Tracking Regret Bounds for Online Submodular Optimization

Tatsuya Matsuoka
NEC Corporation
ta.matsuoka@nec.com

Shinji Ito
NEC Corporation
i-shinji@nec.com

Naoto Ohsaka
NEC Corporation
ohsaka@nec.com

Abstract

In this paper, we propose algorithms for online submodular optimization with tracking regret bounds. *Online submodular optimization* is a generic framework for sequential decision making used to select subsets. Existing algorithms for online submodular optimization have been shown to achieve small *(static) regret*, which means that the algorithm's performance is comparable to the performance of a fixed optimal action. Such algorithms, however, may perform poorly in an environment that changes over time. To overcome this problem, we apply a *tracking-regret*-analysis framework to online submodular optimization, one by which output is assessed through comparison with time-varying optimal subsets. We propose algorithms for submodular minimization, monotone submodular maximization under a size constraint, and unconstrained submodular maximization, and we show tracking regret bounds. In addition, we show that our tracking regret bound for submodular minimization is nearly tight.

1 INTRODUCTION

Submodularity is a property of set functions naturally arising in many machine learning applications, such as influence maximization (Kempe et al., 2003), price optimization (Ito and Fujimaki, 2016), and data summarization (Bach, 2013). A set function $f\colon 2^{[n]} \to \mathbb{R}$ is said to be *submodular* if $f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y)$ whenever $X \subseteq Y \subseteq [n]$ and $e \in [n] \setminus Y$. Optimization of submodular functions, which is a critical component in those applications, has been shown to be computationally tractable over the past few decades; in particular, submodular minimization is solvable in polynomial time, and submodular maximization is constant-factor approximable (see Section 1.2).

Online submodular optimization is a generic framework for sequential decision making, which is a (discrete) variant of online convex optimization, wherein a decision maker needs to choose actions repeatedly in the face of the uncertainty. Formally, an online algorithm sequentially observes T submodular objective functions $f_1, \dots, f_T\colon 2^{[n]} \to \mathbb{R}$, over T rounds. In each round $t \in [T]$, the algorithm must choose an action $X_t \subseteq [n]$ without observing f_t, and it is then given access to f_t and incurs loss $f_t(X_t)$. The performance of an online (submodular) minimization algorithm is often measured in terms of *(static) regret*, which is defined as the difference between the total loss of actions that the algorithm has chosen and that of the best *fixed* action in hindsight; specifically,

$$\sum_{t=1}^{T} f_t(X_t) - \min_{X^* \subseteq [n]} \sum_{t=1}^{T} f_t(X^*).$$

On submodular maximization, we adopt a different definition of regret (α*-regret*) as the offline version of maximization cannot be solved exactly in polynomial time; see Section 2. Efficient algorithms for online submodular optimization with static regret bounds have been developed (see Section 1.2 for more details).

Such online algorithms with static regret bounds, however, may perform poorly if the underlying environment is so dynamic that the best action *changes* over time. We present below two motivating examples for such a situation.

Example 1: Influence Maximization in Dynamic Networks *Influence maximization* (Kempe et al., 2003) is the problem of identifying a small set of "seed" nodes X_t in a social network that maximizes the spread of influence $f_t(X_t)$ initiated by X_t, motivated by the application to viral marketing (Domingos and Richardson, 2001). Since the influence spread

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

Table 1: Tracking regret bounds of the proposed algorithms. $\{f_t\}$: objective, $\{X_t\}$: output of the algorithm, $\{X_t^*\}$: (arbitrary) comparator sequence, EO: the time for single call of evaluation oracle. $\tilde{O}(\cdot)$ ignores polylogarithmic factors in n and T. The highlighted field is results presented in this paper.

Problem setting	Unconstrained submodular minimization	Size-constrained monotone submodular maximization	Unconstrained nonmonotone submodular maximization
Assumption	f_t: submodular	f_t: submodular, monotone	f_t: submodular
Constraint	None	Size constraint: $\lvert X_t \rvert \leq k$	None
Regret	$\sum_{t=1}^T f_t(X_t) - \sum_{t=1}^T f_t(X_t^*)$	$(1-\frac{1}{e})\sum_{t=1}^T f_t(X_t^*) - \sum_{t=1}^T f_t(X_t)$	$\frac{1}{2}\sum_{t=1}^T f_t(X_t^*) - \sum_{t=1}^T f_t(X_t)$
Algorithm	Algorithm 1	Algorithm 2	Algorithm 3
Regret bound	$\tilde{O}\left(\sqrt{T(n+P)}\right)$ (Theorem 2)	$\tilde{O}\left(\sqrt{kT(k+P)}\right)$ (Theorem 3)	$\tilde{O}\left(\sqrt{nT(n+P)}\right)$ (Theorem 4)
Time complexity	$\tilde{O}(n \cdot \mathrm{EO})$ per round	$\tilde{O}(kn \cdot \mathrm{EO})$ per round	$\tilde{O}(n \cdot \mathrm{EO})$ per round
Static regret bounds	$O(\sqrt{nT})$ (Hazan and Kale, 2012)	$\tilde{O}(\sqrt{kT})$ (Streeter and Golovin, 2008; Harvey et al., 2020)	$O(n\sqrt{T})$ (Roughgarden and Wang, 2018) $O(\sqrt{nT})$ (Harvey et al., 2020)

function f_t is submodular (Kempe et al., 2003) and the structure of the underlying network would be unknown a priori, influence maximization naturally fits into the framework of online submodular maximization. In reality, however, the optimal seed set may change drastically as time goes by (Ohsaka et al., 2016; Zhuang et al., 2013) due to the dynamic nature of social networks (Leskovec et al., 2007).

Example 2: Multi-Product Long-Term Price Optimization In the *price optimization* problem, we are given n types of products and choose the set $X_t \subseteq [n]$ that indicates which products are discounted, with the purpose of maximizing gross profit $f_t(X_t)$. Ito and Fujimaki (2016) have shown that $-f_t$ is submodular under certain assumptions, which means that the problem is captured in the framework of online submodular minimization. Hence, one can achieve small static regret using algorithms developed by Hazan and Kale (2012) and Ito (2019). Optimal prices, however, may change day to day as the demand for products is affected by such factors as the appearances of other competitive products.

