\lambda = \mu$, the items in $S \setminus R$ can be packed into $(1 - \mu)|B| + 4 \cdot 4^{\mu^{-2}} + 2\mu$ bins. By using an additional bin for each item in $R$, and assuming $|B|$ is large enough, the items in $S$ can be packed into

$$(1 - \mu)|B| + 4 \cdot 4^{\mu^{-2}} + 2\mu + 2 \cdot (\mu^{-2} + 1) \leq |B| - \frac{|B|}{(\log |B|)^{\frac{1}{4}}} + 5 \cdot 4^{\sqrt{\log |B|}} + 3 \leq |B|$$

bins of capacity $W^*_B$. Recall that the algorithm of [25] returns a packing in at most $\text{OPT} + O(\log^2 \text{OPT})$ bins. Thus, for large enough $|B|$, the number of bins used in Step 7 of

Algorithm 1 is at most

$$|B| - \frac{|B|}{(\log |B|)^{\frac{1}{4}}} + 5 \cdot 4^{\sqrt{\log |B|}} + O(\log^2 |B|) \leq |B|.$$

Finally, we note that Algorithm 1 can be implemented in polynomial time. ◀

## 4 Approximation Algorithm

In this section we present our algorithm for general instances of $d$-MKCP, which gives the result in Theorem 1. In designing the algorithm, a key observation is that we can restrict our attention to $d$-MKCP instances of certain structure, with other crucial properties satisfied by the objective function. For the *structure*, we assume the bins are partitioned into *levels* by capacities, using the following definition of [15].

▶ **Definition 13.** *For any $N \in \mathbb{N}$, a set of bins $B$ and capacities $W : B \to \mathbb{R}_{\geq 0}$, a partition $(K_j)_{j=0}^{\ell}$ of $B$ is $N$-*leveled *if, for all $0 \leq j \leq \ell$, $K_j$ is a block and $|K_j| = N^{\left\lfloor \frac{j}{N^2} \right\rfloor}$. We say that $B$ and $W$ are $N$-*leveled *if such a partition exists.*

For $N, \xi \in \mathbb{N}$, *$(N, \xi)$-restricted $d$-MKCP* is the special case of $d$-MKCP in which for any instance $\mathcal{R} = \left(I, (w_t, B_t, W_t)_{t=1}^{d}, \mathcal{I}, f\right)$ it holds that $B_t$ and $W_t$ are $N$-leveled for all $1 \leq t \leq d$, and $f(\{i\}) - f(\emptyset) \leq \frac{\text{OPT}}{\xi}$ for any $i \in I$, where OPT is the value of an optimal solution for the instance. We assume the input for $(N, \xi)$-restricted $d$-MKCP includes the $N$-leveled partition $(K_j^t)_{j=0}^{\ell_t}$ of $B_t$ for all $1 \leq t \leq d$. Combining standard enumeration with the structuring technique of [15], we derive the next result.

▶ **Lemma 14.** *For any $N, \xi, d \in \mathbb{N}$ and $c \in [0,1]$, a polynomial time $c$-approximation for modular/ monotone/ non-monotone $(N, \xi)$-restricted $d$-MKCP with a matroid/ matroid intersection/ matching/ no additional constraint implies a polynomial time $c \cdot \left(1 - \frac{d}{N}\right)$-approximation for $d$-MKCP, with the same type of function and same type of additional constraint.*

The proof of the lemma is given in [16].

Our algorithm for $(N, \xi)$-restricted $d$-MKCP associates a polytope with each instance. To this end, we first generalize the definition of a block polytope (Definition 6) to represent an MKC. We then use it to define a polytope for the whole instance.

▶ **Definition 15.** *For $\gamma > 0$, the extended $\gamma$-partition polytope of an MKC $(w, B, W)$ and the partition $(K_j)_{j=0}^{\ell}$ of $B$ to blocks is*

$$P^e = \left\{ (\bar{x}, \bar{y}^0, \ldots, \bar{y}^\ell) \;\middle|\; \begin{array}{ll} \bar{x} \in [0,1]^I & \\ \sum_{j=0}^{\ell} \bar{y}^j = \bar{x} & \\ \bar{y}^j \in P_{K_j} & \forall 0 \leq j \leq \ell \\ \bar{y}^j_i = 0 & \forall 0 \leq j \leq \ell, |K_j| = 1, i \in I \setminus L_{K_j,\gamma} \end{array} \right\} \tag{7}$$

*where $P_{K_j}$ is the block polytope of $K_j$, and $L_{K_j,\gamma}$ is the set of $\gamma$-light items of $K_j$. The $\gamma$-partition polytope of $(w, B, W)$ and $(K_j)_{j=0}^{\ell}$ is*

$$P = \left\{ \bar{x} \in [0,1]^I \;\middle|\; \exists \bar{y}^0, \ldots \bar{y}^\ell \in [0,1]^I \textit{ s.t. } (\bar{x}, \bar{y}^0, \ldots, \bar{y}^\ell) \in P^e \right\} \tag{8}$$

The last constraint in (7) forbids the assignment of $\gamma$-heavy items to blocks of a single bin. This technical requirement is used to show a concentration bound.

