Operations Research Letters 39 (2011) 447–451

Contents lists available at SciVerse ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Maximizing non-monotone submodular set functions subject to different constraints: Combined algorithms

Salman Fadaei [a], MohammadAmin Fazli [b,*], MohammadAli Safari [b]

[a] Faculty of Engineering, University of Tehran, Tehran, Iran

[b] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 20 February 2011
Accepted 23 September 2011
Available online 8 October 2011

Keywords:
Non-monotone submodular set functions
Approximation algorithms
Continuous greedy process
Matroid
Knapsack
Cardinality

ABSTRACT

We study the problem of maximizing constrained non-monotone submodular functions and provide approximation algorithms that improve existing algorithms in terms of either the approximation factor or simplicity. Different constraints that we study are exact cardinality and multiple knapsack constraints for which we achieve $(0.25 - \epsilon)$-factor algorithms.

We also show, as our main contribution, how to use the continuous greedy process for non-monotone functions and, as a result, obtain a 0.13-factor approximation algorithm for maximization over any solvable down-monotone polytope.

1. Introduction

Submodularity is the discrete analogous of convexity. Submodular set functions naturally arise in several different important problems including cuts in graphs [16,13], rank functions of matroids [8], and set covering problems [9]. The problem of maximizing a submodular function is NP-hard as it generalizes many important problems such as Maximum Cut [10], Maximum Facility Location [2,1], and the Quadratic Cost Partition Problem with non-negative edge weights [14].

Definition 1. A function $f : 2^X \to \mathbb{R}_+$ is called submodular if and only if $\forall A, B \subseteq X$, $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$. An alternative definition is that the marginal values of items should be non-increasing, i.e., $\forall A, B \subseteq X$, $A \subseteq B \subseteq X$ and $x \in X \setminus B$, $f_A(x) \geq f_B(x)$, where $f_A(x) = f(A \cup \{x\}) - f(A)$; $f_A(x)$ is called the marginal value of x with respect to A.

The Submodular Maximization Problem is a pair (f, Δ), where f is a submodular function and Δ is the search domain. Our aim is to find a set $A^* \in \Delta$ whose value, $f(A^*)$, is maximum. Our focus is on non-monotone submodular functions, i.e., we do not require that $f(A) \leq f(B)$ for $A \subseteq B \subseteq X$.

Definition 2. A packing polytope is a polytope $P \subseteq [0, 1]^X$ that is down-monotone: If $x, y \in [0, 1]^X$ with $x \preceq y$ and $y \in P$, then $x \in P$. A polytope P is solvable if we can maximize linear functions over P in polynomial time [21].

A packing polytope constraint binds the search domain (Δ) to a packing polytope.

Definition 3. For a ground set X, k weight vectors $\{w^i\}_{i=1}^k$, and k knapsack capacities $\{C_i\}_{i=1}^k$ are given. A set $V \subseteq X$ is called packable if $\sum_{j \in V} w_j^i \leq C_i$, for $i = 1, \ldots, k$.

The multiple knapsack constraint forces us to bind the search domain to packable subsets of X. In the exact cardinality constraint, we have $\Delta = \{S \subseteq X : |S| = k\}$.

Background. The problem of maximizing non-monotone submodular functions, with or without some constraints, has been extensively studied in the literature. In [11], a 0.4-factor approximation algorithm was developed for maximizing unconstrained (non-negative, non-monotone) submodular functions. The approximation factor was very recently improved to 0.41 by Oveis Gharan and Vondrák [20].

For the constrained variants, Lee et al. [19], Vondrák [22], and Gupta et al. [15] provide the best approximation algorithms. Lee et al. [19] developed a 0.2-approximation for the problem subject to a constant number of knapsack constraints, followed by a 0.25-approximation for the cardinality constraint and a 0.15-approximation for the exact cardinality constraint. The latter two approximation factors were later improved by Vondrák [22] to

* Corresponding author.
E-mail addresses: salman.fadaei@gmail.com (S. Fadaei), fazli@ce.sharif.edu, mohammadamin.fazli@gmail.com (M. Fazli), msafari@sharif.edu (M. Safari).

0167-6377/$ – see front matter
doi:10.1016/j.orl.2011.10.002

Table 1
Comparison of our results with the existing ones.

