Journal of Combinatorial Optimization (2022) 43:1655–1670
https://doi.org/10.1007/s10878-021-00717-1

Fast algorithms for maximizing monotone nonsubmodular functions

Bin Liu[1] · Miaomiao Hu[1]

Accepted: 22 February 2021 / Published online: 11 March 2021

Abstract
In recent years, with the more and more researchers studying the problem of maximizing monotone (nonsubmodular) objective functions, the approximation algorithms for this problem have gotten much progress by using some interesting techniques. In this paper, we develop the approximation algorithms for maximizing a monotone function f with generic submodularity ratio γ subject to certain constraints. Our first result is a simple algorithm that gives a $(1-e^{-\gamma}-\epsilon)$-approximation for a cardinality constraint using $O(\frac{n}{\epsilon}log\frac{n}{\epsilon})$ queries to the function value oracle. The second result is a new variant of the continuous greedy algorithm for a matroid constraint. We combine the variant of continuous greedy method with the contention resolution schemes to find a solution with approximation ratio $(\gamma^2(1-\frac{1}{e})^2-O(\epsilon))$, and the algorithm makes $O(rn\epsilon^{-4}log^2\frac{n}{\epsilon})$ queries to the function value oracle.

Keywords Nonsubmodular Function · Cardinality constraint · Matroid constraint · Approximation algorithm

1 Introduction

Given a ground set $N=\{1,2,...,n\}$, we say a set function $f:2^N\to R$ is *nonnegative* if $f(S)\geq 0$ for any $S\subseteq N$. The function f is *monotone* if $f(S)\leq f(T)$ whenever $S\subseteq T$, and f is *normalized* if $f(\emptyset)=0$. What's more, the function f is called

This work was supported in part by the National Natural Science Foundation of China (11971447, 11871442), and the Fundamental Research Funds for the Central Universities. A preliminary version of this paper has appeared in the proceeding of conference Algorithmic Aspects in Information and Management, AAIM 2020.

✉ Bin Liu
binliu@ouc.edu.cn

Miaomiao Hu
humiaomiao@stu.ouc.edu.cn

[1] School of Mathematical Sciences, Ocean University of China, Qingdao, China

submodular if $f(S \cup \{j\}) - f(S) \geq f(T \cup \{j\}) - f(T)$ for any $S \subseteq T \subseteq N$, $j \in N \setminus T$. That is, the marginal contribution of any element j to the value of the function $f(S)$ diminishes as the input set S increases. Without loss of generality, we denote by $f_S(T) = f(S \cup T) - f(S)$ the marginal gain of the set T in S for any pair of $S, T \subseteq N$. Specially, for any $S \subseteq N$ and any $j \in N$, we denote by $f_S(j) = f(S \cup \{j\}) - f(S)$ the marginal gain of the singleton set $\{j\}$ in S. In the following we always assume f is nomalized and nonnegative, and there is a value oracle for the objective function f.

In recent years, the theory of submodular optimization has more and more extensive applications, such as machine learning and computer vision. Therefore, researchers paid more attention to the ways to solve these problems. In fact, submodular maximization problems capture well known combinatorial optimization problems such as cut functions of graphs and hypergraphs, rank functions of matroids, entropy, mutual information, coverage functions, budget additive functions and the welfare functions, etc. The reason why the submodular function maximization problems be applied so extensively is that the submodular functions have a good property called diminishing marginal returns. Although the submodular functions have some good properties and important applications, many objective functions in practical problems are not submodular. Therefore, it is necessary to study the nonsubmodular function maximization problem.

The problems of maximizing a submodular or nonsubmodular function under the combinatorial constraints are proved to be generally NP-hard. Thus we turn to find approximation algorithms to solve these problems. Researchers are devoted to find a better approximation ratio and meanwhile with fewer time complexity or fewer queries to the function value oracle.

The common types of submodular maximization problems conclude the submodular function maximization problems under the cardinality constraint or under the matroid constraint. There are lots of results in this area, and the greedy approach is a basic technique for these problems. For example, the greedy algorithm for solving the submodular maximization problem under a cardinality constraint can be divided into three steps successively: firstly start from an empty set; then iteratively add to the current solution set one element that results in the largest marginal gain of the objective function while satisfying the constraints; finally return the solution set of the last step. Also, there are several variations of the greedy approach for different focuses, such as streaming algorithms, parallel algorithms, etc.

1.1 Cardinality constraint

Submodular maximization problems with a cardinality constraint conclude two normally settings, monotone and non-monotone. There are lots of results for submodular maximization under a cardinality constraint. In the following, we will give a brief summary on this constraint.

For the case of monotone, in 1978, Nemhauser et al. (1978) found that the greedy approach is highly effective and it yields a $(1-\frac{1}{e})$-approximation. And Nemhauser and Wolsey (1978) showed that $(1-\frac{1}{e})$ is the best approximation guarantee. Afterwards, Conforti and Cornuéjols (1984) introduced the total curvature $\alpha = 1 - \min_{S, j \notin S} \frac{f_S(j)}{f_\emptyset(j)}$ and proved that the greedy approach can achieve a $\frac{1}{\alpha}(1 - e^{-\alpha})$-approximation under a cardinality constraint. Recently Sviridenko, Sviridenko et al. (2017) showed a new ratio of $(1 - \frac{\alpha}{e})$. Later, researchers paid attention to reduce the query times of the value oracle, and meanwhile it maintains the best approximation guarantee. Badanidiyuru and Vondrák (2014) proposed a simple algorithm which gave a $(1 - \frac{1}{e})$-approximation for a cardinality constraint using $O(\frac{n}{\epsilon} log \frac{n}{\epsilon})$ queries to the function oracle. Mirzasoleiman et al. (2015) proposed a randomized algorithm which can achieve a $(1 - \frac{1}{e})$-approximation guarantee in expectation, and in time linear in the size of the data and independent of the cardinality constraint. In recent years, Mirzasoleiman et al. (2013) proposed a distributed algorithm which can achieve $(1 - \frac{1}{e})$-approximation and the model can be implemented in MapReduce. In addition, due to the rapid increase in dataset sizes, parallel computing has received much interest. Balkanski and Singer (2018) firstly prpoposed the adaptive algorithms to achieve a $(1/3 - \epsilon)$-approximation with adaptivity $O(\log n)$. Later, it is improved to a $(1 - \frac{1}{e} - \epsilon)$-approximation with adaptivity $O(\log n/\epsilon^2)$ Balkanski et al. (2019), Ene and Nguyen (2019). More recently, Breuer et al. (2020) improved the the adaptivity to $O(\log n(\log^2 \frac{\log k}{\epsilon})/\epsilon^2)$ and meanwhile maintaining the optimal approximation ratio.

