Multi-Objective Submodular Maximization by Regret Ratio Minimization with Theoretical Guarantee

Chao Feng,[1,2] **Chao Qian**[1]
[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China
chaofeng@mail.ustc.edu.cn, qianc@lamda.nju.edu.cn

Abstract

Submodular maximization has attracted much attention due to its wide application and attractive property. Previous works mainly considered one single objective function, while there can be multiple ones in practice. As the objectives are usually conflicting, there exists a set of Pareto optimal solutions, attaining different optimal trade-offs among multiple objectives. In this paper, we consider the problem of minimizing the regret ratio in multi-objective submodular maximization, which is to find at most k solutions to approximate the whole Pareto set as well as possible. We propose a new algorithm RRMS by sampling representative weight vectors and solving the corresponding weighted sums of objective functions using some given α-approximation algorithm for single-objective submodular maximization. We prove that the regret ratio of the output of RRMS is upper bounded by $1-\alpha+O(\sqrt{d-1}\cdot(\frac{d}{k-d})^{\frac{1}{d-1}})$, where d is the number of objectives. This is the first theoretical guarantee for the situation with more than two objectives. When $d=2$, it reaches the $(1-\alpha+O(1/k))$-guarantee of the only existing algorithm POLYTOPE. Empirical results on the applications of multi-objective weighted maximum coverage and Max-Cut show the superior performance of RRMS over POLYTOPE.

Introduction

Submodular maximization tries to find a subset maximizing a submodular objective function (i.e., an objective function satisfying the natural diminishing returns property) under some constraints. It arises in various real-world applications, such as influence maximization (Kempe, Kleinberg, and Tardos 2003) and sensor placement (Krause, Singh, and Guestrin 2008). The problems of maximizing submodular functions are usually NP-hard, and many polynomial-time algorithms with bounded approximation guarantees have been proposed. For example, for maximizing a monotone submodular function with a cardinality constraint, the greedy algorithm achieves the optimal polynomial-time approximation guarantee of $(1-1/e)$ (Nemhauser, Wolsey, and Fisher 1978; Nemhauser and Wolsey 1978); for maximizing a non-monotone submodular function, the double greedy algorithm achieves the optimal guarantee of $(1/2)$ (Buchbinder et al. 2015).

Previous works mainly considered one single submodular objective function, while there can be multiple ones in several applications. For example, the variable selection problem under parameter uncertainty (Krause et al. 2008) tries to find a subset of observed variables which simultaneously maximizes information gains (which are submodular) for all possible parameter values; the document summarization task (Lin and Bilmes 2011) is modeled by maximizing two submodular objective functions, i.e., coverage and diversity; multi-objective weighted Max-Cut (Angel, Bampis, and Gourvs 2006; Bhangale et al. 2018) is to find a subset of vertices from a graph (where each edge has a weight vector instead of only a single weight), maximizing the total weight vector of the edges between the selected subset and its complement, each dimension of which corresponds to one submodular objective function.

Multi-objective submodular maximization can be formulated as

$$\max\ (f_1(X),f_2(X),\dots,f_d(X))\ \ s.t.\ \ X\in\mathcal{C}, \quad (1)$$

where $\mathcal{C}$ is the feasible solution set, and $\forall i: f_i(X)$ is the i-th submodular objective function to be maximized. As the objective functions can be conflicting, it is impossible to have one solution (i.e., subset) which is optimal for all objectives, while there is a set of Pareto optimal solutions, attaining different optimal trade-offs among multiple objectives.

However, it is often impractical to find the whole Pareto set, whose size can be even exponentially large. A simplified alternative approach is to consider the robust formulation (Krause et al. 2008; Udwani 2018; Anari et al. 2019):

$$\arg\max_{X\in\mathcal{C}}\ \min_{i\in\{1,2,\dots,d\}} f_i(X), \quad (2)$$

that is to find a solution maximizing the minimum of all submodular objectives. Recently, Soma and Yoshida (2017) considered the formulation of minimizing regret ratio:

$$\arg\min_{S\subseteq\mathcal{C},|S|\le k}\ \max_{\boldsymbol{w}\in\mathbb{R}_+^d} \frac{\max_{X\in\mathcal{C}} f^{\boldsymbol{w}}(X)-\max_{X\in S} f^{\boldsymbol{w}}(X)}{\max_{X\in\mathcal{C}} f^{\boldsymbol{w}}(X)}, \quad (3)$$

where $\mathbb{R}_+$ denotes the set of non-negative reals, and $f^{\boldsymbol{w}}(X)=\sum_{i=1}^d w_i f_i(X)$ is the linear combination of objective functions w.r.t. the weight vector $\boldsymbol{w}$. That is, the goal is to select at most k solutions to approximate the whole Pareto set as well as possible. In fact, robust submodular maximization is a special case of regret ratio minimization. By setting $k = 1$ and $\boldsymbol{w} \in \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_d\}$, where $\forall i : \boldsymbol{e}_i$ is the basis vector with all entries equal to 0, except the i-th entry, which is 1, Eq. (3) is specialized to

$$\arg\max_{X\in\mathcal{C}} \min_{i\in\{1,2,\ldots,d\}} \frac{f_i(X)}{\max_{X\in\mathcal{C}} f_i(X)},$$

which is a normalized version of Eq. (2), and often considered in real-world robust submodular maximization applications, e.g., robust influence maximization (Chen et al. 2016).

Soma and Yoshida (2017) proposed a geometric algorithm POLYTOPE for regret ratio minimization. Given an α-approximation algorithm for single-objective submodular maximization, POLYTOPE first gets an initial set of d solutions by applying the α-approximation algorithm to solve each objective function, and then iteratively generates new solutions until finding k solutions in total. In each iteration, POLYTOPE first computes the convex hull that contains the objective vectors of solutions generated-so-far and the boundary hyperplanes of the first quadrant; then uses the non-negative normal vectors of the frontier facets of the convex hull as the weight vectors to linearly combine the d objective functions; finally applies the α-approximation algorithm to these new single-objective submodular functions to generate new solutions. They proved that when the number d of objectives is two, the set of k solutions output by POLYTOPE has a regret ratio upper bounded by $1 - \alpha + O(1/k)$.

Though the performance of POLYTOPE is theoretically guaranteed for bi-objective (i.e., $d = 2$) submodular maximization, the number of objectives can be larger than two in real-world multi-objective applications, e.g., (Matrosov et al. 2015; Hierons et al. 2020). In this paper, we propose a new algorithm RRMS. The main idea is to sample representative weight vectors such that the distance between any weight vector and the nearest sampled one is small, and then apply the α-approximation algorithm to solve single-objective submodular functions generated by linearly combining all objectives using the sampled weight vectors. We prove that the set of k solutions output by RRMS has a regret ratio upper bounded by $1 - \alpha + O(\sqrt{d-1} \cdot (\frac{d}{k-d})^{\frac{1}{d-1}})$. Note that this is the first theoretical guarantee for $d > 2$. When $d = 1$, it reaches the $(1 - \alpha)$-guarantee of the α-approximation algorithm for single-objective submodular maximization. When $d = 2$, it reaches the $(1 - \alpha + O(1/k))$-guarantee of POLYTOPE. We also empirically compare RRMS with POLYTOPE on the applications of multi-objective weighted maximum coverage and Max-Cut with d from 2 to 7. The results show that they perform similarly for small d, while RRMS becomes better as d increases.

Preliminaries

Let $\mathbb{R}$ and $\mathbb{R}_+$ denote the set of reals and non-negative reals, respectively. Given a finite set V, we study the functions $f : 2^V \to \mathbb{R}$ defined on subsets of V. A set function $f : 2^V \to \mathbb{R}$ is monotone if $\forall X \subseteq Y : f(X) \leq f(Y)$. f is submodular (Nemhauser, Wolsey, and Fisher 1978) if $\forall X \subseteq Y \subseteq V, v \in V \setminus Y : f(X \cup \{v\}) - f(X) \geq f(Y \cup \{v\}) - f(Y)$, which intuitively represents the "diminishing returns" property, i.e., adding an item to a set X does not give a smaller benefit than adding the same item to a superset of X. Any linear combination of submodular functions is still submodular.

