

---

# Stochastic $L^{\natural}$ -convex Function Minimization

---

**Haixiang Zhang**  
Department of Mathematics  
University of California, Berkeley  
Berkeley, CA 94704  
haixiang\_zhang@berkeley.edu

**Zeyu Zheng**  
Department of IEO  
University of California, Berkeley  
Berkeley, CA 94704  
zyzheng@berkeley.edu

**Javad Lavaei**  
Department of IEO  
University of California, Berkeley  
Berkeley, CA 94704  
lavaei@berkeley.edu

## Abstract

We study an extension of the stochastic submodular minimization problem, namely, the stochastic  $L^{\natural}$ -convex minimization problem. We develop the first polynomial-time algorithms that return a near-optimal solution with high probability. We design a novel truncation operation to further reduce the computational complexity of the proposed algorithms. When applied to a stochastic submodular function, the computational complexity of the proposed algorithms is lower than that of the existing stochastic submodular minimization algorithms. In addition, we provide a strongly polynomial approximate algorithm. The algorithm execution also does not require any prior knowledge about the objective function except the  $L^{\natural}$ -convexity. A lower bound on the computational complexity that is required to achieve a high probability error bound is also derived. Numerical experiments are implemented to demonstrate the efficiency of our theoretical findings.

## 1 Introduction

As an important branch of optimization, discrete stochastic optimization problems (Futschik & Pflug, 1995; Gutjahr & Pflug, 1996; Futschik & Pflug, 1997; Kleywegt et al., 2002; Semelhago et al., 2020) appear in a wide range of applications such as image processing (Jegelka & Bilmes, 2011), pricing optimization (Ito & Fujimaki, 2016) and simulation-based optimization (Altman et al., 2003; Freund et al., 2017). Since a discrete optimization problem is generally  $\mathcal{NP}$ -hard to solve, specific structures are necessary to guarantee the existence of polynomial-time algorithms. In the special case when the feasible set is  $\{0, 1\}^d$  for some positive integer  $d$ , submodular functions are viewed as the discrete analogy of convex functions in the field of combinatorial optimization. The submodular function minimization (SFM) problem has wide applications in computer vision, economics, game theory and is well-studied in literature (Lee et al., 2015; Axelrod et al., 2020; Zhang et al., 2020). However, for certain problems, because of the stochastic structure, the exact value of the objective function is not available but only noisy samples of the objective function value are available. These scenarios, for example, include the optimization of expected performances for complex stochastic systems. In contrast to the deterministic problem, the stochastic SFM problem is less understood (Ito, 2019; Chakrabarty et al., 2017; Blais et al., 2019; Axelrod et al., 2020).

In this work, we consider the stochastic minimization problem of a class of generalized submodular functions, named the  $L^{\natural}$ -convex functions (Murota, 2003). More concretely, a function

$f : \{1, \dots, N\}^d \mapsto \mathbb{R}$  is called an  $L^{\natural}$ -convex function if it satisfies the *discrete midpoint convexity*:

$$f(x) + f(y) \geq f(\lceil (x+y)/2 \rceil) + f(\lfloor (x+y)/2 \rfloor), \quad \forall x, y \in \{1, \dots, N\}^d,$$

where  $N, d$  are positive integers and  $\lceil \cdot \rceil, \lfloor \cdot \rfloor$  are the ceiling and flooring functions applied component-wisely to vectors. We note that  $L^{\natural}$ -convex functions are reduced to submodular functions if  $N = 2$ . As an extension to the stochastic SFM problem, the stochastic  $L^{\natural}$ -convex minimization problem also appears in a variety of decision making problems. For example, the user satisfaction function of a bike-sharing system (Freund et al., 2017) and the expected total waiting time of a service network (Altman et al., 2003) are proved to be multimodular functions, which is equivalent to an  $L^{\natural}$ -convex function under a linear transformation. Since the evaluation of both functions includes the simulation of random user behaviors, we only have access to noisy objective function value oracles and thus both problems can be formulated as stochastic  $L^{\natural}$ -convex minimization problems. We refer the readers to the appendix and Murota (2003) for more concrete examples of  $L^{\natural}$ -convex minimization problems.

The contributions of this work are two-fold. For the algorithm design, we provide the first polynomial-time algorithms to achieve high-probability bounds for the stochastic  $L^{\natural}$ -convex function minimization problem. We note that we only consider *the number of objective function value evaluations* in the computational complexity, since the function value evaluation oracle is usually the major contributor to the running time of algorithms. The first algorithm is based on the observation that the Lovász extension of an  $L^{\natural}$ -convex function transforms the original problem to an equivalent continuous convex optimization problem. Therefore, we can apply the stochastic subgradient method to solve the continuous convex problem and then round the solution to an integral solution. The idea of combining the Lovász extension and the stochastic subgradient method has already been considered in Hazan & Kale (2011); Ito (2019) for stochastic online convex optimization and stochastic submodular minimization, respectively. However, the direct application of the stochastic subgradient method will lead to a large computational complexity. We add an extra truncation step to the stochastic subgradient and the computational complexity is reduced by  $O(d)$ . We also implement numerical experiments to show the significance of the truncation step. In the special case when  $N = 2$ , our problem reduces to the stochastic SFM problem and the computational complexity of the truncated subgradient method is  $O(d^2)$  smaller than the stochastic cutting-plane method in Blais et al. (2019). In addition, we propose the dimension reduction method and get a strongly polynomial approximate algorithm (see the definition in Section 4.2) with computational complexity  $O(d^3 N^2 (d + \log(N)) \epsilon^{-2} \log(1/\delta))$ . Finally, the uniform sampling method in Zhang et al. (2021) is extended to the multi-dimensional case and the computational complexity is independent of  $N$ . Table 1 summarizes the upper bounds on the computational complexities to find a solution whose objective value is at most  $\epsilon$  larger than the optimum with probability at least  $1 - \delta$ .

On the other hand, we derive lower bounds on the computational complexity for achieving a high-probability error bound. We construct a set of  $L^{\natural}$ -convex functions whose landscapes are close to each other while their optima are different with each other. Intuitively, the algorithm requires sufficiently many evaluations to differentiate those “similar”  $L^{\natural}$ -convex functions, where the similarity between functions is characterized by the KL-divergence. More rigorously, we utilize the information inequality in Kaufmann et al. (2016) and get the lower bound  $O(d\epsilon^{-2} \log(1/\delta))$  on the complexity. Recently, Ito (2019) proved a lower bound for achieving an  $\epsilon$ -expected error. There is a difference in terms of  $d$  and  $\delta$  regarding the lower bound results between ours and Ito’s. We note that these two lower bounds are established under different settings, i.e., the high-probability setting and the expectation setting. Hence, a directly comparison of the two lower bounds is unavailable; see the discussion in Section 5.

## 1.1 Notation

For  $N \in \mathbb{N}$ , we define  $[N] := \{1, 2, \dots, N\}$ . For a given set  $\mathcal{S}$  and an integer  $d \in \mathbb{N}$ , the product set  $\mathcal{S}^d$  is defined as  $\{(x_1, x_2, \dots, x_d) : x_i \in \mathcal{S}, i \in [d]\}$ . For two vectors  $x, y \in \mathbb{R}^d$ , we use  $(x \wedge y)_i := \min\{x_i, y_i\}$  and  $(x \vee y)_i := \max\{x_i, y_i\}$  to denote the component-wise minimum and maximum. Similarly, the ceiling function  $\lceil \cdot \rceil$  and the flooring function  $\lfloor \cdot \rfloor$  round each component to an integer when applied to vectors. The failing probability of algorithms is denoted as  $\delta$ . The notation  $f = O(g)$  (resp.  $f = \Theta(g)$ ) means that there exist absolute constants  $c_1, c_2 > 0$  such that  $f \leq c_1 g$  (resp.  $c_1 g \leq f \leq c_2 g$ ). Similarly, the notation  $f = \tilde{O}(g)$  (resp.  $f = \tilde{\Theta}(g)$ ) means that there exist absolute constants  $c_1, c_2 > 0$  and constant  $c_3 > 0$  independent of  $\delta$  such that  $f \leq c_1 g + c_3$  (resp.  $c_1 g \leq f \leq c_2 g + c_3$ ).

