Modular and Submodular Optimization with Multiple Knapsack Constraints via Fractional Grouping

Yaron Fairstein* Ariel Kulik† Hadas Shachnai‡

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.

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 \([34, 29, 41, 45]\). 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 \([9, 26, 27]\). 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 \([28, 14]\). 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.

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., \([14, 28]\)). 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.

*Computer Science Department, Technion, Haifa 3200003, Israel. E-mail: yyfairstein@gmail.com
†Computer Science Department, Technion, Haifa 3200003, Israel. E-mail: kulik@cs.technion.ac.il
‡Computer Science Department, Technion, Haifa 3200003, Israel. E-mail: hadas@cs.technion.ac.il.
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 $(1 - \varepsilon^{-1} - o(1))$-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 $A$ 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 $f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T)$. Submodular functions naturally arise in numerous settings. While many submodular functions, such as coverage \cite{18} and matroid rank function \cite{7}, are monotone, i.e., for any $S \subseteq T \subseteq I$, $f(S) \leq f(T)$, this is not always the case (cut functions \cite{17} 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 $f(S \cup \{i\}) - f(S) = f(T \cup \{i\}) - f(T)$.

The problem of Submodular Maximization with $d$-Multiple Knapsack Constraints ($d$-MKC) is defined as follows. The input is $I = (I, (K_t)_{t=1}^d, \mathcal{I}, f)$, where $I$ is a set of items, $K_t$, $1 \leq t \leq d$ 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 \cite{3} or (iii) $\mathcal{I}$ is the intersection of independent sets of two matroids, or (iv) $\mathcal{I}$ is a matching \cite{2}. 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 $K_t$ 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.

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 Appendix \cite{A}).

1.2 Our Results

Our main results are summarized in the next theorem (see also Table \cite{I}).

Theorem 1.1. 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 $(1 - \varepsilon^{-1} - \varepsilon)$-approximation for monotone $d$-MKCP with a matroid constraint.

---

1Alternatively, for every $S, T \subseteq I$: $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$.

2A formal definition for matroid can be found in \cite{17}.

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 1.1 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                 | –                        | –                     |

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. The problem has natural applications in network optimization [13, 40]. The best known approximation ratio for the problem is 2 due to [13]. This approximation ratio is improved by Theorem 1.1, as stated in the following.

**Theorem 1.2.** 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 [9, 26, 27, 16, 38], 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 1.1.

**Theorem 1.3.** For any $\varepsilon > 0$ there is a randomized $(0.385 - \varepsilon)$-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 $\Theta(\frac{1}{\varepsilon})$-MKCP with a matroid constraint (see [15] 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 ($\mathcal{I} = 2^d$ 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 [31]. While there is a well-known PTAS for Multiple Knapsack [2], 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 [35]. The approximation ratio is also tight under $P \neq NP$ due to the special case of coverage functions [18]. Previous works [16, 38] 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 [5]. This approximation ratio matches the current best known ratio for non-monotone submodular maximization with a single knapsack constraint [5]. A 0.491 hardness of approximation bound for non-monotone d-MKCP follows from [22].

---

4That 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 $I$.

5See, e.g., [3] for the Multistage Knapsack model.
The Monotone Submodular Multiple Knapsack Problem with Uniform Capacities (USMKP) is the special case of $d$-MKCP in which $I = 2^{I}$, $d = 1$, $f$ 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 $\varepsilon > 0$ [16, 38]. Another contribution of this paper is an improvement of this ratio.

**Theorem 1.4.** There is a polynomial-time $(1 - e^{-1} - O((\log |B|)^{-\frac{1}{4}}))$-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 [9]. The authors also ruled out the existence of a fully polynomial time approximation scheme (FPTAS). An EPTAS was later developed by Jansen [26, 27].

In the Bin Packing problem, we are given a set $I$ of items, a weight function $w : I \to \mathbb{R}_{\geq 0}$ 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 $\sum_{i \in S_k} w(i) \leq W$ 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 $\text{OPT}$ is the number of bins in a minimal packing. The result was later improved by Rothvoß [30].

Research work on monotone submodular maximization dates back to the late 1970’s. In [35] 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 [35] was later generalized to monotone submodular maximization subject to a knapsack constraint [30, 39].

A major breakthrough in the field of submodular optimization resulted from the introduction of algorithms for optimizing the multilinear extension of a submodular function ([17, 33, 8, 44, 19, 6]). 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 $f : 2^I \to \mathbb{R}_{\geq 0}$, its multilinear extension is $F : [0, 1]^I \to \mathbb{R}_{\geq 0}$ defined as $F(\bar{x}) = \mathbb{E}_{S \sim \bar{x}}[f(S)]$.

The input for the multilinear optimization problem is an oracle for a submodular function $f : 2^I \to \mathbb{R}_{\geq 0}$ and a downward closed solvable polytope $P$. 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 $(1 - e^{-1} - o(1))$-approximation in the monotone case and a $(0.385 + \delta)$-approximation in the non-monotone case (for some small constant $\delta > 0$) due to [8] and [5].

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 Rounding [10], and Contention Resolution Schemes [12]. These techniques led to the state of art results for many problems (e.g., [32, 8, 1, 10]).

A random $(1 - e^{-1} - \varepsilon)$-approximation for the Monotone Submodular Multiple Knapsack problem was presented in [16]. The technique in [16] modifies the objective function and its domain. This modification does not preserve submodularity of a non-monotone 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. [33]. Their algorithm relies on a variant of the submodular greedy of [70] which cannot be extended to the non-monotone case, or easily adapted to handle more than a single MKC.

---

*\[A\] 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 $\lambda \in \mathbb{R}^I$, a point $\bar{x} \in P$ such that $\lambda \cdot \bar{x} = \max_{\bar{y} \in P} \lambda \cdot \bar{y}$ can be computed in polynomial time, where $\lambda \cdot \bar{x}$ is the dot product of $\lambda$ and $\bar{x}$.\[\]
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-MKC, 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.

Fix an arbitrary small \(\mu > 0\). 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 [14] 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 \(\mu^2\) and \(\mu^2 \cdot h^*\) are integers). Specifically, \(S^* \cap H = G_1^* \cup \ldots \cup G_{\mu^2-2}^*\), 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 \(G_1, \ldots, G_{\mu^2-2}\) where \(G_k = \{i \in H \mid w(q_{k+1}) < w(i) \leq w(q_k)\}\) for \(1 \leq k < \mu^2\), and \(G_{\mu^2-2} = \{i \in H \mid w(i) \leq w(q_{\mu^2-2})\}\). 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. The items in \(S \cap G_k\) can be packed in place in 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 \(q_1, \ldots, q_{\mu^2-2}\) and \(h^*\); thus, the sets \(G_1, \ldots, G_{\mu^2-2}\) 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 [32, 5]. 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 \(q_1, \ldots, q_{\mu^2-2}\) and \(h^*\). This prevents us from using the above approach. However, as in [14], we can overcome this difficulty through exhaustive enumeration. Each of \(q_1, \ldots, q_{\mu^2-2}\) and \(h^*\) takes one of \(|I|\) possible values. Thus, by iterating over all \(|I|^{1+\mu^2}\) possible values for \(q_1, \ldots, q_{\mu^2-2}\) 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 [16] 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 \(|I|^{9(\log |B|)}\) 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 \(G_1^\tau, \ldots, G_\tau^\tau\) 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 [27, 20, 4]. 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, \(\sum_{i \in G_k^\tau} \bar{y}_i \approx \mu^2 |B|\) for all \(1 \leq k \leq \tau\). Furthermore, we show that if \(\bar{y}\) is strictly contained in \(P\) then any subset \(S \subseteq I\) satisfying (i) \(|S \cap G_k^\tau| \leq \mu|B|\) 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

\footnote{For a set \(S \subseteq I\) we denote \(w(S) = \sum_{i \in S} w(i)\).}
while avoiding enumeration.

This suggests the following algorithm. Use the algorithm of \cite{5} 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 $\mathbb{E}[|R \cap G_k|] \ll \mu|B|$. Thus, $R$ can be packed into $B$ using a bin packing algorithm. Standard submodular bounds also guarantee 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 \ref{thm:main}. This scaling involves use of existing techniques for submodular optimization (\cite{10, 11, 8, 5}), 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 is presented in Appendix C.

## 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 2.1.** The extended block polytope of $K$ is

$$P^e_K = \left\{ \bar{y} \in [0, 1]^I, \bar{z} \in [0, 1]^{\mathcal{C}_K} \left| \begin{array}{r} \sum_{C \in \mathcal{C}_K} z_C \bar{y}_i \leq |K| \\ \bar{y}_i \leq \sum_{C \in \mathcal{C}_K s.t. i \in C} z_C \end{array} \right. \right\}. \tag{1}$$

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 2.2.** The block polytope of $K$ is

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

While $P^e_K$ and $P_K$ are defined by an exponential number of variables, it follows from standard arguments (see, e.g., \cite{20, 28}) that, for any $\bar{c} \in \mathbb{R}^I$, $\max_{\bar{y} \in P_K} \bar{c} \cdot \bar{y}$ can be approximated.

**Lemma 2.3.** 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 maximal, 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 is given in Appendix D as a special case of Lemma 4.4. We say that $A : K \rightarrow 2^I$ is a feasible assignment for $K$ if $w(A(b)) \leq W^*_K$ for any $b \in K$. Also, we use $1_S = \bar{x} \in \{0, 1\}^I$, 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 $P^e_K$ is sound for the problem.

**Lemma 2.4.** Let $A$ be a feasible assignment for $K$ and $S = \bigcup_{b \in K} A(b)$. Then $1_S \in P_K$.

The lemma is easily proved, by setting $z_C = 1$ if $A(b) = C$ for some $b \in B$, and $z_C = 0$ otherwise. We say an item $i \in I$ is $\mu$-heavy for $\mu > 0$ (w.r.t $K$) if $W^*_K \geq w(i) > \mu \cdot W^*_K$; otherwise, $i \in I$ is $\mu$-light. 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 2.5.** Let \( \bar{y} \in P_K \) and \( \mu \in (0, \frac{1}{2}] \). 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.,

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

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 \( 1 \leq k \leq \tau \).

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 2.6 (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 2.5 that \( G_1, \ldots, G_\tau \) can be computed in polynomial time. Also, \( \sum_{i \in G_k} \bar{y}_i < \mu \cdot |K| \) and

\[
\forall 1 \leq k < \tau : \quad \mu \cdot |K| \leq \sum_{i \in G_k} \bar{y}_i \leq \mu \cdot |K| + 1.
\] (3)

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

Define the type of a configuration \( C \in C_K \), denoted by \( \text{type}(C) \), as the vector \( T \in \mathbb{N}^\tau \) with \( T_k = |C \cap G_k| \). Let \( T = \{\text{type}(C) \mid C \in C_K\} \) 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 \( C \in C_K \) (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 2.4 there exists \( \bar{z} \in [0, 1]^C \) such that \((\bar{y}, \bar{z}) \in (1 - \delta)P_K^c\). For \( T \in \mathcal{T} \), let \( \eta(T) = \sum_{C \in C_K} \text{type}(C) = T \cdot \bar{z}_C \). Then, for any \( 1 \leq k \leq \tau - 1 \), we have

\[
\mu |K| \leq \sum_{i \in G_k} \bar{y}_i \leq \sum_{i \in G_k} \sum_{C \in C_K} \sum_{i \in C} \bar{z}_C = \sum_{C \in C_K} |G_k \cap C| \bar{z}_C = \sum_{T \in \mathcal{T}} T_k \cdot \eta(T)
\] (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 \( \eta = \sum_{T \in \mathcal{T}} [\eta(T)] \) 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 $\lceil n(T) \rceil$ sets associated with the type $T \in T$, using a function $R$. That is, let $R : \{1, 2, \ldots, \eta\} \to T$ such that $|R^{-1}(T)| = \lceil n(T) \rceil$. 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 $C \in C_K$ and $T \in T$ 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

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

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

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

We use First-Fit (see, e.g., Chapter 9 in [42]) 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 (3), it holds that $\eta' \leq \max\{\eta, (|K|(1 - \delta) + \lambda)(1 + 2\mu) + 1\}$. Finally,

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

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.

\[\square\]

### 3 Uniform Capacities

In this section we apply fractional grouping (as stated in Lemma 2.2) 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 [11, 7]. Given a (fractional) solution $\bar{x} \in [0, 1]^I$, a submodular function $f$, and two items $i_1, i_2 \in I$ with costs $c_1, c_2$, let $F$ be the multilinear extension of $f$. Pipage Rounding returns a new random solution $x' \in [0, 1]^I$ such that $E[F(x')] \geq F(\bar{x})$, $x'_i = \bar{x}_i$ for $i \in I \setminus \{i_1, i_2\}$, $x_{i_1} \cdot c_1 + x_{i_2} \cdot c_2 = x'_{i_1} \cdot c_1 + x'_{i_2} \cdot c_2$, and either $x'_{i_1} \in \{0, 1\}$ or $x'_{i_2} \in \{0, 1\}$. Algorithm [10] calls the subroutine Pipage$(\bar{x}, f, G, \bar{c})$, which can be implemented by an iterative application of Pipage rounding, as summarized in the next result.
Lemma 3.1. There is a polynomial time procedure Pipage($x, f, G, c$) for which the following holds. Given $x \in [0, 1]^I$, 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 $c \in \mathbb{R}_{\geq 0}^G$, the procedure returns a random vector $x' \in [0, 1]^I$ such that $\mathbb{E}[F(x')] \geq F(x)$, $x'_i \in [0, 1]$ for $i \in G$, $x'_i = x_i$ for all $i \in I \setminus G$, and there is $i^* \in G$ such that $\sum_{i \in G} x'_i \cdot c_i \leq 0 + \sum_{i \in G} x_i \cdot c_i$.

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 $F(y)$ 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 $(1 - 4\mu)\bar{y}$ 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 $\approx (1 - 4\mu) \cdot \sum_{i \in L_{B,\mu}} y_i \cdot w(i)$. An approximation algorithm for bin packing is then used to find a packing of $S$ to the bins. Lemma 2.6 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.

