AIMS Mathematics, 11(3): 7555–7572.
DOI: 10.3934/math.2026310
Received: 29 October 2025
Revised: 30 January 2026
Accepted: 10 February 2026
Published: 23 March 2026

https://www.aimspress.com/journal/Math

Research article

Streaming algorithm for balance gain and cost with cardinality constraint on the integer lattice[1]

Jingjing Tan[1,2], Cuiping Ge[2,*], Fengmin Wang[3] and Ziyang Li[1]

[1] School of Mathematics and Statistics, Weifang University, Weifang 261061, China
[2] Shandong Key Laboratory of Intelligent Manufacturing Technology for Advanced Power Equipment，Weifang 261061, China
[3] Beijing Jinghang Research Institute of Computing and Communication, Beijing 100074, China

* **Correspondence:** Email: wfgecuiping@163.com; Tel: 0536-8785525.

Abstract: The team formation problem is a very important problem in the labor market that has been proved to be NP-hard. This paper proposes an efficient bicriteria streaming algorithm aimed at striking a balance between gain and cost in team formation problems with cardinality constraints on the integer lattice. To address this, we utilized a model optimized for maximizing the difference between a nonnegative normalized monotone submodular function and a nonnegative linear function. We further consider the case where the first function of the object function is weakly submodular. Combining the lattice binary search with the threshold method, we present an online algorithm called bicriteria streaming algorithmsalgorithm. Concomitantly, we comprehensively analyze both models.

Keywords: knapsack constraint; integer lattice; cardinality function; streaming algorithm
Mathematics Subject Classification: 90C25

1. Introduction

Team formation considerations occur in a diverse range of scenarios, including but not limited to project management, workforce allocation, sports team composition, and academic

[1] A preliminary version of this paper appeared in the 30th International Computing and Combi natorics Combinatorics Conference (COCOON 2024), 2025, pp. 324–331.

collaboration. These problems have received intense attention in recent research efforts [1–4]. In essence, the primary objective of addressing team formation issues is to strike an optimal balance between accruing benefits and the costs involved, elements that can largely fluctuate based on the context. As a case in point, within the realm of project management, one potential goal might be the maximization of team productivity, or alternatively, minimizing the duration required for project completion. Similarly, in sports team composition, the key aim could rest upon crafting a well-balanced team, wherein members' skills and capabilities complement each other, ultimately fostering an enhancement in overall performance levels. These contextualized problems can be carefully crafted into a submodular optimization issue. The primary target herein would involve choosing a subset V from a larger set E with an aim to maximize a specific objective function denoted as $g(V)$. Given the nature of practical circumstances, often it becomes critical to ensure an equilibrium between the advantages reaped by selecting V quantified through the function $g(V)$ and the relative cost implicated, denoted as $c(V)$. To cater to these application domains, Nikolakaki et al. [5] formulated the team formation issue as a maximization problem, which can be characterized as follows:

$$\max_{v \subset E} g(V) - c(V),$$

where the function $g(V)$ is monotone and nonnegative submodular, while $c(V)$ represents a nonnegative linear function that sums up the costs of selected elements. This computational model, through its robust architecture and flexible algorithms, provides an efficient method of generalization for different types of data mining applications. Unconstrained models allow for detailed and thorough exploration in the realm of data mining, ensuring that valuable information is not lost due to arbitrary element restrictions. Therefore, it assists in providing more accurate and high-quality results in various data mining applications.

Consider a constrained model to find the optimal number of elements to be part of the solution. In addressing cardinality constraints, a conventional approach involves employing the standard greedy algorithm and considering online versions of the above questions. By utilizing a variation of the continuous greedy method, Sviridenko et al. [6] obtained a solution set V satisfying inequalities $g(V) - c(V) \geq (1 - e^{-1})$. $g(V) - c(V)$ under identical model conditions with a constraint matroid Constraint Matroid constraint. Du et al. [7] proposed a two-criterion approximation algorithm, which is considered to be a more suitable performance measurement method than traditional approximation algorithms. Feldman [8] designed the bicriteria approximation algorithm and obtained similar conclusions as Sviridenko et al. [6].

In the team formation problem, we also encounter that a fixed skill can be owned by multiple team members at the same time; in that case, we need to decide not only which skill to choose but also the number of choices. Then, the problem becomes an integer lattice submodule maximization problem. The concepts of lattice theory and optimization were first proposed by Lovasz [9]. The modular maximization problem pertaining to integer lattices is a traditional enlargement of the primary modular maximization problem. For a finite set E of size n, where each element e_i belongs to E, we set χ_{e_i} as the i-th unit vector. Then, we use $\mathbb{N}^E$ to represent the integer grid defined over E. $\mathcal{G}_B$ is a set of monotone and nonnegative DR-submodule functions, and $\mathcal{C}_B$ is a set of linear functions. We denote g as a DR-submodular function if the following condition is satisfied: for any z, y, given that $z \leq y$, the inequality

$$g(y + \chi_{e_i}) - g(y) \leq f(z + \chi_{e_i}) - f(z)$$

holds. This property is referred to as the fulfilment of DR-submodularity. So- ma

Soma et al. [10] conducted research on the DR-submodular maximization problem, taking into account constraints such as cardinality, knapsack, and matroid. Utilizing both the threshold technique and binary search technique, they developed a $(1-e^{-1})$-threshold greedy algorithm. Gottschalk et al. [11] cleverly used the bidirectional greed technique: according to the predetermined order of indicators, the upper certainty to rise and fall is iterated until the two are equal to output the final solution. The authors suggested an algorithm that operates in polynomial time and offers a 1/3 approximation ratio, tackling the issue of maximizing non-monotonic modular functions on integer lattices. Nong et al. [12] further improved the approximate ratio to 1/2. Based on the optimal budget allocation problem, Soma et al. [13] established a monotone lattice submodular maximum on integer lattices with knapsack constraints, with an approximation ratio $(1-e^{-1})$. By using the improved threshold technique, Tan et al. [14] proposed an online threshold streaming algorithm with 1/3-approximate ratios for the lattice submodular maximization problems with knapsack constraints in a flow model.

