Resolving the Approximability of Offline and Online Non-monotone DR-Submodular Maximization over General Convex Sets

Loay Mualem
University of Haifa

Moran Feldman
University of Haifa

Abstract

In recent years, maximization of DR-submodular continuous functions became an important research field, with many real-worlds applications in the domains of machine learning, communication systems, operation research and economics. Most of the works in this field study maximization subject to down-closed convex set constraints due to an inapproximability result by Vondrák (2013). However, Dürr et al. (2021) showed that one can bypass this inapproximability by proving approximation ratios that are functions of m, the minimum ℓ_∞-norm of any feasible vector. Given this observation, it is possible to get results for maximizing a DR-submodular function subject to *general* convex set constraints, which has led to multiple works on this problem. The most recent of which is a polynomial time $\frac{1}{4}(1-m)$-approximation offline algorithm due to Du (2022). However, only a sub-exponential time $\frac{1}{3\sqrt{3}}(1-m)$-approximation algorithm is known for the corresponding online problem. In this work, we present a polynomial time online algorithm matching the $\frac{1}{4}(1-m)$-approximation of the state-of-the-art offline algorithm. We also present an inapproximability result showing that our online algorithm and Du's offline algorithm are both optimal in a strong sense. Finally, we study the empirical performance of our algorithm and the algorithm of Du (which was only theoretically studied previously), and show that they consistently outperform previously suggested algorithms on revenue maximization, location summarization and quadratic programming applications.

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

1 INTRODUCTION

Optimization of continuous DR-submodular functions has gained prominence in recent times. Such optimization is an important tractable subclass of non-convex optimization, and captures problems at the forefront of machine learning and statistics with many real-world applications (see, e.g., (Bian et al., 2019; Hassani et al., 2017a; Mitra et al., 2021; Soma and Yoshida, 2017)). The majority of the existing works on DR-submodular optimization (and submodular optimization in general) have been focused either on monotone objective functions, or optimization subject to a down-closed convex set constraint.[1] However, many real-world problems are naturally captured as optimization of a non-monotone DR-submodular function over a constraint convex set that is not down-closed. For example, consider a streaming service that would like to produce a summary of recommended movies for a user. Often the design of the user interface places strong bounds on the size of the summary displayed to the user, leading to a non-down-closed constraint. Furthermore, the quality of the summary is often captured by a non-monotone objective since putting very similar films in the summary is detrimental to both its value and professional look.

Motivated by the above-mentioned situation, a few recent works started to consider DR-submodular maximization subject to a general (not necessarily down-closed) convex set constraint $\mathcal{K}$. In general, no constant approximation ratio can be guaranteed for this problem in sub-exponential time due to an hardness result by Vondrák (2013). However, Dürr et al. (2021) showed that this inapproximability result can be bypassed when the convex set constraint $\mathcal{K}$ includes points whose ℓ_∞-norm is less than the maximal value of 1. Specifically, Dürr et al. (2021) presented a sub-exponential time offline algorithm guaranteeing $\frac{1}{3\sqrt{3}}(1-m)$-approximation for this problem, where m is the minimal ℓ_∞-norm of any vector in $\mathcal{K}$. Later, Thắng and Srivastav (2021) showed how to obtain a similar result in an online (regret minimization) setting, and

[1] A set $\mathcal{K} \subseteq [0,1]^n$ is down-closed if, for every two vectors $\mathbf{x}, \mathbf{y} \in [0,1]^n$, $\mathbf{x} \in \mathcal{K}$ whenever $\mathbf{y} \in \mathcal{K}$ and $\mathbf{y}$ coordinate-wise dominates $\mathbf{x}$.

an improved sub-exponential offline algorithm obtaining $\frac{1}{4}(1-m)$-approximation was suggested by Du et al. (2022). Very recently, Du (2022) provided the first polynomial time algorithm for this setting, obtaining the same offline $\frac{1}{4}(1-m)$-approximation as Du et al. (2022). Nevertheless, and despite all the progress described above, there are still important open questions left regarding this setting.

- What is the best approximation ratio that can be obtained by a polynomial time offline algorithm? In particular, can such an algorithm guarantee a better than $\frac{1}{4}(1-m)$-approximation, and if not, how much slower must be an algorithm that improves over this approximation ratio?
- Is there a polynomial time *online* algorithm guaranteeing any constant approximation ratio? Can such an algorithm match the optimal approximation ratio obtainable by an offline algorithm?

In this work we answer all the above questions, essentially settling the problem of maximizing DR-submodular functions over general convex sets in both the offline and online settings. We also study the empirical performance of the theoretically optimal offline and online algorithms, showing that both algorithms consistently outperform previously suggested algorithms. Below we describe our results in more detail.

Online setting. As mentioned above, the state-of-the-art online (regret minimization) algorithm of Thắng and Srivastav (2021) achieves $\frac{1}{3\sqrt{3}}(1-m)$-approximation, which it does with sub-exponential running time and roughly $O(\sqrt{T})$-regret, where T is the number of time steps.[2] In this paper, we describe a new online algorithm improving both the approximation ratio and the time complexity. Specifically, our algorithm achieves $\frac{1}{4}(1-m)$-approximation in polynomial time and roughly $O(\sqrt{T})$-regret. The approximation guarantee of our algorithm matches an inapproximability that we prove for the offline setting (see below), and is thus, optimal. We also study the empirical performance of our algorithm, and show that it outperforms the algorithm of Thắng and Srivastav (2021) on two applications of revenue maximization and location summarization.

Offline setting. Recall that the state-of-the-art offline algorithm is a recent polynomial time $\frac{1}{4}(1-m)$-approximation algorithm due to Du (2022). Our first contribution to the offline setting is an inapproximability result showing that this algorithm is optimal in a very strong sense. Specifically, we show that no sub-exponential time algorithm can significantly improve over this approximation ratio, even when m is fixed to any particular value in $[0,1]$. Furthermore, since Du (2022) analyzed only the theoretical performance of his algorithm, it is interesting to study the empirical performance of this algorithm, which we do by considering revenue maximization and quadratic programming applications.

Coding the algorithm of Du (2022) for the empirical study is somewhat non-trivial because Du (2022) presented his algorithm as part of a general mathematical framework for designing algorithms for various submodular optimization problems. Therefore, our empirical study is based on an explicit version of this algorithm that we give in this paper, which is not fully identical to the algorithm of Du (2022). Beside being explicit, our version of the algorithm also has the advantage of being more tuned towards practical performance. For completeness, we include a full analysis of our version of the algorithm of Du (2022). This full analysis is also used as a warm-up towards the analysis of our own online algorithm.

1.1 Related work

Next, we provide a brief summary of the most relevant results on DR-submodular maximization. Recently, this field has become the work-horse of numerous applications in the fields of statistics and machine learning, which has lead to a dramatic increase in the number of studies related to it.

Offline DR-submodular optimization. The problem of maximizing monotone DR-functions subject to a down-closed convex set was considered by Bian et al. (2017a), who showed a variant of the Frank-Wolfe algorithm (based on the greedy method proposed by Calinescu et al. (2011) for set functions) that guarantees $(1-1/e)$-approximation for this problem, which is optimal (Nemhauser and Wolsey, 1978). Later, Hassani et al. (2017a) showed that the algorithm of Bian et al. (2017a) is not robust in stochastic settings (i.e., when only an unbiased estimator of gradients is available), and proved that gradient methods are robust in such setting while still achieving $1/2$-approximation. When the objective DR-submodular function is not necessarily monotone, the problem becomes harder to approximate. Bian et al. (2019) and Niazadeh et al. (2020) independently provided two algorithms with the same approximation guarantee of $1/2$ for maximizing non-monotone DR-submodular functions over a hypercube, which is optimal (Feige et al., 2011) (the algorithm of Niazadeh et al. (2020) applies also to non-DR submodular functions). For general down-closed convex sets, Bian et al. (2018) provided a $1/e$-approximation algorithm based on the greedy method of Feldman et al. (2011) for set functions. Using the concept of monotonicity ratio, Mualem and Feldman (2022) were able to smoothly interpolate between the

[2] By changing parameter values, it is possible to reduce the time complexity of the algorithm of Thắng and Srivastav (2021) to be polynomial. However, this comes at the cost of a regret that is nearly-linear in T and an error term in the approximation ratio that diminishes very slowly (linearly in $\log T$).

last result and the $(1 - 1/e)$-approximation obtainable for monotone objectives.

Online DR-submodular optimization. Online optimization of monotone DR-submodular functions over general convex sets (for monotone objective functions, there is no difference between optimization subject to down-closed or general convex sets) was first considered by Chen et al. (2018), who provided two algorithms. One guaranteeing $(1 - 1/e)$-approximation using roughly $O(\sqrt{T})$-regret, and another algorithm which is robust to stochastic settings but guarantees only $1/2$-approximation up to the same regret. Later, Chen et al. (2019) presented an algorithm that combines $(1 - 1/e)$-approximation with roughly $O(\sqrt{T})$-regret and robustness, and Zhang et al. (2019) showed how one can reduce the number of gradient calculations per time step to one, at the cost of increasing the regret to roughly $O(T^{4/5})$. Such a reduction is important for bandit versions of the same problem. Online optimization of DR-submodular functions that are not necessarily monotone was studied by Thắng and Srivastav (2021), who provided three algorithms for it. One of these algorithms applies to general convex set constraints, and was already discussed above. Another algorithm applies to maximization over the entire hypercube, and achieves $1/2$-approximation with roughly $O(\sqrt{T})$-regret; and the last algorithm applies to online maximization of non-monotone DR-submodular functions over down-closed convex sets, and achieves $1/e$-approximation with roughly $O(T^{3/4})$-regret.

1.2 Paper organization

In Section 2, we provide some definitions and important properties of DR-submodular functions. Section 3 describes our explicit version of the offline algorithm of Du (2022), which also serves as warm up for our novel online algorithm described in Section 4. Our inapproximability result, showing that the above offline and online algorithms are both optimal, is proved in Section 5. Finally, in Section 6, we study the empirical performance and robustness of our online algorithm and our version of the algorithm of Du (2022) by comparing them with previously suggested algorithms on multiple machine learning applications.

2 PRELIMINARIES

DR-submodularity (first defined by Bian et al. (2017b)) is an extension of the submodularity notion from set functions to continuous functions. Formally speaking, given a domain $\mathcal{X} = \prod_{i=1}^{n} \mathcal{X}_i$, where $\mathcal{X}_i$ is a closed range in $\mathbb{R}$ for every $i \in [n]$, a function $F\colon \mathcal{X} \to \mathbb{R}$ is *DR-submodular* if for every two vectors $\mathbf{a}, \mathbf{b} \in \mathcal{X}$, positive value k and coordinate $i \in [n]$, the inequality $F(\mathbf{a} + k\mathbf{e}_i) - F(\mathbf{a}) \geq F(\mathbf{b}+k\mathbf{e}_i) - F(\mathbf{b})$ holds whenever $\mathbf{a} \leq \mathbf{b}$ and $\mathbf{b}+k\mathbf{e}_i \in \mathcal{X}$ (here and throughout the paper, $\mathbf{e}_i$ denotes the standard i-th basis vector, and comparison between two vectors should be understood to hold coordinate-wise). Note that if function F is continuously differentiable, then the above definition of DR-submodulrity is equivalent to

$$\nabla F(\mathbf{x}) \leq \nabla F(\mathbf{y}) \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X}, \mathbf{x} \geq \mathbf{y} \ .$$

Furthermore, when F is twice differentiable, it is DR-submodular if and only if its Hessian is non-positive at every vector $\mathbf{x} \in \mathcal{X}$.

In this work, we study the problem of maximizing a non-negative DR-submodular function $F\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ subject to a general convex body $\mathcal{K} \subseteq \mathcal{X}$ (usually polytope) constraint. For simplicity, we assume that $\mathcal{X} = [0, 1]^n$. Note that this assumption is without loss of generality since there is a natural mapping from $\mathcal{X}$ to $[0, 1]^n$. Additionally, as is standard in the field, we assume that F is β-smooth for some parameter $\beta > 0$. Recall that F is β-smooth if it is continuously differentiable, and for every two vectors $\mathbf{x}, \mathbf{y} \in [0, 1]^n$, the function F obeys $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\|_2 \leq \beta \|\mathbf{x} - \mathbf{y}\|_2$.

In the online (regret minimization) version of the above problem, there are T time steps. In every time step $t \in [T]$, the adversary selects a non-negative β-smooth DR-submodular function F_t, and then the algorithm should select a vector $\mathbf{y}^{(t)} \in \mathcal{K}$ without knowing F_t (the function F_t is revealed to the algorithm only after $\mathbf{y}^{(t)}$ is selected). The objective of the algorithm is to maximize $\sum_{i=1}^{T} F_t(\mathbf{y}^{(t)})$, and its success in doing so is measured compared to the best fixed vector $\mathbf{x} \in \mathcal{K}$. More formally, we say that the algorithm achieves an approximation ratio of $c \geq 0$ with regret $\mathcal{R}(T)$ if

$$\mathbb{E}\left[\sum_{t=1}^{T} F_t(\mathbf{y}^{(t)})\right] \geq c \cdot \max_{\mathbf{x} \in \mathcal{K}} \mathbb{E}\left[\sum_{t=1}^{T} F_t(\mathbf{x})\right] - \mathcal{R}(T) \ .$$

The nature of the access that the algorithm has to F_t varies between different versions of the above problem. Some previous works assume access to the exact gradient of F. However, our algorithm applies also to a stochastic version of the problem in which only access to an unbiased estimator of this gradient is available.

We conclude this section by introducing some additional notation and two known lemmata that are useful in our proofs. Given two vectors $\mathbf{x}, \mathbf{y} \in [0, 1]^n$, we denote by $\mathbf{x} \vee \mathbf{y}$ and $\mathbf{x} \wedge \mathbf{y}$ their coordinate-wise maximum and minimum, respectively. Using this notation, we can now state the first known lemma, which can be traced back to Hassani et al. (2017a) (see Inequality 7.5 in the arXiv version (Hassani et al., 2017b) of Hassani et al. (2017a)), and is also explicitly stated and proved in (Dürr et al., 2021).

Lemma 2.1 (Lemma 1 of Dürr et al. (2021))**.** *For every two vectors $\mathbf{x}, \mathbf{y} \in [0, 1]^n$ and any continuously differentiable DR-submodular function $F\colon [0, 1]^n \to \mathbb{R}$, $\langle \nabla F(x), y - x \rangle \geq F(\mathbf{x} \vee \mathbf{y}) + F(\mathbf{x} \wedge \mathbf{y}) - 2F(\mathbf{x})$.*

The following lemma originates from a lemma proved by Feldman et al. (2011) for set functions. Extensions of this lemma to continuous domains have appeared in (Bian et al., 2017a; Chekuri et al., 2015), but for completeness, we prove our exact version of the lemma in Appendix A.

Lemma 2.2. *For every two vectors $\mathbf{x}, \mathbf{y} \in [0,1]^n$ and any continuously differentiable non-negative DR-submodular function $F\colon [0,1]^n \to \mathbb{R}_{\geq 0}$, $F(\mathbf{x} \vee \mathbf{y}) \geq (1 - \|\mathbf{x}\|_\infty) F(\mathbf{y})$.*

3 OFFILINE MAXIMIZATION

In this section, we present and analyze an explicit variant of the offline algorithm of Du (2022) for maximizing a non-negative DR-submodular function F over a general convex set $\mathcal{K}$. Since the algorithm of Du (2022) is related to Frank-Wolfe, we name our variant `Non-mon. Frank-Wolfe`, and its pseudocode appears as Algorithm 1. Algorithm 1 gets a non-negative integer parameter T and a quality control parameter $\varepsilon \in (0,1)$.