1.1 Our Contributions

In this paper, we propose algorithms for online submodular optimization with tracking regret bounds. *Tracking regret* is a performance metric that makes it possible to compare an algorithm's actions to an arbitrary sequence of actions. Given a *comparator sequence* $\{X_t^*\}_{t=1}^T$, tracking regret for the minimization problem is defined as

$$\sum_{t=1}^T f_t(X_t) - \sum_{t=1}^T f_t(X_t^*).$$

For the maximization problem, *α-(approximate) tracking regret* is defined as

$$\alpha \cdot \sum_{t=1}^T f_t(X_t^*) - \sum_{t=1}^T f_t(X_t).$$

Tracking regret is so general that it includes static regret as a special case ($X_1^* = X_2^* = \cdots = X_T^*$), and it can even "track" the best actions that could change over T rounds. Tracking-regret analyses can be found in a large body of literature on online learning, including the expert problem (Herbster and Warmuth, 1998) and online convex optimization (Hall and Willett, 2013), of which details are given in Section 1.2.

We investigate three problem settings: (1) unconstrained submodular minimization, (2) size-constrained monotone submodular maximization, and (3) unconstrained submodular maximization. Table 1 summarizes tracking regret bounds of the proposed algorithms for each of them. Our regret bounds are parameterized by the *path-length* P of a comparator sequence. Intuitively speaking, P captures the degree of action change in a comparator sequence, and $P = 0$ corresponds to static regret bounds. Details regarding each problem setting are given below.

- For unconstrained submodular minimization (e.g., price optimization), we present an algorithm whose expected tracking regret is bounded by $\tilde{O}(\sqrt{T(n+P)})$. Further, we show that this bound is almost tight up to a constant.
- For size-constrained monotone submodular maximization (e.g., influence maximization), we show an online algorithm that has a tracking regret bound of $\tilde{O}(\sqrt{kT(k+P)})$ in expectation, where $k \leq n$ is a parameter specifying a size constraint.
- For unconstrained submodular maximization, we develop an online algorithm having a tracking regret bound of $\tilde{O}(\sqrt{nT(n+P)})$ in expectation.

The main ingredient of the first algorithm is a reduction to online convex optimization involving Lovász extension, while that of both the second and third algorithms is an application of algorithms for the expert problem.

1.2 Related Work

Submodular optimization has been studied over a wide range of research communities, including theoretical computer science (Fujishige, 2005; Lovász, 1983; Iwata, 2008), machine learning (Bach, 2013), and data mining (Kempe et al., 2003; Mossel and Roch, 2007; Bach, 2013). One influential work is the greedy algorithm for size-constrained monotone submodular maximization (Nemhauser et al., 1978), which efficiently finds a $(1 - 1/\mathrm{e})$-approximate solution. This approximation ratio is tight, as no polynomial-time algorithms can achieve a better approximation (Nemhauser and Wolsey, 1978; Feige, 1998). Similarly, unconstrained nonmonotone submodular maximization admits $(1/2)$-approximation algorithms (Buchbinder et al., 2015; Buchbinder and Feldman, 2018) and this ratio is optimal (Feige et al., 2011). In contrast to maximization problems, unconstrained submodular minimization can be solved exactly. The first polynomial-time algorithm was proposed by Grötschel et al. (1981). Many improvements have been made, such as a strongly polynomial-time algorithm (Grötschel et al., 1988), combinatorial strongly polynomial-time algorithms (Iwata et al., 2001; Schrijver, 2000), and fast algorithms later (Orlin, 2009; Iwata, 2003; Lee et al., 2015).

In the research area of online submodular optimization, efficient algorithms with (approximate-)static regret bounds have been found for some problem settings. For online size-constrained monotone submodular maximization, Streeter and Golovin (2008) constructed an algorithm that achieves $(1 - 1/\mathrm{e})$-approximate static regret of $\tilde{\mathrm{O}}(\sqrt{kT})$. Recently, Harvey et al. (2020) gave a $(1 - 1/\mathrm{e})$-approximate regret algorithm of $\tilde{\mathrm{O}}(\sqrt{kT})$, which improve the regret bound by Streeter and Golovin (2008) above by a factor of $\mathrm{O}(\sqrt{\log n}/\sqrt{\log(n/k)})$. In the case of unconstrained nonmonotone submodular maximization, Roughgarden and Wang (2018) proposed a $(1/2)$-approximate regret algorithm of $\mathrm{O}(n\sqrt{T})$ and recently Harvey et al. (2020) proposed a $(1/2)$-approximate regret algorithm of $\mathrm{O}(\sqrt{nT})$. For online submodular minimization, Hazan and Kale (2012) provided an algorithm with an $\mathrm{O}(\sqrt{nT})$-regret bound, and showed its optimality. Additionally, other problem settings have also been extensively studied, such as k-submodular maximization (Soma, 2019), continuous submodular optimization (DR-submodular optimization) (Chen et al., 2018a,b; Zhang et al., 2019), submodular minimization with the constraints of combinatorial structures (Jegelka and Bilmes, 2011). Also, submodular optimization (and other related function-class problems) with limited information feedback has been considered (Hazan and Kale, 2012; Streeter and Golovin, 2008; Zhang et al., 2019).

Studies on tracking regret, which were initiated by Herbster and Warmuth (1998) in the context of the expert problem (Section 3.2), have spread over a variety of problem settings, including multi-armed bandit (Auer et al., 2002), online convex optimization (Hall and Willett, 2013; Zhang et al., 2018, 2017), and online combinatorial optimization with certain structures (György et al., 2005, 2007). Further, *adaptive regret* (Hazan and Seshadhri, 2007) and *strongly adaptive regret* (Daniely et al., 2015) are closely related to tracking regret. In the expert problem, for example, bounds for strongly adaptive regret lead to those for tracking regret. Such connections, however, do not directly apply to the problem settings in which path-lengths are defined with respect to a metric over action spaces, e.g., convex optimization and the problems discussed in this paper.

2 PROBLEM SETTINGS

A player is given T and n, which represent the number of rounds and the size of the underlying set, respectively. In each round $t \in [T]$, the player (randomly) chooses a subset $X_t \subseteq [n]$ of the underlying set $[n] := \{1, 2, \dots, n\}$. After choosing X_t, the player gets *full-information feedback* regarding the objective function $f_t \colon 2^{[n]} \to [0, 1]$; i.e., can observe $f_t(X)$ for any $X \subseteq [n]$ after choosing X_t. Objective functions f_t are here assumed to be submodular and the adversary decides f_t in each round t. Note that the adversary only knows the strategy of the player and does not know what action X_t does the player take until the decision of f_t.