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$_{y \in P_B} F(y)$, 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 $y^0 \leftarrow (1 - 4\mu)\bar{y}$, and let $G_1, \ldots, G_\tau$ be the $\mu$-grouping of $y^0$.
4. for $k = 1, 2, \ldots, \tau$ do $y^k \leftarrow \text{Pipage}(y^{k-1}, f, G_k, \bar{1})$.
5. $y' = \text{Pipage}(y^*, f, L_{B,\mu}, (w(i))_{i \in L_{B,\mu}})$.
6. Let $S = \{i \in I \mid y'_i = 1\}$.
7. Pack the items in $S$ into $B$ using a bin packing algorithm. Return the resulting assignment.

Lemma 3.2. Algorithm 1 yields a $\left(1 - e^{-1} - O\left((\log |B|)^{-\frac{1}{4}}\right)\right)$-approximation for USMKP.

Proof. Let $A^*$ be an optimal solution for the input instance, and $\text{OPT} = f(\bigcup_{b \in B} A^*(b))$ its value. By Lemma 2.4, $1_{\bigcup_{b \in B} A^*(b)} \in P_B$. Let $c = 1 - e^{-1}$. By using the algorithm of 8 we have that $F(y^0) \geq (c - \frac{1}{|B|}) \cdot \text{OPT}$ (where $y^0$ is defined in Step 1 of Algorithm 1). The algorithm of 8 is used with the FPTAS of Lemma 2.3 as an oracle for solving linear optimization problems over $P_B$. We note that this would not affect the approximation guarantee.

Since the multilinear extension has negative second derivatives, it follows that $F(y^0) \geq (1 - 4\mu) \cdot \left(c - \frac{1}{|B|}\right) \cdot \text{OPT}$. Now, consider the vector $y'$ output in Step 5 of the algorithm. By Lemma 3.1, it follows that $\mathbb{E}[F(y')] \geq F(y^0) \geq (1 - 4\mu) \cdot \left(c - \frac{1}{|B|}\right) \cdot \text{OPT}$, and $y' \in \{0, 1\}^I$ (note that $y'_i = y_i = 0$ for any $i$ with $w(i) > W_B$ due to 11). Thus, for the set $S$ defined in Step 5 of the algorithm, we have $\mathbb{E}[f(S)] \geq (1 - 4\mu) \cdot \left(c - \frac{1}{|B|}\right) \cdot \text{OPT} \geq \left(c - O\left((\log |B|)^{-\frac{1}{4}}\right)\right) \cdot \text{OPT}$ (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 3.1, for any $1 \leq k \leq \tau$, it holds that $|S \cap G_k| = \sum_{i \in G_k} y^0_i \leq 1 + \sum_{i \in G_k} y^0_i \leq \mu \cdot |B| + 2$ (the last inequality follows from Lemma 2.6). Similarly, there is $i^* \in L_{B,\mu}$ such that $w(S \cap L_{B,\mu}) = \sum_{i \in L_{B,\mu}} y^0_i \cdot w(i) + \sum_{i \in L_{B,\mu}} y^0_i \cdot w(i) \leq \mu \cdot W_B + \sum_{i \in L_{B,\mu}} y^0_i \cdot w(i)$.

To meet the conditions of Lemma 2.6 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 2.6. 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

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

bins of capacity $W_B$. Recall that the algorithm of [28] returns a packing in at most $\text{OPT} + O((\log^2 |B|)$ bins. Thus, for large enough $|B|$, the number of bins used in Step 7 of Algorithm 4 is at most

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

Finally, we note that Algorithm 4 can be implemented in polynomial time.

\[\square\]

4 Approximation Algorithm

In this section we present our algorithm for general instances of $d$-MKCP, which gives the result in Theorem 1.1. 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 [16].

**Definition 4.1.** 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 $|K_j| = N\lceil \frac{j}{\ell} \rceil$. 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 $\mathcal{R} = \{(I, (w_t, B_t, W_t)_{t=1}^d, I, f)\}$ it holds that $B_t$ and $W_t$ are $N$-leveled for all $1 \leq t \leq d$, and $f(\{i\}) - f(\emptyset) \leq \text{OPT}$ 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 $\{K_j\}_{j=0}^\ell$ of $B_t$ for all $1 \leq t \leq d$. Combining standard enumeration with the structuring technique of [16], we derive the next result, whose proof is given in Appendix [3].

**Lemma 4.2.** 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/matrice intersection/matching/no additional constraint implies a polynomial time $c \cdot (1 - \frac{\xi}{d})$-approximation for $d$-MKCP, with the same type of function and same type of additional constraint.

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 2.2) to represent an MKC. We then use it to define a polytope for the whole instance.

**Definition 4.3.** 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

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

(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

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

(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 \( \left( I, (w_t, B_t, W_t)_{t=1}^d, I, f \right) \) and a partition \( \left( K_j^t \right)_{j=0}^{\ell_t} \) of \( B_t \) to blocks, for \( 1 \leq t \leq d \), is \( P = P(I) \cap \left( \bigcap_{t=1}^d P_t \right) \), where \( P(I) \) is the convex hull of \( I \) 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 \( R \) 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 \( R \). While the problem cannot be solved exactly, it admits an FPTAS.

**Lemma 4.4.** 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. We give the proof in Appendix D.

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

**Lemma 4.5.** Given an \((N, \xi)\)-restricted \( d \)-MKCP instance \( R \) with objective function \( f \), let \( S, (A_t)_{t=1}^d \) be an optimal solution for \( R \) and \( \gamma > 0 \). Then there is \( S' \subseteq S \) such that \( 1_{S'} \in P \) and \( f(S') \geq \left( 1 - \frac{N^2 d}{\xi \gamma} \right) f(S) \), where \( P \) is the \( \gamma \)-instance polytope of \( R \).

Lemma 4.5 is proved constructively by removing the \( \gamma \)-heavy items assigned to blocks of a single bin in \( A_t \), for \( 1 \leq t \leq d \). We give the proof in Appendix E.

Recall that \( F \) is the multilinear extension of the objective function \( f \). Our algorithm finds a vector \( \bar{x} \) in the instance polytope for which \( F(\bar{x}) \) approximates the optimum. The fractional solution \( \bar{x} \) is then rounded to an integral solution. Initially, a random set \( R \in I \) is sampled, with \( \Pr(i \in R) = (1 - \delta)^2 \bar{x}_i \). The technique by which \( R \) is sampled depends on \( I \). If \( I = 2^I \) then \( R \) is sampled according to \( \bar{x} \), i.e., \( R \sim (1 - \delta)^2 \bar{x} \) (as defined in Section 3.3). If \( I \) is a matroid constraint, the sampling of 10 is used. Finally, if \( I \) is a matroid intersection, or a matching constraint, then the dependent rounding technique of 11 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 \( I \).

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 \( f : 2^I \rightarrow \mathbb{R} \), we define a purging function \( \eta_f : 2^I \rightarrow 2^I \) 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 12 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 Grouping (see Lemma 2.6). 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 \( \left( I_j^t \right)_{j=0}^{\ell_t} \) 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. In Appendix C we give a formal definition of block association and its properties.

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

\(^8\) Recall that \( I \) is the additional constraint.
Lemma 4.6. For any $0 < \gamma > d$, 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 is given in Appendix E. Theorem 1.1 follows immediately from Lemmas 4.6 and 4.1.

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

**Input:** An $(N, \xi)$-restricted d-MKCP instance $\mathcal{R}$ defined by $(I, (w_t, B_t, W_t)_{t=1}^d, \mathcal{I}, f)$ and $(K^d_{ij})_{j=0}^d$, the $d$- 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 $\bar{y}^{f,0}, \ldots, \bar{y}^{f,\ell_h} \in [0,1]^I$ such that $(\bar{x}, \bar{y}^{f,0}, \ldots, \bar{y}^{f,\ell_h}) \in P^*_f$, where $P^*_f$ is the extended $\gamma$-partition polytope of $(w_t, B_t, W_t)$ and the partition $(K^d_{ij})_{j=0}^d$, for $1 \leq t \leq d$.
4. Find the block association $(I^d_{ij})_{j=0}^d$ of $(1 - \delta)(\bar{x}, \bar{y}^{f,0}, \ldots, \bar{y}^{f,\ell_h})$ and $\mu = \frac{\delta}{2}$ for $1 \leq t \leq d$.
5. Pack the items of $J \cap I^d_j$ into the bins of $K^d_j$ using an algorithm for bin packing if $|K^d_j| > 1$, or simply assign $J \cap I^d_j$ to $K^d_j$ otherwise.
6. Return $J$ and the resulting assignment if the previous step succeeded; otherwise, return an empty set and an empty packing.

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] Noga Alon and Joel H Spencer. The probabilistic method. John Wiley & Sons, 2004.

[3] 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.

[4] 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, pages 1561–1579. SIAM, 2016.

[5] Niv Buchbinder and Moran Feldman. Constrained submodular maximization via a nonsymmetric technique. Mathematics of Operations Research, 44(3):988–1005, 2019.

[6] 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, pages 1433–1452. SIAM, 2014.

[7] Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In Integer Programming and Combinatorial Optimization, 12th International IPCO Conference, Ithaca, NY, USA, June 25-27, 2007, Proceedings, pages 182–196, 2007.

[8] 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.
[9] Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 35(3):713–728, 2005.

[10] Chandra Chekuri, Jan Vondrak, 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*, page 575–584, 2010.

[11] 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*, pages 1080–1097, SIAM, 2011.

[12] 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.

[13] 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.

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

[15] Yaron Fairstein, Ariel Kulik, Joseph, Naor, and Danny Raz. General knapsack problems in a dynamic setting. *arXiv preprint arXiv:2105.00882*, 2021.

[16] Yaron Fairstein, Ariel Kulik, Joseph (Seffi) Naor, Danny Raz, and Hadas Shachnai. A (1−e−1− ε)-approximation for the monotone submodular multiple knapsack problem. In *28th Annual European Symposium on Algorithms (ESA 2020)*, pages 44:1–44:19, 2020.

[17] 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.

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

[19] 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*, pages 570–579. IEEE, 2011.

[20] 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.

[21] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *J. ACM*, 53(3):324–360, May 2006.

[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*, pages 1098–1116. SIAM, 2011.

[23] M Grötschel, L Lovász, and A Schrijver. Geometric algorithms and combinatorial optimization. *Algorithmics and Combinatorics*, 1993.

[24] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Comb.*, 1(2):169–197, 1981.

[25] Oscar H Ibarra and Chul E Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM (JACM)*, 22(4):463–468, 1975.

[26] Klaus Jansen. Parameterized approximation scheme for the multiple knapsack problem. *SIAM J. Comput.*, 39(4):1392–1412, 2010.
[27] Klaus Jansen. A fast approximation scheme for the multiple knapsack problem. In *SOFSEM’12*, pages 313–324, 2012.

[28] 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.

[29] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. Springer, 2004.

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

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

[32] 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.

[33] 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.

[34] S. Martello and P. Toth. Knapsack problems: algorithms and computer implementations. *Wiley-Interscience series in discrete mathematics and optimization*, 1990.

[35] 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.

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

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

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

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

[40] 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. IEEE, 2006.

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

[42] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[43] Jan Vondrak. A note on concentration of submodular functions, 2010.

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

[45] 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.
A Application of $d$-MKCP in Cloud Data Centers

In the Cross Region Cloud Provider problem, a cloud provider maintains two data centers, located in different geographical regions (e.g., an American and a European data center). Each of the data centers contains servers of varying computing capacities. Additionally, there is a collection of applications which the provider may deploy on its cloud. Each application requires some compute capacity in each region. As the demand for an application differs between the regions, the compute requirement of an application may differ significantly between the regions. Each application also has a specified profit the provider gains if the application is deployed.

The cloud provider needs to select a subset of applications to be deployed, and assign each of these applications to a server in each of the regions. The assignment of applications to servers must preserve the compute capacities of the servers. The provider’s goal is to maximize the total profit obtained from the selected applications. The Cross Region Cloud Provider problem can be easily cast as an instance of 2-MKCP with no additional constraint.

B Reduction to Restricted Instances

In this section we prove Lemma 4.2. Let $T = \left(I, (K_t)^{d}_{t=1}, I, f\right)$ be a $d$-MKCP instance, where $K_t = (w_t, B_t, W_t)$, $1 \leq t \leq d$ are the $d$ multiple knapsack constraints. Also, let $S$ and $(A_t)^{d}_{t=1}$ be some solution for $T$, and let $\xi \in \mathbb{N}$. We define the residual instance of $T, S, (A_t)^{d}_{t=1}$ and $\xi$ as the valid $d$-MKCP instance $T' = \left(I', (K'_t)^{d}_{t=1}, I', g\right)$, where

1. $I' = \left\{i \in I \setminus S \mid f(\{i\} \cup S) - f(S) \leq f(S)\right\}$.

2. The function $g : 2^{I'} \rightarrow \mathbb{R}_{\geq 0}$ is defined by $g(T) = f(S \cup T)$.

3. For any $1 \leq t \leq d$, $K'_t = (w_t, B_t, W'_t)$, where $W'_t(b) = W_t(b) - w_t(A_t(b))$ for all $b \in B_t$.

4. $I' = \{T \subseteq I' \mid T \cap S \subseteq I\}$.

We note that the function $g$ is non-negative and submodular. Furthermore (as shown in Lemma 4.2), $g$ is monotone (modular) if $f$ is monotone (modular). Also, if $I$ describes a constraint (matroid, matroid intersection or matching constraint), $I'$ describes a constraint of the same type. If $T \subseteq I'$ and its assignment $(D_t)^{d}_{t=1}$ define a solution for the residual instance $T'$, then it can be easily verified that $S \cup T$ and $(A_t \cup D_t)^{d}_{t=1}$ form a solution for the original instance $T$.

Furthermore, by definition, $f(S \cup T) = g(T)$.

Residual instances are useful as the marginal value $g(\{i\}) - g(\emptyset)$ of every item $i \in I'$ is bounded.

The next lemma states that the residual instance with a specific solution preserves the optimum.

**Lemma B.1.** Let $\xi \in \mathbb{N}$, $T$ be an instance of $d$-MKCP, and $S^*, (A_t^*)^{d}_{t=1}$ a solution for $T$. Then there is $S \subseteq S^*$, $|S| \leq \xi$ such that $S^* \setminus S$, $(A_t^* \setminus S)^{d}_{t=1}$ is a solution for the residual instance of $T$, $S$, $(A_t^* \cap S)^{d}_{t=1}$ and $\xi$.

