A $(1 - e^{-1} - \varepsilon)$-Approximation for the Monotone Submodular Multiple Knapsack Problem

Yaron Fairstein
Computer Science Department, Technion, Haifa, Israel
yyfairstein@gmail.com

Ariel Kulik
Computer Science Department, Technion, Haifa, Israel
kulik@cs.technion.ac.il

Joseph (Seffi) Naor
Computer Science Department, Technion, Haifa, Israel
naor@cs.technion.ac.il

Danny Raz
Computer Science Department, Technion, Haifa, Israel
danny@cs.technion.ac.il

Hadas Shachnai
Computer Science Department, Technion, Haifa, Israel
hadas@cs.technion.ac.il

Abstract

We study the problem of maximizing a monotone submodular function subject to a Multiple Knapsack constraint (SMKP) . The input is a set I of items, each associated with a non-negative weight, and a set of bins having arbitrary capacities. Also, we are given a submodular, monotone and non-negative function f over subsets of the items. The objective is to find a subset of items $A \subseteq I$ and a packing of these items in the bins, such that $f(A)$ is maximized.

SMKP is a natural extension of both Multiple Knapsack and the problem of monotone submodular maximization subject to a knapsack constraint. Our main result is a nearly optimal polynomial time $(1 - e^{-1} - \varepsilon)$-approximation algorithm for the problem, for any $\varepsilon > 0$. Our algorithm relies on a refined analysis of techniques for constrained submodular optimization combined with sophisticated application of tools used in the development of approximation schemes for packing problems.

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

Keywords and phrases Sumodular Optimization, Multiple Knapsack, Randomized Rounding

Digital Object Identifier 10.4230/LIPIcs.ESA.2020.44

Related Version A full version of the paper is available at `https://arxiv.org/abs/2004.12224`.

Funding *Joseph (Seffi) Naor*: This research was supported in part by US-Israel BSF grant 2018352 and by ISF grant 2233/19 (2027511).

1 Introduction

Submodular optimization has recently attracted much attention as it provides a unifying framework capturing many fundamental problems in combinatorial optimization, economics, algorithmic game theory, networking, and other areas. Furthermore, submodularity also captures many real world practical applications where economy of scale is prevalent. Classic examples of submodular functions are coverage functions [9], matroid rank functions [3] and graph cut functions [10]. A recent survey on submodular functions can be found in [1].

28th Annual European Symposium on Algorithms (ESA 2020).
Editors: Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders; Article No. 44; pp. 44:1–44:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Submodular functions are defined over sets. Given a ground set I, a function $f : 2^I \to \mathbb{R}_{\geq 0}$ is called *submodular* if for every $A \subseteq B \subseteq I$ and $u \in I \setminus B$, $f(A+u)-f(A) \geq f(B+u)-f(B)$.[1] This reflects the diminishing returns property: the marginal value from adding $u \in I$ to a solution diminishes as the solution set becomes larger. A set function $f : 2^I \to \mathbb{R}$ is *monotone* if for any $A \subseteq B \subseteq I$ it holds that $f(A) \leq f(B)$. While in many cases, such as coverage and matroid rank function, the submodular function is monotone, this is not always the case (cut functions are a classic example).

The focus of this work is optimization of monotone submodular functions. In [19] Nemhauser and Wolsey presented a greedy based $(1 - e^{-1})$-approximation for maximizing a monotone submodular function subject to a cardinality constraint, along with a matching lower bound in the oracle model. A $(1 - e^{-1})$ hardness of approximation bound is also known for the problem under P $\neq$ NP, due to the hardness of max-k-cover [9] which is a special case. The greedy algorithm of [19] was later generalized to monotone submodular optimization with a knapsack constraint [16, 21].

A major breakthrough in the field was the continuous greedy algorithm presented in [22]. Initially used to derive a $(1 - e^{-1})$-approximation for maximizing a monotone submodular function subject to a matroid constraint, the algorithm has become a primary tool in the development of monotone submodular maximization algorithms subject to various other constraints. These include d-dimensional knapsack constraints [17], and combinations of d-dimensional knapsack and matroid constraints [7]. A variant of the continuous greedy algorithm for non-monotone functions is given in [11].

In the *multiple knapsack problem* (MKP) we are given a set of items, where each item has a weight and a profit, and a set of bins of arbitrary capacities. The objective is to find a packing of a subset of the items that respects the bin capacities and yields a maximum profit. The problem is one of the most natural extensions of the classic Knapsack problem that arises in the context of Virtual Machine (VM) allocation in cloud computing. The practical task is to assign VMs to physical machines such that capacity constraints are satisfied, while maximizing the profit of the cloud provider. A submodular cost function allows cloud providers to offer complex cost models to high-volume customers, where the price customers pay for each VM can depend on the overall number of machines used by the customer.

A *polynomial time approximation scheme* for MKP was first presented by Chekuri and Khanna [5]. The authors also ruled out the existence of a *fully polynomial time approximation scheme* for the problem. An *efficient* polynomial time approximation scheme was later developed by Jansen [14, 15].

1.1 Our Results

In this paper we consider the *submodular multiple knapsack problem* (SMKP). The input consists of a set of n items I and m bins B. Each item $i \in I$ is associated with a weight $w_i \geq 0$, and each bin $b \in B$ has a capacity $W_b \geq 0$. We are also given an oracle to a non-negative monotone submodular function $f : 2^I \to \mathbb{R}_{\geq 0}$. A feasible solution to the problem is a tuple of m subsets $(A_b)_{b \in B}$ such that for every $b \in B$ it holds that $\sum_{i \in A_b} w_i \leq W_b$. The value of a solution $(A_b)_{b \in B}$ is $f\left(\bigcup_{b \in B} A_b\right)$. The goal is to find a feasible solution of maximum value.[2]

[1] Equivalently, for every $A, B \subseteq I$: $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.

[2] We note that the set of bins B is part of the input for SMKP, thus the number of bins is non-constant. This is one difference between SMKP and the problem of maximizing a submodular set function subject to d knapsack constraints (or, a d-dimensional knapsack constraint) where d is fixed (for more details see, e.g., [17]).

The problem is a natural generalization of both Multiple Knapsack [5] (where f is modular or linear), and the problem of monotone submodular maximization subject to a knapsack constraint [21] (where $m = 1$). Our result is stated in the next theorem.

▶ **Theorem 1.** *For any $\varepsilon > 0$, there is a randomized $(1 - e^{-1} - \varepsilon)$-approximation algorithm for SMKP.*

As mentioned above, a $(1 - e^{-1})$ hardness of approximation bound is known for the problem under P $\neq$ NP, due to the hardness of max-k-cover [9] which is a special case of SMKP. This is a vast improvement over previous results. Feldman presented in [12] a $\left(\frac{e-1}{3e-1} - o(1)\right) \approx 0.24$-approximation for the special case of identical bin capacities, along with a $\frac{1}{9}$-approximation for general capacities. To the best of our knowledge, this is the best known approximation ratio for the problem.[3]

In very recent work, carried out independently of our work, Sun et al. [20] present a *deterministic* greedy based $(1 - e^{-1} - \varepsilon)$-approximation for the special case of identical bins. They also derive a randomized $(1 - e^{-1} - \varepsilon)$-approximation for the special case where the ratio between the capacity of the largest and smallest bins is a constant. In this paper we give a randomized algorithm for the most general case, based on a different approach (as described below).

1.2 Tools and Techniques

Our algorithm relies on a refined analysis of techniques for submodular optimization subject to d-dimensional knapsack constraints [17, 4, 7], combined with sophisticated application of tools used in the development of approximation schemes for packing problems [8].

At the heart of our algorithm lies the observation that SMKP for a large number of identical bins (i.e., $\forall b \in B,\ W_b = W$ for some $W \geq 0$) can be easily approximated via a reduction to the problem of maximizing a submodular function subject to a 2-dimensional knapsack constraint (see, e.g., [17]). Given such an SMKP instance and $\varepsilon > 0$, we partition the items to *small* and *large*, where an item $i \in I$ is small if $w_i \leq \varepsilon W$ and large otherwise. We further define a *configuration* to be a subset of large items which fits into a single bin, and let $\mathcal{C}$ be the set of all configurations. It follows that for fixed $\varepsilon > 0$, the number of configurations is polynomial.

Using the above we define a new submodular optimization problem, to which we refer as the *block-constraint* problem. We define a new universe E which consists of all configurations $\mathcal{C}$ and all small items, $E = \mathcal{C} \cup \{\{i\} |\ i \text{ is small}\}$. We also define a new submodular function $g : 2^E \to \mathbb{R}_{\geq 0}$ by $g(T) = f\left(\bigcup_{A \in T} A\right)$. Now, we seek a subset of elements $T \subseteq E$ such that T has at most $m = |B|$ configurations, i.e., $|T \cap \mathcal{C}| \leq m$, and the total weight of sets selected is at most $m \cdot W$; namely, $\sum_{A \in T} w(A) \leq m \cdot W$, where $w(A) = \sum_{i \in A} w_i$.

It is easy to see that the optimal value of the block-constraint problem is at least the value of the optimum for the original instance. Moreover, a solution T for the block-constraint problem can be used to generate a solution for the SMKP instance with only a small loss in value. As there are no more than m configurations, and all other items are small, the items in T can be easily packed into $(1+\varepsilon)m+1$ bins of capacity W using First Fit. Then, it is possible

[3] Sun et al. [20] indicate that a $\left(1 - e^{1-e^{-1}} - o(1)\right) \approx 0.468$-approximation for the problem can be derived using the techniques of [4]. We note that this derivation is non-trivial (no details were given in [4]).

to remove $\varepsilon m+1$ of the bins while maintaining at least $\frac{m}{\varepsilon m+1} \geq \frac{1}{1+2\varepsilon}$ of the solution value, for $m \geq \frac{1}{\varepsilon}$. Once these $\varepsilon m+1$ bins are removed, we have a feasible solution for the SMKP instance. The block-constraint problem can be viewed as monotone submodular optimization subject to a 2-dimensional knapsack constraint. Thus, a $(1-e^{-1}-\varepsilon)$-approximate solution can be found efficiently [17].

