Regularized Online DR-Submodular Optimization

Abstract

The utilization of online optimization techniques is prevalent in many fields of artificial intelligence, enabling systems to continuously learn and adjust to their surroundings. This paper outlines a regularized online optimization problem, where the regularizer is defined on the average of the actions taken. The objective is to maximize the sum of rewards and the regularizer value while adhering to resource constraints, where the reward function is assumed to be DR-submodular. Both concave and DR-submodular regularizers are analyzed. Concave functions are useful in describing the impartiality of decisions, while DR-submodular functions can be employed to represent the overall effect of decisions on all relevant parties. We have developed two algorithms for each of the concave and DR-submodular regularizers. These algorithms are easy to implement, efficient, and produce sublinear regret in both cases. The performance of the proposed algorithms and regularizers has been verified through numerical experiments in the context of internet advertising.

1 INTRODUCTION

Online optimization encompasses a broad range of scenarios where information is disclosed incrementally and decisions must be made at each step, despite the uncertain future. It has many practical applications in the fields of computer science and operations research, e.g., [Auer et al., 2002, Buchbinder et al., 2007, Mehta et al., 2007, Zinkevich, 2003, Buchbinder and Naor, 2009], among others. For instance, consider an online advertising placement scenario where website visitors are exposed to different types of Ads. Visitors come to the website in a sequential manner, and in each round, the website allocates varying amounts of visitors to each ad type with the purpose of maximizing the number of clicks on those Ads.

In traditional online convex optimization, the majority of existing studies define the total reward simply as the sum of rewards obtained at each step. However, certain applications may benefit from utilizing a more sophisticated utility function to evaluate the solution, instead of just summing up individual rewards. To this end, we propose regularized online optimization, a variation that incorporates a non-linear regularizer that is defined on the average of the actions taken. This enables the capturing of extra characteristics, such as fairness, that are desired in the final outcome. In our problem, the decision maker selects an action in each round and receives a reward based on the reward function of that round. The objective is to maximize the total of the accumulated reward and the regularizer over a specified finite time frame.

While efficient solutions exist for convex optimization problems, there is a growing number of non-convex problems present in the fields of machine learning and statistics. For instance, continuous DR-submodular function[Bian et al., 2017a,b] is a rich subclass of non-convex/non-concave functions, capturing a variety of real-world applications, such as optimal experiment design, non-definite quadratic programming, coverage and diversity functions, and continuous relaxation of discrete submodular functions[Bian et al., 2020]. The combination of continuous DR-submodular and concave functions is commonly found in ML applications, such as maximizing a regularized submodular function[Kazemi et al., 2021] and finding the mode of distributions[Kazemi et al., 2021, Robinson et al., 2019]. In this paper, we shall study online optimization problems with a regularizer. We mainly focus on the case where the reward function is a DR-submodular function, and the regularizer can be either concave or DR-submodular. To the best of our knowledge, we are the first to investigate the use of DR-submodular functions as regularizers in the field of online optimization.

Our primary focus is on investigating online optimization problems that are subject to resource constraints, and the

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

budget B grows at least linearly in the time horizon T. In our study, we denote the budget as $B = \rho T$ where $\rho \in [0, 1]$. When $\rho = 1$, there are no constraints on the budget. The main contribution of this paper can be summarized as follows:

- We first examine the optimization problem with a concave regularizer in an online setting. To address this problem, we introduce the Dual Online Non-oblivious Gradient Ascent algorithm. We demonstrate that our proposed algorithm attains $O(\sqrt{T})$-$(\rho(1 - \frac{1}{e}), \rho)$-regret. Notably, when $\rho = 1$, our algorithm achieves the optimal approximation ratio of $(1 - \frac{1}{e}, 1)$.
- Then we consider the online optimization problem with DR-submodular regularizer. We propose the Online Stochastic Frank-Wolfe for this problem. We prove that the proposed algorithm achieves $O(\sqrt{T})$-$(\frac{\rho}{e^\rho}, \frac{\rho}{e^\rho})$-regret.
- To demonstrate the practicality and efficacy of our solutions, we offer several examples of application scenarios. Additionally, we perform a series of experiments to thoroughly evaluate the effectiveness of our proposed methods.

All missing proofs and materials are moved to supplementary materials.

2 RELATED WORK

Online submodular maximization. Consider a online monotone DR-submodular maximization problem. [Chen et al., 2018b] proposed the Meta-Frank-Wolfe algorithm for this problem and achieves a $(1 - 1/e)$ approximation factor of the best fixed offline solution in hindsight up to an $O(\sqrt{T})$ regret term. The Meta-Frank-Wolfe need the full information of the function's gradient. And [Chen et al., 2018a] develop a projection-Free algorithm which gets the same approximation factor and regret order where only stochastic gradient estimates are available. More recently, [Zhang et al., 2022] have proposed an auxiliary function to boost the approximation ratio of the offline and online gradient ascent algorithms from $1/2$ to $1 - 1/e$.

Online submodular optimization in the i.i.d. model. When the utility functions are drawn i.i.d from a unknown distribution $f \sim \mathcal{D}$, [Chen et al., 2018a] proposed a simple algorithm for stochastic online optimization which requires only a single stochastic gradient estimate in each round. This algorithm achieves a $O(T^{2/3})$ regret. Then [Sadeghi et al., 2021] improved this result to $O(\sqrt{T})$.