Table 1: Computational complexity for stochastic  $L^{\natural}$ -convex function minimization. Constants except for  $d, N, \epsilon, \delta, c$  are omitted in the  $O(\cdot)$  notation. The sub-optimality gap  $c$  refers to the objective function value gap between the global and the best sub-optimal solutions.

Algorithms	Unknown Sub-optimality Gap	Known Sub-optimality Gap $c$
Truncated Subgradient	$O(d^2 N^2 \epsilon^{-2} \log(1/\delta))$	$O(d^2 \log(N) c^{-2} \log(1/\delta))$
Multi-dim Uniform Sampling	$O(C^d \epsilon^{-2} \log(1/\delta))$	$O(C^d c^{-2} \log(1/\delta))$
Stochastic Cutting-plane	$O(d^3 N^2 (d + \log(N)) \epsilon^{-2} \log(1/\delta))$	$O(d^3 \log(N) (d + \log(N)) c^{-2} \log(1/\delta))$

## 2 Problem Formulation

We have in mind a complex stochastic system and consider the following discrete stochastic optimization problem

$$\min_x f(x) := \mathbb{E}[F(x, \xi_x)] \quad \text{s.t. } x \in \mathcal{X}, \quad (1)$$

in which  $\mathcal{X}$  is the discrete feasible set,  $\xi_x$  summarizes all the randomness supported on some proper probability space  $(\mathcal{Y}, \mathcal{B}_{\mathcal{Y}})$  and  $F : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$  is a deterministic function. Specifically,  $\xi_x$  summarizes all the random variables and stochastic processes involved in the system for the decision variable  $x$ . The function  $F$  maps the decision and associated randomness into a random system performance. For example, in service systems such as the bike-sharing service, this function  $F$  captures the full operational logic regarding how services are arranged. We write  $\xi_{x,1}, \xi_{x,2}, \dots, \xi_{x,n}$  as independent and identically distributed (i.i.d.) copies of  $\xi_x$ . We use  $\hat{F}_n(x) := \frac{1}{n} \sum_{j=1}^n F(x, \xi_{x,j})$  to denote the empirical mean of the  $n$  independent evaluations for the choice of decision variables  $x$ . The triad of the set  $\mathcal{X}$ , the space of randomness  $(\mathcal{Y}, \mathcal{B}_{\mathcal{Y}})$  and the function  $F(\cdot, \cdot)$  is called the *model* of problem (1). In this work, we assume the  $L^{\natural}$ -convexity of the objective function. For the ease of presentation, we assume the feasible set  $\mathcal{X} = [N]^d$ , where  $d$  denotes the dimension and  $N$  represents the number of choices on each dimension for the decision variable  $x$ .

**Assumption 1.** The objective function  $f(\cdot)$  is an  $L^{\natural}$ -convex function on  $\mathcal{X} = [N]^d$ , i.e.,

$$f(x) + f(y) \geq f(\lceil (x+y)/2 \rceil) + f(\lfloor (x+y)/2 \rfloor), \quad \forall x, y \in \mathcal{X},$$

*Remark 1.* In general, the definition of  $L^{\natural}$ -convexity also requires the feasible set  $\mathcal{X}$  to be an  $L^{\natural}$ -convex set, namely, the indicator function of  $\mathcal{X}$  should be  $L^{\natural}$ -convex on  $\mathbb{Z}^d$ . In our case, the feasible set  $\mathcal{X} = [N]^d$  is an  $L^{\natural}$ -convex set. Hence, we only need the discrete mid-point convexity property to define  $L^{\natural}$ -convex functions on  $\mathcal{X}$ .

Moreover, we assume that the distribution of the stochastic output  $F(x, \xi_x)$  is sub-Gaussian.

**Assumption 2.** The distribution of  $F(x, \cdot)$  is sub-Gaussian with a *known* parameter  $\sigma^2$  for all  $x \in \mathcal{X}$ .

The knowledge of  $\sigma^2$  represents the maximum level of stochasticity involved in the system performance. For scenarios where this  $\sigma^2$  is not known, it has to be adaptively learned and adds additional complexity to the problem, but is not the focus of this work. We further note that our algorithm design and analysis can be generalized to any probability distribution with provable concentration inequalities. The set of all models satisfying both Assumption 1 and Assumption 2 is denoted as  $\mathcal{MC}(\mathcal{X})$ , or simply  $\mathcal{MC}$ . The main goal is to design algorithms that guarantee the selection of a near-optimal solution with high probability. Formally, this criterion is called the  $(\epsilon, \delta)$ -*Probably Approximately Correct (PAC)* guarantee; see Even-Dar et al. (2002); Kaufmann et al. (2016); Ma & Henderson (2017). We say that a solution  $\hat{x}$ , as the output of an algorithm, satisfies the  $(\epsilon, \delta)$ -PAC guarantee if

$$\mathbb{P} \left[ f(\hat{x}) - \min_{x \in \mathcal{X}} f(x) \geq \epsilon \right] \leq \delta,$$

where the probability is calculated over the distribution of  $\hat{x}$ .

An algorithm is called an  $[(\epsilon, \delta)\text{-PAC}, \mathcal{MC}]$ -algorithm if it always returns an  $(\epsilon, \delta)$ -PAC solution for any model in  $\mathcal{MC}$ . The *computational complexity* of an algorithm is defined as the number of evaluations of  $F(\cdot, \cdot)$  before termination. The difficulty of satisfying the PAC guarantee for any model in  $\mathcal{MC}$  is characterized by the worst-case performance of an optimal algorithm.

**Definition 1.** The **computational complexity** of the  $(\epsilon, \delta)$ -PAC guarantee and class  $\mathcal{MC}$  is defined as

$$T(\epsilon, \delta, \mathcal{MC}) := \inf_{\mathbf{A} \text{ is } ((\epsilon, \delta)\text{-PAC}, \mathcal{MC})} \sup_{M \in \mathcal{MC}} \mathbb{E}[\tau_{\mathbf{A}}],$$

where the random variable  $\tau_{\mathbf{A}}$  is the computational complexity for each implementation of  $\mathbf{A}$ .

The remainder of this paper is focused on providing upper bounds on  $T(\epsilon, \delta, \mathcal{MC})$  by designing algorithms and providing lower bounds by an information-theoretical analysis. In this paper, we consider the behavior of algorithms in the high-probability regime, namely, in the setting when the failing probability  $\delta$  is small enough.

**Assumption 3.** The failing probability  $\delta$  is sufficiently small.

In general, the sub-Gaussian randomness leads to

$$T(\epsilon, \delta, \mathcal{M}) = C(\epsilon, \mathcal{M}) \cdot \log(1/\delta) \quad 0 < \delta \ll 1,$$

where  $C(\epsilon, \mathcal{M})$  is some function only depending on the precision  $\epsilon$  and the model class  $\mathcal{M}$ . For example, it is known that the sample complexity is  $\Theta(N^d \epsilon^{-2}) \log(1/\delta)$  for the general best-arm selection problem (Even-Dar et al., 2002) when no convex structure is in presence.

### 3 Truncated Stochastic Subgradient Algorithms

In this section, we develop gradient-based algorithms for problem (1). We first show that the properties of  $L^{\natural}$ -convex functions make it possible to construct a continuous convex function on the continuous cube  $[1, N]^d$  that has the same set of optima as the original objective function. Then, a rounding process is designed to round the solution of the continuous problem to an integral solution. Finally, the truncated stochastic subgradient algorithm is proposed and its computational complexity is analyzed. Moreover, in this section, we make the assumption that the  $\ell_{\infty}$ -Lipschitz constant of the objective function can be estimated a priori. For example, this estimation is possible for service systems optimization, where it represents an upper bound on how much impact an additional server over a certain time range contributes to the system performance.

**Assumption 4.** The  $\ell_{\infty}$ -Lipschitz constant of  $f(x)$  is known to be  $L$  a priori. Namely, we know beforehand that

$$|f(x) - f(y)| \leq L, \quad \forall x, y \in \mathcal{X} \quad \text{s.t.} \quad \|x - y\|_{\infty} \leq 1.$$

#### 3.1 Reduction to a continuous convex optimization problem

We first show that the original discrete minimization problem (1) can be equivalently transformed to a continuous convex optimization problem. Based on the fact that the restriction of an  $L^{\natural}$ -convex function to a hypercube is a submodular function (Murota, 2003), we show that we can piece together the Lovász extension of the objective function in each hypercube to form a continuous convex function on  $\text{conv}(\mathcal{X}) = [1, N]^d$ . More importantly, it can be shown that the continuous convex function has exactly the same global minima as the original  $L^{\natural}$ -convex function and an unbiased subgradient estimator is available at each point at the cost of  $2d$  evaluations to  $F(\cdot, \cdot)$ .