Constraint	[19]	[22]	[15]	Our result	Claim
Exact cardinality	0.15	0.25	0.17	0.25	Simpler
k-Knapsacks	0.2	–	–	0.25	Better ratio
Packing polytope	–	–	–	0.13	New ratio

0.309 and 0.25, respectively. As a new way of tackling these problems, Gupta et al. [15] provide greedy algorithms that achieve the approximation factor of 0.17 for a knapsack constraint. Greedy algorithms are more common for maximizing monotone submodular functions.

In a recent work, Vondrák [23] and Calinescu et al. [3] used the idea of multilinear extension of submodular functions and achieved optimal approximation algorithms for the problem of maximizing a monotone submodular function subject to a matroid.

1.1. Our results

We consider the problem subject to different constraints. Our results are summarized in Table 1 and are compared with existing results. We obtain simple algorithms for the exact cardinality constraint, multiple knapsack constraints. Moreover, we use the continuous greedy process for non-monotone functions to obtain a 0.13-factor approximation algorithm for maximization over any solvable down-monotone polytope. This implies a 0.13-approximation for several discrete problems, such as maximizing a non-negative submodular function subject to a matroid constraint and/or multiple knapsack constraints.

1.2. Preliminaries

In this section, we introduce the concepts and terms that we often use throughout this paper.

Multilinear extension. For a submodular function $f: 2^X \to \mathbb{R}_+$, the multilinear extension of f is defined as follows [4]: $F: [0,1]^X \to \mathbb{R}_+$ and

$$F(x) = \mathbf{E}[f(x)] = \sum_{S\subseteq X} f(S) \prod_{i\in S} x_i \prod_{i\in X\setminus S} (1-x_i).$$

This concept is frequently used in recent works [4,3,17,19,22]. The multilinear extension of every submodular function is a smooth submodular function [3]. The gradient of F is defined as $\nabla F = (\frac{\partial F}{\partial x_1}, \dots, \frac{\partial F}{\partial x_n})$.

Matroid. A matroid is a pair $\mathcal{M} = (X, \mathcal{I})$ where $\mathcal{I} \subseteq 2^X$ and

- $\forall B \in \mathcal{I}, A \subset B \Rightarrow A \in \mathcal{I}$.
- $\forall A, B \in \mathcal{I}, |A| < |B| \Rightarrow \exists x \in B \setminus A; A \cup \{x\} \in \mathcal{I}$.

Matroid polytopes. A matroid polytope is a solvable packing polytope with special properties. Given a matroid $\mathcal{M} = (X, \mathcal{I})$, we define the matroid polytope as

$$P(\mathcal{M}) = \left\{ x \geq 0 : \forall S \subseteq X; \sum_{j\in S} x_j \leq r_{\mathcal{M}}(S) \right\}$$

where $r_{\mathcal{M}}(S) = \max\{|I| : I \subseteq S; I \in \mathcal{I}\}$ is the rank function of matroid $\mathcal{M}$. This definition shows that the matroid polytope is a packing polytope.

Randomized pipage rounding. Given a matroid $\mathcal{M} = (X, \mathcal{I})$, the randomized pipage rounding converts a fractional point in the matroid polytope, $y \in P(\mathcal{M})$ into a random set $B \in \mathcal{I}$ such that $\mathbf{E}[f(B)] \geq F(y)$, where F is the multilinear extension of the submodular function f [4,3,22].

1.3. Recent developments

There has been some very recent relevant works independent and concurrent to our work. Kulik et al. give an $(0.25 - \epsilon)$-approximation algorithm for maximizing non-monotone submodular functions subject to multiple knapsacks [18]. Chekuri et al. [6] show that, by using a fractional local search, a 0.325-approximation could be achieved for maximizing non-monotone submodular functions subject to any solvable packing polytope. However, our 0.13-factor approximation algorithm is still of independent interest in that it uses the continuous greedy approach rather than local search and, thus, it would be more efficient in practice.

2. Exact cardinality constraint

In this section, we propose very simple algorithm for the exact cardinality constraint problem whose approximation factor matches the best existing one, yet it is much simpler and easy to implement. Our algorithm is a simple combination of existing local search or greedy based algorithms. Our main tool is the following useful lemma from [15].

Lemma 1 (*[15]*). *Given sets $C, S_1 \subseteq X$, let $C' = C \setminus S_1$ and $S_2 \subseteq X \setminus S_1$. Then, $f(S_1 \cup C) + f(S_1 \cap C) + f(S_2 \cup C') \geq f(C)$.*

Let k be the right-hand side of the cardinality constraint.