**Proof.** Let $T = \left(I, (w_t, B_t, W_t)^{d}_{t=1}, I, f\right)$. Also, let $S^* = \{s^*_1, \ldots, s^*_\ell\}$ with the items sorted such that $f(\{s^*_1, \ldots, s^*_\ell\}) = \max_{1 \leq i \leq \ell} f(\{s^*_1, \ldots, s^*_{\ell-1}\} \cup \{s^*_\ell\})$ for every $1 \leq i \leq \ell$. If $\ell \leq \xi$ let $S = S^*$, and the lemma immediately follows. Otherwise, set $S = \{s^*_1, \ldots, s^*_\ell\}$.

Let $T' = \left(I', (w_t, B_t, W'_t)^{d}_{t=1}, I', f\right)$ be the residual instance of $T$, $S$, $(A_t^* \cap S)^{d}_{t=1}$ and $\xi$. By well known properties of submodular functions (see Lemma 4.1), for every $i \in S^* \setminus S$ it holds that $f(S \cup \{i\}) - f(S) \leq \frac{f(S)}{\xi}$, and therefore $S^* \setminus S \subseteq I'$. Furthermore, it can be easily verified that $S^* \setminus S \subseteq I'$ and $A_t^* \setminus S$ is a feasible assignment of $S^* \setminus S$ w.r.t $(w_t, B_t, W'_t)$; thus, $S^* \setminus S$ and $(A_t^* \setminus S)^{d}_{t=1}$ is a solution for the residual instance $T'$.

\[\xi\]

*Given two assignments $A_1, A_2 : B \rightarrow 2^I$, we define $D = A_1 \cup A_2$ by $D(b) = A_1(b) \cup A_2(b)$ for all $b \in B$.

\[\xi\]

*Given an assignment $A : B \rightarrow 2^I$ and a set $S \subseteq I$, we define $A \cap S (A \setminus S)$ to be the assignment $D : B \rightarrow 2^I$ such that $D(b) = A(b) \cap S (D(b) = A(b) \setminus S)$ for every $b \in B$.  

15
The enumeration (Step 2) of Algorithm 3 iterates over all subsets \( S' \subseteq I \), \( |S'| \leq \xi \) and disjoint assignments \( (A_t)_{t=1}^d \) of \( S' \) to the \( d \) MKCs, and finds an approximate solution for each residual instance. The number of possible subsets and assignments is polynomial for a fixed \( \xi \). In one of these iterations, \( S' = S \) and \( A_t = A^*_t \cap S \) for every \( 1 \leq t \leq d \) (\( S \) is the set from Lemma B.1). In this specific iteration, the value of the optimal solution for the residual instance equals to the value of an optimal solution for the original instance; thus, the approximate solution for the residual instance can be used to derive an approximate solution for the input instance.

Algorithm 3 converts each of the \( d \) MKCs into an \( N\)-leveled constraint. The following lemma, adapted from \cite{10}, shows that any MKC can be converted into an \( N\)-leveled constraint with only a small decrease in the value of an optimal solution.

**Lemma B.2.** For any \( N \), set of bins \( B \) and capacities \( W : B \rightarrow \mathbb{R}_{\geq 0} \), there is \( \tilde{B} \subseteq B \), capacities \( \tilde{W} : \tilde{B} \rightarrow \mathbb{R}_{\geq 0} \), and an \( N\)-leveled partition \( (\tilde{K}_j)_{j=0}^f \) of \( \tilde{B} \), such that

1. \( \tilde{B}, \tilde{W} \) and \( (\tilde{K}_j)_{j=0}^f \) can be computed in polynomial time.
2. The bin capacities satisfy \( \tilde{W}(b) \leq W(b) \), for every \( b \in \tilde{B} \).
3. For any set of items \( I \), weight function \( w : I \rightarrow \mathbb{R}_{\geq 0} \), a subset \( S \subseteq I \) feasible for the MKC \((w, B, W)\) over \( I \), and a submodular function \( f : 2^I \rightarrow \mathbb{R}_{\geq 0} \), there is \( \tilde{S} \subseteq S \) feasible for the MKC \((\tilde{w}, \tilde{B}, \tilde{W})\) such that \( f(\tilde{S}) \geq (1 - \frac{1}{N})f(S) \). \[12\]

We refer to \( \tilde{B} \) and \( \tilde{W} \) as the \( N\)-leveled constraint of \( B \) and \( W \).

The proof of the lemma requires a minor adaptation of the proof in \cite{10}. We now proceed to the analysis of Algorithm 3.

**Lemma B.3.** For any \( N, \xi, d \in \mathbb{N} \) and \( c \in [0, 1] \), if \( A \) is a polynomial-time random \( c\)-approximation for modular/ monotone/ non-monotone \((N, \xi)\)-restricted \( d\)-MKCP with a matroid/ matroid intersection/ matching/ no additional constraint then Algorithm 3 configured with \( A \) is a polynomial-time random \( c \cdot (1 - \frac{1}{N})\)-approximation for \( d\)-MKCP with the same type of function and additional constraint.

**Proof.** Let \( T = (\{i,(w_i,B_i,W_i)\}_{i=1}^d,I,f) \) be \( d\)-MKCP instance where \( f \) and \( I \) matches the type of functions and additional constraint solved by \( A \). Also, let \( S^* \) and \( (A^*_t)_{t=1}^d \) be an optimal solution for \( T \), and \( OPT = f(S^*) \). W.l.o.g., we assume that \( A^*_t \) is a disjoint assignment for any \( 1 \leq t \leq d \).

By Lemma B.1, there is \( S \subseteq S^* \), \( |S| \leq \xi \) such that \( S^* \cap S \) and \( (A^*_t \cap S)_{t=1}^d \) is a solution for \( T' \), the residual problem of \( T \), \( S \), \( (A^*_t \cap S)_{t=1}^d \) and \( \xi \). We focus in the analysis on the iteration of the loop in Step 2 in which \( S \) is the set defined above, and \( A_t = A^*_t \cap S \) for any \( 1 \leq t \leq d \). It holds that \( T' \) is a \((N,\xi)\)-restricted \( d\)-MKCP instance with the type of function and additional constraint handled by \( A \). Its constraints are \( N\)-leveled due to the leveling. Furthermore, \( g(\{i\}) - g(\emptyset) = f(S \cup \{i\}) - f(S) \leq \frac{f(S)}{\xi} \leq \frac{OPT}{\xi} \) for any \( i \in I' \) (note that \( S \) and \( (A^*_t \cap S)_{t=1}^d \) form a solution for \( T \), thus \( f(S) \leq OPT \)). Finally, it can be easily verified that the optimum of \( T' \) is at most \( OPT \). We conclude that \( A \) returns a \( c\)-approximation for \( T' \).

By iterative application of Lemma B.2 it follows that there is \( \tilde{S} \subseteq S^* \cap S \subseteq I' \) such that \( \tilde{S} \) is feasible for \((w_i, B_i, W_i)\) for any \( 1 \leq t \leq d \) and \( g(\tilde{S}) \geq (1 - \frac{1}{N})g(S^* \cap S) = (1 - \frac{1}{N})OPT \). Thus, there are assignments with which \( S \) is a feasible solution for \( T \). We conclude that \( \mathbb{E}[g(R)] \geq c \cdot (1 - \frac{1}{N})OPT \) (the randomization stems from \( A \) being possibly random).

Note that \( T^* \) is maintained by the algorithm as the solution of highest value attained so far. Thus, subsequent to the specific iteration it holds that \( \mathbb{E}[f(T^*)] \geq \mathbb{E}[f(S \cup R)] = \mathbb{E}[g(R)] \geq (1 - \frac{1}{N})OPT \). We note that the algorithm has a polynomial running time, as the number of iterations is polynomial for a constant \( \xi \), and that the returned solution is always feasible.

\[\square\]

\footnotetext{11}{An assignment \( A : B \rightarrow 2^I \) is disjoint if for any \( b_1, b_2 \in B, b_1 \neq b_2 \), it holds that \( A(b_1) \cap A(b_2) = \emptyset \).}

\footnotetext{12}{We say \( S \) is feasible for a MKC \((w, B, W)\) if there is a feasible assignment \( A \) of \( S \) w.r.t \((w, B, W)\).}
Lemma C.1
In this section we present our block association technique, formally summarized in the next lemma.

C Block Association

Let \((\bar{y}, \bar{y}^0, \ldots, \bar{y}^r)\) be an MKC, and \((K_j)_{j=0}^r\) a partition of \(B\) into blocks. Suppose that \((\bar{x}, \bar{y}^0, \ldots, \bar{y}^r)\) is a perfect \((\bar{x}, \bar{y}^0, \ldots, \bar{y}^r)\) assignment, such that \((\bar{x}, \bar{y}^0, \ldots, \bar{y}^r)\) satisfies the following conditions for all \(0 \leq j \leq r\):

1. If \(|K_j| > 1\), let \(G_1, \ldots, G_{\tau_j}\) be the \(\mu\)-grouping of \(\bar{y}^j\). Then \(\sum_{i \in I_j \cap G_{\tau_j}} \bar{x}_i \leq \mu|K_j| + 2\) for all \(1 \leq k \leq \tau_j\), and there is \(i^*_j \in I\) such that \(\sum_{i \in I_j \cap L_{K_j} \setminus G_{\tau_j}} \bar{x}_i w(i) \leq \sum_{i \in I_j \setminus G_{\tau_j}} \tilde{y}_i^j w(i)\).

2. If \(|K_j| = 1\) there is \(i^*_j \in I\) such that \(\sum_{i \in I_j \setminus \{i^*_j\}} \bar{x}_i w(i) \leq \sum_{i \in I_j \setminus \{i^*_j\}} \tilde{y}_i^j w(i)\).

3. It holds that \(I_j \subseteq \text{supp}(\bar{y}^j)\).

We refer to \((I_{j})_{j=0}^{r}\) as the Block Association of \((\bar{x}, \bar{y}^0, \ldots, \bar{y}^r)\) in \(P^e\).

Defining a block association for the items relies on an abstract notion of constraints. A constraint is a pair \((\bar{c}, \beta)\) of coefficients \(\bar{c} \in \mathbb{R}_{\geq 0}^I\) and a bound \(\beta \in \mathbb{R}_{\geq 0}\). We say that \(\bar{c} \in \mathbb{R}_{\geq 0}^I\) satisfies the constraint \((\bar{c}, \beta)\) if \(\sum_{i \in I} \bar{c}_i \cdot \bar{c}_i \leq \beta\), and that \(\bar{c}\) is \(\gamma\)-semi-satisfies the constraint \((\bar{c}, \beta)\) if \(\exists i^* \in I\) such that \(\sum_{i \in I \setminus \{i^*\}} \bar{c}_i \cdot \bar{c}_i \leq \beta\).

We say that \((\tilde{\gamma}^r)_{r=1}^p\), \(\tilde{\gamma}^r \in [0,1]^I\) are a decomposition of \(\bar{x} \in [0,1]^I\) if \(\bar{x} = \sum_{r=1}^p \tilde{\gamma}^r\). Item \(i \in I\) is perfect w.r.t. vectors \((\tilde{\gamma}^r)_{r=1}^p\) if there is at most one vector \(\tilde{\gamma}^r\) for which \(\tilde{\gamma}^r_i \neq 0\); otherwise, \(i\) is broken. We say that a vector \(\tilde{\gamma}^r \in [0,1]^I\) is perfect w.r.t \((\tilde{\gamma}^r)_{r=1}^p\) if all items in \(\text{supp}(\tilde{\gamma}^r)\) are perfect (recall \(\text{supp}(\tilde{\gamma}) = \{i \in I \mid \tilde{\gamma}_i > 0\}\) for \(\tilde{\gamma} \in [0,1]^I\)). Otherwise, \(\tilde{\gamma}^r\) is broken. A decomposition \((\tilde{\gamma}^r)_{r=1}^p\) is perfect if all items are perfect (and therefore all vectors \(\tilde{\gamma}^r\) are perfect). Given vectors \((\tilde{\lambda})_{r=1}^p\) as the bipartite graph \(G = (S,T,E)\) where \(S\) is the set of broken items, \(T = \{1 \leq r \leq p \mid \tilde{\lambda}^r\) is broken\} is the set of broken vectors, and \(E = \{(i,r) \mid \tilde{\lambda}^r_i \neq 0\}\) (see Figure 3a).

Lemma C.2. Given a set of constraints \(\{(\tilde{c}^r, \beta^r) \mid 1 \leq r \leq p\}\) and a decomposition \((\tilde{\gamma}^r)_{r=1}^p\) of \(\bar{x} \in [0,1]^I\), such that \(\tilde{\gamma}^r\) satisfies \((\tilde{c}^r, \beta^r)\) for \(1 \leq r \leq p\), Algorithm 3 returns in polynomial time a perfect decomposition \((\tilde{\lambda}^r)_{r=1}^p\) of \(\bar{x}\) such that \(\tilde{\lambda}^r\) semi-satisfies \((\tilde{c}^r, \beta^r)\), and \(\text{supp}(\tilde{\lambda}^r) \subseteq \text{supp}(\tilde{\gamma}^r)\) for \(1 \leq r \leq p\).

\[\text{supp}(\tilde{\gamma}) = \{i \in I \mid \tilde{\gamma}_i > 0\}\) for \(\tilde{\gamma} \in [0,1]^I\).
(a) Broken graph \( G = (S,T,E) \) with \( S = \{1,3\} \), \( T = \{1,3\} \), and \( E = \{(1,1),(1,3),(3,1),(3,3)\} \) for the following instance \( I = \{1,2,3\} \), \( p = 3 \), \( \bar{\gamma}_1 = (\frac{1}{3},0,\frac{2}{3}) \), \( \bar{\gamma}_2 = (0,1,0) \), and \( \bar{\gamma}_3 = (\frac{2}{3},0,\frac{1}{3}) \).

(b) Shift items along cycle \( C = (i_1,r_1,i_2,r_2) \), with \( i_1 = r_1 = 1 \) and \( i_2 = r_2 = 3 \).