Let $\mathcal{C}$ denote the feasible solution set, i.e., the set of solutions satisfying the constraints. Note that a solution corresponds to a subset of V here. For any $S \subseteq \mathcal{C}$, its regret ratio w.r.t. a function $f : 2^V \to \mathbb{R}_+$ is

$$\begin{aligned}\mathrm{rr}_{f,\mathcal{C}}(S) &= \frac{\max_{X\in\mathcal{C}} f(X) - \max_{X\in S} f(X)}{\max_{X\in\mathcal{C}} f(X)} \qquad (4)\\ &= 1 - \frac{\max_{X\in S} f(X)}{\max_{X\in\mathcal{C}} f(X)}.\end{aligned}$$

Obviously, $\mathrm{rr}_{f,\mathcal{C}}(S) \in [0, 1]$, and the smaller the better.

Given d submodular functions $f_1, f_2, \ldots, f_d : 2^V \to \mathbb{R}_+$ and the feasible solution set $\mathcal{C}$, multi-objective submodular maximization tries to find solutions from $\mathcal{C}$ maximizing these d functions simultaneously, as shown in Eq. (1). The objectives are usually conflicting, i.e., optimizing one objective alone will degrade the other ones, and thus it is impossible to have one solution that optimizes all objectives simultaneously. Instead, there is a set of Pareto optimal solutions (Deb et al. 2002; Qian, Yu, and Zhou 2013), attaining different optimal tradeoffs among the d objectives. That is, an optimal solution of the non-negative linear combination (denoted as $f^{\boldsymbol{w}}$) of $f_1, f_2, \ldots, f_d$ w.r.t. any weight vector $\boldsymbol{w} \in \mathbb{R}^d_+$ is a Pareto optimal solution. For any $X \in \mathcal{C}$, $f^{\boldsymbol{w}}(X) = \sum_{i=1}^d w_i f_i(X)$.

For any $S \subseteq \mathcal{C}$, its regret ratio w.r.t. $f_1, f_2, \ldots, f_d$ is

$$\mathrm{rr}_{f_1,f_2,\ldots,f_d,\mathcal{C}}(S) = \max_{\boldsymbol{w}\in\mathbb{R}^d_+} \mathrm{rr}_{f^{\boldsymbol{w}},\mathcal{C}}(S), \qquad (5)$$

which intuitively characterizes the approximation degree of S to the whole Pareto set. The smaller the regret ratio, the better the approximation. The problem of regret ratio minimization is to find at most k solutions best approximating the whole Pareto set, as presented in Definition 1. Note that we consider non-negative functions and $k > d$.

Definition 1 (Regret Ratio Minimization (Soma and Yoshida 2017)). *Given submodular functions $f_1, f_2, \ldots, f_d : 2^V \to \mathbb{R}_+$ and the feasible solution set $\mathcal{C}$, to find a solution set $S \subseteq \mathcal{C}$ with size at most k minimizing the regret ratio $\mathrm{rr}_{f_1,f_2,\ldots,f_d,\mathcal{C}}(S)$, i.e.,*

$$\arg\min_{S\subseteq\mathcal{C},|S|\leq k} \ \mathrm{rr}_{f_1,f_2,\ldots,f_d,\mathcal{C}}(S). \qquad (6)$$

To compute the regret ratio in Eq. (5), one needs to enumerate all non-negative weight vectors. In fact, we only need to consider all non-negative unit vectors, since the regret ratio w.r.t. $f^{\boldsymbol{w}}$ is equivalent to that w.r.t. $f^{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}}$.

Lemma 1. $\forall S \subseteq \mathcal{C} : \mathrm{rr}_{f^{\boldsymbol{w}},\mathcal{C}}(S) = \mathrm{rr}_{f^{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}},\mathcal{C}}(S)$.

Proof. By Eq. (4), we have

$$\mathrm{rr}_{f^{\boldsymbol{w}},\mathcal{C}}(S) = 1 - \frac{\|\boldsymbol{w}\| \cdot \max_{X\in S} f^{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}}(X)}{\|\boldsymbol{w}\| \cdot \max_{X\in\mathcal{C}} f^{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}}(X)} = \mathrm{rr}_{f^{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}},\mathcal{C}}(S),$$

where the first equality holds by $f^{\boldsymbol{w}}(X) = \sum_{i=1}^d w_i \cdot f_i(X) = \|\boldsymbol{w}\| \cdot \sum_{i=1}^d \frac{w_i}{\|\boldsymbol{w}\|} \cdot f_i(X) = \|\boldsymbol{w}\| \cdot f^{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}}(X)$. □

Thus, Eq. (6) is equivalent to

$$\mathop{\arg\min}_{S\subseteq\mathcal{C},|S|\le k} \max_{\boldsymbol{w}\in\mathbb{R}^d_+,\|\boldsymbol{w}\|=1} \mathrm{rr}_{f^{\boldsymbol{w}},\mathcal{C}}(S). \tag{7}$$

The RRMS Algorithm

In this section, we propose a new algorithm, called RRMS, for Regret Ratio Minimization by Sampling representative weight vectors and solving the corresponding weighted sums of objective functions.

As presented in Algorithm 1, RRMS first applies an existing α-approximation algorithm $\mathcal{A}$ to solve each single objective function f_i, and obtains d solutions. Note that each single objective function f_i can be viewed as the linear combination $f^{\boldsymbol{w}}$ w.r.t. $\boldsymbol{w} = \boldsymbol{e}_i$, i.e., the basis vector. After that, RRMS samples $(k-d)$ unit weight vectors, and linearly combines the d objective functions using these vectors. For the linear combination $f^{\boldsymbol{w}}$ w.r.t. each sampled unit weight vector $\boldsymbol{w}$, RRMS applies $\mathcal{A}$ to solve it, and obtains a solution. The resulting $(k-d)$ solutions are combined with the initial d solutions as the final output of RRMS.

Note that there have been many polynomial-time approximation algorithms for diverse single-objective submodular maximization. For example, for maximizing a non-monotone submodular function, simulated annealing is a roughly 0.41-approximation algorithm (Gharan and Vondrák 2011), and the double greedy algorithm is an $(1/2)$-approximation algorithm (Buchbinder et al. 2015). Thus, the α-approximation algorithm for single-objective submodular maximization is left as a parameter of RRMS. An α-approximation algorithm $\mathcal{A}$ w.r.t. a function f implies that the solution X^* output by $\mathcal{A}$ satisfies $f(X^*) \ge \alpha \cdot \max_{X\in\mathcal{C}} f(X)$.

Next, we introduce how to select the representative unit weight vectors, which is inspired by (Agarwal, Har-Peled, and Varadarajan 2004; Xie et al. 2018). Let the hypercube $\mathbb{C}^d_+ = \{\boldsymbol{x} \mid \boldsymbol{x} \in [0,1]^d\}$, which has $2d$ facets. Consider the d facets, denoted by $F_1, F_2, \ldots, F_d$, which don't contain the origin. A hyperplane in a d-dimensional space is a subspace with dimension $d-1$, which can be represented as $h(\boldsymbol{u}, c) = \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{u}^{\mathrm{T}} \cdot \boldsymbol{x} = c\}$ where $\boldsymbol{u}$ is the unit normal vector of the hyperplane and c is the offset. For each facet F_i, where $i \in \{1, 2, \ldots, d\}$, let $\boldsymbol{u}^i$ and c_i denote the unit normal vector and offset, respectively, of the hyperplane containing F_i. Thus, any hyperplane parallel to the hyperplane containing F_i can be represented as $h(\boldsymbol{u}^i, c)$ for some constant c.

