International Journal of Advanced and Applied Sciences, 13(3) 2026, Pages: 105-114

Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: https://www.science-gate.com/IJAAS.html

Streaming algorithm for the team formation problem on an integer lattice

Jingjing Tan [1, 2, *], Meixia Li [3], Meng Sun [1], Ruiqi Yang [3], Xiaoqing Zhang [4]

[1]School of Mathematics and Statistics, Weifang University, Weifang, Shandong 261061, China
[2]Shandong Key Laboratory of Intelligent Manufacturing Technology for Advanced Power Equipment, Weifang 261061, China
[3]School of Mathematics, Statistics and Mechanics, Beijing University of Technology, Beijing 100074, China
[4]Department of General Education, Jinan Vocational College of Nursing, Jinan, Shandong 250100, China

ARTICLE INFO

Article history:
Received 12 September 2025
Received in revised form
5 March 2026
Accepted 9 March 2026

Keywords:
DR-submodular function
Integer lattice
Team formation
Streaming algorithm
Knapsack constraint

ABSTRACT

In social networks, selecting appropriate team members is a key challenge in team formation. This problem has been proven to be NP-hard, meaning that obtaining an optimal solution within polynomial time is generally infeasible. In this study, the team formation problem is formulated as an optimization model that aims to maximize the difference between a monotone DR-submodular function and a non-negative linear function. To address this optimization problem, we propose an online bicriteria streaming algorithm that combines lattice-based binary search with thresholding techniques. Because determining the optimal threshold values in advance is difficult, we introduce Algorithm 3, which dynamically constructs estimation intervals for thresholds and iteratively approaches near-optimal decision boundaries. In addition, we examine a more general case in which the first function is relaxed from a DR-submodular function to a non-submodular function. For this scenario, a corresponding streaming algorithm is also developed. Finally, theoretical analyses of space complexity and time complexity are presented, demonstrating the scalability and computational efficiency of the proposed algorithms for large-scale data environments.

1. Introduction

With the advancement of the internet and information technology, our world has entered the era of big data. Driven by such formats, massive amounts of information are always generated, and in every field, the application of big data is crucial. It plays an important role in saving society, improving efficiency, and providing services. For example, how to build an efficient and cooperative team. Whether communication among team members is smooth and whether the allocation of roles is reasonable are important issues that we need to address. If we cannot solve these problems properly and in time, it may affect the realization of team goals, as well as the overall atmosphere of the team and the participation of members. The widespread adoption and advancement of the internet, coupled with the rapid evolution of information technology, have exerted a far-reaching and multifaceted influence on two critical aspects of organizational dynamics: the very process of team formation and the modes in which team members engage in collaborative efforts. In terms of team formation, the traditional constraints of geographical proximity and physical colocation have been significantly alleviated. Previously, teams were often limited to individuals within the same office building, city, or even country, due to the challenges of communication and coordination across distances. However, with the internet serving as a ubiquitous and instantaneous communication platform supported by tools such as video conferencing software, cloud-based project management systems, and real-time messaging applications, organizations can now assemble teams comprising members from diverse global locations. This not only expands the talent pool available for team formation, allowing companies to recruit individuals with specialized skills regardless of their physical whereabouts, but also fosters greater diversity within teams, as members bring varied cultural backgrounds, perspectives, and experiences to the table. Simultaneously, the way team members cooperate has undergone a transformative shift. In the predigital era, collaboration was heavily reliant on in-person meetings, paper-based documentation, and synchronous communication, which often

* Corresponding Author.
Email Address: tanjingjing1108@163.com (J. Tan)
https://doi.org/10.21833/ijaas.2026.03.011
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-9941-0629

restricted the flexibility of work schedules and the efficiency of information sharing. Today, information technology enables asynchronous collaboration, where team members can contribute to projects at different times, accessing and updating shared digital resources such as cloud storage files, collaborative editing documents, and virtual whiteboards at their convenience. This flexibility not only enhances work-life balance for team members but also accelerates the pace of project execution, as delays caused by scheduling conflicts are minimized. Additionally, technologies like data and artificial intelligence have further optimized collaborative processes by providing teams with actionable insights, automating repetitive tasks, and facilitating more informed decision-making, thereby elevating the overall effectiveness of teamwork. Recently, more research has focused on the issue of team formation, in the hope of building efficient working teams. Nikolakaki et al. (2021) transformed the problem of selecting team members during the team building process into the following maximization problem, which is described as follows:

$$\max_{\{X \subset E\}} u(X)\text{-}v(X),$$

where, the non-negative submodular function $u(X)$ describes the nonlinear increase of benefits (diminishing marginal returns), and the linear function $\mathrm{v}(X)$ describes the linear increase in costs. Maximizing the difference between the two corresponds to the optimal team resource allocation strategy of obtaining maximum benefits at minimum cost. By maximizing the above problem, the optimal strategy for team member selection is obtained. Du et al. (2014) proposed a two-criterion approximation algorithm.

Sviridenko et al. (2017) gave an algorithm with an approximation guarantee for this problem that is best possible in certain intuitive and formal senses. However, this algorithm involves a guess scheme that makes it somewhat messy and significantly hurts its time complexity. Feldman (2021) studied the problem of maximizing the sum of a monotone submodular function and a linear function subject to a matroid constraint. By using a novel weighting technique, they design an elegant alternative algorithm that avoids the problematic guessing step, improving the time complexity in the algorithm designed by Sviridenko et al. (2017) for such problems while keeping the same approximation guarantee. Under the streaming model, Wang et al. (2022) studied the same problem with a cardinality constraint. Using the extended threshold greedy algorithm, a bicriteria streaming algorithm was proposed. Meanwhile, the case where the first function is non-submodular was also considered. There have been many studies on this issue (Lappas et al., 2009; Wang et al., 2021; Anagnostopoulos et al., 2012; Anagnostopoulos et al., 2018; Byrnes, 2015). The integer lattice is an algebraic structure of great importance in mathematics, defined as follows:

Let n be a positive integer. $\mathbb{Z}^n$ denotes the set of n-dimensional integer vectors, i.e.,

$$\mathbb{Z}^n = \{(x_1, x_2, \ldots, x_n) | x_i \in \mathbb{Z}^n, i = 1,2,\ldots,n\}$$

$\mathbb{Z}^n$ together with vector addition and scalar multiplication operations, forms a group, which is called the n-dimensional integer lattice. When studying problems on integer lattices, the following challenges often arise: First, discreteness: Decision variables take values at discrete integer points. Unlike optimization problems in continuous domains, classical optimization methods based on continuous mathematical tools (such as derivatives) cannot be directly applied. Second, strong combinatorial nature: Submodular optimization problems on integer lattices are inherently highly combinatorial.