Figure 1: Visualizations of procedures in Algorithm 4

Algorithm 4: Make Perfect

\[\text{Input:} \text{ Vectors } \bar{x}, (\bar{\gamma}_r^p)_{r=1}^p, \text{ constraints } (\bar{\alpha}^r,\beta^r) \text{ for } 1 \leq r \leq p.\]
1. Initialize \( \bar{\lambda}_r^i \) for \( 1 \leq r \leq p \)
2. while there exists a broken item \( i \) (w.r.t. \( (\bar{\lambda}_r^i)_{r=1}^p \)) do
   3. Update \( G = (S,T,E) \) to be the broken bipartite graph of \( (\bar{\lambda}_r^i)_{r=1}^p \).
   4. if there exists a broken vector \( r \in T \) such that \( \deg_G(r) = 1 \) then
      5. Let \( i \in S \) be the neighbor of \( r \). Set \( \bar{\lambda}_r^i \leftarrow \bar{x}_i \), and for all \( r' \neq r \) set \( \bar{\lambda}_r^{i'} \leftarrow 0 \).
   else
      6. Let \( C = (i_1,r_1,\ldots,i_k,r_k) \) be a simple cycle in \( G \).
      7. Find a vector \( \bar{\nu} \in \mathbb{R}^k, \bar{\nu} \neq 0 \) such that \( \bar{c}_{ij}^{r_j} \cdot \bar{\nu}_j - \bar{c}_{ij}^{r_{j+1}} \cdot \bar{\nu}_{j+1} \leq 0 \) for \( 1 \leq j \leq k \)
         \( (\bar{\nu}_{k+1} = \bar{\nu}_1, i_{k+1} = i_1) \).
      8. Select a maximal \( \delta \) such that \( \bar{\lambda}_{ij}^{r_j} + \delta \bar{\nu}_j, \bar{\lambda}_{ij}^{r_{j+1}} - \delta \bar{\nu}_{j+1} \geq 0 \) for every \( 1 \leq j \leq k \).
      9. Set \( \bar{\lambda}_{ij}^{r_j} \leftarrow \bar{\lambda}_{ij}^{r_j} + \delta \bar{\nu}_j \) and \( \bar{\lambda}_{ij}^{r_{j+1}} \leftarrow \bar{\lambda}_{ij}^{r_{j+1}} - \delta \bar{\nu}_{j+1} \) for \( 1 \leq j \leq k \).
10. Return \( (\bar{\lambda}_r^i)_{r=1}^p \)
We first show that the algorithm is well defined, as stated in the next lemma.

**Lemma C.3.** Algorithm 2 is well defined. That is, if the statement in Step 3 is false, then there exists a cycle and a vector $\bar{\nu}$ as defined in Steps 4, 8.

**Proof.** We note that all of the items in the broken graph (the set $S$) are of degree two or more by definition, and if Step 3 is reached then all the vectors $r \in T$ are of degree at least two. It follows that the broken graph $G$ contains a cycle.

The existence of $\bar{\nu}$ in Step 8 follows from a simple argument in linear algebra. The equations $\bar{c}_{ij} \cdot \bar{v}_j - \bar{c}_{ij+1} \cdot \bar{v}_{j+1} = 0$ for $1 \leq j \leq k - 1$ form $k - 1$ linear homogeneous equations with $k$ variables. Thus, there is $\bar{v}' \in \mathbb{R}^k$, $\bar{v}' \neq 0$ which satisfies the equations. Hence, either $\bar{\nu} = \bar{v}'$ or $\bar{\nu} = -\bar{v}'$ satisfies the condition in Step 8 (i.e., the $k$-th inequality holds as well).

To prove Lemma C.2, we show that the number of edges in $G$ decreases in each iteration, and that the following invariants hold throughout the execution of the algorithm.

1. For any $1 \leq r \leq p$, either $\bar{\lambda}^r$ is perfect and semi-satisfies $(\bar{c}^r, \beta^r)$, or it satisfies $(\bar{c}^r, \beta^r)$.
2. For any $1 \leq r \leq p$ it holds that $\text{supp}(\bar{\lambda}^r) \subseteq \text{supp}(\bar{\gamma}^r)$.
3. $(\bar{\lambda}^r)_{r=1}^p$ is a decomposition of $\bar{x}$.

These remaining parts are fairly simple, hence deferred to Appendix C.1.

**Proof of Lemma C.2 (Block Association).** Let $M = \{0 \leq j \leq \ell \mid |K_j| > 1\}$ be the blocks which contain multiple bins. Also, we use $\wedge$ as an element-wise minimum over two vectors. That is, for $\bar{v}^1, \bar{v}^2 \in \mathbb{R}^I$ we define $\bar{v}^1 \wedge \bar{v}^2 = \bar{\gamma}$, where $\bar{\gamma} \in \mathbb{R}^I$ and $\bar{\gamma}_i = \min\{\bar{v}^1_i, \bar{v}^2_i\}$ for all $i \in I$.

Let $R = \{(j, k) \mid j \in M, 1 \leq k \leq \tau_j \} \cup \{j \mid 0 \leq j \leq \ell\}$. We set a set of constraints $\{(\bar{c}^r, \beta^r) \mid r \in R\}$ for $(\bar{x}, \bar{y}^1, \ldots, \bar{y}^r)$. For every $i \in I$ we define $\bar{c}_i^j = w(i)$ for any $0 \leq j \leq \ell$ and $\bar{c}_i^{j,k} = 1$ for any $j \in M$ and $1 \leq k \leq \tau_j$. Also, we define $S^j = L_{K_j \mu}$ for $j \in M$, $S^{j,k} = G^j_k$ for $j \in M$ and $1 \leq k \leq \tau_j$, and $S^j = \{i \in I \mid w(i) \leq W^*_k\}$ for $j \notin M$. We define $\bar{\gamma}^r = \bar{\gamma}^r \wedge \mathbb{1}_{S^r}$ for $r = (j, k)$ or $r = j$, and $\beta^r = \bar{c}^r \cdot \bar{\gamma}^r$. By definition, it holds that $\bar{\gamma}^r$ satisfies $(\bar{c}^r, \beta^r)$. Furthermore, it can be easily verified that $(\bar{\gamma}^r)_{r \in R}$ is a decomposition of $\bar{x}$.

By Lemma C.2 given $\bar{x}$, $(\bar{\gamma}^r)_{r \in R}$ and the constraints $(\bar{c}^r, \beta^r)_{r \in R}$ Algorithm 1 returns a perfect decomposition $(\bar{\lambda}^r)_{r \in R}$ for $r \in R$ such that $\bar{\lambda}^r$ semi-satisfies $(\bar{c}^r, \beta^r)$. Define $I_j = \text{supp}(\bar{\lambda}^r) \cup (\cup_{i=1}^{j} \text{supp}(\bar{\lambda}^{j,k}))$ for $j \in M$, and $I_j = \text{supp}(\bar{\lambda}^r)$ for $0 \leq j \leq \ell$, $j \notin M$. It follows from Lemma C.2 that $(I_j)_{j=0}^j$ can be computed in polynomial time.

For every $r \in R$ it holds that $\text{supp}(\bar{\lambda}^r) \subseteq \text{supp}(\bar{\gamma}^r) \subseteq S^r$. Furthermore, it can be easily verified that for any $r \in R$, $r = j$ or $r = (j, k)$, it holds that $I_j \cap S^r = \text{supp}(\bar{\lambda}^r)$. By the same argument, it follows that $I_j \subseteq \text{supp}(\bar{y}^r)$. Finally, we note that $(I_j)_{j=0}^j$ is a partition of $\text{supp}(\bar{x})$, since $(\bar{\lambda}^r)_{r \in R}$ is a perfect decomposition of $\bar{x}$.

For every $j \in M$ and $1 \leq k \leq \tau_j$, it holds that $\bar{\lambda}^{j,k}$ semi-satisfies $(\bar{c}^{j,k}, \beta^{j,k})$. Hence, there is $i^{*} \in I$, such that

$$\sum_{i \in I_j \cap G^j_k} \bar{x}_i = \sum_{i \in \text{supp}(\bar{\lambda}^{j,k})} \bar{x}_i = \sum_{i \in I} \bar{\lambda}^{j,k}_i \leq 1 + \sum_{i \in I \setminus \{i^{*}\}} \bar{\lambda}^{j,k}_i \leq 1 + \beta^{j,k} \leq 1 + \sum_{i \in I \setminus \{i^{*}\}} \bar{\gamma}^j_i \leq 2 + \mu \cdot |K_j|.$$
Finally, for $0 \leq j \leq \ell$, $j \notin M$ it holds that $(\bar{\lambda}^j)$ semi-satisfies $(\bar{c}^j, \beta^j)$. Thus, there is $i^*_j \in I$ such that,
\[
\sum_{i \in I \setminus \{i^*_j\}} \bar{x}_i \cdot w(i) = \sum_{i \in \text{supp}(\bar{x}) \setminus \{i^*_j\}} \bar{x}_i \cdot w(i) = \sum_{i \in I \setminus \{i^*_j\}} \bar{\lambda}^j_i \cdot \bar{c}^j_i \leq \beta^j = \sum_{i \in I} w(i) \cdot \bar{y}^j_i.
\]

\[\square\]

C.1 Block Association: Deferred Proofs

Lemma C.4. After each iteration of Algorithm 4 the number of edges in the broken bipartite graph $G$ decreases by at least one.

Proof. It is easy to see that if the statement in Step 4 is true, one broken item becomes perfect and its edges are removed. Otherwise, as proven in Lemma C.3 a non-trivial vector $\bar{\nu}$ exists for which the conditions in Step 8 hold. In Step 9 the maximal $\delta$ is selected such that $\bar{\lambda}^i = 0$ for at least one item-vector pair $(i, r)$. This means that $(i, r)$ is removed from $G$.

Lemma C.5. Given a decomposition $(\gamma^r)_{r=1}^p$ of $\bar{x}$ and constraints $(\bar{c}^r, \beta^r)$ for $1 \leq r \leq p$ such that $\gamma^r$ satisfies $(\bar{c}^r, \beta^r)$, through out its run Algorithm 4 maintains a set of vectors $(\bar{\lambda}^r)_{r=1}^p$ such the following holds.

1. For any $1 \leq r \leq p$, either $\bar{\lambda}^r$ is perfect and semi-satisfies $(\bar{c}^r, \beta^r)$, or it satisfies $(\bar{c}^r, \beta^r)$.
2. For any $1 \leq r \leq p$ it holds that $\text{supp}(\bar{\lambda}^r) \subseteq \text{supp}(\gamma^r)$.
3. $(\bar{\lambda}^r)_{r=1}^p$ is a decomposition of $\bar{x}$.

Proof. For each vector $\bar{\lambda}^r$ let $\bar{\lambda}^r_j$ be the value of $\bar{\lambda}^r$ after the $j$-th iteration of the algorithm $(\bar{\lambda}^0_0)_{r=1}^p$ are the values set upon initialization). It is easy to see that all three requirements hold for $(\bar{\lambda}^0_0)_{r=1}^p$ as $\bar{\lambda}^r_0 = \bar{\gamma}^r$ for $1 \leq r \leq p$, and the requirements hold for $(\bar{\gamma}^r)_{r=1}^p$.

We will prove that the requirements hold after iteration $j + 1$ under the assumption that all requirements hold after the $j$-th iteration, i.e., $(\bar{\lambda}^r_j)_{r=1}^p$ is a decomposition of $\bar{x}$, each vector $\bar{\lambda}^r_j$ is either perfect and semi-satisfies $(\bar{c}^r, \beta^r)$ or it satisfies $(\bar{c}^r, \beta^r)$, and $\text{supp}(\bar{\lambda}^r_j) \subseteq \text{supp}(\gamma^r)$. If a vector $\bar{\lambda}^r_j$ is perfect, no changes are made to it, i.e., $\bar{\lambda}^r_j = \bar{\gamma}^r$. As it did not change, its support does not change as well and it still semi-satisfies $(\bar{c}^r, \beta^r)$.

Next, we will consider the case where it doesn’t perfect. This means that $\bar{\lambda}^r_j$ it satisfies $(\bar{c}^r, \beta^r)$. Changes to vector $\bar{\lambda}^r_j$ can only be made in Steps 5 or 10. If it changed in Step 5 it becomes perfect and its support is unchanged. Let $i$ be the item of Step 5 then
\[
\sum_{i' \in I \setminus \{i\}} (\bar{\lambda}^r_{j+1})_i' \cdot \bar{c}^r_i' = \sum_{i' \in I \setminus \{i\}} (\bar{\lambda}^r_j)_i' \cdot \bar{c}^r_i' \leq \sum_{i' \in I} (\bar{\lambda}^r_j)_i' \cdot \bar{c}^r_i' \leq \beta^r,
\]
where the last inequality holds since $\bar{\lambda}^r_j$ satisfies $(\bar{c}^r, \beta^r)$. That is, $\bar{\lambda}^r_{j+1}$ is perfect and semi-satisfies $(\bar{c}^r, \beta^r)$.

Otherwise, changes were made to vector $\bar{\lambda}^r_j$ in Step 10. As items are shifted along positive edges in cycle $C$, no new items are added to the support of $\bar{\lambda}^r_j$, i.e., $\text{supp}(\bar{\lambda}^r_{j+1}) \subseteq \text{supp}(\bar{\lambda}^r_j) \subseteq \text{supp}(\gamma^r)$. In addition, since $\bar{\lambda}^r_j$ satisfies $(\bar{c}^r, \beta^r)$ and due to the condition in Step 8 it holds that $\bar{c}^r \cdot \bar{\lambda}^r_{j+1} \leq \bar{c}^r \bar{\lambda}^r_j \leq \beta^r$. Thus $\bar{\lambda}^r_{j+1}$ satisfies $(\bar{c}^r, \beta^r)$.

Finally, we note that $\bar{\lambda}^r_{j+1} = 1$ remains a decomposition of $\bar{x}$ as any changes made in Steps 5 and 10 shift items between vectors and do not change their total sum (see illustration in Figure 1b). $\square$

D Solving the Instance Polytope

In this section we prove Lemmas 4.4. The proof utilizes the ellipsoid method with separation oracles via the result of Grötschel, Lovász and Schrijver 24 (see 23 for a comprehensive survey). As we cannot provide an exact separation oracle, or even a weak oracle (as defined in 24), we follow a known scheme,
which appeared for example in [23, 20], in which the oracle may fail, and the ellipsoid method will be aborted. In this case, however, the input for the separation oracle can be used to derive an approximate solution. As in [20] and [23] we use the dual program to replace the exponential number of variables with an exponential number of constraints (for which we provide a separation oracle). However, as the proof of Lemma [4.1] involves matroid and matching polytopes, in a naive representation of the polytope both primal and dual programs have an exponential number of constraints and variables. To circumvent this issue a variant of the technique is used.

We start by providing an approximate separation oracle for the block polytope.