Our approximation algorithm for SMKP is based on a generalization of the above. We refer to a set of bins of identical capacity as a *block*, and show how to reduce an SMKP instance into a submodular optimization problem with a d-dimensional knapsack constraint, in which d is twice the number of blocks plus a constant. While, generally, this problem cannot be solved for non-constant d, we use a refined analysis of known algorithms [17, 7] to show that the problem can be efficiently solved if the blocks admit a certain structure, to which we refer as *leveled*.

We utilize a grouping technique, inspired by the work of Fernandez de la Vega and Lueker [8], to convert a general SMKP instance to a leveled instance. We sort the bins in decreasing order by capacity and then partition them into levels, where level t, $t \geq 0$, has N^{2+t} bins, divided into N^2 consecutive blocks, each containing N^t bins. We decrease the capacity of each bin to the smallest capacity of a bin in the same block. While the decrease in capacity generates the leveled structure required for our algorithm to work, it only slightly decreases the optimal solution value. The main idea is that given an optimal solution, each block of decreased capacity can now be used to store the items assigned to the subsequent block on the same level. Also, the items assigned to N blocks from each level can be evicted, while only causing a reduction of $\frac{1}{N}$ to the profit (as only N of the N^2 blocks of the level are evicted). These evicted blocks are then used for the items assigned to the first block in the next level.

2 Preliminaries

Our analysis utilizes several basic properties of submodular functions. Given a monotone submodular function $f: 2^I \to \mathbb{R}_{\geq 0}$ and a set $S \subseteq I$, we define $f_S: 2^I \to \mathbb{R}_{\geq 0}$ by $f_S(A) = f(S \cup A) - f(S)$. It follows that f_S is a monotone, non-negative submodular function. The proof of the next claim is given in Appendix A.

▷ Claim 2. Let $f: 2^I \to \mathbb{R}_{\geq 0}$ be a non-negative, monotone and submodular function, and let $E \subseteq 2^I \times X$ for some set X (each element of E is a pair (S,h) with $S \subseteq I$ and $h \in X$). Then function $g: 2^E \to \mathbb{R}_{\geq 0}$ defined by $g(A) = f\left(\cup_{(S,h)\in A} S\right)$ is non-negative, monotone and submodular.

While Claim 2 is essential for our algorithm, it is important to emphasize it does not hold for non-monotone submodular functions.

Many modern submodular optimization algorithms rely on the submodular *Multilinear Extension* ([3, 17, 18, 23, 11, 2]). 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:

$$F(\bar{x}) = \sum_{S \subseteq I} f(S) \prod_{i \in S} \bar{x}_i \prod_{i \in I \setminus S} (1 - \bar{x}_i).$$

The multilinear extension can be interpreted as an expectation of a random variable. Given $\bar{x} \in [0,1]^I$ we say that a random set X is distributed according to $\bar{x}$, $X \sim \bar{x}$, if $\Pr(i \in X) = \bar{x}_i$ and the events $(i \in X)_{i \in I}$ are independent. It follows that $F(\bar{x}) = \mathbb{E}_{X \sim \bar{x}}[f(X)]$.

The continuous greedy of [3] can be used to find approximate solution for maximization problems of the form $\max F(\bar{x})$ s.t. $\bar{x} \in P$, where F is the multilinear extension of a monotone submodular function f, and P is a down-monotone polytope. The algorithm uses two oracles, one for f and another which given $\bar{\lambda} \in \mathbb{R}^I$ returns a vector $\bar{x} \in P$ such that $\bar{x} \cdot \bar{\lambda}$ is maximal. The algorithm returns $\bar{x} \in P$ such that $F(\bar{x}) \geq (1 - e^{-1}) \max_{\bar{y} \in P} F(\bar{y})$.

We use $\mathcal{I} = (I, w, B, W, f)$ to denote an SMKP instance consisting of a set of items I with weights w_i for $i \in I$, a set of bins B with capacities W_b for $b \in B$, and objective function f. Given a set $A \subseteq I$, let $w(A) = \sum_{i \in A} w_i$. We denote by $\mathrm{OPT}(\mathcal{I})$ the optimal solution value for the instance $\mathcal{I}$.

3 The Approximation Algorithm

In this section we present our approximation algorithm for SMKP. Given an instance $\mathcal{I}$ of the problem, let $A^* = \cup_{b \in B} A^*_b$ be an optimal solution of value $OPT(\mathcal{I})$. We first observe that there exists a constant size subset $A = \cup_{b \in B} A_b$, where $A_b \subseteq A^*_b$, satisfying the following property: the value gained from any item in $i \in A^* \setminus A$ is small relative to $OPT(\mathcal{I})$. Thus, our algorithm initially enumerates over all possible partial assignments of constant size. Each assignment is then extended to an approximate solution for $\mathcal{I}$. Among all possible partial assignments and the respective extensions the algorithm returns the best solution. Thus, from now on we restrict our attention to finding a solution for the *residual* problem, obtained by fixing the initial partial assignment.

Formally, given an SMKP instance, $\mathcal{I} = (I, w, B, W, f)$, a feasible partial solution $(A_b)_{b \in B}$ and $\xi \in \mathbb{N}$, we define the *residual* instance $\mathcal{I}' = (I', w, B, W', f')$ with respect to $(A_b)_{b \in B}$ and ξ as follows. Let $A = \cup_{b \in B} A_b$ and set $I' = \left\{ i \in I \setminus A \;\middle|\; f_A(\{i\}) \leq \frac{f(A)}{\xi} \right\}$. The weights of the items remain the same and so is the set of bins. For every $b \in B$ we set $W'_b = W_b - w(A_b)$. Finally, the objective function of the residual instance is $f' = f_A$.

▶ **Lemma 3.** *Let $\mathcal{I}$ be an SMKP instance, $\xi \in \mathbb{N}$, and $(A^*_b)_{b \in B}$ an optimal solution for $\mathcal{I}$ such that $A^*_{b_1} \cap A^*_{b_2} = \emptyset$ for any $b_1, b_2 \in B$, $b_1 \neq b_2$. If $\sum_{b \in B} |A^*_b| \geq \xi$ there is a feasible solution $(A_b)_{b \in B}$ for $\mathcal{I}$ such that $A_b \subseteq A^*_b$ for any $b \in B$, $\sum_{b \in B} |A_b| = \xi$, and $(A^*_b \setminus A_b)_{b \in B}$ is a feasible solution for the residual instance of $\mathcal{I}'$ w.r.t $(A_b)_{b \in B}$ and ξ.*

Proof. Let $(A^*_b)_{b \in B}$ be an optimal solution to the SMKP instance. Define $A^* = \cup_{b \in B} A^*_b$ and order the items of A^* by their *marginal values*. That is $A^* = \{a_1, \ldots, a_r\}$ where $f_{T_{\ell-1}}(\{a_\ell\}) = \max_{a \in A^* \setminus T_{\ell-1}} f_{T_{\ell-1}}(\{a\})$ with $T_\ell = \{a_1, \ldots, a_\ell\}$ for every $1 \leq \ell \leq r$ (also, $T_0 = \emptyset$). Define $(A_b)_{b \in B}$ by $A_b = A^*_b \cap \{a_1, \ldots, a_\xi\}$ for every $b \in B$ and $A = \cup_{b \in B} A_b$. We therefore have $A = \{a_1, \ldots, a_\xi\}$.

For any $b \in B$ it holds that $w(A_b) \leq w(A^*_b) \leq W_b$, and thus $(A_b)_{b \in B}$ is a feasible solution for $\mathcal{I}$. Furthermore, for any $b \in B$ it holds that $A_b \subseteq A^*_b$ by definition. As the sets $(A^*_b)_{b \in B}$ are disjoint it follows that $\sum_{b \in B} |A_b| = \xi$.

Let $\mathcal{I}' = (I', w, B, W', f')$ be the residual instance of $\mathcal{I}$ w.r.t $(A_b)_{b \in B}$ and ξ. We are left to show that $(A^*_b \setminus A_b)_{b \in B}$ is a feasible solution for $\mathcal{I}'$. For every $\xi < i \leq r$ and $1 \leq \ell \leq \xi$ it holds that $f_A(\{a_i\}) \leq f_{T_{\ell-1}}(\{a_i\}) \leq f_{T_{\ell-1}}(\{a_\ell\})$ where the first inequality follows from the submodularity of f and the second by the definition of a_ℓ. Combining the last inequality with $f' = f_A$ we obtain,

$$\xi \cdot f'(\{a_i\}) = \xi \cdot f_A(\{a_i\}) \leq \sum_{\ell=1}^{\xi} f_{T_{\ell-1}}(\{a_\ell\}) = f(A) - f(\emptyset) \leq f(A).$$

Thus, $a_i \in I'$, implying that $A^*_b \setminus A_b \subseteq I'$ for any $b \in B$. Furthermore, for any $b \in B$,

$$w(A^*_b \setminus A_b) = w(A^*_b) - w(A_b) \leq W_b - w(A_b) = W'_b.$$

It follows that $(A^*_b \setminus A_b)_{b \in B}$ is a solution to the residual instance. ◀

Next, we observe that instances of SMKP are easier to solve when the number of distinct bin capacities is small (e.g., uniform bin capacities), leading us to consider *bin blocks*:

▶ **Definition 4.** *For a given instance of SMKP we say that a subset of bins $\tilde{B} \subseteq B$ is a* block *if all the bins in $\tilde{B}$ have the same capacity, i.e., for bins b_1 and b_2 belonging to the same block it holds that $W_{b_1} = W_{b_2}$.*

Following an enumeration over partial assignments, our algorithm reduces the number of blocks by altering the bin capacities. To this end, we use a specific structure that we call *leveled*, defined as follows.