**Theorem 1.** *Suppose that function  $f(\cdot)$  is an  $L^{\natural}$ -convex function on  $[N]^d$ . Then, there exists a continuous convex function  $\tilde{f}(\cdot)$  on  $[1, N]^d$ , which we also name as the Lovász extension, such that*

- It holds  $\tilde{f}(x) = f(x)$  for any  $x \in \mathcal{X}$ .
- The minimizers of  $\tilde{f}$  satisfy  $\arg \min_{y \in [1, N]^d} \tilde{f}(y) = \arg \min_{y \in [N]^d} f(y)$ .
- For any point  $x \in [1, N]^d$ , there exists an unbiased estimator  $\hat{g}_x$  to a subgradient of  $\tilde{f}(\cdot)$  at point  $x$ , which can be evaluated via  $2d$  evaluations to  $F(\cdot, \cdot)$ .

---

**Algorithm 1** Rounding process to a feasible solution

---

**Input:** Model  $\mathcal{X}, \mathcal{B}_Y, F(x, \xi_x)$ , optimality guarantee parameters  $\epsilon, \delta$ ,  $(\epsilon/2, \delta/2)$ -PAC solution  $\bar{x}$  to problem (2).

**Output:** An  $(\epsilon, \delta)$ -PAC solution  $x^*$  to problem (1).

- 1: Compute the neighbouring points (defined in the appendix) of  $\bar{x}$ , denoted as  $S^0, \dots, S^d$ .
  - 2: Simulate at  $S^i$  until the  $1 - \delta/(4d)$  confidence half-width of  $\hat{F}_n(S^i)$  is smaller than  $\epsilon/4$  for all  $i$ .
  - 3: Return the optimum  $x^* \leftarrow \arg \min_{S \in \{S^0, \dots, S^d\}} \hat{F}_n(S)$ .
- 

---

**Algorithm 2** Projected and truncated SSGD method for the PAC guarantee

---

**Input:** Model  $\mathcal{X}, \mathcal{B}_Y, F(x, \xi_x)$ , optimality guarantee parameters  $\epsilon, \delta$ , number of iterations  $T$ , step size  $\eta$ , truncation threshold  $M$ .

**Output:** An  $(\epsilon, \delta)$ -PAC solution  $x^*$  to problem (1).

- 1: Choose the initial point  $x^0 \leftarrow ((N + 1)/2, \dots, (N + 1)/2)$ .
  - 2: **for**  $t = 0, \dots, T - 1$  **do**
  - 3:     Generate a stochastic subgradient  $\hat{g}^t$  at  $x^t$ .
  - 4:     Truncate the stochastic subgradient  $\tilde{g}^t \leftarrow \mathcal{T}_M(\hat{g}^t)$ .
  - 5:     Update  $x^{t+1} \leftarrow \mathcal{P}(x^t - \eta \tilde{g}^t)$ .
  - 6: **end for**
  - 7: Compute the averaging point  $\bar{x} \leftarrow (\sum_{t=0}^{T-1} x^t) / T$ .
  - 8: Round  $\bar{x}$  to an integral point by Algorithm 1.
- 

The proof along with a detailed discussion on  $L^{\natural}$ -convex functions is provided in the appendix.

Next, we describe the rounding process that takes an approximate solution to the relaxed problem

$$\min_{x \in [1, N]^d} \tilde{f}(x) \tag{2}$$

to an approximate integral solution to the original problem (1). We list the pseudo-code in Algorithm 1. The following theorem proves the correctness of Algorithm 1 and estimates the computational complexity. Note that all upper bounds on computational complexity in this paper are proved to hold almost surely and thus also in expectation.

**Theorem 2.** *Suppose that Assumptions 1-4 hold. The solution returned by Algorithm 1 satisfies the  $(\epsilon, \delta)$ -PAC guarantee. The computational complexity of Algorithm 1 is at most*

$$O \left[ \frac{d}{\epsilon^2} \log \left( \frac{d}{\delta} \right) + d \right] = \tilde{O} \left[ \frac{d}{\epsilon^2} \log \left( \frac{1}{\delta} \right) \right].$$

Although we transform the original problem to a continuous convex problem, the algorithm design in this work relies heavily on other two important properties of  $L^{\natural}$ -convex functions. The first property is that the global minima are integral points, which makes it possible to design strongly polynomial algorithms; see Section 4.2. The second property is that the Lovász extension is a piecewise linear function. When the sub-optimality gap is known, we prove that the Lovász extension satisfies the Weak Sharp Minima property and the subgradient descent method can be accelerated to achieve a better dependence on the scale  $N$ ; see the appendix.

### 3.2 Algorithm and complexity analysis

Utilizing the tools derived in the last subsection, we can first apply the stochastic subgradient (SSGD) method to find an approximate solution to problem (2) and then round the solution to get an approximate solution to problem (1). The main difficulty of giving sharp upper bounds lies in the fact that the Lovász extension is neither smooth nor strongly-convex. This property of the Lovász extension prohibits the application of Nesterov’s acceleration method (Nesterov, 1983) and variance reduction methods (Johnson & Zhang, 2013; Defazio et al., 2014).

Now, we propose the projected and truncated SSGD method for the  $(\epsilon, \delta)$ -PAC guarantee. The orthogonal projection onto the feasible set, which is defined as

$$\mathcal{P}(x) := (x \wedge N\mathbf{1}) \vee \mathbf{1}, \quad \forall x \in \mathbb{R}^d,$$

is applied after each iteration to ensure the feasibility of iteration point. In addition to the projection, the component-wise truncation of stochastic subgradient is critical in reducing expected computational complexities. The truncation operator with threshold  $M > 0$  is defined as

$$\mathcal{T}_M(g) := (g \wedge M\mathbf{1}) \vee (-M\mathbf{1}), \quad \forall g \in \mathbb{R}^d.$$

The pseudo-code of projected and truncated SSGD method is listed in Algorithm 2. With a suitable choice of the step size, the truncation threshold and the number of iterations, Algorithm 2 returns an  $(\epsilon, \delta)$ -PAC solution and the computational complexity scales as  $O(d^2N^2)$ .

**Theorem 3.** *Suppose that Assumptions 1-4 hold. If we choose*

$$T = \tilde{\Theta} \left[ \frac{dN^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right], \quad M = \tilde{\Theta} \left[ \sqrt{\log\left(\frac{dNT}{\epsilon}\right)} \right], \quad \eta = \frac{N}{M\sqrt{T}},$$

then Algorithm 2 returns an  $(\epsilon, \delta)$ -PAC solution. Furthermore, we have

$$T(\epsilon, \delta, \mathcal{MC}) = O \left[ \frac{d^2N^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right) + \frac{d^3N^2}{\epsilon^2} \log\left(\frac{d^2N}{\epsilon^3}\right) + \frac{d^3N^2L^2}{\epsilon^2} \right] = \tilde{O} \left[ \frac{d^2N^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

The intuition behind the truncation operation is that the truncation makes the norm of the subgradient smaller, which in turn allows the choice of a larger step size. With the larger step size, the convergence rate can be improved. Using the truncation operation to accelerate the algorithms is novel in the context of the stochastic  $L^{\natural}$ -convex function minimization problem. The truncation operation has been considered in other contexts, such as the sparse phase retrieval problem (Wang et al., 2017). We can prove that the computational complexity without the truncation step (i.e.,  $M = \infty$ ) is

$$\tilde{O} \left[ \frac{d^3N^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

Hence, the truncation of stochastic subgradient is necessary for reducing the asymptotic expected computational complexity. This relation has also been observed in our numerical experiments. While the error of the normal SSGD method only contains the optimization residual and the variance terms, the error of the truncated SSGD method has an extra bias term. We note that the bias term can be made arbitrarily small with high probability by choosing large enough  $M$  and utilizing the tail bound for sub-Gaussian random variables, and therefore the total error can be controlled similarly as the normal SSGD method. We mention that Algorithm 2 also applies when  $\mathcal{X}$  is a general  $L^{\natural}$ -convex set.