**Lemma D.1.** There is an algorithm which given an MKC \((w, B, W)\), a block \(K \subseteq I\), \(\vec{y} \in \mathbb{R}^I\) and \(\varepsilon > 0\) either determines that \((1 - \varepsilon) \cdot \vec{y} \in P_K\) or finds \(\nu \in [0, 1]^I\) such that \(\vec{y} \cdot \nu > \vec{y}' \cdot \nu\) for every \(\vec{y}' \in P_K\), where \(P_K\) is the block polytope of \(K\).

**Proof.** Let \((w, B, W)\), \(K\), \(\vec{y}\) and \(\varepsilon\) be as defined in the lemma. We first consider the following linear program, in which the variables are \(\bar{z}_C\) for \(C \in C_K\) (\(C_K\) are the \(K\)-configurations, as defined in Section 2).

\[
\min \quad \sum_{C \in C_K} \bar{z}_C \quad \text{such that} \quad \forall i \in I : \sum_{C \in C_K \text{ s.t. } i \in C} \bar{z}_C \geq \bar{y}_i
\]

\[
\forall C \in C_K : \quad \bar{z}_C \geq 0
\]

By Definition 2.2 it holds that \(\vec{y} \in P_K\) if and only if the optimal solution for (9) is \(|K|\) or less. The dual of the above linear program is the following.

\[
\max \quad \sum_{i \in I} \bar{\beta}_i \cdot \bar{y}_i
\]

\[
\text{such that} \quad \forall C \in C_K : \sum_{i \in C} \bar{\beta}_i \leq 1
\]

\[
\forall i \in I : \quad \bar{\beta}_i \geq 0
\]

We note that the first constraint in (10) is essentially an instance of the knapsack problem, for which there is a known FTPAS [25] (also appears in textbooks such as [42]). Let \(\bar{\beta} \in \mathbb{R}_{\geq 0}^I\) be a vector in the feasible region on (10) and let \(\vec{y}' \in P_K\). Thus there is \(\vec{z}' \in [0, 1]^{C_K}\) such that \((\vec{y}', \vec{z}') \in P_K\) (\(P_K\) is the extended block polytope of \(K\)). It holds that

\[
\bar{\beta} \cdot \vec{y}' = \sum_{i \in I} \bar{\beta}_i \cdot \vec{y}'_i \leq \sum_{i \in I} \bar{\beta}_i \sum_{C \in C_K \text{ s.t. } i \in C} \vec{z}_C = \sum_{C \in C_K} \bar{z}_C \sum_{i \in C} \bar{\beta}_i \leq \sum_{C \in C_K} \bar{z}_C' \leq |K|.
\]

The first and last inequalities follow from the definition of the extended block polytope (Definition 2.1), the second inequality holds since \(\bar{\beta}\) is in the feasible region of (10). In the following we will attempt to find a vector \(\bar{\beta}\) for which \(\bar{\beta} \cdot \vec{y} > |K|\).

For any \(v \geq 0\) define,

\[
D_v = \left\{ \bar{\beta} \in \mathbb{R}_{\geq 0}^I \quad \mid \forall C \in C_K : \sum_{i \in C} \bar{\beta}_i \leq 1, \bar{\beta} \cdot \vec{y} \geq v \right\}.
\]

Clearly, if \(D_v \neq \emptyset\) then the optimal value of (11) is at least \(v\), and otherwise the value is less than \(v\). Given \(v \geq 0\) and \(\delta > 0\) we can use the the ellipsoid method along with a separation oracle to determine if \(D_v\) is empty or not. Given \(\bar{\beta} \in \mathbb{R}_{\geq 0}^I\) the separation oracle will verify that \(\bar{\beta} \cdot \vec{y} \geq v\), and otherwise it will return \(\bar{\beta} \cdot \vec{y} \geq v\) as the separating hyper-plane. If \(\bar{\beta} \cdot \vec{y} \geq v\), the separation oracle executes an FPTAS for the knapsack problem (e.g., [23]) with the following instance and \(\varepsilon\). The set of items is \(I\), with weight \(w(i)\) and profit \(\bar{\beta}_i\) for item \(i \in I\), and knapsack (bin) capacity \(W_K^*\). If the approximation scheme finds a set \(C\) of items such that \(\sum_{i \in C} \bar{\beta}_i > 1\), then \(C \in C_K\) (as this is the capacity constraint of the knapsack). Hence, the separation oracle can return \(\sum_{i \in C} \bar{\beta}_i \leq 1\) as the violated constraint.
If the set $C$ returned by the approximation scheme satisfies $\sum_{i \in C} \bar{\beta}_i \leq 1$ then the separation oracle aborts the ellipsoid algorithm and returns $\bar{\beta}$.

The execution of the ellipsoid given $v$ can end in two possible ways. It will either determine that $D_v = \emptyset$, or the separation oracle will abort it. If the execution is aborted, let $\bar{\beta}$ be the returned vector. Then $\bar{\beta} \cdot \bar{y} \geq v$ and for $C \in \mathcal{C}_K$ it holds that $\sum_{i \in C} \bar{\beta}_i \leq \frac{1}{1 - \varepsilon}$ (otherwise the optimal solution for the knapsack has value greater than $\frac{1}{1 - \varepsilon}$ and the approximation scheme will find a solution of value greater than $(1 - \varepsilon) \frac{1}{1 - \varepsilon} = 1$. Hence $(1 - \varepsilon) \bar{\beta} \in D_{(1 - \varepsilon)v}$. Thus, in time polynomial in the instance and $\varepsilon^{-1}$ the algorithm either finds $(1 - \delta) \bar{\beta} \in D_{(1 - \delta)v}$ or asserts $D_v = \emptyset$.

Thus, using a binary search over the values of $v$ we can find $(1 - \varepsilon)v_2 < v_1 < v_2$ and $\bar{\beta} \in [0, 1]^d$ such that $D_{v_2} = \emptyset$ and $\bar{\beta} \in D_{v_1}$ in time polynomial in the input size and $\varepsilon^{-1}$. If $v_1 > |K|$ then $\bar{\beta} \cdot \bar{y} > |K|$, and the algorithm can return $\bar{v} = \bar{\beta}$ due to (11). Otherwise, $v_2 \leq |K| \frac{1}{1 - \varepsilon}$. Therefore, the optimal solution for (10) is at most $\frac{|K|}{1 - \varepsilon}$, and by duality, the optimal solution for (9) is at most $\frac{|K|}{1 - \varepsilon}$, thus $(1 - \varepsilon)\bar{y} \in P_K$.

The overall running time of the algorithm is polynomial in the input size and $\varepsilon^{-1}$.

**Proof of Lemma 4.4.** Let $(I, (w_t, B_t, W_t)_{t=1}^d, I, f)$ be a $d$-MKCP instance, $(K_j^t)_{j=1}^{\ell_t}$ be a partition of $B_t$ to block for $1 \leq t \leq d$, $\bar{c} \in \mathbb{R}^d$ and $\gamma > 0$. Let $P_t$ be the $\gamma$-partition polytope of $(w_t, B_t, W_t)$ and $(K^t_j)_{j=1}^{\ell_t}$ for $1 \leq t \leq d$ and $P = P(I) \cap \left( \bigcap_{t=1}^d P_t \right)$ be the $\gamma$-instance polytope of the instance. Finally, let $\text{OPT} = \max_{\bar{x} \in P} \bar{c} \cdot \bar{x}$.

As in the proof of Lemma D.1 we will use a binary search and an approximate separation oracle, this time to an extended form of $P$. Let $P_t^I$ be the block polytope of $K_j^t$ in the MKC $(w_t, B_t, W_t)$ for $1 \leq j \leq \ell_t$ and $1 \leq t \leq d$. Given $v \geq 0$, consider the polytope $D_v$, such that each element in $D_v$ is of the form $(\bar{x}, \left( (\bar{y}^{t,j})^l_{j=1} \right)_{t=1}^d)$, $\bar{x}$, $D_v \subseteq [0, 1]^d \times \left( \bigtimes_{t=1}^d \bigtimes_{j=1}^{\ell_t} [0, 1]^d \right)$ and contains all the vectors which satisfy the following constraints.

\begin{align*}
\forall 1 \leq t \leq d, \ i \in I : \\
\bar{x}_i \leq \sum_{j=1}^{\ell_t} (\bar{y}^{t,j}_i)^l_{j=1} \\
\forall 1 \leq t \leq d, \ 1 \leq j \leq \ell_t, \ |K_j^t| = 1, \ i \in I \setminus L_{K_j^t, \gamma} : \\
\bar{y}^{t,j}_i = 0 \\
\forall 1 \leq t \leq d, \ 1 \leq j \leq \ell_t : \\
\bar{x} \in P(I) \\
\bar{y}^{t,j} \in P_{K_j^t}
\end{align*}

(13) (14) (15) (16) (17)

It can be easily observed that $D_v \neq \emptyset$ if and only if $v \geq \text{OPT}$. Furthermore, it holds that if $(\bar{x}, \left( (\bar{y}^{t,j})^l_{j=1} \right)_{t=1}^d) \in D_v$ for some $v \geq 0$ then $\bar{x} \in P$.

Given $v \geq 0$ and $\delta > 0$ we can use the ellipsoid method to determine if $D_v = \emptyset$ using a separation oracle. Given $(\bar{x}, \left( (\bar{y}^{t,j})^l_{j=1} \right)_{t=1}^d)$, the separation oracle will first verify constraints (13), (14) and (15) hold. If not, it will return the constraint as a separation hyper-plane. Next, it will use a separation oracle for $\bar{x} \in P(I)$, we note that such oracles exists for all constraints $I$ considered in this paper (see 37 for further details). Finally, for every $1 \leq t \leq d$ and $1 \leq j \leq \ell_t$ the algorithm of Lemma D.1 will be used with $\bar{y}^{t,j}$, the block $K_j^t$ of the MKC $(w_t, B_t, W_t)$ and $\varepsilon$. If it returns a separating vector $\bar{v}$ it can be used as a separator for the ellipsoid algorithm. If it holds that $(1 - \varepsilon)\bar{y}^{t,j} \in P_{K_j^t}$ for every $1 \leq j \leq \ell_t$ and $1 \leq t \leq d$ then the separation oracle aborts the ellipsoid and returns $(\bar{x}, \left( (\bar{y}^{t,j})^l_{j=1} \right)_{t=1}^d)$.

As in the proof of Lemma D.1 the ellipsoid algorithm can terminate in one of two ways. Either it determines that $D_v = \emptyset$, or the separation oracle aborts the process and returns $(\bar{x}, \left( (\bar{y}^{t,j})^l_{j=1} \right)_{t=1}^d)$. In the latter case it can be easily verified that $(1 - \varepsilon) \cdot (\bar{x}, \left( (\bar{y}^{t,j})^l_{j=1} \right)_{t=1}^d) \in D_{(1 - \varepsilon)v}$. Furthermore, the running time of the ellipsoid method is polynomial in the input size and in $\varepsilon^{-1}$. 

22
Thus, a binary search can be used to find \((1 - \varepsilon)v_2 < v_1 < v_2\) and \(\left(\bar{x}, \left(\left(\bar{y}^tj\right)_{j=1}^{l_t}\right)_{t=1}^{d}\right)\) such that \(D_{v_2} = \emptyset\) and \(\left(\bar{x}, \left(\left(\bar{y}^tj\right)_{j=1}^{l_t}\right)_{t=1}^{d}\right) \in D_{v_1}\). Hence \(\text{OPT} < v_2\), \(\bar{x} \in P\) and
\[
\bar{x} \cdot \bar{c} \geq v_1 \geq (1 - \varepsilon)v_2 \geq (1 - \varepsilon)\text{OPT}.
\]
The process can be implemented in time polynomial in \(\varepsilon^{-1}\) and the input size, thus the promised FPTAS is obtained.

\[\square\]

E Approximation Algorithm: Omitted Proofs

**Proof of Lemma 4.5.** Let \(R\) be the \(d\)-MKCP instance \((I, (w_t, B_t, W_t)_{t=1}^{d}, \mathcal{I}, f)\) and \(\left(K_j\right)_{j=0}^{l_t}\) the \(N\)-leveled partition of \(B_t\) for \(1 \leq t \leq d\). W.l.o.g assume that \(A_t\) is disjoint for any \(1 \leq t \leq d\). That is, \(A_t(b_1) \cap A_t(b_2) = \emptyset\) for any \(b_1 \neq b_2\).

For any \(1 \leq t \leq d\) let \(V_t = \{i \in A(b) \mid 0 \leq j \leq \ell_t, b \in K_j\}, |K_j| = 1, w_i(i) > \gamma \cdot W_i(b)\} be the set of all item assigned to a block of a single bin in the \(t\)-th MKC and are \(\gamma\)-heavy. Since there are at most \(N^2\) blocks of a single bin (by the definition of \(N\)-leveled partition) and a configuration cannot contain \(\gamma^{-1}\) or more \(\gamma\)-heavy items, it follows that \(|V_t| \leq \frac{N^2}{\gamma}\). Let \(V = \bigcup_{t=1}^{d} V_t\), thus \(|V| \leq \frac{N^2d}{\gamma}\).

Define \(S' = S \setminus V\) and denote \(f(S) = \text{OPT}\). By the definition of restricted \(d\)-MKCP it holds that \(f(\{1\}) - f(\emptyset) \leq \frac{\text{OPT}}{\xi}\) for any \(i \in I\). Denote \(V = \{i_1, \ldots, i_n\}\). As \(f\) is submodular we have
\[
f(S) = f(S') + \sum_{j=1}^{\eta} f(S' \cup \{i_1, \ldots, i_j\}) - f(S' \cup \{i_1, \ldots, i_{j-1}\})
\leq f(S') + \sum_{j=1}^{\eta} f(\{i_j\}) - f(\emptyset)
\leq f(S') + |V| \cdot \frac{\text{OPT}}{\xi}.
\]

Hence, \(f(S') \geq \left(1 - \frac{N^2d}{\xi^2\gamma}\right) f(S)\).