▶ **Definition 5.** *For any $N \in \mathbb{N}$, we say that a partition $(B_j)_{j=0}^k$ of a set B of bins with capacities $(W_b)_{b \in B}$ is N*-leveled *if B_j is a block, and $|B_j| = N^{\lfloor \frac{j}{N^2} \rfloor}$ for all $0 \leq j \leq k$.*

By the above definition, we can view each set of consecutive blocks of the same size as a *level*. For $0 \leq j \leq k$, block j belongs to level $\ell = \lfloor \frac{j}{N^2} \rfloor$. Thus, for $\ell \geq 0$, the number of bins in each block increases by factor of N when moving from level ℓ to level $\ell + 1$.

In Section 3.1 we give Algorithm 2 which generates an N-leveled partition of the bins, $\tilde{B} = \cup_{j=0}^k \tilde{B}_j$ with the capacities of the bins $(W_b)_{b \in B}$ modified to $(\tilde{W}_b)_{b \in \tilde{B}}$. We show that solving the problem with these new bin capacities may cause only a small harm to the optimal solution value. In particular, we prove (in Section 3.1) the following.

▶ **Lemma 6.** *Given a set of bins B, capacities $(W_b)_{b \in B}$ and a parameter N, Algorithm 2 returns in polynomial time a subset of bins $\tilde{B} \subseteq B$, capacities $(\tilde{W}_b)_{b \in \tilde{B}}$ and an N-leveled partition $(\tilde{B}_j)_{j=0}^k$ of $\tilde{B}$, such that*

1. *The bin capacities satisfy $\tilde{W}_b \leq W_b$, for all $b \in \tilde{B}$.*
2. $\mathrm{OPT}(\tilde{\mathcal{I}}) \geq \left(1 - \frac{1}{N}\right) \mathrm{OPT}(\mathcal{I})$*, for any SMKP instance $\mathcal{I} = (I, w, B, W, f)$ and $\tilde{\mathcal{I}} = (I, w, \tilde{B}, \tilde{W}, f)$.*

Once the instance is N-leveled, we proceed to solve the problem (fractionally) and apply randomized rounding to obtain an integral solution (see Section 3.2). Algorithm 4 utilizes efficiently the leveled structure of the instance. Instead of having a separate constraint for each bin in a block – to bound the total size of the items packed in this bin – we use only two constraints for each block. The first constraint is a knapsack constraint referring to the total capacity of a block, and the second constraint restricts the number of *configurations* assigned to the block.[4] Thus, the number of constraints significantly decreases if the blocks are large. Since leveled instances also have a constant number of blocks consisting of a *single* bin, those are handled separately via the notion of δ-restricted SMKP.

An input for δ-*restricted SMKP* includes the same parameters as an input for SMKP, and also a subset $B^r \subseteq B$ of *restricted* bins. A solution for δ-restricted SMKP is a feasible assignment $(A_b)_{b \in B}$ satisfying also the property that $\forall b \in B^r$ the items assigned to b are relatively small; namely, for any $i \in A_b$ $w_i \leq \delta W_b$.

[4] We defined a configuration in Section 1.2.

Given the N-leveled instance of our problem, we turn the blocks of a single bin (that is blocks $\tilde{B}_j$ such that $|\tilde{B}_j| = 1$) to be *restricted*. We note that while items of weight greater than δW_b may be assigned to these blocks in some optimal solution, the overall number of such items is bounded by a constant. Indeed, our initial enumeration guarantees that evicting these items from an optimal solution may cause only small harm to the optimal solution value, allowing us to consider the instance as δ-restricted.

In Section 3.2 we show the following bound on the performance guarantee of Algorithm 4, which uses randomized rounding. The algorithm is parameterized by $\mu \in (0, 0.1)$ (to be determined). Suppose we are given a δ-restricted SMKP instance $\mathcal{I}$, such that the unrestricted bins are partitioned into blocks, i.e., $B \setminus B^r = B_1 \cup \ldots \cup B_k$, and $v = \max_{i \in I} f(\{i\}) - f(\emptyset)$.

▶ **Lemma 7.** *For $\mu \in (0, 0.1)$, Algorithm 4 returns a feasible solution $(S_b)_{b \in B}$ such that $\mathbb{E}\left[f(\cup_{b \in B} S_b)\right] \geq (1 - e^{-1}) \frac{(1-\mu)^2}{1+\mu} (1 - \gamma) \mathrm{OPT}(\mathcal{I})$, where*

$$\gamma = \exp\left(-\frac{\mu^3}{16} \cdot \frac{\mathrm{OPT}(\mathcal{I})}{v}\right) + |B^r| \exp\left(-\frac{\mu^2}{12} \cdot \frac{1}{\delta}\right) + 2 \cdot \sum_{j=1}^{k} \exp\left(-\frac{\mu^2}{12} |B_j|\right).$$

Algorithm 1 gives the pseudocode of our approximation algorithm for general SMKP instances. The algorithm uses several configuration parameters that will be set in the proof of Lemma 8.

Algorithm 1 Algorithm for SMKP.

Input : An SMKP instance $\mathcal{I} = (I, w, B, W, f)$ and the parameters N, ξ, δ and μ.
forall *feasible assignments* $A = (A_b)_{b \in B}$ *such that* $\sum_{b \in B} |A_b| \leq \xi$ **do**
  Let $\mathcal{I}' = (I', w, B, W', f')$ be the residual instance of $\mathcal{I}$ w.r.t $(A_b)_{b \in B}$ and ξ.
  Run Algorithm 2 with the bins B and capacities $(W'_b)_{b \in B}$. Let $\tilde{B}$ and $(\tilde{W}_b)_{b \in \tilde{B}}$ be the output, and $\tilde{B} = \cup_{j=0}^{k} \tilde{B}_j$ the partition of $\tilde{B}$ to leveled blocks. Let $\tilde{\mathcal{I}} = (I', w, \tilde{B}, \tilde{W}, f')$ be the resulting instance.
  Let $\tilde{\mathcal{I}}_R$ be the δ-restricted SMKP instance of $\tilde{\mathcal{I}}$ with the restricted bins $\tilde{B}^r = \cup_{j=0}^{\min\{N^2-1, k\}} \tilde{B}_j$.
  Solve $\tilde{\mathcal{I}}_R$ using Algorithm 4 with parameter μ, and the partition $\tilde{B} \setminus \tilde{B}^r = \cup_{j=1}^{k} \tilde{B}_j$. Denote the returned assignment by $(\tilde{S}_b)_{b \in \tilde{B}}$, and let $S_b = \tilde{S}_b$ for $b \in \tilde{B}$ and $S_b = \emptyset$ for $b \in B \setminus \tilde{B}$.
  If $f(\cup_{b \in B}(A_b \cup S_b))$ is higher than the value of the current best solution, set $(A_b \cup S_b)_{b \in B}$ as the current best solution.
end
Return the best solution found.

▶ **Lemma 8.** *For any $\varepsilon > 0$, there are parameters N, ξ, δ, μ such that, for any SMKP instance $\mathcal{I}$, Algorithm 1 returns a solution of expected value at least $(1 - e^{-1} - \varepsilon)\mathrm{OPT}(\mathcal{I})$.*

Proof. We start by setting the parameter values. The reason for selecting these values will become clear later. Given a fixed $\varepsilon \in (0, 0.1)$, there is $\mu \in (0, 0.1)$ such that $\frac{(1-\mu)^2}{1+\mu} \geq (1 - \varepsilon^2)$. By the Monotone Convergence Theorem,

$$\lim_{N \to \infty} 2N^2 \cdot \sum_{t=1}^{\infty} \exp\left(-\frac{\mu^2 \cdot N^t}{12}\right) = \sum_{t=1}^{\infty} \lim_{N \to \infty} 2N^2 \exp\left(-\frac{\mu^2 \cdot N^t}{12}\right) = 0.$$

It follows that there are $N > \frac{1}{\varepsilon^2}$ and $\delta > 0$ such that

$$N^2 \exp\left(-\frac{\mu^2}{12}\cdot\frac{1}{\delta}\right) + 2N^2 \cdot \sum_{t=1}^{\infty} \exp\left(-\frac{\mu^2}{12}N^t\right) < \frac{\varepsilon^2}{2}. \tag{1}$$

Finally, we select ξ such that $\xi \geq \frac{N^2}{\varepsilon^2\delta}$ and $\exp\left(-\frac{\mu^3}{16}\frac{\xi}{5}\right) \leq \frac{\varepsilon^2}{2}$.

Let $\mathcal{I} = (I, w, B, W, f)$ be an SMKP instance, and let $(A^*_b)_{b\in B}$ be an optimal solution for $\mathcal{I}$. Assume w.l.o.g that $A^*_{b_1} \cap A^*_{b_2} = \emptyset$ for any $b_1, b_2 \in B$, $b_1 \neq b_2$. Define $A^* = \cup_{b\in B} A^*_b$. If $|A^*| \leq \xi$, there is an iteration of Line 1 in which $A^*_b = A_b$ for all $b \in B$. Therefore, in this iteration we have at Line 6 $f(\cup_{b\in B}(A_b \cup S_b)) \geq f(A^*)$, and the algorithm returns a solution of value at least $f(A^*)$. Otherwise, by Lemma 3, there is a feasible solution $(A_b)_{b\in B}$ such that $A_b \subseteq A^*_b$, $\sum_{b\in B} |A^*_b| = \xi$ and $(A^*_b \setminus A_b)_{b\in B}$ is a feasible solution to $\mathcal{I}'$, the residual instance of $\mathcal{I}$ w.r.t $(A_b)_{b\in B}$ and ξ. It follows that there is an iteration of Line 1 which considers this solution $(A_b)_{b\in B}$. We focus on this iteration for the rest of the analysis.