Theorem 1. *There is a 0.25-factor approximation algorithm for maximizing a non-monotone submodular function subject to an exact cardinality constraint.*

Proof. First, we use the local search algorithm of [19] and compute a set S_1 whose size is k and $2f(S_1) \geq f(S_1 \cup C) + f(S_1 \cap C)$ for any C with $|C| = |S_1| = k$. Next, we use the greedy algorithm of [15] and compute a set $S_2 \subseteq X \setminus S_1$ of size k such that for any C' with $|C'| \leq k$, $f(S_2) \geq 0.5 f(S_2 \cup C')$. Let C be the true optimum and $C' = C \setminus S_1$. Therefore,

$$\begin{aligned} 2f(S_1) + 2f(S_2) &\geq f(S_1 \cup C) + f(S_1 \cap C) + f(S_2 \cup C') \\ &\geq f(C) = OPT. \end{aligned}$$

Thus, the better of S_1 and S_2 gives an approximation factor 0.25.

Here, we have assumed that $k \leq \frac{|X|}{2}$. If not, we can alternatively solve the problem for the derived submodular function $g(S) = f(X \setminus S)$ subject to cardinality constraint $k' = |X| - k$. □

The approximation factor 0.25 matches that of [22], though our algorithm is simpler and straightforward to implement.

3. Multiple knapsack constraints

Lee et al. [19] propose a 0.2-factor approximation algorithm for the problem. They basically divide the elements into two sets of heavy and light objects and then solve the problem separately for each set and return the maximum of the two solutions.

We improve their result by considering both heavy and light elements together. Our algorithm finds a fractional solution and then integrates it by using independent rounding. We use some of the properties of the independent rounding; For the sake of completeness, we mention it before presenting the main algorithm.

Let $x = (x_1, \dots, x_n)$ be a fractional solution and $(X_1, \dots, X_n) \in \{0,1\}^n$ be an integral solution obtained from x by randomized independent rounding. We observe that $\mathbf{E}[X_i] = x_i$ and for any subset T, $\mathbf{E}[\prod_{i\in T} X_i] = \prod_{i\in T} x_i$, and $\mathbf{E}[\prod_{i\in T}(1 - X_i)] = \prod_{i\in T}(1 - x_i)$. Considering these properties, as in [7] (Theorem II.1) and [12] (Theorem 3.1), we obtain the following Chernoff-type concentration bound for linear functions of $X_1, \dots, X_n$.

Lemma 2. *Let $a_i \in [0, 1]$ and $X = \sum a_i X_i$ where $(X_1, \ldots, X_n)$ are obtained by randomized independent rounding from a starting point $(x_1, \ldots, x_n)$. If $\delta \in [0, 1]$, and $\mu \geq \mathbf{E}[X] = \sum a_i x_i$, then $Pr[X \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}$.*

Moreover, the independent rounding gives a concentration inequality for submodular functions as stated in [24].

Lemma 3 (*[24]*)**.** *If $Z = f(X_1, \ldots, X_n)$ where $X_i \in \{0, 1\}$ are independently random and f is non-negative submodular with marginal values in $[-1, 1]$, then for any $\delta > 0$,*

- $Pr[Z \geq (1+\delta)\mathbf{E}[Z]] \leq e^{-\delta^2 \mathbf{E}[Z]/(4+5\delta/3)}$.
- $Pr[Z \leq (1-\delta)\mathbf{E}[Z]] \leq e^{-\delta^2 \mathbf{E}[Z]/4}$.

We use the lower bound tail for our purpose. Our algorithm (Algorithm 1 below) is based on the algorithm of [7] for maximizing monotone submodular functions subject to one matroid and multiple knapsack constraints. We have made some modifications to use it for non-monotone functions.