For the case of non-monotone, Feige et al. (2011) firstly gave a deterministic local-search $\frac{1}{3}$-approximation and a randomized $\frac{2}{5}$-approximation algorithm for maximizing submodular functions in unconstraint setting. Later, Buchbinder and Feldman (2019) presented a randomized algorithm for optimizing the multilinear relaxation whose guarantee is 0.385. Afterwards, Buchbinder and Feldman (2018) applied the standard derandomization techniques to obtain a deterministic algorithm with an approximation ratio of $\frac{1}{e}$. There also have some fast algorithms in order to deal with the big data. Fahrbach et al. (2019) proposed a distributed algorithm for maximizing a non-monotone submodular function subject to a cardinality constraint k that achieves an expected $(0.039 - \epsilon)$-approximation in $O(\frac{\log n}{\epsilon})$ adaptive rounds using $O(\frac{n \log k}{\epsilon^2})$ expected queries to the function oracle. What's more, Gotovos et al. (2015) presented an adaptive random greedy algorithm for the problem, achieving $\frac{1}{e}$-approximation. Later, Balkanski et al. (2018) showed that there is an algorithm whose approximation is arbitrarily close to $\frac{1}{2e}$ in $O(\log^2 n)$ adaptive rounds.

The unprecedented growth of data streams and the limited storage require that extract useful information from massive data. A recent line of work focused on studying streaming algorithms. Badanidiyuru et al. (2014) presented the first one-pass streaming algorithm whose the approximation ratio achieved $(\frac{1}{2} - \epsilon)$ for maximizing a monotone submodular function with the cardinality constraint k. Feldman et al. (2020) proved that $(\frac{1}{2} - \epsilon)$ is the best possible in a streaming set. Later, Kazemi et al. (2019) improved the memory complexity from $O(k \frac{\log k}{\epsilon})$ to $O(\frac{k}{\epsilon})$ with the same approximation ratio and one-pass. For the non-monotone case, Alaluf et al. (2020) proposed the StreamProcess algorithm to achieve 0.2779 approximation ratio with one-pass.

1.2 Matroid constraint

For the submodular maximization problems under the matroid constraint, researchers usually use the continuous greedy algorithms to solve it. Generally speaking, the continuous greedy algorithm returns a fractional solution, and we need some rounding techniques to convert the fractional solution to an integer one. There are many useful rounding methods which conclude the swap rounding technique Badanidiyuru and Vondrák (2014), the pipage rounding technique Ageev and Sviridenko (2004) and the CR schemes rounding techniqueChekuri et al. (2011). When we use these rounding methods to rounding a fractional solution, there also have an approximation ratio between the integer solution and the fractional one. Therefore we should consider the two parts of the approximation ratio of the process when we deal with the submodular function maximization problems under the matroid constraint.

Nemhauser et al. (1978) firstly proved that when f is monotone and submodular, the greedy approach yield a $\frac{1}{2}$-approximation for a matroid constraint. Then Calinescu et al. (2007) enhanced the result to the optimal ratio $1 - \frac{1}{e}$. After introducing the total curvature α, Conforti and Cornuéjols (1984) showed that the greedy appproach can achieve a $\frac{1}{\alpha}(1 - e^{-\alpha})$-approximation under a uniform matroid constraint and $\frac{1}{1+\alpha}$-approximation under a general matroid constraint. Recently, Sviridenko et al. (2017) designed new approximation algorithms for a single matroid constraint which obtained $(1 - \frac{\alpha}{e})$-approximation, and it was the first improvement over the results of Conforti and Cornuéjols (1984) from 1984. Besides, it's worth noting that the continuous greedy algorithm is a basic approach under the matroid constraints. In the previous research, Badanidiyuru and Vondrák (2014) proposed an accelerated continuous greedy algorithm under a matroid constraint using the multilinear extension and swap rounding, and finally they achieved a $(1 - \frac{1}{e} - \epsilon)$-approximation. Vondrák (2010) showed that there is a $\frac{1}{\alpha}(1 - e^{-\alpha})$-approximation algorithm for any monotone submodular function with the curvature α and under a matroid constraint, achieving by the continuous greedy approach and the pipage rounding technique. The pipage rounding and the swap rounding technique are effective methods, but it depends on the submodularity of the objective function. In addition, Chekuri et al. (2011) firstly proposed the contention resolution schemes (CR schemes), another framework for rounding a fractional solution to an integer one, and achieved a $(1 - \frac{1}{e})$-approximation for the rounding process.

In this paper, we deal with the following two optimization problems:

$$\textbf{Problem (1)} : \max\{f(S) : |S| \leq k, S \subseteq N\}$$
$$\textbf{Problem (2)} : \max\{f(S) : S \in \mathcal{I}, S \subseteq N\}$$

where $f : 2^N \to R_+$ is a monotone (nonsubmodular) function, k is a positive integer, and $(N, \mathcal{I})$ is a matroid.

1.3 Submodularity ratio

Though lots of results obtained for maximizing a submodular function subject to different constraints, there are only a few results for nonsubmodular functions.

In order to use known results or methods for maximizing submodular functions, we can define some parameters, such as submodularity ratio, to deal with the maximazition of nonsubmodular functions. Das and David (2011) firstly defined the *submodularity ratio* $\hat{\gamma} = min_{S,T \subseteq N} \frac{\sum_{j \in T \setminus S} f_S(j)}{f_S(T)}$, which describes how close a function is to being submodular. Later, Elenberg et al. (2018) gave the definition of weak submodularity ratio. A monotone non-negative set function $f : 2^N \rightarrow R_+$ is called γ_f-*weakly submodular* for an integer k if $\gamma_f \leq \gamma_k = \min_{S,T \subset N:|T|,|T \setminus S| \leq k} \frac{\sum_{j \in T \setminus S} f_T(j)}{f_T(S)}$. Recently, Kuhnle et al. (2018) and Nong et al. (2020) independently proposed the *generic submodularity ratio* (or *diminishing-return ratio*) which is the largest scalar γ that satisfies $f_S(j) \geq \gamma f_T(j)$, for any $S \subseteq T \subseteq N$. We should note that all these parameters are not equivalent, although each of them measures how close a set function to being submodular.

