MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Approximations for Monotone and Non-monotone Submodular Maximization with Knapsack Constraints*

Ariel Kulik
Computer Science Department, Technion, Haifa 32000, Israel, ariel.kulik@gmail.com

Hadas Shachnai
Computer Science Department, Technion, Haifa 32000, Israel, hadas@cs.technion.ac.il

Tami Tamir
School of Computer Science, The Interdisciplinary Center (IDC), Herzliya, Israel, tami@idc.ac.il

Submodular maximization generalizes many fundamental problems in discrete optimization, including Max-Cut in directed/undirected graphs, maximum coverage, maximum facility location and marketing over social networks.

In this paper we consider the problem of maximizing any submodular function subject to d knapsack constraints, where d is a fixed constant. We establish a strong relation between the discrete problem and its continuous relaxation, obtained through *extension by expectation* of the submodular function. Formally, we show that, for any non-negative submodular function, an α-approximation algorithm for the continuous relaxation implies a randomized $(\alpha-\varepsilon)$-approximation algorithm for the discrete problem. We use this relation to obtain an $(e^{-1}-\varepsilon)$-approximation for the problem, and a nearly optimal $(1-e^{-1}-\varepsilon)-$approximation ratio for the monotone case, for any $\varepsilon > 0$. We further show that the probabilistic domain defined by a continuous solution can be reduced to yield a polynomial size domain, given an oracle for the extension by expectation. This leads to a deterministic version of our technique.

Key words: Submodular maximization, knapsack constraints, maximum coverage, generalized assignment problem, randomization, approximation algorithms
MSC2000 subject classification: Primary: 68W20, 68W25, 90C27; secondary: 90C05
OR/MS subject classification: Primary: Analysis of Algorithms, Combinatorics; secondary: Suboptimal algorithms
History:

1. Introduction.

A real-valued function f, whose domain is all the subsets of a universe U, is called *submodular* if, for any $S,T \subseteq U$,

$$f(S)+f(T) \geq f(S\cup T)+f(S\cap T).$$

* A preliminary version of this paper appeared in the Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, New York, January 2009.

The concept of submodularity, which can be viewed as a discrete analog of convexity, plays a central role in combinatorial theorems and algorithms (see, e.g., [12] and the references therein, and the comprehensive surveys in [10, 26, 20]). Submodular maximization generalizes many fundamental problems in discrete optimization, including Max-Cut in directed/undirected graphs, maximum coverage, maximum facility location and marketing over social networks (see, e.g., [14]).

In many settings, including set covering or matroid optimization, the underlying submodular functions are monotone, meaning that $f(S) \le f(T)$ whenever $S \subseteq T$. In other settings, the function $f(S)$ is not necessarily monotone. A classic example of such a submodular function is $f(S) = \sum_{e \in \delta(S)} w(e)$, where $\delta(S)$ is a cut in a graph (or hypergraph) $G = (V, E)$ induced by a set of vertices $S \subseteq V$, and $w(e) \ge 0$ is the weight of an edge $e \subseteq E$. An example for a monotone submodular function is $f_{G,\bar{p}} : 2^L \to \mathbb{R}$, defined on a subset of vertices in bipartite graph $G = (L, R, E)$. For any $S \subseteq V$, $f_{G,\bar{p}}(S) = \sum_{v \in N(S)} p_v$, where $N(S)$ is the neighborhood function (i.e., $N(S)$ is the set of neighbors of S), and $p_v \ge 0$ is the profit of v, for any $v \in R$. The problem $\max\{f_{G,\bar{p}}(S) | |S| \le k\}$ is classical maximum coverage.

In this paper we consider the following problem of maximizing a non-negative *submodular* set function subject to d *knapsack constraints*, where d is a fixed constant (d-SUB). Given a d-dimensional budget vector $\bar{L}$, for some $d \ge 1$, and an oracle for a non-negative submodular set function f over a universe U, where each element $i \in U$ is associated with a d-dimensional non-negative cost vector $\bar{c}(i)$, we seek a subset of elements $S \subseteq U$ whose total cost is at most $\bar{L}$, such that $f(S)$ is maximized.

There has been extensive work on maximizing submodular *monotone* functions subject to matroid constraint.[1] For the special case of uniform matroid, i.e., the problem $\{\max f(S) : |S| \le k\}$, for some $k > 1$, Nemhauser et. al showed in [23] that a greedy algorithm yields a ratio of $1 - e^{-1}$ to the optimum. Later works presented greedy algorithms that achieve this ratio for other special matroids or for variants of maximum coverage (see, e.g., [1, 16, 25, 5]). For a general matroid constraint, Calinescu et al. showed in [3] that a scheme based on solving a continuous relaxation of the problem followed by *pipage rounding* (a technique introduced by Ageev and Sviridenko [1]) achieves the ratio of $1 - e^{-1}$ for maximizing submodular monotone functions that can be expressed as a sum of weighted rank functions of matroids. Subsequently, this result was extended by Vondrák [26] to general monotone submodular functions.

The bound of $1 - e^{-1}$ is the best possible for all of the above problems. This follows from the lower bound of Nemhauser and Wolsey [22] in the oracle model, and the later result of Feige [9] for the specific case of maximum coverage, under the assumption that $P \ne NP$.

Other variants of monotone submodular optimization were also considered. In [2], Bansal et al. studied the problem of maximizing a monotone submodular function subject to n knapsack constraints, for arbitrary $n \ge 1$, where each element appears in up to k constraints, and k is fixed. The paper presents a $\frac{8ek}{e-1}$ and $\frac{e^2k}{e-1} + o(k)$ approximations for this problem. Demaine and Zadimoghaddam [7] studied bi-criteria approximations for monotone submodular set function optimization.

The problem of maximizing a *non-monotone* submodular function has been studied as well. Feige et al. [10] considered (unconstrained) maximization of a general non-monotone submodular function. The paper gives several (randomized and deterministic) approximation algorithms, as well as hardness results, also for the special case where the function is *symmetric*.

Lee et al. [20] studied the problem of maximizing a general submodular function under linear and matroid constraints. They proposed algorithms that achieve approximation ratio of $1/5 - \varepsilon$ for the problem with d linear constraints and a ratio of $1/(d + 2 + 1/d + \varepsilon)$ for d matroid constraints, for any fixed integer $d \ge 1$.

[1] A (weighted) matroid is a system of 'independent subsets' of a universe, which satisfies certain *hereditary* and *exchange* properties [24].

Improved lower and upper bounds for non-constrained and constrained submodular maximization were recently derived by Gharan and Vondrák [13]. However, this paper does not consider knapsack constraints.