Input: Elements weights $\{c_{i,j}\}$, parameter $0 < \epsilon < 1/(4k^2)$, and a non-monotone submodular function f

$D \leftarrow \emptyset$.

foreach *subset A of at most $1/\epsilon^4$ elements* **do**

0. Set $D \leftarrow A$ if $f(A) > f(D)$;
1. Redefine $C_j = 1 - \sum_{i \in A} c_{ij}$ for $1 \leq j \leq k$.;
2. Let B be the set of items $i \notin A$ such that either $f_A(i) > \epsilon^4 f(A)$ or $c_{ij} > k\epsilon^3 Cj$ for some j;
3. Let x^* be the fractional solution of the following problem:

$$\max\{G(x) : x \in P(\mathcal{M});\quad \forall j \sum c_{ij} x_i \leq (1-\epsilon)C_j\} \tag{1}$$

where $\mathcal{M}$ is the free matroid over the ground set $X' = X \setminus (A \cup B)$, and $G(x)$ is the multilinear extension of $g(S) = f_A(S), S \subseteq X'$.;

4. Let R be the result of the independent rounding applied to x^* with respect to the matroid polytope $P(\mathcal{M})$. Set $D \leftarrow A \cup R$ if $f(A \cup R) > f(D)$;

end

Return D.

Algorithm 1: Non-Monotone Maximization Subject to Multiple Knapsacks

The following theorem shows how good our algorithm is.

Theorem 2. *Algorithm 1 returns a solution of expected value at least $(0.25 - 2\epsilon)OPT$.*

Proof. The proof follows the line of proofs of [7] with major changes to adapt it for non-monotone case. Let O be the optimal solution and $OPT = f(O)$. Assume $|O| \geq \frac{1}{\epsilon^4}$; otherwise, our algorithm finds the optimal solution in Line 0. Sort the elements of O by their decreasing marginal values, and let $A \subseteq O$ be the first $\frac{1}{\epsilon^4}$ elements. Consider the iteration in which this set A is chosen. Since A has $\frac{1}{\epsilon^4}$ elements, the marginal value of its last element and every element not in A is at most $\epsilon^4 f(A) \leq \epsilon^4 OPT$. So, throwing away elements whose marginal value is bigger than $\epsilon^4 f(A)$ does not hurt. We also throw away the set $B \subseteq N \setminus A$ of items whose size in some knapsack is more than $k\epsilon^3 Cj$. In $O \setminus A$, there can be at most $1/(k\epsilon^3)$ such items for each knapsack, i.e., $1/\epsilon^3$ items in total. Since their marginal values with respect to A are bounded by $\epsilon^4 OPT$, these items together have value $g(O \backslash B) = f_A(O \backslash B) \leq \epsilon OPT$. The set $O' = O \setminus (A \cup B)$ is still a feasible set for the maximization problem, and by submodularity has value:

$$\begin{aligned} g(O') &= g((O \setminus A) \setminus (O \setminus B)) \\ &\geq g(O \setminus A) - g(O \cap B) \geq OPT - f(A) - \epsilon OPT. \end{aligned}$$

The indicator vector $(1-\epsilon)1_{O'}$ is a feasible solution for Problem (1) (specified at step 3 of Algorithm 1). By the concavity of $G(x)$ along the line from the origin to $1_{O'}$, we have $G((1-\epsilon)1_{O'}) \geq (1-\epsilon)g(O') \geq (1-2\epsilon)OPT - f(A)$. By Theorem 4 of [19] we can compute in polynomial time a fractional solution x^* with value:

$$G(x^*) \geq \frac{1}{4} G((1-\epsilon)1_{O'}) \geq \left(\frac{1}{4} - 2\epsilon\right) OPT - f(A).$$

Finally, we apply independent rounding to x^* and call the resulting set R. By the construction of independent rounding, we have $\mathbf{E}[g(R)] = G(x^*)$. However, R might violate some of the knapsack constraints. Consider a fixed knapsack constraint, $\sum_{i \in S} c_{ij} \leq C_j$. Our fractional solution x^* satisfies $\sum c_{ij} x_i^* \leq (1-\epsilon)C_j$. Also, we know that all sizes in the reduced instance are bounded by $c_{ij} \leq k\epsilon^3 C_j$. By scaling, $c'_{ij} = c_{ij}/(k\epsilon^3 C_j)$, we use Lemma 2 with $\mu = (1-\epsilon)/(k\epsilon^3)$:

$$\begin{aligned} Pr\left[\sum_{i \in R} c_{ij} > C_j\right] &\leq Pr\left[\sum_{i \in R} c'_{ij} > (1+\epsilon)\mu\right] \\ &\leq e^{-\mu\epsilon^2/3} < e^{-1/4k\epsilon}. \end{aligned}$$