Algorithm 1: `Non-mon. Frank-Wolfe` (T, ε)

Let $\mathbf{y}^{(0)} \leftarrow \arg\min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty$.
for $i = 1$ **to** T **do**
  Let $\mathbf{s}^{(i)} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{K}} \langle \nabla F(\mathbf{y}^{(i-1)}), \mathbf{x} \rangle$
  Let $\mathbf{y}^{(i)} \leftarrow (1-\varepsilon) \cdot \mathbf{y}^{(i-1)} + \varepsilon \cdot \mathbf{s}^{(i)}$
return the vector maximizing F among $\{\mathbf{y}^{(0)}, \ldots, \mathbf{y}^{(T)}\}$.

For completeness, and as a warmup for Section 4, we present a full analysis of Algorithm 1, independent of the analysis presented by Du (2022). The conclusions of our analysis are summarized by the following theorem. We note that, for the purpose of this theorem, it would have sufficed for Algorithm 1 to return $\mathbf{y}^{(T)}$ rather than the best solution among $\mathbf{y}^{(0)}, \ldots, \mathbf{y}^{(T)}$. However, returning the best of these solutions results in a better empirical performance at almost no additional cost.

Theorem 3.1. *Let $\mathcal{K} \subseteq [0,1]^n$ be a general convex set, and let $F\colon [0,1]^n \to \mathbb{R}_{\geq 0}$ be a non-negative β-smooth DR-submodular function. Then,* `Non-mon. Frank-Wolfe` *(Algorithm 1) outputs a solution $\mathbf{w} \in \mathcal{K}$ obeying*

$$F(\mathbf{w}) \geq (1-2\varepsilon)^{T-1}[(1+\varepsilon)^T - 1](1 - \min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2 \beta D^2 T \ ,$$

where D is the diameter of $\mathcal{K}$ and $\mathbf{o} \in \arg\max_{\mathbf{x} \in \mathcal{K}} F(\mathbf{x})$. In particular, when T is set to be $\lfloor \ln 2/\varepsilon \rfloor$,

$$F(\mathbf{w}) \geq (1/4 - 3\varepsilon)(1 - \min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon\beta D^2 \ .$$

We begin the proof of Theorem 3.1 with the following observation, which bounds the rate in which the infinity norm of the solution maintained by Algorithm 1 can be increase. The proof of this observation is done by induction on the number of iterations, and can be found in Appendix B (like all the other proofs of this section).

Observation 3.2. *For every integer $0 \leq i \leq T$, $1 - \|\mathbf{y}^{(i)}\|_\infty \geq (1-\varepsilon)^i \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty)$.*

By combining the last observation and Lemma 2.2, we can prove the following lemma about the rate in which the value of $F(y^{(i)})$ increases as a function of i. The proof gives a bound on the rate of increase in terms of $\langle \mathbf{s}^{(i)}, \nabla F(\mathbf{y}^{(i-1)}) \rangle$, and then lower bounds this inner product by observing that $\mathbf{o}$ is one possible candidate to be $\mathbf{s}^{(i)}$.

Lemma 3.3. *For every integer $1 \leq i \leq T$, $F(\mathbf{y}^{(i)}) \geq (1-2\varepsilon) \cdot F(\mathbf{y}^{(i-1)}) + \varepsilon(1-\varepsilon)^{i-1} \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2 \beta D^2$.*

Theorem 3.1 is proved by using Lemma 3.3 repeatedly.

4 ONLINE MAXIMIZATION

In this section, we consider the problem of maximizing a non-negative DR-submodular function F over a general convex set $\mathcal{K}$ in the online setting. The only currently known algorithm for this problem is an algorithm due to Thắng and Srivastav (2021) which guarantees $\frac{1 - \min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty}{3\sqrt{3}}$-approximation. One drawback of this algorithm is that its regret is roughly T over the logarithm of the running time, and therefore, to make this regret less than nearly-linear in T one has to allow for a super-polynomial time complexity (furthermore, a sub-exponential time complexity is necessary to get a regret of T^c for any constant $c \in (0,1)$). Our algorithm, given as Algorithm 2, combines ideas from our offline algorithm and the Meta-Frank-Wolfe algorithm suggested in (Chen et al., 2018), and guarantees both $\frac{1}{4}(1 - \min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty)$-approximation and roughly $O(\sqrt{T})$-regret in polynomial time.

Like the original Meta-Frank-Wolfe algorithm of Chen et al. (2018), our algorithm uses in a black-box manner multiple instances $\mathcal{E}$ of an online algorithm for linear optimization. More formally, we assume that every instance $\mathcal{E}$ has the following behavior and guarantee. There are T time steps. In every time step $t \in [T]$, $\mathcal{E}$ selects a vector $\mathbf{u}^{(t)} \in \mathcal{K}$, and then an adversary reveals to $\mathcal{E}$ a vector $\mathbf{d}^{(t)}$ that was chosen independently of $\mathbf{u}^{(t)}$. The algorithm $\mathcal{E}$ guarantees that

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbf{u}^{(t)}, \mathbf{d}^{(t)} \rangle\right] \geq \max_{\mathbf{x} \in \mathcal{K}} \mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbf{x}, \mathbf{d}^{(t)} \rangle\right] - \mathcal{R}(T)$$

for some regret function $\mathcal{R}(T)$ that depends on the particular linear optimization algorithm chosen as the black-box (and may depend on the convex body $\mathcal{K}$ and the bounds

available on the adversarially chosen vectors $\mathbf{d}^{(t)}$). One possible choice for an online linear optimization algorithm is Regularized-Follow-the-Leader due to Abernethy et al. (2008) that has $\mathcal{R}(T) \leq DG\sqrt{2T}$, where D is the diameter of $\mathcal{K}$ and $G = \max_{1 \leq t \leq T} \|\mathbf{d}^{(t)}\|_2$.

Algorithm 2 runs in each time step a procedure similar to our version of the offline algorithm (`Non-mon. Frank-Wolfe`). However, instead of calculating a point $\mathbf{s}$ that is good with respect to the gradient at the current solution, Algorithm 2 asks an instance of an online linear optimization algorithm to provide such a point. At the end of the time step, the online linear optimization algorithm gets an estimate of the gradient as the adversarial vector, and therefore, on average, the points it produces are a good approximation of the optimal point in retrospect. Algorithm 2 gets three parameters. The parameters L and ε correspond to the parameters T and ε of `Non-mon. Frank-Wolfe` (Algorithm 1),[3] respectively, and the parameter T is the number of time steps.

Algorithm 2: `Non-mon. Meta-Frank-Wolfe` (L, ε, T)

for $i = 1$ **to** L **do** Initialize an instance $\mathcal{E}_i$ of some online algorithm for linear optimization.
for $t = 1$ **to** T **do**
  Let $\mathbf{y}^{(0,t)} \leftarrow \arg\min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty$.
  for $i = 1$ **to** L **do**
    Let $\mathbf{s}^{(i,t)} \in \mathcal{K} \leftarrow$ be the vector picked by $\mathcal{E}_\ell$ in time step t.
    Let $\mathbf{y}^{(i,t)} \leftarrow (1-\varepsilon) \cdot \mathbf{y}^{(i-1,t)} + \varepsilon \cdot \mathbf{s}^{(i,t)}$.
  Play $\mathbf{y}^{(t)} = \mathbf{y}^{(L,t)}$.
  for $i = 1$ **to** L **do**
    Observe an unbiased estimator $\mathbf{g}^{(i,t)}$ of $\nabla F_t(\mathbf{y}^{(i-1,t)})$.
    Pass $\mathbf{g}^{(i,t)}$ as the adverserially chosen vector $\mathbf{d}^{(t)}$ for $\mathcal{E}_i$.

The main result that we prove regarding the online setting is given by the next theorem.

Theorem 4.1. *Let $\mathcal{K}$ be a general convex set with diameter D. Assume that for every $1 \leq t \leq T$, $F_t \colon [0,1]^n \to \mathbb{R}_{\geq 0}$ is a β-smooth DR-submodular function, then*

$$\sum_{t=1}^{T} \mathbb{E}[F_t(\mathbf{y}^{(t)})] \geq (1-2\varepsilon)^{T-1}[(1+\varepsilon)^T - 1](1 - \min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty) \cdot \mathbb{E}\left[\sum_{t=1}^{T} F(\mathbf{o})\right] - \varepsilon L \cdot \mathcal{R}(T) - 0.5\varepsilon^2 \beta D^2 T L \ ,$$

where D is the diameter of $\mathcal{K}$, $\mathbf{o}$ is a vector in $\mathcal{K}$ maximizing $\mathbb{E}[\sum_{t=1}^{T} F_t(\mathbf{o})]$, and $\mathcal{R}(T)$ is the regret of the online linear optimization algorithm over the domain $\mathcal{K}$ when the adversarial vectors $\mathbf{d}^{(t)}$ are the estimators $\mathbf{g}^{(i,t)}$ calculated by Algorithm 2. In particular, when L is set to be $\lfloor \ln 2/\varepsilon \rfloor$, ε is set to be $1/\sqrt{T}$ and $\mathcal{E}_i$ is chosen as an instance of Regularized-Follow-the-Leader,

$$\sum_{t=1}^{T} \mathbb{E}[F_t(\mathbf{y}^{(t)})] \geq (1/4 - 3\varepsilon)(1 - \min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|_\infty) \cdot \mathbb{E}\left[\sum_{t=1}^{T} F_t(\mathbf{o})\right] - (G + \beta D) D \sqrt{T} \ ,$$

where $G = \max_{1 \leq i \leq L, 1 \leq t \leq T} \|\mathbf{g}^{(i,t)}\|_2$.

Remark: In the last theorem we have set ε to $1/\sqrt{T}$, which requires pre-knowledge of T. This can be avoided by using a dynamic value for ε that changes as a function of the number of time slots that have already passed.

We begin the proof of Theorem 4.1 by observing that a repetition of the first half of the proof of Lemma 3.3 leads to the following lemma.

Lemma 4.2. *For every two integers $1 \leq t \leq T$ and $1 \leq i \leq L$, $F_t(\mathbf{y}^{(i,t)}) \geq F_t(\mathbf{y}^{(i-1,t)}) + \varepsilon \cdot \langle \mathbf{s}^{(i,t)} - \mathbf{y}^{(i-1,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) \rangle - 0.5\varepsilon^2 \beta D^2$.*

Using the guarantee of $\mathcal{E}_i$, it is possible to get the following lemma from the previous one.

Lemma 4.3. *For every integer number $1 \leq i \leq L$, $\mathbb{E}[\sum_{t=1}^{T} F_t(\mathbf{y}^{(i,t)})] \geq \mathbb{E}[\sum_{t=1}^{T} F_t(\mathbf{y}^{(i-1,t)}) + \varepsilon \cdot \sum_{t=1}^{T} \langle \mathbf{o} - \mathbf{y}^{(i-1,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) \rangle] - \varepsilon \cdot \mathcal{R}(T) - 0.5\varepsilon^2 \beta D^2 T$.*

Proof. Summing up Lemma 4.2 over all t values, we get

$$\begin{aligned}
&\sum_{t=1}^{T} F_t(\mathbf{y}^{(i,t)}) \geq \sum_{t=1}^{T} F_t(\mathbf{y}^{(i-1,t)}) - 0.5\varepsilon^2 \beta D^2 T + \varepsilon \cdot \sum_{t=1}^{T} \langle \mathbf{s}^{(i,t)} - \mathbf{y}^{(i-1,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) \rangle \\
&= \sum_{t=1}^{T} F_t(\mathbf{y}^{(i-1,t)}) - 0.5\varepsilon^2 \beta D^2 T + \varepsilon \cdot \left[\sum_{t=1}^{T} \langle \mathbf{s}^{(i,t)}, \mathbf{g}^{(i,t)} \rangle + \sum_{t=1}^{T} \langle \mathbf{s}^{(i,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) - \mathbf{g}^{(i,t)} \rangle - \sum_{t=1}^{T} \langle \mathbf{y}^{(i-1,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) \rangle\right] .
\end{aligned}$$

Additionally, since $\mathbf{g}^{(i,t)}$ is independent of $\mathbf{s}^{(i,t)}$, by the

[3] The parameter T of `Non-mon. Frank-Wolfe` was renamed to L here to accommodate the standard notation in both offline and online algorithms. In offline Frank-Wolfe-like algorithms, the number of iterations is usually denoted by T, and in online algorithms T is reserved to the number of time steps.

guarantee of $\mathcal{E}_i$,

$$\mathbb{E}\left[\sum_{t=1}^{T}\langle \mathbf{s}^{(i,t)}, \mathbf{g}^{(i,t)}\rangle\right] \geq \mathbb{E}\left[\sum_{t=1}^{T}\langle \mathbf{o}, \mathbf{g}^{(i,t)}\rangle\right] - \mathcal{R}(T) \ .$$

Finally, since $\mathbf{g}^{(i,t)}$ is chosen after $\mathbf{y}^{(i-1,t)}$,

$$\begin{aligned}
&\mathbb{E}[\langle \mathbf{s}^{(i,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) - \mathbf{g}^{(i,t)}\rangle \mid \mathbf{s}^{(i,t)}, \mathbf{y}^{(i-1,t)}] \\
&= \langle \mathbf{s}^{(i,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) - \mathbb{E}[\mathbf{g}^{(i,t)} \mid \mathbf{y}^{(i-1,t)}]\rangle \\
&= \langle \mathbf{s}^{(i,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) - \nabla F_t(\mathbf{y}^{(i-1,t)})\rangle = 0 \ ,
\end{aligned}$$

which by the law of total expectation implies the equality $\mathbb{E}[\langle \mathbf{s}^{(i,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)}) - \mathbf{g}^{(i,t)}\rangle] = 0$. Combining all the above inequalities yields

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} F_t(\mathbf{y}^{(i,t)})\right] \\
&\geq \mathbb{E}\left[\sum_{t=1}^{T} F_t(\mathbf{y}^{(i-1,t)})\right] + \varepsilon \cdot \left\{\sum_{t=1}^{T}\langle o, \mathbb{E}[\mathbf{g}^{(i,t)}]\rangle - \mathcal{R}(T)\right. \\
&\qquad \left. -\mathbb{E}\left[\sum_{t=1}^{T}\langle \mathbf{y}^{(i-1,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)})\rangle\right]\right\} - 0.5\varepsilon^2\beta D^2 T \\
&= \mathbb{E}\left[\varepsilon \cdot \sum_{t=1}^{T}\langle o - \mathbf{y}^{(i-1,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)})\right. \\
&\qquad \left. + \sum_{t=1}^{T} F_t(\mathbf{y}^{(i-1,t)})\rangle\right] - \varepsilon \cdot \mathcal{R}(T) - 0.5\varepsilon^2\beta D^2 T \ . \quad \square
\end{aligned}$$

Corollary 4.4. *For every integer number $1 \leq i \leq L$, $\mathbb{E}[\sum_{t=1}^{T} F_t(\mathbf{y}^{(i,t)})] \geq \mathbb{E}[(1-2\varepsilon) \cdot \sum_{t=1}^{T} F_t(\mathbf{y}^{(i-1,t)}) + \varepsilon(1-\varepsilon)^{i-1} \cdot \sum_{t=1}^{T}(1 - \|\mathbf{y}^{(0,t)}\|_\infty) \cdot F_t(\mathbf{o})] - \varepsilon \cdot \mathcal{R}(T) - 0.5\varepsilon^2\beta D^2 T$.*

Proof. To see why this corollary follows from Lemma 4.3, it suffices to observe that, for every integer $1 \leq t \leq T$,

$$\begin{aligned}
&\langle \mathbf{o} - \mathbf{y}^{(i-1,t)}, \nabla F_t(\mathbf{y}^{(i-1,t)})\rangle \\
&\geq F_t(\mathbf{o} \vee \mathbf{y}^{(i-1,t)}) + F_t(\mathbf{o} \wedge \mathbf{y}^{(i-1,t)}) - 2F_t(\mathbf{y}^{(i-1,t)}) \\
&\geq F_t(\mathbf{o} \vee \mathbf{y}^{(i-1,t)}) - 2F_t(\mathbf{y}^{(i-1)}) \\
&\geq (1-\varepsilon)^{i-1} \cdot (1 - \|\mathbf{y}^{(0,t)}\|_\infty) \cdot F_t(\mathbf{o}) - 2F_t(\mathbf{y}^{(i-1,t)}) \ ,
\end{aligned}$$

where the first inequality follows from Lemma 2.1, the second inequality holds by the non-negativity of F_t, and the last inequality follows from Lemma 2.2 and the observation that the proof of Observation 3.2 extends to Algorithm 2 and yields $1 - \|y^{(i,t)}\|_\infty \leq (1-\varepsilon)^i \cdot (1 - \|\mathbf{y}^{(0,t)}\|_\infty)$. $\square$

One can observe that Corollary 4.4 is very similar to Lemma 3.3 (the main difference between the two is that in Corollary 4.4 the sum $\sum_{t=1}^{T} F_t$ replaces the function F from Lemma 3.3). This similarity means that the proof of Theorem 3.1 can work with Corollary 4.4 instead of Lemma 3.3, which yields Theorem 4.1.