For a cardinality constraint, Bian et al. (2017) proved that the greedy approach gives a $\frac{1}{\alpha}(1 - e^{-\alpha\hat{\gamma}})$-approximation for maximizing a monotone nonsubmodular function with the curvature α. Then, Kohara et al. (2020) apllied the algorithm to solve the sensor placement problem. Afterwards, Nong et al. (2020) showed that the greedy algorithm can achieve a $(1 - e^{-\gamma})$-approximation for maximizing a strictly monotone nonsubmodular function, and the queries of the function value oracle is $O(kn)$. Recently, Santiago and Yoshida (2020) proved that there existed an efficient randomized greedy algorithm which has an approximation ratio of at least $\gamma_f e^{-\frac{1}{\gamma_f}}$ on expectation for the problem of maximizing a non-monotone γ_f-weakly submodular function. On the data streaming summarization, Wang et al. (2019) designed sieve-streaming algorithm which can achieve a $1 - \frac{1}{2^\gamma - \epsilon}$ approximation ratio for maximizing a monotone nonsubmmodular function. Further more, Li et al. (2020) proposed sieve-streaming^{++} algorithm to reach a $\min\{\frac{(1-\epsilon)\gamma}{2^\gamma}, 1 - \frac{1}{2^\gamma}\}$ approximation ratio for maximizing a monotone non-submmodular function.

For knapsack constraints, Nong et al. (2020) showed that the greedy algorithm can achieve $(1-e^{-\gamma})$ approximation for maximizating a strictly monotone nonsubmodular function under a knapsack constraint. Later, Zhang et al. (2019) proved that the greedy algorithm gives a tight approximation guarantee of $\frac{1}{\alpha}(1 - e^{-\alpha\hat{\gamma}})$ for maximizing a monotone nonsubmodular function with the carvature α under a knapsack constraint. On the data streaming summarization, for maximizing a monotone nonsubmodular function with d-knapsack constraints, Jiang et al. (2019) gave an algorithm whose approximation ratio is, $\min\{\frac{\gamma^2}{d}(\frac{d}{1+d})^\gamma(1 - \beta)(1 - \epsilon), 1 - (\frac{d}{1+d})^\gamma\}$ when $\beta < 0.5$; and $\min\{\frac{\gamma^3}{2d}(\frac{d}{1+d})^\gamma(1 - \epsilon), 1 - (\frac{d}{1+d})^\gamma\}$ when $\beta \geq 0.5$.

For matroid constraints, Nong et al. (2020) showed that the greedy algorithm can achieve a $\frac{\gamma}{1+\gamma}$-approximation for maximizing a strictly monotone nonsubmodular function under a matroid constraint. The queries of the function value oracle of this algorithm is $O(n^2)$. Afterwards,Gong et al. (2019) combined the continuous greedy

Table 1 Overview of previous results

Setting	Constraint	Approximation ratio	Query complexity
Submodular	Cardinality	$1-\frac{1}{e}$	O(kn)Nemhauser et al. (1978)
	Matroid	$1-\frac{1}{e}$Calinescu et al. (2007)	–
Nonsubmodular	Cardinality	$1-e^{-\gamma}$	O(kn)Nong et al. (2020)
	Matroid	$\frac{\gamma}{1+\gamma}$	$O(n^2)$Nong et al. (2020)
	Matroid	$\gamma(1-\frac{1}{e})(1-e^{-\gamma})$	$O(n^2r^2)$Gong et al. (2019)
	Knapsack	$1-e^{-\gamma}$Nong et al. (2020)	–
	Knapsack	$\frac{1}{\alpha}(1-e^{-\alpha\widehat{\gamma}})$Zhang et al. (2019)	–

Table 2 Results in this paper

Constraint	Approximation ratio	Query complexity
Cardinality	$1-e^{-\gamma}$	$O(\frac{n}{\epsilon}log\frac{n}{\epsilon})$
Matroid	$\gamma^2(1-\frac{1}{e})^2$	$O(\frac{rn}{\epsilon^4}log^2\frac{n}{\epsilon})$

algorithm and the contention resolution schemes to achieve a $\gamma(1-e^{-1})(1-e^{-\gamma}-O(1))$-approximation.

The results mentioned above are listed in Table 1. In this paper, we will use the generic submodularity ratio of the objective function, and we give an overview of the results of this paper in Table 2. For nonsubmodular functions optimization, there are other research efforts in application-driving, such as supermodular-degree Feldman and Izsak (2014),Feige and Izsak (2013), difference of submodular functions Iyer and Bilmes (2012),Narasimhan and Bilmes (2005),Wu et al. (2019), and discrete difference of convex functions Maehara and Murota (2015),Wu et al. (2018).

1.4 Our result

In this paper, our main contribution is to develop algorithms that have both theoretical approximation guarantees, and fewer queries of the function value oracle. We use the simple decreasing threshold algorithm to solve the problem of maximizing a nonsubmodular function under a cardinality constraint. The following Theorem 1 implies the result in Badanidiyuru and Vondrák (2014) for submodular functions (the case that $\gamma = 1$ in Theorem 1). Besides, we use the continuous greedy approach and the contention resolution schemes to resolve the nonsubmodular maximizing problem under a matroid constraint. In Theorem 2, we improve the query times of a former result in Nong et al. (2020), from $O(n^2)$ to $O(rn\epsilon^{-4}log^2\frac{n}{\epsilon})$. Formally, we obtain the following two theorems.

Theorem 1 *For the problem of maximizing a monotone function with the generic submodularity ratio γ subject to a cardinality constraint, there exists a $(1-e^{-\gamma}-\epsilon)$-approximation algorithm, using $O(\frac{n}{\epsilon}log\frac{n}{\epsilon})$ queries to the function oracle.*

Theorem 2 *For the problem of maximizing a monotone function with the generic submodularity ratio γ subject to a matroid constraint, there is a $(\gamma^2(1-e^{-1})^2-O(\epsilon))$-approximation algorithm, using $O(rn\epsilon^{-4}log^2\frac{n}{\epsilon})$ queries to the function oracle.*

2 Preliminary

In this section, we propose some definitions and properties that we will use in the following of the paper.

Definition 1 (The Multilinear Extension Gong et al. (2019)) Given an increasing function $f : 2^N \to R_+$, a function $F : [0, 1]^N \to R_+$ is the multilinear extension of f if

$$F(\mathbf{x}) = E_{R\leftarrow R(\mathbf{x})}[f(R)] = \sum_{R\subseteq N} f(R) \cdot Pr[R(\mathbf{x}) = R],$$

where $Pr[R(\mathbf{x}) = R] = \prod_{u\in R} x_u \cdot \prod_{v\notin R}(1 - x_v)$, and $R(\mathbf{x})$ is a random set where element i appears independently with probability x_i.