Several fundamental algorithms for submodular maximization (see, e.g., [1, 3, 26, 20]) use a continuous extension of submodular function, to which we refer as *extension by expectation*. Given a submodular function $f: 2^U \to \mathbb{R}$, we define $F: [0,1]^U \to \mathbb{R}$. For any $\bar{y} \in [0,1]^U$, let $D \subseteq U$ be a random variable such that each element $i \in U$ is chosen independently to be in D with probability y_i (we say that $D \sim \bar{y}$). Then

$$F(\bar{y}) = E[f(D)] = \sum_{D \subseteq U} \left(f(D) \prod_{i \in D} y_i \prod_{i \notin D} (1 - y_i) \right).$$

The general framework of these algorithms is to obtain first a fractional solution for the continuous extension, followed by rounding which yields a solution for the discrete problem. In this paper we deal mainly in the rounding step within the scheme above.

Using the definition of F, we define the continuous relaxation of our problem called *continuous d-SUB*. Let $P = \{\bar{y} \in [0,1]^U \,|\, \sum_{i \in U} y_i \bar{c}(i) \le \bar{L}\}$ be the polytope of the instance, then the problem is to find $\bar{y} \in P$ for which $F(\bar{y})$ is maximized. For $\alpha \in (0,1]$, an algorithm $\mathcal{A}$ yields α-approximation for the continuous problem with respect to a submodular function f, if for any assignment of non-negative costs to the elements, and for any non-negative budget, $\mathcal{A}$ finds a feasible solution for continuous d-SUB of value at least $\alpha\mathcal{O}$, where $\mathcal{O}$ is the value of an optimal (integral) solution for d-SUB with the given costs and budget.

For some specific families of submodular functions, linear programming can be used to derive such approximation algorithms (see e.g [1, 3]). For monotone submodular functions, Vondrák presented in [26] a $(1 - e^{-1} - o(1))$-approximation algorithm for the continuous problem. Subsequently, Lee et al. [20] considered the problem of maximizing *any* submodular function with multiple knapsack constraints and developed a $(\frac{1}{4} - o(1))$-approximation algorithm for the continuous problem; however, noting that the rounding method of [19],[2] which proved useful for monotone functions, cannot be applied in the non-monotone case, a $(\frac{1}{5} - \varepsilon)$-approximation was obtained for the discrete problem, by using simple randomized rounding. This gap of approximation ratio between the continuous and the discrete case led us to further develop the technique in [19], so that it can be applied also for non-monotone functions.

Subsequent to our study of maximizing monotone submodular functions subject to multiple knapsack constraints [19], Chekuri et al. [6] showed that, by using a more sophisticated rounding technique, the algorithm in [19] can be applied to derive a $(1 - e^{-1} - \varepsilon)$-approximation for maximizing a submodular function subject to d knapsack constraints and a matroid constraint. Specifically, given a fractional solution for the problem, the authors define a probability distribution over the solution space, such that all of elements in the domain of the distribution are inside the matroid; these elements also satisfy Chernoff-type concentration bounds, which can be used to prove some of the probabilistic claims in [19]. The desired approximation ratio is obtained by using the algorithm of [19] with sampling replaced by the above distribution in the rounding step.

Independently, Fadaei et al. considered d-SUB in [8]. The paper claims to obtain a bound of $(0.25 - \varepsilon)$ for the problem with non-monotone submodular function. However, we could not verify the correctness of the results.[3]

[2] The paper [19] is a preliminary version of this paper.

[3] Specifically, in the proof of Theorem 3, the function G can get negative values; thus, the algorithm of [20] cannot be applied. Overcoming this obstacle is one of the major contributions of our paper while extending the results in [19].

In [27] Chekuri et al. presented a 0.325 approximation algorithm for d-SUB. To do so, the authors obtain a $(0.325+\delta)$-approximation for continuous d-SUB, where $\delta > 0$ is a small fixed constant. The paper also presents a generic approach, called *contention resolution (CR)*, for rounding continuous solutions subject to various constraints. The approach used for the special case of linear constraints is based on the technique developed in Section 2, while a main contribution of CR schemes is in the integration of different rounding techniques, which enables to simultaneously handle multiple types of constraints. Combining the algorithm for continuous d-SUB with CR scheme the authors obtain the improved ratio of 0.325.

Recently, Feldman et al. [11] presented a continuous greedy algorithm that yields an e^{-1}-approximation for the problem $\max\{F(x)|x \in P\}$, where P is a down-closed solvable polytope. By coupling the continuous greedy algorithm with the rounding technique presented in this paper, they obtained an $(e^{-1} - \varepsilon)$ approximation for d-SUB.

1.1. Our results. In this paper we establish a strong relation between the problem of maximizing any submodular function subject to d knapsack constraints and its continuous relaxation.[4] Formally, we show (in Theorem 2) that for any non-negative submodular function, an α-approximation algorithm for the continuous relaxation implies a randomized $(\alpha - \varepsilon)$-approximation algorithm for the discrete problem. We use this relation to obtain approximation ratio of $(e^{-1} - \varepsilon)$ for d-SUB, for any $\varepsilon > 0$ by applying the continuous greedy algorithm of [11]. For the case where the objective function is monotone, we obtain a nearly optimal $(1 - e^{-1} - \varepsilon)$ approximation using the results of [26], for any $\varepsilon > 0$. An important consequence of the above relation is that for any class of submodular functions, a future improvement of the approximation ratio for the continuous problem, to a factor of α, immediately implies an approximation ratio of $(\alpha - \varepsilon)$ for the original instance.

Our technique applies random sampling on the solution space, using a distribution defined by the fractional solution for the problem. In Section 2.5 we show how to convert a feasible solution for the continuous problem to another feasible solution with up to $O(\log |U|)$ fractional entries, given an oracle to the extension by expectation. This facilitates the usage of exhaustive search instead of sampling, which leads to a deterministic version of our technique. Specifically, we obtain a deterministic $(e^{-1} - \varepsilon)$-approximation for general instances and $(1 - e^{-1} - \varepsilon)$-approximation for instances where the submodular function is monotone. For the special case of maximum coverage with d knapsack constraints, that is, d-SUB where the objective function is $f = f_{G,\bar{p}}$ for a given bipartite graph G and profits $\bar{p}$, this result leads to a deterministic $(1 - e^{-1} - \varepsilon)-$approximation algorithm, since the extension by expectation of $f_{G,\bar{p}}$ can be deterministically evaluated. We note that none of the earlier results leads to a deterministic algorithm for this problem, or to a deterministic rounding procedure for the fractional solution obtained for continuous maximization of submodular function with knapsack constraints.