The solution space consists of combinatorial structures formed by discrete integer points, requiring specialized combinatorial optimization algorithms for solving. Finally, high computational complexity: Due to the discreteness and combinatorial nature of the solution space, solving submodular optimization problems on integer lattices typically involves higher computational complexity than those in continuous or simple discrete domains. Lovász's (1979) research on lattices has promoted the development of lattice theory. The concept of lattice theory and optimization was first put forward by Lovász (1979). His work laid the foundation for subsequent research in this field. There is a wealth of remarkable results, see Soma and Yoshida (2018), Sviridenko et al. (2017), Sarpatwar et al. (2019), and Kapralov et al. (2013).

Let $\mathbb{N}^E$ stand for the integer lattice constructed on E, which is a finite set with n elements. For each element $e_i \in E$, we take χ_{e_i} as the i-th unit vector. For any x, y such that $x \leq y$, when the subsequent inequality holds:

$$u(y + \chi_{e_i}) - u(y) \leq u(x + \chi_{e_i}) - u(x)$$

we call u is a DR-submodular function. For DR-submodular optimization problems with cardinality constraints, knapsack constraints, and matroid constraints, Soma and Yoshida (2018) designed a pseudo-polynomial time algorithm with an approximation ratio of $1 - 1/e$ has been developed. By leveraging the bidirectional greedy technique, Gottschalk and Peis (2015) designed a 1/3-approximate algorithm. Nong et al. (2020) further improved the approximate ratio to 1/2. Tan et al. (2023) proposed a 1/3 - approximate online threshold stream algorithm.

Despite the growing attention to team building optimization, existing studies suffer from a critical research gap: they fail to adequately model the benefit-cost trade-off under streaming data scenarios and knapsack constraints on integer lattices.

Most prior works either overemphasize collaborative benefit maximization while neglecting linear resource costs (e.g., human capital, temporal investments) or focus on static optimization frameworks that cannot adapt to real data streams, limiting their applicability to dynamic team formation contexts where resources are constrained and information arrives sequentially. Moreover, few studies have leveraged submodular function theory to capture the cumulative synergy and diminishing marginal returns inherent in team interactions, especially under the combined constraints of streaming data and knapsack resource limits.

To address this gap, this paper addresses the team formation problem on integer lattices subject to knapsack constraints (e.g., finite resource budgets), presenting a streaming algorithm designed to achieve an optimal tradeoff between cumulative gains and incurred costs. Its core can be described as: under resource constraints (knapsack constraints), by optimizing variables (taking values on the integer lattice), maximizing the difference between the submodular function and the linear function to achieve the optimal balance between team benefits and costs. The problem is described as follows: Let x be an n-dimensional vector taken from $\mathbb{N}^E$. Consider u, a normalized submodular function defined on $\mathbb{N}^E$. This function u satisfies the properties of nonnegativity and monotonicity. Additionally, v represents a linear function, also defined on $\mathbb{N}^E$, maximize $u(x) - v(x)$

$$\text{s.t. } x \leq b \qquad (1)$$
$$\omega^T x \leq K.$$

where, $B = \{x \in \mathbb{N}^E : x \leq b\}$ forms a box within the set $\{\mathbb{N} \cup \{+\infty\}\}^E$, The variable $K \in \mathbb{N}$ which is a natural number, represents the total budget. In the context of the knapsack problem, $\omega(e)$ indicates the cost of element e, and $\omega^T x$ is calculated as the sum $= \sum_{e \in \{x\}} \omega(e)$ (Tan et al., 2024). This paper makes three core contributions:

1. Novel Problem Formulation: We propose a DR-submodular minus linear function framework to model the benefit-cost balance in team building, where the DR- submodular component captures collaborative gains and the linear term quantifies resource expenditures, filling the void of integrated modeling for trade-off under knapsack constraints.
2. Efficient Algorithm Design: We develop two bicriteria streaming algorithms (Algorithm 1 and Algorithm 3) for the constrained maximization problem on integer lattices. Algorithm 3 further addresses the practical challenge of unknown threshold by constructing a data-driven estimation interval, enhancing real-world applicability. The core reason for our adoption of the bicriteria approach lies in the fact that the problem addressed in this study involves two mutually conflicting yet indispensable core requirements, such as solution efficiency and solution optimality, or resource consumption and task completion quality. Single criterion methods typically can only prioritize one of these requirements, leading to a significant degradation in the performance of the other (e.g., excessively long runtime when pursuing the optimal solution, or a substantial compromise in solution quality when emphasizing efficiency). The bicriteria approach, by constructing a multi-objective optimization model, can find the Pareto optimal tradeoff between the two conflicting objectives. It not only avoids the limitations of single-criterion methods but also adapts to practical scenarios where balanced performance under multiple constraints is required (e.g., in large-scale data processing, there is a need to output results quickly while ensuring the practicality of the solution). This choice is not a simple superposition of objectives, but an inevitable design based on the essence of the problem, ensuring that the algorithm possesses stronger applicability and flexibility in real-world scenarios.
3. Rigorous Theoretical Analysis: We provide comprehensive performance guarantees for the proposed algorithms, deriving tight bounds on their time complexity and space complexity, which validate their efficiency in streaming scenarios.

This work extends our preliminary conference paper (Tan et al., 2024), which first explored the maximization of a DR-submodular minus linear function for team building benefit-cost optimization under knapsack constraints on integer lattices. While the conference version proposed a basic bicriteria streaming algorithm (Algorithm 1) with an assumed known threshold, this manuscript significantly advances the research by addressing critical limitations and expanding the scope of investigation. Specifically, the key extensions and enhancements are threefold: first, we develop a novel threshold adaptive streaming algorithm (Algorithm 3) to tackle the practical challenge of unknown thresholds, which was not considered in the conference work; second, we extend the problem setting to non-submodular function scenarios and design a corresponding streaming algorithm, filling the gap of handling non-submodular benefit structures; third, we conduct more comprehensive theoretical analysis and performance validation, including refined complexity derivations and additional robustness checks that were not included in the preliminary conference study. These extensions collectively improve the practical applicability and theoretical depth of the original framework, making it more suitable for real-world dynamic team-building contexts.