Let $f : 2^N \to R_+$ be an increasing function with generic submodularity ratio γ and multilinear extension F. For a settled vector $\mathbf{x} \in [0, 1]^N$, the multilinear extension has some useful properties:

(a) $\frac{\partial F}{\partial x_i} = E[f_{R\setminus\{i\}}(i)]$;

(b) $\frac{\partial^2 F}{\partial x_j \partial x_i} = E[f_{R\cup\{j\}\setminus\{i\}}(i) - f_{R\setminus\{i,j\}}(i)]$;

(c) $|\frac{\partial^2 F}{\partial x_j \partial x_i}| \leq \frac{1}{\gamma} \max_{u\in N} f(u)$.

Definition 2 (Matroid Edmonds (2003)) Given a ground set N, a pair $(N, \mathcal{I})$ is called independent system if $\mathcal{I} \subseteq 2^N$ is satisfied that for any set $T \in (N, \mathcal{I})$, every set $S \subseteq T$ is also in $\mathcal{I}$. If an independent system satisfied for every two set $S, T \in \mathcal{I}$, s.t. $|S| < |T|$, there is an element $e \in T \setminus S$, s.t. $S \cup \{e\} \in \mathcal{I}$. Then the independent system be a matroid, and this is the augmentation property of a matroid. That is :

A *matroid* $\mathcal{M} = (N, \mathcal{I})$ can be defined as a finite N and a nonempty family $\mathcal{I} \subset 2^N$ such that:

(i) $S \subset T, T \in \mathcal{I}$, then $S \in \mathcal{I}$;

(ii) $S, T \in \mathcal{I}, |S| \leq |T|$, then there is an element $j \in T\backslash S$, $S + j \in \mathcal{I}$.

Definition 3 (Matroid Polytope Gong et al. (2019)) Given a matroid $(N, \mathcal{I})$, the *matroid polytope* is defined as $\mathcal{P}_{\mathcal{I}} = conv\{1_I : I \in \mathcal{I}\} = \{\mathbf{x} \geq 0$: for any $S \subset N; \sum_{j\in S} x_j \leq r_{\mathcal{I}}(S)$, where $r_{\mathcal{I}}(S) = max\{|I| : I \subset S, I \subset \mathcal{I}\}$ is the rank function of matroid $(N, \mathcal{I})$(hereinafter called r).

Lemma 1 *(*Property of the generic submodularity ratio Gong et al. (2019)*) For an increasing set function $f : 2^N \to R$, with generic submodularity ratio γ, it holds that:*

(a) $\gamma \in [0, 1]$;

(b) f *is submodular iff* $\gamma = 1$*;*
(c) $\sum_{j \in T \setminus S} f_S(j) \geq \gamma f_S(T)$*, for any* $S, T \subseteq N$*.*

Lemma 2 *(*Sviridenko (2004)*) Let* $\mathcal{M} = (N, \mathcal{I})$ *be a matroid, and* $B_1, B_2 \in \mathcal{B}$ *be two bases. Then there is a bijection* $\phi : B_1 \rightarrow B_2$ *such that for every* $b \in B_1$*, we have* $B_1 - b + \phi(b) \in \mathcal{B}$*.*

Lemma 3 *(*Badanidiyuru and Vondrák (2014)*) (Relative + Additive Chernoff Bound) Let* $X_1, X_2, ..., X_m$ *be independent random variables such that for each* i*,* $X_i \in [0, 1]$*, and let* $X = \frac{1}{m} \sum_{i=1}^{m} X_i$ *and* $\mu = E[X]$*. Then*

$$Pr[X > (1 + \alpha)\mu + \beta] \leq e^{-\frac{m\alpha\beta}{3}},$$

and

$$Pr[X < (1 - \alpha)\mu - \beta] \leq e^{-\frac{m\alpha\beta}{2}}.$$

Note that the generic submodularity ratio of a strictly monotone function is greater than 0. In the following of the paper, we consider the problem of maximizing a strictly monotone and normalized set function under certain constraints.

3 Cardinality constraint

First we present a simple algorithm, Simple Decreasing Threshold Algorithm, for **Problem (1)**: $\max\{f(S) : |S| \leq k, S \subseteq N\}$, where f is a monotone function with the generic submodularity ratio γ. Our goal is to develop an algorithm that have both theoretical approximation guarantee and fewer queries of the function value oracle.

Algorithm 1 Simple Decreasing Threshold Algorithm

```
Input: $f : 2^N \rightarrow R_+, k \in \{1, 2, ..., n\}$.
Output: A set $S \subset N$ satisfying $|S| \leq k$.
$S \leftarrow \emptyset$;
$d \leftarrow max_{j \in N} f(j)$;
for ($w = \frac{d}{\gamma}$; $w \geq \frac{\epsilon d}{n\gamma}$; $w \leftarrow \frac{1-\epsilon}{\gamma} w$) do
    for all $i \in N$ do
        if $|S \cup \{j\}| \leq k$ and $f_S(j) \geq w$ then
            $S \leftarrow S \cup \{j\}$
        end if
    end for
end for
return $S$
```

Next we prove Theorem 1. For the approximation ratio, it is necessary to proved the following claim.

Claim 1 *Let* O *be an optimal solution. Given a current solution* S *at the beginning of each iteration, the gain of the element added to* S *is at least* $\frac{1-\epsilon}{k} \sum_{a \in O \setminus S} f_S(a)$*.*

Proof Suppose that the next element chosen is a and the current threshold value is w. Then it implies the following inequalities

$$\begin{cases} f_S(x) \geq w, & if\ x = a; \\ f_S(x) \leq \frac{w}{1-\epsilon}, & if\ x \in O \setminus S \cup \{a\}. \end{cases}$$

The above inequalities imply that $f_S(a) \geq (1-\epsilon) f_S(x)$ for each $x \in O \backslash (S \cup \{a\})$. Taking an average over these inequations, we have

$$f_S(a) \geq \frac{1-\epsilon}{|O \backslash S|} \sum_{x \in O \backslash S} f_S(x) \geq \frac{1-\epsilon}{k} \sum_{x \in O \backslash S} f_S(x).$$

Now we finish the proof of Claim 1.