Next, we consider a special case with additional condition that the norm of stochastic subgradients is bounded almost surely. We then show that this additional condition, whenever available, can contribute to further complexity reduction of our proposed algorithm.

**Assumption 5.** There exist a constant  $G$  and an unbiased subgradient estimator  $\hat{g}$  such that

$$\mathbb{P}[\|\hat{g}\|_1 \leq G] = 1.$$

Moreover, the cost of generating each  $\hat{g}$  is at most  $\beta$  evaluations of  $F(\cdot, \cdot)$ .

We note that  $G$  and  $\beta$  may depend on  $d$  and  $N$ . This assumption is common when analyzing the high-probability error bound of stochastic subgradient methods (Hazan & Kale, 2011; Xu et al., 2016). We first give two examples where Assumption 5 holds.

**Example 1.** Suppose that the randomness of each feasible  $x \in \mathcal{X}$  shares the same measure space, i.e., there exists a measure space  $(Z, \mathcal{B}_Z)$  such that  $\xi_x$  can be any element in the measure space for all  $x \in \mathcal{X}$ . Moreover, for any fixed  $\xi \in \mathcal{B}_Z$ , the function  $F(\cdot, \xi)$  is also  $L^{\natural}$ -convex and has  $\ell_{\infty}$ -Lipschitz constant  $\tilde{L}$ . Then, the properties of  $L^{\natural}$ -convex functions imply that there exists an unbiased subgradient estimator  $\hat{g}$  such that the cost of evaluating  $\hat{g}$  is  $d + 1$  and  $\|\hat{g}\|_1 \leq 3\tilde{L}/2$ . Therefore, the Assumption 5 holds with  $G = 3\tilde{L}/2$  and  $\beta = d + 1$ . We note that this example is similar to the case when  $k = d$  in Ito (2019) while the noise of our problem can be unbounded.

When the distribution at each feasible point is bounded, we show that Assumption 5 also holds.

**Example 2.** Suppose that the distribution at each point  $x \in \mathcal{X}$  is bounded, namely, there exists a constant  $C \geq 0$  such that

$$\mathbb{P}_{\xi_x}[|F(x, \xi_x)| \leq C] = 1, \quad \forall x \in \mathcal{X}.$$

We note that bounded distributions are sub-Gaussian with parameter  $2C^2$ . In this case, there exists an unbiased subgradient estimator such that Assumption 5 holds with  $G = 2dC$  and  $\beta = 2$ .

More details about these two examples are provided in the appendix. Since the stochastic subgradient is bounded, the truncation step is unnecessary in Algorithm 2. The computational complexity of Algorithm 2 is estimated in the following theorem. The proof is similar to Lemma 10 in Hazan & Kale (2011) and, since the feasible set is the hypercube  $[1, N]^d$ , we replace the  $\ell_2$ -norm with the  $\ell_\infty$ -norm to bound distances between points.

**Theorem 4.** *Suppose that Assumptions 1-5 hold. If we skip the truncation step in Algorithm 2 (i.e., set  $M = \infty$ ) and choose*

$$T = \tilde{\Theta} \left[ \frac{N^2(L+G)^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right], \quad \eta = \sqrt{\frac{dN^2}{TG^2}},$$

then Algorithm 2 returns an  $(\epsilon, \delta)$ -PAC solution. Furthermore, we have

$$\begin{aligned} T(\epsilon, \delta, \mathcal{MC}) &= O \left[ \frac{\beta N^2(L+G)^2 + d}{\epsilon^2} \log\left(\frac{1}{\delta}\right) + \frac{d^2 G^2}{\epsilon^2} \right] \\ &= \tilde{O} \left[ \frac{\beta N^2(L+G)^2 + d}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right]. \end{aligned}$$

In the case of Example 1, we have  $\beta = d + 1, G = 3\tilde{L}/2$  and then the computational complexity of Algorithm 2 is at most  $\tilde{O}[dN^2(L + \tilde{L})^2\epsilon^{-2} \log(1/\delta)]$ . If both Lipschitz constants are independent of  $d$  and  $N$ , the computational complexity becomes  $\tilde{O}[dN^2\epsilon^{-2} \log(1/\delta)]$ , which is  $O(d)$  better than the general case without Assumption 5. Moreover, in the case of Example 2, we have  $G = d$  and  $\beta = 2$ . Hence, the computational complexity is at most  $\tilde{O}[d^2N^2\epsilon^{-2} \log(1/\delta)]$ , even without the truncation.

Finally, we consider the case when the global optimum is unique and the sub-optimality gap is known to be at least  $c > 0$ . In this case, the Lovász extension satisfies the *weak sharp minimum* condition (Burke & Ferris, 1993) and we can design an acceleration scheme similar to that in Xu et al. (2016). The acceleration scheme reduces the dependence on  $N$  from  $N^2$  to  $\log(N)$  and we provide detailed discussions in the appendix.

## 4 Stochastic Localization Algorithms

Although the algorithms developed in Section 3 are guaranteed to find an approximate solution in polynomial time, gradient-based algorithms require the prior knowledge about the Lipschitz constant of the objective function to choose the optimal step size. In addition, the computational complexity of subgradient methods has a quadratic dependence on the Lipschitz constant. However, for certain applications, the upper bound on the Lipschitz constant may be difficult to estimate a priori or is very large so that the computational complexity is prohibitive for large-scale problems. To deal with those difficult scenarios, we utilize the discrete natural of problem (1) to design algorithms that do not require the prior knowledge about the objective function except the  $L^{\natural}$ -convexity and whose computational complexity has an upper bound that is independent of the Lipschitz constant. More specifically, we propose two approaches to deal with this problem. The first approach is to generalize the uniform sampling algorithm in Zhang et al. (2021) to the multi-dimensional case. In the second approach, we propose a strongly polynomial approximate algorithm by gradually reducing the dimension.

### 4.1 Multi-dimensional uniform sampling algorithm

We first generalize the uniform sampling algorithm in Zhang et al. (2021) to the multi-dimensional case. The basic idea is that if the computational complexity for finding an  $(\epsilon, \delta)$ -PAC solution has the form  $C\epsilon^{-2} \log(1/\delta)$ , we can view the algorithm as a (possibly biased) estimator with sub-Gaussian tails to the optimal value, where  $C > 0$  is a constant. For example, the uniform sampling algorithm provides a sub-Gaussian estimator for the optimal value of problem (1) in the one-dimensional case. Now suppose that we have derived a sub-Gaussian estimator for the optimal value for the  $(d - 1)$ -dimensional case. Consider the function

$$f^1(x_1) := \min_{x_2, \dots, x_d \in [N]} f(x_1, \dots, x_d), \quad \forall x_1 \in [N], \quad (3)$$

which is an one-dimensional convex function. Since the  $(d - 1)$ -dimensional algorithm has a sub-Gaussian tail, we can view each implementation of the algorithm as a noisy function value oracle to  $f^1(\cdot)$  and the uniform sampling algorithm can be modified to solve the convex problem (3). By a similar method as Zhang et al. (2021), we can prove that the computational complexity of the resulting  $d$ -dimensional algorithm is a constant multiple of that of the  $(d - 1)$ -dimensional algorithm. By the induction method, we get an approximate algorithm for the multi-dimensional problem (1) whose computational complexity is independent of the Lipschitz constant  $L$  and the scale  $N$  in the high-probability regime. A rigorous proof to the following result is provided in the appendix.

**Theorem 5.** *Suppose that Assumptions 1-3 hold. There exists an  $[(\epsilon, \delta)$ -PAC, MC]-algorithm with computational complexity*

$$T(\epsilon, \delta, \mathcal{MC}) = \tilde{O} \left[ \frac{C^d}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right],$$

where  $C$  is an absolute constant.

The exponential dependence on the dimension  $d$  makes it impossible to use the generalized algorithm even for moderate dimension problems. However, this result implies that the computational complexity almost does not grow with  $N$  in the regime  $N \gg d$  in the asymptotic regime with the failing probability  $\delta$  is very small. Therefore this algorithm can be preferable when  $d$  is small and  $N$  is large.