For each $i \in \{1, 2, \ldots, d\}$, the facet F_i is uniformly partitioned into m^{d-1} small facets as follows. For any F_j with $j \ne i$, we generate m hyperplanes $\{h(\boldsymbol{u}^j, lc_j/m) \mid l \in \{1, 2, \ldots, m\}\}$ parallel to the hyperplane containing F_j. Thus, $(d-1)\cdot m$ hyperplanes $\{h(\boldsymbol{u}^j, lc_j/m) \mid j \in \{1, 2 \ldots, d\} \setminus \{i\}, l \in \{1, 2, \ldots, m\}\}$ are generated, which are then used to partition F_i into m^{d-1} small facets. By repeating this process for each F_i, we get $d \cdot m^{d-1}$ small facets in total. The unit vector starting from the origin and passing the central point of a small facet is selected as a representative unit weight vector. Thus, $d \cdot m^{d-1}$ unit weight vectors are generated. These selected unit vectors are actually a good approximation of the space of all unit vectors, since the distance between any unit vector and the nearest selected one is small, which will be proved in theoretical analysis.

Algorithm 1 RRMS Algorithm

```
Input: submodular objective functions $f_1, f_2, \ldots, f_d : 2^V \to \mathbb{R}_+$; feasible solution set $\mathcal{C} \subseteq 2^V$; integer $k > d$
Parameter: $\alpha$-approximation algorithm $\mathcal{A}$ for single-objective submodular maximization
Output: a set of $k$ solutions
Process:
for each $i \in \{1, 2, \ldots, d\}$ do
    $X_i \leftarrow$ the solution obtained by applying $\mathcal{A}$ to $f_i$
end for
let $S = \{X_1, X_2, \ldots, X_d\}$;
apply Algorithm 2 to sample $(k-d)$ unit vectors;
for each sampled unit vector $\boldsymbol{w}$ do
    $X \leftarrow$ the solution obtained by applying $\mathcal{A}$ to $f^{\boldsymbol{w}}$;
    $S = S \cup \{X\}$
end for
return $S$
```

Algorithm 2 Weight Vector Sampling Procedure

```
Output: $(k-d)$ unit vectors
Process:
let $m = \lfloor (\frac{k-d}{d})^{\frac{1}{d-1}} \rfloor$, $P = \emptyset$, and $U = \emptyset$;
for each $i \in \{1, 2, \ldots, d\}$ do
    construct $(d-1)\cdot m$ hyperplanes $\{h(\boldsymbol{u}^j, lc_j/m) \mid j \in \{1, 2 \ldots, d\} \setminus \{i\}, l \in \{1, 2, \ldots, m\}\}$, where $\boldsymbol{u}^j$ and $c_j$ denote the unit normal vector and offset, respectively, of the hyperplane containing $F_j$, and $h(\boldsymbol{u}^j, lc_j/m) = \{\boldsymbol{x} \in \mathbb{R}^d \mid (\boldsymbol{u}^j)^{\mathrm{T}} \cdot \boldsymbol{x} = lc_j/m\}$;
    partition $F_i$ into $m^{d-1}$ small facets using these $(d-1)\cdot m$ hyperplanes;
    add the central points of these small facets into $P$
end for
for each $\boldsymbol{p} \in P$ do
    add the unit vector, which starts from the origin and passes $\boldsymbol{p}$, into $U$
end for
add $(k-d-dm^{d-1})$ random unit vectors into $U$
return $U$
```

RRMS requires to sample $(k-d)$ unit weight vectors. Thus, m is set to $\lfloor (\frac{k-d}{d})^{\frac{1}{d-1}} \rfloor$, and the remaining $(k-d-dm^{d-1})$ unit weight vectors are randomly generated. The whole procedure of sampling representative weight vectors is presented in Algorithm 2.

Theoretical Analysis

In this section, we prove that the regret ratio of the solution set generated by RRMS can be well upper bounded.

Theorem 1. *For regret ratio minimization in Definition 1, the solution set S generated by* RRMS *satisfies*

$$\mathrm{rr}_{f_1,f_2,\ldots,f_d,\mathcal{C}}(S) \le 1-\alpha + O\left(\sqrt{d-1}\cdot\left(\frac{d}{k-d}\right)^{\frac{1}{d-1}}\right).$$

When $d = 1$, the upper bound on the regret ratio by RRMS becomes $1 - \alpha$. This is reasonable, because RRMS will degenerate to applying the α-approximation algorithm $\mathcal{A}$ to solve the only objective f. Note that α-approximation implies that the generated solution X^* satisfies $f(X^*) \geq \alpha \cdot \max_{X \in \mathcal{C}} f(X)$, and thus the regret ratio $1 - f(X^*)/\max_{X \in \mathcal{C}} f(X) \leq 1 - \alpha$. When $d = 2$, the upper bound becomes $1 - \alpha + O(1/k)$, which is the same as that by the only existing algorithm POLYTOPE (Soma and Yoshida 2017). When $d \geq 3$, this is the first theoretical guarantee.

To prove Theorem 1, we first show in Lemma 2 that the set U of unit vectors sampled by Algorithm 2 can approximate the space of all unit vectors well. That is, for any unit vector $\boldsymbol{w}$, U must contain one unit vector close to $\boldsymbol{w}$.

Lemma 2. *Let U denote the set of unit vectors output by Algorithm 2. For any unit vector $\boldsymbol{w} \in \mathbb{R}^d_+$, there exists one unit vector $\boldsymbol{w}' \in U$ such that the angle between $\boldsymbol{w}$ and $\boldsymbol{w}'$ is no greater than $2 \cdot \arcsin(\frac{\sqrt{d-1}}{4}/\lfloor(\frac{k-d}{d})^{\frac{1}{d-1}}\rfloor)$.*

Proof. For any unit vector $\boldsymbol{w} \in \mathbb{R}^d_+$, the ray starting from the origin and going along $\boldsymbol{w}$ must have one intersection point (denoted by $\boldsymbol{s}$) with one of the dm^{d-1} small facets generated in Algorithm 2. Let $\boldsymbol{s}'$ denote the central point of this small facet, and let $\boldsymbol{w}'$ be the unit vector starting from the origin and passing $\boldsymbol{s}'$. By lines 7–9 of Algorithm 2, it holds that $\boldsymbol{w}' \in U$. According to the analysis in (Agarwal, Har-Peled, and Varadarajan 2004), we have $\|\boldsymbol{w} - \boldsymbol{w}'\| \leq \|\boldsymbol{s} - \boldsymbol{s}'\|$.

By the procedure of generating small facets in lines 3–4 of Algorithm 2, we know that the diameter (i.e., the longest distance between any two points in the facet) of each small facet is $\frac{\sqrt{d-1}}{m}$. This implies that in a small facet, the distance between any point and the central point is at most $\frac{\sqrt{d-1}}{2m}$. Thus, $\|\boldsymbol{s} - \boldsymbol{s}'\| \leq \frac{\sqrt{d-1}}{2m}$, implying $\|\boldsymbol{w} - \boldsymbol{w}'\| \leq \frac{\sqrt{d-1}}{2m}$.

By simple calculation, the angle between $\boldsymbol{w}$ and $\boldsymbol{w}'$ is at most $2 \cdot \arcsin(\frac{\sqrt{d-1}}{4m})$. Because $m = \lfloor(\frac{k-d}{d})^{\frac{1}{d-1}}\rfloor$ as shown in line 1 of Algorithm 2, the lemma holds. □

Lemma 3 gives the relationship among the angles of three vectors, which will also be used in the proof of Theorem 1.