Let $A = \cup_{b\in B} A_b$. If $f(A) \geq (1 - e^{-1})f(A^*)$ then when the algorithm reaches Line 6 it holds that $f(\cup_{b\in B}(A_b \cup S_b)) \geq f(A) \geq (1 - e^{-1})f(A^*)$; therefore, the algorithm returns a $(1 - e^{-1})$-approximation in this case. Henceforth, we assume that $f(A) \leq (1 - e^{-1})f(A^*)$. Then,

$$f'(\cup_{b\in B}(A^*_b \setminus A_b)) = f'(A^* \setminus A) = f(A^*) - f(A) \geq \frac{f(A)}{1 - e^{-1}} - f(A) = \frac{1}{e-1} f(A). \tag{2}$$

As $(A^*_b \setminus A_b)_{b\in B}$ is a feasible solution for $\mathcal{I}'$, it holds that $\mathrm{OPT}(\mathcal{I}') \geq f'(A^* \setminus A)$. Therefore, by Lemma 6 and the choice of N, it holds that

$$\mathrm{OPT}(\tilde{\mathcal{I}}) \geq \left(1 - \frac{1}{N}\right) f'(A^* \setminus A) \geq (1 - \varepsilon^2) f'(A^* \setminus A), \tag{3}$$

where $\mathcal{I}'$ is the instance output by Algorithm 2. Let $(D_b)_{b\in\tilde{B}}$ by an optimal solution for $\tilde{\mathcal{I}}$. Consider $(D^r_b)_{b\in\tilde{B}}$ where $D^r_b = D_b \setminus \{i \in D_b | w_i > \delta \cdot \tilde{W}_b\}$ for $b \in \tilde{B}^r$ (the set $\tilde{B}^r$ is defined in Line 4) and $D^r_b = D_b$ for $b \in \tilde{B} \setminus \tilde{B}^r$. It follows that D^r_b is a solution for the δ-restricted SMKP instance $\tilde{\mathcal{I}}_R$. As for any $b \in \tilde{B}^r$, $|\{i \in D_b | w_i > \delta \cdot \tilde{W}_b\}| \leq \frac{1}{\delta}$, we have that

$$\mathrm{OPT}\left(\tilde{\mathcal{I}}_R\right) \geq f'\left(\cup_{b\in\tilde{B}} D^r_b\right) \geq \mathrm{OPT}\left(\tilde{\mathcal{I}}\right) - \frac{N^2}{\delta\cdot\xi} f(A) \geq (1 - \varepsilon^2) f'\left(A^* \setminus A\right) - \varepsilon^2 \cdot f(A). \tag{4}$$

The second inequality follows from the definition of residual instance, and the third inequality from (3) and the choice of ξ. Since $f'(A^* \setminus A) \geq \frac{1}{e-1} f(A)$ and $\varepsilon \in (0, 0.1)$, it follows that $\mathrm{OPT}(\tilde{\mathcal{I}}_R) \geq \frac{f(A)}{5}$.

By Lemma 7, we have that

$$\mathbb{E}\left[f'(\cup_{b\in\tilde{B}}\tilde{S}_b)\right] \geq (1-e^{-1})\frac{(1-\mu)^2}{1+\mu}(1-\gamma)\mathrm{OPT}(\tilde{\mathcal{I}}_R) \geq (1-e^{-1})(1-\varepsilon^2)(1-\gamma)\mathrm{OPT}(\tilde{\mathcal{I}}_R), \tag{5}$$

where

$$\begin{aligned} \gamma &= \exp\left(-\tfrac{\mu^3}{16}\cdot\tfrac{\mathrm{OPT}(\tilde{\mathcal{I}}_R)}{\xi^{-1}f(A)}\right) + |\tilde{B}^r| \exp\left(-\tfrac{\mu^2}{12}\cdot\tfrac{1}{\delta}\right) + 2\cdot\textstyle\sum_{j=N^2}^{k} \exp\left(-\tfrac{\mu^2}{12}|\tilde{B}_j|\right) \\ &\leq \exp\left(-\tfrac{\mu^3}{16}\cdot\tfrac{\xi}{5}\right) + N^2 \exp\left(-\tfrac{\mu^2}{12}\cdot\tfrac{1}{\delta}\right) + 2\cdot N^2 \cdot \textstyle\sum_{t=1}^{\infty} \exp\left(-\tfrac{\mu^2}{12}N^t\right) \leq \varepsilon^2. \end{aligned} \tag{6}$$

The first equality uses $f'(\{i\}) \leq \xi^{-1} f(A)$ (by the definition of $\mathcal{I}'$). The first inequality holds since $\mathrm{OPT}(\tilde{\mathcal{I}}_R) \geq \frac{f(A)}{5}$, $|\tilde{B}^r| \leq N^2$ and there are at most N^2 blocks $\tilde{B}_j$ of size N^t. The second inequality uses (1) and the choice of ξ. Combining (6) with (5) and (4), we obtain

$$\begin{aligned}\mathbb{E}\left[f(\cup_{b\in B}(A_b\cup S_b))\right] &\geq f(A)+\mathbb{E}\left[f'(\cup_{b\in\tilde{B})}\tilde{S}_b)\right]\geq f(A)+(1-e^{-1})(1-\varepsilon^2)^2\mathrm{OPT}(\tilde{\mathcal{I}}_R)\\ &\geq f(A)+(1-e^{-1})(1-\varepsilon^2)^3 f'(A^*\setminus A)-\varepsilon^2 f(A)\geq(1-e^{-1}-\varepsilon)f(A^*).\end{aligned}$$

Hence, in this iteration the solution considered in Line 6 has expected value at least $(1-e^{-1}-\varepsilon)f(A^*)$. This completes the proof of the lemma. ◀

▶ **Lemma 9.** *For any constant parameters N, ξ, δ and μ, Algorithm 1 returns a feasible solution for the input instance in polynomial time.*

Proof. We first note that for any fixed parameter values the algorithm has a polynomial running time. The number of assignments considered in Line 1 can be trivially bounded by $(n\cdot m)^{\xi}$. As Algorithms 2 and 4 are polynomial in their input size, the operations in each iteration are also done in polynomial time.

For each iteration of Line 1, by Lemma 7, $(\tilde{S}_b)_{b\in\tilde{B}}$ is a feasible solution to $\tilde{\mathcal{I}}_R$. Therefore, for any $b\in B$ either $w(S_b)=w(\emptyset)\leq W'_b$ or $w(S_b)=w(\tilde{S}_b)\leq\tilde{W}_b\leq W'_b$, where the last equality follows from Lemma 6. Therefore, $w(A_b\cup S_b)\leq w(A_b)+W'_b\leq W_b$. Hence, the solution considered in each iteration is feasible for the input instance. ◀

Theorem 1 follows from Lemmas 8 and 9.

3.1 Structuring the Instance

In this section we present Algorithm 2 and prove Lemma 6. Our technique for generating an N-leveled partition can be viewed as a variant of the linear grouping technique of [8] which requires the use of *non-uniform* group sizes (each group of bins then becomes a *block*). Given a set of bins B with capacities $(W_b)_{b\in B}$, we sort the bins in non-increasing order by capacities. We now use the numbering $B=\{1,\ldots,m\}$, where $W_1\geq W_2\geq\ldots\geq W_m$. To generate an N-leveled partition of the bins and the modified capacities, we define groups (or blocks) of bins, where each group j consists of $N^{\lfloor\frac{j}{N^2}\rfloor}$ consecutive bins, for $j\geq 0$. Starting from the first bin, we keep generating such groups as long as there are enough bins to form a group of the desired size. We omit the remaining bins and decrease the capacity of each bin to the minimal capacity of a bin in its group. We formalize this procedure in Algorithm 2.

Algorithm 2 Structure in Blocks.

Input : A set of bins B, capacities $(W_b)_{b\in B}$ and N.
Let $B=\{1,\ldots,m\}$ where $W_1\geq W_2\geq\ldots\geq W_m$.
Let $k=\max\left\{\ell\in\mathbb{N}\;\middle|\;\sum_{r=0}^{\ell}N^{\lfloor\frac{r}{N^2}\rfloor}\leq m\right\}$.
Define $\tilde{B}_j=\left\{b\;\middle|\;\sum_{r=0}^{j-1}N^{\lfloor\frac{r}{N^2}\rfloor}<b\leq\sum_{r=0}^{j}N^{\lfloor\frac{r}{N^2}\rfloor}\right\}$ for $0\leq j\leq k$.
Let $\tilde{B}=\cup_{j=0}^{k}\tilde{B}_j$, and $\tilde{W}_b=\min_{b'\in\tilde{B}_j}W_{b'}$ for all $0\leq j\leq k$ and $b\in\tilde{B}_j$.
Return $\tilde{B}$, $(\tilde{W}_b)_{b\in\tilde{B}}$ and the partition $(\tilde{B}_j)_{j=0}^{k}$.

The following standard result for submodular functions is used in the proof of Lemma 6.

▶ **Lemma 10.** *Let $h:2^{\Omega}\to\mathbb{R}_{\geq 0}$ be a monotone submodular function, and let $S_{i,1},\ldots,S_{i,N}\subseteq\Omega$ for $1\leq i\leq M$. Then for every $1\leq i\leq M$ there is $1\leq j_i^*\leq N$ such that*

$$h\left(\bigcup_{i=1}^{M}\bigcup_{1\leq j\leq N,\;j\neq j_i^*}S_{i,j}\right)\geq\left(1-\frac{1}{N}\right)h\left(\bigcup_{i=1}^{M}\bigcup_{j=1}^{N}S_{i,j}\right).$$

The proof for the Lemma is given in Appendix A.

Proof of Lemma 6. By construction, we have that $(\tilde{B}_j)_{j=0}^k$ is an N-leveled partition of $(\tilde{W}_b)_{b\in\tilde{B}}$ and $\tilde{W}_b \leq W_b$ for any $b \in \tilde{B}$. Also, Algorithm 2 has a polynomial running time.

To complete the proof we need to show Property 2 in the lemma. Let $\mathcal{I} = (I, w, B, W, f)$ be an SMKP instance, and let $\tilde{\mathcal{I}} = (I, w, \tilde{B}, \tilde{W}, f)$ be the instance with bins and capacities generated (as output) by Algorithm 2. We need to show that $\mathrm{OPT}(\tilde{\mathcal{I}}) \geq \left(1 - \frac{1}{N}\right)\mathrm{OPT}(\mathcal{I})$.