## 4.2 Dimension reduction algorithm

Next, we take an alternative approach and derive a strongly polynomial approximate algorithm for problem (1). In this work, we do not treat the parameters  $d$  and  $N$  as input parameters. The message we hope to convey is that the sample complexities are upper bounded by a constant independent of the objective function. This property is important for some applications of the stochastic  $L^{\natural}$ -function minimization problem where the objective function has unfavorable properties (e.g. large Lipschitz constant, small sub-optimality gap). Such unfavorable properties may not be known a priori before the problem is solved. If the parameters  $d$  and  $N$  is treated as an input parameter, the proposed algorithms are pseudo-polynomial. We observe that a subgradient of a convex function provides a separation oracle to the global minima, which can be used to gradually shrink the set that contains global minima. This idea leads to the deterministic cutting-plane methods (Vaidya, 1996; Bertsimas & Vempala, 2004; Lee et al., 2015). In this work, we develop a framework for extending cutting-plane methods to the stochastic case. We prove that if we average  $\tilde{O}[dN^2\epsilon^{-2} \log(1/\delta)]$  stochastic subgradients, we can get a separation oracle for the set of  $\epsilon$ -approximate solutions with probability at least  $1 - \delta$ . Utilizing this stochastic separation oracle, we may extend the deterministic cutting-plane methods to the stochastic case to find PAC solutions.

However, the stochastic cutting-plane methods still require the knowledge of the Lipschitz constant of the objective function. To design a strongly polynomial algorithm, we observe that if a convex body  $P \subset \mathbb{R}^d$  has a volume smaller than  $(d!)^{-1} = O[\exp(-d \log(d))]$ , then all integral points in  $P$  must lie on a hyperplane. Otherwise, if there exist  $d + 1$  points  $x_0, \dots, x_d \in P$  that are not on the same hyperplane, then the convex body  $P$  contains the polytope  $\text{conv}\{x_0, \dots, x_d\}$ , which has the volume

$$\frac{1}{d!} |\det(x_1 - x_0, \dots, x_d - x_0)| \geq \frac{1}{d!} > \text{vol}(P),$$

where  $\text{vol}(P)$  is the volume of the polytope  $P$ . This leads to a contradiction. Hence, we may use the stochastic version of Vaidya's method (Vaidya, 1996) or the random walk-based (Bertsimas & Vempala, 2004) cutting-plane method to reduce the volume of the search polytope  $P$  to  $O[\exp(-d \log(d))]$  and then reduce the problem dimension by projecting the polytope onto the hyperplane. After  $d - 1$  dimension reductions, we have an one-dimensional convex problem and algorithms in Zhang et al. (2021) can be applied.

**Theorem 6.** *Suppose that Assumptions 1-3 hold. Then, the dimension reduction algorithm is an  $[(\epsilon, \delta)$ -PAC, MC]-algorithm with computational complexity*

$$T(\epsilon, \delta, \mathcal{MC}) = \tilde{O} \left[ \frac{d^3 N^2 (d + \log(N))}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

The dimension reduction method is the first algorithm for the stochastic  $L^{\natural}$ -convex function minimization that is strongly polynomial and does not require prior knowledge about the objective function

except the  $L^{\natural}$ -convexity. Recently, Jiang (2020) proved that the LLL algorithm (Lenstra et al., 1982) is able to reduce the number of arithmetic operations to be polynomial and make the algorithm more practical, but the LLL algorithm will not reduce the number of evaluations to the stochastic objective values. Their results can be naturally extended to the stochastic case to achieve the PAC guarantee. We note that if we allow exponentially many arithmetic operations, the LLL algorithm is not necessary and the computational complexity can be reduced to  $\tilde{O}[d^4 N^2 \epsilon^{-2} \log(1/\delta)]$ . Similar to the gradient-based method, the acceleration scheme can reduce the number of required objective function value evaluations if we have a positive lower bound to the sub-optimality gap.

## 5 Lower Bounds

In this section, we provide a lower bound result on  $[(\epsilon, \delta)$ -PAC,  $\mathcal{MC}]$ -algorithms. This result shows the limit of algorithms that achieve the PAC guarantee for any stochastic  $L^{\natural}$ -convex functions. To prove lower bounds, we make non-trivial extensions to the construction in Graur et al. (2020) to get several convex models that are “similar” to each other but they have distinct optimal solutions, where the similarity between two models is characterized by the KL-divergence between their distributions. Hence, algorithms need a large number of evaluations to differentiate these models. More rigorously, the information-theoretical inequality in Kaufmann et al. (2016) is utilized to get the following lower bound:

**Theorem 7.** *Suppose Assumptions 1-3 hold. We have*

$$T(\epsilon, \delta, \mathcal{MC}) \geq \Theta \left[ \frac{d}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

Comparing with the lower bound for the expected error in Ito (2019), our high-probability lower bound is  $O(d)$  smaller but includes an extra term  $\log(1/\delta)$ . Since the failing probability  $\delta$  can be sufficiently small, the two lower bounds are not directly comparable. The results in Sections 3-4 imply that

$$T(\epsilon, \delta, \mathcal{MC}) = \tilde{O} \left[ \frac{\min\{d^2 N^2, M^d\}}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

Therefore, we conjecture that the optimal computational complexity is asymptotically independent of  $N$  and leave the design of optimal algorithms for future works.

## 6 Numerical Experiments

In this section, we present a brief account of numerical experiments to illustrate several of our theoretical findings. First, we show that the extra new truncation step in the gradient-based method can achieve a better performance, compared to the gradient-based method without the truncation step. We implement Algorithm 2 with and without the truncation operation on a staffing allocation stochastic optimization problem with staffing cost constraint, which can be formulated into the regularized and the constrained forms. Details about the problem setup are provided in the appendix. We summarize the computational complexities and the objective values in Table 2. We can see that the two algorithms achieve similar objective values but the “Truncated” version observes an advantage in the computational cost compared to the “Not Truncated” version. We note that our algorithms do not require the feasible region to be a hypercube and in fact the problems considered in this section have more general feasible set with constraints. Especially, we can see that the truncation plays an important role in reducing the computational complexity, especially when the dimension is high.

Next, we compare the performances of the gradient-based methods and stochastic localization algorithms on multi-dimensional problems. To facilitate the comparison, we construct a separable stochastic convex function, which is an  $L^{\natural}$ -convex function by the results in Murota (2003). We have observed that using  $(N\epsilon/4, \delta/4)$ - $\mathcal{SO}$  oracles is enough for producing high-probability guarantees using the dimension reduction method within the range  $1 \leq N \leq 150$ . More details about the experiment setup are provided in the appendix. We summarize the results in Table 3, where the coverage rate refers to the percentage of implementations that produce an  $\epsilon$ -optimal solution. Since the coverage rates of the algorithms are 100%, the PAC guarantee is satisfied by all of the algorithms except the  $(d, N) = (20, 50)$  case for the dimension reduction method. This failure is because of

Table 2: Complexity and returned objective value of Algorithm 2 on the optimal allocation problem. The cost refers to the average computational complexity in all the following tables.

Params.		Regularized				Constrained			
		Truncated		Not truncated		Truncated		Not truncated	
d	N	Cost	Obj.	Cost	Obj.	Cost	Obj.	Cost	Obj.
4	10	2.99e5	2.10e2	6.56e5	2.11e2	3.00e5	4.76e1	4.99e5	4.97e1
4	20	1.21e5	3.53e2	2.61e5	3.53e2	1.14e5	5.23e1	1.77e5	5.38e1
4	30	8.85e4	4.75e2	1.68e5	4.76e2	7.38e4	5.24e1	1.23e5	5.21e1
4	40	6.25e4	5.91e2	1.34e5	6.07e2	5.28e4	5.31e1	9.24e4	5.28e1
4	50	5.34e4	7.07e2	1.08e5	7.07e2	4.66e4	5.64e1	6.61e4	5.51e1
8	10	1.19e6	1.75e2	3.80e6	1.76e2	1.20e6	3.11e1	2.23e6	3.02e1
12	10	2.68e6	1.59e2	9.48e6	1.59e2	2.69e6	1.87e1	5.36e6	1.86e1
16	10	6.35e6	1.49e2	1.31e7	1.50e2	4.78e6	1.49e1	1.08e7	1.41e1
20	10	9.91e6	1.43e2	2.09e7	1.46e2	9.43e6	1.17e1	1.70e7	1.28e1
24	10	1.50e7	1.35e2	3.09e7	1.41e2	1.36e7	9.43e0	2.10e7	1.17e1

Table 3: Complexity and coverage rate of different algorithms on separable convex functions.