Lemma 3. *For any three non-zero vectors $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^d_+$, let θ_1 denote the angle between $\boldsymbol{u}$ and $\boldsymbol{w}$, θ_2 denote the angle between $\boldsymbol{v}$ and $\boldsymbol{w}$, and θ denote that between $\boldsymbol{u}$ and $\boldsymbol{v}$. Then,*

$$|\theta_1 - \theta_2| \leq \theta \leq \theta_1 + \theta_2. \tag{8}$$

Proof. Since $\cos(\cdot)$ is decreasing in $[0, \pi/2]$ and $\theta_1, \theta_2, \theta \in [0, \pi/2]$, $|\theta_1 - \theta_2| \leq \theta$ is equivalent to $\cos(\theta_1 - \theta_2) \geq \cos(\theta)$. It is clear that $\theta_1 + \theta_2 \in [0, \pi]$. When $\theta_1 + \theta_2 \geq \pi/2$, $\theta \leq \theta_1 + \theta_2$ trivially holds. Thus, we consider $\theta_1 + \theta_2 < \pi/2$. In this case, $\theta_1 + \theta_2 \geq \theta$ is equivalent to $\cos(\theta_1 + \theta_2) \leq \cos(\theta)$. Thus, to prove Eq. (8), we only need to prove $\cos(\theta_1 + \theta_2) \leq \cos(\theta) \leq \cos(\theta_1 - \theta_2)$, i.e.,

$$|\cos(\theta_1)\cos(\theta_2) - \cos(\theta)| \leq \sin(\theta_1)\sin(\theta_2). \tag{9}$$

Taking the square of both sides of Eq. (9), we are to prove

$$\begin{aligned} &\cos^2(\theta_1) + \cos^2(\theta_2) + \cos^2(\theta) \\ &\quad - 2\cos(\theta_1)\cos(\theta_2)\cos(\theta) \leq 1. \end{aligned} \tag{10}$$

Assume that $\|\boldsymbol{u}\| = 1$, $\|\boldsymbol{v}\| = 1$ and $\|\boldsymbol{w}\| = 1$, because scaling the length of vectors will not change the angles between them. Thus, $\cos(\theta_1) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{u}$, $\cos(\theta_2) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{v} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{w}$, $\cos(\theta) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}$, and Eq. (10) can be rewritten as

$$\begin{aligned} &\boldsymbol{w}^{\mathrm{T}}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u} \\ &\quad - 2 \cdot \boldsymbol{w}^{\mathrm{T}}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} \leq 1. \end{aligned} \tag{11}$$

Because $\|\boldsymbol{w}\|^2 = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} = 1$, $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u} = \boldsymbol{w}^{\mathrm{T}} \cdot (\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}) \cdot \boldsymbol{w}$, and $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{w} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{w}$, Eq. (11) can be rewritten as

$$\begin{aligned} &\boldsymbol{w}^{\mathrm{T}} \cdot (\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} + \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} - \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} - \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} \\ &\quad + \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u} \cdot \mathbf{I}_d - \mathbf{I}_d) \cdot \boldsymbol{w} \leq 0, \end{aligned} \tag{12}$$

where $\mathbf{I}_d$ denotes the identity matrix of size d.

Next, we are to show that Eq. (12) holds. We rotate the coordinate system to make $\boldsymbol{u} = [1, 0, \ldots, 0]^{\mathrm{T}}$, which will not change the angles. Let $\hat{\boldsymbol{v}} = [v_2, v_3, \ldots, v_d]^{\mathrm{T}}$, and $\mathbf{0}_{p,q}$ denote the zero matrix of size $p \times q$. By calculation, we can derive that $\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}} + \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} - \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} - \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ is equal to

$$\begin{bmatrix} 1 - v_1^2 & 0 & \ldots & 0 \\ 0 & v_2^2 & \ldots & v_2 v_d \\ \vdots & \vdots & \vdots & \vdots \\ 0 & v_d v_2 & \ldots & v_d^2 \end{bmatrix} = \begin{bmatrix} 1 - v_1^2 & \mathbf{0}_{1,d-1} \\ \mathbf{0}_{d-1,1} & \hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^{\mathrm{T}} \end{bmatrix},$$

and $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u} \cdot \mathbf{I}_d - \mathbf{I}_d$ is equal to

$$(v_1^2 - 1) \cdot \mathbf{I}_d = \begin{bmatrix} v_1^2 - 1 & \mathbf{0}_{1,d-1} \\ \mathbf{0}_{d-1,1} & -\hat{\boldsymbol{v}}^{\mathrm{T}}\hat{\boldsymbol{v}} \cdot \mathbf{I}_{d-1} \end{bmatrix},$$

where the last equality holds because $\|\boldsymbol{v}\|^2 = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} = v_1^2 + \hat{\boldsymbol{v}}^{\mathrm{T}}\hat{\boldsymbol{v}} = 1$. Let $\hat{\boldsymbol{w}} = [w_2, w_3, \ldots, w_d]^{\mathrm{T}}$. Thus, the left side of Eq. (12) is equal to

$$\begin{aligned} &\begin{bmatrix} w_1, \hat{\boldsymbol{w}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{0}_{1,d-1} \\ \mathbf{0}_{d-1,1} & \hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} w_1 \\ \hat{\boldsymbol{w}} \end{bmatrix} - \\ &\quad \begin{bmatrix} w_1, \hat{\boldsymbol{w}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{0}_{1,d-1} \\ \mathbf{0}_{d-1,1} & \hat{\boldsymbol{v}}^{\mathrm{T}}\hat{\boldsymbol{v}} \cdot \mathbf{I}_{d-1} \end{bmatrix} \begin{bmatrix} w_1 \\ \hat{\boldsymbol{w}} \end{bmatrix} \\ &= \hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^{\mathrm{T}}\hat{\boldsymbol{w}} - \hat{\boldsymbol{v}}^{\mathrm{T}}\hat{\boldsymbol{v}}\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{w}} = (\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{v}})^2 - \|\hat{\boldsymbol{v}}\|^2\|\hat{\boldsymbol{w}}\|^2, \end{aligned}$$

which is obviously no greater than 0. Thus, Eq. (12) holds, implying that the lemma holds. □

Before proving Theorem 1, we briefly introduce the proof idea. For any unit weight vector $\boldsymbol{w}$, Lemma 2 shows that there is a "neighbor" (denoted by $\boldsymbol{u}$) in the unit vectors sampled by line 5 of RRMS. That is, the angle between $\boldsymbol{w}$ and $\boldsymbol{u}$ is small. RRMS applies an α-approximation algorithm to solve the objective $f^{\boldsymbol{u}}$. As $\boldsymbol{w}$ and $\boldsymbol{u}$ are close, the resulting solution is also a good approximation solution for $f^{\boldsymbol{w}}$. Thus, the regret ratio in Eq. (7), i.e., $\max_{\boldsymbol{w} \in \mathbb{R}^d_+, \|\boldsymbol{w}\|=1} 1 - \frac{\max_{X \in S} f^{\boldsymbol{w}}(X)}{\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)}$, can be well upper bounded.