We are left to show that \(\mathbb{1}_{S'} \in P\). For \(1 \leq t \leq d\) and \(0 \leq j \leq \ell_t\) define \(S_{t,j}' = \bigcup_{b \in K_j} A_t(b) \cap S_t\), \(\bar{y}^{t,j} = \mathbb{1}_{S_{t,j}'}\) and \(\bar{z}^{t,j} \in [0, 1]^{\ell_{K_j}}\) (recall \(K_j\) is the set of \(K_j\)-configurations) by \(\bar{z}^{t,j}_C = 1\) if there is \(b \in K_j\) such that \(C = A_t(b)\) and \(\bar{z}^{t,j}_C = 0\) otherwise. It can be easily verified that \((\mathbb{1}_{S_{t,j}'} , \bar{z}^{t,j}) \in P_{K_j}^e\) where \(P_{K_j}^e\) is the extended block polytope of \(K_j\). Thus, \(\bar{y}^{t,j} = \mathbb{1}_{S_{t,j}'} \in P_{K_j}^e\) where \(P_{K_j}^e\) is the block polytope of \(K_j\).

Let \(1 \leq t \leq d\), \(0 \leq j \leq \ell_t\) and \(i \in I \setminus K_j\) with \(K_j = \{b\}\) for some \(b \in B_t\). If \(i \in A_t(b)\) it follows that \(i \in V_t\), therefore \(i \notin S_t'\) and \(i \notin S_{t,j}'\). If \(i \notin A_t(b)\) then \(i \notin S_{t,j}'\) as well. Hence \(\bar{y}^{t,j}_i = 0\).

As \(\mathbb{1}_{S'} = \sum_{j=0}^{\ell_t} \bar{y}^{t,j}\) we can conclude that \((\mathbb{1}_{S'} , \bar{y}^{t,0}, \ldots, \bar{y}^{t,\ell_t}) \in P_t^e\) for any \(1 \leq t \leq d\), where \(P_t^e\) is the extended \(\gamma\)-partition polytope of \((w_t, B_t, W_t)\) and the partition \((K_j)_{j=0}^{l_t}\).

Finally, as \(S' \subseteq S\) and \(S \in \mathcal{I}\) it follows that \(S' \in \mathcal{I}\) and \(\mathbb{1}_{S'} \in P(\mathcal{I})\). Thus we can conclude \(\mathbb{1}_{S'} \in P\).
3. The function $f$ is modular and $\mathcal{I}$ defines the independent sets of a matroid, the intersection of the independent sets of two matroids, or a matching.

If follows that Lemma 4.6 only deals with $d$-MKCP instances in which $(f, \mathcal{I})$, the function and additional constraint, are valid.

We use the following lemma to provide a unified probabilistic analysis which does not depend on the sampling method used in Step 2 of Algorithm 2.

**Lemma E.1.** Let $(\mathcal{I}, f)$ be a valid pair, $\bar{x} \in P(\mathcal{I})$ (recall $P(\mathcal{I})$ is the convex hull of $\mathcal{I}$) and $\delta > 0$. Also, let $R$ be a random set sampled by $\bar{x}$, $\delta$, and $\mathcal{I}$. Then,

1. For any $i \in I$, $\Pr(i \in R) = (1 - \delta)^2 \bar{x}_i$.

2. For any $\bar{a} \in [0,1]^I$, $\epsilon > 0$ and $\zeta \geq (1 - \delta)^2 \bar{a} \cdot \bar{x}$,

$$\Pr\left(\sum_{i \in R} a_i \geq (1 + \epsilon) \zeta\right) \leq \exp\left(-\frac{\zeta \cdot \epsilon^2}{20}\right).$$

3. Let $\epsilon > 0$, $f_{\text{max}} = \max_{i \in I} f(\{i\}) - f(\emptyset)$ and $\zeta \leq (1 - \delta)^2 F(\bar{x})$, where $F$ is the multilinear extension of $f$. Also, assume $f$ is monotone. Then,

$$\Pr(f(R) \leq (1 - \epsilon) \zeta) \leq \exp\left(-\frac{\zeta \cdot \epsilon^2}{20 \cdot f_{\text{max}}}\right).$$

4. If $\mathcal{I} = 2^I$ then the events $(i \in R)_{i \in I}$ are independent.

5. The sampling of $R$ can be implemented in polynomial time.

Lemma E.1 follows directly from [10], [11], the concentration bound of [43] and the Chernoff bounds of [21]. The requirement for $(\mathcal{I}, f)$ to be valid stems from the limitation of the concentration bound in [43] and the FKG inequality (as in [12]).

Before we proceed to the proof of Lemma 4.6 we present its main structure. We use Lemma 4.5 to show that there is $S' \subseteq I$ such that $1_{S'} \in P$ and $f(S') \geq \left(1 - \frac{N^2 \cdot d}{\xi^2}\right) \text{OPT} \geq (1 - \epsilon^2) \text{OPT}$, where the last inequality follows from a proper selection of the parameters. Thus, it holds that $F(\bar{x}) \geq c \cdot (1 - \epsilon^2)^2 \text{OPT}$. We use standard concentration bounds to show that with high probability (w.h.p.) $J \cap I'_j$ satisfies the conditions of Fractional Grouping (Lemma 2.6); thus, w.h.p. the packing in Step 3 succeeds. In the monotone (and modular) case, concentration bounds are used to show that $f(J)$ is sufficiently large w.h.p. In the non-monotone case, the expected value of the returned solution is bounded using the FKG inequality (as in [12]).

**Proof of Lemma 4.6.** Let $d \in \mathbb{N}$, $\epsilon > 0$ and $M > 0$. W.l.o.g assume $\epsilon < 0.1$. Define $\delta = \epsilon^2$ and $\mu = \frac{\delta}{4}$ (as in Algorithm 2). Let $\rho > 0$ be a constant such that the algorithm of [28] returns a bin packing using $\text{OPT} + \rho \log^2 \text{OPT}$ bins for every bin packing instance. By the monotone convergence theorem it holds that

$$\lim_{N \to \infty} \sum_{j=1}^{\infty} N^2 \left(16 \cdot \delta^{-2} + 2\right) \cdot \exp\left(-\frac{\delta^4}{640} \cdot N^j\right) = 0.$$

Thus there is $N > M$ such that $-\frac{\delta^2}{2} N' + 4 \cdot 4^\mu \cdot \rho \log^2 N' \leq 0$ for any $N' \geq N$, $N > 32 \cdot \delta^{-2}$ and

$$\sum_{j=1}^{\infty} N^2 \left(16 \cdot \delta^{-2} + 2\right) \cdot \exp\left(-\frac{\delta^4}{640} \cdot N^j\right) < \frac{\epsilon^2}{4 \cdot d}.$$  \hfill (18)

Select $0 < \gamma < \frac{\delta}{4}$ such that

$$N^2 \cdot \exp\left(-\frac{\delta^3}{160} \gamma^{-1}\right) < \frac{\epsilon^2}{4 \cdot d}.$$ \hfill (19)
Also, select $\xi \in \mathbb{N}$ such that $\exp\left(\frac{0.1\cdot(1-\varepsilon^2)^{\frac{2}{1-\varepsilon^2}}}{20} \cdot \xi\right) \leq \frac{\varepsilon^2}{2}$ and $\frac{N^2 \cdot d}{\gamma \cdot \xi} \leq \varepsilon^2$.

Let $\mathcal{R}$ be an $(N, \xi)$-restricted $d$-MKCP instance defined by $(\bar{I}, (w_t, B_t, W_t)_{t=1}^d, \mathcal{I}, f)$, such that $(f, \mathcal{I})$ is valid, and $(K^t_j)_{j=0}^\infty$ the $N$-leveled partition of $B_t$ for $1 \leq t \leq d$. Let $\text{OPT}$ be value of the optimal solution for $\mathcal{R}$. By Lemma 4.3 there is $S' \subseteq I$ such that $1_{S'} \in P$ and $f(S') \geq \left(1 - \frac{N^2 \cdot d}{\gamma \cdot \xi}\right) \text{OPT} \geq (1 - \varepsilon^2)\text{OPT}$, where $P$ is the $\gamma$-instance polytope of $\mathcal{R}$.

To execute Step 1 of Algorithm 3 the algorithms of [37] can be used to obtain $\bar{x} \in P$ such that $F(\bar{x}) \geq c \cdot (1 - \varepsilon^2) f(S')$ ($c$ is defined in the statement of the lemma). The algorithms require an optimization oracle which given $\bar{x} \in \mathbb{R}^I$, the oracle returns $\bar{\beta} \in P$ such that $\bar{c} \cdot \bar{\beta}$ is maximized. However, the algorithms can be easily adapted to use the FPTAS given in Lemma 4.4 instead of the optimization oracle, while preserving the polynomial running time and the approximation guarantees.

In case $f$ is modular Lemma 1.2 can be used directly to find $\bar{x} \in P$ s.t. $F(\bar{x}) \geq (1 - \varepsilon^2) f(S')$ as the multilinear extension $F$ in linear when $f$ is linear (see Lemma 1.1). Thus, we always have that

$$F(\bar{x}) \geq c \cdot (1 - \varepsilon^2) f(S') \geq c \cdot (1 - \varepsilon^2)^2 \text{OPT} \quad (20)$$

As in the algorithm, we use $\mu = \frac{\delta}{4}$. For any $1 \leq t \leq d$ and $0 \leq j \leq \ell_t$ let $G^{t,j}_1, \ldots, G^{t,j}_{\ell_t}$ be the $\mu$-grouping of $(1 - \delta)\bar{y}^{t,j}$ with respect to the block $K^t_j$ (Lemma 2.6). Note that $I^t_j$ and $\bar{y}^{t,j}$ are defined in Steps 3 and 4 of the algorithm.

We utilize the following definition.

**Definition E.2.** For $1 \leq t \leq d$ and $0 \leq j \leq \ell_t$, a subset $Q \subseteq \text{supp}(\bar{x})$ is $(t, j)$-compliant the following holds.

1. If $j < N^2$ (i.e., $j$ for which $|K^t_j| = 1$) then $w_t(Q \cap I^t_j) \leq W^*_{K^t_j}$.

2. If $j \geq N^2$ then

\[ \forall 1 \leq k \leq \tau_{t,j} : \quad |Q \cap G^{t,j}_k \cap I^t_j| \leq \mu |K^t_j| \quad \text{and} \quad w_t(Q \cap I^t_j \cap L_{K^t_j, \mu}) \leq \sum_{i \in L_{K^t_j, \mu}} (1 - \delta)\bar{y}^{t,j}_i \cdot w_t(i) + \frac{\mu}{4} W^*_{K^t_j} \cdot |K^t_j| \]

Furthermore, we say $Q$ is compliant if it is $(t, j)$-compliant for every $1 \leq t \leq d$ and $0 \leq j \leq \ell_t$.

We proceed with the following claims.

**Claim E.3.** For $1 \leq t \leq d$ and $0 \leq j \leq \ell_t$, if $R$ is $(t, j)$-compliant then Algorithm 3 succeeds in packing $J \cap I^t_j$ into $K^t_j$ in Step 3.

**Proof.** If $j < N^2$ then $|K^t_j| = 1$. Since $R$ is $(t, j)$-compliant the items of $R \cap I^t_j$ fit into the single bin in $K^t_j$. As $J \subseteq R$ it follows that the items of $J \cap I^t_j$ fit into the single bin in $K^t_j$ as well.

If $N^2 \leq j$ it holds that $|K^t_j| \geq N$. By Lemma 2.6 since $R$ is $(t, j)$-compliant, it follows that $R \cap I^t_j$ can be packed into at most

\[ (1 - \mu) |K^t_j| + 4 \cdot 4^{\mu - 2} + 2 \cdot \frac{\mu}{4} |K^t_j| = \left(1 - \frac{\mu}{2}\right) |K^t_j| + 4 \cdot 4^{\mu - 2} \leq |K^t_j| \]

bins of capacity $W^*_{K^t_j}$. Since $J \cap I^t_j \subseteq R \cap I^t_j$ it holds that the items of $J \cap I^t_j$ can also be packed into the above number of bins. Therefore, the bin packing algorithm of 25 packs the items into at most

\[ \left(1 - \frac{\mu}{2}\right) |K^t_j| + 4 \cdot 4^{\mu - 2} + \rho \log^2 |K^t_j| \leq |K^t_j| \]

bins of capacity $W^*_{K^t_j}$. \qed

For every $1 \leq t \leq d$ and $0 \leq j \leq \ell_t$ we define an event $\Psi_{t,j} = \{ R \text{ is } (t, j)\text{-compliant} \}$. 

25
Claim E.4.

\[ \Pr(\exists 1 \leq t \leq d \text{ and } 1 \leq j \leq \ell_t : -\Psi_{t,j}) < \frac{\varepsilon^2}{2} \]

As the proof of the lemma is technical, we first sketch an overview of the proof. It holds that each of the events \( \Psi_{t,j} \) consists of a few linear constraints over \( R \), of the form \( \sum_{i \in R} \bar{a}_i < L \), for some \( L > 0 \) and \( \bar{a} \in \mathbb{R}_{\geq 0}^d \). For each of these constraints, we can show that \( \mathbb{E}[\sum_{i \in R} \bar{a}_i] \) is strictly smaller than \( L \). This property follows from the Block Association and from the distribution by which items are sampled (i.e., \( \Pr(i \in R) = (1 - \delta)^2 \bar{x} \)). As the instance is restricted and \( \bar{x} \in P \), it also follows that the ratio \( \frac{\bar{a}}{\bar{x}} \) is always bounded, and thus the concentration bounds of Lemma [E.1] can be used to show a constraint is violated only with an exponentially small probability. The union bound is then used to show that the probability that one of these constraints is violated is bounded by \( \frac{\varepsilon^2}{2} \).