Although the submodular function maximization problem is widely used, many problems that cannot be properly described remain. Research on the maximization of non-submodular functions has shown significant progress within the realm of mathematical modelling and optimization. This field is relevant for a broad spectrum of applications, contributing to our comprehension regarding highly complex systems. The research of non-submodular function maximization is also expanding to new application areas, such as social network influence maximization and resource allocation optimization. These applications are advancing the field and inspiring new research directions [15–17]. First, scholars used a submodular rate, which quantifies the difference between the function and the submodular function to describe the non-submodular function. The concept of submodular rate for a set function was initially proposed by Das et al. [18]. Bian et al. [19] developed a greedy algorithm for non-submodular maximization problems with cardinality constraints on finite sets using submodular rates. By introducing the generalized curvature c of the set function, they achieved an approximation ratio of $(1-e^{\alpha c})/c$ and proved its compactness. Liu and colleagues [20] conducted research on the issue of maximizing the sum of weakly monotonic submodular functions and supermodular functions, offering both offline and streaming algorithms for this problem. Kuhnle et al. [21] extended the notion of submodular rate to lattice submodular function and designed a threshold greedy algorithm on an integer lattice to solve this problem efficiently. Zhang et al. [22] devised a greedy algorithm that incorporates curvature to significantly improve the approximation ratio. Using the threshold technique and lattice binary search technique, Tan [23] proposed a threshold flow algorithm for the above problems.

Inspired by the above results, we design an efficient bicriteria streaming algorithm to construct a balance between gain and cost in a team formation problem $(G-C)$, with cardinality constraint on the integer lattice. The problem is formulated as follows: Let E be a ground set of size n, z an n-dimensional vector from $\mathbb{N}^E$, g a normalized submodular function defined on $\mathbb{N}^E$ satisfying nonnegativity and monotonicity, and c a linear function also defined on $\mathbb{N}^E$,

$$\begin{aligned} &\text{maximize} \quad g(Z)-c(Z), \\ &\text{s.t.} \qquad\quad z \le b \\ &z(E) \le k \,.\, z(E) \le k, \end{aligned} \tag{1}$$

where the total budget is $k \in \mathbb{N}$, and $\mathbf{B} = \{z \in \mathbb{N}^E : z \le b\}$ is a box in $\{\mathbb{N} \bigcup \{+\infty\}\}^E$. $\mathcal{G}_B$ is a set of monotone and non-negativenonnegative DR-submodule functions, and $\mathcal{C}_B$ is a set of nonnegative linear functions. To address this problem, we initially propose Algorithm 1 when the objective function is a monotone, nonnegative, submodular function. In Algorithm 1, we call a binary search algorithm to decide whether to keep the currently arrived element and set a level for the remaining elements in memory. The effect of Algorithm 2 on the time complexity of the whole algorithm is so slight that we can ignore it. We continue to delve into the situation where the objective function stands as a non-submodular function. A streaming algorithm 3 is conceptualized leveraging the lattice submodular rate. The performance of this particular algorithm is then meticulously analyzed, namely the space complexity and the memory of the whole algorithm related to the lattice submodular rate, which is $O(k \log(k/\alpha)/\varepsilon)$.

Tan et al. [23] focuses on streaming algorithms for maximizing on the integer lattice under a knapsack constraint, which takes the form of $\sum_i \omega_i x_i \le B$ (where ω_i denotes the weight of element i, and B is the weight upper bound). The design of this algorithm needs to account for the heterogeneity and accumulativeness of weights. In contrast, this paper switches the constraint to a cardinality constraint $x(E) \le k$ (where k is the upper bound on the number of elements). Under the cardinality constraint, the algorithm must directly control the quantity threshold of selected elements instead of the total weight, making the weight-based greedy strategies in traditional knapsack constraint scenarios no longer applicable. The cardinality constraint has stronger universality and practicality in real-world scenarios. For instance, in resource allocation problems, when the quantity of available resources (e.g., the number of servers, personnel quotas) is the core limiting factor rather than the total weight, the model proposed in this paper is more aligned with practical needs. In large-scale data processing scenarios, the algorithm implementation under the cardinality constraint is more lightweight, as it does not require storing and calculating element weights, thus reducing the memory overhead and time complexity of streaming processing.

We will divide the main results of this paper into three parts. In Section 2, we will elaborate on the definitions and properties of integer lattices and submodules. In Section 3, we design two combination algorithms to solve the above problems and analyze the performance of the two algorithms. In Section 4, the performance of the algorithm is verified through numerical experiments. Finally, we summarize the main results in Section 5.

2. Preliminaries

In this section, we elucidate numerous symbols, conceptual definitions, and axioms.

We use $[k]$ to denote all positive integers between 1 and k. We establish a base set $E = \{e_1, e_2, ..., e_n\}$ and z as an n-dimensional vector in $\mathbb{N}^E$. The segment of coordinate $e_i \in E$ for z is represented as $z(e_i)$. We employ 0 to depict the null vector and χ_{e_i} as the standard base vector. Thus, all components are at zero, except for the i-th component that stands at unity. For any subset $Z \subseteq E$, its characteristic vector is χ_Z, and $z(Z)$ amounts to the summation of $z(e_i)$ over all elements e_i belonging to Z. Within the set $z \in \mathbb{N}^E$, $supp^+(z) = \{e \in E \mid z(e) > 0\}$ signifies the support set of z. We define a multi-set $\{z\}$, where e appears $z(e)$ times, and use $|\{z\}| := z(E)$ to express the total amount of $z(E)$, and $z \wedge y$ stands for the coordinate-wise

maximum and minimum of vectors z and y, respectively. Multi-sets $\{z\}$ and $\{y\}$ have their difference denoted as $\{z\}\setminus\{y\} := \{(z \setminus y) \vee 0\}$.

A function $f : \mathbb{N}^E \to \mathbb{R}_+$ is known as monotone non-decreasing if and only if $f(z) \le f(y)$ can be satisfied for any $z \le y$. A function f is deemed nonnegative and normalized if it invariably fulfills the conditions $f(z) \ge 0$ (for all $z \in \mathbb{N}^E$) and $f(0) = 0$.

Definition 1. *A function $f : \mathbb{N}^E \to \mathbb{R}_+$ is termed as DR-submodular if f fulfills this particular inequality for every z and y in $\mathbb{N}^E$:*

$$f(y + \chi_e) - f(y) \le f(z + \chi_e) - f(z).$$

The value denoted by $f(z \mid y)$ presents the marginal increase of a vector z concerning y expressed as

$$f(z \mid y) = f(z + y) - f(y).$$

Finally, denote z^* as the ideal solution vector.