5 INAPPROXIMABILITY

This section includes our inapproximability result, which is given by the following theorem. Our result shows that the known offline result (reproved in Section 3) for maximizing a DR-submodular function subject to a general convex set is optimal. Notice that this implies that our online algorithm from Section 4 is also optimal (at least in terms of the approximation ratio) unless one allows for an exponential time complexity.

Theorem 5.1. *For every two constants $h \in [0,1)$ and $\varepsilon > 0$, no sub-exponential time algorithm can obtain $(1/4(1-h)+\varepsilon)$-approximation for the problem of maximizing a continuously differentiable non-negative DR-submodular function $F\colon [0,1]^n \to \mathbb{R}_{\geq 0}$ subject to a solvable polytope $\mathcal{K}$ obeying $\min_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x}\|_\infty = h$. Furthermore, this is true even if we are guaranteed that $\max_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x}) = \Omega(n^{-1})$ and F is β-smooth for some β that is polynomial in n.*

The last part of Theorem 5.1 specifies some additional conditions under which the inapproximability stated in the theorem still applies. These conditions are important because under them our algorithm from Section 3 can be made to have a clean approximation guarantee of $1/4(1 - \min_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x}\|_\infty) - \varepsilon'$, for any constant $\varepsilon' > 0$, by choosing a polynomially small value for the parameter ε of the algorithm (to see that this is indeed the case, it is important to observe that since $\mathcal{K} \subseteq [0,1]^n$, the diameter D of $\mathcal{K}$ is at most $\sqrt{n}$).

Theorem 5.1 is unconditional, i.e., it does not rely on any complexity assumption. Instead, Theorem 5.1 assumes a constraint on the way in which the algorithm may access the objective F. It is standard in the field to assume that the algorithm can access F only by querying the value or gradient of F at a given point $\mathbf{x}$. Theorem 5.1 applies under this standard assumption, and furthermore, it applies even when the algorithm is allowed any query about F whose output is determined by the values of F in an arbitrarily small neighborhood of a point $\mathbf{x}$. Note that the standard queries of value and gradient at $\mathbf{x}$ both fall within this class of queries, and the same is true for other natural kind of queries (such as higher order derivatives of F).

The proof of Theorem 5.1 is based on the symmetry gap framework of Vondrák (2013). To use this framework, we first need to choose a submodular set function f_k ($k \geq 1$ is an integer parameter of the function). We choose the same function that was used by Vondrák (2013) to prove his hardness for maximizing a submodular function subject to a matroid base constraint. Specifically, the ground set of f_k is the set $\mathcal{N}_k = \{a_i, b_i \mid i \in [k]\}$, and for every set $S \subseteq \mathcal{N}_k$,

$$f_k(S) = \sum_{i=1}^{k} \mathbf{1}[a_i \in S] \cdot \mathbf{1}[b_i \notin S] \ .$$

One can verify that f_k is non-negative and submodular since it is the cut function of a directed graph consisting of k vertex-disjoint arcs.

We now would like to convert f_k into two DR-submodular functions, which we do using the following lemma of Vondrák (2013). This lemma refers to the multilinear extension of a set function $f\colon 2^{\mathcal{N}} \to \mathbb{R}$ over a ground set $\mathcal{N}$. This extension is a function $F\colon [0,1]^{\mathcal{N}} \to \mathbb{R}$ defined for every vector $\mathbf{x} \in [0,1]^{\mathcal{N}}$ by $F(\mathbf{x}) = \mathbb{E}[f(\mathtt{R}(\mathbf{x}))]$, where $\mathtt{R}(\mathbf{x})$ is a random subset of $\mathcal{N}$ that includes every element $u \in \mathcal{N}$ with probability x_u, independently.

Lemma 5.2 (Lemma 3.2 of Vondrák (2013))**.** *Consider a function $f\colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ invariant under a group of permutations $\mathcal{G}$ on the ground set $\mathcal{N}$. Let $F(\mathbf{x})$ be the multilinear extension of f, define $\bar{x} = \mathbb{E}_{\sigma\in\mathcal{G}}[\mathbf{1}_{\sigma(\mathbf{x})}]$ and fix any $\varepsilon' > 0$. Then, there is $\delta > 0$ and functions $\hat{F}, \hat{G}\colon [0,1]^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$ (which are also symmetric with respect to $\mathcal{G}$), satisfying the following:*

1. *For all $\mathbf{x} \in [0,1]^{\mathcal{N}}$, $\hat{G}(\mathbf{x}) = \hat{F}(\bar{\mathbf{x}})$.*
2. *For all $\mathbf{x} \in [0,1]^{\mathcal{N}}$, $|\hat{F}(\mathbf{x}) - F(\mathbf{x})| \leq \varepsilon'$.*
3. *Whenever $\|\mathbf{x} - \bar{\mathbf{x}}\|_2 \leq \delta$, $\hat{F}(\mathbf{x}) = \hat{G}(\mathbf{x})$ and the value depends only on $\bar{\mathbf{x}}$.*
4. *The first partial derivatives of $\hat{F}$ and $\hat{G}$ are absolutely continuous.*
5. *If f is monotone, then, for every element $u \in \mathcal{N}$, $\frac{\partial \hat{F}}{\partial x_u} \geq 0$ and $\frac{\partial \hat{G}}{\partial x_u} \geq 0$ everywhere.*
6. *If f is submodular then, for every two elements $u, v \in \mathcal{N}$, $\frac{\partial^2 \hat{F}}{\partial x_u \partial x_v} \leq 0$ and $\frac{\partial^2 \hat{G}}{\partial x_u \partial x_v} \leq 0$ almost everywhere.*

Observe that f_k is invariant to exchanging the identities of a_i and b_i with a_j and b_j, respectively, for any choice of $i, j \in [k]$. Therefore, we can choose $\mathcal{G}$ in the last lemma as the group of permutations that can be obtained by any number of such exchanges. In the rest of this section, we assume that $\hat{F}_k$ and $\hat{G}_k$ are functions $\hat{F}$ and $\hat{G}$ obtained using Lemma 5.2 for this choice of $\mathcal{G}$, f_k and $\varepsilon' = 1/(2k)$. It is also important to note that for this choice of $\mathcal{G}$ we have for every vector $\mathbf{x} \in [0,1]^{\mathcal{N}_k}$ and $i \in [k]$

$$\bar{x}_{a_i} = \frac{1}{k}\sum_{j=1}^{k} x_{a_j} \qquad \text{and} \qquad \bar{x}_{b_i} = \frac{1}{k}\sum_{j=1}^{k} x_{b_j} \ .$$

Let us now define a family of polytopes. The polytope $\mathcal{P}_{h,k}$ is the convex hull of the $k+1$ vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}$ and $\mathbf{u}$ defined as follows. For every $j \in [k]$, $u_{a_j} = 0$ and $u_{b_j} = h$. For every $i, j \in [k]$,

$$v^{(i)}_{a_j} = \begin{cases} 1 & \text{if } i = j \ , \\ 0 & \text{otherwise} \ , \end{cases} \quad \text{and} \quad v^{(i)}_{b_j} = \begin{cases} 1 & \text{if } i \neq j \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

Using the above definitions, we can state two instances of the problem we consider

$$\max \hat{F}_k(\mathbf{x}) \\ \mathbf{x} \in \mathcal{P}_{h,k} \qquad \text{and} \qquad \max \hat{G}_k(\mathbf{x}) \\ \mathbf{x} \in \mathcal{P}_{h,k} \ .$$

In Appendix C we refer to these instances as the *basic* instances. We show there that by "scrambling" these instances in an appropriate way, they can be made indistinguishable. This yields Theorem 5.1 as we also prove in Appendix C that the scrambled instances obey the properties assumed in the theorem, and furthermore, that there is a large gap between the optimal values of scrambled instances derived from the two basic instances.

6 APPLICATIONS AND EXPERIMENTAL RESULTS

Up until recently, all the algorithms suggested for submodular maximization subject to general convex set constraints had a sub-exponential execution time. As mentioned above, Du (2022) has recently shown the first polynomial time offline algorithm for this problem, and in this paper we have shown another polynomial time algorithm obtaining a similar guarantee for the online (regret minimization) setting. In this section (and Appendix D), we study the empirical performance of these algorithms on the machine learning applications of revenue maximization, location summarization and quadratic programming. We note that these are just a few examples of standard applications to which our results can be applied (other possible applications include, for example, movie recommendation and image summarization).

In the case of the offline algorithm, it is important to note that (i) we analyze our explicit version of the algorithm, rather than the original version of Du (2022); and (ii) it is interesting to study the empirical performance of the algorithm of Du (2022) because only a theoretical analysis of this algorithm appeared in (Du, 2022).

Since the previously suggested algorithms require sub-exponential execution time, and thus cannot be used as is, we allowed all algorithms in our experiments the same number of iterations. This makes all the algorithms terminate in roughly the same amount of time, and allows for a fair comparison between the quality of their solutions. In a nutshell, our experiments show that our online algorithm and the offline algorithm of Du (2022) provide better solutions (often much better) compared to their state-of-the-art sub-exponential time counterparts.

6.1 Revenue Maximization

Following Thắng and Srivastav (2021), our first set of experiments considers revenue maximization in the following setting. The goal of a company is to advertise a product

to users so that the revenue increases through the "word-of-mouth" effect. Formally, the input for the problem is a weighted undirected graph $G = (V, E)$ representing a social network graph, where w_{ij} denotes the weight of the edge between vertex i and vertex j ($w_{ij} = 0$ if the edge (i, j) is missing from the graph). If the company invests x_i unit of cost in a user $i \in V$, then this user becomes an advocate of the product with probability $1 - (1 - p)^{x_i}$, where $p \in (0, 1)$ is a parameter. Note that this means that each ε unit of cost invested in the user has an independent chance to make the user an advocate, and that by investing a full unit in the user, she becomes an advocate with probability p (Soma and Yoshida, 2017).

Let $S \subseteq V$ be a set of users who ended up being advocates for the product. Then, the revenue obtained is represented by the total influence of the users of S on non-advociate users, or more formally, by $\sum_{i \in S} \sum_{j \in V \setminus S} w_{ij}$. The objective function $f \colon [0, 1]^V \to \mathbb{R}_{\geq 0}$ of the experiments is accordingly defined as the expectation of the above expression, i.e.,

$$\begin{aligned} f(\mathbf{x}) &= \mathbb{E}_S \left[\sum_{i \in S} \sum_{j \in V \setminus S} w_{ij} \right] \\ &= \sum_{i \in V} \sum_{\substack{j \in V \\ i \neq j}} w_{ij} (1 - (1 - p)^{x_i})(1 - p)^{x_j} \quad . \end{aligned}$$

It has been shown that f is a non-monotone DR-submodular function (Soma and Yoshida, 2017).

In both the online and offline settings, we experimented on instances of the above setting based on two different datasets. The first is a Facebook network (Viswanath et al., 2009), and includes $64K$ users (vertices) and $1M$ unweighted relationships (edges). The second dataset is based on the Advogato network (Massa et al., 2009), and includes $6.5K$ users (vertices) as well as $61K$ weighted relationships (edges).

6.1.1 Online setting

When performing our experiments in the online settings, we tried to closely mimic the experiment of Thắng and Srivastav (2021). Therefore, we chose the number of time steps to be $T = 1000$, and the parameter $p = 0.0001$. In each time step t, the objective function is defined in the following way. A subset $V^t \subseteq V$ is selected, and only edges connecting two vertices of V^t are kept. In the case of the Advogato network, V_t is a uniformly random subset of V of size 200, and in the case of the much larger Facebook network, V_t is a uniformly random subset of V of size 15,000. The optimization is done subject to the constraint $0.1 \leq \sum_i x_i \leq 1$, which represents both minimum and maximum investment requirements. Note that the intersection of this constraint with the implicit box constraint represents a non-down-monotone feasibility polytope.

In our experiments, we have compared our algorithm from Section 4 with the algorithm of Thắng and Srivastav (2021), which is the only other algorithm for the online setting currently known. In both algorithms, we have set the number of online linear optimizers used to be $L = 100$, and in our algorithm we have set the error parameter $\varepsilon = 0.03$ (there is no error parameter in the algorithm of Thắng and Srivastav (2021)). The results of these experiments on the Advogato and Facebook networks can be found in Figures 1a and 1b, respectively. One can observe that our algorithm significantly outperforms the state-of-the-art algorithm for any number of time steps.

6.1.2 Offline setting

Our experiments in the offline setting are similar to the ones done in the online setting, with two differences. First, since there is only one objective function in the offline setting, we base it on the entire network graph rather than on a subset of its vertices. Second, for the sake of diversity, we changed the constraint to be $0.25 \leq \sum_i x_i \leq 1$ (but we note that the results of the experiments remain essentially unchanged if one reuse the constraint from the online setting).

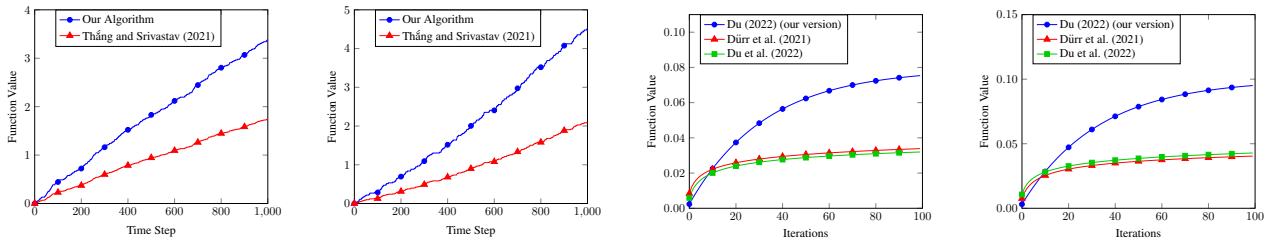
In our experiments, we have compared our explicit version from Section 3 of the algorithm of Du (2022) with the previous algorithms of Dürr et al. (2021) and Du et al. (2022). All the algorithms have been executed for $T = 100$ iterations,[4] and the error parameter ε was set 0.03 in (our version of) the algorithm of Du (2022). The results of these experiments on the Advogato and Facebook networks can be found in Figures 1c and 1d, respectively. One can observe that our version of the polynomial time algorithm of Du (2022) clearly outperforms the two previous algorithms, except when the number of iterations is very low.

6.2 Location Summarization

In this section we consider a location summarization task based on the Yelp dataset (Yelp), which is a subset of Yelp's businesses, reviews and user data. This dataset contains information about local businesses across 11 metropolitan areas, and we have followed the technique of Kazemi et al. (2021) for generating symmetry scores between these locations based on features extracted from the descriptions of the locations and their related user reviews (such as parking options, WiFi access, having vegan menus).