2. Preliminaries

In this section, we denote the set of natural numbers ranging from 1 to K as $[K]$. For a vector x its component at the coordinate $e_i \in E$ is represented by. The null vector is denoted by 0. We define χ_{e_i} as the standard basis vector, which has all zero components except for the *i-th* component, which is equal to 1. For a subset $X \subseteq E$, its characteristic vector is denoted by χ_X. The sum of $x(e_i)$ over all elements e_i in X is denoted by $x(X)$. We use x to represent a multi-set where each element e appears $x(e)$ times, and the cardinality of this multi-set is defined as $|\{x\}| := x(E)$ which represents the total sum of $x(e)$ over all $e \in E$. Within the realm of vector operations, we explicitly define the component wise maximum and component-wise minimum of two vectors x and y, denoting these two operations as $x \vee y$ and $x \wedge y$ respectively. Specifically, the component-wise maximum $x \vee y$ refers to a vector formed by comparing the elements at corresponding positions in vectors x and y one by one and taking the larger value among them as the component at the corresponding position of the new vector. In contrast, the component-wise minimum $x \wedge y$ is a vector constructed by comparing the corresponding components of vectors x and y and selecting the smaller value as the component at the corresponding position of the new vector. For multi-sets, the difference between $\{x\}$ and $\{y\}$ is denoted by $\{x\} \setminus \{y\}$ which is formally defined as $\{(x \setminus y) \vee 0\}$. This operation ensures that the resulting multi-set contains only non-negative components, obtained by taking the component-wise maximum of the vector difference with the zero vector.

Consider a function $u : \mathbb{N}^E \to \mathbb{R}_+$,

Monotonicity: for any vectors $x \leq y$, if inequality $u(x) \leq u(y)$ holds, then u is said to be monotone non-decreasing.

Nonnegativity and Normalization: for all $x \in \mathbb{N}^E$, if $u(x) \geq 0$ and $u(0) = 0$, then u is called nonnegative and normalized.

We define $u(x|y)$ as the marginal increase of a vector x with respect to y, and its specific expression is,

$$u(x|y) = u(x + y) - u(y).$$

The core function of this formula is to quantitatively describe the increment in the function value after adding vector x to the vector y. Essentially, $u(x|y)$ focuses on the marginal contribution of vector x to the overall change in the function value. It does not examine the function values of x or y in isolation; instead, it accurately captures the additional change brought about by the inclusion of x by comparing the function value of the vector y with the function value of the new vector $x + y$ formed by adding x to y .

3. Maximizing the difference between a DR-submodulal function and a linear function

In this section, we will study the algorithm design problem for Problem 1 and analyze the performance of the algorithm.

In the design process of Algorithm 1, we first construct an iterative function $f(x) = u(x) - cv(x)$. Meanwhile, assuming that the threshold (related to the optimal values of the function f and function c) can be known in advance, where the threshold is a key parameter for regulating the tradeoff between the two objectives, and its connection to the optimal solution is reflected in the fact that it defines the preference boundary of bicriteria optimization it quantifies the weight ratio of the two objectives (e.g., the degree of bias toward efficiency or optimality), narrows the search space for solutions, and ultimately screens out the subset of Pareto optimal solutions that align with this preference. As the threshold changes, the preference boundary adjusts accordingly, and the corresponding subset of optimal solutions is updated simultaneously. Under this condition, we use Algorithm 2 to find l such that the following equation holds. Let x^* denote the optimal solution to the problem and let x stands for the final output of Algorithm 1. At the initial stage of our analysis, we construct an intermediate function of the form $u(x) - cv(x)$, where the parameter c satisfies the constraint $c \geq 1$. In terms of the rationale behind this definition, the intermediate function is introduced to bridge the gap for subsequent analysis of the relationship between the optimal solution and the algorithm's output. Through Algorithm 2, we can select the largest l such that when le is added to the current solution, its marginal gain does not exceed the threshold. The specific algorithm design is shown in Algorithm 1 and Algorithm 2. In the algorithm design framework of this paper, the function u is explicitly defined as a submodular function, while the function v is classified as a linear function.

Algorithm 1. Streaming algorithm with knapsack constraint

Input: $\boldsymbol{E}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{K} \in \mathbb{N}_+, \boldsymbol{\tau}, \boldsymbol{c} \geq \mathbf{1}$.
Output: $\boldsymbol{x} \in \mathbb{N}^{\mathbf{E}}$.
$\boldsymbol{x} \leftarrow \mathbf{0}$;
$\boldsymbol{f} = \boldsymbol{u} - \boldsymbol{cv}$:
for $\boldsymbol{e} \in \boldsymbol{E}$ do:
  if $\boldsymbol{\omega}^{\boldsymbol{T}}(\boldsymbol{x}) < \boldsymbol{K}$, then
    determine the level $\boldsymbol{l}$ using the BS-Algorithm$(\boldsymbol{f}, \boldsymbol{\omega}, \boldsymbol{x}, \boldsymbol{b}, \boldsymbol{e}, \boldsymbol{K}, \boldsymbol{\tau})$;
    if $\boldsymbol{\omega}^{\boldsymbol{T}}(\boldsymbol{x} + \boldsymbol{l}_{\boldsymbol{\chi e_i}} \leq \boldsymbol{K})$, then
      update $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{l}_{\boldsymbol{\chi_{e_i}}}$
    end if
  end if
end for
return $\boldsymbol{x}$

Lemma 1: Denote $\mathrm{x_m}$ as the output of Algorithm 1 after the m−th iteration. We establish that $\mathrm{f(x_m)} \geq \tau \mathrm{x_m(E)}$, where τ is a predefined threshold (Tan et al., 2024).