3. Bicriteria streaming algorithms

For the case where the first function is submodular and non-submodular, we will design two combinatorial algorithms and analyze their performance.

3.1. Maximizing $G - C$ function

First, we give a streaming algorithm for G-C on the integer lattice, which is stated in Algorithm 1. In the process of running the algorithm, we not only need to decide whether the current element is preserved, but also count the level of the elements that are preserved. At this point, we need to call Algorithm 2 to perform the calculation. The Algorithm 2 details are as follows.

Let z^* denote the optimal solution, and z is the final solution vector returned by Algorithm 1, $p = z^*(E), q = z(E)$. Assume that z^{i-1} is the set when the element e_i arrives, $i = 1, 2, ...q, z_0 = 0, z_q = 0$. There are two cases $z(E) = k$ (Lemma 1) and $z(E) < k$ (Lemma 2).

Algorithm 1 Streaming algorithm for G-C on the integer lattice

Input: Data stream E, $g \in G_b$, $c \in C_b$, cardinality constraint $k \in \mathbb{N}_+$, threshold value τ.

Output: a vector $z \in \mathbb{N}^E$.

```
z ← 0;
for e ∈ E do
    if z(E) < k, then
        find the level t with BS Algorithm (g, c, z, b, e, k, τ);
        if z + tχ_{e_i}(E) ≤ k, then
            update z ← z + tχ_{e_i};
        end if
    end if
end for
return z
```

First, we prove the existence of lower bounds of function $g(z)-c(z)$ for the case $z(E)=k$.

Lemma 1. Let z represent the result acquired from Algorithm 1. Provided $z(E)=k$, it follows that the computation of $g(z)-c(z)$ yields a value greater than or equal to $k\tau$.

Proof. From Algorithm 2, for any element e_i in the set z, we get the following inequality:

$$g(t_i\chi_{e_i} \mid z_{i-1})-sc(t_i\chi_{e_i})\geq t_i\tau .$$

By summing up the above inequalities for each element e_i in the set z, we can get the following conclusion:

$$\sum_{e_i\in\{z\}}[g(t_i\chi_{e_i} \mid z_{i-1})-sc(t_i\chi_{e_i})]\geq \sum_{e_i\in\{z\}} t_i\tau=\tau z(E).$$

By using the submodularity of function g and the linearity of function c, we can change the above inequalities as follows:

$$g(z)-sc(z)\geq \tau z(E)=\tau k .$$

Since $s\geq 1$, we have

$$g(z)-c(z)\geq g(z)-sc(z)\geq \tau k ,$$

which completes the proof.

Algorithm 2 BS Algorithm (g,c,z,b,e,k,τ)

Input: Data stream E, $e\in E$, submodular function $g\in G_b$, $c\in C_b, z\in\mathbb{N}^E$, and $\tau\in\mathbb{R}_+, s\geq 1$.

Output: t

$t_m\leftarrow 1$;

$t_n\leftarrow \min\{b(e)-z(e),k-z(e)\}$;

if $g(\chi_e \mid z)-sc(\chi_e)<\tau$ **then**

    return 0.

end if

if $\frac{g(t_n\chi_e \mid z)-sc(t_n\chi_e)}{t_n}\geq\tau$ **then**

    return t_n

end if

while $t_m<t_n+1$, **do**

    $a=\left\lfloor\frac{t_n+t_m}{2}\right\rfloor$

    if $\frac{g(t_n\chi_e \mid z)-sc(t_n\chi_e)}{t_n}\geq\tau$ **then**

        $t_m=a$,

    else

        $t_n=a$,

    end if

end while

return t_m

Further, we prove the existence of lower bounds of function $g(z)-c(z)$ for the case $z(E)< k$.

Lemma 2. *Suppose z is the output solution of Algorithm 1, and the condition $z(E)< k$. Then, the function $g(z)-c(z)$ satisfies the following inequalities:*

$$g(z)-c(z)\geq(1-\frac{1}{s})g(z^*)-(s-1)c(z^*)+k\tau[\frac{1}{s}\nu-(1-\frac{1}{s})\mu]$$

where $\mu,\nu:0<\mu,\nu\leq 1,s\geq 1.$

Proof. We denote $\{\overline{z}\}=\{z^*\}\setminus\{z\},b=\overline{z}(E)$. Let z_{i-1} be the corresponding output solution for each arrived element $e_i(i\in\{1,...,b\})$, in which χ_{e_i} has not been added to z.

Now, we will complete the proof of this lemma in two steps.

Step 1: We will prove that the function $g(z)$ has a lower bound and $c(z)$ has an upper bound. First, we prove the existence of a lower bound of function $g(z)$. From Algorithm 2, for any element $e_i\in\{\overline{z}\}$, we obtain

$$g(\chi_{e_i}\mid z_{i-1})-sc(\chi_{e_i})\geq\tau.$$

By summing up the above inequalities for each element e_i in the set $\{\overline{z}\}$, we obtain the following conclusion:

$$\sum_{e_i\in\{z\}}g(\chi_{e_i}\mid z_{i-1})-s\sum_{e_i\in\{z\}}c(\chi_{e_i})\geq\sum_{e_i\in\{z\}}\tau=\tau\left|\{\overline{z}\}\right| \tag{2}$$

With the submodularity of the function g, we get

$$\begin{aligned}&\sum_{e_i\in\{\overline{z}\}}g(\chi_{e_i}\mid z_{i-1})\\ \geq&\sum_{e_i\in\{\overline{z}\}}g(\chi_{e_i}\mid z)\\ \geq& g(\overline{z}\mid z)\\ \geq& g(z^*+z)-g(z).\end{aligned}$$

Since function c is a nonnegative linear function, we get

$$\sum_{e_i\in\{\overline{z}\}}c(\chi_{e_i})=c(\overline{z})\leq c(x^*)$$

From the above two inequalities and the monotonicity of function g, we can get the following inequalities:

$$g(z^*)-g(z)-sc(z^*)\leq\tau\mid\overline{z}(E)\mid,$$

that is,

$$g(z)\geq g(z^*)-sc(z^*)-\tau\mid\overline{z}(E)\mid \tag{3}$$

Second, we show that the function $c(z)$ has an upper bound associated with function $g(z)$. According to the running process of the algorithm, we get that all the elements in the output solution z satisfy the following inequalities:

$$g(t_i\chi_{e_i} \mid z_{i-1}) - sc(t_i\chi_{e_i}) \geq t_i\tau .$$

By summing up the above inequalities for each element e_i in the set $\{z\}$, we have

$$\sum_{e_i\in\{z\}} g(t_i\chi_{e_i} \mid z_{i-1}) - s\sum_{e_i\in\{z\}} c(t_i\chi_{e_i}) \geq \sum_{e_i\in\{z\}} t_i\tau = \tau z(E).$$