Regularized optimization In a recent study, [Mitra et al., 2021] examined the maximization of functions that are composed of a continuous DR-submodular function and a concave function. They proposed multiple algorithms for various offline settings. Separately, [Balseiro et al., 2021] studied online allocation problems with a concave regularizer, their model is capable of capturing extra objectives like fairness.

3 PRELIMINARIES

We use bold letters, such as $\mathbf{x}$, to denote a vector, and x_i is the ith entry of $\mathbf{x}$. Given two vectors $\mathbf{x}$ and $\mathbf{y}$, the notation $\mathbf{x} \leq \mathbf{y}$ implies that $x_i \leq y_i$ for all i. We use the symbol ∇ to denote the gradient of a function. Given two vectors $\mathbf{u}$ and $\mathbf{v}$, $\langle \mathbf{u}, \mathbf{v} \rangle$ is the inner product of these two vectors. $\| \cdot \|$ is the ℓ_2 norm in Euclidean space. We define the projection to the domain $\mathcal{P}$ as $\Pi_{\mathcal{P}}(\mathbf{x}) = \arg\min_{\mathbf{y} \in \mathcal{P}} \|\mathbf{x} - \mathbf{y}\|$

The DR-submodular function encompasses many real-life scenarios of diminishing returns, and represents a generalization of submodular set functions in the continuous domain. We say that f is continuous DR-submodular [Bian et al., 2017a, Calinescu et al., 2011] if f is differentiable and

$$\nabla f(\mathbf{x}) \geq \nabla f(\mathbf{y})$$

for all $\mathbf{x} \leq \mathbf{y}$. A noteworthy characteristic of continuous DR-submodular functions is that they are concave in positive directions; that is, for all $\mathbf{x} \leq \mathbf{y}$,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle.$$

A function f is monotone if $f(\mathbf{x}) \leq f(\mathbf{y})$ for all $\mathbf{x} \leq \mathbf{y}$. A function f is L-smooth if $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y}$.

A typical online learning protocol operates as follows: In each iteration t (ranging from 1 to T), the algorithm chooses an action $\mathbf{x}_t \in \mathcal{P} \subseteq \mathbb{R}^n$, where d is the dimension of the action. Upon making the selection, the reward function $f_t \in \mathbb{R}^n \to \mathbb{R}_+$ is revealed and the algorithm obtains a reward of $f_t(x_t)$. In the traditional setting where there is no regularizer, the objective is to minimize the difference between the total reward accumulated by the algorithm and that of the best fixed decision made with hindsight. Our study not only examines the scenario where f_t is a concave function but also explores the case when f_t is a monotone DR-submodular function. It is important to note that, even in offline settings, maximizing a monotone DR-submodular function subject to a convex constraint can only be approximately solved in polynomial time unless $\mathbf{RP} = \mathbf{NP}$[Bian et al., 2017b]. As a result, we define the α-regret[Kakade et al., 2007, Streeter and Golovin, 2008] of an algorithm as follows:

$$\alpha\text{-}\mathcal{R}_T \triangleq \alpha \max_{\mathbf{x} \in \mathcal{P}} \sum_{t=1}^{T} f_t(\mathbf{x}) - \sum_{t=1}^{T} f_t(\mathbf{x}_t)$$

where α represents the optimal approximation ratio of an offline solution.

4 PROBLEM FORMULATION

We consider the following online optimization problem with a finite horizon of T time periods, resource constraints ρT, and a regularizer r:

$$\max_{x:\mathbf{x}_t \in \mathcal{P}} \sum_{t=1}^{T} f_t\left(\mathbf{x}_t\right) + Tr\left(\frac{1}{T}\sum_{t}^{T}\mathbf{x}_t\right) \quad s.t. \sum_{t}^{T} c_t(\mathbf{x}_t) \leq \rho T \tag{1}$$

where $c_t \in \mathbb{R}^n \to [0,1]$ is the resources consumption function at time t, $\rho \in [0,1]$ is the per-iteration budget, $B = \rho T$ is the total budget, r is a known regularizer which is defined on the average of the actions taken. Although we only consider a single resource constraint in this study, extending the model to accommodate multiple resource constraints is a straightforward task. Here, $c_t(\mathbf{x})$ is transformed into an m-dimensional vector, with m denoting the number of resources, while ρ also becomes an m-dimensional vector.

Note that the regularizer term is not an additively separable objective. The existence of such non-separable regularizer makes the theoretical analysis much harder than the one without the regularizer. We can find this form of regularizer or reward function in [Balseiro et al., 2021, Agrawal and Devanur, 2014a,b]. As for the regularizer, let $\boldsymbol{x}_t^* \in \mathcal{P}$ denote the the optimal offline solution that maximizes the function r of the average of these vectors. We denote the value of this solution as OPT_r. We can define the following average regret measures:

$$\text{avg-regre}(T) \triangleq \text{OPT}_r - r(\frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t)$$

This notation was adopted in [Agrawal and Devanur, 2014a,b] where they obtain an average regret bound of $O(\sqrt{\frac{1}{T}})$ in their settings. However, in general online learning, the emphasis is placed on the total regret at the end of the entire time horizon T, rather than the time average regret. To align the two forms of regret, we use $Tr\left(\frac{1}{T}\sum_t^T \mathbf{x}_t\right)$ as our regularizer, instead of $r\left(\frac{1}{T}\sum_t^T \mathbf{x}_t\right)$.