As for Algorithm 1, firstly, we check the number of queries of it. Obviously, there are two loops in Algorithm 1. In each inner loop, for all $i \in N$, we must get the value of $f_S(j)$ from the function value oracle, therefore each inner loop executes n queries of value oracle. According to the termination condition of the outer loop: $w \geq \frac{\epsilon d}{n\gamma}$. When per inner loop runs over, the threshold w will turn down by a ratio $\frac{1-\epsilon}{\gamma}$. Then we can calculate the query numbers of per outer loop is $O(\frac{1}{\epsilon} log \frac{n}{\epsilon})$. Therefore the algorithm using $O(\frac{n}{\epsilon} log \frac{n}{\epsilon})$ queries of the value oracle.

Next we prove the approximation ratio of Algorithm 1.

Proof Consider on a solution $S_i = \{a_1, a_2, ..., a_i\}$. After i steps, by Claim 1, we have

$$f_{S_i}(a_{i+1}) \geq \frac{1-\epsilon}{k} \sum_{a \in O \backslash S_i} f_{S_i}(a)$$

.

Then

$$\sum_{a \in O \backslash S_i} f_{S_i}(a) \geq \gamma f_{S_i}(O) \geq \gamma (f(O) - f(S_i)).$$

Therefore,

$$f(S_{i+1}) - f(S_i) = f_{S_i}(a+1) \geq \frac{1-\epsilon}{k} \gamma (f(O) - f(S_i)).$$

The second and the fourth inequations follows from that f is γ-submodular. We have

$$\begin{aligned} f(S_k) &\geq \left(1 - \left(1 - \frac{(1-\epsilon)\gamma}{k}\right)^k\right) f(O) \\ &\geq \left(1 - e^{-\gamma(1-\epsilon)}\right) f(O) \\ &\geq \left(1 - e^{-\gamma} - \epsilon\right) f(O). \end{aligned}$$

So we complete the proof of Theorem 1.

4 Matroid contraint

In this section we introduce the CR schemes roughly and we give a $(\gamma^2(1-\frac{1}{e})^2-O(\epsilon))$-approximation algorithm for maximizing a monotone nonsubmodular function under a matroid constraint.

4.1 Contention resolution schemes(CR schemes)

In this subsection we will study the CR schemes, an efficient rounding method, which is firstly researched by Chekuri et al. Chekuri et al. (2011) in 2011, in order to solve nonsubmodular functions maximization problem under a matroid constraint. Since Chekuri et al. and Gong Gong et al. (2019) have studied the theorem of CR schemes in detail, here we just give a rough description.

Given a matroid $(N, \mathcal{I})$, there is matroid polytope $\mathcal{P}_{\mathcal{I}} = conv\{\mathbf{1}_{\mathcal{I}} : I \in \mathcal{I}\}$ over matroid $(N, \mathcal{I})$. We denote a random set where each element $i \in N$ appears independently with probability x_i by $R(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathcal{P}_{\mathcal{I}}$. $Pr[R(\mathbf{x}) = R] = \prod_{i\in R} x_i \cdot \prod_{i\notin R}(1 - x_i)$. Then, for any fixed element $j \in N$, $Pr[R(\mathbf{x}) = R|j \in R] = \prod_{i\in R\setminus\{j\}} x_i \cdot \prod_{i\notin R\cup\{j\}}(1 - x_i)$.

Definition 4 (CR schemes Gong et al. (2019)) For any vector $\mathbf{x} \in \mathcal{P}_{\mathcal{I}}$ and any subset $A \subseteq N$, a CR scheme ß for $\mathcal{P}_{\mathcal{I}}$ is a removal procedure that returns a random set $\text{ß}_{\mathbf{x}}(A)$ such that $\text{ß}_{\mathbf{x}}(A) \subseteq A \cap support(\mathbf{x})$ where support $(\mathbf{x}) = \{j \in N|\mathbf{x}_j > 0\}$ and $\text{ß}_{\mathbf{x}}(A) \in \mathcal{I}$ with probability 1(since the definition of $\mathcal{P}_{\mathcal{I}}$).

A discussion of existence of CR schemes is as follows. Remark that a CR scheme π is a collection of $\text{ß}_{\mathbf{x}}$, for each $\mathbf{x} \in \mathcal{P}_{\mathcal{I}}$, related to the matroid polytope $\mathcal{P}_{\mathcal{I}}$, and every $\text{ß}_{\mathbf{x}}$ corresponds to a probability distribution. Now let $\varphi : 2^N \to \mathcal{I}$ be a valid mapping that for any $A \subseteq N$, $\varphi(A) \subseteq A$ and $\varphi(A) \in \mathcal{I}$. Denote the family of all valid mapping from 2^N to $\mathcal{I}$ by ψ^*. Easy to see, since $\pi(A) \subseteq A$ and $\pi(A) \in \mathcal{I}$, then $\text{ß}_{\mathbf{x}} \in \psi^*$. On the one hand, the probability distribution picks and applies mapping φ with probability λ_φ from ψ^*, since each $\text{ß}_{\mathbf{x}}$ corresponds to a probability distribution, then $\text{ß}_{\mathbf{x}}$ can written as $(\lambda_\varphi)_{\varphi\in\psi^*}$. On the other hand, any probability distribution $(\lambda_\varphi)_{\varphi\in\psi^*}$ can respond to a random scheme $\text{ß}_{\mathbf{x}}$, that is for any subset $A \subseteq N$, $\text{ß}_{\mathbf{x}}$ picks $\varphi \in \psi^*$ with probability λ_φ and returns $\varphi(A)$. Therefore, given a $\mathbf{x} \in \mathcal{P}_{\mathcal{I}}$, if the probability distribution $(\lambda_\varphi)_{\varphi\in\psi^*}$ exists, the CR scheme $\text{ß}_{\mathbf{x}}$ also exists.

Since the $R(\mathbf{x})$ may not be an independent set of the matroid, researchers introduced the c-balanced CR schemes, and it's function is to remove some elements in $R(\mathbf{x})$ in order to guarantee that the $R(\mathbf{x})$ be an independent set. The definition of c-balanced CR schemes is as follows.

Definition 5 (C-balanced schemes Gong et al. (2019)) A CR scheme π for $\mathcal{P}_{\mathcal{I}}$ is c-balanced if it satisfies $Pr[j \in \text{ß}_{\mathbf{x}}(R(\mathbf{x}))|j \in R(\mathbf{x})] \geq c$, for any vector $\mathbf{x} \in \mathcal{P}_{\mathcal{I}}$ and any element $j \in$ support $(\mathbf{x})$.