Proof of Theorem 1. Let S denote the solution set generated by RRMS. The regret ratio of S in Eq. (7) is

$$\max_{\boldsymbol{w} \in \mathbb{R}^d_+, \|\boldsymbol{w}\|=1} \mathrm{rr}_{f^{\boldsymbol{w}},\mathcal{C}}(S) = \max_{\boldsymbol{w} \in \mathbb{R}^d_+, \|\boldsymbol{w}\|=1} 1 - \frac{\max_{X \in S} f^{\boldsymbol{w}}(X)}{\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)}.$$

For any unit weight vector $\boldsymbol{w} \in \mathbb{R}_+^d$, let $X_{\mathcal{C}}^{\boldsymbol{w}}$ and $X_S^{\boldsymbol{w}}$ denote an optimal solution w.r.t. $f^{\boldsymbol{w}}$ in $\mathcal{C}$ and S, respectively. That is, $X_{\mathcal{C}}^{\boldsymbol{w}} = \arg\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)$ and $X_S^{\boldsymbol{w}} = \arg\max_{X \in S} f^{\boldsymbol{w}}(X)$. By Lemma 2, there must exist one unit vector $\boldsymbol{u}$ sampled in line 5 of Algorithm 1, satisfying that the angle between $\boldsymbol{w}$ and $\boldsymbol{u}$ is at most $2\arcsin(\frac{\sqrt{d-1}}{4}/\lfloor(\frac{k-d}{d})^{\frac{1}{d-1}}\rfloor)$. Let $X_{\mathcal{C}}^{\boldsymbol{u}}$ and $X_S^{\boldsymbol{u}}$ denote an optimal solution w.r.t. $f^{\boldsymbol{u}}$ in $\mathcal{C}$ and S, respectively. That is, $X_{\mathcal{C}}^{\boldsymbol{u}} = \arg\max_{X \in \mathcal{C}} f^{\boldsymbol{u}}(X)$ and $X_S^{\boldsymbol{u}} = \arg\max_{X \in S} f^{\boldsymbol{u}}(X)$. Next we utilize $X_{\mathcal{C}}^{\boldsymbol{u}}$ and $X_S^{\boldsymbol{u}}$ to analyze

$$1 - \frac{\max_{X \in S} f^{\boldsymbol{w}}(X)}{\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)} = 1 - \frac{f^{\boldsymbol{w}}(X_S^{\boldsymbol{w}})}{f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})}. \tag{13}$$

We use $\boldsymbol{f}(X) = [f_1(X), f_2(X), \dots, f_d(X)]^{\mathrm{T}}$ to denote the objective vector of a solution X. Let $\theta_{\boldsymbol{w}}$ denote the angle between $\boldsymbol{f}(X_S^{\boldsymbol{u}})$ and $\boldsymbol{w}$, and $\theta_{\boldsymbol{u}}$ denote that between $\boldsymbol{f}(X_S^{\boldsymbol{u}})$ and $\boldsymbol{u}$. Let $\gamma_{\boldsymbol{w}}$ denote the angle between $\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}})$ and $\boldsymbol{w}$, and $\gamma_{\boldsymbol{u}}$ denote that between $\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}})$ and $\boldsymbol{u}$. We have

$$\begin{aligned}
& f^{\boldsymbol{w}}(X_S^{\boldsymbol{w}}) \geq f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}}) \qquad (14)\\
& = f^{\boldsymbol{u}}(X_S^{\boldsymbol{u}}) - f^{\boldsymbol{u}}(X_S^{\boldsymbol{u}}) + f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}})\\
& \geq \alpha \cdot f^{\boldsymbol{u}}(X_{\mathcal{C}}^{\boldsymbol{u}}) - f^{\boldsymbol{u}}(X_S^{\boldsymbol{u}}) + f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}})\\
& \geq \alpha \cdot f^{\boldsymbol{u}}(X_{\mathcal{C}}^{\boldsymbol{w}}) - f^{\boldsymbol{u}}(X_S^{\boldsymbol{u}}) + f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}})\\
& = \alpha\|\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}})\|\cos(\gamma_{\boldsymbol{u}}) - \|\boldsymbol{f}(X_S^{\boldsymbol{u}})\|\cos(\theta_{\boldsymbol{u}}) + f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}})\\
& = \alpha f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})\frac{\cos(\gamma_{\boldsymbol{u}})}{\cos(\gamma_{\boldsymbol{w}})} + f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}})\left(1 - \frac{\cos(\theta_{\boldsymbol{u}})}{\cos(\theta_{\boldsymbol{w}})}\right),
\end{aligned}$$

where the first inequality holds because $X_S^{\boldsymbol{w}}$ is an optimal solution w.r.t. $f^{\boldsymbol{w}}$ in S and $X_S^{\boldsymbol{u}} \in S$, the second inequality holds because an α-approximation algorithm is applied to solve $f^{\boldsymbol{u}}$ in Algorithm 1, the third inequality holds because $X_{\mathcal{C}}^{\boldsymbol{u}}$ is an optimal solution w.r.t. $f^{\boldsymbol{u}}$ in $\mathcal{C}$, the second equality holds because $f^{\boldsymbol{u}}(X_{\mathcal{C}}^{\boldsymbol{w}}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}}) = \|\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}})\| \cdot \cos(\gamma_{\boldsymbol{u}})$ and $f^{\boldsymbol{u}}(X_S^{\boldsymbol{u}}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{f}(X_S^{\boldsymbol{u}}) = \|\boldsymbol{f}(X_S^{\boldsymbol{u}})\| \cdot \cos(\theta_{\boldsymbol{u}})$, and the last holds because $f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}}) = \|\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}})\| \cdot \cos(\gamma_{\boldsymbol{w}})$ and $f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{f}(X_S^{\boldsymbol{u}}) = \|\boldsymbol{f}(X_S^{\boldsymbol{u}})\| \cdot \cos(\theta_{\boldsymbol{w}})$.

Applying Eq. (14) to Eq. (13) leads to

$$1 - \frac{f^{\boldsymbol{w}}(X_S^{\boldsymbol{w}})}{f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})} \leq 1 - \alpha\frac{\cos(\gamma_{\boldsymbol{u}})}{\cos(\gamma_{\boldsymbol{w}})} + \frac{f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}})}{f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})}\left(\frac{\cos(\theta_{\boldsymbol{u}})}{\cos(\theta_{\boldsymbol{w}})} - 1\right).$$

Because the goal is to derive an upper bound, we pessimistically assume that $\cos(\theta_{\boldsymbol{u}})/\cos(\theta_{\boldsymbol{w}}) - 1 \geq 0$. By the definition of $X_{\mathcal{C}}^{\boldsymbol{w}}$, i.e., $X_{\mathcal{C}}^{\boldsymbol{w}}$ is an optimal solution w.r.t. $f^{\boldsymbol{w}}$ in $\mathcal{C}$, it holds that $f^{\boldsymbol{w}}(X_S^{\boldsymbol{u}}) \leq f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})$. Thus, we have

$$\begin{aligned}
1 - \frac{f^{\boldsymbol{w}}(X_S^{\boldsymbol{w}})}{f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})} & \leq 1 - \alpha\frac{\cos(\gamma_{\boldsymbol{u}})}{\cos(\gamma_{\boldsymbol{w}})} + \frac{\cos(\theta_{\boldsymbol{u}})}{\cos(\theta_{\boldsymbol{w}})} - 1 \qquad (15)\\
& = 1 - \alpha + \alpha\left(1 - \frac{\cos(\gamma_{\boldsymbol{u}})}{\cos(\gamma_{\boldsymbol{w}})}\right) + \frac{\cos(\theta_{\boldsymbol{u}})}{\cos(\theta_{\boldsymbol{w}})} - 1.
\end{aligned}$$

We have known that the angle between $\boldsymbol{w}$ and $\boldsymbol{u}$ is at most $2\arcsin(\frac{\sqrt{d-1}}{4}/\lfloor(\frac{k-d}{d})^{\frac{1}{d-1}}\rfloor)$, denoted by δ for convenience. Because $\gamma_{\boldsymbol{w}}$ is the angle between $\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}})$ and $\boldsymbol{w}$, and $\gamma_{\boldsymbol{u}}$ is the angle between $\boldsymbol{f}(X_{\mathcal{C}}^{\boldsymbol{w}})$ and $\boldsymbol{u}$, applying Lemma 3 leads to $\gamma_{\boldsymbol{u}} \leq \gamma_{\boldsymbol{w}} + \delta$. Similarly, we can get $\theta_{\boldsymbol{w}} \leq \theta_{\boldsymbol{u}} + \delta$. Applying these two inequalities to Eq. (15), we have

$$\begin{aligned}
& 1 - \frac{f^{\boldsymbol{w}}(X_S^{\boldsymbol{w}})}{f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})} \qquad (16)\\
& \leq 1 - \alpha + \alpha\left(1 - \frac{\cos(\gamma_{\boldsymbol{w}} + \delta)}{\cos(\gamma_{\boldsymbol{w}})}\right) + \frac{\cos(\theta_{\boldsymbol{u}})}{\cos(\theta_{\boldsymbol{u}} + \delta)} - 1\\
& = 1 - \alpha + \alpha(1 - \cos(\delta) + \tan(\gamma_{\boldsymbol{w}})\sin(\delta))\\
& \quad + \frac{1}{\cos(\delta) - \tan(\theta_{\boldsymbol{u}})\sin(\delta)} - 1.
\end{aligned}$$