We would like to pick a non-empty set of up to 2 locations that summarizes the existing locations, while not being too far from the current location of the user. A natural objective function for this task (which is very similar to the objective function used in (Kazemi et al., 2021)) is the following set function. Assume that the set of locations is $[n]$, $M_{i,j}$ is

[4] Recall that the number of iterations corresponds to the parameter L in the online setting, which was also set to 100 above.



(a) Online Algorithms on the Advogato network.

(b) Online Algorithms on the Facebook network.

(c) Offline Algorithms on the Advogato network.

(d) Offline Algorithms on the Facebook network.

Figure 1: Results of the Revenue Maximization Experiments

the similarity score between locations i and j, and d_i is the distance of location i from the user (in units of 200KM); then for every set $S \subseteq [n]$, the value of the objective is $f(S) = \frac{1}{n}\sum_{i=1}^{n} \max_{j \in S} M_{i,j} - \sum_{i \in S} d_i$.

Since f is a set function, and the tools we have developed in this work apply only to continuous functions, we optimize the multilinear extension F of f,[5] which is given for every vector $\mathbf{x} \in [0,1]^n$ by

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[x_j M_{i,j} \cdot \prod_{j' | M_{i,j} \prec M_{i,j'}} (1 - x_{j'})\right] - \sum_{i=1}^{n} x_i d_i \ .$$

The multilinear extension F is DR-submodular since f is submodular. Moreover, any solution obtained while optimizing F can be rounded into a solution obtaining the same approximation guarantee for f using either pipage or swap rounding (Calinescu et al., 2011; Chekuri et al., 2010).

In our experiment, we restricted attention to a single metropolitan area (Charlotte), and assumed there are 100 time steps. In each time step, a new user u arrives, and her location is determined uniformly at random within the rectangle containing the metropolitan area. Let us denote by F_u the function F when the distances are calculated based on the location of u. When user u arrives, we would like to choose a vector $\mathbf{x}^{(u)}$ maximizing F_u among all vectors obeying $\|\mathbf{x}\|_1 \in [1,2]$ (recall that we look for solutions that include 1 or 2 locations). Furthermore, we would like to do that before learning the location of u (to speed up the response and for privacy reasons); thus, we need to consider online optimization algorithms. Specifically, like in Section 6.1.1, we compared our algorithm from Section 4 with the algorithm of Thắng and Srivastav (2021). In both algorithms, we have set the number of online linear optimizers used to be $L = 100$, and in our algorithm we have set the error parameter $\varepsilon = 0.03$. The results of the experiment can be found in Figure 2, and they show that our algorithm (again) significantly outperforms the state-of-the-art algorithm for any number of time steps.

[5] See Section 5 for a definition of the multi-linear extension.

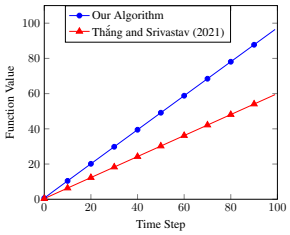


Figure 2: Location Summarization Experiment

7 CONCLUSION

In this work, we have considered the problem of maximizing a DR-submodular function over a general convex set in both the offline and the online (regret minimization) settings. For the online setting we provided the first polynomial time algorithm. Our algorithm matches the approximation guarantee of the only polynomial time algorithm known for the offline setting. Moreover, we presented a hardness result showing that this approximation guarantee is optimal for both settings. Finally, we have run experiments to study the empirical performance of both our algorithm and the (recently suggested) polynomial time offline algorithm. Our experiments show that both these algorithms outperform previous benchmarks.

Acknowledgements

This work was supported in part by Israel Science Foundation (ISF) grant number 459/20.

References

Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory (COLT)*, pages 263–273, 2008. URL https://www.learningtheory.org/colt2008/papers/123-Abernethy.pdf.

An Bian, Kfir Yehuda Levy, Andreas Krause, and Joachim M. Buhmann. Non-monotone continuous DR-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 486–496, 2017a. URL https://proceedings.neurips.cc/paper/2017/hash/58238e9ae2dd305d79c2ebc8c1883422-Abstract.html.

An Bian, Kfir Y Levy, Andreas Krause, and Joachim M Buhmann. Non-monotone continuous DR-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 487–497. Curran, 2018.

Andrew An Bian, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *International Conference on Machine Learning (ICML)*, pages 498–507, 2017b. URL http://proceedings.mlr.press/v70/bian17a.html.

Yatao Bian, Joachim Buhmann, and Andreas Krause. Optimal continuous DR-submodular maximization and applications to provable mean field inference. In *International Conference on Machine Learning (ICML)*, pages 644–653. PMLR, 2019.

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.

Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Foundations of Computer Science (FOCS)*, pages 575–584. IEEE Computer Society, 2010. doi: 10.1109/FOCS.2010.60. URL https://doi.org/10.1109/FOCS.2010.60.

Chandra Chekuri, T. S. Jayram, and Jan Vondrák. On multiplicative weight updates for concave and submodular function maximization. In Tim Roughgarden, editor, *Innovation in Theoretical Computer Science (ITCS)*, pages 201–210. ACM, 2015. doi: 10.1145/2688073.2688086. URL https://doi.org/10.1145/2688073.2688086.

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

Lin Chen, Mingrui Zhang, and Amin Karbasi. Projection-free bandit convex optimization. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pages 2047–2056. PMLR, 2019. URL http://proceedings.mlr.press/v89/chen19f.html.

Donglei Du. Lyapunov function approach for approximation algorithm design and analysis: with applications in submodular maximization. *CoRR*, abs/2205.12442, 2022. doi: 10.48550/arXiv.2205.12442. URL https://doi.org/10.48550/arXiv.2205.12442.

Donglei Du, Zhicheng Liu, Chenchen Wu, Dachuan Xu, and Yang Zhou. An improved approximation algorithm for maximizing a DR-submodular function over a convex set. *arXiv preprint arXiv:2203.14740*, 2022.

Christoph Dürr, Nguyên Kim Thắng, Abhinav Srivastav, and Léo Tible. Non-monotone DR-submodular maximization over general convex sets. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2148–2154, 2021.

Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011.

Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Foundations of Computer Science (FOCS)*, pages 570–579, 2011.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017a.

S. Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *CoRR*, abs/1708.03949, 2017b. URL http://arxiv.org/abs/1708.03949.

Ehsan Kazemi, Shervin Minaee, Moran Feldman, and Amin Karbasi. Regularized submodular maximization at scale. In Marina Meila and Tong Zhang, editors, *International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 5356–5366. PMLR, 2021. URL http://proceedings.mlr.press/v139/kazemi21a.html.

Paolo Massa, Martino Salvetti, and Danilo Tomasoni. Bowling alone and trust decline in social network sites. In *IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC)*, pages 658–663. IEEE Computer Society, 2009. doi: 10.1109/DASC.2009.130. URL https://doi.org/10.1109/DASC.2009.130.

Siddharth Mitra, Moran Feldman, and Amin Karbasi. Submodular + concave. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11577–11591, 2021. URL https://proceedings.

neurips.cc/paper/2021/hash/602443a3d6907117d8b4a308844e963e-Abstract.html.

Loay Mualem and Moran Feldman. Using partial monotonicity in submodular maximization. *arXiv preprint arXiv:2202.03051*, 2022.

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

Rad Niazadeh, Tim Roughgarden, and Joshua R Wang. Optimal algorithms for continuous non-monotone submodular and DR-submodular maximization. *Journal of Machine Learning Research*, 21(1):4937–4967, 2020.

Tasuku Soma and Yuichi Yoshida. Non-monotone DR-submodular function maximization. In Satinder Singh and Shaul Markovitch, editors, *AAAI Conference on Artificial Intelligence*, pages 898–904. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14483.

Nguyễn Kim Thắng and Abhinav Srivastav. Online non-monotone DR-submodular maximization. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 9868–9876. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17186.

Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in facebook. In *ACM SIGCOMM Workshop on Social Networks (WOSN)*, August 2009.

Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM J. Comput.*, 42 (1):265–304, 2013. doi: 10.1137/110832318. URL https://doi.org/10.1137/110832318.

Wei Xia, Juan-Carlos Vera, and Luis F. Zuluaga. Globally solving nonconvex quadratic programs via linear integer programming techniques. *INFORMS J. Comput.*, 32(1):40–56, 2020. doi: 10.1287/ijoc.2018.0883. URL https://doi.org/10.1287/ijoc.2018.0883.

Yelp. Yelp Dataset. https://www.yelp.com/dataset, 2019.

Mingrui Zhang, Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization: From full-information to bandit feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

A PROOF OF LEMMA 2.2

In this section we prove Lemma 2.2, which we repeat here for convenience.

Lemma 2.2. *For every two vectors $\mathbf{x}, \mathbf{y} \in [0,1]^n$ and any continuously differentiable non-negative DR-submodular function $F\colon [0,1]^n \to \mathbb{R}_{\geq 0}$, $F(\mathbf{x} \vee \mathbf{y}) \geq (1 - \|\mathbf{x}\|_\infty) F(\mathbf{y})$.*

Proof. If $\|\mathbf{x}\|_\infty = 0$, then $\mathbf{x}$ is the all zeros vector, and the lemma becomes trivial. Thus, we may assume in the rest of this proof that $\|\mathbf{x}\|_\infty > 0$. Let $\mathbf{z} = \mathbf{x} \vee \mathbf{y} - \mathbf{y}$. Then,

$$\begin{aligned} F(\mathbf{x} \vee \mathbf{y}) - F(\mathbf{y}) &= \int_0^1 \left.\frac{dF(\mathbf{y} + r \cdot \mathbf{z})}{dr}\right|_{r=t} dt = \int_0^1 \sum_{i=1}^n \langle \mathbf{z}, \nabla F(\mathbf{y} + t \cdot \mathbf{z}) \rangle dt \\ &= \|\mathbf{x}\|_\infty \cdot \int_0^{1/\|\mathbf{x}\|_\infty} \sum_{i=1}^n \langle \mathbf{z}, \nabla F(\mathbf{y} + \|\mathbf{x}\|_\infty \cdot t' \cdot \mathbf{z}) \rangle dt' \\ &\geq \|\mathbf{x}\|_\infty \cdot \int_0^{1/\|\mathbf{x}\|_\infty} \sum_{i=1}^n \langle \mathbf{z}, \nabla F(\mathbf{y} + t' \cdot \mathbf{z}) \rangle dt' \ , \end{aligned} \tag{1}$$

where the last equality holds by changing the integration variable to $t' = t/\|\mathbf{x}\|_\infty$, and the inequality follows from the DR-submodularity of F because $\mathbf{y} + t' \cdot \mathbf{z} \in [0,1]^n$. To see that the last inclusion holds, note that, for every $i \in [n]$, if $x_i \leq y_i$, then $y_i + t' \cdot z_i = y_i \leq 1$, and if $x_i \geq y_i$, then

$$y_i + t' \cdot z_i \leq y_i + \frac{z_i}{\|\mathbf{x}\|_\infty} = y_i + \frac{x_i - y_i}{\|\mathbf{x}\|_\infty} \leq \frac{x_i}{\|\mathbf{x}\|_\infty} \leq 1 \ .$$

Observe now that we also have

$$\begin{aligned} \int_0^{1/\|\mathbf{x}\|_\infty} \sum_{i=1}^n \langle \mathbf{z}, \nabla F(\mathbf{y} + t' \cdot \mathbf{z}) \rangle dt' &= \int_0^{1/\|\mathbf{x}\|_\infty} \left.\frac{dF(\mathbf{y} + r \cdot \mathbf{z})}{dr}\right|_{r=t'} dt' \\ &= F\left(\mathbf{y} + \frac{\mathbf{z}}{\|\mathbf{x}\|_\infty}\right) - F(\mathbf{y}) \geq -F(\mathbf{y}) \ , \end{aligned}$$

where the inequality follows from the non-negativity of F. The lemma now follows by plugging this inequality into Inequality (1), and rearranging. □

B MISSING PROOFS OF SECTION 3

B.1 Proof of Observation 3.2

In this section we prove observation 3.2, which we repeat here for convenience.

Observation 3.2. *For every integer $0 \leq i \leq T$, $1 - \|\mathbf{y}^{(i)}\|_\infty \geq (1-\varepsilon)^i \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty)$.*

Proof. To prove the observation, we show by induction that for every fixed coordinate $j \in [n]$, we have $1 - y_j^{(i)} \geq (1-\varepsilon)^i \cdot (1 - y_j^{(0)})$. For $i = 0$, this inequality trivially holds. Furthermore, assuming this inequality holds for $i-1$, it also holds for i because

$$\begin{aligned} 1 - y_j^i &= 1 - (1-\varepsilon) y_j^{(i-1)} - \varepsilon s_j^{(i)} \\ &\geq 1 - (1-\varepsilon) y_j^{(i-1)} - \varepsilon \\ &= (1-\varepsilon)(1 - y_j^{(i-1)}) \\ &\geq (1-\varepsilon)^i \cdot (1 - y_j^{(0)}) \ , \end{aligned}$$

where the second inequality follows from the induction hypothesis. □

B.2 Proof of Lemma 3.3

In this section we prove Lemma 3.3, which we repeat here for convenience.

Lemma 3.3. *For every integer* $1 \leq i \leq T$, $F(\mathbf{y}^{(i)}) \geq (1-2\varepsilon)\cdot F(\mathbf{y}^{(i-1)}) + \varepsilon(1-\varepsilon)^{i-1}\cdot(1-\|\mathbf{y}^{(0)}\|_\infty)\cdot F(\mathbf{o}) - 0.5\varepsilon^2\beta D^2$.

Proof. By the chain rule,

$$\begin{aligned}
F(\mathbf{y}^{(i)}) - F(\mathbf{y}^{(i-1)}) &= F((1-\varepsilon)\cdot\mathbf{y}^{(i-1)} + \varepsilon\cdot\mathbf{s}^{(i)}) - F(\mathbf{y}^{(i-1)}) \\
&= \int_0^\varepsilon \left.\frac{F((1-z)\cdot\mathbf{y}^{(i-1)} + z\cdot\mathbf{s}^{(i)})}{dz}\right|_{z=r} dr \\
&= \int_0^\varepsilon \langle \mathbf{s}^{(i)} - \mathbf{y}^{(i-1)}, \nabla F((1-r)\cdot\mathbf{y}^{(i-1)} + r\cdot\mathbf{s}^{(i)})\rangle dr \\
&\geq \int_0^\varepsilon \Big[\langle \mathbf{s}^{(i)} - \mathbf{y}^{(i-1)}, \nabla F(\mathbf{y}^{(i-1)})\rangle - r\beta D^2\Big] dr \\
&= \varepsilon\cdot\langle \mathbf{s}^{(i)} - \mathbf{y}^{(i-1)}, \nabla F(\mathbf{y}^{(i-1)})\rangle - 0.5\varepsilon^2\beta D^2,
\end{aligned}$$

where the inequality follows from the β-smoothness of F. Recall now that $s^{(i)}$ is the maximizer found by Algorithm 1 in its i-th iteration, and $\mathbf{o}$ is one of the values in the domain on which the maximum is calculated. Therefore,

$$\begin{aligned}
F(\mathbf{y}^{(i)}) - F(\mathbf{y}^{(i-1)}) &\geq \varepsilon\cdot\langle \mathbf{s}^{(i)} - \mathbf{y}^{(i-1)}, \nabla F(\mathbf{y}^{(i-1)})\rangle - 0.5\varepsilon^2\beta D^2 \\
&\geq \varepsilon\cdot\langle \mathbf{o} - \mathbf{y}^{(i-1)}, \nabla F(\mathbf{y}^{(i-1)})\rangle - 0.5\varepsilon^2\beta D^2 \\
&\geq \varepsilon\cdot\Big[F(\mathbf{o}\vee\mathbf{y}^{(i-1)}) + F(\mathbf{o}\wedge\mathbf{y}^{(i-1)}) - 2F(\mathbf{y}^{(i-1)})\Big] - 0.5\varepsilon^2\beta D^2 \\
&\geq \varepsilon\cdot\Big[(1-\varepsilon)^{i-1}\cdot(1-\|\mathbf{y}^{(0)}\|_\infty)\cdot F(\mathbf{o}) - 2F(\mathbf{y}^{(i-1)})\Big] - 0.5\varepsilon^2\beta D^2.
\end{aligned}$$

where the third inequality follows from Lemma 2.1, and the last inequality from Lemma 2.2, Observation 3.2 and the non-negativity of F. The lemma now follows by rearranging the last inequality. □

B.3 Proof of Theorem 3.1

In this section we prove Theorem 3.1, which we repeat here for convenience.