Proof: Given that Equation $f(x_m) \geq \tau x_m(E)$ holds, we proceed to prove Equation $f(x_{m+1}) \geq \tau x_{m+1}(E)$ below. From Algorithm 2, we have $\frac{f(l_{m+1\chi_e}|x_m)}{l_{m+1}\omega(e)} \geq \tau$. Let $x_{m+1} = l_{m+1\chi_e} + x_m$, we have

$$
\begin{aligned}
f(x_{m+1}) &\geq l_{m+1^{\omega(e)\tau}} + f(x_m) \\
&\geq l_{m+1^{\omega(e)\tau}} + \tau x_m(E) \\
&= \tau\big(l_{m+1^{\omega(e)\tau}} + l_{m+1^{\omega(e)\tau}} + x_m(E)\big) \\
&= \tau x_{m+1}(E)
\end{aligned}
$$

We conclude that the result holds by the method of recursion.

Lemma 2: Suppose $x(E) < k$ and x be the output of Algorithm 1, we have$f(\chi_e|x) < \omega(e)\tau$, for any $e \in \{x^*\}\backslash\{x\}$ (26).

Proof: When element e arrives, let the output solution be denoted as $\tilde{\boldsymbol{x}} \leq \boldsymbol{x}$, where represents the specified order relation. Assume that l_e is the output of Algorithm 2. Regarding the value of l, we discuss it under two scenarios. The first case, if $l_e = 0$, we have $f(\chi_e|\tilde{x}) \leq \omega(e)\tau$. The second case, if $0 < l_e < b(e) - \omega^T x$, we have

$f(\chi_e|\tilde{x}) \leq l_e\omega(e)\tau$ and $f((l_e+1)\chi_e|\tilde{x}) \leq (l_e+1)\omega(e)\tau$.

Algorithm 2. BS Algorithm (*u, v, x, b, e, K, τ*)

Input: $E, e \in E, u, v, x \in \mathbb{N}^E, and\ \tau \in \mathbb{R}_+$.
Output: l
$l_m \leftarrow 1$;
$l_m \leftarrow \min\left[\frac{b(e)-\omega^T(x)}{\omega(e)}\right]$;
if $\frac{f(\chi_e|x)}{\omega(e)} < \tau$ then;
  return 0.
end if
if $\frac{f(l_n\chi_e|x)}{l_n\,\omega(e)} \geq \tau$ then
return l_n
end if
while $l_m < l_n + 1$,do
  $a = \left[\frac{l_n+l_m}{2}\right]$
  if $\frac{f(a\chi_e|x)}{a\omega(e)} \geq \tau$ then
    $l_m = a$,
  else
    $l_n = a$,
  end if
end while
return l_m

Let $x_e = \tilde{x} + l_e\chi_e \leq x$, we obtain

$$
\begin{aligned}
f(\chi_e|\tilde{x}) &= f(l_e+1)\chi_e|\tilde{x}) - \boldsymbol{f}(l_e\chi_e|\tilde{x}) \\
&< (l_e+1)\boldsymbol{\omega}(e)\tau - l_e\boldsymbol{\omega}(e)\tau \\
&= \tau\boldsymbol{\omega}(e)
\end{aligned}
$$

We now apply the DR-submodularity to the $f(\chi_e|x)$, we get

$$f(\chi_e|x) \leq f(\chi_e|x_e) < \tau\omega(e)$$

which completes the proof.

Theorem 1: Suppose $c = \frac{3+\sqrt{5}}{2}$,$\tau = \frac{(3-\sqrt{5})f(x^*)-2h(x^*)}{2K}$ Then, the output solution of Algorithm 1 satisfying $u(x) - v(x) \geq \frac{3-\sqrt{5}}{2}u(x^*) - v(x^*)$.

By the design of Algorithm 1, it holds that $x(E) \leq K$. We analyze this inequality through two scenarios:

Case 1: According to Lemma 1, we derive $f(x) \geq K\tau$. This implies the intermediate function $u(x) - cv(x) \geq K\tau$. Given $v(x) \geq 0$, *and* $c \geq 1$, we further obtain

$$u(x) - v(x) \geq u(x) - cv(x) \geq K\tau. \tag{2}$$

Case 2: When $x(E) < K$, we have

$$
\begin{aligned}
&f(x^* \vee x) - f(x) \\
&\leq \textstyle\sum_{e\in\{x^*\}\backslash\{x\}}(\chi_e|x) \\
&\leq K\tau
\end{aligned}
\tag{3}
$$

where the first inequality follows from the DR-sub modularity of function f , and the second inequality leverages the conclusion of Lemma 2. Given that $u(x^* \vee x) \geq u(x^*)$ and $v(x^* - x) \vee 0$, the following inequality holds due to the definition of the function.

$$
\begin{aligned}
&f(x^* \vee x) - f(x) \\
&= u(x^* \vee x) - cv(x^* \vee x) - (u(x) - cv(x)) \\
&\geq u(x^*) - u(x) - cv(x^*)
\end{aligned}
\tag{4}
$$

by combining Eqs. 3 and 4, we have

$$u(x^*) \geq u(x^*) - cv(x^*) - K\tau. \tag{5}$$

Furthermore, when $x(E) < K$, the result of Lemma 1 also holds. So, we get

$$u(\boldsymbol{x}) - cv(\boldsymbol{x}) \geq \boldsymbol{x}(E)\tau \geq 0 \tag{6}$$

From Eqs. 5 and 6, we obtain

$$u(\boldsymbol{x}) - v(\boldsymbol{x}) \geq \frac{c-1}{c}(u(x^*) - cv(x^*) - K\tau) \tag{7}$$

by combining Eqs. 2 and 7, we have

$$u(\boldsymbol{x}) - v(\boldsymbol{x}) \geq \min\{K\tau, \frac{c-1}{c}\ (u(x^*) - cv(x^*) - K\tau\ \}$$

so, $K\tau = \frac{c-1}{c}(u(x^*) - cv(x^*) - K\tau$,

rearranging the terms allows us to obtain the following equality

$K\tau = \frac{c-1}{2c-1}(u(x*) - cv(x*))$.
Let $\frac{c(c-1)}{2c-1} = 1$,

we get $c = \frac{3+\sqrt{5}}{2} \geq 1$.

Consequently, we may draw the conclusion that

$$u(x) - v(x) \geq \frac{3-\sqrt{5}}{2}u(x^*) - v(x^*)$$

with the threshold $\tau = \frac{1}{K}\left[\frac{3-\sqrt{5}}{2}u(x^*) - v(x^*)\right]$ of Algorithm 1.