Finally, the *$\gamma$-instance polytope* of $\left(I,(w_t,B_t,W_t)_{t=1}^{d},\mathcal{I},f\right)$ and a partition $\left(K_j^t\right)_{j=0}^{\ell_t}$ of $B_t$ to blocks, for $1 \leq t \leq d$, is $P = P(\mathcal{I}) \cap \left(\bigcap_{t=1}^{d} P_t\right)$, where $P(\mathcal{I})$ is the convex hull of $\mathcal{I}$ and $P_t$ is the $\gamma$-partition polytope of $(w_t, B_t, W_t)$ and $\left(K_j^t\right)_{j=0}^{\ell_t}$. In the *instance polytope optimization problem*, we are given a $d$-MKCP instance $\mathcal{R}$ with a partition of the bins to blocks for each MKC, $\bar{c} \in \mathbb{R}^I$ and $\gamma > 0$. The objective is to find $\bar{x} \in P$ such that $\bar{x} \cdot \bar{c}$ is maximized, where $P$ is the $\gamma$-instance polytope of $\mathcal{R}$. While the problem cannot be solved exactly, it admits an FPTAS.

▶ **Lemma 16.** *There is an FPTAS for the instance polytope optimization problem.*

The lemma follows from known techniques for approximating an exponential size linear program using an approximate separation oracle for the dual program. The full proof appears in [16].

The next lemma asserts that the $\gamma$-instance polytope provides an approximate representation for the instance as a polytope.

▶ **Lemma 17.** *Given an $(N, \xi)$-restricted d-MKCP instance $\mathcal{R}$ with objective function $f$, let $S, (A_t)_{t=1}^{d}$ be an optimal solution for $\mathcal{R}$ and $\gamma > 0$. Then there is $S' \subseteq S$ such that $\mathbb{1}_{S'} \in P$ and $f(S') \geq \left(1 - \frac{N^2 \cdot d}{\xi \cdot \gamma}\right) f(S)$, where $P$ is the $\gamma$-instance polytope of $\mathcal{R}$.*

Lemma 17 is proved constructively by removing the $\gamma$-heavy items assigned to blocks of a single bin in $A_t$, for $1 \leq t \leq d$. The full proof appears in [16].

Recall that $F$ is the multiliear extension of the objective function $f$. Our algorithm finds a vector $\bar{x}$ in the instance polytope for which $F(x)$ approximates the optimum. The fractional solution $\bar{x}$ is then rounded to an integral solution. Initially, a random set $R \in \mathcal{I}$ is sampled, with $\Pr(i \in R) = (1-\delta)^2 \bar{x}_i$.[9] The technique by which $R$ is sampled depends on $\mathcal{I}$. If $\mathcal{I} = 2^I$ then $R$ is sampled according to $\bar{x}$, i.e., $R \sim (1-\delta)^2 \bar{x}$ (as defined in Section 1.3). If $\mathcal{I}$ is a matroid constraint, the sampling of [9] is used. Finally, if $\mathcal{I}$ is a matroid intersection, or a matching constraint, then the dependent rounding technique of [10] is used. Each of the distributions admits a Chernoff-like concentration bound. These bounds are central to our proof of correctness. We refer to the above operation as sampling $R$ by $\bar{x}$, $\delta$ and $\mathcal{I}$.

Given the set $R$, the algorithm proceeds to a *purging* step. While this step does not affect the content of $R$ if $f$ is monotone, it is critical in the non-monotone case. Given a submodular function $f : 2^I \to \mathbb{R}$, we define a purging function $\eta_f : 2^I \to 2^I$ as follows. Fix an arbitrary order over $I$ (which is independent of $S$), initialize $J = \emptyset$ and iterate over the items in $S$ by their order in $I$. For an item $i \in S$, if $f(J \cup \{i\}) - f(J) \geq 0$ then $J \leftarrow J \cup \{i\}$; else, continue to the next item. Now, $\eta_f(S) = J$, where $J$ is the set at the end of the process. The purging function was introduced in [11] and is used here similarly in conjunction with the FKG inequality.