Theorem 3.1. *Let* $\mathcal{K} \subseteq [0,1]^n$ *be a general convex set, and let* $F\colon [0,1]^n \to \mathbb{R}_{\geq 0}$ *be a non-negative* β*-smooth DR-submodular function. Then,* `Non-mon. Frank-Wolfe` *(Algorithm 1) outputs a solution* $\mathbf{w} \in \mathcal{K}$ *obeying*

$$\begin{aligned}
F(\mathbf{w}) \geq (1-2\varepsilon)^{T-1}[(1+\varepsilon)^T - 1](1 - \min_{\mathbf{x}\in\mathcal{K}}\|\mathbf{x}\|_\infty)\cdot F(\mathbf{o}) \\
- 0.5\varepsilon^2\beta D^2 T \ ,
\end{aligned}$$

where D *is the diameter of* $\mathcal{K}$ *and* $\mathbf{o} \in \arg\max_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x})$. *In particular, when* T *is set to be* $\lfloor \ln 2/\varepsilon \rfloor$,

$$F(\mathbf{w}) \geq (1/4 - 3\varepsilon)(1 - \min_{\mathbf{x}\in\mathcal{K}}\|\mathbf{x}\|_\infty)\cdot F(\mathbf{o}) - 0.5\varepsilon\beta D^2 \ .$$

Proof. To see that the second part of the theorem follows from the first part, note that for $T = \lfloor \ln 2/\varepsilon \rfloor$ and $\varepsilon < 1/4$,

$$\begin{aligned}
(1-2\varepsilon)^{T-1}[(1+\varepsilon)^T - 1] &\geq e^{-2\varepsilon T}(1-4\varepsilon^2 T)[e^{\varepsilon T}(1-\varepsilon^2 T) - 1] \\
&\geq e^{-2\ln 2}(1 - 4\varepsilon\ln 2)[e^{\ln 2 - \varepsilon}(1 - \varepsilon\ln 2) - 1] \\
&= \left(\frac{1}{4} - \varepsilon\ln 2\right)\left[\frac{2 - 2\varepsilon\ln 2}{e^\varepsilon} - 1\right] \\
&\geq \left(\frac{1}{4} - \varepsilon\right)\left[\frac{2-2\varepsilon}{1+2\varepsilon} - 1\right] \\
&= \left(\frac{1}{4} - \varepsilon\right)\cdot\frac{1-4\varepsilon}{1+2\varepsilon} \\
&\geq \frac{1}{4} - 3\varepsilon \ .
\end{aligned}$$

For $\varepsilon \geq 1/4$, the second part of the theorem is an immediate consequence of the non-negaitivity of F.

It remains to prove the first part of the theorem. We do that by proving by induction the stronger claim that for every integer $0 \leq i \leq T$,

$$F(\mathbf{y}^{(i)}) \geq (1-2\varepsilon)^{i-1}\big[(1+\varepsilon)^i - 1\big] \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2 \beta D^2 i \ . \tag{2}$$

Note that the theorem indeed follows from this claim because w is the best vector within a set that includes $\mathbf{y}^{(T)}$, and $\mathbf{y}^{(0)} \in \arg\min_{\mathbf{x}\in\mathcal{K}} \|x\|_\infty$. For $i = 0$, Equation (2) follows directly from the non-negativity of F. Hence, we only need to show that for $1 \leq i \leq T$, if we assume that Equation (2) holds for $i-1$, then it holds for i as well. This is indeed the case because Lemma 3.3 yields

$$\begin{aligned}
F(\mathbf{y}^{(i)}) &\geq (1-2\varepsilon) \cdot F(\mathbf{y}^{(i-1)}) + \varepsilon(1-\varepsilon)^{i-1} \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2\beta D^2 \\
&\geq (1-2\varepsilon) \cdot \{(1-2\varepsilon)^{i-2}\big[(1+\varepsilon)^{i-1} - 1\big] \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2\beta D^2 (i-1)\} \\
&\quad + \varepsilon(1-\varepsilon)^{i-1} \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2\beta D^2 \\
&\geq \{(1-2\varepsilon)^{i-1}\big[(1+\varepsilon)^i - \varepsilon(1+\varepsilon)^{i-1} - 1\big] + \varepsilon(1-\varepsilon)^{i-1}\} \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2\beta D^2 i \\
&\geq (1-2\varepsilon)^{i-1}\big[(1+\varepsilon)^i - 1\big] \cdot (1 - \|\mathbf{y}^{(0)}\|_\infty) \cdot F(\mathbf{o}) - 0.5\varepsilon^2\beta D^2 i \ ,
\end{aligned}$$

where the second inequality follows from the induction hypothesis, and the last inequality holds since

$$(1-2\varepsilon)^{i-1} \cdot \varepsilon(1+\varepsilon)^{i-1} = \varepsilon(1-\varepsilon-2\varepsilon^2)^{i-1} \leq \varepsilon(1-\varepsilon)^{i-1} \ . \qquad \square$$

C CONTINUING THE PROOF OF THEOREM 5.1

In this section, we complete the proof of Theorem 5.1. As explained in Section 5, the proof of Theorem 5.1 is based on showing that: (i) by "scrambling" the basic instances defined in Section 5 in an appropriate way, they can be made indistinguishable, (ii) the scrambled instances obey the properties assumed in the theorem, and (iii) there is a large gap between the optimal values of scrambled instances derived from the two basic instances. Towards this goal, we first study the properties of the basic instances themselves, and the gap between their optimal values. Let us begin with the following lemma, which gives some properties of the objective functions of the basic instances.

Lemma C.1. *The functions $\hat{F}_k$ and $\hat{G}_k$ are continuously differentiable, non-negative and DR-submodular. Furthermore, they are β-smooth for a value β that is polynomial in k.*

Proof. The non-negativity of $\hat{F}_k$ and $\hat{G}_k$ is explicitly guaranteed by Lemma 5.2, and Part 4 of the lemma shows that $\hat{F}$ and $\hat{G}$ are also continuously differentiable. Finally, Parts 4 and 6 of Lemma 5.2 imply together that $\hat{F}_k$ and $\hat{G}_k$ are DR-submodular (see the proof of Lemma 3.1 of Vondrák (2013) for a formal argument).

It remains to bound the smoothness of $\hat{F}_k$ and $\hat{G}_k$. Notice that the following claim implies that both functions are β-smooth for a β value that is polynomial in k. Unfortunately, the proof of this claim is technically quite involved (and not very insightful) as it requires us to look into the proof Lemma 5.2, and therefore, we defer the proof of this claim to Section C.1.

Claim C.2. *The absolute values of the second order partial derivatives of the functions $\hat{F}_k$ and $\hat{G}_k$ are bounded by $16k+2$ almost everywhere, and therefore, both functions are β-smooth for a β value that is polynomial in k.* $\square$

Next, we observe that the common constraint polytope of the basic instances is solvable since $\mathcal{P}_{h,k}$ is a polytope over $2k$ variables defined as the convex-hall of $k+1$ vectors. The next observation proves another property of this polytope.

Observation C.3. *If $k \geq 1/(1-h)$, $\min_{\mathbf{x}\in\mathcal{P}_{h,k}} \|\mathbf{x}\|_\infty = h$.*

Proof. Since $\mathbf{u} \in \mathcal{P}_{h,k}$, $\min_{\mathbf{x}\in\mathcal{P}_{h,k}} \|\mathbf{x}\|_\infty \leq h$. Thus, we only need to show that no point in $\mathcal{P}_{h,k}$ has an infinity norm less than h. Recall that every point in $\mathcal{P}_{h,k}$ is a convex combination $\sum_{i=1}^{k} c_i \mathbf{v}^{(i)} + d\mathbf{u}$ (where c_i is the coefficient of $\mathbf{v}^{(i)}$ in the combination, and d is the coefficient of $\mathbf{u}$), and assume without loss of generality that $c_1 = \min\{c_1, c_2, \ldots, c_k\}$. Then,

$$\left\|\sum_{i=1}^{k} c_i \mathbf{v}^{(i)} + d\mathbf{u}\right\|_\infty \geq \sum_{i=1}^{k} c_i v^{(i)}_{b_1} + d u_{b_1} = \sum_{i=2}^{k} c_i + dh \geq \frac{k-1}{k}\sum_{i=1}^{k} c_i + dh \geq h\sum_{i=1}^{k} c_i + dh = h \ ,$$

where the last inequality holds by the condition of the observation, and the last equality holds since the fact that $\sum_{i=1}^{k} c_i \mathbf{v}^{(i)} + d\mathbf{u}$ is a convex combination implies $\sum_{i=1}^{k} c_i + d = 1$. $\square$

The last properties that we need to prove for the basic instances are about the optimal values of these instances. Specifically, we need to show that both their optimal values are significant (at least $\Omega(k^{-1})$), but there is a large gap between them. The following two lemmata show these properties, respectively.

Lemma C.4. $\max_{\mathbf{x}\in\mathcal{P}_{h,k}} \hat{F}_k(\mathbf{x}) = \Omega(k^{-1})$ *and* $\max_{\mathbf{x}\in\mathcal{P}_{h,k}} \hat{G}_k(\mathbf{x}) = \Omega(k^{-1})$.

Proof. We prove the lemma by considering the vector $\mathbf{y} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{u}^{(i)}$. Since $\mathbf{y} \in \mathcal{P}_{h,k}$ and $\bar{\mathbf{y}} = \mathbf{y}$, $\hat{F}_k(\mathbf{y})$ lower bounds both $\max_{\mathbf{x}\in\mathcal{P}_{h,k}} \hat{F}_k(\mathbf{x})$ and $\max_{\mathbf{x}\in\mathcal{P}_{h,k}} \hat{G}_k(\mathbf{x})$. Thus, it remains to show that $\hat{F}_k(\mathbf{y}) = \Omega(k^{-1})$. By Lemma 5.2,

$$\hat{F}_k(\mathbf{y}) \geq F_k(\mathbf{y}) - \varepsilon' = \sum_{i=1}^{k} y_{a_i}(1-b_i) - \varepsilon' = \sum_{i=1}^{k} \frac{1}{k}\cdot\left(1-\left(1-\frac{1}{k}\right)\right) - \varepsilon' = \frac{1}{k} - \varepsilon' = \frac{1}{2k} \ ,$$

where F_k is the multilinear extension of f_k. □

Lemma C.5. $\max_{\mathbf{x}\in\mathcal{P}_{h,k}} \hat{F}_k(\mathbf{x}) \geq 1 - 1/(2k)$ *and* $\max_{\mathbf{x}\in\mathcal{P}_{h,k}} \hat{G}_k(\mathbf{x}) \leq (1-h)/4 + 3/(2k)$.

Proof. To prove the first part of the lemma, it suffices to observe that $\mathbf{v}^{(1)} \in \mathcal{P}_{h,k}$ and

$$\hat{F}_k(\mathbf{v}^{(1)}) \geq F_k(\mathbf{v}^{(1)}) - \varepsilon' = f_k(\{a_1\} \cup \{b_i \mid 2 \leq i \leq k\}) - \varepsilon' = 1 - 1/(2k) \ ,$$

where F_k is the multilinear extension of f_k.

Let us now prove the second part of the lemma. Fix an arbitrary vector $\mathbf{x} \in \mathcal{P}_{h,k}$, and let d be the coefficient of $\mathbf{u}$ in the convex combination that shows that $\mathbf{x}$ belongs to $\mathcal{P}_{h,k}$. Then,

$$\sum_{i=1}^{k} x_{a_i} = 1 - d \qquad \text{and} \qquad \sum_{i=1}^{k} x_{b_i} = dkh + (1-d)(k-1) = k(dh+1-d) + d - 1 \ .$$

Thus,

$$\begin{aligned}\hat{G}_k(\mathbf{x}) = \hat{F}_k(\bar{\mathbf{x}}) \leq F_k(\bar{\mathbf{x}}) + \varepsilon' &= \sum_{1=1}^{k} \frac{\sum_{i=1}^{k} x_{a_i}}{k}\left(1 - \frac{\sum_{i=1}^{k} x_{b_i}}{k}\right) + \varepsilon' \\ &= (1-d)\left(d - dh + \frac{1-d}{k}\right) + \varepsilon' \leq d(1-d)(1-h) + \frac{1}{k} + \varepsilon' \leq \frac{1-h}{4} + \frac{3}{2k} \ .\end{aligned}$$

□

We now would like to describe how the two basic instances are scrambled. Intuitively, the constraint polytope $\mathcal{K}_{h,k,\ell}$ of a scrambled instance is obtained by combining ℓ orthogonal instances of $\mathcal{P}_{h,k}$. Each element a_i or b_i has a copy in all the orthogonal instances, and the objective function treats every such copy as representing ℓ^{-1} of the original element. For example, if one would like to construct a solution assigning a value of $1/2$ to a_i, then the copies of a_i in $\mathcal{K}_{h,k,\ell}$ should get an average value of $1/2$. By randomly permuting the names of the elements in each orthogonal instance of $\mathcal{P}_{h,k}$, we make it difficult for the algorithm to construct solutions that do not correspond to symmetric vectors in $\mathcal{P}_{h,k}$. More formally, the constraint polytope $\mathcal{K}_{h,k,\ell}$ is a subset of $[0,1]^{\mathcal{M}_{k,\ell}}$, where

$$\mathcal{M}_{k,\ell} = \{a_{i,j}, b_{i,j} \mid i \in [k], j \in [\ell]\} \ .$$

A vector $\mathbf{x} \in [0,1]^{\mathcal{M}_{k,\ell}}$ belongs to $\mathcal{K}_{h,k,\ell}$ if for every $j \in [\ell]$ we have $\mathbf{x}^{(j)} \in \mathcal{P}_{h,k}$, where the vector $\mathbf{x}^{(j)} \in [0,1]^{\mathcal{N}_k}$ is defined by

$$\mathbf{x}^{(j)}_{a_i} = \mathbf{x}_{a_{i,j}} \qquad \text{and} \qquad \mathbf{x}^{(j)}_{b_i} = \mathbf{x}_{b_{i,j}} \ .$$

The following lemma is an immediate corollary of the definition of $\mathcal{K}_{h,k,\ell}$, Observation C.3 and the discussion before this observation.

Lemma C.6. *When* $k \geq 1/(1-h)$, $\mathcal{K}_{h,k,\ell}$ *is solvable and* $\max_{\mathbf{x}\in\mathcal{K}_{h,k,\ell}} \|\mathbf{x}\|_\infty = h$.