Let $A_1^*, \ldots, A_m^*$ be an optimal solution for $\mathcal{I}$, and $A^* = \cup_{b\in B} A_b^*$. We modify this solution using a sequence of steps, eventually obtaining a feasible solution for $\tilde{\mathcal{I}}$. The latter is used to lower bound $\mathrm{OPT}(\tilde{\mathcal{I}})$. Define $\tilde{B}_{k+1} = B \setminus \tilde{B}$. We note that $\tilde{B}_{k+1}$ may be empty. We partition $\{\tilde{B}_j | \; 0 \leq j \leq k+1\}$ into *levels* and *super-blocks*. We consider each N^2 consecutive blocks to be a *level*, and each N consecutive blocks within a level to be a *super-block*. Formally, level t is

$$\mathcal{L}_t = \left\{ j \mid t \cdot N^2 \leq j < \min\{(t+1)N^2, \; k+2\} \right\}$$

for $0 \leq t \leq \ell$ with $\ell = \left\lfloor \frac{k+1}{N^2} \right\rfloor$. Also, super-block r of level t is

$$\mathcal{S}_{t,r} = \left\{ j \mid t \cdot N^2 + r \cdot N \leq j < t \cdot N^2 + (r+1) \cdot N \right\}$$

for $0 \leq r < N$ and level $0 \leq t < \ell$ (we do not partition the last level to super-blocks). It follows that $B = \cup_{t=0}^{\ell} \cup_{j\in\mathcal{L}_t} \tilde{B}_j$ and $\mathcal{L}_t = \cup_{r=0}^{N-1} \mathcal{S}_{t,r}$ for $0 \leq t < \ell$. Furthermore, for any $j \in \mathcal{L}_t$, $j \neq k+1$ it holds that $|\tilde{B}_j| = N^t$ and $|\tilde{B}_{k+1}| < N^\ell$. Essentially, all the blocks of level t are of the same size.

We modify $A_1^*, \ldots, A_m^*$ using the following steps. First, in each level (except the last one) we evict all the bins from a singe super-block. Since there are N super-blocks in each of these levels, this may decrease the value of the assignment at most by factor $\frac{1}{N}$. Then, we slightly shuffle the content of the bins in all levels (except the last one). In each level, we place the content of the bins of the last super-block in bins of the evicted super-block in the same level. As the bins are ordered by capacity, this will keep the assignment feasible with respect to the original capacities. In the last step, for each level (except level 0) we move the content of the bins from the first block to the bins of the last super-block in the previous level (the two sets of bins have the same cardinality), and content of bins from other blocks (except level 0) to the previous block from the same level. This yields a feasible assignment for the leveled instance. We formally describe these steps in the following.

Eviction: We first evict a super-block of bins from each level (except the last one). Let $R = \cup_{b\in\mathcal{L}_\ell} A_b^*$ be the subset of items assigned to the last level. By Lemma 10, for any $0 \leq t < \ell$ there is r_t^* such that

$$\begin{aligned} f_R\left(\bigcup_{t=0}^{\ell-1} \bigcup_{0\leq r<N, \; r\neq r_t^*} \bigcup_{j\in\mathcal{S}_{t,r}} \bigcup_{b\in\tilde{B}_j} A_b^* \right) &\geq \left(1 - \frac{1}{N}\right) f_R\left(\bigcup_{t=0}^{\ell-1} \bigcup_{r=0}^{N-1} \bigcup_{j\in\mathcal{S}_{t,r}} \bigcup_{b\in\tilde{B}_j} A_b^* \right) \\ &= \left(1 - \frac{1}{N}\right) f_R(A^*). \end{aligned}$$

Define $T_1, \ldots, T_m$ by $T_b = \emptyset$ for any $b \in \bigcup_{t=0}^{\ell-1} \bigcup_{j\in\mathcal{S}_{t,r_t^*}} \tilde{B}_j$, and $T_b = A_b^*$ for any $b \in B \setminus \left(\bigcup_{t=0}^{\ell-1} \bigcup_{j\in\mathcal{S}_{t,r_t^*}} \tilde{B}_j \right)$. Then,

$$f\left(\bigcup_{b\in B} T_b\right) = f(R) + f_R\left(\bigcup_{t=0}^{\ell-1} \bigcup_{0\leq r<N-1, \; r\neq r_t^*} \bigcup_{j\in\mathcal{S}_{t,r}} \bigcup_{b\in\tilde{B}_j} A_b^* \right) \geq \left(1 - \frac{1}{N}\right) f(A^*).$$

It also holds that $T_1, \ldots, T_m$ is a feasible solution for $\mathcal{I}$.

Shuffling: We generate a new assignment $\tilde{T}_1, \ldots, \tilde{T}_m$ such that $\cup_{b\in B}\tilde{T}_b = \cup_{b\in B} T_b$ and the last super-blocks in each level (except the last one) is empty. This property is obtained by moving the content of the bins in super-block $N-1$ to the bins of super-block r_t^* for every $0 \le t < \ell$.

We define $(\tilde{T}_b)_{b\in B}$ as follows. For any $0 \le t < \ell$, $j \in \mathcal{S}_{t,r_t^*}$ let $\varphi_t : \bigcup_{j\in\mathcal{S}_{t,r_t^*}} \tilde{B}_j \to \bigcup_{j\in\mathcal{S}_{t,N-1}} \tilde{B}_j$ be a bijection (since $\left|\bigcup_{j\in\mathcal{S}_{t,N-1}} \tilde{B}_j\right| = \left|\bigcup_{j\in\mathcal{S}_{t,r_t^*}} \tilde{B}_j\right| = N\cdot N^t$, such a function exists). For any $b \in \bigcup_{j\in\mathcal{S}_{t,r_t^*}} \tilde{B}_j$ set $\tilde{T}_b = T_{\varphi_t(b)}$. By definition we have $\varphi_t(b) \ge b$; therefore, in this case

$$w(\tilde{T}_b) = w(T_{\varphi_t(b)}) \le W_{\varphi_t(b)} \le W_b.$$

For any $0 \le t < \ell$, $j \in \mathcal{S}_{t,N-1}$ and $b \in \tilde{B}_j$ set $\tilde{T}_b = \emptyset$. For any other bin $b \in B$ set $\tilde{T}_b = T_b$.

It follows that $\cup_{b\in B}\tilde{T}_b = \cup_{b\in B} T_b$, since $T_b = \emptyset$ for every $0 \le t < \ell$, $j \in \mathcal{S}_{t,r_t^*}$ and $b \in B_j$. Also, $(\tilde{T}_b)_{b\in B}$ is a feasible solution for $\mathcal{I}$.

Shifting: In this step we generate the assignment $(A_b)_{b\in\tilde{B}}$ which satisfies the properties in the lemma. As the bins of the last super-block in each level (except the last one) are vacant in $\tilde{T}_1, \ldots, \tilde{T}_m$, we use them for the content assigned to the first block of the next level. This can be done since N blocks of level t contain the same number of bins as a single block of level $t+1$. We also use blocks in levels greater than 0 which are not in the last super-block to store the content of the next block in the same level.

For any $0 < t \le \ell$ and $j \in \mathcal{L}_t$, consider a block $\tilde{B}_j$. Suppose that $j \ne t \cdot N^2$ and $j \notin \mathcal{S}_{t,N-1}$ where $t \ne \ell$; that is, $\tilde{B}_j$ is not the first block in the level, and is not in the last super-block of a level other than the last one. Then, let $\psi_j : \tilde{B}_j \to \tilde{B}_{j-1}$ be a bijection, and define $A_{\psi_t(b)} = \tilde{T}_b$ for any $b \in \tilde{B}_j$. If $j = t \cdot N^2$ (that is, $\tilde{B}_j$ is the first block in a level), let $\psi_j : \tilde{B}_j \to \bigcup_{j'\in\mathcal{S}_{t-1,N-1}} \tilde{B}_{j'}$ be a bijection, and define $A_{\psi_t(b)} = \tilde{T}_b$ for any $b \in \tilde{B}_j$. Finally, for any $0 \le j < N^2 - N$, let $\psi_j : \tilde{B}_j \to \tilde{B}_j$ be the identity function and define $A_{\psi_t(b)} = A_b = \tilde{T}_b$ for the single bin $b \in \tilde{B}_j$. For any bin $b \in \tilde{B}$ not handled in this process, set $A_b = \emptyset$.

We note that the definition is sound as the ranges of the functions ψ_j above do not intersect. Let $b \in \tilde{B}$. If $A_b = \emptyset$ then $w(A_b) \le \tilde{W}_b$. Otherwise, if $b \in \tilde{B}_j$ for $0 \le j < N^2 - N$ then $w(A_b) = w(\tilde{T}_b) < W_b \le \tilde{W}_b$. Finally, the only option left is that $b = \psi_{j'}(b')$ for some $0 < t \le \ell$, $j' \in \mathcal{L}_t$ and $b' \in \tilde{B}_{j'}$, such that $j' \ne t\cdot N^2$ and $j' \notin S_{t,N-1}$ if $t \ne \ell$. By definition, it holds that $b' \in \tilde{B}_j$ for some $j < j'$. Since the bins were ordered by capacity, we have

$$w(A_b) = w(A_{\phi_{j'}(b')}) = w(\tilde{T}_{b'}) \le W_{b'} \le \min_{b''\in\tilde{B}_j} W_{b''} = \tilde{W}_b.$$

Thus, the assignment is feasible with respect to the bins $\tilde{B}$ with capacities $\left(\tilde{W}_b\right)_{b\in\tilde{B}}$.

It also holds that for any $b \in B$ such that $\tilde{T}_b \ne 0$ there is $b' \in \tilde{B}$ such that $A_{b'} = \tilde{T}_b$. Therefore, $\bigcup_{b\in\tilde{B}} A_b = \bigcup_{b\in B} \tilde{T}_b = \bigcup_{b\in B} T_b$. Hence, $f(\cup_{b\in\tilde{B}} A_b) = f(\cup_{b\in B} T_b) \ge \left(1 - \frac{1}{N}\right) f(A^*)$. We conclude that $\mathrm{OPT}(\tilde{\mathcal{I}}) \ge \left(1 - \frac{1}{N}\right) \mathrm{OPT}(\mathcal{I})$. ◀

3.2 Solving a Continuous Relaxation and Rounding

In this section we give Algorithm 4 which outputs a solution satisfying Lemma 7. The input for the algorithm is a δ-restricted SMKP instance along with a partition $B \setminus B^r = B_1 \cup \ldots \cup B_k$ of the bins, where B_j is a block for all $1 \le j \le k$. Algorithm 4 uses Algorithm 3 as a subroutine which converts a solution for an auxiliary *block-constraint* problem into a solution for δ-restricted SMKP.