Definition 6 (Monotone schemes Gong et al. (2019)) A CR scheme π for $\mathcal{P}_{\mathcal{I}}$ is monotone if it satisfies $Pr[j \in \text{ß}_{\mathbf{x}}(A_1)] \geq Pr[j \in \text{ß}_{\mathbf{x}}(A_2)]$, for any $\mathbf{x} \in \mathcal{P}_{\mathcal{I}}$, $j \in A_1 \subseteq A_2$.

After given the two definitions, Chekuri proposed and proved the folloeing important claim.

Claim 2 *Chekuri et al. (2011) For any matroid polytope $\mathcal{P}_{\mathcal{I}}$, there exists a monotone $(1-\frac{1}{e})$-balanced CR scheme and no c-balanced scheme with better c.*

The existence of c-balanced CR scheme and Claim 2 had been proved by Chekuri in detail. We only quote here.

As for CR schemes, the most important and useful result is the approximation ratio of rounding a fractional solution. Gong et al.Gong et al. (2019) proved that there is a $\gamma(1-\frac{1}{e})$ approximation ratio when rounding a fractional solution in the nonsubmodular setting by using CR schemes. Here we do not give unnecessary details.

4.2 Matroid contraint

In this subsection, we give a $(\gamma^2(1-\frac{1}{e})^2 - O(\epsilon))$-approximation algorithm for **Problem (2)**: $\max\{f(S) : S \in \mathcal{I}, S \subseteq N\}$, where f is a monotone function with the generic submodularity ratio γ, using $O(rn\epsilon^{-4}log^2\frac{n}{\epsilon})$ queries to the value oracle. The genenral outline of our algorithm follows from the continuous greedy algorithm in Badanidiyuru and Vondrák (2014). With a fractional solution being built up gradually from $\mathbf{x} = \mathbf{0}$, and finally using the contention resolution schemes from Chekuri et al. (2011) to convert the fractional solution to an integer one.

4.3 Notation

In the following context, for $\mathbf{x} \in [0, 1]^N$, a random set that contains each element $i \in N$ independently with probability x_i is denoted by $R(\mathbf{x})$. We denote $R(\mathbf{x}+\epsilon\mathbf{1}_S)$ as $R(\mathbf{x}, S)$. Before we analyse the approximation of Algorithm 2, we give and analyse a subroutine-Algorithm 3, which is used in Algorithm 2. This subroutine takes a current fractional solution $\mathbf{x}$ and adds to it an increment corresponding to an independent set B, to obtain $\mathbf{x} + \epsilon\mathbf{1}_B$. The B in the Algorithm 3 can be seemed as the fastest increasing direction of the value f. The way we find B in Algorithm 3 is similar to that in Algorithm 1.

Claim 3 *Let O be an optimal solution. Given a fractional solution $\mathbf{x}$, by the Algorithm 3 we can get a new fractional solution $\mathbf{x}' = \mathbf{x} + \epsilon\mathbf{1}_B$ such that*

$$F(\mathbf{x}') - F(\mathbf{x}) \geq \epsilon\left(\gamma(1-\epsilon) - \frac{2\epsilon}{\gamma}\right) f(O) - F(\mathbf{x}').$$

Proof Actually, the Algorithm 3 gave the fastest increasing direction of the $f(S)$. It is because that B corresponding to a direction vector $\mathbf{1}_B$. We suppose that $B = \{b_1, b_2, ..., b_r\}$, $b_i (i \in \{1, 2, ..., r\})$ is the element that joins in the B at the ith step. If

Algorithm 2 Revised Continuous Greedy Algorithm

```
Input: $f : 2^N \rightarrow R_+,\ \mathcal{I} \subseteq 2^N$.
Output: A set $S \subseteq N$ satisfying $S \in \mathcal{I}$.
$\mathbf{x} \leftarrow \mathbf{0}$;
for ($t \leftarrow \epsilon; t \leq 1; t \leftarrow t + \epsilon$) do
    $B \leftarrow$ the output of Algorithm 3
    $\mathbf{x} \leftarrow \mathbf{x} + \epsilon \mathbf{1}_B$
end for
$S \leftarrow$ contention resolution schemes $(\mathbf{x}, \mathcal{I})$
return $S$
```

Algorithm 3 Decreasing Threshold Procedure

```
Input: $f : 2^N \rightarrow R_+,\ \mathbf{x} \in [0,1]^N,\ \epsilon \in [0,1],\ \mathcal{I} \subseteq 2^N$.
Output: A set $B \subset N$ satisfying $B \in \mathcal{I}$
$B \leftarrow \emptyset$;
$d \leftarrow max_{j \in N} f(j)$;
for ($w = \frac{d}{\gamma}; w \geq \frac{\epsilon d}{r\gamma}; w \leftarrow \frac{1-\epsilon}{\gamma} w$) do
    for all $e \in N$ do
        $w_e(B, \mathbf{x}) \leftarrow$ estimate of $E[f_{R(\mathbf{x}+\epsilon \mathbf{1}_B)}(e)]$ by averaging $\frac{rlogn}{\epsilon^2}$ random samples.
        if $B \cup \{e\} \in \mathcal{I}$ and $w_e(B, \mathbf{x}) \geq w$ then
            $B \leftarrow B \cup \{e\}$
        end if
    end for
end for
return $B$
```

the threshold has turned down to the $\frac{\epsilon d}{r\gamma}$ before the algorithm terminates, in this case we can add the fictitious elements in order to make sure $|B| = r$.

Next we should build up the connection between the output solution and the optimal solution. We suppose that the $O = \{o_1, o_2, ..., o_r\}$ be the optimal solution and according to Lemma 2, there is a bijection between O and B s.t. $\phi(b_i) = o_i$. Additionally, the first i elements of B are denoted as B_i , and the first i elements of O are denoted as O_i.