**Proof of Lemma [E.4]** For \( 1 \leq t \leq d \) and \( 0 \leq j \leq \min\{N^2 - 1, \ell_t\} \), by Lemma [C.1] it holds that there is \( i^* \in I \) such that

\[
\sum_{i \in I_t^j \setminus \{i^*\}} (1 - \delta)\bar{x}_i \cdot w_l(i) \leq \sum_{i \in I} (1 - \delta)\bar{y}_{i,j}^t \cdot w_l(i) \leq (1 - \delta)W_{K_j^t}^k.
\]

Since \( \Pr(i \in R) = (1 - \delta)^2 \bar{x}_i \) for any \( i \in I \), it follows that

\[ \mathbb{E}[w_l(R \cap I_t^j \setminus \{i^*\})] = \sum_{i \in I_t^j \setminus \{i^*\}} (1 - \delta)^2 \bar{x}_i \cdot w_l(i) \leq (1 - \delta)^2W_{K_j^t}^k. \]

As \( I_t^j \subseteq \text{supp}\{\bar{y}_{i,j}^t\} \subseteq \{i \in I | w_l(i) \leq \gamma \cdot W_{K_j^t}^k\} \) (due to Lemma [C.1] and Definition [E.3]), and \( \gamma < \frac{\varepsilon}{2} \), it follows that \( \mathbb{E}[w_l(R \cap I_t^j)] \leq (1 - \frac{\varepsilon}{2}) \cdot W_{K_j^t}^k \). Hence, \( \mathbb{E}\left[\sum_{i \in R} 1_{i \in I_t^j} \cdot w_l(i) \frac{1}{\gamma \cdot W_{K_j^t}^k}\right] \leq (1 - \frac{\varepsilon}{2}) \cdot \frac{1}{\gamma} \cdot \frac{1}{4} \frac{1}{160} \)

Following the same argument and by Lemma [E.1] it holds that

\[
\Pr(-\Psi_{t,j}) = \Pr(w_l(R \cap I_t^j) > W_{K_j^t}^k) = \Pr\left(\frac{\sum_{i \in I_t^j} w_l(i)}{\gamma \cdot W_{K_j^t}^k} > \gamma^{-1}\right)
\]

\[
\leq \Pr\left(\sum_{i \in R} 1_{i \in I_t^j} \cdot w_l(i) \frac{\gamma \cdot W_{K_j^t}^k}{\gamma^{-1}} \right) \geq \left(1 + \frac{\varepsilon}{2}\right) \left(1 - \frac{\varepsilon}{2}\right) \frac{1}{\gamma} \leq \exp\left(-\frac{3}{80} \left(1 - \frac{\varepsilon}{2}\right) \gamma^{-1}\right) \leq \exp\left(-\frac{3}{160} \gamma^{-1}\right).
\]

For \( 1 \leq t \leq d \) and \( N^2 \leq j \leq \ell_t \), by Lemmas [C.1] and [E.1] for any \( 1 \leq k \leq \tau_{t,j} \) it holds that

\[ \mathbb{E}\left[|R \cap I_t^j \cap G_{K_j^t}^{t,j}|\right] \leq (1 - \delta) (\mu |K_j^t| + 2) \leq \left(1 - \frac{\varepsilon}{2}\right) \mu |K_j^t|, \]

where the last transition holds since \( |K_j^t| \geq N \) and thus \( |K_j^t| > 32 \cdot \delta^{-2} = 8 \cdot \delta^{-1} \cdot \mu^{-1} \). Hence, following Lemma [E.1] it holds that

\[
\Pr\left(|R \cap I_t^j \cap G_{K_j^t}^{t,j}| > \mu |K_j^t|\right) \leq \Pr\left(\sum_{i \in R} 1_{i \in I_t^j \cap G_{K_j^t}^{t,j}} \geq \left(1 + \frac{\varepsilon}{2}\right) \left(1 - \frac{\varepsilon}{2}\right) \mu |K_j^t|\right)
\]

\[
\leq \exp\left(-\frac{3}{80} \left(1 - \frac{\varepsilon}{2}\right) \mu |K_j^t|\right) = \exp\left(-\frac{3}{160} \mu |K_j^t|\right) \cdot \left(1 - \frac{\varepsilon}{2}\right) \cdot \mu |K_j^t|.
\]

Similarly, by Lemma [C.1] there is \( i^* \in I \) such that \( \sum_{i \in I_t^j \cap L_{K_j^t}^{t,j}} (1 - \delta)\bar{x}_i \cdot w_l(i) \leq \sum_{i \in L_{K_j^t}^{t,j}} (1 - \delta)\bar{y}_{i,j}^t \cdot w_l(i) + \mu W_{K_j^t}^k \leq \sum_{i \in L_{K_j^t}^{t,j}} (1 - \delta)\bar{y}_{i,j}^t \cdot w_l(i) + \frac{\mu}{4} |K_j^t| W_{K_j^t}^k.
\]

\[14\text{For } i \in I \text{ and } S \subseteq I \text{ we use } 1_{i \in S} = 1 \text{ if } i \in s \text{ and } 1_{i \in S} = 0 \text{ otherwise.} \]
where the last transition requires \(|K_j^*| \geq 4\), which holds due to the parameter selection. Let \(\zeta = \sum_{i \in L_{K_j^*}} (1 - \delta) y_i \cdot w_i(i) + \frac{2}{|K_j^*|} \cdot W_{K_j^*}^*\). Subsequently, by Lemma E.1 we have \(\mathbb{E} [w_i (I_j \cap L_{K_j^*} \cap R)] \leq (1 - \delta) \zeta\) and

\[
\Pr \left( w_i (I_j \cap L_{K_j^*} \cap R) > \zeta \right) \leq \Pr \left( \sum_{i \in R} \mathbb{1}_{i \in I_j \cap L_{K_j^*} \cap R} \cdot \frac{w_i(i)}{W_{K_j^*}^*} \geq (1 + \delta) (1 - \delta) \frac{\zeta}{W_{K_j^*}^*} \right)
\]

\[
\leq \exp \left( -\frac{\delta^3}{20} \cdot (1 - \delta) \cdot \frac{\zeta}{W_{K_j^*}^*} \right) \leq \exp \left( -\frac{\delta^4}{640} \cdot |K_j^*| \right). \tag{23}
\]

The last inequality used \(\zeta \geq \frac{\delta}{16} \cdot |K_j^*| \cdot W_{K_j^*}^* = \frac{\delta}{16} \cdot |K_j^*| \cdot W_{K_j^*}^*\). By (22) and (23), it follows that

\[
\Pr (\neg \Psi_{t,j}) \leq (1 + \tau_{t,j}) \exp \left( -\frac{\delta^4}{640} \cdot |K_j^*| \right) \leq (16 \cdot \delta^{-2} + 2) \exp \left( -\frac{\delta^4}{640} \cdot |K_j^*| \right), \tag{24}
\]

where the second inequality follows from \(\tau_{t,j} \leq \mu^{-2} + 1 = 16 \cdot \delta^{-2} + 1\) (Lemma 2.6).

By (21) and (24), it holds that

\[
\Pr (\exists t \leq d \text{ and } 0 \leq j \leq \ell_t : \neg \Psi_{t,j}) \leq d \cdot N^2 \exp \left( -\frac{\delta^3}{160 \gamma^{-1}} \right) + d \cdot \sum_{j=1}^{\infty} N^2 \cdot (16 \cdot \delta^{-2} + 2) \cdot \exp \left( -\frac{\delta^4}{640} \cdot N^2 \right) < \frac{\varepsilon^2}{2},
\]

where the last inequality is by (19) and (18). \(\Box\)

**Claim E.5.** If \(f\) is monotone (or modular) then the expected value of the returned solution is at least \(c \cdot (1 - \varepsilon) \cdot \mathbb{OPT}\).

**Proof.** Since \(f\) is monotone we have \(J = \eta_f(R) = R\).

Denote \(f_{\max} = \max_{i \in I} f(\{i\}) - f(\emptyset)\). Since \(R\) is \((N, \xi)\)-restricted \(d\)-MCKP instance it holds that \(f_{\max} \leq \frac{\mathbb{OPT}}{\xi}\). Using (20) and Lemma E.1 it holds that

\[
\Pr \left( f(R) < c \cdot (1 - \varepsilon^2)^5 \cdot \mathbb{OPT} \right) \leq \exp \left( -\frac{(1 - \varepsilon^2)^4 \cdot \varepsilon^2 \cdot \mathbb{OPT}}{20 \cdot f_{\max}} \right)
\]

\[
\leq \exp \left( -\frac{(1 - \varepsilon^2)^4 \cdot \varepsilon^6 \cdot c}{20 \cdot \xi} \right) \leq \frac{\varepsilon^2}{2}. \tag{25}
\]

Let \(\Phi\) be the event in which \(\Psi_{t,j}\) occurs for every \(1 \leq t \leq d\) and \(0 \leq j \leq \ell_t\), and \(f(R) \geq c \cdot (1 - \varepsilon^2)^5 \cdot \mathbb{OPT}\). By Claim E.4 and (25) it holds that \(\Pr (\Phi) \geq (1 - \varepsilon^2)\). By Claim E.3 if \(\Phi\) occurs then the algorithm returns \(J = R\). The solution along with corresponding assignments to the MKCs (Step 3 of the algorithm). Therefore, the expected value of the solution returned by the algorithm at least

\[
\Pr(\Phi) \cdot \mathbb{E} [f(R) \mid \Phi] \geq (1 - \varepsilon^2) \cdot c \cdot (1 - \varepsilon^2)^5 \cdot \mathbb{OPT} \geq c \cdot (1 - \varepsilon) \cdot \mathbb{OPT}
\]

Claim E.5 completes the proof for the case that \(f\) is monotone. We are left to handle the case in which \(f\) is non-monotone. As we assume \((\mathcal{I}, f)\) is a valid pair, it follows that \(\mathcal{I} = 2^I\).

In the following we show that our algorithm behaves similarly to a contention resolution scheme, and prove the approximation guarantee similarly to [12]. It seems possible to use the theorems of [12] directly to show the correctness, instead of adjusting their proofs to our setting. However, the technical overhead required to present our procedure in a way which matches the definitions in [12] appears to surpass the benefits of such an approach. The technical overhead stems from the fact that our rounding depends on the vectors \(\tilde{y}^S\) and not just on \(x\), the rounding does not strictly adhere to the definition of monotonicity in [12], and since the polytope used is only an approximate representation of the problem.

The following is a variant of Claim E.4 and is proved similarly.
Claim E.6. Assume $\mathcal{I} = 2^I$. Then for any $Q \subseteq \text{supp}(\hat{x})$, $|Q| \leq 1$ it holds that

$$\Pr(\exists 1 \leq t \leq d \text{ and } 1 \leq j \leq \ell_t : \neg \Psi_j \mid Q \subseteq R) < \frac{\epsilon^2}{2}.$$ 

Proof. As $\mathcal{I} = 2^I$, by Lemma [E.1] it follows that the events $(i \in R)_{i \in I}$ are independent.

For $1 \leq t \leq d$ and $0 \leq j \leq \min\{N^2 - 1, \ell_t\}$, by Lemma [C.1] it holds that there is $i^* \in I$ such that

$$\sum_{i \in I^t \setminus \{i^*\}} (1 - \delta)\hat{x}_i \cdot w_t(i) \leq \sum_{i \in I}(1 - \delta)\hat{y}^{i,j}_t \cdot w_t(i) \leq (1 - \delta)W^*_j.$$

Since $\Pr(i \in R) = (1 - \delta)^2\hat{x}_i$ for any $i \in I$, it follows that

$$\mathbb{E}[w_t(R \cap I^t \setminus \{i^*\} \setminus Q)] = \sum_{i \in I^t \setminus \{i^*\} \setminus Q} (1 - \delta)^2\hat{x}_i \cdot w_t(i) \leq (1 - \delta)^2W^*_j.$$

As $I^t_j \subseteq \text{supp}\{\hat{y}^{i,j}\} \subseteq \{i \in I \mid w_t(i) \leq \gamma \cdot W^*_j\}$ (due to Lemma [C.1] and Definition [E.3], and $\gamma < \frac{\delta}{2}$ it follows that,

$$\mathbb{E}[w_t(R \cap I^t_j \setminus Q)] \leq \gamma \cdot W^*_j + (1 - \delta)^2W^*_j \leq \left(1 - \frac{3 \cdot \delta}{4}\right) \cdot W^*_j.$$ 

Hence, $\mathbb{E}\left[\sum_{i \in R \cap Q} 1_{i \in I^t_j} \cdot w_t(i) \frac{1}{\gamma \cdot W^*_j}\right] \leq \left(1 - \frac{3 \cdot \delta}{4}\right) \cdot \frac{1}{\gamma}$. Recall that as $\mathcal{I} = 2^I$, by Lemma [E.1] it holds that the sum $\sum_{i \in R \cap Q} 1_{i \in I^t_j} \cdot w_t(i) \frac{1}{\gamma \cdot W^*_j}$ is independent of the event $Q \subseteq R$. Thus,

$$\Pr(\neg \Psi_{t,j} \mid Q \subseteq R) = \Pr(w_t(R \cap I^t_j \setminus Q) > W^*_j - w(I^t \cap Q) \mid Q \subseteq R)$$

$$\leq \Pr\left(\sum_{i \in R \cap Q} 1_{i \in I^t_j} \cdot w_t(i) \frac{1}{\gamma \cdot W^*_j} > \gamma^{-1} \left(1 - \frac{\delta}{4}\right)\right)$$

$$\leq \Pr\left(\sum_{i \in R \cap Q} 1_{i \in I^t_j} \cdot w_t(i) \frac{1}{\gamma \cdot W^*_j} \geq \left(1 + \frac{\delta}{2}\right) \left(1 - \frac{3 \cdot \delta}{4}\right) \frac{1}{\gamma}\right)$$

$$\leq \exp\left(-\frac{\delta^3}{80} \left(1 - \frac{3 \cdot \delta}{4}\right) \cdot \gamma^{-1}\right)$$

$$\leq \exp\left(-\frac{\delta^3}{160} \cdot \gamma^{-1}\right),$$

where first equality follows from the definition of $\Psi_{t,j}$, the first inequality uses $|Q| \leq 1$ and $\gamma < \frac{\delta}{4}$ as well as the independence of the sum from $Q \subseteq R$, the third inequality is by Lemma [E.1].