In the design process of Algorithm 1, we assumed that the threshold value was known. However, in practical problems, it is very difficult for us to obtain this value. Therefore, when designing the algorithm, we need to improve this issue. In this paper, we can estimate the value of $\hat{\tau} = \frac{3-\sqrt{5}}{2}u(x^*) - v(x^*)$by using the following method.

For any element e in E, we can calculate the maximum value of $\hat{\tau}(\chi_e)$ and denote M as this maximum value. With the help of this maximum value, we construct the set

$$V_\varepsilon = \{(1+\varepsilon)^i : i \in \mathbb{Z}_+, \mathcal{E} \le (1+\varepsilon)^i \le KM\}.$$

Denote

$$x^* = \sum_{e\in\{x^*\}} \chi e.$$

From the DR-submodularity, we can get the following inequality

$$\hat{\tau}(x*) = \hat{\tau}(\sum_{e\in\{x*\}}) \le \sum_{e\in\{x*\}} \hat{\tau}(\chi e) \le KM.$$

In the process of designing the algorithm, we only consider the case where the threshold value is greater than 0. So, there exists a value $e > 0$ such that $\hat{\tau}(\boldsymbol{x}^*) \ge \varepsilon$, i.e., $\varepsilon \le \hat{\tau}(\boldsymbol{x}^*) \le KM$.

According to the definition of V_ε, we know that there must exist an $\bar{\iota}$ such that $\hat{\tau}(\boldsymbol{x}^*) = (1+\varepsilon)^{\bar{\iota}}$ We just need to let

$$\tilde{\iota} = [\log_{1+\varepsilon} \tilde{\iota}(\boldsymbol{x}^*)].$$

So,

$$d* = (1+\varepsilon)^i \le \hat{\tau}(\boldsymbol{x}*),$$

and

$$d^* \ge \frac{\hat{\tau}(x^*)}{1+\varepsilon} \ge \max(1-\varepsilon)\hat{\tau}(\boldsymbol{x}^*)\frac{\varepsilon}{1+\varepsilon}.$$

Therefore, there exist a $d^* \in V_\varepsilon$, such that

$$(1-\varepsilon)\tau\hat{}(x^*) \le d^* \le \tau\hat{}(x^*),$$

which shows the estimated interval of $\hat{\tau}(x^*)$.

In the design process of Algorithm 3, We continuously update the estimated interval of $\hat{\tau}(x^*)$ according to the newly arrived elements. After reading all the elements, we can obtain an accurate estimated interval. When the element e_i arrives, we denote the corresponding interval as V_ε^i. To reduce the complexity of the algorithm, we will delete the element d that is not in V_ε^i. For specific details, refer to Algorithm 3.

Theorem 2: Suppose $0 < \varepsilon < 1$. Then, the output solution of Algorithm 3 satisfying

$$u(x) - v(x) \ge \left[\frac{3-\sqrt{5}}{2} - \varepsilon\right](u(x^*) - v(x^*). \quad (8)$$

The space memory complexity of Algorithm 3 is $O(K\log K/\varepsilon)$, and the time complexity is $\log^2 O(K/\varepsilon)$.

Proof: From Theorem 1 and the definition of V_ε^n , we know there exists a $d \in V_\varepsilon^n$ such that

$$u(x^d) - v(x^d) \ge (1-\varepsilon)\hat{\tau}(x^*).$$

Algorithm 3. Streaming algorithm with knapsack constraint

Input: $E, u, v,\ K \in \mathbb{N}+, \tau,\ c \ge 1,\ 0 < \varepsilon < 1,\ \hat{\tau}(x) = \frac{1}{2}(3-\sqrt{5})u(x) - v(x)$.
Output: $x \in \mathbb{N}^E$.
1: $V = \{(1+\varepsilon)^m : m \in \mathbb{Z}_+\}$;
2: $f(x) = u(x) - \frac{3+\sqrt{5}}{2}v(x)$
3: *for each* $d \in V$, *set* $x^d = 0$
4 : $M \leftarrow 0$
5: for $i = 1,2,\dots,n$ do
6: $M \leftarrow max\{M, \widehat{\tau(\chi_{e_i})}\}$
7: $\{V_\varepsilon^i = (1+\varepsilon)^m : \varepsilon \le (1+\varepsilon)^m \le KM\}$;
8: delete all x^d, *where* $d \notin V_\varepsilon^i$;
9: for $d \in V_\varepsilon^i$ do
10: if $\omega^T(x^d) < K$, then
11: determine the level l using the BS-Algorithm$\left(f, \omega, x, b, e, K, \frac{d}{K}\right)$
12 : update $x^d \leftarrow x^d + l\chi_{e_i}$;
13: end if
14: end for
15: end for
16:return $\arg\max_{d\in V_\varepsilon^i} f(x^d)$

Suppose x is the solution output by Algorithm 3. From the proof process of Theorem 1, we obtain the following inequality

$$u(x) - v(x) \ge u(x^d) - v(x^d) \ge (1-\varepsilon)\hat{\tau}(x^*).$$

Thus, we have proved Eq. 8. Next, we discuss the complexity of Algorithm 3. First, the cardinality of V_ε^i is $O(\log K/\varepsilon)$ and the scale of the solution output by Algorithm 3 is $O(K)$, so the space complexity is $O((K\log K)/\varepsilon)$. For the time complexity, as known from Algorithm 3, the time complexity for each d is $O(\log K)$. Summing over all $d \in V_\varepsilon^i$, the time complexity is $O(log^2 K/\varepsilon)$.

4. Maximizing the difference between a non-submodular function and a linear function

In team formation problem, the consistency of team members' preferences affects decision- making efficiency, thus forming a non-submodular benefit model. For example, in a committee decision-making team, if a majority vote is required, when the number of team members supporting a proposal increases from 4 (below the majority) to 5 (reaching the majority), the marginal benefit (proposal approval) will increase sharply, reflecting the voting threshold characteristic of non-submodularity. In design teams, introducing members with significantly different creative styles may intensify opinion conflicts, causing the marginal benefit (creative output) to first rise and then fall as the team size expands. This violates the diminishing marginal returns assumption of submodularity. These practical problems all require consideration of non-submodular scenarios.