Remark: Our study of maximizing submodular functions encompasses also a generalization of *maximum coverage* and a budgeted variant of the *generalized assignment problem (GAP)*. These two problems can be cast as submodular optimization problems, however, the resulting universe sizes are non-polynomial in the input size. As our algorithms cannot be directly applied to these problems, more specialized techniques need to be used to obtain approximation algorithm for each of the problems. Using the technique in Section 2, a $(1 - e^{-1} - \varepsilon)$-approximation can be derived for both problems. The detailed results can be found in [17]. Here, we focus on our general technique for maximizing submodular functions.

[4] Some basic properties of submodular functions are given in Appendix A.

2. Maximizing submodular functions. In this section we describe our framework for maximizing a submodular set function subject to multiple linear constraints. For short, we call this problem d-SUB.

2.1. Preliminaries.

Notation: An essential component in our framework is the distinction between elements by their costs. We say that an element i is *small* if $\bar{c}(i) \le \varepsilon^3 \bar{L}$; otherwise, the element is *big*.

Given a universe U, we call a subset of elements $S \subseteq U$ *feasible* if the total cost of elements in S is bounded by $\bar{L}$. We say that S is ε-*nearly feasible* (or *nearly feasible*, if ε is known from the context) if the total cost of the elements in S is bounded by $(1+\varepsilon)\bar{L}$. We refer to $f(S)$ as the value of S. Similar to the discrete case, $\bar{y} \in [0,1]^U$ is feasible if $\bar{y} \in P$.

For any subset $T \subseteq U$, we define $f_T : 2^U \to \mathbb{R}_+$ by $f_T(S) = f(S \cup T) - f(T)$. It is easy to verify that if f is a submodular set function then f_T is also a submodular set function. Finally, for any set $S \subseteq U$, we define $c_r(S) = \sum_{i \in S} c_r(i)$, where $1 \le r \le d$, and $\bar{c}(S) = \sum_{i \in S} \bar{c}(i)$. For a fractional solution $\bar{y} \in [0,1]^U$, we define $c_r(\bar{y}) = \sum_{i \in U} c_r(i) \cdot y_i$ and $\bar{c}(\bar{y}) = \sum_{i \in U} \bar{c}(i) \cdot y_i$.

Overview: Our algorithm consists of two phases, to which we refer as *rounding procedure* and *profit enumeration*. The rounding procedure yields an $(\alpha - O(\varepsilon))$-approximation for instances in which there are no big elements, using an α-approximate solution for the continuous problem. It relies heavily on Theorem 1 that gives some conditions on the probabilistic domain of solutions; these conditions guarantee that the expected profit of the resulting nearly feasible solution is high. This solution is then converted to a feasible one, by using a fixing procedure. We first present a randomized version and later show how to derandomize the rounding procedure.

The profit enumeration phase uses enumeration over the most profitable elements in an optimal solution; then it reduces a general instance to another instance with no big elements, on which we apply the rounding procedure.

Finally, we combine the above results with an algorithm for the continuous problem (e.g., the algorithm of [26], or [11]) to obtain approximation algorithm for d-SUB.

2.2. A probabilistic theorem. We first prove a general probabilistic theorem.

THEOREM 1. *Given a d-SUB instance with no big elements, let $\bar{x} \in [0,1]^U$ be a feasible fractional solution such that $F(\bar{x}) \ge \mathcal{O}/5$, where $\mathcal{O}$ is the optimal solution for the d-SUB instance. Let $D \subseteq U$ be a random set such that $D \sim \bar{x}$ (i.e., for all $i \in U$, $i \in D$ independently with probability x_i), and let D' be a random set such that $D' = D$ if D is ε-nearly feasible, and $D' = \emptyset$ otherwise. Then D' is always ε-nearly feasible, and $E[f(D')] \ge (1 - O(\varepsilon))F(\bar{x})$.*

Proof. Let X_i $(1 \le i \le n)$ be an indicator random variable for $i \in D$. By the definition of D these indicators are independent. For any dimension $1 \le r \le d$, let $R_r = \frac{c_r(D)}{L_r}$, and define $R = \max_r R_r$, then R denotes the maximal relative deviation of the cost from the r-th entry in the budget vector, where the maximum is taken over $1 \le r \le d$.

CLAIM 1. *For any $\ell > 1$,*

$$Pr[R > \ell] < \frac{d\varepsilon^3}{(\ell-1)^2}.$$

Proof. For any dimension $1 \le r \le d$, it holds that $E[c_r(D)] = \sum_{i=1}^n E[c_{i,r} \cdot X_i] \le L_r$. Then,

$$\begin{aligned} Var[c_r(D)] &= \sum_{i=1}^n Var[c_{i,r} \cdot X_i] \le \sum_{i=1}^n E[c_{i,r}^2 \cdot X_i] \\ &\le \sum_{i=1}^n E[c_{i,r} \cdot X_i] \cdot \varepsilon^3 L_r = \varepsilon^3 L_r \sum_{i=1}^n E[c_{i,r} \cdot X_i] \le \varepsilon^3 L_r^2. \end{aligned}$$

The first inequality holds since $Var[X] \leq E[X^2]$, and the second inequality follows from the fact that $c_{i,r} \leq \varepsilon^3 L_r$ for $1 \leq i \leq n$ as all elements are small. Recall that, by the Chebyshev-Cantelli inequality, for any $t > 0$ and a random variable Z,

$$Pr\left[Z - E[Z] \geq t\right] \leq \frac{Var[Z]}{Var[Z] + t^2}.$$

Thus, for any dimension $1 \leq r \leq d$,

$$\begin{aligned} Pr[R_r > \ell] &= Pr[c_r(D) > \ell \cdot L_r] \\ &\leq Pr\left[c_r(D) - E[c_r(D)] > (\ell - 1)L_r\right] \\ &\leq \frac{\varepsilon^3 L_r^2}{(\ell-1)^2 L_r^2} \leq \frac{\varepsilon^3}{(\ell-1)^2}, \end{aligned}$$

and by the union bound, we get that

$$Pr[R > \ell] \leq \frac{d\varepsilon^3}{(\ell-1)^2}.$$

■

Define an indicator random variable N such that $N = 1$ if D is ε-nearly feasible, and $N = 0$ otherwise. Thus,we get that $N = 0$ iff $R > (1+\varepsilon)$, and by the previous claim we have

Claim 2. *$Pr[N = 0] \leq d\varepsilon$.*

Claim 3. *For any integer $\ell > 1$, if $R \leq \ell$ then*

$$f(D) \leq 2d\ell \cdot \mathcal{O}.$$

Proof Sketch: We note that the set D can be partitioned to $2d\ell$ sets $D_1, \ldots D_{2d\ell}$ such that each of these sets is a feasible solution. Hence, $f(D_i) \leq \mathcal{O}$. By Lemma 8, we have that $f(D) \leq f(D_1) + \ldots + f(D_{2d\ell}) \leq 2d\ell\mathcal{O}$.[5] □

Combining the above results we have

Claim 4. $E[f(D')] \geq (1 - O(\varepsilon))E[f(D)]$.