Since $g(z)$ is a monotonic submodular function and c is a linear function, we can obtain the upper bound $c(z)$ that is associated with function $g(z)$, that is,

$$c(z) \leq \frac{1}{s}g(z) - \frac{\tau}{s}z(E). \tag{4}$$

Step 2: We prove the existence of a lower bound of function $g(z)-c(z)$. According to inequality (3) and inequality (4), we get

$$\begin{aligned} & g(z)-c(z) \\ & \geq g(z) - \frac{1}{s}g(z) + \frac{\tau}{s}z(E) \\ & \geq \frac{s-1}{s}g(z^*) - (s-1)c(z*) - \frac{s-1}{s}\overline{z}(E)\tau + \frac{\tau}{s}z(E) \end{aligned} \tag{5}$$

For the remaining part of the proof, we approach this by introducing two parameters, μ (where $0 \leq \mu \leq 1$) and ν (where $0 \leq \nu \leq 1$), that are reliant on either the ultimate output set or the present set. Without loss of generality, we can assume that $\overline{z}(E) = \mu k$ and $z(E) = \nu k$. In the algorithm analysis below, we use only *mu* and *nu* to represent multiple factors, so they need to be consistent with the other parameters in the current analysis. Then, we can translate the inequality (5) into the following form:

$$\begin{aligned} & g(z)-c(z) \\ & \geq (1-\frac{1}{s})g(z^*) - (s-1)c(z^*) - (1-\frac{1}{s})\mu k\tau + \frac{\tau}{s}\nu k \\ & = (1-\frac{1}{s})g(z^*) - (s-1)c(z^*) + k\tau[\frac{1}{s}\nu - (1-\frac{1}{s})\mu]. \end{aligned}$$

Therefore, we get

$$g(z)-c(z) \geq (1-\frac{1}{s})g(z^*) - (s-1)c(z^*) + k\tau[\frac{1}{s}\nu - (1-\frac{1}{s})\mu]$$

for $0 < \mu, \nu \leq 1, s \geq 1$, which completes the proof.

Theorem 1. *Suppose* $s = \dfrac{2+\mu+\sqrt{\delta}}{2}, \delta = \mu^2 + 4 - 4\nu, 0 < \mu, \nu \leq 1, \rho = \dfrac{(s-1)}{s+\mu(s-1)-\nu}$, *then Algorithm 1's bicriteria ratio is* $(\rho, \rho \cdot s)$.

Proof. For the case of $z(E) = k$, Lemma 1 shows that

$$g(z)-c(z) \geq k\tau .$$

For the case of $z(E) < k$, Lemma 2 shows that

$$g(z) - c(z) \geq (1 - \frac{1}{s}) g(z^*) - (s-1)c(z^*) + k\tau[\frac{1}{s}\nu - (1 - \frac{1}{s})\mu]$$

for any $0 < \mu, \nu \leq 1, s \geq 1$. So, the value of the final objective function is the minimum of $k\tau$ and $(1 - \frac{1}{s}) g(z^*) - (s-1)c(z^*) + k\tau[\frac{1}{s}\nu - (1 - \frac{1}{s})\mu]$. Let

$$k\tau = (1 - \frac{1}{s}) g(z^*) - (s-1)c(z^*) + k\tau[\frac{1}{s}\nu - (1 - \frac{1}{s})\mu],$$

we have

$$k\tau = \rho g(z^*) - \rho \cdot s \cdot c(z^*).$$

We choose the value of satisfying

$$\rho \cdot s = 1,$$

that is, by choosing the value of s, we can make the coefficient of $c(z^*)$ equal to 1. From the above equation, we can calculate the value of s

$$s_1 = \frac{2 + \mu + \sqrt{\delta}}{2} > 1 + \mu,$$

and

$$s_2 = \frac{2 + \mu - \sqrt{\delta}}{2} < 1 + \mu,$$

where $\delta = (2+\mu)^2 - 4(\mu + \nu) = 4 + \mu^2 - 4\nu$ and $\delta > 0, \sqrt{\delta} > \mu$ under the assumption $0 \leq \mu \leq 1$ and $0 \leq \nu \leq 1$.

From the range of s, we can get the value of s, which is $s = s_1$. Therefore,

$$k\tau = \rho g(z^*) - \rho \cdot s \cdot c(z^*),$$

and the threshold value

$$\tau = \frac{1}{k} \rho g(z^*) - \frac{1}{k} \rho \cdot s \cdot c(z^*),$$

where

$$s = \frac{2 + \mu + \sqrt{\delta}}{2} > 1 + \mu,$$

$$\delta = 4 + \mu^2 - 4\nu,$$

$$0 \leq \mu \leq 1,\ 0 \leq \nu \leq 1.$$

So, we get the bicriteria ratio as $(\rho, \rho \cdot s)$.

From our earlier discussion on the above theorem, it is evident that

$$\tau = \rho g(z^*) - \frac{1}{k}\rho \cdot s \cdot c(z^*).$$

So, we need to calculate the optimal value of function

$$f(z) = \rho g(z) - \rho \cdot s \cdot c(z).$$

In order to estimate the value $f(z^*)$, we opt for taking the averaged maximum singleton value m instead, where $m = \max_{e \in E} f(x_e)$, an approach inspired by the concepts proposed by Ene [5]. Also, form their discussion above the OPT, we can ensure that $m \le OPT \le km$. It takes $O(log\ k/\varepsilon)$ values falling in $[m, km]$ to get a $(1+\varepsilon)$ - approximation to OPT. Unfortunately, with this approach, we need to read the whole stream data two passes. In order to reduce the number of data reads to one, we set m equal to the maximum singleton value among all the elements that have arrived. When different thresholds are selected, we can parallel-run the original algorithm with a run count of $O(\log\ k/\varepsilon)$. Therefore, the total memory is $O(k \log\ k/\varepsilon)$.