The objective functions of the scrambled instances are formally defined using a vector $\boldsymbol{\sigma}$ of ℓ permutations over $[k]$ (in other words, $\sigma_1, \sigma_2, \ldots, \sigma_\ell$ are all permutations over $[k]$). Given such a vector $\boldsymbol{\sigma}$ and a vector $\mathbf{x} \in [0,1]^{\mathcal{M}_{k,\ell}}$, we define the vector $\mathbf{x}^{(\boldsymbol{\sigma})} \in [0,1]^{\mathcal{N}_k}$ as follows.

$$\mathbf{x}^{(\boldsymbol{\sigma})}_{a_i} = \tfrac{1}{\ell} \sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}} \qquad \text{and} \qquad \mathbf{x}^{(\boldsymbol{\sigma})}_{b_i} = \tfrac{1}{\ell} \sum_{j=1}^{\ell} \mathbf{x}_{b_{\sigma_j(i),j}} \enspace .$$

Then, the functions $\bar{F}_{k,\boldsymbol{\sigma}} \colon [0,1]^{\mathcal{M}_{k,\ell}} \to \mathbb{R}_{\geq 0}$ and $\bar{G}_{k,\boldsymbol{\sigma}} \colon [0,1]^{\mathcal{M}_{k,\ell}} \to \mathbb{R}_{\geq 0}$ are defined for every vector $\mathbf{x} \in [0,1]^{\mathcal{M}_{k,\ell}}$ by

$$\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) = \hat{F}(\mathbf{x}^{(\boldsymbol{\sigma})}) \qquad \text{and} \qquad \bar{G}_{k,\boldsymbol{\sigma}}(\mathbf{x}) = \hat{G}(\mathbf{x}^{(\boldsymbol{\sigma})}) \enspace .$$

The following lemma shows that the functions $\bar{F}_{k,\boldsymbol{\sigma}}$ and $\bar{G}_{k,\boldsymbol{\sigma}}$ inherit all the good properties of $\hat{F}_k$ and $\hat{G}_k$ promised by Lemma C.1. Since the proof of this lemma is technical and quite straightforward given Lemma C.1, we defer it to Section C.1.

Lemma C.7. *The functions $\bar{F}_{k,\boldsymbol{\sigma}}$ and $\bar{G}_{k,\boldsymbol{\sigma}}$ are continuously differentiable, non-negative and DR-submodular. Furthermore, they are β-smooth for a value β that is polynomial in k and ℓ.*

We can now formally state the scrambled instances that we consider.

$$\begin{array}{l} \max \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) \\ \mathbf{x} \in \mathcal{K}_{h,k,\ell} \end{array} \qquad \text{and} \qquad \begin{array}{l} \max \bar{G}_{k,\boldsymbol{\sigma}}(\mathbf{x}) \\ \mathbf{x} \in \mathcal{K}_{h,k,\ell} \end{array} \enspace .$$

The next lemma shows that these scrambled instances inherit the values of their optimal solutions from the basic instances, which in particular, implies that they also inherit the gap between these solutions.

Lemma C.8. *We have both* $\max_{\mathbf{x} \in \mathcal{K}_{h,k,\ell}} \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) = \max_{\mathbf{x} \in \mathcal{P}_{h,k}} \hat{F}_k(\mathbf{x})$ *and* $\max_{\mathbf{x} \in \mathcal{K}_{h,k,\ell}} \bar{G}_{k,\boldsymbol{\sigma}}(\mathbf{x}) = \max_{\mathbf{x} \in \mathcal{P}_{h,k}} \hat{G}_k(\mathbf{x})$.

Proof. We prove below only the first equality of the lemma. The proof of the other equality is analogous. We begin by arguing that $\max_{\mathbf{x} \in \mathcal{K}_{h,k,\ell}} \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) \geq \max_{\mathbf{x} \in \mathcal{P}_{h,k}} \hat{F}_k(\mathbf{x})$. To show this inequality, we start with an arbitrary vector $\mathbf{x} \in \mathcal{P}_{h,k}$, and we construct a vector $\mathbf{y} \in \mathcal{K}_{h,k,\ell}$ such that $\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{y}) = \hat{F}_k(\mathbf{x})$. Formally, the vector $\mathbf{y}$ is defined as follows. For every $i \in [k]$ and $j \in [\ell]$,

$$\mathbf{y}_{a_{i,j}} = \mathbf{x}_{a_{\sigma_j^{-1}(i)}} \qquad \text{and} \qquad \mathbf{y}_{b_{i,j}} = \mathbf{x}_{b_{\sigma_j^{-1}(i)}} \enspace .$$

One can observe that $\mathbf{x} = \mathbf{y}^{(\boldsymbol{\sigma})}$, and therefore, we indeed have $\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{y}) = \hat{F}_k(\mathbf{x})$; which means that we are only left to show that $\mathbf{y} \in \mathcal{K}_{h,k,\ell}$. Recall that, by the definition of $\mathcal{K}_{h,k,\ell}$, to prove this inclusion, we need to argue that $\mathbf{y}^{(j)} \in \mathcal{P}_{h,k}$ for every $j \in [\ell]$, where $\mathbf{y}^{(j)}$ is the restriction of $\mathbf{y}$ to elements of $\{a_{i,j}, b_{i,j} \mid i \in [k]\}$.

Below, given a vector $\mathbf{z} \in \mathcal{P}_{h,k}$, we denote by $\sigma_j(\mathbf{z})$ the following vector.

$$(\sigma_j(\mathbf{z}))_{a_i} = \mathbf{z}_{a_{\sigma_j^{-1}(i)}} \qquad \text{and} \qquad (\sigma_j(\mathbf{z}))_{b_i} = \mathbf{z}_{b_{\sigma_j^{-1}(i)}} \enspace .$$

Observe that this definition implies $\sigma_j(\mathbf{u}) = \mathbf{u}$ and $\sigma_j(\mathbf{v}^{(i)}) = \mathbf{v}^{(\sigma_j(i))}$, where $\mathbf{u}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}$ are the vectors whose convex-hall defines $\mathcal{P}_{h,k}$. Since $\mathbf{x} \in \mathcal{P}_{h,k}$, it must be given by some convex combination of the vectors $\mathbf{u}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}$. In other words,

$$\mathbf{x} = \sum_{i=1}^{k} c_i \cdot \mathbf{v}^{(i)} + d \cdot \mathbf{u} \enspace .$$

Thus,

$$\mathbf{y}^{(j)} = \sigma_j(\mathbf{x}) = \sigma_j\left(\sum_{i=1}^{k} c_i \cdot \mathbf{v}^{(i)} + d \cdot \mathbf{u}\right) = \sum_{i=1}^{k} c_i \cdot \mathbf{v}^{(\sigma_j(i))} + d \cdot \mathbf{u} \enspace .$$

The rightmost side of the last equality is another convex combination of the vectors $\mathbf{u}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}$, and thus, the equality shows that $\mathbf{y}^{(j)} \in \mathcal{P}_{h,k}$, as desired.

We now get to the proof that $\max_{\mathbf{x} \in \mathcal{K}_{h,k,\ell}} \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) \leq \max_{\mathbf{x} \in \mathcal{P}_{h,k}} \hat{F}_k(\mathbf{x})$. Consider an arbitrary vector $\mathbf{x} \in \mathcal{K}_{h,k,\ell}$. By the definition of $\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x})$, $\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) = \hat{F}_k(\mathbf{x}^{(\boldsymbol{\sigma})})$. Thus, to prove the last inequality, it suffices to show that $\mathbf{x}^{(\boldsymbol{\sigma})} \in \mathcal{P}_{h,k}$, which is done by the next claim. Since the proof of this claim is very similar to the above proof that $\mathbf{y} \in \mathcal{K}_{h,k,\ell}$, we defer it to Section C.1.

Claim C.9. *For every vector* $\mathbf{x} \in \mathcal{K}_{h,k,\ell}$, $\mathbf{x}^{(\boldsymbol{\sigma})} \in \mathcal{P}_{h,k}$. □

Corollary C.10. *It holds that* $\max_{\mathbf{x}\in\mathcal{K}_{h,k,\ell}} \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) \geq 1 - 1/(2k) = \Omega(k^{-1})$ *and* $(1-h)/4 + 3/(2k) \geq \max_{\mathbf{x}\in\mathcal{K}_{h,k,\ell}} \bar{G}_{k,\boldsymbol{\sigma}}(\mathbf{x}) = \Omega(k^{-1})$.

Lemmata C.6, C.7 and C.8 show that the scrambled instances we have constructed have all the properties stated in Theorem 5.1 when $k \geq 1/(1-h)$). Therefore, to prove the theorem it suffices to show that no sub-exponential time algorithm can obtain a good approximation guarantee given these instances when ℓ is large enough compared to k. We do this by showing that when $\boldsymbol{\sigma}$ is chosen uniformly at random, it is difficult to distinguish between the two scrambled instances, and therefore, no sub-exponential time algorithm can obtain an approximation ratio better than the (large) gap between their optimal values. The first step in this proof is done by the next lemma, which shows that any single access to the objective function almost always returns the same answer given either of the two scrambled instances. To understand why the lemma implies this, it is important to recall that we assume that the algorithm is able to access F only by making queries whose outputs are determined by the values of F in an arbitrary small neighborhood of a given point $\mathbf{x}$ (this kind of queries includes the standard value and gradient queries).

Lemma C.11. *Assume* $\boldsymbol{\sigma}$ *is drawn uniformly at random, i.e.,* σ_j *is an independently chosen uniformly random permutation of* $[k]$ *for every* $j \in [\ell]$. *Given any vector* $\mathbf{x} \in [0,1]^{\mathcal{M}_k}$, *with probability at least* $1 - 4k \cdot e^{-\ell \cdot \frac{\delta_k}{6\sqrt{2k}}}$ *we have* $\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{y}) = \bar{G}_{k,\boldsymbol{\sigma}}(\mathbf{y})$ *for every vector* $\mathbf{y}$ *such that* $\|\mathbf{x}-\mathbf{y}\|_2 \leq (\sqrt{\ell}/4)\cdot\delta_k$, *where* δ_k *is the value of* δ *when Lemma 5.2 is applied to* f_k.

Proof. Below, we show that $\|\mathbf{x}^{(\boldsymbol{\sigma})} - \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}\|_2 \leq \delta_k/2$ with probability at least $1 - 4k \cdot e^{-\ell\delta_k/(6\sqrt{2k})}$. However, before getting to this proof, let us show that, whenever this inequality holds, we also have $\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{y}) = \bar{G}_{k,\boldsymbol{\sigma}}(\mathbf{y})$. By the definitions of $\bar{F}_{k,\boldsymbol{\sigma}}$ and $\bar{G}_{k,\boldsymbol{\sigma}}$, the last equality is equivalent to $\hat{F}_k(\mathbf{y}^{(\boldsymbol{\sigma})}) = \hat{G}_k(\mathbf{y}^{(\boldsymbol{\sigma})})$, and this equality holds by Lemma 5.2 since

$$\|\mathbf{y}^{(\boldsymbol{\sigma})} - \bar{\mathbf{y}}^{(\boldsymbol{\sigma})}\|_2 \leq \|\mathbf{y}^{(\boldsymbol{\sigma})} - \mathbf{x}^{(\boldsymbol{\sigma})}\|_2 + \|\bar{\mathbf{y}}^{(\boldsymbol{\sigma})} - \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}\|_2 + \|\mathbf{x}^{(\boldsymbol{\sigma})} - \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}\|_2 \leq 2\|\mathbf{y}^{(\boldsymbol{\sigma})} - \mathbf{x}^{(\boldsymbol{\sigma})}\|_2 + \delta_k/2 \leq \delta_k \ ,$$

where the first inequality is the triangle inequality, the second inequality holds since averaging two vectors in the same way can only decrease their distance from each other, and the last inequality holds because Sedrakyan's inequality (or Cauchy-Schwarz inequality) implies

$$\begin{aligned}\|\mathbf{y}^{(\boldsymbol{\sigma})} - \mathbf{x}^{(\boldsymbol{\sigma})}\|_2^2 &= \frac{\sum_{i=1}^{k}[\sum_{j=1}^{\ell}(\mathbf{y}_{a_{\sigma_j(i),j}} - \mathbf{x}_{a_{\sigma_j(i),j}})]^2 + \sum_{i=1}^{k}[\sum_{j=1}^{\ell}(\mathbf{y}_{b_{\sigma_j(i),j}} - \mathbf{x}_{b_{\sigma_j(i),j}})]^2}{\ell^2} \\ &\leq \frac{\sum_{i=1}^{k}\sum_{j=1}^{\ell}(\mathbf{y}_{a_{\sigma_j(i),j}} - \mathbf{x}_{a_{\sigma_j(i),j}})^2 + \sum_{i=1}^{k}\sum_{j=1}^{\ell}(\mathbf{y}_{b_{\sigma_j(i),j}} - \mathbf{x}_{b_{\sigma_j(i),j}})^2}{\ell} = \frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{\ell} \ .\end{aligned}$$

It now remains to prove that the inequality $\|\mathbf{x}^{(\boldsymbol{\sigma})} - \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}\|_2 \leq \delta_k/2$ holds with probability at least $1 - 4k\cdot e^{-\ell\delta_k/(6\sqrt{2k})}$. By the union bound, to prove this inequality it suffices to show that, for every $i \in [k]$, the probabilities of the two inequalities $|\mathbf{x}^{(\boldsymbol{\sigma})}_{a_i} - \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}_{a_i}| > \delta_k/\sqrt{8k}$ and $|\mathbf{x}^{(\boldsymbol{\sigma})}_{b_i} - \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}_{b_i}| > \delta_k/\sqrt{8k}$ to hold are both at most $2e^{-\ell\delta_k/(6\sqrt{2k})}$. The rest of this proof is devoted to showing that this is indeed the case for the first inequality as the proof for the second inequality is analogous. Recall that

$$\mathbf{x}^{(\boldsymbol{\sigma})}_{a_i} = \tfrac{1}{\ell}\sum_{j=1}^{\ell}\mathbf{x}_{a_{\sigma_j(i),j}} \ . \tag{3}$$

Thus,

$$\bar{\mathbf{x}}^{(\boldsymbol{\sigma})}_{a_i} = \frac{1}{k}\sum_{i'=1}^{k}\mathbf{x}^{(\boldsymbol{\sigma})}_{a_{i'}} = \frac{1}{k}\sum_{i'=1}^{k}\left(\tfrac{1}{\ell}\sum_{j=1}^{\ell}\mathbf{x}_{a_{\sigma_j(i'),j}}\right) = \frac{1}{k\ell}\sum_{i'=1}^{k}\sum_{j=1}^{\ell}\mathbf{x}_{a_{\sigma_j(i'),j}} = \frac{1}{k\ell}\sum_{i'=1}^{k}\sum_{j=1}^{\ell}\mathbf{x}_{a_{i',j}} \ , \tag{4}$$

where the last equality holds since σ_j is a permutation over $[k]$. Similarly, we also have

$$\mathbb{E}[\mathbf{x}^{(\boldsymbol{\sigma})}_{a_i}] = \frac{1}{\ell}\sum_{j=1}^{\ell}\mathbb{E}[\mathbf{x}_{a_{\sigma_j(i),j}}] = \frac{1}{\ell}\sum_{j=1}^{\ell}\left(\tfrac{1}{k}\sum_{i'=1}^{k}\mathbb{E}[\mathbf{x}_{a_{i',j}}]\right) = \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}_{a_i} \ .$$

Hence, the claim that we want to prove bounds the probability that $\mathbf{x}^{(\boldsymbol{\sigma})}_{a_i}$ significantly deviates from its expectation. Furthermore, Equation (3) shows that $\ell \cdot \mathbf{x}^{(\boldsymbol{\sigma})}_{a_i}$ is the sum of ℓ random variables taking values from the range $[0,1]$. Since σ_j

is chosen independently for every $j \in [\ell]$, these ℓ random variables are independent, which allows us to use Chernoff's inequality to bound their sum. Therefore,

$$\begin{aligned}\Pr\left[|\mathbf{x}^{(\boldsymbol{\sigma})}_{a_i} - \bar{\mathbf{x}}^{(\boldsymbol{\sigma})}_{a_i}| > \frac{\delta_k}{\sqrt{8k}}\right] &= \Pr\left[\left|\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}} - \mathbb{E}\left[\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}}\right]\right| > \frac{\ell\delta_k}{\sqrt{8k}}\right] \\ &\leq 2e^{-\frac{\mathbb{E}[\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}}] \cdot \min\left\{\frac{\ell\delta_k}{\sqrt{8k}\cdot\mathbb{E}[\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}}]}, \frac{\ell^2\delta_k^2}{8k\cdot\mathbb{E}[\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}}]^2}\right\}}{3}} \\ &= 2e^{-\frac{\min\left\{\frac{\ell\delta_k}{\sqrt{8k}}, \frac{\ell^2\delta_k^2}{8k\cdot\mathbb{E}[\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}}]}\right\}}{3}} \leq 2e^{-\frac{\frac{\ell\delta_k}{\sqrt{8k}}\cdot\min\left\{1, \frac{\delta_k}{\sqrt{8k}}\right\}}{3}} = 2e^{-\ell\cdot\frac{\delta_k}{6\sqrt{2k}}} \ .\end{aligned}$$ □

Equation (4) in the last proof has another interesting consequence. This equation shows that $\bar{\mathbf{x}}^{(\boldsymbol{\sigma})}$ is independent of $\boldsymbol{\sigma}$. Since Lemma 5.2 shows that $\hat{G}_k(\mathbf{x}) = \hat{F}_k(\bar{\mathbf{x}})$ for every $\mathbf{x} \in [0,1]^{\mathcal{N}_k}$, this implies the following observation.