Proof. By Claims 1, 2 and 3, we have that

$$\begin{aligned} E[f(D)] &= E\left[f(D)|\ N=1\right] \cdot Pr\left[N=1\right] + E\left[f(D)|\ N=0 \wedge (R<2)\right] \cdot Pr\left[N=0 \wedge (R<2)\right] \\ &\quad + \sum_{\ell \geq 1} E\left[f(D)|\ N=0 \wedge (2^\ell \leq R < 2^{\ell+1})\right] \cdot Pr\left[N=0 \wedge (2^\ell \leq R < 2^{\ell+1})\right] \\ &\leq E[f(D)|\ N=1] \cdot Pr\left[N=1\right] + 4d^2\varepsilon \cdot \mathcal{O} + \ d^2\varepsilon^3 \cdot \mathcal{O} \cdot \sum_{\ell \geq 1} \frac{2^{\ell+2}}{(2^{\ell-1})^2}. \end{aligned}$$

Since the last summation is a constant, and $E[f(D)] \geq \mathcal{O}/5$, we have that

$$E[F(D)] \leq E[f(D)|N=1]Pr\left[N=1\right] + \varepsilon \cdot c \cdot E[F(D)],$$

[5] We give the detailed proof in Appendix B

where $c > 0$ is some constant. It follows that

$$(1 - O(\varepsilon))E[f(D)] \le E[f(D)|N = 1] \cdot Pr[N = 1].$$

Finally, since $D' = D$ if $N = 1$ and $D' = 0$ otherwise, we have that

$$E[f(D')] = E[f(D)|N = 1] \cdot Pr[N = 1] \ge (1 - O(\varepsilon))E[f(D)].$$

■

By definition, D' is always ε-nearly feasible. This completes the proof of the theorem. ■

2.3. Rounding instances with no big elements. In this section we present an $(\alpha - O(\varepsilon))$-approximation algorithm for d-SUB inputs with no big elements, given an α-approximate solution for the continuous problem. Inputs with no big elements are easier to tackle. Indeed, any nearly feasible solution for such input can be converted to a feasible one, with only a small harm to the total value.

LEMMA 1. *Let $S \subseteq U$ be an ε-nearly feasible solution with no big elements, then S can be converted in polynomial time to a feasible solution $S' \subseteq S$, such that $f(S') \ge (1 - O(\varepsilon)) f(S)$.*

Proof. In fixing the solution S we handle each dimension separately. For any dimension $1 \le r \le d$, if $c_r(S) \le L_r$ then no modification is needed; otherwise, $c_r(S) > L_r$. Since all elements in S are small, we can partition S into ℓ disjoint subsets $S_1, S_2, \ldots, S_\ell$ such that $\varepsilon L_r \le c_r(S_j) < (\varepsilon + \varepsilon^3)L_r$ for any $1 \le j \le \ell$, where $\ell = \Omega(\varepsilon^{-1})$. Since the function f is submodular, by Lemma 10, we have that $f(S) \ge \sum_{j=1}^{\ell} f_{S \setminus S_j}(S_j)$. Hence, there exists a value $j \in \{1, 2 \ldots, \ell\}$ such that $f_{S \setminus S_j}(S_j) \le \frac{f(S)}{\ell} = f(S) \cdot O(\varepsilon)$ (note that $f_{S \setminus S_j}(S_j)$ may be negative). Now, $c_r(S \setminus S_j) \le L_r$, and $f(S \setminus S_j) \ge (1 - O(\varepsilon))f(S)$. We repeat this step for all $1 \le r \le d$ to obtain a feasible set S' satisfying $f(S') \ge (1 - O(\varepsilon))f(S)$. ■
Combined with Theorem 1, we have the following rounding algorithm.

Randomized Rounding Algorithm for d-SUB with No Big Elements
Input: A d-SUB instance, a feasible solution $\bar{x}$ for the continuous problem, with $F(\bar{x}) \ge \mathcal{O}/5$.
1. Define a random set $D \sim \bar{x}$. Let $D' = D$ if D is ε-nearly feasible, and $D' = \emptyset$ otherwise.
2. Convert D' to a feasible set D'' as in the proof of Lemma 1 and return D''.

Clearly, the algorithm returns a feasible solution for the problem. By Theorem 1, $E[f(D')] \ge (1 - O(\varepsilon))F(\bar{x})$. By Lemma 1, $E[f(D'')] \ge (1 - O(\varepsilon))F(\bar{x})$. Hence, we have

LEMMA 2. *For any instance of d-SUB with no big elements, any feasible solution $\bar{x}$ for the continuous problem with $F(\bar{x}) \ge \mathcal{O}/5$ can be converted to a feasible solution for d-SUB in polynomial running time with expected profit at least $(1 - O(\varepsilon)) \cdot F(\bar{x})$.*

2.4. Approximation algorithm for d-SUB. Given an instance of d-SUB and a subset $T \subseteq U$, define another instance of d-SUB, to which we refer as the *residual problem with respect to* T, with f remaining the objective function. The budget for the residual problem is $\bar{L}' = \bar{L} - \bar{c}(T)$, and the universe U' consists of all elements $i \in U \setminus T$ such that $\bar{c}(i) \le \varepsilon^3 \bar{L}'$, and all elements in T. Formally,

$$U' = T \cup \left\{ i \in U \setminus T | \ \bar{c}(i) \le \varepsilon^3 \bar{L}' \right\}.$$

The new cost of element i is $c'(i) = c(i)$ for any $i \in U' \setminus T$, and $c'(i) = 0$ for any $i \in T$. It follows that there are no big elements in the residual problem. Let S be a feasible solution for the residual problem with respect to T. Then $\bar{c}(S) \le \bar{c}'(S) + \bar{c}(T) \le \bar{L}' + \bar{c}(T) = \bar{L}$. Thus, any feasible solution for the residual problem is also feasible for the original instance.

Let $\mathcal{O}_T$ denote the optimal solution for the residual problem with respect to T. Let $\mathcal{O} = \{i_1, \ldots, i_m\}$ be an optimal solution for a d-SUB instance (we use $\mathcal{O}$ to denote both an optimal sub-collection of elements and the optimal value). For $\ell \geq 1$, let $K_\ell = \{i_1, \ldots, i_\ell\}$, and assume that the elements are ordered by their residual profits, i.e., $i_\ell = \arg\max_{i \in \mathcal{O} \setminus K_{\ell-1}} f_{K_{\ell-1}}(\{i\})$.