In addition, we discuss the query number per element. During the run of Algorithm 1, we need to call Algorithm 2 to determine the elements retained in storage and the level of each preserved element, that is, any l should satisfy both $\frac{f(l\chi_e \mid z)}{l} \ge \tau$ and $f(\chi_e \mid z + l\chi_e) < \tau$. In order to obtain a valid level l, we need to query function f with $O(log\ k)$ times in Algorithm 2. So, we can obtain that the query number per element is $O(log\ k)$.

3.2. Streaming algorithm for $\alpha G - C$

In numerous real-world applications, the marginal gain of adding an element to a set is not necessarily non-increasing at all times (the core property of submodularity); instead, it may fluctuate slightly within a limited range. The weak submodularity assumption is an extension of the classical submodular property. It allows for a certain degree of supermodularity, making it more suitable for real-world scenarios since pure submodularity is often an oversimplification in many practical problems.

In this section, we shall discuss the case where the first function g in the objective function does not satisfy submodularity. We assume that the first function g in the objective satisfies α -weakly submodular, nonnegative, and monotone. The second function c in the objective satisfies nonnegative linearity. The problem is described as follows:

$$\begin{aligned} \text{maximize} \quad & g(z) - c(z), \\ \text{s.t.} \quad & z \le b, \\ & z(E) \le k, \end{aligned} \tag{6}$$

We first define α -weakly submodular function in the integer lattice.

In Algorithm 3, the first step involves setting a specific threshold value for the element present in the data stream. This becomes our preliminary standard when determining whether these data points should be incorporated into the current solution set. However, this decision-making process is not solely reliant on the set threshold. We also refer to Algorithm 4, which helps us finalize the exact level (denoted as l here) of the current element that should be included in the solution. To better understand this, consider reaching an element in the stream in which the value $g(\chi_e \mid z) - (1 + \alpha)c(\chi_e)$ does not exceed the previously established threshold. Under such circumstances, we follow through with the decision to discard that element, choosing not to maintain it within our current solution. The rationale behind this decision arises from the fact that this element does not meet the defined threshold condition, indicating that its contribution to resolving the problem is likely insignificant or counterproductive. Conversely, if the same evaluation results in a value exceeding the threshold, the element is deemed essential and retained within the solution. When it comes to determining the number of the element that can be kept, we rely on inequality

$$g(l_n\chi_e \mid z) - (1 + \alpha)c(l_n\chi_e) \geq \tau_\alpha .$$

The intricacies of these mechanisms and steps are better explained and extensively covered within the respective descriptions of Algorithm 3 and Algorithm 4.

The analytical approach is similar to the proof methodology detailed in Section 3.1. To make explanations clearer and easier to understand, we have based our discussions on several assumptions. First, we define z^* to be an optimal solution, with z representing the final solution vector generated by Algorithm 3. Subsequently, from the vectors mentioned earlier, we identify two key values: p and q. Here, p equals the value of $z^*(E)$, while q corresponds to the value of $z(E)$. Following this, we assume z_{i-1} represents the set when element e_i arrives, for each instance of i ranging from 1 to q, $z_0 = 0, z_q = z$. Within the context of this discussion, we mainly examine two special scenarios; when $z(E)$ equals k, our discussion primarily revolves around Lemma 3, and when $z(E)$ is less than k, our discussion is included in Lemma 4.

Algorithm 3 Streaming algorithm for $\alpha G - C$ on the integer lattice

Input: Data stream E, function g and c, cardinality constraint $k \in \mathbb{N}_+$, threshold value $\tau_{\alpha,} 0 < \alpha \leq 1$.

Output: a vector $z \in \mathbb{N}^E$.

```
z ← 0;
for e ∈ E do
    if z(E) < k, then
        find the level l with BS Algorithm (g, c, z, b, e, k, τ_α)
        if z + lχ_{e_i}(E) ≤ k, then
            update z ← z + lχ_{e_i};
        end if
    end if
end for
return z
```

Definition 2. [21] *For any* $s,t \in B_c$ *with* $s \le t$ *and* $e \in E$, *the* α*-weakly submodular function* g *in* G_c *is the maximum scalar* α_g *such that*

$$\alpha_g g(\chi_e \mid t) \le g(\chi_e \mid s),$$

where $t + \chi_e \in B_c$.

Algorithm 4 BS algorithm $(g, c, z, b, e, k, \tau_\alpha)$

Input: Data stream E, $e \in E$, function g and c, $z \in \mathbb{N}^E$, and $\tau_\alpha \in \mathbb{R}_+$.

Output: l.

$l_m \leftarrow 1$;

$l_n \leftarrow \min\{b(e) - z(e), k - z(e)\}$;

if $g(\chi_e \mid z) - (1 + \alpha)c(\chi_e) < \tau_\alpha$ **then**

    return 0.

end if

if $\dfrac{g(l_n\chi_e \mid z) - (1 + \alpha)c(l_n\chi_e)}{l_n} \ge \tau_\alpha$

    return l_n

end if

while $l_m < l_n + 1$, **do**

    $a = \left\lfloor \dfrac{l_m + l_n}{2} \right\rfloor$

    if $\dfrac{g(l_n\chi_e \mid z) - (1 + \alpha)c(l_n\chi_e)}{l_n} \ge \tau_\alpha$ **then**

        $l_m = a$,

    else

        $l_n = a$,

    end if

end while

return l_m

Lemma 3. *Assume* z *is the solution output by Algorithm 3, when* $z(E) = k$, *then the lower bound offunction* $g(z) - c(z)$ *is* $k\tau_\alpha$.

Proof. For each element selected into the output solution, according to the selection rule in Algorithm 3, the following inequality must be satisfied:

$$\frac{g(l_i\chi_{e_i} \mid z_{i-1}) - (1 + \alpha)c(l_i\chi_{e_i})}{l_i} \ge \tau_\alpha$$

For all elements selected into the output solution z, by adding the inequalities they satisfy, we can obtain the following comprehensive inequality:

$$\sum_{e_i \in \{z\}} g(l_i\chi_{e_i} \mid z_{i-1}) - (1 + \alpha) \sum_{e_i \in \{z\}} c(l_i\chi_{e_i}) \ge \sum_{e_i \in \{z\}} \tau_\alpha = l_i\tau_\alpha z(E)$$

.

Rearranging the inequality, we can deduce the following inequality:

$$g(z)-(1+\alpha)c(z)\geq\tau_{\alpha}z(E)=\tau_{\alpha}k.$$

Since $1+\alpha\geq 1$, we can draw the following conclusions:

$$g(z)-c(z)\geq g(z)-(1+\alpha)c(z)=k\tau_{\alpha}.$$

The proof of the conclusion of Lemma 3 is completed.