Analogous to the previously introduced α-regret, the α, β-regret of an algorithm for the given problem can be defined as:

$$\alpha, \beta{-}\mathcal{R}_T \triangleq \left(\alpha \sum_{t=1}^{T} f_t(\mathbf{x}^*) + \beta Tr\left(\mathbf{x}^*\right)\right) - \left(\sum_{t=1}^{T} f_t\left(\mathbf{x}_t\right) + Tr\left(\frac{1}{T}\sum_{t}^{T}\mathbf{x}_t\right)\right)$$

where $\mathbf{x}^*$ is the optimal fixed action for Problem (1). We introduce two approximation ratios α and β to account for the fact that optimizing the reward function f_t and the regularizer r might have different levels of intrinsic difficulty.

Remark: Note that in our study, we consider the resource constraints as hard, meaning they must be strictly followed, and no violation is allowed. Therefore, our focus is only on the regret of the objective function, rather than considering both the objective function and the constraint violation. This approach is in contrast to studies such as [Agrawal and Devanur, 2014b], where the resource constraints are treated as soft and violations are allowed, requiring an additional regret term to quantify the degree of constraint violation.

Thus far, we have considered a adversarial setting where no assumptions are made about the generation of reward functions. Another commonly utilized setting is the stochastic setting, in which the functions are assumed to be independently and identically distributed (i.i.d) from an unknown distribution $f_t \sim \mathcal{D}$. In this case, the goal is to minimize the stochastic regret defined as

$$\alpha, \beta{-}\mathcal{SR}_T \triangleq \left(\alpha T f(\mathbf{x}^*) + \beta Tr\left(\mathbf{x}^*\right)\right) - \left(\sum_{t=1}^{T} f_t\left(\mathbf{x}_t\right) + Tr\left(\frac{1}{T}\sum_{t}^{T}\mathbf{x}_t\right)\right)$$

where $f(\mathbf{x}) = \mathbb{E}_{f_t \sim \mathcal{D}}\left[f_t(\mathbf{x})\right]$ represents the expected function. This framework is commonly used in many machine learning and statistical applications, such as empirical risk minimization, where the objective is unknown but can be estimated through sampled data points and labels. Typically, the stochastic setting admits more effective algorithms as compared to the adversarial setting.

5 MAIN RESULTS

We shall now introduce algorithmic results. (i) For the case of concave regularizer, we consider a adversarial setting and assume that the reward function revealed in each round is a DR-submodular function. (ii) For the more complicated case of DR-submodular regularizer, we consider a stochastic setting and examine two sub-cases in which the reward function in each round is either a DR-submodular function or a concave function.

5.1 CONCAVE REGULARIZER

In this subsection , we consider the case where the reward function f_t is a monotone DR-submodular function, and the regularizer r is a concave function. Applications such as ad allocation can significantly benefit from the capability of maximizing a concave reward function. While the reward in most pay-per-click advertising systems is a simple linear function, the actual situation is more complex. Advertisers (in display advertising) often consider the requirement for a specific mix of different demographics, such as an equal

number of male and female clicks or clicks from different cities. These are not strict constraints; the closer it is to the ideal mix, the better. We will now provide some examples of the regularizer.

Example 1. (Max-min fairness) The regularizer is defined as $r(\mathbf{x}) = \lambda \min_i x_i$. If we view $\mathbf{x}$ as an ad resource allocation method, this regularizer represents the minimum amount of visitors allocated among all types of advertisements. This guarantees that certain forms of advertising will not receive too few resources. Here, λ is a parameter.

Example 2. (Santa Claus Regularizer[Bansal and Sviridenko, 2006]) This can be considered as another form of max-min fairness. Unlike the goal of distributing resources evenly, we desire that each advertiser's actual reward will be fair. If we consider the reward function $f_t(\mathbf{x}) = \mathbf{q}_t^T \mathbf{x}$, then the regularizer can be $r(\mathbf{x}) = \lambda \min_i (q_t^T x)_i$.

Example 3. (Online joke recommendation with max-min fairness regularizer) The aim of this problem is to design a joke recommendation algorithm that assigns jokes to a sequence of online users in a way that maximizes their overall impression within a fixed time horizon $B_T = \Omega(T)$. At each step $t \in [T]$, a user arrives and the algorithm should assign up to m jokes, represented as $\mathbf{x}_t \in \left\{\mathbf{x} \in \{0,1\}^n : \mathbf{1}^T \mathbf{x} \leq m\right\}$. If joke i is assigned to user t, they will spend $p_{t(i)}$ time reading it and submit a rating of $r_{t(i)}$. The overall impression is represented by the submodular set function $f_t(\mathbf{x}) = \mathbf{r}_t^T \mathbf{x} + \sum_{i,j:i<j} \theta_{t(ij)} x_i x_j$, where $\theta_{t(ij)} \leq 0$ is used to discourage similarity between jokes i and j in order to promote diversity. To handle continuous scenarios, $\mathbf{x}_t$ is relaxed to $\mathbf{x}_t \in \left\{\mathbf{x} \in [0,1]^n : \mathbf{1}^T \mathbf{x} \leq m\right\}$ and treated as the probability of each joke being selected in each round, making $f_t(\mathbf{x})$ a DR-submodular function. However, there is a fairness problem that arises when two similar jokes, A and B, are recommended. Assuming that joke A is more appealing than joke B when recommended separately, this may result in a significantly decreased probability of recommending joke B due to their similarity, unfairly favoring joke A. To mitigate this issue, a max-min fairness regularizer can be employed to guarantee that the probability of joke B being recommended is not too small. This ensures fairness, not by requiring each round's selection to meet specific criteria, but by taking into account the average probability in the entire recommendation process. Therefore, the final form of this problem is

$$\begin{aligned} \max_{\mathbf{x}:\mathbf{x}_t \in \mathcal{P}} \sum_{t=1}^{T} (\mathbf{r}_t^T \mathbf{x}_t + \sum_{i,j:i<j} \theta_{t(ij)} x_{t(i)} x_{t(j)}) + \\ T\lambda \min_i \left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t \right)_i \\ s.t. \sum_t^T \langle \mathbf{p}_t, \mathbf{x}_t \rangle \leq B_T = \rho T. \end{aligned}$$