Algorithm 4. Streaming algorithm with knapsack constraint

Input: $E, u, linear\ v, K \in \mathbb{N}_+, \tau, c \geq 1$.
Output: $x \in \mathbb{N}^E$.
$x \leftarrow 0$;
$f = u - (1+\alpha)v$;
for $e \in E$ do
  if $\omega^T(X) < K$, then
    $determine\ the\ level\ l\ using\ the$ BS − Algorithm$(f, \omega, x, b, e, K, \tau_a)$;
    if $\omega^T\left(x + l_{\chi_{e_i}}\right) \leq K$, then
      update $x \leftarrow x + l_{\chi_{e_i}}$;
    $end\ if$
  $end\ if$
$end\ for$
$return\ x$

Lemma 3: Let x_m represent the output after the m-th iteration acquired from Algorithm 4, in this case, the expression $u(x_m) - (1+\alpha)v(x_m)$ produces a value that is no less than $\tau_\alpha x_m(E)$.

Proof: From Algorithm 5, we get

$\frac{u(l_{m+1}\chi_e|x_m) - (1+\alpha)v(l_{m+1}\chi_e)}{l_{m+1}\omega(e)} \geq \tau_\alpha$, Let $x_{m+1} = l_{m+1}\chi_e + x_m$,

we have

$$u(x_{m+1}) - (1+\alpha)v(x_{m+1}) \geq l_{m+1}\omega(e)\tau_\alpha + u(x_m) - (1+\alpha)v(x_m)$$
$$\geq l_{m+1}\omega(e)\tau\alpha + \tau_\alpha x_m(E)$$
$$= \tau_\alpha(l_{m+1}\omega(e) + x_m(E)$$
$$= \tau_\alpha x_{m+1}(E)$$

thus, finalizing the demonstration.

Lemma 4: Assume that $x(E) < K$ and x be the output of Algorithm 1. Then, for any $e \in \{x^*\}\backslash\{x\}$, the inequality

$$u(\chi_e|x) - (1+\alpha)v(\chi_e) < \omega(e)\tau_\alpha$$

holds.

Proof: Let $\tilde{x} \leq x$ represent the solution output when elemen e arrives. Suppose l_e is the result returned by Algorithm 5. In such a case, if $l_e = 0$, the inequality

$u(\chi_e|\tilde{x}) - (1+\alpha)v(\chi_e) \leq \omega(e)\tau_\alpha$ holds. If $0 < l_e < b(e) - \omega^T x$,

we have

$$u(l_e\chi_e|\tilde{x}) - (1+\alpha)v(l_e\chi_e) \geq l_e\omega(e)\tau_\alpha$$

and

$$u((l_e+1)\chi_e|\tilde{x}) - (1+\alpha)v((l_e+1)\chi_e) < (l_e+1)\omega(e)\tau_\alpha.$$

Let $x_e = \tilde{x} + l_e\chi_e \leq x$. Under this condition, the following relationship holds:

$$u(\chi_e|x_e) - (1+\alpha)v(\chi_e)$$
$$= u((l_e+1)\chi_e|\tilde{x}) - (1+\alpha)v((l_e+1)\chi_e) - u(l_e\chi_e|\tilde{x}) - (1+\alpha)v(l_e\chi_e)$$
$$< (l_e+1)\omega(e)\tau_\alpha - l_e\omega(e)\tau$$
$$= \tau_\alpha\omega(e)$$

By DR-submodularity, we get

$$u(\chi_e|x) - (1+\alpha)v(\chi_e) \leq u(\chi_e|x_e) - (1+\alpha)v(\chi_e) < \tau\omega(e).$$

which completes the proof.

Algorithm 5. BS Algorithm (*u, v, x, b, e, K, τ*)

Input: $E, e \in E, u, v, x \in \mathbb{N}^E, and\ \tau \in \mathbb{R}_+$.
Output: l
$l_m \leftarrow 1$;
$l_n \leftarrow min\left\lfloor\frac{b(e) - \omega^T(x)}{\omega(e)}\right\rfloor$
if $u(\chi_e|x) - (1+\alpha)v(\chi_e) < \tau_\alpha$ then
  return 0.
end if
if $\frac{u(l_n\chi_e|x - (1+\alpha)v(l_n\chi_e)}{l_n\omega(e)} \geq \tau_\alpha$ then
  return l_n
end if
while $l_m < l_n + 1$, do
  $a = \left\lfloor\frac{l_m + l_m}{2}\right\rfloor$
  if=$\frac{u(a\chi_e|x - (1+\alpha)v(a\chi_e)}{a\omega(e)}$
    $l_m = a$,
  else
    $l_n = a$,
  end if
end while
return l_m

Theorem 3: Suppose $\tau\alpha = \frac{\alpha}{2K(1+\alpha)}u(x^*) - \frac{\alpha}{2K}v(x^*)$. Then, the output solution of Algorithm 1 satisfying

$$u(x) - v(x) \geq \frac{\alpha}{2(1+\alpha)}u(x^*) - \frac{\alpha}{2}v(x^*)$$

Proof: Algorithm 4 ensures that $x(E) \leq K$. We analyze this in two scenarios. When $x(E) \leq K$, Lemma 1 implies that $u(x) - (1+\alpha)v(x) \geq K\tau_\alpha$. Given that $v(x) \geq 0$, and $\alpha > 0$, it follows that

$$u(x) - v(x) \geq u(x) - (1+\alpha)v(x) \geq K\tau_\alpha. \quad (9)$$

Second, in the scenario where $x(E) \leq K$, the following conclusion can be drawn based on the DR-submodularity of function u and Lemma 4,

$$(u(x^* \vee x) - (1+\alpha)v(x^* \vee x)) - (u(x) - (1+\alpha)v(x))$$
$$\leq \sum u(\chi_e|x) - (1+\alpha)v(\chi_e) \quad (10)$$
$$\leq K\tau_\alpha.$$

since

$$u(x^* \vee x) \geq u(x^*)$$

and

$$v(x^* - x) \vee 0,$$

we have

$$(u(x^* \vee x) - (1+\alpha)v(x^* \vee x) - (u(x) - (1+\alpha)v(x)) \geq u(x^*) - u(x) - (1+\alpha)v(x^*)$$