Lemma 4. *Suppose z is the output solution of Algorithm 3 and satisfying $z(E)<k$, then we have*

$$g(z)-c(z)\geq\frac{\alpha}{1+\alpha}g(z^{*})-c(z^{*})+\frac{\nu-\mu}{1+\alpha}k\tau_{\alpha},$$

where $0\leq\mu\leq 1,\ 0\leq\nu\leq 1,\ \alpha\geq 0.$

Proof. Suppose $\{\bar{z}\}=\{z^{*}\}\setminus\{z\}, l=\bar{z}(E)$. Let $z_{i-1}(i\in\{1,...,l\})$ be the corresponding output solution when the current element e_i arrives but χ_{e_i} is not added to the output solution. Next, we will separately prove the lower bound of $g(\bar{z})$ and the upper bound of $c(z)$.

According to the rule of selecting elements in Algorithm 3, for each element $e_i\in\{\bar{z}\}$, we have

$$g(\chi_{e_i}\mid z_{i-1})-(1+\alpha)c(\chi_{e_i})<\tau_{\alpha}.$$

For all elements selected into the output solution $e_i\in\{\bar{z}\}$, by adding the inequalities they satisfy, we can obtain the following comprehensive inequality:

$$\sum_{e_i\in\{z\}}[g(\chi_{e_i}\mid z_{i-1})-(1+\alpha)c(\chi_{e_i})]<\sum_{e_i\in\{z\}}\tau_{\alpha}=\tau_{\alpha}\left|\{\bar{z}\}\right| \tag{7}$$

Further, from the definition of weakly submodular (Definition 2), we can derive

$$\sum_{e_i\in\{\bar{z}\}}g(\chi_{e_i}\mid z_{i-1})\geq\alpha g(\bar{z}\mid z)\geq\alpha g(\bar{z}+z)-g(z).$$

Consider the monotonicity of function g, and we have

$$g(z^{*})-g(z)\leq g(z^{*}+z)-g(z)\ =\ g(\bar{z}+z)-g(z).$$

Since function c is nonnegative linear, we get

$$\sum_{e_i\in\{\bar{z}\}}c(\chi_{e_i})=c(\bar{z})\leq c(z^{*}).$$

Considering the inequalities satisfied by $\sum_{e_i\in\{\bar{z}\}}g(\chi_{e_i}\mid z_{i-1})$ and $\sum_{e_i\in\{\bar{z}\}}c(\chi_{e_i})$ together, we can rewrite inequality (7) as follows:

$$\alpha[g(z^{*})-g(z)]-(1+\alpha)c(z^{*})\leq\tau_{\alpha}\left|\bar{z}(E)\right|,$$

which can be formulated as

$$g\left(z\right) \geq g(z^*) - \frac{1+\alpha}{\alpha} c(z^*) - \frac{\tau_\alpha \left|\overline{z}(E)\right|}{\alpha}.$$

Now, we are in the position to prove the upper bound of $c(z)$.

According to Algorithm 3, all the elements in the output solution z satisfy the following inequalities:

$$g(l_i \chi_{e_i} \mid z_{i-1}) - \left(1 + \alpha\right) c(l_i \chi_{e_i}) \geq l_i \tau_\alpha.$$

Summing up all the above inequalities, we get

$$\sum_{e_i \in \{z\}} g(l_i \chi_{e_i} \mid z_{i-1}) - \left(1 + \alpha\right) \sum_{e_i \in \{z\}} c(l_i \chi_{e_i}) \geq \sum_{e_i \in \{z\}} l_i \tau_\alpha = \tau_\alpha z(E).$$

Given the monotonic and submodular nature of function g, and the linear aspect of function c, we can transform the inequality as follows:

$$c(z) \leq \frac{1}{1+\alpha} g(z) - \frac{\tau_\alpha}{1+\alpha} z(E) \cdot$$

So, we can get the bound of the value of objective function

$$\begin{aligned} & g(z) - c(z) \\ & \geq g(z) - \frac{1}{1+\alpha} g(z) + \frac{\tau_\alpha}{1+\alpha} z(E) \\ & = \frac{\alpha}{1+\alpha} g(z) + \frac{\tau_\alpha}{1+\alpha} z(E) \\ & \geq \frac{\alpha}{1+\alpha} [g(z^*) - \frac{\alpha}{1+\alpha} c(z^*) - \frac{\tau_\alpha}{\alpha} \overline{z}(E)] + \frac{\tau_\alpha}{1+\alpha} z(E) \\ & = \frac{\alpha}{1+\alpha} g(z^*) - c(z^*) + \frac{z(E) - \overline{z}(E)}{1+\alpha} z(E). \end{aligned}$$

Similar to Lemma 2, we also define parameters $\overline{z}\left(E\right) = \mu k$ and $z\left(E\right) = \nu k$ by introducing parameters μ and ν with the range of values for μ and ν: $0 \leq \mu \leq 1, 0 \leq \nu \leq 1$. Then we have

$$\begin{aligned} & g(z) - c(z) \\ & \geq \frac{\alpha}{1+\alpha} g(z^*) - c(z^*) - \frac{k\nu - k\mu}{1+\alpha} \tau_\alpha \\ & = \frac{\alpha}{1+\alpha} g(z^*) - c(z^*) + \frac{\nu - \mu}{1+\alpha} k \tau_\alpha \cdot \end{aligned}$$

So, the output solution, denoted as z, in Lemma 4 satisfies

$$g(z) - c(z) \geq \frac{\alpha}{1+\alpha} g(z^*) - c(z^*) + \frac{\nu - \mu}{1+\alpha} k \tau_\alpha$$

for $0 \leq \mu \leq 1, 0 \leq \nu \leq 1, t \geq 1$.
The proof of the conclusion of Lemma 4 is completed.

Theorem 2. Assume $0<\alpha\le 1, 0\le\mu\le 1, 0\le\nu\le 1$, then we have that the bicriteria ratio for Algorithm 3 is $(\frac{\alpha}{1+\alpha+\mu-\nu}, \frac{1+\alpha}{1+\alpha+\mu-\nu})$.