Assumption 1. We make the following assumptions on the utility function and regularizer:

1. f_t is a monotone DR-submodolar function, $c_t \in [0,1]$ is a convex function and r is concave function.
2. $f_t(\mathbf{0}) = r(\mathbf{0}) = c_t(\mathbf{0}) = 0$

When r is a concave function, using the Jensen's inequality, we can get

$$r(\frac{1}{T} \sum_{t=1}^{T} \mathbf{x}_t) \geq \frac{1}{T} \sum_{t=1}^{T} r(\mathbf{x}_t).$$

Hence, we can get a lower bound on the optimal solution of Problem (1) by solving the following problem

$$\begin{aligned} \max_{x:\mathbf{x}_t \in \mathcal{P}} \sum_{t=1}^{T} \left(f_t\left(\mathbf{x}_t\right) + r(\mathbf{x}_t)\right) \\ s.t. \sum_t^T c_t(\mathbf{x}_t) \leq \rho T. \end{aligned} \tag{2}$$

This problem can be regraded as a new online optimization problem: at iteration t, after taking the action, we get $f_t(\mathbf{x}) + r(\mathbf{x})$ as the new reward function, which is formed by the sum of a DR-submodular function and a concave function. Given that the optimal solution of Problem (2) is a lower bound of that of Problem (1), so any algorithm that has a α, β-regret of Problem (2) is also a α, β-regret algorithm of Problem (1).

The algorithm we propose builds upon the work of [Castiglioni et al., 2022], which leverages the classic primal-dual approach commonly used in online problems with packing constraints. Our method can be seen as solving two online problems simultaneously. Specifically, we derive a Lagrangian function for the original problem at each iteration. We consider this function to be the primal problem:

$$\mathcal{L}_t^P = f_t(\mathbf{x}) + r(\mathbf{x}) - \langle \lambda_t, c_t(\mathbf{x}) \rangle \tag{3}$$

At the same time, we have a corresponding dual problem

$$\mathcal{L}_t^D = -\langle \lambda, \rho - c_t(\mathbf{x}_t) \rangle \tag{4}$$

At each iteration, we apply the projected gradient method to tackle the two problems. For Problem (4), the feasible domain of λ can be $\mathbb{R}$. However, as we will demonstrate in the proof of Theorem 1, restricting λ to $\left[-\frac{1}{\rho}, \frac{1}{\rho}\right]$ is sufficient for the algorithm's performance. The algorithm terminates when either the agent exhausts their budget or the time horizon T concludes.

The dual problem of the two problems is a linear optimization problem, which can be easily solved. However, the primal problem is a sum of a concave function and a continuous DR-submodular function, which requires a new approach to solve. The theoretical guarantee of gradient ascent

Algorithm 1 Dual Online Non-oblivious Gradient Ascent

Input: parameters $B_1, T, \{\eta_t\}_1^T, \{\gamma_t\}_1^T$
Initialization: $\forall i \in [m], B_1, \rho \leftarrow B_1/T$
1: **for** $t \leftarrow 1, 2, 3..., T$ **do**
2: **Primal decision**:

$$\tilde{\mathbf{x}}_t = \Pi_{\mathcal{P}}\left(\mathbf{x}_{t-1} + \eta_{t-1}\tilde{\nabla}\mathcal{L}_{t-1}^P\left(\mathbf{x}_{t-1}\right)\right)$$
$$\mathbf{x}_t \leftarrow \begin{cases} \tilde{\mathbf{x}}_t & \text{if} B_t \geq 1 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

3: **Dual decision:**

$$\lambda_t \leftarrow \Pi_{\mathcal{D}}\left(\lambda_{t-1} + \gamma_{t-1}\nabla\mathcal{L}_t^D(\lambda_{t-1})\right)$$

4: **Observe request**: observe f_t, c_t and update available resources: $B_{t+1} \leftarrow B_t - c_t(\mathbf{x}_t)$
5: **Primal update**

$$\mathcal{L}_t^P = f_t(\mathbf{x}) + r(\mathbf{x}) - \langle\lambda_t, c_t(\mathbf{x})\rangle$$
$$\nabla\tilde{\mathcal{L}_t^P} = \nabla\left(F_t(\mathbf{x}) + r(\mathbf{x}) - \langle\lambda_t, c_t(\mathbf{x})\rangle\right)$$

6: **Dual update** $\mathcal{L}_t^D = -\langle\lambda, \rho - c_t(\mathbf{x}_t)\rangle$
7: **end for**

methods applied to concave functions is based on a key property that characterizes concavity: if g is a concave function, then $g(\mathbf{y}) - g(\mathbf{x}) \leq \langle\nabla g(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle$. Fortunately, there is a similar property that holds for monotone DR-submodular functions, which is presented in Lemma 1.