While the above sampling and purging steps can be used to select a set of items for the solution, they do not determine how these items are assigned to the bins. We now show that it suffices to associate the selected items with blocks and then use a Bin Packing algorithm for finding their assignment to the bins in the blocks, as in Algorithm 1.

Intuitively, we would like to associate a subset of items $I_j^t$ with a block $K_j^t$ in a way that enables to assign the items in $I_j^t \cap R$ to $|K_j^t|$ bins, for $1 \leq t \leq d$ and $1 \leq j \leq \ell_t$. Consider two cases. If $|K_j^t| > 1$ then we ensure $I_j^t \cap R$ satisfies conditions that allow using Fractional

[9] Recall that $\mathcal{I}$ is the additional constraint.

Grouping (see Lemma 10). On the other hand, if $|K_j^t| = 1$, it suffices to require that $R \cap I_j^t$ adheres to the capacity constraint of this bin. Such a partition $(I_j^t)_{j=0}^{\ell_t}$ of $\text{supp}(\bar{x})$ can be computed for each of the MKCs. We refer to this partition as the *Block Association* of a point in the $\gamma$-partition polytope and $\mu$, on which the partition depends. The formal definition of block association and its properties can be found in [16].

We proceed to analyze our algorithm (see the pseudocode in Algorithm 2).

**Algorithm 2** $(N, \xi)$-restricted $d$-MKCP.

**Input:** An $(N, \xi)$-restricted $d$-MKCP instance $\mathcal{R}$ defined by $\left(I, (w_t, B_t, W_t)_{t=1}^d, \mathcal{I}, f\right)$ and $(K_j^t)_{j=0}^{\ell_t}$, the $N$-leveled partition of $B_t$ for $1 \leq t \leq d$.

**Configuration:** $\gamma > 0$, $\delta > 0$, $N \in \mathbb{N}$, $\xi \in \mathbb{N}$,

1. Optimize $F(\bar{x})$ with $\bar{x} \in P$, where $P$ is the $\gamma$-instance polytope of $\mathcal{R}$, and $F$ is the multilinear extension of $f$.
2. Let $R$ be a random set sampled by $\bar{x}$, $\delta$ and $\mathcal{I}$. Define $J = \eta_f(R)$ ($\eta_f$ is the purging function).
3. Let $\bar{y}^{t,0}, \ldots, \bar{y}^{t,\ell_t} \in [0,1]^I$ such that $(\bar{x}, \bar{y}^{t,0}, \ldots, \bar{y}^{t,\ell_t}) \in P_t^e$, where $P_t^e$ is the extended $\gamma$-partition polytope of $(w_t, B_t, W_t)$ and the partition $(K_j^t)_{j=0}^{\ell_t}$, for $1 \leq t \leq d$.
4. Find the block association $(I_j^t)_{j=0}^{\ell_t}$ of $(1-\delta)(\bar{x}, \bar{y}^{t,0}, \ldots, \bar{y}^{t,\ell_t})$ and $\mu = \frac{\delta}{4}$ for $1 \leq t \leq d$.
5. Pack the items of $J \cap I_j^t$ into the bins of $K_j^t$ using an algorithm for bin packing if $|K_j^t| > 1$, or simply assign $J \cap I_j^t$ to $K_j^t$ otherwise .
6. Return $J$ and the resulting assignment if the previous step succeeded; otherwise, return an empty set and an empty packing.

▶ **Lemma 18.** *For any $d \in \mathbb{N}$, $\varepsilon > 0$ and $M > 0$, there are parameters $N \in \mathbb{N}$ satisfying $N > M$, $\xi \in \mathbb{N}$, $\gamma > 0$ and $\delta > 0$ such that Algorithm 2 is a randomized $(c - \varepsilon)$-approximation for $(N, \xi)$-restricted d-MKCP, where $c = 1$ for modular instances with any type of additional constraint, $c = 1 - e^{-1}$ for monotone instances with a matroid constraint, and $c = 0.385$ for non-monotone instances with no additional constraint.*

A formal proof of the lemma appears in [16]. Theorem 1 follows immediately from Lemmas 18 and 14.

## References

1. Alexander A Ageev and Maxim I Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.
2. Evripidis Bampis, Bruno Escoffier, and Alexandre Teiller. Multistage Knapsack. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, pages 22:1–22:14, 2019.
3. Nikhil Bansal, Marek Eliáš, and Arindam Khan. Improved approximation for vector bin packing. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1561–1579, 2016.
4. Niv Buchbinder and Moran Feldman. Constrained submodular maximization via a nonsymmetric technique. *Mathematics of Operations Research*, 44(3):988–1005, 2019.