Proof. Based on the cardinality of z, we start to prove the conclusion. Lemma 3 and Lemma 4 show that for any $0\le\mu\le 1, 0\le\nu\le 1, t\ge 1$, the objective value satisfies

$$g(z)-c(z)\ge k\tau_\alpha \ , z(E)=k,$$

and

$$g(z)-c(z)\ge \frac{\alpha}{1+\alpha}g(z^*)-c(z^*)+k\tau_\alpha(\frac{\nu-\mu}{1+\alpha}) \ , z(E)<k$$

Therefore, the value of the final objective function is minimum of $k\tau_\alpha$, and

$$\frac{\alpha}{1+\alpha}g(z^*)-c(z^*)+k\tau_\alpha(\frac{\nu-\mu}{1+\alpha}).$$

Denote $k\tau_\alpha=\frac{\alpha}{1+\alpha}g(z^*)-c(z^*)+k\tau_\alpha(\frac{\nu-\mu}{1+\alpha})$,

$$k\tau_\alpha=\frac{\alpha}{1+\alpha+\mu-\nu}g(z^*)-\frac{1+\alpha}{1+\alpha+\mu-\nu}c(z^*)$$

So, the threshold value

$$\tau_\alpha=\frac{1}{k}\frac{\alpha}{1+\alpha+\mu-\nu}g(z^*)-\frac{1}{k}\frac{1+\alpha}{1+\alpha+\mu-\nu}c(z^*).$$

Therefore, the solution z output by Algorithm 3 satisfies

$$g(z)-c(z)\ge k\tau_\alpha\ge\frac{\alpha}{1+\alpha+\mu-\nu}g(z^*)-\frac{1+\alpha}{1+\alpha+\mu-\nu}c(z^*),$$

so the bicriteria ratio is $(\frac{\alpha}{1+\alpha+\mu-\nu}, \frac{1+\alpha}{1+\alpha+\mu-\nu})$,

where $0<\alpha\le 1, 0\le\mu\le 1, 0\le\nu\le 1$.

We denote

$$h_\alpha(z^*)\triangleq\frac{\alpha}{1+\alpha+\mu-\nu}g(z)-\frac{1+\alpha}{1+\alpha+\mu-\nu}c(z)$$

According to the definition of α-weakly submodularity, we can easily prove that $h_\alpha(z^*)$ also satisfies α-weakly submodularity. Using the same proof method applied in Section 3, we can obtain that $h_\alpha(z^*)$ falls in $[m, km/\alpha]$.

The memory is $O(k\log(k/\alpha)/\varepsilon)$.

4. Numerical examples

We now conduct numerical experiments to compare our algorithm with the stream greedy algorithm and validate the effectiveness of our approach. This paper considers the influence

maximization problem with weighted budget allocation under the stream model. Specifically, elements from the set of marketing channel nodes arrive sequentially in a stream, and only information about the already arrived nodes and their adjacent customer node sets is known. The budget allocated to a node must be determined before the next node arrives. After processing the entire set of marketing channel nodes once, the budget allocated to each marketing channel node is finalized, aiming to maximize the expected number of activated customers. Simultaneously, this process consumes certain resources, requiring the subtraction of a cost function; thus, the objective function is a nonnegative submodular function minus a linear function. The experimental data comes from the MovieLens movie rating data set. From this dataset, we obtain a bipartite graph $G(V1,V2,E)$. We select a subset of nodes (—$V1$—=500, —$V2$—=2000). We choose some nodes from $V1$ as initially activated seeds. Through network propagation, we aim to activate as many nodes in $V2$ as possible. We set epsilon = 0.1. For each selected seed node and its neighbors, a random number between $(0,\ 1)$ is chosen as the initial activation probability. If a node is selected multiple times, the probability of activating its neighbor for the second time is $1/5$ of the initial activation probability, and so forth. The knapsack constraint weights and the weights for the linear function are obtained by random sampling of integers between 1 and 10. The upper bounds for the integer lattice are obtained by random sampling of integers between 1 and 5. By treating the cardinality n and the knapsack capacity K as independent variables, we obtain the results of the numerical experiments.

Figure 1 shows the changes in the outputs of the two algorithms as the cardinality of the ground set increases. The graph shows that the result of Algorithm 1 is always above that of the stream greedy algorithm. Furthermore, the stream greedy algorithm is insensitive to the increase in cardinality, whereas the output of Algorithm 1 increases with the cardinality, indicating that our algorithm performs better than the stream greedy algorithm.

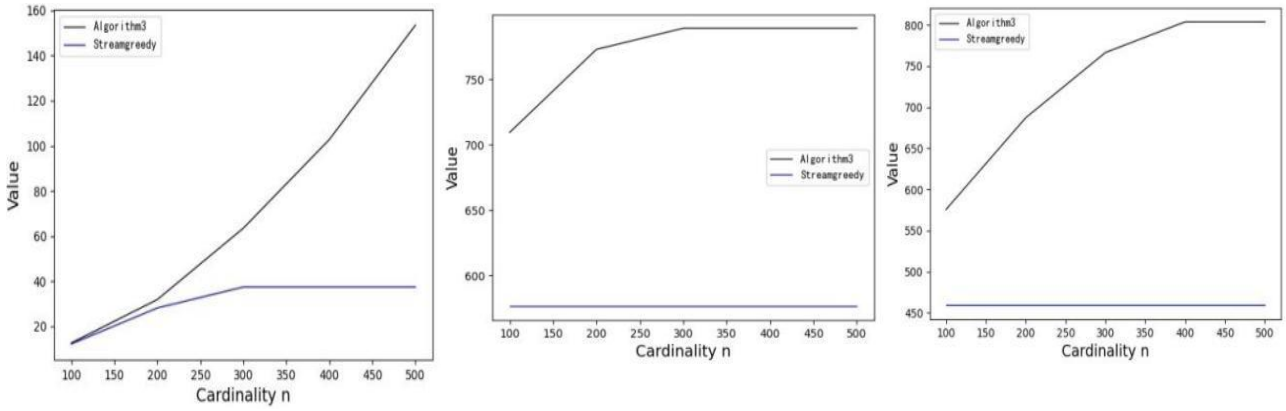

Figure 1. Solution quality when cardinality n is a variable.