Lemma 1 *[Bian et al., 2017a] Let $f : \mathcal{P} \to \mathbb{R}_+$be a monotone DR-submodular function. For any two vector $\mathbf{x}, \mathbf{y} \in \mathcal{P}$, we have*

$$\frac{1}{2}f(\mathbf{y}) - f(\mathbf{x}) \leq \langle\nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle.$$

But if we use the original function f_t directly for gradient descent, we can only achieve a $\frac{1}{2}$ approximation ratio for α. Therefore, we introduce some auxiliary functions, called non-oblivious functions, to obtain a better approximation ratio.

Lemma 2 *Let $f : \mathcal{P} \to R^+$ is a monotone, differentiable, DR-submodular function, and $F(\mathbf{x}) = \int_0^1 \frac{e^{z-1}}{z} f(z * \mathbf{x})dz$ is the non-oblivious function of f. For any vector $\mathbf{x}, \mathbf{y} \in \mathcal{P}$, we have*

$$(1 - e^{-1})f(\mathbf{y}) - f(\mathbf{x}) \leq \langle\nabla F(\mathbf{x}), \mathbf{x} - \mathbf{y}\rangle.$$

Lemma 2, which is essential in obtaining the optimal approximation ratio of $1 - \frac{1}{e}$, can be derived from [Zhang et al., 2022]. The key idea in the gradient descent method for DR-submodular functions is to use the gradient of non-oblivious functions instead of the gradient of the original function for computation. The advantage of Lemma 2 is that it is also compatible with the gradient descent method for concave functions, allowing for the handling of combinations of concave functions and continuous DR-submodular functions with ease.

Now we can provide the regret bound of Algorithm 1.

Theorem 1 *Let $\eta_t = \frac{1}{\sqrt{t}}, \gamma_t = \frac{1}{\sqrt{t}}$ and $\mathbf{x}_t : 1 \leq t \leq T$ be the choices of Algorithm 1, then we have*

$$(\rho(1 - \frac{1}{e}), \rho) - \mathcal{R} = O(\sqrt{T})$$

where $\rho \in [0, 1]$ is the resource constraint ratio.

Notably, when $\rho = 1$, which means there is no resource constraints (since $c_t(x) \in [0, 1]$), we get the optimal $(1 - \frac{1}{e}, 1)$ ratio for this problem.

The proof of Theorem 1 can be divided into three main steps. First, we obtain an approximate regret for the primal problem. Then, in the second step, we derive an approximate regret for the dual problem. Finally, we combine the outcomes of these two steps to establish a bound on the regret of the entire problem, thus proving Theorem 1.

Calculating the gradient of non-oblivious functions $F(\mathbf{x})$ can be challenging. To overcome this, an approximation in the form of $G(\mathbf{x}) = \varepsilon \cdot \sum_{j=1}^{\varepsilon^{-1}} \frac{e^{\varepsilon j} \cdot f(\varepsilon j \cdot \mathbf{x})}{\varepsilon j}$ is used to calculate the gradient of $F(\mathbf{x})$. This method, however, results in a loss of approximation α. When only unbiased estimates of the gradient are available, a calculation approach to obtain a unbiased estimate of the gradient of $F(x)$ is presented in [Zhang et al., 2022] through sampling. With only the availability of the stochastic gradient, we can get the results consistent with the above results, in expectation.

5.2 DR-SUBMODULAR REGULARIZER

In this section, we consider the scenario where the Regularizer is a monotone DR-submodular function. The property of submodularity, representing the concept of diminishing returns, frequently appears in various real-world situations. We will begin by providing a motivating example to illustrate this problem.

Online Advertising with Influence Maximization In this problem, we consider how advertisers place online advertisements. We assume that there are currently n channels for advertising and that advertisers have budget constraints. In each round, we must allocate our budget wisely in order to maximize the rewards we receive. Formally, in each round t,

we make a decision $\mathbf{x}_t \in \{\mathbf{1}^T\mathbf{x} \leq B\}$, and then we get a reward $f_t(\mathbf{x}_t)$. For simplicity, we assume the reward function is a linear function $f_t(\mathbf{x}_t) = \mathbf{r}_t^T\mathbf{x}_t$. This reward function can be utilized to represent the click-through rate. However, in addition to the click-through rate, other metrics are often considered when it comes to advertising. We can model the input of this problem as a bipartite graph $(S, G; W)$, where S and G are sets of advertising channels and customer groups, respectively. The edge weight $p_{sg} \in W$ represents the probability of group g being activated by one unit of investment in channel s. p can be obtained from the previous browsing data of the website. $(1-p_{sg})^{x_s}$ is the probability that group g will not be activated when we have x_s investment in channel s. The overall influence customer group g received from all channels can be modeled by a proper monotone DR-submodular function $I_g(\mathbf{x})$[Bian et al., 2020], such as $I_g(\mathbf{x}) = 1 - \prod_{(s,g)\in W}(1-p_{sg})^{x_s}$, where $\mathbf{x} \in \mathbb{R}_+^S$ represents the budget allocation among the advertising channels. Therefore, the final form of this problem is

$$\max_{\mathbf{x}_t \in \{\mathbf{1}^T\mathbf{x} \leq B\}} \sum_t^T \mathbf{r}_t^T\mathbf{x}_t + T\lambda \sum_{g\in G}\left(1 - \prod_{(s,g)\in W}(1-p_{sg})^{\frac{1}{T}\sum_{t=1}^T x_{t(s)}}\right)$$

In the previous section, we employed Jensen's inequality for concave functions to transform Problem (1) into Problem (2). Then, we only needed to develop an algorithm for Problem (2). However, Jensen's inequality does not always hold for a monotone DR-submodular function. Therefore, we can only establish a weaker property for a monotone DR-submodular function.