Finally, we show that $g(R)$ has a high value with respect to $G(x^*)$. In the reduced instance, all items have value $g(i) \leq \epsilon^4 OPT$. Let $\mu = G(x^*)/(\epsilon^4 OPT)$. Using Lemma 3, we get

$$\begin{aligned} Pr[g(R) \leq (1-\delta)G(x^*)] &= Pr[g(R)/\epsilon^4 OPT \leq (1-\delta)\mu] \\ &\leq e^{-\delta^2\mu/4} = e^{-\delta^2 G(x^*)/(4\epsilon^4 OPT)}. \end{aligned}$$

By setting $\delta = \frac{OPT}{G(x^*)}\epsilon$, we obtain

$$Pr[g(R) \leq G(x^*) - \epsilon OPT] \leq e^{-OPT/(4\epsilon^2 G(x^*))} \leq e^{-1/4\epsilon^2}.$$

Therefore, $Pr[g(R) \leq G(x^*) - \epsilon OPT$ or $\exists j : \sum_{j \in R} c_{ij} > C_j] \leq e^{-1/4\epsilon^2} + ke^{-1/4k\epsilon}$. For $\epsilon \leq 1/(4k^2)$ this probability is at most $e^{-4k^4} + ke^{-k} < 1$. Finally, we have a feasible solution of expected value $\mathbf{E}[f(R)] = f(A) + \mathbf{E}[g(R)] = f(A) + G(x^*) \geq (\frac{1}{4} - 2\epsilon)OPT$. □

4. Packing polytope constraint

In this section, we adapt the continuous greedy process for non-monotone submodular functions and propose an algorithm for solving the optimization problems subject to a packing polytope constraint. As an application of the technique, we then consider the problem of submodular maximization subject to both one matroid and multiple knapsacks constraints. Finally, we briefly show how to replace this continuous process with a polynomial time discrete process without suffering much.

4.1. Continuous greedy process for non-monotone functions

Similar to [3], the greedy process starts with $y(0) = \mathbf{0}$ and increases over a unit time interval as follows:

$$\frac{dy}{dt} = v_{\max}(y),$$

where $v_{\max}(y) = \operatorname{argmax}_{v \in P}(v.\nabla F(y))$. For the case where F is a non-monotone smooth submodular function, we have the following lemma.

Lemma 4. *$y(1) \in P$ and $F(y(1)) \geq (1 - e^{-1})(F(x \vee y(1)) - F_{DMAX})$, where $x \in P$, and $F_{DMAX} = \max_{0 \leq t \leq 1} F(y(1) - y(t))$.*

Proof. The proof is essentially similar to that of [3] with some modifications to adapt it for non-monotone functions. First, the trajectory for $t \in [0, 1]$ is contained in P since

$$y(t) = \int_0^t v_{\max}(y(\tau))d\tau$$

is a convex linear combination of vectors in P. To prove the approximation guarantee, fix a point y. Consider a direction $v^* = (x \vee y) - y = (x - y) \vee 0$. This is a non-negative vector; since $v^* \leq x \in P$ and P is down-monotone, we also have $v^* \in P$. Consider the ray of direction v^* starting at y, and the function $F(y + \xi v^*)$, $\xi \geq 0$. The directional derivative of F along this ray is $\frac{dF}{d\xi} = v^*.\nabla F$. Since F is smooth submodular (that means, each entry $\frac{\partial F}{\partial y_j}$ of ∇F is non-increasing with respect to y_j) and v^* is nonnegative, $\frac{dF}{d\xi}$ is non-increasing too and $F(y + \xi v^*)$ is concave in ξ. By concavity, we have

$$F(y(1) + v^*) - F(y(t)) \leq F(y(t) + v^*) - F(y(t)) + F(y(1) - y(t)) \leq v^*.\nabla F(y(t)) + F_{DMAX}.$$

Since $v^* \in P$ and $v_{\max}(y) \in P$ maximizes $v.\nabla F(y)$ over all vectors $v \in P$, we get

$$v_{\max}(y).\nabla F(y) \geq v^*.\nabla F(y) \geq F(y(1) + v^*) - F_{DMAX} - F(y). \quad (2)$$

We now get back to the continuous process and analyze $F(y(t))$. Using the chain rule and the inequality (2), we get

$$\frac{dF}{dt} = \sum_j \frac{\partial F}{\partial y_j}\frac{dy_j}{dt} = v_{\max}(y(t)).\nabla F(y(t)) \geq F(x \vee y(1)) - F_{DMAX} - F(y(t)).$$