According to Lemma 3, we know that there is an error while using $w_e(B_i, \mathbf{x})$ to estimate $E[f_{R(\mathbf{x}, B_i)}(e)]$, with high probability we have the following inequality

$$|w_e(B_i, \mathbf{x}) - E[f_{R(\mathbf{x}, B_i)}(e)]| \leq \frac{\epsilon f(O)}{\gamma r} + \epsilon E[f_{R(\mathbf{x}, B_i)}(e)]. \tag{1}$$

We get that when an element b_i is chosen, o_i is a candidate element which could have been chosen instead of b_i. Therefore, according to the process of Algorithm 3, and because either o_i is a potential candidate of value within a factor of $1-\epsilon$ of the element we chose instead, or the algorithm terminated and all remaining elements have value below $\frac{\epsilon d}{r\gamma}$, we have

$$w_{b_i}(B_{i-1}, \mathbf{x}) \geq (1-\epsilon) w_{o_i}(B_{i-1}, \mathbf{x}) - \frac{\epsilon d}{\gamma r}. \tag{2}$$

Combining (1) and (2), and the fact that $f(O) \geq d$, we have

$$E[f_{R(\mathbf{x},B_{i-1})}(b_i)] \geq (1-\epsilon)E[f_{R(\mathbf{x},B_{i-1})}(o_i)] - 2\frac{\epsilon f(O)}{\gamma r}. \tag{3}$$

Then at each step in Algorithm 2:

$$\begin{aligned}
F(\mathbf{x}') - F(\mathbf{x}) &= F(\mathbf{x} + \epsilon \mathbf{1}_B) - F(\mathbf{x}) \\
&= \sum_{i=1}^{r} (F(\mathbf{x} + \epsilon \mathbf{1}_{B_i}) - F(\mathbf{x} + \epsilon \mathbf{1}_{B_{i-1}})) \\
&= \sum_{i=1}^{r} \epsilon \frac{\partial F}{\partial x_{b_i}}\Big|_{\mathbf{x}+\epsilon \mathbf{1}_{B_{i-1}}} \\
&\geq \sum_{i=1}^{r} \epsilon E[f_{\mathbf{x}+\epsilon \mathbf{1}_{B_{i-1}}})(b_i)] && (monotone) \\
&\geq \sum_{i=1}^{r} \epsilon \left((1-\epsilon)E[f_{\mathbf{x}+\epsilon \mathbf{1}_{B_{i-1}}})(o_i)] - 2\frac{\epsilon f(O)}{\gamma r} \right) \\
&\geq \sum_{i=1}^{r} \epsilon \left((1-\epsilon)\gamma E[f_{R(\mathbf{x}+\epsilon \mathbf{1}_{B \cup \{o_1, o_2, \ldots, o_{i-1}\}})}(o_i)] - 2\frac{\epsilon f(O)}{\gamma r} \right) \\
&= \epsilon((1-\epsilon)\gamma E[f(R(\mathbf{x}') \cup O) - f(R(\mathbf{x}'))] - 2\frac{\epsilon f(O)}{\gamma} \\
&\geq \left(\gamma\epsilon(1-\epsilon) - \frac{2\epsilon^2}{\gamma} \right) f(O) - \epsilon F(\mathbf{x}') && (monotone) \\
&= \epsilon \left(\left(\gamma(1-\epsilon) - \frac{2\epsilon}{\gamma} \right) f(O) - F(\mathbf{x}') \right).
\end{aligned}$$

The second inequality follows from that o_i is the candidate element when b_i is chosen, and the third inequality is due to the definition of generic submodularity ratio γ.

Claim 4 *Algorithm 3 makes $O(\frac{1}{\epsilon^3}nrlog^2\frac{n}{\epsilon})$ queries to the function oracle.*

Proof Obviously, there are two loops in Algorithm 3. According to the termination condition of the outer loop, we get the query numbers of per outer loop is $O(\frac{1}{\epsilon}log\frac{n}{\epsilon})$. The number of iterations in the inner loop is n, and the number of samples per evaluation of F is $\frac{1}{\epsilon^2}rlogn$ in per inner loop. Therefore, Algorithm 3 makes $O(\frac{1}{\epsilon^3}nrlog^2\frac{n}{\epsilon})$ queries to the value oracle.

Claim 5 *Algorithm 2 has an approximation ratio of $\gamma^2(1-\frac{1}{e})^2 - O(\epsilon)$.*

Proof Define $\Omega = (\gamma(1-\epsilon) - \frac{2\epsilon}{\gamma})f(O)$. Substituting this in the result of Claim 2, we have

$$F(\mathbf{x}(t+\epsilon)) - F(\mathbf{x}(t)) \geq \epsilon(\Omega - F(\mathbf{x}(t+\epsilon))).$$

Add $\Omega - F(\mathbf{x}(t+\epsilon))$ to the inequation we have

$$\Omega - F(\mathbf{x}(t+\epsilon)) \leq \frac{\Omega - F(\mathbf{x}(t))}{1+\epsilon}.$$

Using induction to this inequation, we have

$$\Omega - F(\mathbf{x}(t)) \leq \frac{\Omega}{(1+\epsilon)^{\frac{t}{\epsilon}}}.$$

Substituting $t = 1$ and rewriting the inequation, we have

$$\begin{aligned} F(\mathbf{x}(t)) &\geq \left(1 - \frac{1}{(1+\epsilon)^{\frac{1}{\epsilon}}}\right)\Omega \\ &= \left(1 - \frac{1}{(1+\epsilon)^{\frac{1}{\epsilon}}}\right)\left(\gamma(1-\epsilon) - \frac{2\epsilon}{\gamma}\right) \\ &\geq \gamma\left(1 - \frac{1}{e}\right) - O(\epsilon). \end{aligned}$$

Besides, when we use CR schemes to convert the fractional solution to the integer one, there also have an approximation ratio which is $\gamma(1-\frac{1}{e})$.

Therefore, the approximation ratio of Algorithm 2 is $\gamma^2(1-\frac{1}{e})^2 - O(\epsilon)$.

Claim 6 *Algorithm 2 makes $O(\frac{1}{\epsilon^4} nrlog^2 \frac{n}{\epsilon})$ queries to the function oracle.*

Proof Observe that in Algorithm 2, the queries to the function oracle is only related to Algorithm 3. Therefore the total number of oracle calls to the function is equal to the number of the loop multiplied with the number of oracle calls in one iteration. So we get the queries to the function oracle are at most $O(\frac{1}{\epsilon^4} nrlog^2 \frac{n}{\epsilon})$.