combining (10) and (11), we have

$$u(x) \geq u(x^*) - (1+\alpha)v(x^*) - K\tau_\alpha. \quad (11)$$

Moreover, when $x(E) < K$, the conclusion from Lemma 3 remains valid. Thus, we obtain

$$v(x) \leq \frac{1}{1+\alpha} u(x) \quad (12)$$

from Eqs. 11 and 12, we obtain

$$u(x) - v(x) \geq \frac{\alpha}{(1+\alpha)} u(x^*) - (1+\alpha)v(x^*) - K\tau_\alpha \quad (13)$$

combining Eqs. 9 and 13, we get

$$u(x) - v(x) \geq min\{K\tau_\alpha, \frac{\alpha}{(1+\alpha)} u(x^*) - (1+\alpha)v(x^*) - K\tau_\alpha\}$$

so,

$$K\tau_\alpha = \frac{\alpha}{2(1+\alpha)} u(x^*) - \frac{\alpha}{2} v(x^*).$$

Thus, the threshold value is given by $\tau_\alpha = \frac{\alpha}{2K(1+\alpha)} u(x^*) - \frac{\alpha}{2K} v(x^*)$. From this, we may draw the conclusion that. Therefore, the bicriteria ratio can be acquired as $(\alpha/2(1+\alpha), \alpha/2)$, where $0 < \alpha \leq 1$.

5. Numerical experiments

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 nodes 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 non-negative submodular function minus a linear function.

The experimental evaluation is conducted on three real-world datasets: email-enron-large, soc-Epinions1, and MovieLens. The email-enron-large dataset represents an email communication network, where nodes correspond to users and edges indicate email interactions. The soc-Epinions1 dataset is a trust network, where nodes represent users and edges denote trust relationships. The MovieLens dataset is a user–item interaction network, where users are connected to movies through rating relationships. From each dataset, we construct a bipartite graph G(V1, V2, E), where V1 represents candidate seed nodes and V2 represents target nodes to be activated. For all datasets, we fix |V1| = 500, |V2| = 2000.

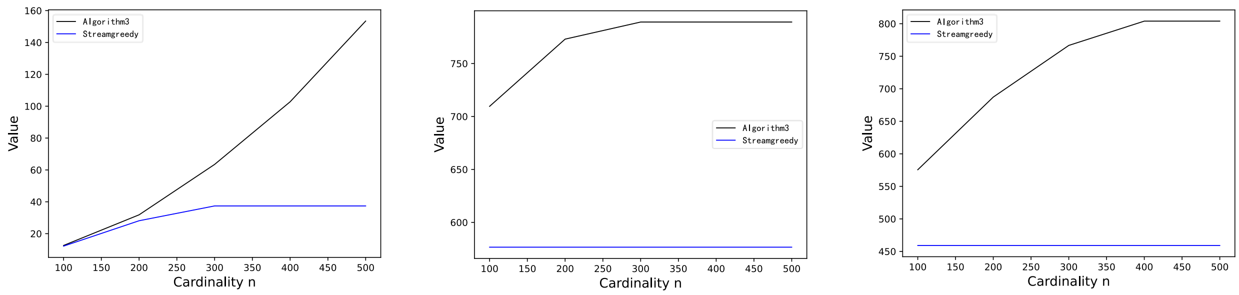
We choose some nodes from V1 as initially activated seeds. Through network propagation, we aim to activate as many nodes in V2 as possible. We set ϵ=0.1. For each selected seed node and its neighbors, a random number between (0, 1) is chosen as the initial activation probability. If a node is selected 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.

Fig. 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 3 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 3 increases with cardinality, indicating that our algorithm performs better than the Stream Greedy algorithm.

Fig. 2 shows the changes in the outputs of the two algorithms as the knapsack capacity increases. It can be observed that the results of both algorithms increase with the knapsack capacity, but the result of Algorithm 3 is consistently above that of the Stream Greedy algorithm, meaning our algorithm yields better results. Therefore, in terms of solution quality, our algorithm outperforms the Stream Greedy algorithm, although the Stream Greedy algorithm is faster in terms of computational speed.

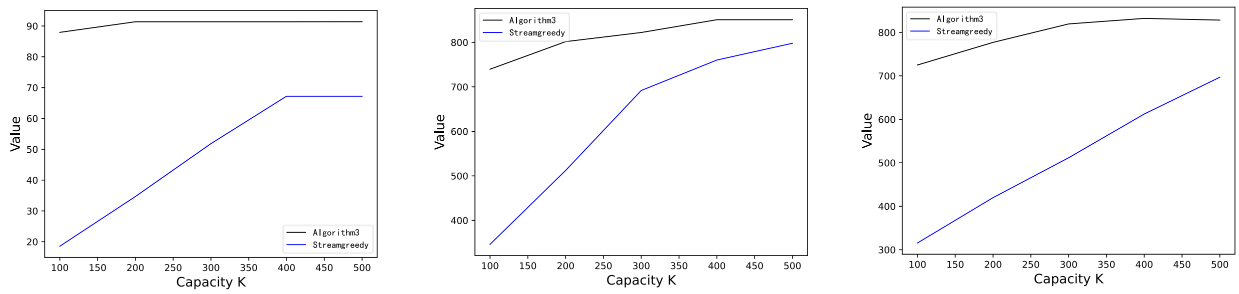
6. Conclusions

The balance between benefits and costs constitutes a core issue in team building. This problem can be efficiently addressed by transforming it into an optimization task maximizing the difference between a DR-submodular function and a linear function. The former captures the collaborative gains derived from team interactions, embodying the properties of cumulative synergy and diminishing marginal returns among members, while the latter quantifies resource expenditures (e.g., human capital, temporal investments) through a linear cost structure. This paper focuses on the maximization problem of a monotone non-negative submodular function minus a non-negative linear function with knapsack constraints on integer lattices. We design an efficient bicriteria streaming algorithm (Algorithm 1) and conduct a comprehensive analysis of its performance. To address the contradiction between the assumption of a priori known threshold in Algorithm 1 and the practical challenge of threshold inaccessibility, Algorithm 3 constructs a threshold estimation interval by leveraging the maximum value of a single point. Performance analysis demonstrates that the proposed algorithm achieves a space complexity of $O((K\log K)/\varepsilon)$ and a time complexity as $O(\log^2 K/\varepsilon)$. Additionally, for the extended scenario where the first function is non-submodular, this paper develops a corresponding streaming algorithm and conducts in-depth discussions on its performance.