Proposition 1. (Jensen's inequality for DR-Submodular Function) Let f be a continuous monotone DR-submodular function on set $\mathcal{P}$. If $\mathbf{x_1} \leq \mathbf{x_2} \leq \cdots \leq \mathbf{x_n} \in \mathcal{P}$, and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^n \lambda_i = 1$, we have

$$f(\sum_{i=1}^n \lambda_i \mathbf{x_i}) \geq \sum_i^n \lambda_i f(\mathbf{x_i})$$

Proposition 1 implies that if we can find an algorithm for Problem (2), and the output value of the algorithm in each round meets the partial order condition of Proposition 1, then the algorithm is an efficient solution for Problem (1) as well.

In the stochastic online setting, where functions are i.i.d. sampled as $f_t \sim \mathcal{D}$, it is possible to design an algorithm that satisfies the conditions specified in Proposition 1. To start, we make the following assumptions regarding the utility functions and regularizer:

Assumption 2.

1. $\mathcal{P} \subset \mathbb{R}_+^n$ is a down-closed convex set.
2. $f(\mathbf{0}) = f_t(\mathbf{0}) = r(\mathbf{0}) = 0$.
3. There exists $\sigma > 0$ such that for any $x \in \mathcal{P}$ and $t \in [T]$, $\|\nabla f_t(\mathbf{x}) - \nabla f(\mathbf{x})\|_2 \leq \sigma$ holds.
4. f is monotone DR-submodular and L-smooth.
5. r is monotone DR-submodular and L-smooth.
6. f_t is L-smooth.

Algorithm 2 Online Stochastic Frank-Wolfe

Input: convex set $\mathcal{P}, T, \mathbf{x}_1 = \mathbf{0}$, step sizes $\{\eta_t\}$, parameters B_1, T
Output: $\mathbf{x}_t : 1 \leq t \leq T$

```
for t ← 1, 2, 3..., T do
    if B_t ≥ 1 then
        Play x_t and observe f_t, c_t and update available
        resources: B_{t+1} ← B_t − c_t(x).
        if t=1 then
            d_t = ∇f_t(x_t)
        else
            d_t = ∇f_t(x_t) + (1 − η_t)(d_{t−1} − ∇f_t(x_{t−1}))
        end if
        v_t = arg max_{x∈P} ⟨x, d_t + ∇r(x_t)⟩
        Set x_{t+1} = x_t + (1/T) v_t
    else
        x_t = 0
    end if
end for
```

Algorithm 2 is a variant of the Frank-Wolfe algorithm. The decision variables in the algorithm are updated using $\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{1}{T}\mathbf{v}_t$, where $\mathbf{v}_t$ is a positive vector. As a result, the sequence $\{\mathbf{x}_t\}_{t=1}^{t=T}$ satisfies the partial order condition of Proposition 1.

Suppose f is known in advance, we would employ the Frank-Wolfe algorithm for offline DR-submodular maximization. Specifically, starting from $\mathbf{x}_0 = \mathbf{0}$, we would perform T Frank-Wolfe updates. At each iteration t, we choose $\mathbf{v}_t$ according to $\mathbf{v}_t = \arg\max_{\mathbf{x}\in\mathcal{P}} \langle \mathbf{x}, \nabla f(\mathbf{x}_t)\rangle$, and then perform the update $\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{1}{T}\mathbf{v}_t$. Howeve, if f is not known beforehand, we estimate $\nabla f(\mathbf{x}_t)$ by employing the recursive estimator $\mathbf{d}_t = \nabla f_t(\mathbf{x}_t) + (1-\eta_t)(\mathbf{d}_{t-1} - \nabla f_t(\mathbf{x}_{t-1}))$, which is inspired by variance-reduction techniques.

Because the functions are i.i.d. sampled as $f_t \sim \mathcal{D}$, employing variance reduction together with one Frank-Wolfe step is sufficient to achieve a sublinear regret bound.

Theorem 2 *If Assumption 2 holds, let $\mathbf{x}_t : 1 \leq t \leq T$ be the choices of Algorithm 2 and let $\eta_t = \frac{1}{t+1}$, then we have the following stochastic bound in expectation:*

$$\mathbb{E}((\frac{\rho}{e^\rho}, \frac{\rho}{e^\rho}) - \mathcal{SR}) = O(\sqrt{T}).$$

It is noteworthy that although we presume our reward function to be monotone DR-submodular, both Algorithm 2 and Theorem 2 can be extended to scenarios where the reward function is a monotone concave function. This is because Algorithm 2 primarily utilizes the characteristic that DR-submodular functions are concave in positive directions, and this trait holds true for concave functions as well.

6 NUMERICAL EXPERIMENTS

In order to verify our theoretical findings, we assess the efficacy of our algorithms through two numerical experiments.