Params.		Truncated subgradient		Dimension reduction	
d	N	Cost( $10^7$ )	Rate(%)	Cost( $10^7$ )	Rate(%)
2	50	0.39	100.0	0.19	100.0
2	100	0.44	100.0	0.22	100.0
2	150	0.51	100.0	0.25	100.0
6	50	0.96	100.0	1.51	100.0
6	100	1.07	100.0	1.76	100.0
6	150	1.24	100.0	1.92	100.0
10	50	1.26	100.0	3.47	100.0
10	100	1.43	100.0	4.02	100.0
10	150	1.64	100.0	4.33	100.0
15	50	1.52	100.0	4.14	100.0
20	50	1.69	100.0	4.28	99.00

our loosened choice of the separation oracles. When the dimension  $d$  is not large, the dimension reduction method has the best performance and has the advantage of not requiring any knowledge about the Lipschitz constant; when the dimension  $d$  is large, the higher order dependence on  $d$  makes the dimension reduction method inferior to the stochastic subgradient method. In addition, the computational complexity of the dimension reduction method has little dependence on  $N$ . This is because we choose the separation oracle to be  $(N\epsilon/4, \delta/4)$ - $\mathcal{SO}$  and the computational complexity of each evaluation is independent of  $N$ .

## 7 Conclusions

In this work, we provide the first polynomial-time algorithms for the stochastic  $L^h$ -convex minimization problem, which serves as an important extension of the stochastic submodular minimization problem. Comparing to existing stochastic submodular minimization algorithms, our algorithms either achieve a lower computational complexity or do not require extra information about the objective function. We also derive a lower bound on the computational complexity to achieve the PAC guarantee. Numerical results are implemented to verify our theoretical findings.

## Acknowledgments and Disclosure of Funding

Haixiang Zhang and Javad Lavaei were supported by grants from AFOSR, ARO, ONR, NSF and C3.ai Digital Transformation Institute.

## References

- Altman, E., Gaujal, B., and Hordijk, A. *Discrete-event control of stochastic networks: Multimodularity and regularity*. Springer, 2003.
- Axelrod, B., Liu, Y. P., and Sidford, A. Near-optimal approximate discrete and continuous submodular function minimization. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 837–853. SIAM, 2020.
- Bach, F. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1-2):419–459, 2019.
- Balasubramanian, K. and Ghadimi, S. Zeroth-order nonconvex stochastic optimization: Handling constraints, high-dimensionality and saddle-points. *arXiv preprint arXiv:1809.06474*, 2018.
- Bertsimas, D. and Vempala, S. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.
- Blais, E., Canonne, C. L., Eden, T., Levi, A., and Ron, D. Tolerant junta testing and the connection to submodular optimization and function isomorphism. *ACM Transactions on Computation Theory (TOCT)*, 11(4):1–33, 2019.
- Borjesson, P. and Sundberg, C.-E. Simple approximations of the error function  $q(x)$  for communications applications. *IEEE Transactions on Communications*, 27(3):639–643, 1979.
- Burke, J. V. and Ferris, M. C. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993.
- Chakrabarty, D., Lee, Y. T., Sidford, A., and Wong, S. C.-w. Subquadratic submodular function minimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1220–1231, 2017.
- Chen, X., Zhou, E., and Hu, J. Discrete optimization via gradient-based adaptive stochastic search methods. *IIE Transactions*, 50(9):789–805, 2018.
- Chick, S. E. Subjective probability and bayesian methodology. *Handbooks in Operations Research and Management Science*, 13:225–257, 2006.
- Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.
- Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Wibisono, A. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.
- Eckman, D. J. and Henderson, S. G. Fixed-confidence, fixed-tolerance guarantees for selection-of-the-best procedures. Technical report, Working paper, Cornell University, School of Operations Research and . . . , 2018a.
- Eckman, D. J. and Henderson, S. G. Guarantees on the probability of good selection. In *2018 Winter Simulation Conference (WSC)*, pp. 351–365. IEEE, 2018b.
- Eckman, D. J., Plumlee, M., and Nelson, B. L. Plausible screening using functional properties for simulations with large solution spaces. Working paper, 2020.
- Even-Dar, E., Mannor, S., and Mansour, Y. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pp. 255–270. Springer, 2002.
- Fan, W., Hong, L. J., and Nelson, B. L. Indifference-zone-free selection of the best. *Operations Research*, 64(6):1499–1514, 2016.
- Favati, P. Convexity in nonlinear integer programming. *Ricerca operativa*, 53:3–44, 1990.

- Frazier, P., Powell, W., and Dayanik, S. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.
- Frazier, P. I. A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. *Operations Research*, 62(4):926–942, 2014.
- Freund, D., Henderson, S. G., and Shmoys, D. B. Minimizing multimodular functions and allocating capacity in bike-sharing systems. In *International Conference on Integer Programming and Combinatorial Optimization*, pp. 186–198. Springer, 2017.
- Fu, M. C. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.
- Fujishige, S. Theory of submodular programs: A fenchel-type min-max theorem and subgradients of submodular functions. *Mathematical programming*, 29(2):142–155, 1984.
- Fujishige, S. *Submodular functions and optimization*. Elsevier, 2005.
- Futschik, A. and Pflug, G. Confidence sets for discrete stochastic optimization. *Annals of Operations Research*, 56(1):95–108, 1995.
- Futschik, A. and Pflug, G. C. Optimal allocation of simulation experiments in discrete stochastic optimization and approximative algorithms. *European Journal of Operational Research*, 101(2):245–260, 1997.
- Ghadimi, S. and Lan, G. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- Graur, A., Pollner, T., Ramaswamy, V., and Weinberg, S. M. New query lower bounds for submodular function minimization. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- Gutjahr, W. J. and Pflug, G. C. Simulated annealing for noisy cost functions. *Journal of global optimization*, 8(1):1–13, 1996.
- Hassani, H., Soltanolkotabi, M., and Karbasi, A. Gradient methods for submodular maximization. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5843–5853, 2017.
- Hassidim, A. and Singer, Y. Submodular optimization under noise. In *Conference on Learning Theory*, pp. 1069–1122. PMLR, 2017.
- Hazan, E. and Kale, S. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 421–436, 2011.
- Homem-de Mello, T. and Bayraksan, G. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, 19(1):56–85, 2014.
- Hong, L. J. and Nelson, B. L. Discrete optimization via simulation using compass. *Operations Research*, 54(1):115–129, 2006.
- Hong, L. J., Nelson, B. L., and Xu, J. Speeding up compass for high-dimensional discrete optimization via simulation. *Operations Research Letters*, 38(6):550–555, 2010.
- Hong, L. J., Nelson, B. L., and Xu, J. Discrete optimization via simulation. In *Handbook of simulation optimization*, pp. 9–44. Springer, 2015.
- Hong, L. J., Fan, W., and Luo, J. Review on ranking and selection: A new perspective. *arXiv preprint arXiv:2008.00249*, 2020.
- Hu, J., Fu, M. C., and Marcus, S. I. A model reference adaptive search method for global optimization. *Operations Research*, 55(3):549–568, 2007.