Thus, $F(y(t))$ dominates the solution of the differential equation

$$\frac{d\phi}{dt} = F(x \vee y(1)) - F_{DMAX} - \phi(t)$$

which means $\phi(t) = (1 - e^{-t})(F(x \vee y(1)) - F_{DMAX})$. Therefore, $F(y(t)) \geq (1 - e^{-t})(F(x \vee y(1)) - F_{DMAX})$. □

4.2. Extending smooth local search

As our final tool for obtaining the main algorithm of this section, we propose an algorithm for the following problem: Let f be a submodular function and F be its multilinear extension. Let $u_i \in [0, 1]$, $1 \leq i \leq n$, be a set of upper bound variables and $\mathcal{U} := \{0 \leq y_i \leq u_i \ \forall i \in X\}$. We want to maximize F over the region $\mathcal{U}$:

$$\max\{F(y) : y \in \mathcal{U}\}$$

For this, we extend the 0.4-approximation algorithm (Smooth Local Search or SLS) of [11] as follows. We call our algorithm FMV_Y.

We define a discrete set ζ of values in [0, 1], where $\zeta = \{p.\delta : 0 \leq p \leq 1/\delta\}, \delta = \frac{1}{8n^4}$ and p is integer. The algorithm returns a vector whose values come from the discrete set ζ. We show that such a discretization does not substantially harm our solution, yet it reduces the running time.

Let U be a multiset containing $s_i = \lfloor \frac{1}{\delta} u_i \rfloor$ copies of each element $i \in X$. We define a set function $g : 2^U \rightarrow \mathbb{R}_+$ with $g(T) = F(\ldots, \frac{|T_i|}{s_i}, \ldots)$, where $T \subseteq U$ and T_i contains all copies of i in T. The function g has been previously introduced in [19] and proved to be submodular. Let B be the solution of running the SLS algorithms for maximizing g and y be its corresponding vector.

Based on [11], we have $g(B) \geq 0.4g(A)$, $\forall A \in U$; thus

$$F(y) \geq 0.4F(z), \quad \forall z \in \mathcal{U} \cap \zeta^n. \quad (3)$$

and we can prove the following claim.

Claim 1. *For any $x \in \mathcal{U}$, $2.5F(y) \geq F(x) - \frac{f_{\max}}{4n^2}$, where $f_{\max} = \max\{f(i) : i \in X\}$.*

Proof. Let z be the point in $\zeta^n \cap \mathcal{U}$ that minimizes $\sum_{i=1}^n (x_i - z_i)$. By Claim 3 of [19], $F(z) \geq F(x) - \frac{f_{\max}}{4n^2}$. Using the inequality (3), we get $F(y) \geq 0.4(F(x) - \frac{f_{\max}}{4n^2})$. This completes the proof. □

4.3. The algorithm

We now present our algorithm for maximizing a smooth submodular function over a solvable packing polytope:

Input: A packing polytope P and a smooth submodular function F
1. $y_1 \longleftarrow$ The result of running the continuous greedy process.
2. $y'_1 \longleftarrow \text{argmax}_{0 \leq t \leq 1} F(y_1 - y(t))$.
3. $y_{1\max} \longleftarrow$ The result of running FMV_Y with the upper bound y_1.
4. $y_2 \longleftarrow$ The result of running the greedy process over the new polytope P' which is built by adding constraints $y_i \leq 1 - y_{1_i}$ for any $1 \leq i \leq n$ to P. Note that P' is a down-monotone polytope.
5. $y'_2 \longleftarrow \text{argmax}_{0 \leq t \leq 1} F(y_2 - y(t))$.
6. Return $\text{argmax}(F(y_1), F(y_2), F(y_{1\max}), F(y'_1), F(y'_2))$.