For $1 \leq t \leq d$ and $N^2 \leq j \leq \ell_t$, by Lemmas [C.1] and [E.1] for any $1 \leq k \leq \tau_{t,j}$ it holds that

$$\mathbb{E}\left[|R \cap I^t_j \cap G^{t,j}_k \setminus Q|\right] \leq (1 - \delta) \left(\mu|K^j_t| + 2\right) \leq (1 - \frac{3 \cdot \delta}{4}) \mu|K^j_t|,$$

where the last transition holds since $|K^j_t| \geq N$ and thus $|K^j_t| > 32 \cdot \delta^{-2} = 8 \cdot \delta^{-1} \cdot \mu^{-1}$. Hence, following an argument similar to that of (26), we have,

$$\Pr\left(|R \cap I^t_j \cap G^{t,j}_k| > \mu|K| \mid Q \subseteq R\right)$$

$$= \Pr\left(|R \cap I^t_j \cap G^{t,j}_k \setminus Q| > \mu|K| - |Q \cap I^t_j \cap G^{t,j}_k| \mid Q \subseteq R\right)$$

$$\leq \Pr\left(|R \cap I^t_j \cap G^{t,j}_k \setminus Q| > \mu|K| \left(1 - \frac{\delta}{4}\right)\right)$$

$$\leq \Pr\left(\sum_{i \in R \cap Q} 1_{i \in I^t_j \cap G^{t,j}_k} \geq \left(1 + \frac{\delta}{2}\right) \left(1 - \frac{3 \cdot \delta}{4}\right) \mu|K^j_t|\right)$$

$$\leq \exp\left(-\frac{\delta^3}{80} \left(1 - \frac{\delta}{2}\right) \mu|K^j_t|\right) = \exp\left(-\frac{\delta^4}{640} \cdot |K^j_t|\right).$$
Similarly, by Lemma [C.1] there is $i^* \in I$ such that
\[
\sum_{i \in I^* \cap L_{K_i^*} \setminus \{i^*\}} (1 - \delta) \bar{x}_i \cdot w_t(i) \leq \sum_{i \in L_{K_i^*}} (1 - \delta) \bar{y}_{i}^j \cdot w_t(i).
\]

Define $\zeta = \sum_{i \in L_{K_i^*}} (1 - \delta) \bar{y}_{i}^j \cdot w_t(i) + \frac{\mu}{8} |K_i^*| \cdot W_{K_i^*}^*$. Thus,
\[
\sum_{i \in I^* \cap L_{K_i^*} \setminus \{i^*\}} (1 - \delta) \bar{x}_i \cdot w_t(i) \leq \sum_{i \in L_{K_i^*}} (1 - \delta) \bar{y}_{i}^j \cdot w_t(i) + \frac{\mu}{8} |K_i^*| W_{K_i^*}^* = \zeta - \frac{\mu}{8} |K_i^*| W_{K_i^*}^*,
\]
where the last inequality requires $|K_i^*| \geq 8$ which holds due to the parameter selection. Subsequently, by Lemma [E.1] we have $\mathbb{E} \left[w_t(I_j \cap L_{K_i^*} \cap R \setminus Q)\right] \leq (1 - \delta) \left(\zeta - \frac{\mu}{8} |K_i^*| W_{K_i^*}^*\right)$ and
\[
\Pr \left(w_t(I_j \cap L_{K_i^*} \cap R) > \zeta \mid Q \subseteq R\right) = \Pr \left(w_t(I_j^f \cap L_{K_i^*} \cap R \setminus Q) > \zeta - w_t(I_j^f \cap L_{K_i^*} \cap Q) \mid Q \subseteq R\right) \\
\leq \Pr \left(w_t(I_j^f \cap L_{K_i^*} \cap R \setminus Q) > \zeta - \mu W_{K_i^*}^*\right) \\
\leq \Pr \left(\sum_{i \in R \cap Q} \mathbb{1}_{i \in I^* \cap L_{K_i^*}} \cdot \frac{w_t(i)}{\mu W_{K_i^*}^*} \geq (1 + \delta)(1 - \delta) \frac{\zeta - \frac{\mu}{8} |K_i^*| \cdot W_{K_i^*}^*}{\mu W_{K_i^*}^*}\right) \\
\leq \exp \left(-\frac{\delta^2}{20} \cdot (1 - \delta) \cdot \frac{\zeta - \frac{\mu}{8} |K_i^*| \cdot W_{K_i^*}^*}{\mu W_{K_i^*}^*}\right) \leq \exp \left(-\frac{\delta^4}{640} \cdot |K_i^*|\right). \tag{28}
\]

The last inequality used $\zeta \geq \frac{\mu}{4} |K_i^*| \cdot W_{K_i^*}^*$. By (27) and (28), it follows that
\[
\Pr(\neg \Psi_{t,j} \mid Q \subseteq R) \leq (1 + \tau_{t,j}) \exp \left(-\frac{\delta^4}{640} |K_i^*|\right) \leq (16 \cdot \delta^{-2} + 2) \exp \left(-\frac{\delta^4}{640} |K_i^*|\right), \tag{29}
\]
where the second inequality follows from $\tau_{t,j} \leq \mu^{-2} + 1 = 16 \cdot \delta^{-2} + 1$ (Lemma [2.6]).

By (20) and (29), it holds that
\[
\Pr(\exists t \leq d \text{ and } 0 \leq j \leq \ell_t: \neg \Psi_{t,j} \mid Q \subseteq R) \\
\leq d \cdot \sum_{Q \subseteq I} N^2 \exp \left(-\frac{\delta^3}{160} \gamma^{-1}\right) + d \cdot \sum_{j=1}^{\infty} N^2 (16 \cdot \delta^{-2} + 2) \cdot \exp \left(-\frac{\delta^4}{640} \cdot N^2\right) < \frac{\epsilon^2}{2},
\]
where the last inequality is by (19) and (18).

W.l.o.g we assume $I = \{1, 2, \ldots, n\}$ and that the order in which $\eta_f$ iterates over the items is $1, 2, \ldots, n$. Denote $[i] = \{1, 2, \ldots, i\}$ for $i \in I$, $[0] = \emptyset$, and
\[
\forall Q \subseteq I: \quad \kappa(Q) = \begin{cases} 1 & Q \text{ is compliant} \\ 0 & \text{otherwise} \end{cases}.
\]

**Claim E.7.** Assume $I = 2^I$. For any $i \in I$ it holds that
\[
\mathbb{E} \left[\kappa(R) \cdot (f(J \cap [i]) - f(J \cap [i - 1]))\right] \geq (1 - \varepsilon^2) \mathbb{E} \left[f(R \cap [i]) - f(R \cap [i - 1])\right].
\]

29
Hence \( \pi \) where the first inequality is by Claims E.4 and E.7, and the last inequality is due to (20).

As \( -763 \) and (31).

where the second inequality is due to Claim E.6. The statement of the claim is obtained by combining (30) and (31).

In the following we use the FKG inequality. We refer the reader to the relevant chapter in [2] for the definition of lattice and log-supermodular functions. Since the events \( (i \in R)_{i \in I} \) are independent, it follows that for any \( X_1, X_2 \subseteq I \setminus \{i\} \) we have,

\[
\pi(X_1) \cdot \pi(X_2) = \pi(X_1 \cup X_2) \cdot \pi(X_1 \cap X_2).
\]

Hence \( \pi \) is log supermodular over the lattice of subsets of \( I \setminus \{i\} \). Additionally, the functions \( \kappa(X \cup \{i\}) \) and \( \max \{0, f_{X \cap [i-1]}(\{i\})\} \) is decreasing as functions of \( X \). Finally, it holds that \( \sum_{X \subseteq \Gamma \setminus \{i\}} \pi(X) = 1 \).

Thus, by the FKG inequality (Theorem 2.1 in [2]) it holds that,

\[
\Pr(Q \subseteq R) \sum_{X \subseteq \Gamma \setminus \{i\}} \pi(X) \cdot \kappa(X \cup \{i\}) \cdot \max \{0, f_{X \cap [i-1]}(\{i\})\} \geq \Pr(Q \subseteq R) \cdot \left( \sum_{X \subseteq \Gamma \setminus \{i\}} \pi(X) \cdot \kappa(X \cup \{i\}) \right) \cdot \left( \sum_{X \subseteq \Gamma \setminus \{i\}} \pi(X) \cdot \max \{0, f_{X \cap [i-1]}(\{i\})\} \right)
\]

\[
= \Pr(Q \subseteq R) \cdot \mathbb{E}[\kappa(R) \mid Q \subseteq R] \cdot \mathbb{E}[\max \{0, f_{R \cap [i-1]}(\{i\})\} \mid Q \subseteq R] \geq (1 - \varepsilon^2) \cdot \mathbb{E}[\max \{0, f_{R \cap [i-1]}(\{i\})\}] \geq (1 - \varepsilon^2) \cdot \mathbb{E}[f(R \cap [i]) - f(R \cap [i-1])],
\]

where the second inequality is due to Claim E.6. The statement of the claim is obtained by combining (30) and (31).

Claim E.8. If \( f \) is non-monotone then the expected value of the solution returned by the algorithm is at least \( c \cdot (1 - \varepsilon) \cdot \text{OPT} \).

Proof. As \( f \) is non-monotone we have \( I = 2^I \). By Claim E.3 the expected value of the solution returned by the algorithm is at least \( \mathbb{E}[\kappa(R) \cdot f(J)] \). By Claim E.7 we have,

\[
\mathbb{E}[\kappa(R) \cdot f(J)] = \mathbb{E} \left[ \kappa(R) \left( f(\emptyset) + \sum_{i=1}^{n} (f(J \cap [i]) - f(J \cap [i-1])) \right) \right]
\]

\[
= \mathbb{E}[\kappa(R) \cdot f(\emptyset)] + \sum_{i=1}^{n} \mathbb{E}[\kappa(R) \cdot (f(J \cap [i]) - f(J \cap [i-1]))]
\]

\[
\geq (1 - \varepsilon^2) \cdot f(\emptyset) + (1 - \varepsilon^2) \cdot \sum_{i=1}^{n} \mathbb{E}[f(R \cap [i]) - f(R \cap [i-1])]
\]

\[
\geq (1 - \varepsilon^2) \cdot \mathbb{E}[f(R)] \geq (1 - \varepsilon) \cdot c \cdot \text{OPT},
\]

where the first inequality is by Claims E.4 and E.7 and the last inequality is due to (20).

\( \square \)
The statement of the lemma follows from Claims \[E.5\] and \[E.8\].

## F Properties of Submodular Functions

**Lemma F.1.** Let \( f : 2^I \to \mathbb{R}_{\geq 0} \) be a submodular function and let \( S = \{s_1, \ldots, s_\ell\} \subseteq I \), \( |S| = \ell \), such that \( f(\{s_1, \ldots, s_r\}) = \max_{r-1 < k \leq \ell} f(\{s_1, \ldots, s_{r-1}\} \cup \{s_k\}) \) for every \( 1 \leq r \leq \ell \). Also, let \( E \in \mathbb{N} \) and \( S_E = \{s_r \mid 1 \leq r \leq \min\{E, \ell\}\} \). Then for every \( i \in S \setminus S_E \) it holds that \( f(S_E \cup \{i\}) - f(S_E) \leq \frac{f(S_E)}{E} \).

**Proof.** If \( E \geq \ell \) then \( S \setminus S_E = \emptyset \) and the statement trivially holds. Otherwise, for every \( i \in S \setminus S_E \) it holds that

\[
\begin{align*}
f(S_E \cup \{i\}) - f(S_E) &= \frac{1}{E} \sum_{r=1}^{E} (f(S_E \cup \{i\}) - f(S_E)) \\
&\leq \frac{1}{E} \sum_{r=1}^{E} (f(\{s_1, \ldots, s_{r-1}\} \cup \{i\}) - f(\{s_1, \ldots, s_{r-1}\})) \\
&\leq \frac{1}{E} \sum_{r=1}^{E} (f(\{s_1, \ldots, s_{r-1}\} \cup \{s_r\}) - f(\{s_1, \ldots, s_{r-1}\})) \\
&\leq \frac{1}{E} (f(S_E) - f(\emptyset)) \\
&\leq \frac{f(S_E)}{E}.
\end{align*}
\]

The first inequality follows from the submodularity of \( f \). The second inequality follows from \( f(\{s_1, \ldots, s_r\}) = \max_{r-1 < k \leq \ell} f(\{s_1, \ldots, s_{r-1}\} \cup \{s_k\}) \) for every \( 1 \leq r \leq \ell \). \[\square\]

**Lemma F.2.** Let \( f : 2^I \to \mathbb{R} \) be a set function and \( R \subseteq I \). Define \( g : 2^I \to \mathbb{R} \) by \( g(S) = f(S \cup R) \) for any \( S \subseteq I \). Then,

1. If \( f \) is submodular then \( g \) is submodular.
2. If \( f \) is monotone then \( g \) is monotone.
3. If \( f \) is modular then \( g \) is modular.

**Proof.**

1. Assume \( f \) is submodular. Let \( S, T \subseteq I \). Then,

\[
g(S) + g(T) = f(R \cup S) + f(R \cup T) \\
\geq f((R \cup S) \cup (T \cup R)) + f((R \cup S) \cap (T \cup R)) \\
= f(R \cup (S \cup T)) + f(R \cup (S \cap T)) \\
= g(S \cup T) + g(S \cap T)
\]

Thus \( g \) is submodular.

2. Assume \( f \) is monotone and let \( S \subseteq T \subseteq I \). Then \( R \cup S \subseteq R \cup T \) and therefore,

\[
g(S) = f(R \cup S) \leq f(R \cup T) = g(T).
\]

Thus \( g \) is monotone.

3. If \( f \) is modular then both \( f \) and \( -f \) are submodular. Thus, by the first property both \( g \) and \( -g \) are submodular and therefore \( g \) is modular. \[\square\]
G The Multilinear Extension of a Modular Function

In this section we show several well known properties of the multilinear extension of a linear function.

**Lemma G.1.** Let \( f : 2^I \to \mathbb{R} \) such that \( f(S) = a + \sum_{i \in S} \bar{p}_i \) where \( a \in \mathbb{R} \) and \( \bar{p} \in \mathbb{R}^I \). Then, \( F \) the multilinear extension of \( f \) satisfies

\[
\forall \bar{x} \in [0, 1]^I :\quad F(\bar{x}) = a + \bar{x} \cdot \bar{p}
\]

**Proof.** We use the notation

\[
\forall S \subseteq I, \ i \in I :\quad \mathbb{1}_{i \in S} = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}
\]

Let \( \bar{x} \in [0, 1]^I \), then

\[
F(\bar{x}) = \mathbb{E}_{S \sim \bar{x}} [f(S)]
\]

\[
= \mathbb{E}_{S \sim \bar{x}} \left[ a + \sum_{i \in S} \bar{p}_i \right]
\]

\[
= a + \mathbb{E}_{S \sim \bar{x}} \left[ \sum_{i \in I} \mathbb{1}_{i \in S} \cdot \bar{p}_i \right]
\]

\[
= a + \sum_{i \in I} \bar{p}_i \cdot \mathbb{E}_{S \sim \bar{x}} [\mathbb{1}_{i \in S}]
\]

\[
= a + \sum_{i \in I} \bar{x}_i \cdot \bar{p}_i = a + \bar{x} \cdot \bar{p}
\]

\( \square \)