References

Ageev AA, Sviridenko MI (2004) Pipage rounding: a new method of constructing algorithms with proven performance guarantee. J Combinatorial Optim 8(3):307–328

N. Alaluf, A. Ene, M. Feldman, H. Nguyen, A. Suh, Optimal Streaming Algorithms for Submodular Maximization with Cardinality Constraints. In LIPI (2020)

Badanidiyuru A, Mirzasoleiman B, Karbasi A et al (2014) Streaming Submodular Maximization: Massive AData Summarization on the Fly. ACM 671–680

Badanidiyuru A, Vondrák J (2014) Fast Algorithm for Maximizing Submodular Functions. SODA 1497–1514

Balkanski E, Breuer A, Singer Y (2018) Non-monotone Submodular Maximization in Exponentially Fewer Iterations. NIPS 2353–2364

Balkanski E, Rubinstein A, Singer Y (2019) An Exponential Speedup in Parallel Running Time for Submodular Maximization without Loss in Approximation. SODA 283–302

Balkanski E, Singer Y (2018) The Adaptive Complexity of Maximizing a Submodular Function. STOC 1138–1151

A. A. Bian, J. M. Buhmann, A. Krause, S. Tschiatschek, Guarantees for Greedy Maximization of Nonsubmodular Functions with Applications. In ICML (2017)

Breuer A, Balkanski E, Singer Y (2020) The FAST Algorithm for Submodular Maximization. ICML 1134–1143

Buchbinder N, Feldman M (2018) Deterministic algorithms for submodular maximization problems. ACM 14(3):1–20

Buchbinder N, Feldman M (2019) Constrained submodular maximization via a nonsymmetric technique. Math Op Res 44(3):988–1005

Calinescu G, Chekuri C, Pál M, Vondrák J (2007) Maximizing a Submodular Set Function Subject to a Matroid Constraint. IPCO 182–196

Chekuri C, Vondrák J, Zenklusen R (2011) Submodular Function Maximization Via the Multilinear Relaxation and Contention Resolution Schemes. STOC 783–792

Conforti M, Cornuéjols G (1984) Submodular set functions, matroids and the greedy algorithm: tight worstcase bounds and some generalizations of the Rado-Edmonds theorem. Discret Appl Math 7(3):251–274

Das A, David K (2011) Submodular meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection. ICML 1057–1064

Edmonds J (2003) Submodular functions, matroids, and certain polyhedra. LNCS 2570:11–26

Elenberg ER, Khanna R, Dimakis AG, Negahban S (2018) Restricted strong convexity implies weak submodularity. Annals Stat 46(6B):3539–3568

Ene A, Nguyen HL (2019) Submodular Maximization with Nearly-optimal Approximation and Adaptivity in Nearly-linear Time. SODA 274–282

Fahrbach M, Mirrokni V, Zadimoghaddam M (2019) Non-monotone Submodular Maximization with Nearly Optimal Adaptivity and Query Complexity. PMLR 1833–1842

U. Feige, R. Izsak, Welfare Maximization and the Supermodular Degree. In ITCS(2013), 247–256

Feige U (1998) A threshold of ln n for approximating set cover. JACM 45(4):634–652

Feige U, Mirrokni VS, Vondrák J (2011) Maximizing non-monotone submodular functions. J Comput 40(4):1133–1153

Feldman M, Izsak R (2014) Constrained monotone function maximization and the supermodular degree. LIP I:160–175

Feldman M, Norouzi-Fard A, Svensson O, Zenklusen R (2020) The One-way Communication Complexity of Submodular Maximization with Applications to Streaming and Robustness. STOC 1363–1374

Gong S, Nong Q, Liu W, Fang Q (2019) So parametric monotone function maximization with matroid constraints. J Glob Optim 75(3):833–849

Gotovos A, Karbasi A, Krause A (2015) Non-Monotone Adaptive Submodular Maximization. In IJCA I:1996–2003

Iyer R, Bilmes J (2012) Algorithms for Approximate Minimization of the Difference between Submodular Functions, with Applications. UAI 407–417

Jiang Y, Wang Y, Xu D, Yang R, Zhang Y (2019) Streaming algorithm for maximizing a monotone nonsubmodular function under D-knapsack constraint. Optim Lett 14(2):1–14

E. Kazemi, M. Mitrovic, Zadimoghaddam, S. Lattanzi, A. Karbasi, Submodular Streaming in all its Glory: Tight Approximation, Minimum Memory and Low Adaptive Complexity. In ICML (2019) 5767–5784

A. Kohara, K. Okano, K. Hirata, et al. Sensor Placement Minimizing the State Estimation Mean Square Error: Performance Guarantees of Greedy Solutions (2020). arXiv:2004.04355

Kuhnle A, Smith JD, Crawford VG, Thai MT (2018) Fast maximization of Non-submodular, Monotonic Functions on the Integer Lattice. ICML 2791–2800

Li M, Zhou X, Tan J, Wang W (2020) Non-Submodular Streaming Maximization with Minimum Memory and Low Adaptive Complexity. LNCS 214–224

Maehara T, Murota K (2015) A framework of discrete DC programming by discrete convex analysis. Math Programm 152(1–2):435–466

B. Mirzasoleiman, A. Karbasi, R. Sarkar, A. Sarkar, Distributed Submodular Maximization: Identifying Representative Elements in Massive Datas. In NIPS (2013)

Mirzasoleiman B, Badanidiyuru A, Karbasi A, Vondrák J, Krause A (2015) Lazier Than Lazy Greedy. In AAA I:1812–1818

Narasimhan M, Bilmes JA (2005) A Submodular-Supermodular Procedure with Applications to Discriminative Structure Learning. UA I:404–412

Nemhauser GL, Wolsey LA (1978) Best algorithms for approximating the maximum of a submodular set functions. Math Op Res 3(3):177–188

Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions-I. Math Programm 14(1):265–294

Nong Q, Sun T, Gong S, Sun T, Fang Q, Du D, Shao X (2020) Maximize a monotone function with a generic submodularity ratio. Theor Computer Sci. https://doi.org/10.1016/j.tcs.2020.05.018

R. Santiago, Y. Yoshida, Weakly Submodular Function Maximization Using Local Submodularity Ratio(2020). arXiv:2004.14650

Sviridenko M (2004) A note on maximizing a submodular set function subject to a Knapsack constraint. Op Res Lett 32(1):41–43

Sviridenko M, Vondrák J, Ward J (2017) Optimal approximation for submodular and supermodular optimization with bounded curvature. Math Op Res 42(4):1197–1218

Vondrák J (2008) Optimal Approximation for the Submodular Welfare Problem in the Value Oracle Model. STOC 67–74

Vondrák J (2010) Submodularity and curvature: the optimal algorithm. RIMS Kokyuroku Bessatsu B 23:253–266

Wang Y, Xu D, Wang Y, Zhang D (2019) Non-submodular maximization on massive data streams. J Glob Optim 30:1–15

Wu C, Wang Y, Lu Z et al (2018) Solving the degree-concentrated fault-tolerant spanning subgraph problem by DC programming. Math Programm 169(1):255–275

Wu W, Zhang Z, Du D (2019) Set function optimization. J Op Res Soc Chin 7(2):183–193

Zhang Z, Liu B, Wang Y, Xu D, Zhang D (2019) Greedy Algorithm for Maximization of Non-submodular Functions Subject to Knapsack Constraint. Computing and Combinatorics 651–662