The goal of the player is to minimize/maximize $\sum_{t=1}^{T} f_t(X_t)$, the sum of the values of objective functions. In this paper, we consider three different problem settings for online submodular optimization: (unconstrained nonmonotone) submodular minimization, size-constrained monotone submodular maximization, and unconstrained nonmonotone submodular maximization.

Online Submodular Minimization In online submodular minimization, there is no constraint, i.e., the player can choose an arbitrary subset X_t of $[n]$. The performance of the player is evaluated by means of *tracking regret* defined as

$$R_T(\{X_t^*\}_{t=1}^T) = \sum_{t=1}^T f_t(X_t) - \sum_{t=1}^T f_t(X_t^*) \qquad (1)$$

for an arbitrary *comparator sequence* $\{X_t^*\}_{t=1}^T \subseteq [n]$.

Online Submodular Maximization In online size-constrained monotone submodular maximization, the player is given a parameter $k \in [n]$ before the game starts, and the available actions are constrained so that any chosen subset has a size of at most k, i.e., $|X_t| \leq k$. In this setting, objective functions are assumed to be *monotone*, i.e., $f_t(X) \leq f_t(Y)$ hold for any $X, Y \subseteq [n]$ satisfying $X \subseteq Y$. By way of contrast, in the problem setting of unconstrained nonmonotone submodular maximization, there is no constraint and the objective can be nonmonotone. In both problem settings, the performance of the player is measured in terms of *α-(approximate) tracking regret* defined as

$$R_T^\alpha(\{X_t^*\}_{t=1}^T) = \alpha \sum_{t=1}^T f_t(X_t^*) - \sum_{t=1}^T f_t(X_t), \qquad (2)$$

where the parameter $\alpha > 0$ corresponds to the approximation ratios.[1] In the size-constrained monotone setting (Section 4.2), we set $\alpha = (1 - \frac{1}{\mathrm{e}})$. In the unconstrained nonmonotone setting (Section 4.3), we set $\alpha = \frac{1}{2}$. In the size-constrained setting, the comparator sequence $\{X_t^*\}_{t=1}^T$ is assumed to be constrained as well, i.e., $|X_t^*| \leq k$.

For a comparator sequence $\{X_t^*\}_{t=1}^T$, we define its *path-length* P by

$$P = \sum_{t=1}^{T-1} |X_t^* \triangle X_{t+1}^*|, \qquad (3)$$

where $X \triangle Y$ represents the symmetric difference of X and Y, i.e., $X \triangle Y = (X \setminus Y) \cup (Y \setminus X)$. Intuitively speaking, the value of P measures the length of the path that the comparator X_t^* moves along over T rounds. In the special case of $P = 0$, tracking regrets defined in (1) and (2) correspond, respectively, to the static regret and the α-static regret. In our analysis, we provide bounds for tracking regrets parametrized with the path-lengths P.

3 PRELIMINARIES

This section introduces certain known techniques that are key ingredients in our proposed algorithms.

[1]Since offline counterparts are computationally hard, we employ these approximate regrets.

3.1 Lovász Extension

In the construction of the algorithm for submodular minimization, we reduce the problem to (continuous) convex optimization via a technique referred to as *Lovász extension* (Lovász, 1983), which is defined as follows: For $x \in [0,1]^n$ and $u \in [0,1]$, denote $H_u(x) = \{i \in [n] \mid x_i \geq u\}$. Given a submodular function $f\colon 2^{[n]} \to \mathbb{R}$, define the Lovász extension $\tilde{f}\colon [0,1]^n \to \mathbb{R}$ of f by

$$\tilde{f}(x) = \mathbb{E}_{u \sim \mathrm{Unif}([0,1])}[f(H_u(x))], \qquad (4)$$

where $\mathrm{Unif}([0,1])$ represents a uniform distribution over $[0,1]$. This $\tilde{f}$ is indeed an extension of f as it holds for all $X \subseteq [n]$ that $\tilde{f}(\chi(X)) = f(X)$, where $\chi(X) \in \{0,1\}^n$ represents the indicator vector of X.

Given a permutation $\sigma\colon [n] \to [n]$ over $[n]$, denote $S_\sigma(i) := \{\sigma(j) \mid j \in [i]\}$. If $x \in [0,1]^n$ satisfies $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \cdots \geq x_{\sigma(n)}$, Lovász extension can be expressed as

$$\tilde{f}(x) = f(\emptyset) + \sum_{i=1}^n (f(S_\sigma(i)) - f(S_\sigma(i-1))) x_{\sigma(i)}.$$

Given a submodular function $f\colon 2^{[n]} \to \mathbb{R}$ and a permutation $\sigma\colon [n] \to [n]$, define $g(\sigma) \in \mathbb{R}^n$ as follows:

$$g(\sigma) = \sum_{i=1}^n (f(S_\sigma(i)) - f(S_\sigma(i-1))) \chi(\sigma(i)), \qquad (5)$$

where $\chi(i) \in \{0,1\}^n$ is the indicator vector, i.e., $\chi(i)_i = 1$ and $\chi(i)_j = 0$ for $j \in [n] \setminus \{i\}$. If $x \in [0,1]^n$ satisfies $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \cdots \geq x_{\sigma(n)}$, then $g(\sigma)$ is a subgradient of $\tilde{f}$ at x. Since $\tilde{f}$ is a convex function (Lovász, 1983), we have

$$\tilde{f}(y) - \tilde{f}(x) \geq g(\sigma)^\top (y - x)$$

for $x, y \in [0,1]^n$ and a permutation $\sigma\colon [n] \to [n]$ such that $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \cdots \geq x_{\sigma(n)}$.

Lemma 1. *If $f\colon 2^{[n]} \to \mathbb{R}$ is a submodular function, $g(\sigma)$ defined by (5) satisfies $\|g(\sigma)\|_1 \leq 4 \max_{X \subseteq [n]} |f(X)|$ for any permutation σ.*

For the proof of this lemma, see, e.g., Lemma 8 of Hazan and Kale (2012).

3.2 Algorithms for the Expert Problem

When designing the algorithms for online submodular maximization, we use techniques for the *expert problem*, a fundamental problem in online optimization.