**5** Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 1433–1452. SIAM, 2014.

**6** Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Proceedings of Integer Programming and Combinatorial Optimization (IPCO)*, pages 182–196, 2007.

**7** Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. `doi:10.1137/080733991`.

**8** Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 35(3):713–728, 2005. `doi:10.1137/S0097539700382820`.

**9** Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS)*, page 575–584, 2010.

**10** Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pages 1080–1097, 2011.

**11** Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.

**12** Reuven Cohen and Guy Grebla. Efficient allocation of periodic feedback channels in broadband wireless networks. *IEEE/ACM Transactions on Networking*, 23(2):426–436, 2014.

**13** W Fernandez De La Vega and George S. Lueker. Bin packing can be solved within 1+ $\varepsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

**14** Yaron Fairstein, Ariel Kulik, Joseph Naor, and Danny Raz. General knapsack problems in a dynamic setting. To appear in APPROX, 2021.

**15** Yaron Fairstein, Ariel Kulik, Joseph (Seffi) Naor, Danny Raz, and Hadas Shachnai. A $(1-e^{-1}-\varepsilon)$-approximation for the monotone submodular multiple knapsack problem. In *28th Annual European Symposium on Algorithms (ESA 2020)*, pages 44:1–44:19, 2020.

**16** Yaron Fairstein, Ariel Kulik, and Hadas Shachnai. Modular and submodular optimization with multiple knapsack constraints via fractional grouping, 2020. `arXiv:2007.10470`.

**17** Yaron Fairstein, Ariel Kulik, and Hadas Shachnai. Tight approximations for modular and submodular optimization with $d$-resource multiple knapsack constraints, 2020. (second version). `arXiv:2007.10470v2`.

**18** U. Feige and M. Goemans. Approximating the value of two power proof systems, with applications to max 2sat and max dicut. In *Proceedings of the 3rd Israel Symposium on the Theory of Computing Systems*, ISTCS '95, pages 182–189, 1995.

**19** Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

**20** Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 570–579, 2011.

**21** Lisa Fleischer, Michel X Goemans, Vahab S Mirrokni, and Maxim Sviridenko. Tight approximation algorithms for maximum separable assignment problems. *Mathematics of Operations Research*, 36(3):416–431, 2011.

**22** Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pages 1098–1116, 2011.

**23** Klaus Jansen. Parameterized approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 39(4):1392–1412, 2010.

**24** Klaus Jansen. A fast approximation scheme for the multiple knapsack problem. In *SOFSEM'12*, pages 313–324, 2012. `doi:10.1007/978-3-642-27660-6_26`.

25 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, 1982.

26 Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. Springer, 2004.

27 Samir Khuller, Anna Moss, and Joseph Seffi Naor. The budgeted maximum coverage problem. *Information processing letters*, 70(1):39–45, 1999.

28 Ariel Kulik and Hadas Shachnai. There is no EPTAS for two-dimensional knapsack. *Information Processing Letters*, 110(16):707–710, 2010.

29 Ariel Kulik, Hadas Shachnai, and Tami Tamir. Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Math. Oper. Res.*, 38(4):729–739, 2013. doi:10.1287/moor.2013.0592.

30 Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4):2053–2078, 2010.

31 S. Martello and P. Toth. Knapsack problems: algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimiza tion*, 1990.

32 G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

33 Thomas Rothvoß. Approximating bin packing within o (log opt* log log opt) bins. In *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 20–29, 2013.

34 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

35 Xiaoming Sun, Jialin Zhang, and Zhijie Zhang. Tight algorithms for the submodular multiple knapsack problem. *arXiv preprint arXiv:2003.11450*, 2020.

36 Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

37 Deepak S Turaga, Krishna Ratakonda, and Junwen Lai. QoS support for media broadcast in a services oriented architecture. In *2006 IEEE International Conference on Services Computing (SCC'06)*, pages 127–134, 2006.

38 Hien Nguyen Van, Frédéric Dang Tran, and Jean-Marc Menaud. Performance and power management for cloud infrastructures. In *IEEE 3rd international Conference on Cloud Computing (CLOUD)*, pages 329–336. IEEE, 2010.

39 Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

40 Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.

41 Quanqing Xu, Khin Mi Mi Aung, Yongqing Zhu, and Khai Leong Yong. A blockchain-based storage system for data analytics in the internet of things. In *New Advances in the Internet of Things*, pages 119–138. Springer, 2018.