(1) *Online joke recommendation with max-min fairness regularizer* We choose $n = 100$ jokes. We consider the lengths of horizon $T \in \{10^2, 10^3, 2 \cdot 10^3, \ldots, 10^4\}$. In each round, we are required to choose 20 jokes for recommendation, so we set $\mathcal{P} = \{\mathbf{x} \in [0,1]^n : 1^T\mathbf{x} \leq 20\}$. The reward functions we consider here is $f_t(\mathbf{x}) = r_t^T\mathbf{x} + \sum_{i,j:i<j} \theta_{t(ij)} x_i x_j \forall t \in [T]$. The original ratings in the *Jester* dataset[1] are within the range of $[-10, 10]$. To normalize the ratings, we rescale them to fit within the range of $[1, 10]$. We randomly select $\theta_{ij}^{(t)}$ from the range of $[-0.05, 0]$. In this case $f_t(x)$ in each round is a monotone DR-submodular function. Also, $p_{t(i)}$ is chosen randomly from range [0,0.05]. The regularizer is $r(\mathbf{x}) = \lambda \min_i \mathbf{x}_i$.

First, we set $\rho = 1$, which means there is no constraints. We use the algorithm in [Mitra et al., 2021] to solve the offline problem as our $(1 - \frac{1}{e}, 1)$-regret benchmark. We compare our Algorithm 1 with Online Gradient Ascent[Chen et al., 2018b] when $\lambda = 1$. Figure 1(a) shows that each algorithm get a sublinear regret. But our algorithm performs much better than Online Gradient Ascent. We also tested the performance of our algorithm at different regularization levels $\lambda \in \{1, 3, 5, 7, 9\}$. Figure 1(b) suggests that regret grows at rate $O(\sqrt{T})$ for all regularization levels. The reason why we choose this regularization levels is that the gradient of $f_t(x)$ in each round is in $[0, 10]^n$, and the size of the gradient of $r(\mathbf{x}) = \lambda \min_i \mathbf{x}_i$ is λ. Using these regularization levels can significantly affect the behavior of the algorithm. Figure 1(c) presents the trade-off between reward and fairness. Although employing a higher value of λ will result in a reduction of the reward. However, we can enhance the fairness level from 1 to 9 by incurring a mere 1.4% reduction in reward. The actual cumulative reward of Algorithm 1 at different ρ levels is depicted in Figure 1(d). According to Theorem 1, our approximation ratio increases linearly with ρ. It should be noted that Theorem 1 is based on a worst-case scenario where the resources consumed in each round are always the largest ($c_t(\mathbf{x}_t) = 1$). However, in practice and in the present experiment, the resources consumed may not always be the largest. Therefore, the reward of Algorithm 1 will not decrease significantly as long as the resources are sufficient (i.e., when ρ is still large). It is only when resources are scarce that the performance of the algorithm will be affected.

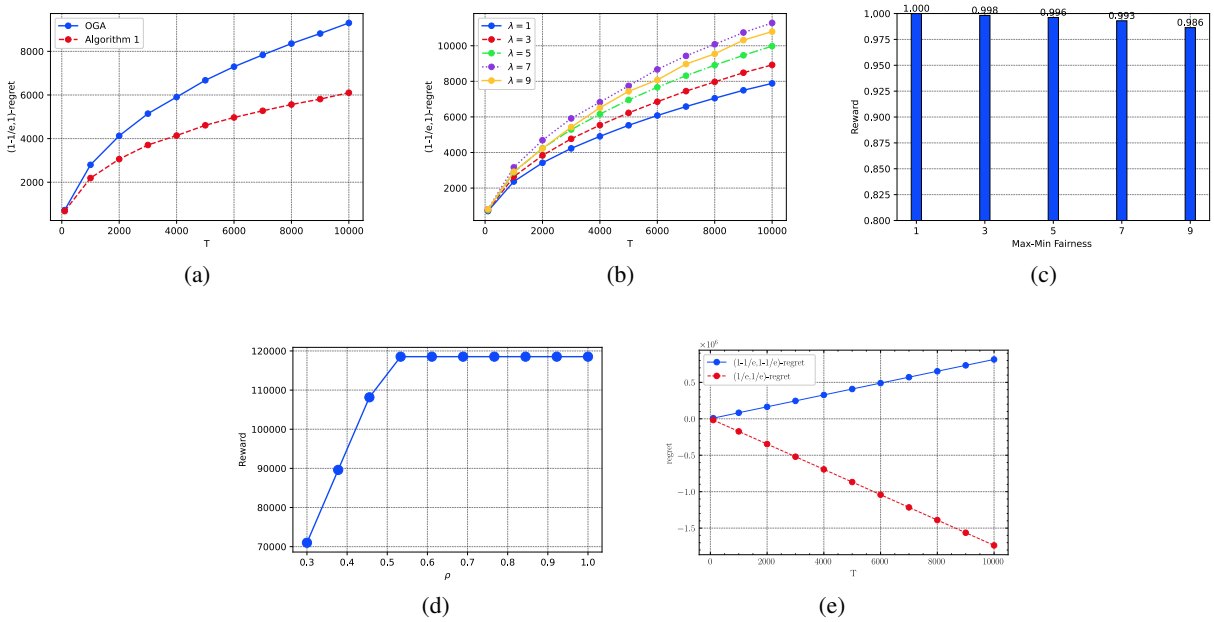
(2) *Online Advertising with Influence Maximization* Here we consider the problem of online advertising investment. Suppose we have $n = 100$ channels to invest, and the investment budget for each round is $B = 20$. This process is iterated for a total of 10000 rounds, with each round involving the selection of an action $\mathbf{x}_t : \{\mathbf{x} \in [0,1]^n : 1^T\mathbf{x} \leq 20\}$, followed by the acquisition of a reward $f_t(\mathbf{x}_t) = \mathbf{r}_t^T\mathbf{x}_t$. We generate $r_{t(i)}$ uniformly from the interval $[0, 10]$. Furthermore, we make the assumption that users who are presented with recommended advertisements can be categorized into 20 distinct groups. For each group and channel, we assign a weight value p_{sg}, which denotes the likelihood of group g being activated by a single unit of investment in channel s. p_{sg} is chosen randomly from $[0, 1]$. We consider $r(\mathbf{x}) = \lambda \sum_{g \in G} \left(1 - \prod_{(s,g) \in W} (1 - p_{sg})^{x_s}\right)$ as our regularizer to maximize the investment influence.