In the expert problem, a player is given a number T of rounds and a number m of actions before the game starts. In each round $t \in [T]$, the player chooses a distribution $p_t \in \Delta^m := \{p \in [0,1]^m \mid \|p\|_1 = 1\}$ over $[m]$, and the environment then reveals the loss vector $\ell_t = (\ell_{t1}, \ell_{t2}, \ldots, \ell_{tm})^\top \in [-1,1]^m$, for which the player suffers a loss of $\ell_t^\top p_t$.

The *fixed share forecaster* (Herbster and Warmuth, 1998) is an algorithm for the expert problem and can be described as follows: Set a parameter $\gamma > 0$ and $\beta \in [0,1]$. Initialize the *weight vector* $w_t \in \mathbb{R}^m_{>0}$ by $w_1 = \mathbf{1}$. The algorithm outputs a distribution proportional to w_t, i.e., $p_t = \frac{w_t}{\|w_t\|_1}$. In each round t, the weight vector w_t is updated on the basis of the observation of ℓ_t as follows:

$$v_{ti} = w_{ti} \exp(-\gamma \ell_{ti}) \qquad (i \in [m]), \qquad (6)$$

$$w_{t+1,i} = \beta \frac{W_t}{m} + (1-\beta) v_{ti} \qquad (i \in [m]), \qquad (7)$$

where $W_t = \sum_{i=1}^n v_{ti}$. The special case of the fixed share forecaster in which parameter β is chosen as $\beta = 0$ is called the *multiplicative weight update (MWU)* method. The fixed share forecaster achieves the following performance:

Theorem 1 (Herbster and Warmuth (1998)). *Suppose that p_t is given by $p_t = \frac{w_t}{\|w_t\|_1}$ with (6) and (7). We denote $P' = |\{t \in [T-1] \mid i_t^* \neq i_{t+1}^*\}|$. Then, for any sequence $\{i_t^*\}_{t=1}^T \subseteq [m]$,*

$$\sum_{t=1}^T \left(\ell_t^\top p_t - \ell_{ti_t^*}\right) \le \gamma T + \frac{1}{\gamma}\left((2P'+1)\log m + \log \frac{1}{\beta^{P'}(1-\beta)^{T-P'-1}}\right)$$

holds.

From Theorem 1, we can achieve a tracking regret of $O(\sqrt{P'T\log(mT)})$ for the expert problem if the parameters β and γ can be tuned depending on P'. Such an approach of parameter tuning is, however, impossible as P' is an arbitrary integer in $[T-1]$ and is not given in advance. This issue can be resolved by a strategy of maintaining multiple learning rates, which can be found in the literature (van Erven and Koolen, 2016; Zhang et al., 2018). In this strategy, we run multiple $O(\log T)$ fixed share forecaster algorithms with different parameters, and combine their outputs using multiplicative weight update method. A more specific description of the algorithm is given in Algorithm 4 in the appendix (Section A). We call this algorithm as FSF*.

Corollary 1. *There is an algorithm* FSF* *(in appendix) for the expert problem such that*

$$\sum_{t=1}^T \left(\ell_t^\top p_t - \ell_{ti_t^*}\right) \le 8\sqrt{T\left((P'+1)\log(mT) + \log(1+\log T)\right)}$$

holds for any sequence $\{i_t^*\}_{t=1}^T \subseteq [m]$, where P' is defined by $P' = |\{t \in [T-1] \mid i_t^* \neq i_{t+1}^*\}|$.

The computational time of FSF* (in Corollary 1) is of $O(m \log T)$ per round. Proof of the above corollary is given in the appendix (Section B.1).

Note that the online submodular optimization problem setting is a special case of the expert problem, but if we apply an algorithm for the expert problem like MWU it takes exponential time (the regret is polynomial). Thus, by utilizing the submodularity, we construct polynomial-time algorithms, with the assumption that we can call evaluation oracles.

4 PROPOSED ALGORITHMS

4.1 Online Submodular Minimization

In this subsection, we introduce an algorithm for online submodular minimization. The proposed algorithm is described in Algorithm 1. Below, we show that Algorithm 1 enjoys the following regret bound:

Theorem 2. *If $\{X_t\}_{t=1}^T$ is produced by Algorithm 1, it holds for any comparator sequence $\{X_t^*\}_{t=1}^T \subseteq 2^{[n]}$ that*

$$\mathbb{E}\left[R_T(\{X_t^*\}_{t=1}^T)\right] \le 16\sqrt{T\left(n + P + \log(5+\log T)\right)},$$

where R_T and P are defined in (1) and (3), respectively, and $\mathbb{E}[\cdot]$ represents expectation taken with respect to the algorithm's internal randomness.

Reduction to Online Linear Optimization Let $\tilde{f}_t$ denote the Lovász extension of f_t defined as in (4). In the proposed algorithm, we maintain real vectors $x_t \in [0,1]^n$, and output a subset $X_t(H_{u_t}(x_t)) = \{i \in [n] \mid x_{ti} \ge u_t\}$ in each round, where u_t is chosen from a uniform distribution over $[0,1]$. Then from (4), we have

$$\mathbb{E}[f_t(X_t)] = \mathbb{E}\left[f_t(H_{u_t}(x_t))\right] = \mathbb{E}\left[\tilde{f}_t(x_t)\right]. \qquad (8)$$

From this, for any $\{X_t^*\}_{t=1}^T \subseteq 2^{[n]}$,

$$\mathbb{E}\left[\sum_{t=1}^T f_t(X_t) - \sum_{t=1}^T f_t(X_t^*)\right] = \mathbb{E}\left[\sum_{t=1}^T \tilde{f}_t(x_t) - \sum_{t=1}^T \tilde{f}_t(x_t^*)\right] \le \mathbb{E}\left[\sum_{t=1}^T g_t^\top (x_t - x_t^*)\right] \qquad (9)$$