Let $\eta = \frac{\sqrt{d-1}}{4}/\lfloor(\frac{k-d}{d})^{\frac{1}{d-1}}\rfloor$. By $\delta = 2\arcsin(\eta)$, we can derive that $\cos(\delta) = 1 - 2\eta^2$ and $\sin(\delta) = 2\eta\sqrt{1-\eta^2} \leq 2\eta$. Applying them to Eq. (16) leads to

$$\begin{aligned}
& 1 - \frac{f^{\boldsymbol{w}}(X_S^{\boldsymbol{w}})}{f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})}\\
& \leq 1 - \alpha + 2\alpha(\eta^2 + \tan(\gamma_{\boldsymbol{w}})\eta) + \frac{1}{\frac{1}{2\eta^2 + 2\tan(\theta_{\boldsymbol{u}})\eta} - 1}.
\end{aligned}$$

As the number k of selected solutions is usually much larger than the number d of objectives, η can be very small. Furthermore, $\tan(\gamma_{\boldsymbol{w}})$ and $\tan(\theta_{\boldsymbol{u}})$ are usually constantly large. Thus, we have

$$1 - \frac{f^{\boldsymbol{w}}(X_S^{\boldsymbol{w}})}{f^{\boldsymbol{w}}(X_{\mathcal{C}}^{\boldsymbol{w}})} \leq 1 - \alpha + O(\eta).$$

Because the above inequality holds for any $\boldsymbol{w}$, we have

$$\max_{\boldsymbol{w} \in \mathbb{R}_+^d, \|\boldsymbol{w}\|=1} \mathrm{rr}_{f^{\boldsymbol{w}}, \mathcal{C}}(S) \leq 1 - \alpha + O(\eta).$$

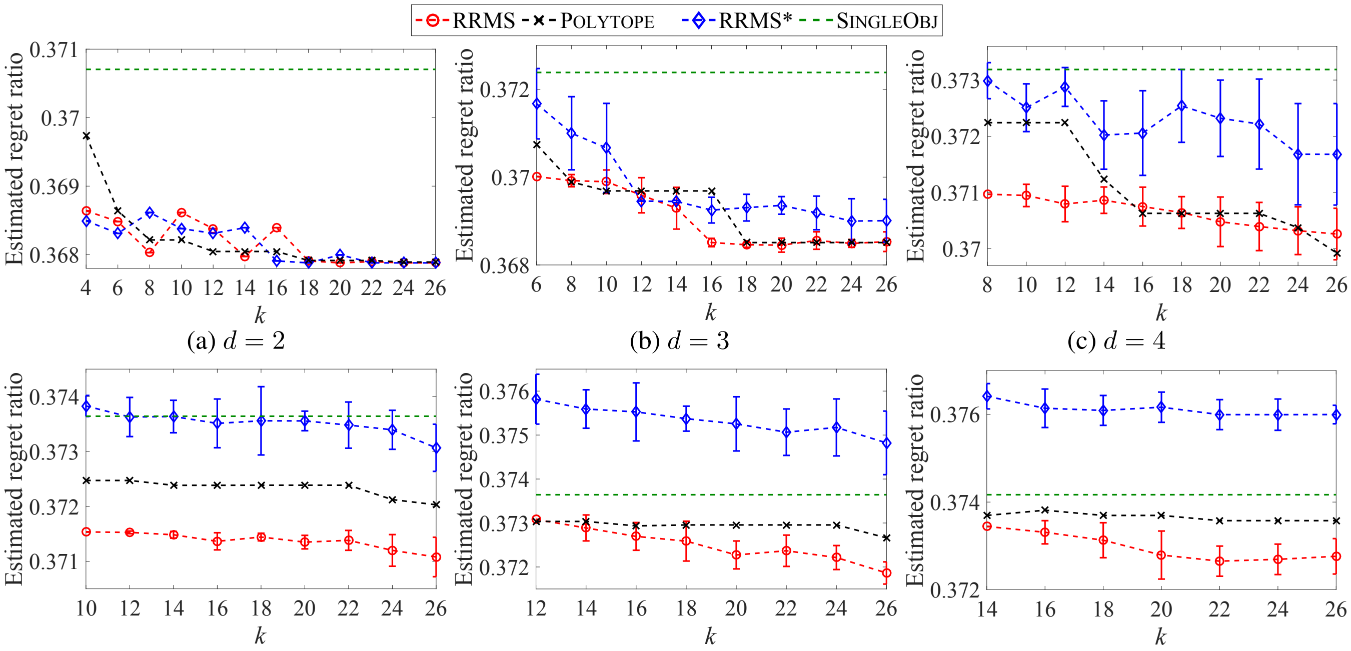
As $\eta = \frac{\sqrt{d-1}}{4}/\lfloor(\frac{k-d}{d})^{\frac{1}{d-1}}\rfloor$, the theorem holds. □

Empirical Study

In this section, we empirically compare RRMS with the only existing algorithm POLYTOPE (Soma and Yoshida 2017) on the applications of multi-objective weighted maximum coverage and Max-Cut. The number d of objectives is set from 2 to 7. For each d, the number k of selected solutions is set from $2d$ to 26 with an interval of 2.

Note that for the solution set S output by an algorithm, it is hard to directly compute the regret ratio $\mathrm{rr}_{f_1, f_2, \dots, f_d, \mathcal{C}}(S) = \max_{\boldsymbol{w} \in \mathbb{R}_+^d, \|\boldsymbol{w}\|=1} \mathrm{rr}_{f^{\boldsymbol{w}}, \mathcal{C}}(S)$, as all unit weight vectors have to be considered. Soma and Yoshida (2017) have provided a feasible way to compute it. Let $C_{\boldsymbol{f}}(S)$ denote the convex hull of $\{\boldsymbol{f}(X) \mid X \in S\}$, and let $P(S) = \{\boldsymbol{x} \in \mathbb{R}_+^d \mid \exists \boldsymbol{y} \in C_{\boldsymbol{f}}(S) : \boldsymbol{x} \leq \boldsymbol{y}\}$, where $\boldsymbol{x} \leq \boldsymbol{y}$ means $\forall i : x_i \leq y_i$. They proved that $\mathrm{rr}_{f_1, f_2, \dots, f_d, \mathcal{C}}(S) = \max_{\boldsymbol{w}} \mathrm{rr}_{f^{\boldsymbol{w}}, \mathcal{C}}(S)$, where $\boldsymbol{w}$ runs over the non-negative unit normal vectors of all frontiers of $P(S)$.

We also note that $\mathrm{rr}_{f^{\boldsymbol{w}}, \mathcal{C}}(S) = 1 - \frac{\max_{X \in S} f^{\boldsymbol{w}}(X)}{\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)}$ requires the optimal value of $f^{\boldsymbol{w}}$, i.e., $\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)$, which is usually NP-hard to be computed. As an α-approximation algorithm is given, we can apply it to obtain a solution X^* with $f^{\boldsymbol{w}}(X^*)/\alpha \geq \max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X) \geq f^{\boldsymbol{w}}(X^*)$. Soma and Yoshida (2017) used the lower bound $f^{\boldsymbol{w}}(X^*)$ to approximate $\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)$, which will, however, make the estimated regret ratio quite close to 0. In the experiments, we use the upper bound $f^{\boldsymbol{w}}(X^*)/\alpha$ to approximate.