We consider the case there is no constraints ($\rho = 1$).We adopt the Frank-Wolfe algorithm described in [Bian et al., 2017b] as our benchmark to solve the offline problem. While the initial design of the method was intended for DR-submodular functions, it is capable of accommodating the sum of both DR-submodular and concave functions. This is due to the property of concave functions being concave in the positive direction, allowing the method to handle the combination of the two types of functions. The algorithm provides a $(1 - \frac{1}{e}, 1 - \frac{1}{e})$ approximation ratio. Nonetheless, to attain our target $(\frac{1}{e}, \frac{1}{e})$ approximation ratio, we adjusted the reward achieved by the Frank-Wolfe algorithm by scaling it based on the approximation ratio. By doing so, we established a new benchmark that aligns with our desired approximation ratio. In Figure 1(e), the regret under both approximation ratios is depicted. The graph reveals that the curve corresponding to the $(\frac{1}{e}, \frac{1}{e})$-regret is consistently below zero. As a result, we can conclude that our approximation ratio is at least $(\frac{1}{e}, \frac{1}{e})$ in this particular experiment, and it falls within the range of $(\frac{1}{e}, \frac{1}{e})$ and $(1 - \frac{1}{e}, 1 - \frac{1}{e})$. Furthermore, since the regret is below zero, it is sublinear ($0 \leq \sqrt{T}$). This provides compelling evidence for the efficacy of our proposed solutions.

7 CONCLUSION

This paper addresses the regularized online DR-submodular optimization problem, where the regularizer is either a concave or a DR-submodular function. We provide application scenarios for each type of regularizer, and present efficient algorithms that come with theoretical performance guarantees. Finally, we have confirmed the validity of our theoretical findings through numerical experiments.

[1] http://eigentaste.berkeley.edu/dataset/



(a) (b) (c)

(d) (e)

Figure 1: (a) contrasts the performance of Algorithm 1 and OGA. (b) illustrates the regret of Algorithm 1 for different levels of regularization. (c) displays the trade-off between reward and fairness. (d) showcases the actual cumulative reward of Algorithm 1 at various ρ levels. (e) demonstrates the regret of Algorithm 2 with two benchmark methods.

References

Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006, 2014a.

Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1405–1424. SIAM, 2014b.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Regularized online allocation problems: Fairness and beyond. In *International Conference on Machine Learning*, pages 630–639. PMLR, 2021.

Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 31–40, 2006.

An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous dr-submodular maximization: Structure and algorithms. *Advances in Neural Information Processing Systems*, 30, 2017a.

Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *Artificial Intelligence and Statistics*, pages 111–120. PMLR, 2017b.

Yatao Bian, Joachim M Buhmann, and Andreas Krause. Continuous submodular function maximization. *arXiv preprint arXiv:2006.13474*, 2020.

Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009.

Niv Buchbinder, Kamal Jain, and Joseph Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.

Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

Matteo Castiglioni, Andrea Celli, and Christian Kroer. Online learning with knapsacks: the best of both worlds. In *International Conference on Machine Learning*, pages 2767–2783. PMLR, 2022.

Lin Chen, Christopher Harshaw, Hamed Hassani, and Amin Karbasi. Projection-free online optimization with stochas-

tic gradient: From convexity to submodularity. In *International Conference on Machine Learning*, pages 814–823. PMLR, 2018a.

Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 1896–1905. PMLR, 2018b.

Sham M Kakade, Adam Tauman Kalai, and Katrina Ligett. Playing games with approximation algorithms. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 546–555, 2007.

Ehsan Kazemi, Shervin Minaee, Moran Feldman, and Amin Karbasi. Regularized submodular maximization at scale. In *International Conference on Machine Learning*, pages 5356–5366. PMLR, 2021.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.

Siddharth Mitra, Moran Feldman, and Amin Karbasi. Submodular+ concave. *Advances in Neural Information Processing Systems*, 34:11577–11591, 2021.

Joshua Robinson, Suvrit Sra, and Stefanie Jegelka. Flexible modeling of diversity with strongly log-concave distributions. *Advances in Neural Information Processing Systems*, 32, 2019.

Omid Sadeghi, Prasanna Raut, and Maryam Fazel. Improved regret bounds for online submodular maximization. *arXiv preprint arXiv:2106.07836*, 2021.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. *Advances in Neural Information Processing Systems*, 21, 2008.

Qixin Zhang, Zengde Deng, Zaiyi Chen, Haoyuan Hu, and Yu Yang. Stochastic continuous submodular maximization: Boosting via non-oblivious function. In *International Conference on Machine Learning*, pages 26116–26134. PMLR, 2022.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.