Algorithm 1 Algorithm for online submodular minimization with full-information

```
Require: The number $T$ of rounds and the size $n$ of the ground set.
 1: Set $d = \lceil \log_2 T \rceil + 4$ and let $p_1 = \frac{1}{d}\mathbf{1} \in \Delta^d$ and $w_1 = \mathbf{1} \in \mathbb{R}^d$. Set $\gamma = \sqrt{\frac{\log d}{T}}$. For each $j \in [d]$, initialize $x_t^{(j)}$ by $x_1^{(j)} = \mathbf{0} \in \mathbb{R}^n$.
 2: for $t = 1, 2, \ldots, T$ do
 3:   Set $x_t = \sum_{j=1}^d p_{tj} x_t^{(j)}$.
 4:   Pick $u_t$ from a uniform distribution over $[0,1]$ and output $X_t = H_{u_t}(x_t) = \{i \in [n] \mid x_{ti} \geq u_t\}$.
 5:   Get feedback of $f_t$ and compute $g_t$, a subgradient of $\tilde{f}_t$ at $x_t$ given by (5).
 6:   for $j = 1, 2, \ldots, d$ do
 7:     Compute $x_{t+1}^{(j)}$ as in (10), where $\eta^{(j)} = \sqrt{\frac{n}{2^j}}$.
 8:   end for
 9:   Compute $p_{t+1} = \frac{w_{t+1}}{\|w_{t+1}\|_1}$, where $w_t$ is updated as in (6) and (7) with $m = d$, $\beta = 0$, and $\ell_{tj} = g_t^\top x_t^{(j)}/4$ for $j \in [d]$.
10: end for
```

holds, where $x_t^* \in \{0,1\}^n$ is the indicator vector of X_t^*, and g_t represents a subgradient of $\tilde{f}_t$ at x_t. In (9), the equality follows from (8), and the inequality follows from the fact that $\tilde{f}_t$ is convex.

Online Gradient Descent Online gradient descent (OGD) method is expressed by the following:

$$x_{t+1} \in \operatorname*{argmin}_{x \in [0,1]^n} \|x - (x_t - \eta g_t)\|_2^2, \tag{10}$$

where $\eta > 0$ is a parameter referred to as learning rate. Then x_t satisfies the lemma below:

Lemma 2. *Suppose that x_t is given by* (10). *Then, for any $\{x_t^*\}_{t=1}^T \subseteq [0,1]^n$, we have*

$$\sum_{t=1}^T g_t^\top x_t - \sum_{t=1}^T g_t^\top x_t^* \leq \frac{\eta}{2}\sum_{t=1}^T \|g_t\|_2^2 + \frac{n}{2\eta} + \frac{1}{\eta}\sum_{t=1}^{T-1} \|x_t^* - x_{t+1}^*\|_1.$$

Proof of the above lemma is in the appendix (Section B.2).

Adaptive Learning Rate via MWU From Lemmas 1 and 2, by running OGD with parameter $\eta = \sqrt{\frac{n+\sum_{t=1}^T \|x_t^* - x_{t+1}^*\|_1}{T}}$, we obtain

$$\sum_{t=1}^T g_t^\top x_t - \sum_{t=1}^T g_t^\top x_t^* = \mathrm{O}\left(\sqrt{T(n+P)}\right). \tag{11}$$

However, we here need to choose parameter η depending on $\sum_{t=1}^T \|x_t^* - x_{t+1}^*\|_1$, the path-length of the comparator sequence. This means that (11) holds only for comparator sequences $\{x_t^*\}_{t=1}^T$ with a *fixed* path-length.

To achieve (11) for all comparator sequences with *arbitrary* path-lengths, we run OGD algorithms with different learning rates in parallel, and combine them using the multiplicative weight update method.

Denote $d = \lceil \log_2 T \rceil + 4$. Set $\eta^{(j)} = \sqrt{n 2^{-j}}$ for $j = 1, 2, \ldots, d$. For each j, update $x_t^{(j)} \in [0,1]^n$ using OGD with learning rate $\eta^{(j)}$. Set x_t as $x_t = \sum_{j=1}^d p_{tj} x_t^{(j)}$, where $p_t \in \Delta^d$ is given by multiplicative weight update method, with $\ell_{tj} = g_t^\top x_t^{(j)}/4$ and $\gamma = \sqrt{\frac{\log d}{T}}$, i.e., p_t is defined as $p_t = \frac{w_t}{\|w_t\|_1}$ where w_t is updated by (6) and (7) with $\beta = 0$.

Proof of Theorem 2. From Theorem 1, we have

$$\sum_{t=1}^T g_t^\top x_t - \sum_{t=1}^T g_t^\top x_t^{(j)} = 4\left(\sum_{t=1}^T \ell_t^\top p_t - \sum_{t=1}^T \ell_{tj}\right) \leq 4\gamma T + \frac{4\log d}{\gamma} \leq 8\sqrt{T \log d}$$

for all $j \in [d]$, where the first equality follows from the definition of ℓ_{tj}, the inequality follows from Theorem 1 with $\beta = 0$ and $P' = 0$, and the last equality follows from $\gamma = \sqrt{\frac{\log d}{T}}$. Further, from Lemmas 2 and 1, we have

$$\sum_{t=1}^T g_t^\top x_t^{(j)} - \sum_{t=1}^T g_t^\top x_t^* \leq 8\eta^{(j)} T + \frac{n}{2\eta^{(j)}} + \frac{1}{\eta^{(j)}}\sum_{t=1}^{T-1} \|x_t^* - x_{t+1}^*\|_1.$$

Combining the above two inequalities and (9), we obtain

$$\mathbb{E}\left[\sum_{t=1}^T f_t(X_t) - \sum_{t=1}^T f_t(X_t^*)\right] \leq 8\eta^{(j)} T + \frac{n}{2\eta^{(j)}} + \frac{1}{\eta^{(j)}}\sum_{t=1}^{T-1} \|x_t^* - x_{t+1}^*\|_1 + 8\sqrt{T\log d}$$

for all $j \in [d]$. From the definition of $\eta^{(j)} = \sqrt{\frac{n}{2^j}}$, there exists $j \in [d]$ such that

$$\frac{\eta^{(j)}}{\sqrt{2}} \leq \sqrt{\frac{n + 2\sum_{t=1}^{T-1} \|x_t^* - x_{t+1}^*\|_1}{16T}} \leq \eta^{(j)}.$$

Hence, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} f_t(X_t) - \sum_{t=1}^{T} f_t(X_t^*)\right] \\
&\leq \sqrt{32T\left(n + 2\sum_{t=1}^{T-1}\|x_t^* - x_{t+1}^*\|_1\right)} + 8\sqrt{T\log d} \\
&= \sqrt{32T\,(n+2P)} + 8\sqrt{T\log(\lceil \log_2 T\rceil + 4)},
\end{aligned}$$

where the equality follows from the definition of x_t^* (indicator vector of X_t^*) and the definition of d. □