Lemma 3. *Consider $T = K_h$ where $h = \lceil d \cdot \varepsilon^{-4} \rceil$, then $\mathcal{O}_T \geq (1-\varepsilon)\mathcal{O}$.*

Proof. Define $\mathcal{O}' = \mathcal{O} \cap U'$ (U' is the universe of the residual problem with respect to T). Clearly, the set $\mathcal{O}'$ is a feasible solution for the residual problem with respect to T. We show a lower bound for $f(\mathcal{O}')$. The set $R = \mathcal{O} \setminus \mathcal{O}'$ consists of elements in $\mathcal{O} \setminus T$ that are big with respect to the residual instance. The total cost of elements in R is bounded by $\bar{L}'$ (since $\mathcal{O}$ is a feasible solution), and thus $|R| \leq \varepsilon^{-3} \cdot d$.

Since $T = K_h$, for any $j \in \mathcal{O} \setminus T$ it holds that $f_T(j) \leq \frac{f(T)}{|T|}$, and we get that

$$f_T(R) \leq \sum_{j \in R} f_T(\{j\}) \leq \varepsilon^{-3} \cdot d \frac{f(T)}{|T|} = \varepsilon f(T) \leq \varepsilon \mathcal{O}.$$

Thus, $f_{\mathcal{O}'}(R) \leq f_T(R) \leq \varepsilon \mathcal{O}$. Since

$$f(\mathcal{O}) = f(\mathcal{O}') + f_{\mathcal{O}'}(R) \leq f(\mathcal{O}') + \varepsilon f(\mathcal{O}),$$

we have that $f(\mathcal{O}') \geq (1-\varepsilon) f(\mathcal{O})$. ∎

Consider the following algorithm.

Approximation Algorithm for d-SUB

Input: A d-SUB instance and an α-approximation algorithm $\mathcal{A}$ for the continuous problem with respect to the function f.

1. For any $T \subseteq U$ such that $|T| \leq h = \lceil d \cdot \varepsilon^{-4} \rceil$
   (a) Use $\mathcal{A}$ to obtain an α-approximate solution $\bar{x}$ for the continuous residual problem with respect to T.
   (b) Use the Randomized Rounding Algorithm of Section 2.3 to convert $\bar{x}$ to a feasible solution S for the residual problem.
2. Return the best solution found.

Lemma 4. *The above approximation algorithm returns an $(\alpha - O(\varepsilon))$-approximate solution for d-SUB and uses a polynomial number of calls to algorithm $\mathcal{A}$.*

Proof. By Lemma 2, in each iteration the algorithm finds a feasible solution S for the residual problem. Hence, the algorithm always returns a feasible solution for the given d-SUB instance.

Consider the iteration in which $T = K_h$. By Lemma 3, we have $f(\mathcal{O}_T) \geq (1-\varepsilon) f(\mathcal{O})$. Thus, in this iteration we get a solution $\bar{x}$ for the residual problem with $F(\bar{x}) \geq \alpha(1-\varepsilon) f(\mathcal{O})$, and the solution S obtained after the rounding satisfies $f(S) \geq (1 - O(\varepsilon))\alpha f(\mathcal{O})$. ∎

We summarize in the next result.

Theorem 2. *Let f be a submodular function, and suppose there is a polynomial time α-approximation algorithm for the continuous problem with respect to f. Then there is a polynomial time randomized $(\alpha - \varepsilon)$-approximation algorithm for d-SUB with respect to f, for any $\varepsilon > 0$.*

Since there is a e^{-1}-approximation algorithm for general instances of continuous d-SUB [11], we have

Theorem 3. *There is a polynomial time randomized $(e^{-1} - \varepsilon)$-approximation algorithm for d-SUB, for any $\varepsilon > 0$.*

Since there is a $(1-e^{-1}-o(1))$ approximation algorithm for d-SUB with monotone objective function [26], we have

THEOREM 4. *There is a polynomial time randomized $(1-e^{-1}-\varepsilon)$-approximation algorithm for d-SUB with monotone objective function, for any $\varepsilon > 0$.*

2.5. Derandomization. In this section we show how the algorithm of Section 2.3 can be derandomized, assuming we have an oracle for F, the extension by expectation of f. For some families of submodular functions, F can be directly evaluated; for a general function f, F can be evaluated with high accuracy by sampling f, as in [26].

The main idea is to reduce the number of fractional entries in the fractional solution $\bar{x}$, so that the number of values a random set $D \sim \bar{x}$ can get is polynomial in the input size (for a fixed value of ε). Then, we go over all the possible values, and we are promised to obtain a solution of high value.

A key tool in our derandomization is the *pipage rounding* technique of Ageev and Sviridenko [1]. We give below a brief overview of the technique. For any element $i \in U$, define the unit vector $\bar{i} \in \{0,1\}^U$, in which $\bar{i}_j = 0$ for any $j \neq i$, and $\bar{i}_i = 1$. Given a fractional solution $\bar{x}$ for the problem and two elements i, j, such that x_i and x_j are both fractional, consider the vector function $\bar{x}_{i,j}(\delta) = \bar{x} + \delta\bar{i} - \delta\bar{j}$ (Note that $\bar{x}_{i,j}(\delta)$ is equal to $\bar{x}$ in all entries except i, j). Let $\delta^+_{\bar{x},i,j}$ and $\delta^-_{\bar{x},i,j}$ (for short, δ^+ and δ^-) be the maximal and minimal value of δ for which $\bar{x}_{i,j}(\delta) \in [0,1]^U$. In both $\bar{x}_{i,j}(\delta^+), \bar{x}_{i,j}(\delta^-)$, the entry of either i or j is integral.

Define $F^{\bar{x}}_{i,j}(\delta) = F(\bar{x}_{i,j}(\delta))$ over the domain $[\delta^-, \delta^+]$. The function $F^{\bar{x}}_{i,j}$ is convex (see [4] for a detailed proof), thus $\bar{x}' = \arg\max_{\{\bar{x}_{i,j}(\delta^+), \bar{x}_{i,j}(\delta^-)\}} F(\bar{x})$ has fewer fractional entries than $\bar{x}$, and $F(\bar{x}') \geq F(\bar{x})$. By appropriate selection of i, j, such that $\bar{x}'$ maintains feasibility (in some sense), we can repeat the above step to gradually decrease the number of fractional entries. We use the technique to prove the next result.