3.2.1 The Block-Constraint Problem

We now define the *block-constraint problem*, to be solved for a given instance $\mathcal{I}$ of δ-restricted SMKP, using the partition $B \setminus B^r = B_1 \cup \ldots \cup B_k$ and the parameter μ. The input for the block-constraint problem is a universe of elements E, a monotone submodular function $g : 2^E \to \mathbb{R}_{\geq 0}$ and a polytope $P \subseteq [0,1]^E$ (see below).

For simplicity, let $\{B_{k+1}, \ldots, B_\ell\} = \{\{b\} | \ b \in B^r\}$ be the set of blocks, each consisting of a single bin. Thus, $B = \cup_{j=1}^{\ell} B_j$. Denote the (uniform) capacity of the bins in block B_j by W_j^*, for $1 \leq j \leq \ell$. That is, for any $b \in B_j$ it holds that $W_j^* = W_b$. For $1 \leq j \leq k$, we say that an item $i \in I$ is *j-small* if $w_i \leq \mu \cdot W_j^*$, otherwise i is *j-large*. Let $I_j = \{\{i\} | \ i \text{ is } j\text{-small}\}$ for $1 \leq j \leq k$. For $k < j \leq \ell$ define $I_j = \{\{i\} \mid w_i \leq \delta W_j^*\}$.

A *j-configuration* is a subset of j-large items which can be packed into a single bin in B_j. That is, $C \subseteq I$ is a j-configuration if every item $i \in C$ is j-large and $w(C) \leq W_j^*$. Let $\mathcal{C}_j$ be the set of all j-configurations for $1 \leq j \leq k$ and $\mathcal{C}_j = \emptyset$ for $k < j \leq \ell$. As any j-configuration has at most μ^{-1} items, it follows that $|\mathcal{C}_j| \leq |I|^{\mu^{-1}}$, i.e., the number of configurations is polynomial in the size of $\mathcal{I}$. Furthermore, for $A \subseteq I$ such that $w(A) \leq W_j^*$, $1 \leq j \leq k$, there are $C \in \mathcal{C}_j$ and $S \subseteq I$ such that all the items in S are j-small and $A = C \cup S$. Our algorithm exploits this property.

Towards solving the block-constraint problem we define a submodular function g over a new universe of elements. Let $E = \{(S,j) | \ S \in \mathcal{C}_j \cup I_j, \ 1 \leq j \leq \ell\}$. Informally, the element $(S,j) \in E$ represents an assignment of all the items in S to a single bin $b \in B_j$. We now define $g : 2^E \to \mathbb{R}_{\geq 0}$ by $g(T) = f\left(\bigcup_{(S,j) \in T} S\right)$. By Claim 2, g is a submodular, monotone and non-negative function‘.

We define a polytope P for the instance $\mathcal{I}$ as follows.

$$P = \left\{ \bar{x} \in [0,1]^E \;\middle|\; \begin{array}{lr} \displaystyle\sum_{C \in \mathcal{C}_j} \bar{x}_{(C,j)} \leq |B_j| & \forall 1 \leq j \leq k \\ \displaystyle\sum_{S \in \mathcal{C}_j \cup I_j} w(S) \cdot \bar{x}_{(S,j)} \leq |B_j| \cdot W_j^* & \forall 1 \leq j \leq \ell \end{array} \right\} \tag{7}$$

The polytope represents a relaxed version of the capacity constraints over the bins. For each block B_j, $1 \leq j \leq k$, we only require that the *total weight* of items assigned to bins in B_j does not exceed the total capacity of the bins in this block. We also require that the number of j-configurations selected for B_j is no greater than the number of bins in this block.

Given an instance $\mathcal{I}$ of δ-restricted SMKP, along with the partition $B \setminus B^r = B_1 \cup \ldots \cup B_k$ and the parameter μ, we use for the *block-constraint problem* the universe E, the function g and the polytope $P \subseteq [0,1]^E$ as defined above.

We start by establishing a connection between the solution for the block-constraint problem for the given instance $\mathcal{I}$ and the optimal solution for δ-restricted SMKP on this instance. For a set $T \subseteq E$, we use $\bar{x}^T$ to denote the vector $\bar{x}^T \in \{0,1\}^E$ defined by $\bar{x}^T_e = 1$ for $e \in T$, and $\bar{x}^T_e = 0$ for $e \in E \setminus T$. Also, given a polytope Q and $\eta \geq 0$ we use the notation $\eta \cdot Q = \{\eta \bar{x} \mid \bar{x} \in Q\}$.

Our algorithm for solving δ-restricted SMKP on $\mathcal{I}$ solves first the block-constraint problem on this instance and then transforms the solution into a feasible solution for δ-restricted SMKP. We give the pseudocode for the transformation in Algorithm 3.

Algorithm 3 Employ a Block-Constraint Solution for SMKP.

Input : A δ-restricted SMKP instance $\mathcal{I} = (I, w, B, W, f)$, the partition of bins to block $\cup_{j=1}^{\ell} B_j$ and $T \subseteq E$.

1 Set $A_b = \emptyset$ for every $b \in B$.
2 Sort the elements (S, j) in T in decreasing order by the $w(S)$ values.
3 **for** *each* $(S, j) \in T$ *in the sorted order* **do**
4 | Set $A_b \leftarrow A_b \cup S$ where $b = \arg\min_{b \in B_j} w(A_b)$.
5 **end**
6 Return $(A_b)_{b \in B}$.

▶ **Lemma 11.** *Given an instance $\mathcal{I}$ of δ-restricted SMKP, consider the universe E and the polytope P as defined above. Then the following hold:*

1. *There is $T \subseteq E$, $\bar{x}^T \in P$ such that $g(T) \geq \mathrm{OPT}(\mathcal{I})$, where $\mathrm{OPT}(\mathcal{I})$ is the optimal solution value for δ-restricted SMKP on $\mathcal{I}$.*
2. *Given $T \subseteq E$ such that $\bar{x}^T \in (1-\mu) \cdot P$, Algorithm 3 returns in polynomial time a feasible solution $(A_b)_{b \in B}$ for δ-restricted SMKP instance $\mathcal{I}$ satisfying $f(\cup_{b \in B} A_b) = g(T)$.*

Proof. We start by proving part 1. Let $(A_b^*)_{b \in B}$ be an optimal solution for the δ-restricted SMKP instance, and let L_j be the set of all j-large items for $1 \leq j \leq k$ and $L_j = \emptyset$ for $k < j \leq \ell$. Define

$$T = \left(\bigcup_{j=1}^{k} \{(A_b^* \cap L_j, j) \mid b \in B_j\} \right) \cup \left(\bigcup_{j=1}^{\ell} \bigcup_{b \in B_j} \{(\{i\}, j) \mid i \in A_b^* \setminus L_j\} \right).$$

It can be easily shown that $g(T) = f(\cup_{b \in B} A_b^*)$. Furthermore, as $(A_b^*)_{b \in B}$ is a feasible solution, it holds that $\bar{x}^T \in P$.

We now prove part 2. Let $(A_b)_{b \in B}$ be the output of Algorithm 3 for the given input. We first note that $\cup_{b \in B} A_b = \cup_{(S,j) \in T} S$, and thus $g(T) = f(\cup_{b \in B} A_b)$.

For any $b \in B^r$, there is $k < j \leq \ell$ such that $B_j = \{b\}$. Therefore $A_b = \{i | (\{i\}, j) \in T\}$, and since $\bar{x}^T \in (1-\mu)P$ it follows that $w(A_b) \leq W_j^* = W_b$.

Let $1 \leq j \leq k$ and $b \in B_j$. Assume by negation that $w(A_b) > W_b = W_j^*$. Let $(S, j) \in T$ be the last element in T such that $S \neq \emptyset$ and S was added to A_b in Line 4. We conclude that $w(A_b \setminus S) > 0$, as otherwise $w(A_b) = w(S) \leq W_b$, by the definition of E. Therefore there are at least $|B_j|$ elements $(S', j) \in T$ such that $w(S') \geq w(S)$ (else, on the iteration of (S, j) there must be $b \in B_j$ with $A_b = \emptyset$). If $S \in \mathcal{C}_j$ then $w(S) > \mu \cdot W_j^*$ and thus

$$|\{S' \neq \emptyset | \ (S', j) \in T, \ S' \in \mathcal{C}_j\}| \ \geq \ |\{S' | \ (S', j) \in T, \ w(S') \geq w(S)\}| \ > \ |B_j|,$$

contradicting $\bar{x}^T \in (1-\mu)P$.

Therefore $S \notin \mathcal{C}_j$, and we can conclude that $S = \{i\}$ with $w_i \leq \mu \cdot W_j^*$. Thus, $w(A_b \setminus S) > (1-\mu) \cdot W_j^*$. Here, S has been allocated to A_b (which is itself a set of minimum weight). Then, for any $b' \in B_j$, we have $w(A_{b'}) \geq w(A_b) > (1-\mu) \cdot W_j^*$. Thus,

$$\sum_{(S',j) \in T} w(S') \geq \sum_{b' \in B_j} w(A_{b'}) > |B_j|(1-\mu) \cdot W_j^*,$$

contradicting $\bar{x}^T \in (1-\mu)P$. We conclude that $w(A_b) \leq W_b$.