4.2 Online Size-Constrained Monotone Submodular Maximization

The proposed algorithm for online size-constrained monotone submodular maximization is based on the algorithms for the expert problem introduced in Section 3.2. More specifically, the algorithm runs k instances of FSF* (in Corollary 1) in parallel and chooses X_t on the bases of the outputs of these instances. The procedure is summarized in Algorithm 2. In each round t, after getting the outputs $p_t^{(1)}, p_t^{(2)}, \ldots, p_t^{(k)}$ from the k copies of FSF* (in Corollary 1), the algorithm constructs the output X_t as

$$X_t = \{i_{t1}, i_{t2}, \ldots, i_{tk}\},$$

where i_{ts} follows $p_t^{(s)}$, i.e., $i_{ts} = i$ with probability $p_{ti}^{(s)}$ for each $s \in [k]$. Then the algorithm feed the loss vector $\ell_t^{(s)} \in [-1,1]^n$ to the s-th copy FSF$^{*(s)}$, where $\ell_t^{(s)}$ is defined by

$$X_{ts} = \{i_{t1}, i_{t2}, \ldots, i_{ts}\} \qquad (s \in [k]), \tag{12}$$

$$\ell_{ti}^{(s)} = f_t(X_{ts}) - f_t(X_{ts} \cup \{i\}) \quad (s \in [k], i \in [n]). \tag{13}$$

Let us next show that the proposed algorithm enjoys the regret bound described below.

Theorem 3. *Suppose that $\{X_t\}_{t=1}^T$ is produced by Algorithm 2. Then, for any comparator sequence $\{X_t^*\}_{t=1}^T \subseteq 2^{[n]}$ such that $|X_t^*| = k$, we have*

$$\mathbb{E}\left[R_T^{(1-1/\mathrm{e})}(\{X_t^*\}_{t=1}^T)\right] = \tilde{\mathrm{O}}\left(\sqrt{kT\,(k+P)}\right),$$

where $R_T^{(1-1/\mathrm{e})}$ and P are defined in (2) and (3), respectively, and where $\mathbb{E}[\cdot]$ represents the expectation taken with respect to the algorithm's internal randomness.

To show Theorem 3, we use the following lemma, which connects the approximate tracking regret defined in (2) and the bounds in Corollary 1.

Algorithm 2 Algorithm for online monotone submodular maximization under size constraint

```
Require: The number $T$ of rounds, the size $n$ of the ground set, the size-constrained parameter $k$ such that $1 \leq k < n$.
 1: Initialize $k$ copies FSF$^{*(1)}$, FSF$^{*(2)}, \ldots,$ FSF$^{*(k)}$ of FSF$^*$ (in Corollary 1) with parameters $T$ and $m = n$.
 2: for $t = 1, 2, \ldots, T$ do
 3:   Set $X_{t0} = \emptyset$.
 4:   for $s = 1, 2, \ldots, k$ do
 5:     Get the $t$-th output $p_t^{(s)}$ from FSF$^{*(s)}$.
 6:     Draw an item $i_{ts}$ from the distribution $p_t^{(s)}$.
 7:     Set $X_{ts}$ by $X_{ts} := X_{t,s-1} \cup \{i_{ts}\}$.
 8:   end for
 9:   Output $X_t = X_{tk}$ and get feedback of $f_t$.
10:   for $s = 1, 2, \ldots, k$ do
11:     Set $\ell_t^{(s)} = (\ell_{t1}^{(s)}, \ell_{t2}^{(s)}, \ldots, \ell_{tn}^{(s)})$ by $\ell_{ti}^{(s)} = f(X_{t,s-1}) - f(X_{t,s-1} \cup \{i\})$ for each $i \in [n]$.
12:     Feed $\ell_t^{(s)}$, as the $t$-th input, to FSF$^{*(s)}$.
13:   end for
14: end for
```

Lemma 3. *For each $s \in [k]$, denote*

$$B_T^{(s)} = \sum_{t=1}^{T}\left(k\ell_{ti_{ts}}^{(s)} - \sum_{i \in X_t^*} \ell_{ti}^{(s)}\right), \tag{14}$$

where $\ell_{ti}^{(s)}$ is defined with (12) and (13). Then, for any $\{X_t\}_{t=1}^T$ such that $|X_t^*| = k$ for all $t \in [T]$, the regret defined in (2) is bounded as

$$R_T^{(1-1/\mathrm{e})}(\{X_t^*\}_{t=1}^T) \leq \frac{1}{k}\sum_{s=1}^{k}\left(1 - \frac{1}{k}\right)^{k-s} B_T^{(s)}.$$

This can be shown via the assumption that each f_t is a monotone submodular function. The proof of this lemma is given in the appendix (Section B.3).

Proof of Theorem 3. From Lemma 3, it suffices to give bounds for each $B_T^{(s)}$ defined in (14). For any sequence $\{X_t^*\}_{t=1}^T$ such that $|X_t^*| = k$, we can choose $\{i_{ts}^*\}_{t\in[T], s\in[k]} \subseteq [n]$ so that $X_t^* = \{i_{t1}^*, i_{t2}^*, \ldots, i_{tk}^*\}$ and

$$P_j' := |\{t \in [T-1] \mid i_{tj}^* \neq i_{t+1,j}^*\}|$$

satisfies

$$2\sum_{j=1}^{k} P_j' = P = \sum_{t=1}^{T-1} |X_t^* \triangle X_{t+1}^*|. \tag{15}$$

In fact, (15) holds when we choose $\{i_{ts}^*\}_{t\in[T], s\in[k]}$ so that $i_{t+1,s}^* = i_{ts}^*$ for each $s \in X_{t+1}^* \cap X_t^*$. Hence, for each $s \in [k]$, since $p_t^{(s)}$ are the output of FSF*,

$$\begin{aligned}\mathbb{E}\left[B_T^{(s)}\right] &= \sum_{j=1}^{k} \mathbb{E}\left[\sum_{t=1}^{T}(\ell_t^{(s)\top} p_t^{(s)} - \ell_{ti_{tj}^*}^{(s)})\right] \\ &\leq 8\sum_{j=1}^{k}\sqrt{T\left((P'_j+1)\log(nT)+\log(1+\log T)\right)} \\ &\leq 8\sqrt{kT\sum_{j=1}^{k}\left((P'_j+1)\log(nT)+\log(1+\log T)\right)} \\ &\leq 8\sqrt{kT\left(P+k\log(nT)+k\log(1+\log T)\right)},\end{aligned}$$

where the first inequality follows from Corollary 1, the second comes from the Cauchy–Schwarz inequality, and the last inequality follows from (15). By combining this bound for $B_T^{(s)}$ and Lemma 3, we obtain the regret bound in Theorem 3. □

4.3 Online Unconstrained Nonmonotone Submodular Maximization

The proposed algorithm for online unconstrained nonmonotone submodular maximization relies on Algorithm 4, in a way similar to that in which Algorithm 2 relies on it. Its procedure is summarized in Algorithm 3. The algorithm runs n copies $\mathrm{FSF}^{*(1)}, \mathrm{FSF}^{*(2)}, \ldots, \mathrm{FSF}^{*(n)}$ of FSF* (in Corollary 1), each of which is used to decide if any individual element $s \in [n]$ should be included in X_t or not. Each of n copies solves an expert problem with two actions, i.e., $m = 2$.