LEMMA 5. *Let $\bar{x} \in [0,1]^U$ be a solution having k or less fractional entries (i.e., $|\{i \mid 0 < x_i < 1\}| \leq k$), and $\bar{c}(\bar{x}) \leq \bar{L}$ for some $\bar{L}$. Then $\bar{x}$ can be converted to a vector $\bar{x}'$ with at most $k' = \left(\frac{8\ln(2k)}{\varepsilon}\right)^d = O(ln^d(k))$ fractional entries, such that $\bar{c}(\bar{x}') \leq (1+\varepsilon)\bar{L}$, and $F(\bar{x}') \geq F(\bar{x})$, in time polynomial in k.*

Proof. Let $U' = \{i \mid 0 < x_i < 1\}$ be the set of all fractional entries. We define a new cost function $\bar{c}'$ over the elements in U.

$$c'_r(i) = \begin{cases} c_r(i) & i \notin U' \\ 0 & c_r(i) \leq \dfrac{\varepsilon \cdot L_r}{2k} \\ \dfrac{\varepsilon \cdot L_r}{2k}(1+\varepsilon/2)^j & \dfrac{\varepsilon \cdot L_r}{2k}(1+\varepsilon/2)^j \leq c_r(i) < \dfrac{\varepsilon \cdot L_r}{2k}(1+\varepsilon/2)^{j+1} \end{cases}$$

Note that for any $i \in U'$, $\bar{c}'(i) \leq \bar{c}(i)$, and

$$c_r(i) \leq (1+\frac{\varepsilon}{2})c'_r(i) + \frac{\varepsilon \cdot L_r}{2k},$$

for all $1 \leq r \leq d$. The number of different values $c'_r(i)$ can get for $i \in U'$ is bounded by $\frac{8\ln(2k)}{\varepsilon}$ (since all elements are small, and $\ln(1+x) \geq x/2$). Hence, the number of different values $\bar{c}'(i)$ can get for $i \in U'$ is bounded by $k' = \left(\frac{8\ln(2k)}{\varepsilon}\right)^d$.

We start with $\bar{x}' = \bar{x}$, and while there are $i, j \in U'$ such that x'_i and x'_j are both fractional, and $\bar{c}'(i) = \bar{c}'(j)$, define $\delta^+ = \delta^+_{\bar{x}',i,j}$ and $\delta^- = \delta^-_{\bar{x}',i,j}$. Since i and j have the same cost (by $\bar{c}'$), it holds that $\bar{c}'(\bar{x}_{i,j}(\delta^+)) = \bar{c}'(\bar{x}_{i,j}(\delta^-)) = \bar{c}'(\bar{x})$. If $F^{\bar{x}}_{i,j}(\delta^+) \geq F(\bar{x})$, then set $\bar{x}'' = \bar{x}_{i,j}(\delta^+)$, otherwise $\bar{x}'' = \bar{x}_{i,j}(\delta^-)$.

In both cases $F(\bar{x}'') \geq F(\bar{x}')$ and $\bar{c}'(\bar{x}'') = \bar{c}'(\bar{x}')$. Now, repeat this step with $\bar{x}' = \bar{x}''$. Since in each iteration the number of fractional entries in $\bar{x}'$ decreases, the process will terminate (after at most k iterations) with a vector $\bar{x}'$ such that $F(\bar{x}') \geq F(\bar{x})$, $\bar{c}'(\bar{x}') = \bar{c}'(\bar{x}) \leq \bar{L}$, and there are no two elements $i, j \in U'$ with $\bar{c}'(i) = \bar{c}'(j)$, where x'_i and x'_j are both fractional. Also, for any $i \notin U'$, the entry x'_i is integral (since x_i was integral and the entry was not modified by the process). Thus, the number of fractional entries in $\bar{x}'$ is at most k'. Now, for any dimension $1 \leq r \leq d$,

$$
\begin{aligned}
c_r(\bar{x}') &= \sum_{i \notin U'} x'_i c_r(i) + \sum_{i \in U'} x'_i c_r(i) \\
&\leq (1+\varepsilon/2) \cdot \sum_{i \notin U'} x'_i \cdot c'_r(i) + \sum_{i \in U'} x'_i \left((1+\varepsilon/2) c'_r(i) + \frac{\varepsilon \cdot L_r}{2k} \right) \\
&= (1+\varepsilon/2) \cdot \sum_{i \in U} x'_i \cdot c'_r(i) + \sum_{i \in U'} x_i \frac{\varepsilon \cdot L_r}{2k} \leq (1+\varepsilon) L_r.
\end{aligned}
$$

This completes the proof. ∎

Using the above lemma, we can reduce the number of fractional entries in $\bar{x}$ to a number that is poly-logarithmic in k. However, the number of values $D \sim \bar{x}$ remains super-polynomial. To reduce further the number of fractional entries, we apply the above step twice, that is, we convert $\bar{x}$ with at most $|U|$ fractional entries to $\bar{x}'$ with at most $k' = c \cdot \ln^d |U|$ entries, where $c = c(\varepsilon, d)$ is a fixed constant. Repeating the conversion, we obtain $\bar{x}''$, in which the number of fractional entries is bounded by

$$k'' = c \cdot \ln^d(k') = c \cdot \ln^d(c \cdot \ln^d |U|) \leq c'(\ln \ln |U|)^d = o(\ln |U|),$$

where c' is some constant (for a constant ε and d).

Lemma 6. *Given a vector $\bar{L}$ and a constant $\varepsilon > 0$, let $\bar{x} \in [0,1]^U$ be a vector satisfying $\bar{c}(\bar{x}) \leq \bar{L}$. Then $\bar{x}$ can be converted in time polynomial in $|U|$ to a vector $\bar{x}'$ with at most $k'' = O(\log |U|)$ fractional entries, such that $\bar{c}(\bar{x}') \leq (1+\varepsilon)^2 \bar{L}$, and $F(\bar{x}') \geq F(\bar{x})$,*

The next result follows immediately from Lemma 1 ($\mathcal{O}$ is the value of an optimal solution for d-SUB).

Lemma 7. *Given $\bar{x} \in [0,1]^U$ such that $\bar{x}$ is a feasible fractional solution with $F(\bar{x}) \geq \mathcal{O}/5$ for a d-SUB instance with no big elements, there exists a realization of the random variable $D \sim \bar{x}$, such that the solution $\mathcal{D}$ is nearly feasible, and $F(\mathcal{D}) \geq (1 - O(\varepsilon)) F(\bar{x})$.*

Consider the following rounding algorithm.