(a) $d = 2$ (b) $d = 3$ (c) $d = 4$

(d) $d = 5$ (e) $d = 6$ (f) $d = 7$

Figure 1: Comparison among RRMS, POLYTOPE, RRMS* and SINGLEOBJ on multi-objective weighted maximum coverage with the number d of objectives from 2 to 7. Estimated regret ratio: the smaller, the better.

Multi-objective Weighted Maximum Coverage

Given a set U of elements (where each element e has a weight $w(e)$), a collection $V = \{S_1, S_2, \ldots, S_n\}$ of subsets of U, and a budget b, the weighted maximum coverage problem is to find at most b subsets from V maximizing the weighted sum of covered elements, i.e.,

$$\arg\max_{X \subseteq V} \sum_{e \in \cup_{S_i \in X} S_i} w(e) \quad \text{s.t.} \quad |X| \leq b.$$

The objective function is monotone submodular. For multi-objective weighted maximum coverage, each element e has a weight vector $\boldsymbol{w}(e) = [w_1(e), w_2(e), \ldots, w_d(e)]^{\mathrm{T}}$, and there are d objectives $f_1, f_2, \ldots, f_d$, where $\forall i : f_i(X) = \sum_{e \in \cup_{S_i \in X} S_i} w_i(e)$. The goal is to maximize these d objectives simultaneously with the feasible solution set $\mathcal{C} = \{X \mid X \subseteq V, |X| \leq b\}$. This is the dual of the well-studied multi-objective weighted set cover problem (Jaszkiewicz 2003; Lust and Tuyttens 2014; Weerasena, Wiecek, and Soylu 2017). For the single-objective approximation algorithm $\mathcal{A}$ employed by RRMS and POLYTOPE, we use the greedy algorithm, which iteratively selects one subset with the largest marginal gain. The greedy algorithm achieves the $(1-1/e)$-approximation ratio (Nemhauser, Wolsey, and Fisher 1978), and thus the parameter α in Algorithm 1 is $1 - 1/e$.

We use the real-world data set *email-Eu-core* from http://snap.stanford.edu/data/#email, which is a directed graph with 1,005 vertices and 25,571 edges. We create a set for each vertex that contains the vertex itself and its adjacent vertices. Each weight of a vertex is uniformly randomly sampled from $[0, 1]$. Here a vertex corresponds to an element. The budget b is set to 100. As RRMS has a random step (i.e., line 10 of Algorithm 2) when sampling weights, we repeat the running ten times independently and report the mean and standard deviation of the estimated regret ratio.

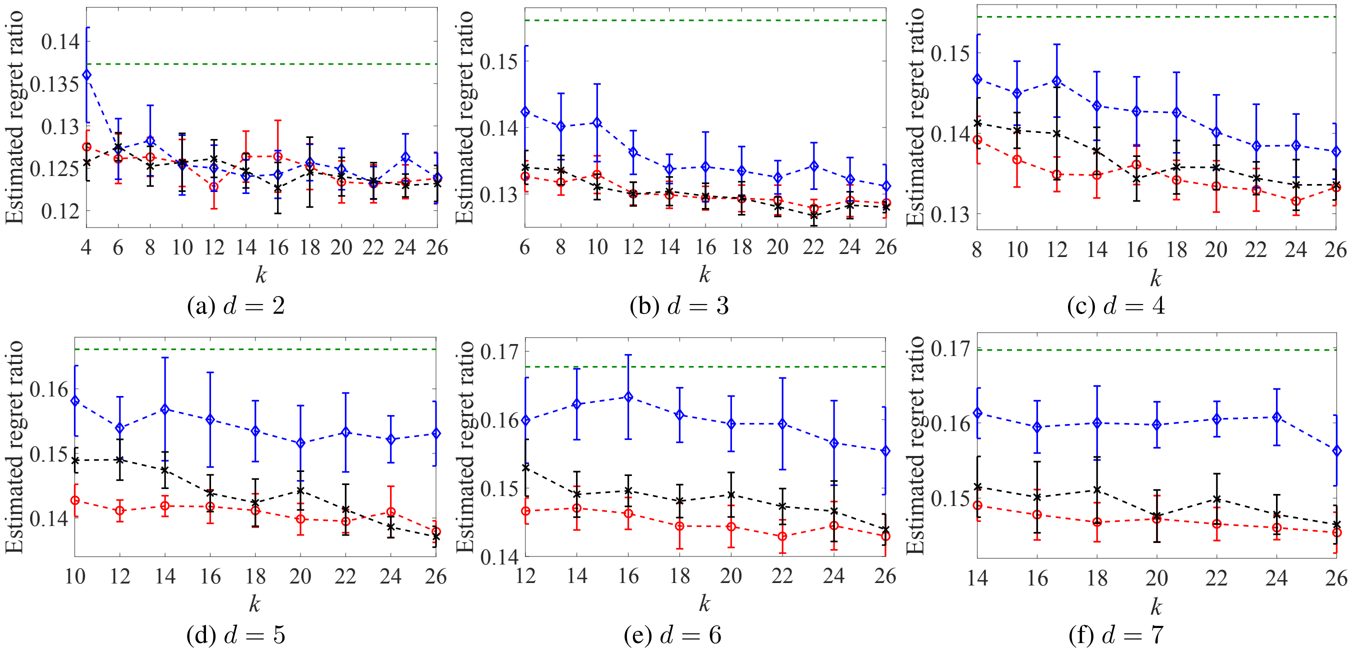
The results are plotted in Figure 1, showing that when the number d of objectives is small, RRMS and POLYTOPE perform similarly; but as d continues to increase, the advantage of RRMS becomes clear. This verifies the theoretical results that the performance of RRMS is theoretically guaranteed for any d while that of POLYTOPE is guaranteed only for $d = 2$. As d increases, maybe POLYTOPE is more likely to sample weight vectors nonuniformly, degrading its performance. We can also observe that the estimated regret ratio of both algorithms has the trend of decreasing with k but increasing with d. This is expected because more solutions are selected as k increases, and thus the approximation can be better; while the problem becomes harder as d increases.

Multi-objective Weighted Max-Cut

Given an undirected graph $G = (V, E)$, the weighted Max-Cut problem is to find a subset $X \subseteq V$ maximizing the weighted sum of edges connecting X and $V \setminus X$, i.e.,

$$\arg\max_{X \subseteq V} \sum_{e \in (X, V \setminus X)} w(e),$$

where $(X, V \setminus X)$ denotes the set of edges whose two vertices are in X and $V \setminus X$, respectively. The objective function is non-monotone submodular. For multi-objective weighted Max-Cut (Angel, Bampis, and Gourvs 2006; Bhangale et al. 2018), each edge e has a weight vector $\boldsymbol{w}(e) = [w_1(e), w_2(e), \ldots, w_d(e)]^{\mathrm{T}}$, and there are d objectives $f_1, f_2, \ldots, f_d$, where $\forall i : f_i(X) = \sum_{e \in (X, V \setminus X)} w_i(e)$. The goal is to maximize these d objectives simultaneously



(a) $d = 2$

(b) $d = 3$

(c) $d = 4$

(d) $d = 5$

(e) $d = 6$

(f) $d = 7$

Figure 2: Comparison among RRMS, POLYTOPE, RRMS* and SINGLEOBJ on multi-objective weighted Max-Cut with the number d of objectives from 2 to 7. Estimated regret ratio: the smaller, the better.

with the feasible solution set $\mathcal{C} = 2^V$. The randomized algorithm using semidefinite programming (Goemans 1995) is employed as the single-objective approximation algorithm $\mathcal{A}$, which can achieve the 0.87856-approximation ratio in expectation, and thus the parameter $\alpha = 0.87856$.