5. Conclusions

In this paper, we present the development of combinatorial approximation algorithms designed to address the maximization problem of a monotone nonnegative submodular function minus a nonnegative linear function in an integer lattice. The problem is tackled under a cardinality constraint and by establishing an appropriate threshold, starting with the creation of a bicriteria streaming algorithm. Running parallel to this, we utilize the lattice binary search algorithm as a subalgorithm. Further into the subject matter, we explore scenarios where the first function of the objective function is

α-weakly submodular and a non-submodular function. For these situations, we design a combinatorial bicriteria streaming algorithm. Simultaneously, we present comprehensive analyses for both models. Features of these models include that the overall memory is while the query number per is.

Acknowledgments

The first author is supported by National Natural Science Foundation of China (No. 12301417).

Author contributions

Jingjing Tan conceived the problem and designed the algorithms. Cuiping Ge and Fengmin Wang performed theoretical analysis and proved the approximation guarantees. Ziyang Li wrote and revised the manuscript together.

Use of Generative-AI tools declaration

Generative AI tools (e.g., grammar and academic English polishing software) were solely used for language refinement and readability improvement. No AI was involved in problem formulation, combinatorial algorithm design, submodular analysis, theoretical proofs, or numerical derivation. All intellectual contents and mathematical results are fully original by the authors.

References

1. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, S. Leonardi, Online team formation in social networks, in Proceedings of the 21st International Conference on World Wide Web, (2012), 839–848. https://doi.org/10.1145/2187836.2187950
2. A. Anagnostopoulos, C. Castillo, A. Fazzone, S. Leonardi, E. Terzi, Algorithms for hiring and outsourcing in the online labor market, in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2018), 1109–1118. https://doi.org/10.1145/3219819.3220056
3. T. Lappas, K. Liu, E. Terzi, Finding a team of experts in social networks, in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2009), 467–476. https://doi.org/10.1145/1557019.1557074
4. Y. Benoist, P. Foulon, F. Labourie, Flots d'Anosov a distributions stable et instable differentiables (French) [Anosov flows with stable and unstable differentiable distributions], *J. Amer. Math. Soc.,* **5** (1992), 33–74. https://doi.org/10.1090/S0894-0347-1992-1124979-1
5. S. Nikolakaki, A. Ene, E. Terzi, An Efficient Framework for Balancing Submodularity and Cost, ArXiv preprint arXiv:2002.07782, (2020). https://doi.org/10.1145/3447548.3467367
6. M. Sviridenko, J. Vondrák, J. Ward, Optimal approximation for submodular and supermodular optimization with bounded curvature, *Math. Oper. Res.*, **42** (2017), 1197–1218. https://doi.org/10.1287/moor.2016.0842
7. D. Du, Y. Li, N. Xiu, D. Xu, Simultaneous approximation of multi-criteria submodular functions maximization, *J. Oper. Res. Soc. China*, **2** (2014), 271–290. https://doi.org/10.1007/s40305-014-0053-z

8. M. Feldman, Guess free maximization of submodular and linear sums, in Proceedings of the 16th International Conference Workshop on Algorithms and Data Structures, (2019), 380–394. https://doi.org/10.1007/978-3-030-24766-9_28
9. L. Lovasz, On the Shannon capacity of a graph, *IEEE T. Inf. Theory*, **25** (1979), 1–7. https://doi.org/10.1109/TIT.1979.1055985
10. T. Soma, Y. Yoshida, Maximization monotone submodular functions over the integer lattice, *Math. Program.*, **172** (2018), 539–563. https://doi.org/10.1007/s10107-018-1324-y
11. C. Gottschalk, B. Peis, Submodular function maximization on the bounded integer lattice, in Proceedings of WAOA, (2015), 133–144. https://doi.org/10.1007/978-3-319-28684-6_12
12. Q. Nong, J. Fang, S. Gong, D. Du, Y. Feng, X. Qu, A 1/2-approximation algorithm for maximizing a non-monotone weak-submodular function on a bounded integer lattice, *J. Comb. Optim.*, **39** (2020), 1208–1220. https://doi.org/10.1007/s10878-020-00558-4
13. T. Soma, N. Kakimura, K. Inaba, K. Kawarabayashi, Optimal budget allocation: Theoretical guarantee and efficient algorithm, in Proceedings of ICML, (2014), 351–359.
14. J. Tan, Y. Xu, D. Zhang, X. Zhang, On streaming algorithms for maximizing a supermodular function plus a MDR-submodular function on the integer lattice, *J. Comb. Optim.*, **45** (2023), 1–19. https://doi.org/10.1007/s10878-023-00986-y
15. M. Kapralov, I. Post, J. Vondrák, Online submodular welfare maximization: Greedy is optimal, in Proceedings of SODA, (2012), 1216–1225. https://doi.org/10.1137/1.9781611973105.88
16. I. Simon, N. Snavely, S. Seitz, Scene summarization for online image collections, in Proceedings of ICCV, (2007), 1–8. https://doi.org/10.1109/ICCV.2007.4408863
17. J. Tan, Y. Xu, D. Zhang, X. Zhang, On streaming algorithms for maximizing a supermodular function plus a MDR-submodular function on the integer lattice, *J. Comb. Optim.*, **45** (2023), 1–19. https://doi.org/10.1007/s10878-023-00986-y
18. A. Das, D. Kempe, Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection, in Proceedings of ICML, (2011), 1057–1064.
19. A. Bian, J. Buhmann, A. Krause, S. Tschiatschek, Guarantees for greedy maximization of non-submodular functions with applications, in Proceedings of ICML, (2017), 498–507.
20. Z. Liu, H. Chang, D. Du, X. Zhang, Improved algorithms for non-submodular function maximization problem, in Proceedings of AAIM, (2021), 190–199. https://doi.org/10.1007/978-3-030-93176-6_17
21. A. Kuhnle, J. Smith, V. Crawford, M. Yhai, Fast maximization of nonsubmodular, monotonic functions on the integer lattice, in Proceedings of ICML, (2018), 2791–2800.
22. Z. Zhang, B. Liu, Y. Wang, D. Xu, D. Zhang, Greedy algorithm for maximization of non-submodular functions subject to knapsack constraint, in Proceedings of COCOON, (2019), 651–662. https://doi.org/10.1007/978-3-030-26176-4_54
23. J. Tan, F. Wang, W. Ye, Q. Zhang, Y. Zhou, Streaming algorithms for monotone non-submodular function maximization under a knapsack constraint on the integer lattice, *Theor. Comput. Sci.*, **937** (2022), 39–49. https://doi.org/10.1016/j.tcs.2022.09.028