Observation C.12. *For every $\mathbf{x} \in [0,1]^{\mathcal{M}_{k,\ell}}$, the value of $\bar{G}_{k,\boldsymbol{\sigma}}(\mathbf{x}) = \hat{G}_k(\mathbf{x}^{(\boldsymbol{\sigma})}) = \hat{F}_k(\bar{\mathbf{x}}^{\boldsymbol{\sigma}})$ is independent of $\boldsymbol{\sigma}$.*

In light of the above observation, we use below $\bar{G}_k$ to denote the function $\bar{G}_{k,\boldsymbol{\sigma}}$. We are now ready to prove Theorem 5.1.

Proof of Theorem 5.1. Fix an arbitrary sub-exponential function $P(\cdot)$. Below, we show that there is a distribution of instances on which no deterministic algorithm making at most $P(n)$ accesses to the objective function, where n is the dimension, can obtain an approximation ratio of $(1-h)/4+\varepsilon$. By Yao's principle, this will imply the same result also for randomized algorithms running in time $P(n)$ (notice that running in time $P(n)$ implies making at most $P(n)$ accesses to the objective function).

The distribution of instances we consider is the scrambled instance $\max_{vx\in\mathcal{K}_{h,k,\ell}} \mathcal{F}_{k,\boldsymbol{\sigma}}$, where $k \geq 1/(1-h)$ and ℓ are deterministic values to be determined below, and $\boldsymbol{\sigma}$ is chosen at random according to the distribution defined in Lemma C.11. Assume towards a contradiction that there exists a deterministic algorithm ALG that accesses the objective function at most $P(|\mathcal{M}_{k,\ell}|) = P(2k\ell)$ times, and given a random instance from the above distribution obtains an approximation ratio of $(1-h)/4+\varepsilon$. More formally, if we denote $OPT = \max_{\mathbf{x}\in\mathcal{P}_{h,k}} \hat{F}_k(\mathbf{x})$, then ALG guarantees that its output vector $\mathbf{a}$ obeys

$$\mathbb{E}[\mathcal{F}_{k,\boldsymbol{\sigma}}(\mathbf{a})] \geq [(1-h)/4+\varepsilon] \cdot \mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{K}_{h,k,\ell}} \hat{F}_{k,\boldsymbol{\sigma}}(\mathbf{x})\right] = [(1-h)/4+\varepsilon] \cdot OPT \ , \tag{5}$$

where the equality holds by Lemma C.8.

Consider now an execution of ALG on the instance $\max_{\mathbf{x}\in\mathcal{K}_{h,k,\ell}} \bar{G}_k(\mathbf{x})$, and let us denote by $A_1, A_2, \ldots, A_r$ the accesses made by ALG (each access A_i consists of a vector $\mathbf{x}$ and the type of access, namely whether ALG evaluates the objective function at $\mathbf{x}$ or calculates the gradient of the objective function at $\mathbf{x}$). It is convenient to assume that the last access made by ALG is to evaluate the value of its output set $\mathbf{a}$. If this is not the case, we can add such an access to the end of the execution of ALG, and still have $r \leq P(2k\ell)+1$. Let $\mathcal{E}$ be the event that all the accesses $A_1, A_2, \ldots, A_r$ return the same value given that the objective is either $\bar{G}_k$ or $\bar{F}_{k,\boldsymbol{\sigma}}$. Clearly, ALG follows the same execution path given either $\bar{G}_k$ or $\bar{F}_{k,\boldsymbol{\sigma}}$ when the event $\mathcal{E}$ happens, and therefore, it outputs the same vector $\mathbf{a} \in \mathcal{K}_{h,k,\ell}$ in this case. Furthermore, $\mathcal{E}$ also implies that $\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{a}) = \bar{G}_k(\mathbf{a})$, and thus, conditioned on $\mathcal{E}$,

$$\begin{aligned}\mathcal{F}_{k,\boldsymbol{\sigma}}(\mathbf{a}) &\leq \max_{\mathbf{x}\in\mathcal{K}_{h,k,\ell}} \hat{G}_k(\mathbf{x}) \leq (1-h)/4 + 3/(2k) \leq \frac{(1-h)/4+3/(2k)}{1-1/(2k)} \cdot OPT \\ &\leq \left[\frac{1-h}{4-2/k} + \frac{3}{k}\right] \cdot OPT \leq \left[\frac{1-h}{4} + \frac{4}{k}\right] \cdot OPT \ ,\end{aligned}$$

where the second inequality holds by Corollary C.10, the third inequality follows from Lemma C.5, and two last inequalities hold since $k \geq 1$ and $h \in [0,1]$.

We would like to use the last inequality to upper bound $\mathbb{E}[\mathcal{F}_{k,\boldsymbol{\sigma}}(\mathbf{a})]$. For that purpose, we need to lower bound the probability of the event $\mathcal{E}$. By Lemma C.11 and the union bound,

$$\Pr[\mathcal{E}] \geq 1 - 4kr \cdot e^{-\ell\cdot\frac{\delta_k}{6\sqrt{2k}}} \geq 1 - 4k[P(2k\ell)+1] \cdot e^{-\ell\cdot\frac{\delta_k}{6\sqrt{2k}}} \ .$$

Consider the second term in the rightmost side of the last inequality. This term is a function of k and ℓ alone, and for a fixed value of k it is the product of a sub-exponential function of ℓ and an exponentially decreasing function of ℓ. Therefore, for any fixed value of k, we can choose a large enough value for ℓ to guarantee that $2k[P(2k\ell)+1]\cdot e^{-\ell\cdot\frac{\delta_k}{6\sqrt{k}}} \leq \varepsilon/2$. In the rest of the proof we assume that ℓ is chosen in such a way. Then, since we always have $\mathcal{F}_{k,\boldsymbol{\sigma}}(\mathbf{a}) \leq OPT$ and $\Pr[\mathcal{E}] \leq 1$, we get by the law of total expectation,

$$\mathbb{E}[\mathcal{F}_{k,\boldsymbol{\sigma}}(\mathbf{a})] \leq \Pr[\bar{\mathcal{E}}]\cdot OPT + \mathbb{E}[\mathcal{F}_{k,\boldsymbol{\sigma}}(\mathbf{a}) \mid \mathcal{E}] \leq \frac{\varepsilon}{2}\cdot OPT + [(1-h)/4 + 4/k]\cdot OPT \ ,$$

which contradicts Equation (5) (and thus, the existence of ALG) when k is chosen to be $\max\{\lceil 1/(h-1)\rceil, 8/\varepsilon\}$. □

C.1 Missing Proofs

C.1.1 Proof of Claim C.2

In this section we prove Claim C.2, which we repeat here for convenience.

Claim C.2. *The absolute values of the second order partial derivatives of the functions $\hat{F}_k$ and $\hat{G}_k$ are bounded by $16k+2$ almost everywhere, and therefore, both functions are β-smooth for a β value that is polynomial in k.*

Proof. Recall that $\hat{F}_k$ and $\hat{G}_k$ are the functions $\hat{F}$ and $\hat{G}$ whose existence is guaranteed by Lemma 5.2 for $f = f_k$. The functions $\hat{F}$ and $\hat{G}$ are obtained in the proof of Lemma 5.2 in a series of steps involving multiple intermediate functions. The first of these functions are F (the multilinear extension of f), the function $G(\mathbf{x}) = F(\bar{\mathbf{x}})$ and the function $H(\mathbf{x}) = F(\mathbf{x}) - G(\mathbf{x})$. The proof of Lemma 3.5 of Vondrák (2013) shows that the absolute values of the second partial derivatives of these functions are bounded by $4M$, $4M$ and $8M$, respectively, where M is the maximum value that the function f can take. Since in our case f is f_k, the maximum value it can take is k, and therefore, the absolute values of the second partial derivatives of all three functions can be upper bounded by $8k$.

The next function we consider is a function denoted by $\tilde{F}$ in the proof of Lemma 5.2. The proof of Lemma 3.8 of Vondrák (2013) shows that for every two elements $u, v \in \mathcal{N}$, this function obeys almost everywhere the inequality

$$\left|\frac{\partial^2\tilde{F}(\mathbf{x})}{\partial u\partial v} - \frac{\partial^2 F(\mathbf{x})}{\partial u\partial v} + \phi(D(\mathbf{x}))\cdot\frac{\partial^2 H(\mathbf{x})}{\partial u\partial v}\right| \leq 512M|\mathcal{N}|\alpha = \frac{512\varepsilon'}{2000|\mathcal{N}|^2} \leq 1 \ ,$$

where ϕ is a function defined by Vondrák (2013) whose range is $[0,1]$, $D(\mathbf{x})$ is another function defined by Vondrák (2013) and $\alpha = \varepsilon'/(2000M|\mathcal{N}|^3)$. Since $|\phi(D(\mathbf{x}))| \leq 1$, the last inequality implies that the absolute values of the second partial derivatives of $\tilde{F}$ are upper bounded by $16k+1$ because the second partial derivatives of F and H have absolute values bounded by $8k$.

The functions $\hat{F}$ and $\hat{G}$ are obtained from $\tilde{F}$ and G, respectively, by adding $256M|\mathcal{N}|\alpha J(\mathbf{x}) = \frac{256\varepsilon'}{2000|\mathcal{N}|^2}\cdot J(\mathbf{x})$, where

$$J(\mathbf{x}) = |\mathcal{N}|^2 + 3|\mathcal{N}|\|\mathbf{x}\|_1 - (\|\mathbf{x}\|_1)^2 \ .$$

Since the second order partial derivatives of $J(\mathbf{x})$ are all -2, and the coefficient of $J(\mathbf{x})$ is $\frac{256\varepsilon'}{2000|\mathcal{N}|^2} \leq 1/2$, adding $\frac{256\varepsilon'}{2000|\mathcal{N}|^2}\cdot J(\mathbf{x})$ cannot increase the absolute value of the second order partial derivatives by more than 1. □

C.1.2 Proof of Lemma C.7

In this section we prove Lemma C.7, which we repeat here for convenience.

Lemma C.7. *The functions $\bar{F}_{k,\boldsymbol{\sigma}}$ and $\bar{G}_{k,\boldsymbol{\sigma}}$ are continuously differentiable, non-negative and DR-submodular. Furthermore, they are β-smooth for a value β that is polynomial in k and ℓ.*

Proof. We prove the lemma below for $\bar{F}_{k,\boldsymbol{\sigma}}$. The proof for $\bar{G}_{k,\boldsymbol{\sigma}}$ is analogous. The non-negativity of $\bar{F}_{k,\boldsymbol{\sigma}}$ follows immediately from their definitions and the non-negativity of $\hat{F}_k$ and $\hat{G}_k$. Furthermore, by the chain-rule, for every pair of $i \in [k]$ and $j \in [\ell]$, we have

$$\frac{\partial\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x})}{\partial\mathbf{x}_{a_{i,j}}} = \frac{1}{\ell}\cdot\left.\frac{\partial\hat{F}_k(\mathbf{z})}{\partial\mathbf{z}_{a_{\sigma_j(i)}}}\right|_{\mathbf{z}=\mathbf{x}(\boldsymbol{\sigma})} \quad \text{and} \quad \frac{\partial\bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x})}{\partial\mathbf{x}_{b_{i,j}}} = \frac{1}{\ell}\cdot\left.\frac{\partial\hat{F}_k(\mathbf{z})}{\partial\mathbf{z}_{b_{\sigma_j(i)}}}\right|_{\mathbf{z}=\mathbf{x}(\boldsymbol{\sigma})} . \tag{6}$$

Thus, the continuous differentiability of $\hat{F}_k$ implies that $\bar{F}_{k,\boldsymbol{\sigma}}$ is also continuously differentiable.

Taking the derivative of the last equalities with respect to $a_{i',b'}$ for another pair $i' \in [k], j' \in [\ell]$, the chain-rule gives us the equalities

$$\frac{\partial^2 \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x})}{\partial \mathbf{x}_{a_{i',j'}} \partial \mathbf{x}_{a_{i,j}}} = \frac{1}{\ell^2} \cdot \left. \frac{\partial^2 \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{a_{\sigma_{j'}(i')}} \partial \mathbf{z}_{a_{\sigma_j(i)}}} \right|_{\mathbf{z}=\mathbf{x}^{(\boldsymbol{\sigma})}}$$

and

$$\frac{\partial^2 \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x})}{\partial \mathbf{x}_{a_{i',j'}} \partial \mathbf{x}_{b_{i,j}}} = \frac{1}{\ell^2} \cdot \left. \frac{\partial^2 \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{a_{\sigma_{j'}(i')}} \partial \mathbf{z}_{b_{\sigma_j(i)}}} \right|_{\mathbf{z}=\mathbf{x}^{(\boldsymbol{\sigma})}} .$$

Since similar equalities hold also when we take the derivative of the equalities in Equation (6) with respect to $b_{i',j'}$, the DR-submodularity of $\hat{F}_k$ implies the same property for $\bar{F}_{k,\boldsymbol{\sigma}}$.