Also, by definition, we have that for any $b \in B^r$ and $i \in A_b$ it holds that $w_i \leq \delta W_b$. Hence, $(A_b)_{b \in B}$ is a solution to the restricted SMKP instance. ◀

3.2.2 An Algorithm for δ-restricted SMKP

We are now ready to present our algorithm for δ-restricted SMKP. We note that in Line 3 of Algorithm 4 we use sampling by a solution vector $\bar{x}^*$, as defined in Section 2.

Algorithm 4 Solve and Round.

Input: A δ-restricted SMKP instance $\mathcal{I}$, a partition to blocks $B \setminus B^r = \cup_{j=k}^{j} B_j$, and a parameter $\mu > 0$.

1 Define E, g and P for the block-constraint problem on $\mathcal{I}$ and $\cup_{j=1}^{k} B_j$.
2 Let $G : [0,1]^E \to \mathbb{R}_{\geq 0}$ be the multilinear extension of g. Find a solution $\bar{x}^*$ for $\max_{\bar{x} \in \frac{1-\mu}{1+\mu} P} G(\bar{x})$ using the continuous greedy of [4].
3 Sample a set $T \sim \bar{x}^*$.
4 **if** $T \in (1-\mu)P$ **then**
5   Use Algorithm 3 to convert T into a solution $(A_b)_{b \in B}$ for δ-restricted SMKP on $\mathcal{I}$ and return $(A_b)_{b \in B}$.
6 **else**
7   Return $(A_b)_{b \in B}$ with $A_b = \emptyset$ for every $b \in B$.
8 **end**

For the analysis, consider first the running time. We note that, for any $\bar{\lambda} \in \mathbb{R}^E$, a vector $\bar{x} \in \frac{1-\mu}{1+\mu} P$ which maximizes $\bar{x} \cdot \bar{\lambda}$ can be found in polynomial time. Therefore, the continuous greedy in Line 2 runs in polynomial time. Thus, Algorithm 4 has a polynomial running time.

It remains to show that the algorithm returns a solution of expected value as stated in Lemma 7. The approach we use to prove the statement of the lemma is similar to the approach taken in [6]. In fact, it is possible to prove a variant of this claim using an approach of [17]. While eliminating the dependency on υ, this will result in a more involved proof.

Proof of Lemma 7. For any $e \in E$ define X_e to be a random variable such that $X_e = 1$ if $e \in T$ and $X_e = 0$ otherwise. It follows that $(X_e)_{e \in E}$ are independent Bernoulli random variables, $\mathbb{E}[X_e] = \bar{x}^*_e$ and $T = \{e \in E | X_e = 1\}$.

We first consider blocks $k < j \leq \ell$. Let $k < j \leq \ell$ and $B_j = \{b\}$. Since $\bar{x}^* \in \frac{1-\mu}{1+\mu} P$, it follows that $\mathbb{E}\left[\sum_{(S,j) \in E} w(S) \cdot X_{(S,j)}\right] \leq \frac{1-\mu}{1+\mu} \cdot W_b$. Also, $w_{(S,j)} \leq \delta \cdot W_b$ for every $(S,j) \in E$. Using Chernoff's bound (Theorem 3.1 in [13], see also Lemma 15), we have

$$\Pr\left(\sum_{(S,j) \in T} w(S) > (1-\mu) W_b\right) \leq \exp\left(-\frac{\mu^2}{3} \cdot \frac{1-\mu}{1+\mu} \cdot \frac{1}{\delta}\right) \leq \exp\left(-\frac{\mu^2}{12} \cdot \frac{1}{\delta}\right), \tag{8}$$

with the last inequality following from $\mu \in (0, 0.1)$.

Now, let $1 \leq j \leq k$. For every $(S,j) \in E$ it holds that $w(S) \leq W^*_j$. Also, since $\bar{x}^* \in \frac{1-\mu}{1+\mu} P$, $\mathbb{E}\left[\sum_{(S,j) \in E} w(S) \cdot X_{(S,j)}\right] \leq \frac{1-\mu}{1+\mu} \cdot |B_j| W^*_j$, and $\mathbb{E}\left[\sum_{(S,j) \in E:\ S \in \mathcal{C}_j} 1 \cdot X_{(S,j)}\right] \leq \frac{1-\mu}{1+\mu} \cdot |B_j|$. Therefore, by Chernoff's bound (Theorem 3.1 in [13] and Lemma 15), we have

$$\Pr\left(\sum_{(S,j) \in T} w(S) > (1-\mu) |B_j| W^*_j\right) \leq \exp\left(-\frac{\mu^2}{3} \cdot \frac{1-\mu}{1+\mu} \cdot |B_j|\right) \leq \exp\left(-\frac{\mu^2}{12} \cdot |B_j|\right) \tag{9}$$

$$\Pr\left(\sum_{(S,j) \in T:\ S \in \mathcal{C}_j} 1 > (1-\mu) |B_j|\right) \leq \exp\left(-\frac{\mu^2}{3} \cdot \frac{1-\mu}{1+\mu} \cdot |B_j|\right) \leq \exp\left(-\frac{\mu^2}{12} \cdot |B_j|\right). \tag{10}$$

By Lemma 11, $\max_{\bar{z}\in P} G(\bar{z}) \geq \text{OPT}(\mathcal{I})$. Since the second derivatives of G are non-positive (see [4]) it follows that $\max_{\bar{z}\in \frac{1-\mu}{1+\mu}P} G(\bar{z}) \geq \frac{1-\mu}{1+\mu}\text{OPT}(\mathcal{I})$. As the continuous greedy of [4] yields a $(1-e^{-1})$-approximation for the problem of maximizing the multilinear extension subject to a polytope constraint, it follows that

$$G(\bar{x}^*) \geq (1-e^{-1})\frac{1-\mu}{1+\mu}\text{OPT}(\mathcal{I}). \tag{11}$$

For any $(S,j)\in E$ we have $|S|\leq \mu^{-1}$, and from the submodularity of f, $g(\{(S,j)\}) - g(\emptyset) \leq \mu^{-1}\upsilon$ (recall that υ is defined in Lemma 7). Therefore, by the concentration bound of [6] (see Lemma 16), we have

$$\begin{aligned} &\Pr\left(g(T) \leq (1-e^{-1})\frac{(1-\mu)^2}{1+\mu}\text{OPT}(\mathcal{I})\right) \leq \Pr\left(g(\{e\in E | X_e = 1\}) \leq (1-\mu)G(\bar{x}^*)\right) \\ &\leq \exp\left(-\frac{\mu^3\cdot G(\bar{x}^*)}{2\upsilon}\right) \leq \exp\left(-\frac{\mu^3(1-e^{-1})}{2\upsilon}\frac{1-\mu}{1+\mu}\text{OPT}(\mathcal{I})\right) \leq \exp\left(-\frac{\mu^3\cdot \text{OPT}(\mathcal{I})}{16\cdot\upsilon}\right) \end{aligned} \tag{12}$$

The first and third inequality are due to (11).

Let ω be the event $\bar{x}^T \in (1-\mu)P$ and $g(T) \geq \frac{(1-\mu)^2}{1+\mu}(1-e^{-1})\text{OPT}(\mathcal{I})$. By applying the union bound over (8), (9), (10) and (12), we have

$$\Pr(\omega) \geq 1-\left(|B^r|\exp\left(-\frac{\mu^2}{12}\frac{1}{\delta}\right) - 2\sum_{j=1}^{k}\exp\left(-\frac{\mu^2}{12}|B_j|\right) - \exp\left(-\frac{\mu^3}{16}\frac{\text{OPT}(\mathcal{I})}{\upsilon}\right)\right) = 1-\gamma.$$

In case the event ω occurs, the algorithm executes Line 5, and by Lemma 11, $f(\cup_{b\in B}A_b) = f(T)$. Hence,

$$\mathbb{E}\left[f(\cup_{b\in B}A_b)\right] = \Pr(\omega)\cdot E\left[f(\cup_{b\in B}A_b)|\omega\right] \geq (1-\gamma)\frac{(1-\mu)^2}{1+\mu}(1-e^{-1})\text{OPT}(\mathcal{I}).$$

Also, the algorithm either returns an empty solution when Line 7 executes, or Line 5 executes. In the latter case the solution is feasible by Lemma 11. Therefore the algorithm always returns a feasible solution. ◀

4 Discussion

In this paper we presented a randomized $(1-e^{-1}-\varepsilon)$-approximation for the monotone submodular multiple knapsack problem. Our algorithm relies on three main building blocks. The structuring technique (Section 3.1) which converts a general instance to a leveled instance, the reduction to the block-constraint problem (Section 3.2.1) and a refined analysis of known algorithms for submodular optimization with a d-dimensional knapsack constraint (Section 3.2.2). While the structuring technique and the refined analysis seem to be fairly robust, the reduction to the block-constraint problem proved to be limiting when generalizations of the problem were considered.

A notable example is the *non-monotone* submodular multiple knapsack problem, in which the set function f is non-monotone. Unfortunately, when f is non-monotone the function g used for solving the block-constraint problem is not submodular. A variant of the block-constraint problem which does not alter the set function may be used to overcome this hurdle. However, this variant limits the knapsacks utilization and degrades the approximation ratio. Our preliminary results for the non-monotone case guarantee an approximation ratio of $\frac{1}{2}\cdot e^{-\frac{1}{2}} - \varepsilon \approx 0.303 - \varepsilon$ using this approach.

Another natural generalization of SMKP is monotone submodular optimization subject to a multiple knapsack and a *matroid* constraints, in which the solution $(A_b)_{b\in B}$ must also satisfy $\cup_{b\in B} A_b \in \mathcal{M}$ for a matoid $\mathcal{M}$. However, the matroid properties are not preserved throughout the reduction to the block-constraint problem, rendering existing techniques for submodular optimization with matroid and d-dimensional knapsack constraints [6] ineffective.

On the positive side, we believe that the techniques described in this paper can be extended to handle the problem for maximizing a monotone submodular function subject to a multiple knapsack constraint and an additional d-dimensional knapsack constraint, for a fixed d. We defer the details to the full version of the paper.

References

1 Niv Buchbinder and Moran Feldman. Submodular functions maximization problems. *Handbook of Approximation Algorithms and Metaheuristics*, 1:753–788, 2017.