We show that Algorithm 3 achieves the regret bound described below.

Theorem 4. *Suppose that $\{X_t\}_{t=1}^T$ is produced by Algorithm 3. Then, for any comparator sequence $\{X_t^*\}_{t=1}^T \subseteq 2^{[n]}$, we have*

$$\mathbb{E}\left[R_T^{(1/2)}(\{X_t^*\}_{t=1}^T)\right] = \tilde{\mathrm{O}}\left(\sqrt{nT\,(n+P)}\right),$$

where $R_T^{(1/2)}$ and P are defined in (2) and (3), respectively, and $\mathbb{E}[\cdot]$ represents the expectation taken with respect to the algorithm's internal randomness.

The lemma below plays an essential role in our proof of Theorem 4.

Lemma 4. *For the output of Algorithm 3 and any comparator sequence $\{X_t^*\}_{t=1}^T \subseteq 2^{[n]}$, we have*

$$\mathbb{E}\left[R_T^{(1/2)}(\{X_t^*\}_{t=1}^T)\right] \leq$$

$$\frac{1}{2}\sum_{s=1}^{n}\mathbb{E}\left[\sum_{t\in[T]}\ell_t^{(s)\top}p_t^{(s)} - \sum_{t\in[T]:s\in X_t^*}\ell_{t1}^{(s)} - \sum_{t\in[T]:s\notin X_t^*}\ell_{t2}^{(s)}\right],$$

where $\ell_t^{(s)}$ and $p_t^{(s)}$ are as defined in Algorithm 3.

This lemma can be shown via the diminishing returns property of submodular functions and the technique of reduction from Blackwell's approachability problem to regret minimization (Abernethy et al., 2011; Soma, 2019). A complete proof of this lemma can be found in the appendix (Section B.4).

Proof of Theorem 4. Set $i_{ts}^* = 1$ if $s \in X_{ts}$ and $i_{ts}^* = 2$ if $s \notin X_{ts}$ for each $t \in [T]$ and $s \in [n]$. The values P'_s defined by $P'_s := |\{t \in [T-1] \mid i_{ts}^* \neq i_{t+1,s}^*\}|$ then satisfy $\sum_{s=1}^n P'_s = P$. From this and the fact that $p_j^{(s)}$ are the output from FSF* (in Corollary 1), we have

$$\begin{aligned}&\sum_{s=1}^{n}\left(\sum_{t\in[T]}\ell_t^{(s)\top}p_t^{(s)} - \sum_{t\in[T]:s\in X_t^*}\ell_{t1}^{(s)} - \sum_{t\in[T]:s\notin X_t^*}\ell_{t2}^{(s)}\right) \\ &= \sum_{s=1}^{n}\left(\sum_{t\in[T]}\ell_t^{(s)\top}p_t^{(s)} - \sum_{t\in[T]}\ell_{ti_{ts}^*}^{(s)}\right) \\ &\leq 8\sum_{s=1}^{n}\sqrt{T\left((P'_s+1)\log(nT)+\log(1+\log T)\right)} \\ &\leq 8\sqrt{nT\left(P+n\log(nT)+n\log(1+\log T)\right)},\end{aligned}$$

where the first inequality follows from Corollary 1 and the second comes from the Cauchy–Schwarz inequality and $\sum_{s=1}^n P'_s = P$. From this and Lemma 4, we obtain the regret bound in Theorem 4.

4.4 Time Complexity of Our Algorithms

The time complexity of all three algorithms in this paper are polynomial in n, T, and EO, which represents the time for single call of evaluation oracle. Indeed, each round $t \in [T]$ requires computational time polynomial in n and EO as written in Table 1.

5 LOWER BOUND

The regret bound given in Theorem 2 achieved by Algorithm 1 is nearly tight up to a constant. In fact, we can show the following regret lower bound:

Theorem 5. *For any T, P, n such that $0 \leq P \leq nT$ and any algorithm for online submodular minimization there is a sequence of submodular functions $f_t\colon 2^{[n]} \to [0,1]$ and a comparator sequence $\{X_t^*\}_{t=1}^T$ such that $\sum_{t=1}^{T-1}|X_t^* \triangle X_{t+1}^*| \leq P$, and the following holds:*

$$\mathbb{E}\left[R_T(\{X_t^*\}_{t=1}^T)\right] = \Omega(\min\{\sqrt{T(n+P)}, T\}).$$

Proof of the above theorem is written in appendix (Section B.5).