a b c

Fig. 1: Solution quality versus cardinality n for Algorithm 3 and the Stream Greedy algorithm on three datasets: (a) email-enron-large, (b) soc-Epinions1, and (c) MovieLens. Algorithm 3 consistently achieves higher values as n increases



a b c

Fig. 2: Solution quality versus knapsack capacity K for Algorithm 3 and the Stream Greedy algorithm on three datasets: (a) email-enron-large, (b) soc-Epinions1, and (c) MovieLens. Algorithm 3 consistently outperforms the Stream Greedy algorithm

7. Research limitations and future work

This study has several limitations that point to promising directions for future research. First, the algorithm's performance relies on the assumptions of DR-submodularity and linear costs. Future work can explore non-DR-submodular and nonlinear cost models that better align with complex real-world team scenarios (e.g., heterogeneous member contributions, dynamic resource allocation). Second, the current research does not address real-time optimization for dynamic teams, such as those involving member turnover or task iteration. Subsequent studies could develop adaptive streaming algorithms to enhance adaptability to such dynamic contexts. Third, the numerical experiments in this paper only focus on verifying theoretical complexity. Future research should conduct empirical analyses using real team-building data (e.g., corporate project teams, academic research groups) to validate the algorithms, practical applicability, and effectiveness.

Acknowledgment

The first author is supported by the Natural Science Foundation of China (No.12301417) and the National Natural Science Foundation of Shandong Province (No. ZR2022MA034). The second author is supported by the Natural Science Foundation of China (Nos.11401438, 11571120). The fourth author is supported by the National Natural Science Foundation of China (No.12101587).

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Anagnostopoulos A, Becchetti L, Castillo C, Gionis A, and Leonardi S (2012). Online team formation in social networks. In the Proceedings of the 21st International Conference on World Wide Web, ACM, Lyon, France: 839-848. https://doi.org/10.1145/2187836.2187950 **PMCid:PMC9645621**

Anagnostopoulos A, Castillo C, Fazzone A, Leonardi S, and Terzi E (2018). Algorithms for hiring and outsourcing in the online labor market. In the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, London, UK: 1109-1118. https://doi.org/10.1145/3219819.3220056 **PMid:30278844**

Byrnes KM (2015). A tight analysis of the submodular–supermodular procedure. Discrete Applied Mathematics, 186: 275-282. https://doi.org/10.1016/j.dam.2015.01.026

Du DL, Li Y, Xiu NH, and Xu DC (2014). Simultaneous approximation of multi-criteria submodular function maximization. Journal of the Operations Research Society of

China, 2: 271-290.
https://doi.org/10.1007/s40305-014-0053-z

Feldman M (2021). Guess free maximization of submodular and linear sums. Algorithmica, 83: 853-878.
https://doi.org/10.1007/s00453-020-00757-9

Gottschalk C and Peis B (2015). Submodular function maximization on the bounded integer lattice. In: Sanità L and Skutella M (Eds.), International workshop on approximation and online algorithms: 133-144. Springer, Cham, Switzerland.
https://doi.org/10.1007/978-3-319-28684-6_12

Kapralov M, Post I, and Vondrák J (2013). Online submodular welfare maximization: Greedy is optimal. In the Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, New Orleans, USA: 1216-1225.
https://doi.org/10.1137/1.9781611973105.88

Lappas T, Liu K, and Terzi E (2009). Finding a team of experts in social networks. In the Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Paris, France: 467-476.
https://doi.org/10.1145/1557019.1557074
PMCid:PMC9645621

Lovász L (1979). On the Shannon capacity of a graph. IEEE Transactions on Information Theory, 25(1): 1-7.
https://doi.org/10.1109/TIT.1979.1055985

Nikolakaki SM, Ene A, and Terzi E (2021). An efficient framework for balancing submodularity and cost. In the Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, ACM, Singapore, Singapore: 1256-1266.
https://doi.org/10.1145/3447548.3467367

Nong Q, Fang J, Gong S, Du D, Feng Y, and Qu X (2020). A 1/2-approximation algorithm for maximizing a non-monotone weak-submodular function on a bounded integer lattice. Journal of Combinatorial Optimization, 39(4): 1208-1220.
https://doi.org/10.1007/s10878-020-00558-4

Sarpatwar KK, Schieber B, and Shachnai H (2019). Constrained submodular maximization via greedy local search. Operations Research Letters, 47(1): 1-6.
https://doi.org/10.1016/j.orl.2018.11.002

Soma T and Yoshida Y (2018). Maximizing monotone submodular functions over the integer lattice. Mathematical Programming, 172: 539-563. https://doi.org/10.1007/s10107-018-1324-y

Sviridenko M, Vondrák J, and Ward J (2017). Optimal approximation for submodular and supermodular optimization with bounded curvature. Mathematics of Operations Research, 42(4): 1197-1218.
https://doi.org/10.1287/moor.2016.0842

Tan J, Li M, Sun M, and Yang R (2024). Streaming algorithm for balance gain and cost with knapsack constraint on the integer lattice. In: Li Y, Zhang Y, and Xu J (Eds.), Parallel and distributed computing: Applications and technologies: 534-540. Springer, Singapore, Singapore.
https://doi.org/10.1007/978-981-96-4207-6_48

Tan J, Xu Y, Zhang D, and Zhang X (2023). On streaming algorithms for maximizing a supermodular function plus a MDR-submodular function on the integer lattice. Journal of Combinatorial Optimization, 45: 55.
https://doi.org/10.1007/s10878-023-00986-y

Wang Y, Xu D, Du D, and Jiang Y (2022). Bicriteria streaming algorithms to balance gain and cost with cardinality constraint. Journal of Combinatorial Optimization, 44: 2946-2962. https://doi.org/10.1007/s10878-021-00827-w

Wang Y, Xu Y, and Yang X (2021). On maximizing the difference between an approximately submodular function and a linear function subject to a matroid constraint. In: Du DZ, Du D, Wu C, and Xu D (Eds.), Combinatorial optimization and applications: 75-85. Springer International Publishing, Cham, Switzerland. https://doi.org/10.1007/978-3-030-92681-6_7