We use the real-world data set *American College football* from http://www-personal.umich.edu/~mejn/netdata/, which is an undirected graph with 115 vertices and 613 edges. Each edge weight is uniformly randomly sampled from $[0, 1]$. As the employed single-objective approximation algorithm $\mathcal{A}$ is randomized, the running of RRMS and POLYTOPE is repeated ten times independently, and we report the mean and standard deviation of the estimated regret ratio. The results in Figure 2 are similar to that observed for multi-objective weighted maximum coverage. That is, RRMS gets better as the number of objectives increases.

Discussion

When estimating the regret ratio, we pessimistically use the upper bound $f^{\boldsymbol{w}}(X^*)/\alpha$ to approximate $\max_{X \in \mathcal{C}} f^{\boldsymbol{w}}(X)$, where X^* is the solution generated by applying the α-approximation algorithm $\mathcal{A}$ to $f^{\boldsymbol{w}}$. Thus, the best regret ratio for algorithms using $\mathcal{A}$ is $1 - \alpha$, which can be achieved by applying $\mathcal{A}$ to each possible $f^{\boldsymbol{w}}$. For these two applications, the values of $(1 - \alpha)$ are $1 - (1 - 1/e) \approx 0.36788$ and $1 - 0.87856 = 0.12144$, respectively. We can see from Figures 1-2 that the regret ratios achieved by RRMS are close to them, implying that RRMS has performed quite well.

Before sampling unit weight vectors by Algorithm 2, RRMS first solves each single objective function f_i independently, as shown in lines 1–3 of Algorithm 1. Each f_i can be viewed as the linear combination $f^{\boldsymbol{w}}$ w.r.t. $\boldsymbol{w} = \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the basis vector with all entries equal to 0, except the i-th entry, which is 1. RRMS employs this process, because Algorithm 2 cannot generate these basis weight vectors, which might, however, influence the performance significantly. To verify the effectiveness of this process, we implement the RRMS* algorithm which samples k unit weight vectors directly by Algorithm 2. It can be observed from Figures 1-2 that RRMS* is almost always worse than RRMS and POLYTOPE. A natural question is then whether the solutions obtained by solving the single-objective functions $f_1, f_2, \ldots, f_d$ are sufficient. The corresponding estimated regret ratios are denoted by SINGLEOBJ in Figures 1-2. We can see that SINGLEOBJ almost always performs the worst.

Conclusion

In this paper, we study the regret ratio minimization problem in multi-objective submodular maximization. We propose the RRMS algorithm and prove that its performance is theoretically guaranteed for any number d of objectives. For $d \geq 3$, this guarantee is the first theoretical one; for $d = 2$, it reaches the guarantee of the only existing algorithm POLYTOPE. Empirical results on the applications of multi-objective weighted maximum coverage and Max-Cut show that RRMS and POLYTOPE have similar performance for small d, while RRMS performs better as d increases.

Both RRMS and POLYTOPE convert multi-objective submodular maximization into single-objective one by weighted sum. In the future, it is interesting to study the algorithms based on Pareto dominance, e.g., (Friedrich and Neumann 2015; Qian, Yu, and Zhou 2015; Qian et al. 2017, 2019), which solve the multi-objective problem directly.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2018AAA0101100), the NSFC (62022039), the Jiangsu NSF (BK20201247) and the project of HUAWEI-LAMDA Joint Laboratory of Artificial Intelligence. Chao Qian is the corresponding author.

References

Agarwal, P. K.; Har-Peled, S.; and Varadarajan, K. R. 2004. Approximating extent measures of points. *Journal of the ACM* 51(4): 606–635.

Anari, N.; Haghtalab, N.; Naor, S.; Pokutta, S.; Singh, M.; and Torrico, A. 2019. Structured robust submodular maximization: Offline and online Algorithms. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS'19)*, 3128–3137. Naha, Japan.

Angel, E.; Bampis, E.; and Gourvs, L. 2006. Approximation algorithms for the bi-criteria weighted MAX-CUT problem. *Discrete Applied Mathematics* 154(12): 1685–1692.

Bhangale, A.; Khot, S.; Kopparty, S.; Sachdeva, S.; and Thiruvenkatachari, D. 2018. Near-optimal approximation algorithm for simultaneous Max-Cut. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'18)*, 1407–1425. New Orleans, LA.

Buchbinder, N.; Feldman, M.; Seffi, J.; and Schwartz, R. 2015. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing* 44(5): 1384–1402.

Chen, W.; Lin, T.; Tan, Z.; Zhao, M.; and Zhou, X. 2016. Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, 795–804. San Francisco, CA.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2): 182–197.

Friedrich, T.; and Neumann, F. 2015. Maximizing submodular functions under matroid constraints by evolutionary algorithms. *Evolutionary Computation* 23(4): 543–558.

Gharan, S. O.; and Vondrák, J. 2011. Submodular maximization by simulated annealing. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'11)*, 1098–1116. San Francisco, CA.

Goemans, M. X. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM* 42(6): 1115–1145.

Hierons, R. M.; Li, M.; Liu, X.; Parejo, J. A.; Segura, S.; and Yao, X. 2020. Many-objective test suite generation for software product lines. *ACM Transactions on Software Engineering and Methodology* 29(1): 1–46.

Jaszkiewicz, A. 2003. Do multiple-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem. *IEEE Transactions on Evolutionary Computation* 7(2): 133–143.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, 137–146. Washington, DC.

Krause, A.; McMahan, H. B.; Guestrin, C.; and Gupta, A. 2008. Robust submodular observation selection. *Journal of Machine Learning Research* 9: 2761–2801.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9: 235–284.

Lin, H.; and Bilmes, J. A. 2011. A class of submodular functions for document summarization. In *Proceeddings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL'11)*, 510–520. Portland, OR.

Lust, T.; and Tuyttens, D. 2014. Variable and large neighborhood search to solve the multiobjective set covering problem. *Journal of Heuristics* 20(2): 165–188.

Matrosov, E. S.; Huskova, I.; Kasprzyk, J. R.; Harou, J. J.; Lambert, C.; and Reed, P. M. 2015. Many-objective optimization and visual analytics reveal key trade-offs for London's water supply. *Journal of Hydrology* 531: 1040–1053.

Nemhauser, G. L.; and Wolsey, L. A. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research* 3(3): 177–188.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14(1): 265–294.

Qian, C.; Shi, J.-C.; Tang, K.; and Zhou, Z.-H. 2017. Constrained monotone k-submodular function maximization using multiobjective evolutionary algorithms with theoretical guarantee. *IEEE Transactions on Evolutionary Computation* 22(4): 595–608.

Qian, C.; Yu, Y.; Tang, K.; Yao, X.; and Zhou, Z.-H. 2019. Maximizing submodular or monotone approximately submodular functions by multi-objective evolutionary algorithms. *Artificial Intelligence* 275: 279–294.

Qian, C.; Yu, Y.; and Zhou, Z.-H. 2013. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence* 204: 99–119.

Qian, C.; Yu, Y.; and Zhou, Z.-H. 2015. Subset selection by Pareto optimization. In *Advances in Neural Information Processing Systems 28 (NIPS'15)*, 1765–1773. Montreal, Canada.

Soma, T.; and Yoshida, Y. 2017. Regret ratio minimization in multi-objective submodular function maximization. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*, 905–911. Austin, TX.

Udwani, R. 2018. Multi-objective maximization of monotone submodular functions with cardinality constraint. In *Advances in Neural Information Processing Systems 31 (NeurIPS'18)*, 9513–9524. Montréal, Canada.

Weerasena, L.; Wiecek, M. M.; and Soylu, B. 2017. An algorithm for approximating the Pareto set of the multiobjective set covering problem. *Annals of Operations Research* 248(1-2): 493–514.

Xie, M.; Wong, R. C.-W.; Li, J.; Long, C.; and Lall, A. 2018. Efficient k-regret query algorithm with restriction-free bound for any dimensionality. In *Proceedings of the 37th International Conference on Management of Data (SIGMOD'18)*, 959–974. Houston, TX.