Algorithm 2: Continuous greedy process for non-monotone functions

Theorem 3. *The above algorithm is a $\frac{2e-2}{13e-9}$-approximation algorithm for the problem of maximizing a smooth submodular function F over a solvable packing polytope P.*

Proof. Suppose $x^* \in P$ is the optimum and $F(x^*) = OPT$. By Lemma 4, $F(y_1) \geq (1 - e^{-1})(F(x^* \vee y_1) - F(y'_1))$. We also have $F(y_2) \geq (1 - e^{-1})(F(x' \vee y_2) - F(y'_2))$, where $x' = x^* - (x^* \wedge y_1)$. Note that $x' \in P'$. By Claim 1, we also have $F(y_{1\max}) \geq 0.4(F(x^* \wedge y_1) - \frac{f_{\max}}{4n^2})$ as $x^* \wedge y_1 \preceq y_1$.

By adding up the above inequalities, we get

$$\frac{e}{e-1}(F(y_1) + F(y_2)) + F(y'_1) + F(y'_2) + 2.5F(y_{1\max}) \geq F(x^* \vee y_1) + F(x' \vee y_2) + F(x^* \wedge y_1) - \frac{f_{\max}}{4n^2} \geq F(x^*) - \frac{f_{\max}}{4n^2} = OPT - \frac{f_{\max}}{4n^2}.$$

Therefore, the approximation factor of the algorithm is at least $\frac{2e-2}{13e-9} OPT$. □

Both one matroid and multiple knapsacks. As a direct result of the above theorem, we propose the first approximation algorithm for maximizing a submodular function subject to both one matroid and multiple knapsacks. This problem was solved (approximately) in [5] for monotone submodular functions.

Theorem 4. *There exists an algorithm with expected value of at least $(\frac{2e-2}{13e-9} - 3\epsilon)OPT$ for the problem of maximizing any non-monotone submodular function subject to one matroid and multiple knapsacks.*

Proof. The intersection of the polytopes corresponding to one matroid and multiple knapsacks is still a solvable packing polytope. Thus, we can achieve a fractional solution by using Algorithm 2 together with the enumeration phase (as in Algorithm 1), and then we can round the fractional solution into the integral one using randomized pipage rounding.

Our algorithm is very similar to that of [5] with some modifications to adapt it for non-monotone functions. As the two algorithms are similar, we only highlight the modifications to our algorithm.

The algorithm in [5] is for maximizing monotone submodular functions subject to one matroid and multiple knapsacks and uses partial enumeration. At each iteration, after getting rid of all items of large value or size, it defines an optimization problem with scaled down constraints. Since the objective function is monotone, the reduced problem at each iteration is solved using continuous greedy algorithm to find a fractional solution within a factor $1-1/e$ of the optimal.

For our case, we cannot use the continuous greedy algorithm as our function is not monotone. Instead, we use Algorithm 2 to solve the reduced problem and achieve a fractional solution with approximation factor $\frac{2e-2}{13e-9}$. The final step of the two algorithms are identical. At each iteration, we apply randomized pipage rounding to the fractional solution with respect to contracted matroid of the that iteration. The result is the set with maximum objective functions over all iterations.

Our analysis is similar to that of [5] except that our approximation factor for the reduced problem (at each iteration) is $\frac{2e-2}{13e-9}$ as opposed to $1-1/e$ of [5]. So, the same analysis works with the two approximation factors exchanged.

Note that, because of considerations in the design of the algorithm, randomized pipage rounding does not violate, with high probability, the capacity constraints on knapsacks and, therefore, our solution is a feasible one with constant probability. We remark that the argument for the concentration bound in [5] is applicable to our analysis, as well. □

4.4. Discretizing continuous process

In order to obtain a polynomial time, we discretize the continuous greedy process for non-monotone functions and show that by taking small enough time steps, this process only introduces a small error that is negligible and the solution to the differential inequality does not significantly change.

Let $\delta = \frac{1}{n^2}$, and $\zeta = \{p.\delta : 0 \leq p \leq 1/\delta\}$ be a set of discrete values. We set the unit time interval equal to δ in Algorithm 2, and change lines 2 and 5 of it as follows.

2 $y_1' \longleftarrow \text{argmax} F(y_1 - y(t)), \quad \forall t \in \zeta$
5 $y_2' \longleftarrow \text{argmax} F(y_2 - y(t)), \quad \forall t \in \zeta$

and obtain the following lemma which is weaker (but not very different) than Lemma 4.

Lemma 5. *$y(1), y_1' \in P$ and $F(y(1)) \geq (1-e^{-1})(F(x \vee y(1)) - F(y_1')) - o(1)OPT$, where $x \in P$, where P is any solvable packing polytope.*

Acknowledgment

The authors are grateful to Vahab Mirrokni for his help in all the steps of the preparation of this paper. The first author is also thankful to Ali Moeini (his M.Sc. advisor), Dara Moazzami, and Jasem Fadaei for their help and advice. The authors also would like to acknowledge the anonymous referees for their useful comments and suggestions.