- Hu, J., Fu, M. C., Marcus, S. I., et al. A model reference adaptive search method for stochastic global optimization. *Communications in Information & Systems*, 8(3):245–276, 2008.
- Hunter, S. R. and Nelson, B. L. Parallel ranking and selection. In *Advances in Modeling and Simulation*, pp. 249–275. Springer, 2017.
- Ito, S. Submodular function minimization with noisy evaluation oracle. In *Advances in Neural Information Processing Systems*, pp. 12103–12113, 2019.
- Ito, S. and Fujimaki, R. Large-scale price optimization via network flow. In *NIPS*, pp. 3855–3863, 2016.
- Jegelka, S. and Bilmes, J. Submodularity beyond submodular energies: coupling edges in graph cuts. In *CVPR 2011*, pp. 1897–1904. IEEE, 2011.
- Jian, N., Freund, D., Wiberg, H. M., and Henderson, S. G. Simulation optimization for a large-scale bike-sharing system. In *2016 Winter Simulation Conference (WSC)*, pp. 602–613. IEEE, 2016.
- Jiang, H. Minimizing convex functions with integral minimizers. *arXiv preprint arXiv:2007.01445*, 2020.
- Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.
- Karimi, M. R., Lucic, M., Hassani, H., and Krause, A. Stochastic submodular maximization: the case of coverage functions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6856–6866, 2017.
- Kaufmann, E., Cappé, O., and Garivier, A. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.
- Kim, S.-H. and Nelson, B. L. Selecting the best system. *Handbooks in operations research and management science*, 13:501–534, 2006.
- Kleywegt, A. J., Shapiro, A., and Homem-de Mello, T. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.
- Larson, J., Menickelly, M., and Wild, S. M. Derivative-free optimization methods. *arXiv preprint arXiv:1904.11585*, 2019.
- Lee, Y. T., Sidford, A., and Wong, S. C.-w. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 1049–1065. IEEE, 2015.
- Lenstra, A. K., Lenstra, H. W., and Lovász, L. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261(ARTICLE):515–534, 1982.
- Lim, E. Stochastic approximation over multidimensional discrete sets with applications to inventory systems and admission control of queueing networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 22(4):1–23, 2012.
- Lovász, L. Submodular functions and convexity. In *Mathematical programming the state of the art*, pp. 235–257. Springer, 1983.
- Ma, S. and Henderson, S. G. An efficient fully sequential selection procedure guaranteeing probably approximately correct selection. In *2017 Winter Simulation Conference (WSC)*, pp. 2225–2236. IEEE, 2017.
- Mokhtari, A., Hassani, H., and Karbasi, A. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *International Conference on Artificial Intelligence and Statistics*, pp. 1886–1895. PMLR, 2018.
- Murota, K. Discrete convex analysis. In *Society for Industrial and Applied Mathematics*. Citeseer, 2003.

- Nelson, B. L. Optimization via simulation over discrete decision variables. In *Risk and Optimization in an Uncertain World*, pp. 193–207. Informs, 2010.
- Nemirovsky, A. S. and Yudin, D. B. *Problem complexity and method efficiency in optimization*. 1983.
- Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.
- Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.
- Nesterov, Y. E. A method for solving the convex programming problem with convergence rate  $O(1/k^2)$ . In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.
- Park, C. and Kim, S.-H. Penalty function with memory for discrete optimization via simulation with stochastic constraints. *Operations Research*, 63(5):1195–1212, 2015.
- Park, C., Telci, I. T., Kim, S.-H., and Aral, M. M. Designing an optimal water quality monitoring network for river systems using constrained discrete optimization via simulation. *Engineering Optimization*, 46(1):107–129, 2014.
- Sekar, S., Vojnovic, M., and Yun, S.-Y. A test score-based approach to stochastic submodular optimization. *Management Science*, 67(2):1075–1092, 2021. doi: 10.1287/mnsc.2020.3585. URL <https://doi.org/10.1287/mnsc.2020.3585>.
- Semelhago, M., Nelson, B. L., Song, E., and Wächter, A. Rapid discrete optimization via simulation with Gaussian Markov random fields. *INFORMS journal on Computing*, Articles in Advance, 2020.
- Shaked, M. and Shanthikumar, J. G. Stochastic convexity and its applications. *Advances in Applied Probability*, 20(2):427–446, 1988.
- Shamir, O. A variant of azuma’s inequality for martingales with subgaussian tails. *arXiv preprint arXiv:1110.2392*, 2011.
- Singhvi, D., Singhvi, S., Frazier, P. I., Henderson, S. G., O’Mahony, E., Shmoys, D. B., and Woodard, D. B. Predicting bike usage for new york city’s bike sharing system. In *AAAI Workshop: Computational Sustainability*. Citeseer, 2015.
- Sun, L., Hong, L. J., and Hu, Z. Balancing exploitation and exploration in discrete optimization via simulation through a gaussian process-based search. *Operations Research*, 62(6):1416–1438, 2014.
- Vaidya, P. M. A new algorithm for minimizing convex functions over convex sets. *Mathematical programming*, 73(3):291–341, 1996.
- Wang, G., Zhang, L., Giannakis, G. B., Akçakaya, M., and Chen, J. Sparse phase retrieval via truncated amplitude flow. *IEEE Transactions on Signal Processing*, 66(2):479–491, 2017.
- Wang, H., Pasupathy, R., and Schmeiser, B. W. Integer-ordered simulation optimization using r-spline: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(3):1–24, 2013.
- Wolff, R. W. and Wang, C.-L. On the convexity of loss probabilities. *Journal of applied probability*, pp. 402–406, 2002.
- Xu, J., Nelson, B. L., and Hong, J. L. Industrial strength compass: A comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 20(1):1–29, 2010.
- Xu, Y., Lin, Q., and Yang, T. Accelerated stochastic subgradient methods under local error bound condition. *arXiv preprint arXiv:1607.01027*, 2016.
- Yang, T. and Lin, Q. Rsg: Beating subgradient method without smoothness and strong convexity. *The Journal of Machine Learning Research*, 19(1):236–268, 2018.

- Zhang, H., Zheng, Z., and Lavaei, J. Selection of the best under convexity. *SSRN preprint 3848865*, 2021.
- Zhang, J., Lin, H., Sra, S., and Jadbabaie, A. On complexity of finding stationary points of nonsmooth nonconvex functions. *arXiv preprint arXiv:2002.04130*, 2020.
- Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

## A Literature Review

In contrast to continuous optimization, where problems with the convex structure are studied in depth, most works on discrete stochastic optimization (Futschik & Pflug, 1995; Gutjahr & Pflug, 1996; Futschik & Pflug, 1997; Kleywegt et al., 2002; Semelhago et al., 2020) do not consider the convex structure. The main obstacle to the development of discrete convex optimization lies in the lack of a suitable definition of the discrete convex structure. A natural definition of the discrete convex functions would be functions that are extensible to continuous convex functions. However, for that class of functions, the local optimality does not imply the global optimality and therefore it is not suitable for the purpose of optimization. Later, Favati (1990) proposed a stronger condition, named the integral convexity, which ensures that the local optimality is equivalent to the global optimality. On the other hand, similar properties of continuous convex functions have been proved for submodular functions. Lovász (1983) showed the equivalence between the submodularity of a function and the convexity of its Lovász extension, and later Fujishige (1984) established the Fenchel-type min-max duality theorem for submodular functions. Moreover, the Lovász extension along with the subgradient convinced in Fujishige (2005) provide a good framework of applying gradient-based method to the submodular function minimization (SFM) problem. In the case when the objective function is an integer-valued submodular function, the best strongly and weakly polynomial algorithms proposed in Lee et al. (2015) respectively return the optimal solution with  $O(d^3 \log^2(d))$  and  $O(d^2 \log(Md))$  function value evaluations, where  $d$  is the dimension and  $M$  is the maximal absolute objective function value. The state-of-the-art approximate algorithm designed in (Axelrod et al., 2020) returns an  $\epsilon$ -approximate solution with  $O(d \log(d)/\epsilon^2)$  function value evaluations. Our problem can be viewed as stochastic versions of the problems in Chakrabarty et al. (2017); Axelrod et al. (2020), which also naturally cause the difference in algorithm design and analysis. The efficiency of the algorithms in Chakrabarty et al. (2017); Axelrod et al. (2020) comes from on sampling a stochastic subgradient with  $O(1)$  variance within  $O(1)$  evaluations, which is based on the preprocessing phase (Lemma 3.4 in Axelrod et al. (2020)). However, the sampling phase requires an importance sampling based on the exact function values difference that is computed in the preprocessing phase. It is challenging to extend the importance sampling to the case when we only have access to noisy function values. The major techniques in our gradient-based algorithm is the truncation step, which is proved to reduce the sample complexity by  $O(d)$ . In Bach (2019); Axelrod et al. (2020), the authors extend the domain of submodular function to  $\{1, 2, \dots, N\} \times \dots \times \{1, 2, \dots, N\}$  and show that the generalized submodular function is equivalent to a submodular function on a sub-lattice of  $\mathbb{Z}^{(N-1)d}$ .