2 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.

3 Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 182–196. Springer, 2007.

4 Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

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

6 Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding for matroid polytopes and applications. *arXiv preprint*, 2009. `arXiv:0909.4348`.

7 Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584. IEEE, 2010.

8 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.

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

10 Uriel Feige and Michel Goemans. Approximating the value of two power proof systems, with applications to max 2sat and max dicut. In *Proceedings Third Israel Symposium on the Theory of Computing and Systems*, pages 182–189. IEEE, 1995.

11 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.

12 Moran Feldman and Seffi Naor. *Maximization problems with submodular objective functions*. PhD thesis, Computer Science Department, Technion, 2013.

13 Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3):324–360, 2006.

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

15 Klaus Jansen. A fast approximation scheme for the multiple knapsack problem. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 313–324. Springer, 2012.

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

17 Ariel Kulik, Hadas Shachnai, and Tami Tamir. Approximations for monotone and nonmonotone submodular maximization with knapsack constraints. *Mathematics of Operations Research*, 38(4):729–739, 2013.

18 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.

19 George L Nemhauser and Laurence A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

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

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

22 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74, 2008.

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

A Basic Properties of Submodular Functions

▷ Claim 12. Let $f : 2^I \to \mathbb{R}_{\geq 0}$ be monotone and submodular function, then for any $A \subseteq B \subseteq I$ and $S \subseteq I$ it holds that $f(A \cup S) - f(A) \geq f(B \cup S) - f(B)$.

Proof. By the submodularity of f, we have

$$f(A \cup S) + f(B) \geq f(A \cup S \cup B) + f((A \cup S) \cap B) \geq f(B \cup S) + f(A)$$

where the second inequality follows from $A \subseteq (A \cup S) \cap B$ and the monotonicity of f. By rearranging the terms in the above we get

$$f(A \cup S) - f(A) \geq f(B \cup S) - f(B)$$

as required. ◁

▷ Claim 13. Let $f : 2^I \to \mathbb{R}_{\geq 0}$ be be a non-negative, monotone and submodular function, and let $S \subseteq I$. Then f_S is a submodular, monotone and non-negative function.

Proof. Let $A \subseteq I$. As f is monotone, we have

$$f_S(A) = f(S \cup A) - f(S) \geq 0.$$

That is, f is non-negative.

By claim 12, for any $A \subseteq B \subseteq I$ and $x \in I \setminus B$ it holds that

$$\begin{aligned} f_S(A \cup \{x\}) - f_S(A) &= f(S \cup A \cup \{x\}) - f(S \cup A) \\ &\geq f(S \cup B \cup \{x\}) - f(S \cup B) = f_S(B \cup \{x\}) - f_S(B). \end{aligned}$$

Therefore, f_S is submodular.

Finally, for $A \subseteq B \subseteq I$, as f is monotone we have that

$$f_S(A) = f(S \cup A) - f(S) \geq f(S \cup B) - f(S) = f_S(B).$$

Thus, f_S is also monotone. ◁

Proof of Claim 2. It is easy to see that g is non-negative, as f is non negative. In addition, for any two subsets $A \subseteq B \subseteq E$ we have $\cup_{(S,h)\in A} s \subseteq \cup_{(S,h)\in B} s$. Thus, since f is monotone, g is monotone as well.

All that is left to prove is that g is submodular. Consider subsets $A \subseteq B \subseteq E$ and $(S,h) \in E \setminus B$.

$$\begin{aligned} g(A \cup \{(S,h)\}) - g(A) &= f(\cup_{(S',h')\in A} S' \cup S) - f(\cup_{(S',h')\in A} s) \\ &\leq f(\cup_{(S',h')\in B} S' \cup S) - f(\cup_{(S',h')\in B} S') \\ &= g(B \cup \{(S,h)\}) - g(B). \end{aligned}$$

The inequality follows from Claim 12 and $\cup_{(S',h')\in A} S' \subseteq \cup_{(S',h')\in B} S'$. ◁

To prove Lemma 10 we first prove a special case of the lemma.

▶ **Lemma 14.** *Let $h : 2^\Omega \to \mathbb{R}_{\geq 0}$ be a submodular monotone and non-negative function, and let $S_1, \ldots, S_N \subseteq \Omega$. Then there is $1 \leq j^* \leq N$ such that*

$$h\left(\bigcup_{1\leq j\leq N,\ j\neq j^*} S_j\right) \geq \left(1 - \frac{1}{N}\right) h(S_1 \cup \ldots \cup S_N).$$

Proof. As h is submodular and monotone, using Claim 2, we have

$$\begin{aligned} h(S_1 \cup \ldots \cup S_N) - h(\emptyset) &= \sum_{j=1}^{N} \left(h(S_1 \cup \ldots \cup S_j) - h(S_1 \cup \ldots \cup S_{j-1})\right) \\ &\geq \sum_{j=1}^{N} \left(h\left(\bigcup_{j'=1}^{N} S_{j'}\right) - h\left(\bigcup_{1\leq j'\leq N, j'\neq j} S_{j'}\right)\right) \end{aligned}$$

Therefore there is $1 \leq j^* \leq N$ such that

$$h\left(\bigcup_{j=1}^{N} S_j\right) - h\left(\bigcup_{1\leq j\leq N, j\neq j^*} S_j\right) \leq \frac{1}{N}\left(h(S_1 \cup \ldots \cup S_N) - h(\emptyset)\right).$$

By rearranging the terms and using $h(\emptyset) \geq 0$ we obtain

$$h\left(\bigcup_{1\leq j\leq N, j\neq j^*} S_j\right) \geq \left(1 - \frac{1}{N}\right) h(S_1 \cup \ldots \cup S_N),$$

as required. ◀

Proof of Lemma 10. Let $h : 2^\Omega \to \mathbb{R}_+$ be a submodular, non-negative and monotone function, and $S_{i,1}, \ldots, S_{i,N} \subseteq \Omega$ for every $1 \leq i \leq M$.

Define $T_i = \bigcup_{j=1}^{N} S_{i,j}$. Now,

$$h\left(\bigcup_{i=1}^{M}\bigcup_{j=1}^{N} S_{i,j}\right) - h(\emptyset) = \sum_{i=1}^{M} h_{\left(\bigcup_{i'=1}^{i-1} T_{i'}\right)}(T_i) = \sum_{i=1}^{M} h_{\left(\bigcup_{i'=1}^{i-1} T_{i'}\right)}\left(\bigcup_{j=1}^{N} S_{i,j}\right).$$

By Lemma 14 for every $1 \leq i \leq M$ there is $1 \leq j_i^* \leq N$ such that

$$h_{\left(\bigcup_{i'=1}^{i-1} T_{i'}\right)}\left(\bigcup_{1\leq j\leq N,\ j\neq j_i^*} S_{i,j}\right) \geq \left(1 - \frac{1}{N}\right) h_{\left(\bigcup_{i'=1}^{i-1} T_{i'}\right)}(T_i).$$

Therefore,

$$
\begin{aligned}
&h\left(\bigcup_{i=1}^{M} \bigcup_{1\le j\le M,\ j\neq j^*_i} S_{i,j}\right) - h(\emptyset) = \sum_{i=1}^{M} h_{\left(\bigcup_{i'=1}^{i-1}\bigcup_{1\le j\le M,\ j\neq j^*_{i'}} S_{i',j}\right)}\left(\bigcup_{1\le j\le M,\ j\neq j^*_i} S_{i,j}\right)\\
&\geq \sum_{i=1}^{M} h_{\left(\bigcup_{i'=1}^{i-1} T_{i'}\right)}\left(\bigcup_{1\le j\le M,\ j\neq j^*_i} S_{i,j}\right) \geq \left(1-\frac{1}{N}\right)\sum_{i=1}^{M} h_{\left(\bigcup_{i'=1}^{i-1} T_{i'}\right)}(T_i)\\
&= \left(1-\frac{1}{N}\right)\left(h\left(\bigcup_{i=1}^{M}\bigcup_{j=1}^{N} S_{i,j}\right) - h(\emptyset)\right).
\end{aligned}
$$

The first inequality follows from $h_{T_1}(A) \geq h_{T_2}(A)$ for any $T_1 \subseteq T_2 \subseteq \Omega$ and $A \subseteq \Omega$ due to Claim 2. As h is non-negative, we conclude that

$$
h\left(\bigcup_{i=1}^{M} \bigcup_{1\le j\le M,\ j\neq j^*_i} S_{i,j}\right) \geq \left(1-\frac{1}{N}\right)\cdot h\left(\bigcup_{i=1}^{M}\bigcup_{j=1}^{N} S_{i,j}\right).
$$

◀

B Chernoff Bounds

In the analysis of the algorithm we use the following Chernoff-like bounds.

▶ **Lemma 15** (Theorem 3.1 in [13]). *Let $X = \sum_{i=1}^{n} X_i \cdot \lambda_i$ where $(X_i)_{i=1}^{n}$ is a sequence of independent Bernoulli random variable and $\lambda_i \in [0,1]$ for $1 \leq i \leq n$. Then for any $\varepsilon \in (0,1)$ and $\eta \geq \mathbb{E}[X]$ it holds that*

$$
\Pr\left(X > (1+\varepsilon)\eta\right) < \exp\left(-\frac{\varepsilon^2}{3}\eta\right)
$$

▶ **Lemma 16** (Theorem 1.3 in [6]). *Let $I = \{1,\ldots,n\}$, $\upsilon > 0$ and $f : 2^I \to \mathbb{R}_+$ be a monotone submodular function such that $f(\{i\}) - f(\emptyset) \leq \upsilon$ for any $i \in I$. Let $X_1,\ldots,X_n$ be independent random variables and $\eta = \mathbb{E}[f(\{i \in I | X_i = 1\})]$. Then for any $\varepsilon > 0$ it holds that*

$$
\mathbb{E}[f(\{i \in I | X_i = 1\}) \leq (1-\varepsilon)\eta] \leq \exp\left(-\frac{\eta\cdot\varepsilon^2}{2\upsilon}\right)
$$