References

[1] A.A. Ageev, M.I. Sviridenko, An 0.828-approximation algorithm for the uncapacitated facility location problem, Discrete Applied Mathematics 93 (1999) 149–156.
[2] S. Benati, An improved Branch & Bound method for the uncapacitated competitive location problem, Annals of Operations Research 122 (1) (2003) 43–58.
[3] G. Calinescu, C. Chekuri, M. Pál, J. Vondrák, Maximizing a submodular set function subject to a matroid constraint. SIAM Journal on Computing (in press).
[4] G. Calinescu, C. Chekuri, M. Pál, J. Vondrák, Maximizing a monotone submodular function under a matroid constraint, in: Proc. of 12th IPCO, 2007, pp. 182–196.
[5] C. Chekuri, J. Vondrák, Randomized pipage rounding for matroid polytopes and applications, 2009. arXiv:0909.4348v1.
[6] C. Chekuri, J. Vondrák, R. Zenklusen, Submodular function maximization via the multilinear relaxation and contention resolution schemes, in: Proceedings of the 43rd annual ACM symposium on Theory of computing, STOC, 2011, pp. 783–792.
[7] C. Chekuri, J. Vondrák, R. Zenklusen, Dependent randomized rounding via exchange properties of combinatorial structures, in: Proc. of IEEE FOCS, 2010. A more detailed version available at arXiv:0909.4348v2.
[8] J. Edmonds, Matroids, submodular functions, and certain polyhedra, Journal of Combinatorial Structures and their Applications (1970) 69–87.
[9] U. Feige, A threshold of ln n for approximating set cover, Journal of the ACM 45 (4) (1998) 634–652.
[10] U. Feige, M.X. Goemans, Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT, in: Proceedings of the Third Israel Symposium on Theory of Computing and Systems, vol. 388, 1995.
[11] U. Feige, V.S. Mirrokni, J. Vondrák, Maximizing non-monotone submodular functions, in: FOCS, 2007.
[12] R. Gandhi, S. Khuller, S. Parthasarathy, A. Srinivasan, Dependent rounding and its applications to approximation algorithms, Journal of the ACM 53 (2006) 324–360.
[13] M. Goemans, P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, Journal of the ACM (JACM) 42 (4) (1994) 1115–1145.
[14] B. Goldengorin, D. Ghosh, The multilevel search algorithm for the maximization of submodular functions applied to the quadratic cost partition problem, Journal of Global Optimization 32 (1) (2005) 65–82.
[15] A. Gupta, A. Roth, G. Schoenebeck, K. Talwar, Constrained Non-Monotone Submodular Maximization: Offline and Secretary Algorithms, 2010.
[16] S. Iwata, L. Fleischer, S. Fujishige, A combinatorial strongly polynomial algorithm for minimizing submodular functions, Journal of the ACM (JACM) 48 (4) (2001) 777.
[17] A. Kulik, H. Shachnai, T. Tamir, Maximizing submodular set functions subject to multiple linear constraints, in: Proc. of ACM-SIAM SODA, 2009 pp. 545–554.
[18] A. Kulik, H. Shachnai, T. Tamir, Approximations for Monotone and Non-monotone Submodular Maximization with Knapsack Constraints, 2011. arXiv:1101.2940v1.
[19] J. Lee, V. Mirrokni, V. Nagarajan, M. Sviridenko, Maximizing non-monotone submodular functions under matroid and knapsack constraints, in: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC, 2009.
[20] S. Oveis Gharan, J. Vondrák, Submodular maximization by simulated annealing, in: Proc. of the 22nd Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, SODA, 2011.
[21] A. Schrijver, Combinatorial Optimization — Polyhedra and Efficiency, Springer, 2003.
[22] J. Vondrák, Symmetry and approximability of submodular maximization problems, in: Proc. of IEEE FOCS, 2009, pp. 251–270.
[23] J. Vondrák, Optimal approximation for the submodular welfare problem in the value oracle model, in: STOC, 2008, pp. 67–74.
[24] J. Vondrák, A note on concentration of submodular functions. Manuscript, 2010.