It remains to bound the smoothness of $\bar{F}_{k,\boldsymbol{\sigma}}$. For every two vectors $\mathbf{x}, \mathbf{y} \in [0,1]^{\mathcal{M}_k}$, we have by Equation (6) that

$$\begin{aligned}
&\|\nabla \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) - \nabla \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{y})\|_2^2 = \sum_{i=1}^{k} \sum_{j=1}^{\ell} \left(\frac{1}{\ell} \cdot \left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{a_{\sigma_j(i)}}} \right|_{\mathbf{z}=\mathbf{x}^{(\boldsymbol{\sigma})}} - \frac{1}{\ell} \cdot \left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{a_{\sigma_j(i)}}} \right|_{\mathbf{z}=\mathbf{y}^{(\boldsymbol{\sigma})}} \right)^2 \\
&+ \sum_{i=1}^{k} \sum_{j=1}^{\ell} \left(\frac{1}{\ell} \cdot \left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{b_{\sigma_j(i)}}} \right|_{\mathbf{z}=\mathbf{x}^{(\boldsymbol{\sigma})}} - \frac{1}{\ell} \cdot \left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{b_{\sigma_j(i)}}} \right|_{\mathbf{z}=\mathbf{y}^{(\boldsymbol{\sigma})}} \right)^2 = \frac{1}{\ell} \cdot \sum_{i=1}^{k} \left(\left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{a_i}} \right|_{\mathbf{z}=\mathbf{x}^{(\boldsymbol{\sigma})}} - \left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{a_i}} \right|_{\mathbf{z}=\mathbf{y}^{(\boldsymbol{\sigma})}} \right)^2 \\
&+ \frac{1}{\ell} \cdot \sum_{i=1}^{k} \left(\left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{b_i}} \right|_{\mathbf{z}=\mathbf{x}^{(\boldsymbol{\sigma})}} - \left. \frac{\partial \hat{F}_k(\mathbf{z})}{\partial \mathbf{z}_{b_i}} \right|_{\mathbf{z}=\mathbf{y}^{(\boldsymbol{\sigma})}} \right)^2 = \frac{\|\nabla \hat{F}_k(\mathbf{x}^{(\boldsymbol{\sigma})}) - \nabla \hat{F}_k(\mathbf{y}^{(\boldsymbol{\sigma})})\|_2^2}{\ell} \leq \frac{\beta^2 \|\mathbf{x}^{(\boldsymbol{\sigma})} - \mathbf{y}^{(\boldsymbol{\sigma})}\|_2^2}{\ell} \\
&= \frac{\beta^2 \cdot \sum_{i=1}^{k} [(\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}} - \sum_{j=1}^{\ell} \mathbf{y}_{a_{\sigma_j(i),j}})^2 + (\sum_{j=1}^{\ell} \mathbf{x}_{b_{\sigma_j(i),j}} - \sum_{j=1}^{\ell} \mathbf{y}_{b_{\sigma_j(i),j}})^2]}{\ell^3} ,
\end{aligned}$$

where β is the smoothness parameter of $\hat{F}_k$, and the second equality holds since the entries of $\boldsymbol{\sigma}$ are permutations. Using Sedrakyan's inequality (or Cauchy–Schwarz inequality), we also have, for every $i \in [k]$,

$$\left(\sum_{j=1}^{\ell} \mathbf{x}_{a_{\sigma_j(i),j}} - \sum_{j=1}^{\ell} \mathbf{y}_{a_{\sigma_j(i),j}} \right)^2 \leq \ell \cdot \sum_{j=1}^{\ell} (\mathbf{x}_{a_{\sigma_j(i),j}} - \sum_{j=1}^{\ell} \mathbf{y}_{a_{\sigma_j(i),j}})^2$$

and

$$\left(\sum_{j=1}^{\ell} \mathbf{x}_{b_{\sigma_j(i),j}} - \sum_{j=1}^{\ell} \mathbf{y}_{b_{\sigma_j(i),j}} \right)^2 \leq \ell \cdot \sum_{j=1}^{\ell} (\mathbf{x}_{b_{\sigma_j(i),j}} - \sum_{j=1}^{\ell} \mathbf{y}_{b_{\sigma_j(i),j}})^2 .$$

Combining all the above inequalities yields

$$\begin{aligned}
\|\nabla \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{x}) - \nabla \bar{F}_{k,\boldsymbol{\sigma}}(\mathbf{y})\|_2 &\leq \frac{\beta \cdot \sqrt{\sum_{i=1}^{k} [\sum_{j=1}^{\ell} (\mathbf{x}_{a_{\sigma_j(i),j}} - \mathbf{y}_{a_{\sigma_j(i),j}})^2 + \sum_{j=1}^{\ell} (\mathbf{x}_{b_{\sigma_j(i),j}} - \mathbf{y}_{b_{\sigma_j(i),j}})^2]}}{\ell} \\
&= \frac{\beta \cdot \|\mathbf{x} - \mathbf{y}\|_2}{\ell} ,
\end{aligned}$$

which completes the proof of the lemma since the smoothness parameter β of $\hat{F}_k$ is polynomial in k. □

C.1.3 Proof of Claim C.9

In this section we prove Claim C.9, which we repeat here for convenience.

Claim C.9. *For every vector* $\mathbf{x} \in \mathcal{K}_{h,k,\ell}$, $\mathbf{x}^{(\boldsymbol{\sigma})} \in \mathcal{P}_{h,k}$.

Proof. By the definition of $\mathcal{K}_{h,k,\ell}$, the membership of $\mathbf{x}$ in $\mathcal{K}_{h,k,\ell}$ implies that for every $j \in [\ell]$ we must have $\mathbf{x}^{(j)} \in \mathcal{P}_{h,k}$. Thus, $\mathbf{x}^{(j)}$ can be represented by a convex combination of the vectors $\mathbf{u}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(k)}$ as follows.

$$\mathbf{x}^{(j)} = \sum_{i=1}^{k} c_{i,j} \cdot \mathbf{v}^{(j)} + d_j \cdot \mathbf{u} \ .$$

Similarly to the proof of Lemma C.8, let us define $\sigma_j^{-1}(\mathbf{x}^{(j)})$ to be the following vector. For every $i \in [k]$,

$$(\sigma_j^{-1}(\mathbf{x}^{(j)}))_{a_i} = \mathbf{x}^{(j)}_{a_{\sigma(i)}} \qquad \text{and} \qquad (\sigma_j^{-1}(\mathbf{x}^{(j)}))_{b_i} = \mathbf{x}^{(j)}_{b_{\sigma(i)}} \ .$$

Using the above notation, we get

$$\begin{aligned}
\mathbf{x}^{(\boldsymbol{\sigma})} &= \tfrac{1}{\ell} \sum_{j=1}^{\ell} \sigma_j^{-1}(\mathbf{x}^{(j)}) = \tfrac{1}{\ell} \sum_{j=1}^{\ell} \sigma_j^{-1} \left(\sum_{i=1}^{k} c_{i,j} \cdot \mathbf{v}^{(i)} + d_j \cdot \mathbf{u} \right) \\
&= \tfrac{1}{\ell} \sum_{j=1}^{\ell} \left[\sum_{i=1}^{k} c_{i,j} \cdot \sigma_j^{-1}(\mathbf{v}^{(i)}) + d_j \cdot \sigma_j^{-1}(\mathbf{u}) \right] = \tfrac{1}{\ell} \sum_{j=1}^{\ell} \left[\sum_{i=1}^{k} c_{i,j} \cdot \mathbf{v}^{(\sigma_j^{-1}(i))} + d_j \cdot \mathbf{u} \right] \\
&= \sum_{i=1}^{k} \frac{\sum_{j=1}^{\ell} c_{\sigma_j(i),j}}{\ell} \cdot \mathbf{v}^{(i)} + \frac{\sum_{j=1}^{\ell} d_j}{\ell} \cdot \mathbf{u} \ .
\end{aligned}$$

The last step in the proof of the claim is to show that the rightmost side is a convex combination, which implies $\mathbf{x}^{(\boldsymbol{\sigma})} \in \mathcal{P}_{h,k}$ by the definition of $\mathcal{P}_{h,k}$. To see that this is indeed the case, we observe that the coefficients of all the vectors in this rightmost side are averages of non-negative numbers, and therefore, are non-negative as well. Furthermore,

$$\sum_{i=1}^{k} \frac{\sum_{j=1}^{\ell} c_{\sigma_j(i),j}}{\ell} + \frac{\sum_{j=1}^{\ell} d_j}{\ell} = \tfrac{1}{\ell} \sum_{j=1}^{\ell} \left[\sum_{i=1}^{k} c_{\sigma_j(i),j} + d_j \right] = \tfrac{1}{\ell} \sum_{j=1}^{\ell} \left[\sum_{i=1}^{k} c_{i,j} + d_j \right] = \tfrac{1}{\ell} \sum_{j=1}^{\ell} 1 = 1 \ ,$$

where the second equality holds since σ_j is a permutation for every $j \in \ell$. □

D QUADRATIC PROGRAMMING

In this section, we complement the study of (our version) of the offline algorithm of Du (2022), by checking its empirical performance for down-closed polytopes. Algorithms with better approximation guarantees are known when one is guaranteed to have such a constraint (Bian et al., 2017a). However, it is still important to understand the performance of algorithms designed for general polytope constraint when they happen to get a down-closed polytope. In particular, we note that Dürr et al. (2021) studied the empirical performance of their algorithm compared to the performance of the algorithm of Bian et al. (2017a) subject to such constraints, and we extend here their work by comparing the performance of their algorithm with that of newer algorithms. All the experiments presented in this section closely follow settings studied in (Dürr et al., 2021).

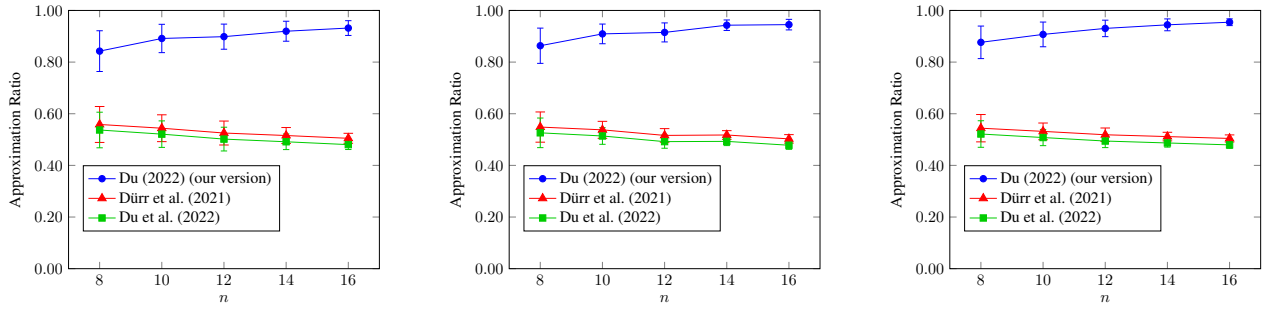
Consider the down-closed polytope given by

$$\mathcal{K} = \{\mathbf{x} \in \mathbb{R}^n_{\geq 0} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \leq \mathbf{u}, \mathbf{A} \in \mathbb{R}^{m \times n}_{\geq 0}, \mathbf{b} \in \mathbb{R}^m_{\geq 0}\} \ ,$$

where $\mathbf{A}$ is a non-negative matrix chosen in a way described below, $\mathbf{b}$ is the all ones vector, and $\mathbf{u}$ is a vector that acts as an upper bound on $\mathcal{K}$ and is given by $u_j = \min_{j \in [m]} b_i / A_{i,j}$ for every $j \in [n]$. We now describe a function F that we would like to maximize subject to $\mathcal{K}$. For every vector $\bar{0} \leq \mathbf{x} \leq \mathbf{u}$ (where $\bar{0}$ is the all zeros vector),

$$F(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{h}^T \mathbf{x} + c \ ,$$

where $\mathbf{H}$ is a matrix, $\mathbf{h}$ is a vector and c is a scalar. The matrix $\mathbf{H}$ is chosen in a way described below, and it is always non-positive, which guarantees that F is DR-submodular. Furthermore, once $\mathbf{H}$ is chosen, we follow Bian et al. (2017a) and set $\mathbf{h} = -0.1 \cdot \mathbf{H}^T \mathbf{u}$. Finally, to make sure that F is also non-negative, the value of c should be at least $M =$



(a) $m = \lfloor 0.5n \rfloor$ (b) $m = n$ (c) $m = \lfloor 1.5n \rfloor$

Figure 3: Quadratic Programming with Uniform Distribution

$-\min_{\bar{0} \leq \mathbf{x} \leq \mathbf{u}} \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{h}^T\mathbf{x}$. The value of M can be approximately obtained using QUADPROGIP[6] (Xia et al., 2020), and c is chosen to be $M + 0.1|M|$, which is a bit larger than the necessary minimum.

It remains to describe the way in which the entries of the matrices $\mathbf{H}$ and $\mathbf{A}$ are chosen. Below we describe two different random ways in which this can be done, and study the performance of the various algorithms on the instances generated in this way.

D.1 Uniform distribution

The first way to choose the matrices $\mathbf{H}$ and $\mathbf{A}$ is using a uniform distribution. Here, the matrix $\mathbf{H} \in \mathbb{R}^{n \times n}$ is a randomly generated symmetric matrix whose entries are drawn uniformly at random (and independently) from $[-1, 0]$, and $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a randomly generated matrix whose entries are drawn uniformly at random from $[v, v + 1]$ for $v = 0.01$ (this choice of v guarantees that the entries of $\mathbf{A}$ are strictly positive).

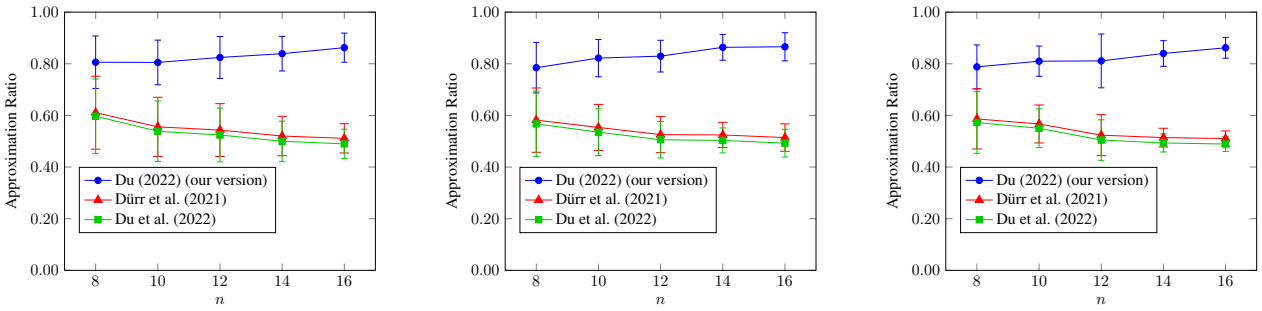
In each one of our experiments, we chose a different set of values for the dimensions n and m, and then drew an instance from the above distribution and executed on it 100 iterations of three algorithms: our explicit version from Section 3 of the algorithm of Du (2022) (with $\varepsilon = 0.03$), and the previous algorithms of Dürr et al. (2021) and Du et al. (2022). Each such experiment was repeated 100 times, and the results are depicted in Figure 3. In each plot of this figure, the x-axis represents the value of n, and the caption of the plot specifies how the value of m was calculated based on the value of n. The y-axis of the plots represents the approximation ratios obtained by the various algorithms compared to the optimum computed using a quadratic programming solver. One can observe that the two sub-exponential time algorithms of Dürr et al. (2021) and Du et al. (2022) exhibit similar performance, and (our version) of the newer algorithm of Du (2022) consistently and significantly outperforms them.

D.2 Exponential distribution

The other way to choose the matrices $\mathbf{H}$ and $\mathbf{A}$ is using an exponential distribution. Recall that given $\lambda > 0$, the exponential distribution $\exp(\lambda)$ is given by a density function assigning a density of $\lambda e^{-\lambda y}$ for every $y \geq 0$ and density 0 for negative y values. Then, $\mathbf{H} \in \mathbb{R}^{n \times n}$ is randomly generated symmetric matrix whose entries are drawn independently from $-\exp(1)$, and $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a randomly generated matrix whose entries are drawn independently from $\exp(0.25) + 0.01$.

For this way of generating $\mathbf{H}$ and $\mathbf{A}$, we repeated that same set of experiments as for the previous way of generating these matrices. The results of these experiments (averaged over 100 repetitions) are depicted in Figure 4. Again, we note that the two sub-exponential time algorithms of Dürr et al. (2021) and Du et al. (2022) exhibit similar performance, and (our version) of the newer algorithm of Du (2022) significantly outperforms them, especially as the dimension n grows.

[6]We have used IBM CPLEX optimization studio `https://www.ibm.com/products/ilog-cplex-optimization-studio`.



(a) $m = \lfloor 0.5n \rfloor$ (b) $m = n$ (c) $m = \lfloor 1.5n \rfloor$

Figure 4: Quadratic Programming with Exponential Distribution