Algorithm 3 Algorithm for online unconstrained nonmonotone submodular maximization

```
Require: The number $T$ of rounds, the size $n$ of the base set.
 1: Initialize $n$ copies $\mathrm{FSF}^{*(1)}, \mathrm{FSF}^{*(2)}, \ldots, \mathrm{FSF}^{*(n)}$ of $\mathrm{FSF}^*$ (in Corollary 1) with parameters $T$ and 2.
 2: for $t = 1, 2, \ldots, T$ do
 3:     Set $X_{t0} = \emptyset$ and $Y_{t0} = [n]$.
 4:     for $s = 1, 2, \ldots, n$ do
 5:         Get the $t$-th output $p_t^{(s)}$ from $\mathrm{FSF}^{*(s)}$, and set $q_t^{(s)} = \frac{1}{4}(1 + 2p_{t1}^{(s)})$.
 6:         With probability $q_t^{(s)}$, set $X_{ts} := X_{t,s-1} \cup \{s\}$ and $Y_{ts} := Y_{t,s-1}$. Otherwise, (with probability $(1 - q_t^{(s)})$,) set $X_{ts} := X_{t,s-1}$ and $Y_{ts} := Y_{t,s-1} \setminus \{s\}$.
 7:     end for
 8:     Output $X_t := X_{tn} = Y_{tn}$ and get feedback of $f_t$.
 9:     for $s = 1, 2, \ldots, n$ do
10:         Set $a_{ts} = f_t(X_{t,s-1} \cup \{s\}) - f_t(X_{t,s-1})$ and $b_{ts} = f_t(Y_{t,s-1} \setminus \{s\}) - f_t(Y_{t,s-1})$.
11:         Set $\ell_t^{(s)} = (\ell_{t1}^{(s)}, \ell_{t2}^{(s)})^\top$ by $\ell_{t1}^{(s)} = -(1 - q_t^{(s)})a_{ts}$ and $\ell_{t2}^{(s)} = -q_t^{(s)} b_{ts}$.
12:         Feed $\ell_t^{(s)}$, as the $t$-th input, to $\mathrm{FSF}^{*(s)}$.
13:     end for
14: end for
```

6 CONCLUSION

In this paper, we considered the tracking regret bounds for three online submodular optimization problems; (i) (unconstrained) submodular minimization, (ii) size-constrained monotone submodular maximization, and (iii) unconstrained nonmonotone submodular maximization. For the above problems, we gave (i) tracking regret bound of $\tilde{\mathrm{O}}(\sqrt{T(n+P)})$, (ii) $(1-1/\mathrm{e})$-tracking regret of $\tilde{\mathrm{O}}(\sqrt{kT(k+P)})$, and (iii) 1/2-tracking regret of $\tilde{\mathrm{O}}(\sqrt{nT(n+P)})$, respectively. Also, for (i), we gave the lower bound $\Omega(\min\{\sqrt{T(n+P)}, T\})$, which means that tracking regret bound $\tilde{\mathrm{O}}(\sqrt{T(n+P)})$ is tight if we ignore the polynomial terms of logarithmics of T and n when we focus on order and coefficient of polynomials of T.

As future work directions, other themes related to both tracking regret bounds and (generalization or variant of) submodularity can be considered, such as tracking regret bounds of online optimization for k-submodular functions, other functions with the generalized property of submodularity, or DR-submodular functions.

Acknowledgements

The authors are grateful to the anonymous reviewers for their comments. The second author is supported by JST, ACT-I, Grant Number JPMJPR18U5, Japan.

References

J. Abernethy, P. L. Bartlett, and E. Hazan. Blackwell approachability and no-regret learning are equivalent. In *Conference on Learning Theory*, pages 27–46, 2011.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

N. Buchbinder and M. Feldman. Deterministic algorithms for submodular maximization problems. *ACM Transactions on Algorithms*, 14(3):32:1–20, 2018.

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

L. Chen, C. Harshaw, H. Hassani, and A. Karbasi. Projection-free online optimization with stochastic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, pages 814–823, 2018a.

L. Chen, H. Hassani, and A. Karbasi. Online continuous submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 1896–1905, 2018b.

A. Daniely, A. Gonen, and S. Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning*, pages 1405–1411, 2015.

P. Domingos and M. Richardson. Mining the network value of customers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

U. Feige, V. S. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

S. Fujishige. *Submodular Functions and Optimization, Volume 58, 2nd Edition.* Elsevier, 2005.

M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

A. György, T. Linder, and G. Lugosi. Tracking the best of many experts. In *International Conference on Computational Learning Theory*, pages 204–216, 2005.

A. György, T. Linder, G. Lugosi, and G. Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8:2369–2403, 2007.

E. Hall and R. Willett. Dynamical models and tracking regret in online convex programming. In *International Conference on Machine Learning*, pages 579–587, 2013.

N. Harvey, C. Liaw, and T. Soma. Improved algorithms for online submodular maximization via first-order regret bounds. In *Advances in Neural Information Processing Systems*, 2020.

E. Hazan and S. Kale. Online submodular minimization. *Journal of Machine Learning Research*, 13: 2903–2922, 2012.

E. Hazan and C. Seshadhri. Adaptive algorithms for online decision problems. In *Electronic Colloquium on Computational Complexity*, volume 14, 2007.

M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.

S. Ito. Submodular function minimization with noisy evaluation oracle. In *Advances in Neural Information Processing Systems*, pages 12103–12113, 2019.

S. Ito and R. Fujimaki. Large-scale price optimization via network flow. In *Advances in Neural Information Processing Systems*, pages 3855–3863, 2016.

S. Iwata. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32(4):833–840, 2003.

S. Iwata. Submodular function minimization. *Mathematical Programming*, 112(1):45, 2008.

S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4): 761–777, 2001.

S. Jegelka and J. Bilmes. Online submodular minimization for combinatorial structures. In *International Conference on Machine Learning*, pages 345–352, 2011.

D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

Y. T. Lee, A. Sidford, and S. C. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Symposium on Foundations of Computer Science*, pages 1049–1065, 2015.

J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1):2, 2007.

L. Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

E. Mossel and S. Roch. On the submodularity of influence in social networks. In *ACM Symposium on Theory of Computing*, pages 128–134, 2007.

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

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

N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi. Dynamic influence analysis in evolving networks. *Proceedings of the VLDB Endowment*, 9(12):1077–1088, 2016.

J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

T. Roughgarden and J. R. Wang. An optimal learning algorithm for online unconstrained submodular maximization. In *Conference on Learning Theory*, pages 1307–1325, 2018.

A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2): 346–355, 2000.

T. Soma. No-regret algorithms for online k-submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 1205–1214, 2019.

M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 1577–1584, 2008.

T. van Erven and W. M. Koolen. MetaGrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems*, pages 3666–3674, 2016.

L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou. Improved dynamic regret for non-degenerate functions.

In *Advances in Neural Information Processing Systems*, pages 732–741, 2017.

L. Zhang, S. Lu, and Z.-H. Zhou. Adaptive online learning in dynamic environments. In *Advances in Neural Information Processing Systems*, pages 1323–1333, 2018.

M. Zhang, L. Chen, H. Hassani, and A. Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In *Advances in Neural Information Processing Systems*, pages 9210–9221, 2019.

H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun. Influence maximization in dynamic social networks. In *IEEE International Conference on Data Mining*, pages 1313–1318, 2013.