Deterministic Rounding Algorithm for d-SUB with No Big Elements
Input: A d-SUB instance, a feasible solution $\bar{x}$ for the continuous problem, with $F(\bar{x}) \geq \mathcal{O}/5$.

1. Define $\bar{x}' = (1+\varepsilon)^{-2} \cdot \bar{x}$ (note that $F(\bar{x}') \geq (1+\varepsilon)^{-2} \cdot F(\bar{x})$).
2. Convert $\bar{x}'$ to $\bar{x}''$ such that $\bar{x}''$ is fractionally feasible, the number of fractional entries in $\bar{x}''$ is $O(\log |U|)$, and $F(\bar{x}) \geq (1+\varepsilon)^{-2} F(\bar{x}'')$, as in Lemma 6.
3. Enumerate over all possible realizations of $D \sim \bar{x}''$. For each such realization, if the solution $\mathcal{D}$ is ε-nearly feasible convert it to a feasible solution $\mathcal{D}'$ (see Lemma 1). Return the solution with maximum value among the feasible solutions found.

By Theorem 1, the algorithm returns a feasible solution of value at least $(1 - O(\varepsilon)) F(\bar{x})$. Also, the running time of the algorithm is polynomial when ε is a fixed constant. Replacing the randomized rounding step in the algorithm of Section 2.4 with the above Deterministic Rounding Algorithm, we get the following result.

THEOREM 5. *Let f be a submodular function, and assume we have an oracle for F. If there is a deterministic polynomial time α-approximation algorithm for the continuous problem with respect to f, then there is a polynomial time deterministic $(\alpha-\varepsilon)$-approximation algorithm for d-SUB with respect to f, for any $\varepsilon > 0$.*

We note that, given an oracle to F, both the algorithms of [26] and [11] for the continuous problem are deterministic, thus we get the following.

THEOREM 6. *Given an oracle for F, there is a polynomial time deterministic $(1-e^{-1}-\varepsilon)$-approximation algorithm for d-SUB with a monotone function, for any $\varepsilon > 0$.*

THEOREM 7. *Given an oracle for F, there is a polynomial time deterministic $(e^{-1}-\varepsilon)$-approximation algorithm for d-SUB for any $\varepsilon > 0$.*

For the problem of maximum coverage with d knapsack constraints, i.e., d-SUB where the objective function is $f = f_{G,\bar{p}}$, for a given bipartite graph $G=(L,R,E)$ and profits $\bar{p}$, there is a deterministic $\left(1-(1-\frac{1}{\alpha})^{\alpha}-\varepsilon\right)$-approximation for the continuous problem using linear programming (see [1]), where α is the maximal degree of any vertex in R. This yields the following result.

THEOREM 8. *There is a polynomial time deterministic $\left(1-(1-\frac{1}{\alpha})^{\alpha}-\varepsilon\right)$-approximation algorithm for maximum coverage with a constant number of knapsack constraints, where α is the maximal degree of any vertex in R.*[6]

3. Discussion. In this paper we established a strong relation between the continuous relaxation of d-SUB and the discrete problem. This relation is nearly optimal and suggests that future research should focus on deriving better approximation ratios for the continuous problem.

The question whether better rounding exists remains open; namely, is it possible to obtain an $\alpha-$approximation algorithm for d-SUB, given an $\alpha < 1$ approximation algorithm for the continuous problem? And more specifically, is there a polynomial time $(1-e^{-1})-$approximation for d-SUB with monotone objective function?

Finally, the running times of our algorithms are exponential in $1/\varepsilon$, thus rendering them impractical. Yet, the hardness results for d-dimensional Knapsack (see, e.g., [15, 21, 18]), a special case of d-SUB, hint that significant improvements over these running times may be impossible.

Appendix A: Basic properties of submodular functions. In this section we give some simple properties of submodular functions. Recall that $f: 2^U \to \mathbb{R}$ is a submodular function if $f(S)+f(T) \geq f(S\cup T)+f(T\cap S)$ for any $S,T \subseteq U$. We define $f_T(S) = f(S\cup T) - f(T)$.

LEMMA 8. *Let $f: 2^U \to \mathbb{R}$ be a submodular function with $f(\emptyset) \geq 0$, and let $S = S_1 \cup S_2 \cup \ldots \cup S_k$, where S_i are disjoint sets. Then*

$$f(S) \leq f(S_1) + f(S_2) + \ldots f(S_k).$$

Proof. By induction on k. For $k=2$, since f is a submodular function, we have that

$$f(S_1) + f(S_2) \geq f(S_1 \cup S_2) + f(S_1 \cap S_2) = f(S) + f(\emptyset),$$

and since $f(\emptyset) \geq 0$, we get that $f(S) \leq f(S_1) + f(S_2)$.

For $k > 2$, using the induction hypothesis twice, we have

$$f(S) \leq f(S_1) + f(S_2) + \ldots f(S_{k-2}) + f(S_{k-1} \cup S_k) \leq f(S_1) + f(S_2) + \ldots f(S_k).$$

■

[6] The vertices in R represent the elements in the instance, while the vertices in L represent subsets of the elements.

LEMMA 9. *Let $f: 2^U \to \mathbb{R}_+$ be a submodular function, and let $S, T_1, T_2 \subseteq U$ such that $T_1 \subseteq T_2$ and $S \cap T_2 = \emptyset$. Then, $f_{T_2}(S) \leq f_{T_1}(S)$.*

Proof. Since f is submodular,

$$f(S \cup T_1) + f(T_2) \geq f(S \cup T_1 \cup T_2) + f((S \cup T_1) \cap T_2) = f(S \cup T_2) + f(T_1).$$

Hence, $f_{T_2}(S) \leq f_{T_1}(S)$. ∎

LEMMA 10. *Let $f: 2^U \to \mathbb{R}_+$ be a submodular function, and let $S = S_1 \cup S_2 \cup \ldots \cup S_k$, where S_i are disjoint sets. Then,*

$$f(S) \geq \sum_{i=1}^{k} f_{S \setminus S_i}(S_i).$$

Proof. We note that

$$f(S) = \sum_{i=1}^{k} f_{S_1 \cup \ldots \cup S_{i-1}}(S_i).$$

By Lemma 9, for each $i > 1$, $f_{S_1 \cup \ldots \cup S_{i-1}}(S_i) \geq f_{S \setminus S_i}(S_i)$. Hence,

$$f(S) \geq \sum_{i=1}^{k} f_{S \setminus S_i}(S_i).$$

∎

Appendix B: Proof of Claim 3. We first show that D can be partitioned into $2d\ell$ sets $D_1, \ldots D_{2d\ell}$, such that each of the sets is a feasible solution. W.l.o.g assume that $L_r = 1$ for all $1 \leq r \leq d$,[7] and define the max-cost of element i to be $cm(i) = \max_{1 \leq r \leq d} c_r(i)$. Let $D = \{i_1, \ldots, i_t\}$ be the set of elements in non-decreasing order by their max-costs, i.e., $cm(i_1) \geq cm(i_2) \geq \ldots \geq cm(i_t)$. Now, we set $D_k = \emptyset$ for all $1 \leq k \leq 2\ell$, and for any $1 \leq j \leq t$ we find a set D_k such that $D_k \cup \{i_j\}$ is feasible; we then add i_j to D_k. Assume towards contradiction that, for some i_j, such D_k does not exist. We say that a set D_k is half-full if $c_r(D_k)$ is greater than $1/2$ in some dimension $1 \leq r \leq d$. We distinguish between two cases:

(i) If $cm(i_j) > 1/2$ then, clearly, all the sets D_k are half-full, since none of these sets is empty, and all elements placed in these sets have a dimension in which the cost is greater than $1/2$ (as the elements are ordered by their max-cost).

(ii) If $cm(i_j) \leq 1/2$ then all the sets D_k are half-full, else it would be possible to add i_j to one of them.

Since all the sets D_k are half full, for some $1 \leq r \leq d$ there are at least 2ℓ sets in $D_1, \ldots D_{2d\ell}$ whose costs are greater than $1/2$ in dimension r. Assume these sets are $D_1, \ldots, D_{2\ell}$. Then $\sum_{k=1}^{2\ell} c_r(D_k) > 2\ell \cdot 1/2 = \ell$, and therefore $c_r(D) > \ell$, in contradiction to $R \leq \ell$.

Thus, the process terminates successfully with the elements of D partitioned into $2d\ell$ sets, $D_1, \ldots D_{2d\ell}$, such that each of the sets is a feasible solution. Hence, $f(D_i) \leq \mathcal{O}$, and by Lemma 8, $f(D) \leq f(D_1) + \ldots + f(D_{2d\ell}) \leq 2d\ell f(\mathcal{O})$. ∎

[7] This can be attained by a proper scaling of the element costs.

References

[1] Ageev, A.A., M. Sviridenko. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Combinatorial Optimization* **8**(3) 307–328.

[2] Bansal, N., N. Korula, V. Nagarajan, A. Srinivasan. 2010. On k-column sparse packing programs. *Proceedings of the 14th international conference on Integer Programming and Combinatorial Optimization*. Springer-Verlag, Berlin, Heidelberg, 369–382.

[3] Calinescu, G., C. Chekuri, M. Pál, J. Vondrák. 2007. Maximizing a submodular set function subject to a matroid constraint (extended abstract). *Proceedings of the 12th international conference on Integer Programming and Combinatorial Optimization*. Springer-Verlag, Berlin, Heidelberg, 182–196.

[4] Calinescu, G., C. Chekuri, M. Pál, J. Vondrák. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing* **40**(6) 1740–1766.

[5] Chekuri, C., A. Kumar. 2004. Maximum coverage problem with group budget constraints and applications. *APPROX-RANDOM*. 72–83.

[6] Chekuri, C., J. Vondrák, R. Zenklusen. 2010. Dependent randomized rounding via exchange properties of combinatorial structures. *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. 575–584.

[7] Demaine, E. D., M. Zadimoghaddam. 2010. Scheduling to minimize power consumption using submodular functions. *Proceedings of the 22nd ACM Symposium on Parallelism in Algorithms and Architectures*. 21–29.

[8] Fadaei, S., MohammadAli S., MohammadAmin F. 2011. Maximizing submodular set functions subject to different constraints: Combined algorithms. *CoRR* **abs/1101.2973**.

[9] Feige, U. 1998. A threshold of $\ln n$ for approximating set cover. *J. ACM* **45**(4) 634–652.

[10] Feige, U., V. S. Mirrokni, J. Vondrák. 2007. Maximizing non-monotone submodular functions. *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. 461–471.

[11] Feldman, M., J. Naor, R. Schwartz. 2011. A unified continuous greedy algorithm for submodular maximization. *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. 570–579.

[12] Fujito, T. 2000. Approximation algorithms for submodular set cover with applications. *IEICE Trans. Inf. and Systems* **E83-D(3)**.

[13] Gharan, S. O., J. Vondrák. 2011. Submodular maximization by simulated annealing. *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 1098–1116.

[14] Hartline, J., V. Mirrokni, M. Sundararajan. 2008. Optimal marketing strategies over social networks. *Proceedings of the 17th international conference on World Wide Web*. ACM, 189–198.

[15] Kellerer, H., U. Pferschy, D. Pisinger. 2004. *Knapsack Problems*. 1st ed. Springer.

[16] Khuller, S., A. Moss, J. Naor. 1999. The budgeted maximum coverage problem. *Inf. Process. Lett.* **70**(1) 39–45.

[17] Kulik, A. 2011. *Submodular and Linear Maximization with Knapsack Constraints*. MSc Thesis, Computer Science Department, Technion.

[18] Kulik, A., H. Shachnai. 2010. There is no EPTAS for two-dimensional knapsack. *Inf. Process. Lett.* **110**(16) 707–710.

[19] Kulik, A., H. Shachnai, T. Tamir. 2009. Maximizing submodular set functions subject to multiple linear constraints. *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 545–554.

[20] Lee, J., V. S. Mirrokni, V. Nagarajan, M. Sviridenko. 2009. Non-monotone submodular maximization under matroid and knapsack constraints. *Proceedings of the 41st annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 323–332.

[21] Magazine, Michael J., Maw-Sheng Chern. 1984. A note on approximation schemes for multidimensional knapsack problems. *Mathematics of Operations Research* **9**(2) 244–247.

[22] Nemhauser, G., L. Wolsey. 1978. Best algorithms for approximating the maximum of submodular set function. *Mathematics of Operations Research* **3**(3) 177–188.

[23] Nemhauser, G., L. Wolsey, M. Fisher. 1978. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming* **14** 265–294.

[24] Schrijver, A. 2003. *Combinatorial Optimization - polyhedra and efficiency*. Springer Verlag - Berlin Heidelberg.

[25] Sviridenko, M. 2004. A note on maximizing a submodular set function subject to knapsack constraint. *Operations Research Letters* **32** 41–43.

[26] Vondrák, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. *Proceedings of the 40th annual ACM symposium on Theory of computing*. 67–74.

[27] Vondrák, J., C. Chekuri, R. Zenklusen. 2011. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *Proceedings of the 43rd annual ACM symposium on Theory of computing*. New York, NY, USA, 783–792.