Instead of the deterministic SFM problem, the stochastic SFM problem is far from being well-understood. Recently, Ito (2019) provided upper bounds on the computational complexity for expectation error bounds. In addition, the authors of Ito (2019) proved a lower bound of computational complexity for achieving the given precision and showed that the stochastic SFM problem is essentially more difficult than the deterministic SFM problem. The first polynomial-time approximate algorithm to achieve a high-probability error bound is given in Blais et al. (2019). In this work, we also focus on finding solutions with a small error with high probability. We propose algorithms that either have a smaller computational complexity compared to the algorithms in Blais et al. (2019) or do not require information about the objective function. An alternative problem, the stochastic submodular maximization problem, has been studied in depth under the assumption that the objective function is monotone; see Hassani et al. (2017); Karimi et al. (2017); Hassidim & Singer (2017); Mokhtari et al. (2018); Sekar et al. (2021). We note that the approximate solutions to the maximization problem often include a multiplicative error instead of an additive error for the SFM problem. We refer the readers to Ito (2019) for a detailed discussion on related works for the stochastic SFM problem.

In Murota (2003), a generalization of submodular functions, called the  $L^{\natural}$ -convex functions, are defined through the translation submodularity or the discrete midpoint convexity. The  $L^{\natural}$ -convex functions are equivalent to functions that are both submodular and integrally convex on integer lattice. In addition, the  $L^{\natural}$ -convex function has a convex extension that shares similar properties as the Lovász extension and therefore gradient-based methods are also applicable for the  $L^{\natural}$ -convex function minimization problem. We refer the readers to Murota (2003) for more discussion on the discrete convex functions.

Since we are optimizing the Lovász extension of the submodular function without accessing its derivatives, we can view submodular minimization algorithms as zeroth-order convex optimization

algorithms. The idea of approximating gradient using finite difference appeared as early as Nemirovsky & Yudin (1983) and was elaborated in Nesterov & Spokoiny (2017) to derive bounds of zeroth-order methods on smooth and non-smooth convex problems. Later, Duchi et al. (2015) utilized this idea to design the optimal zeroth-order mirror descent method for stochastic convex problems. Furthermore, the RSGF method in Ghadimi & Lan (2013) also incorporates the Gaussian smoothing technique and gives the first bound on stochastic nonconvex smooth problems. In the follow-up work by Balasubramanian & Ghadimi (2018), the zeroth-order conditional gradient method is designed to handle the constrained case. In the same literature, a linear-time zeroth-order estimator of the Hessian matrix was proposed and was used to construct a saddle-point avoiding method. We refer the readers to Homem-de Mello & Bayraksan (2014); Larson et al. (2019) for a review of recent developments of zeroth-order algorithms.

The optimization problems via stochastic objective value evaluations have also been widely studied in the simulation literature. The problem is often called the ranking-and-selection (R&S); see Hong et al. (2020) for a recent review. There have been two approaches to categorize the R&S literature. One approach is differentiating the frequentist view and the Bayesian view; see Kim & Nelson (2006); Chick (2006). The other approach differentiates the fixed-confidence procedures and the fixed-budget procedures; see Frazier et al. (2009); Hunter & Nelson (2017); Hong et al. (2020). In particular, the probability of correct selection (PCS) of the best decision has been a widely used guarantee for both types of procedures. In general, the R&S procedures under fixed-precision and fixed-budget were often classified into the frequentist and Bayesian procedures in the literature. However, there are some exceptions Frazier (2014). A large number of R&S procedures based on the PCS guarantee adopt the indifference zone (IZ) formulation, called PCS-IZ. The IZ parameter is typically assumed to be known, while Fan et al. (2016), as a notable exception, provides selection guarantees without the knowledge of the indifference-zone parameter. In practice, for some problem settings, this IZ parameter may be unknown a priori. This naturally gives rise to a notion of probability of good selection (PGS), which is the optimality guarantee considered in this work. Eckman & Henderson (2018a,b) have thoroughly discussed settings when the use of PGS is preferable compared to the use of PCS-IZ. Generally in the R&S problems, there is no structural information such as convexity that is considered.

The optimization problems considered in this work also belong to the class of discrete optimization via simulation problems, the discussions of which can be found in Fu (2002); Nelson (2010); Sun et al. (2014); Hong et al. (2015); Chen et al. (2018). Discrete optimization via simulation problems naturally arise in many operations research and management science applications, including queueing networks, supply chain networks, sharing economy operations, financial markets, etc.; see Shaked & Shanthikumar (1988); Wolff & Wang (2002); Altman et al. (2003); Singhvi et al. (2015); Jian et al. (2016); Freund et al. (2017) for example. Hu et al. (2007, 2008) have discussed model reference adaptive search algorithms in order to ensure global convergence. Hong & Nelson (2006); Hong et al. (2010); Xu et al. (2010) propose and study algorithms based on the convergent optimization via most-promising-area stochastic search (COMPASS) that can be used to solve general simulation optimization problems with discrete decision variables. The proposed algorithms are computationally efficient and are proved to convergence with probability one to optimal points. Lim (2012) studies simulation optimization problems over multidimensional discrete sets where the objective function adopts multimodularity. They propose algorithms that converge almost surely to the global optimal. Wang et al. (2013) discusses stochastic optimization problems with integer-ordered decision variables. Park & Kim (2015) and Park et al. (2014) develop and examine the Penalty Function with Memory (PFM) method for discrete optimization via simulation with stochastic constraints. Sun et al. (2014) discusses an exploration-exploitation balancing approach using Gaussian process based search. Eckman et al. (2020) discusses a statistically guaranteed screening to rule out decisions based on initial simulation experiments utilizing the convex structure. Sekar et al. (2021) proposes an adaptive line search method that guarantees the convergence to local minima.

## B Preliminaries on $L^h$ -convex functions

In this section, we provide a detailed discussion on the properties of  $L^h$ -convex functions. The following property shows that  $L^h$ -convex functions can be viewed as a generalization of submodular functions.

**Lemma 1** (Murota (2003)). *Suppose that the function  $f(x) : \mathcal{X} \mapsto \mathbb{R}$  is  $L^{\natural}$ -convex. Then, the following translation submodularity holds:*

$$f(x) + f(y) \geq f((x - \alpha \mathbf{1}) \vee y) + f(x \wedge (y + \alpha \mathbf{1})),$$

$$\forall x, y \in \mathcal{X}, \alpha \in \mathbb{N} \quad \text{s.t.} \quad (x - \alpha \mathbf{1}) \vee y, x \wedge (y + \alpha \mathbf{1}) \in \mathcal{X}.$$

By the translation submodularity, the  $L^{\natural}$ -convex function restricted to a cube  $x + \{0, 1\}^d \subset \mathcal{X}$  is a submodular function. Therefore, the Lovász extension (Lovász, 1983) can be constructed as the convex piecewise linear extension inside each cube. In addition,  $L^{\natural}$ -convex functions are integrally convex functions (Murota, 2003). Hence, we can obtain a continuous convex function on  $[1, N]^d$  by piecing together the Lovász extension in each cube. More importantly, we can calculate a subgradient of the convex extension with  $O(d)$  function value evaluations. Hence,  $L^{\natural}$ -convex functions provide a good framework for extending the continuous convex optimization theory to the discrete case. In the remainder of this subsection, we specify this intuition of  $L^{\natural}$ -convex functions in a rigorous way. We first define the Lovász extension of submodular functions and give an explicit subgradient of the Lovász extension at each point.

**Definition 2.** Suppose that  $f(x) : \{0, 1\}^d \mapsto \mathbb{R}$  is a submodular function. For any  $x \in [0, 1]^d$ , we say that a permutation  $\alpha_x : [d] \mapsto [d]$  is a **consistent permutation** of  $x$ , if

$$x_{\alpha_x(1)} \geq x_{\alpha_x(2)} \geq \dots \geq x_{\alpha_x(d)}.$$

For each  $i \in \{0, 1, \dots, d\}$ , the  $i$ -th neighbouring points of  $x$  is defined as

$$S^{x,i} := \sum_{j=1}^i e_{\alpha_x(j)} \in \mathcal{X},$$

where vector  $e_k$  is the  $k$ -th unit vector of  $\mathbb{R}^d$ . We define the **Lovász extension**  $\tilde{f}(x) : [0, 1]^d \mapsto \mathbb{R}$  as

$$\tilde{f}(x) := f(S^{x,0}) + \sum_{i=1}^d [f(S^{x,i}) - f(S^{x,i-1})] x_{\alpha_x(i)}. \quad (4)$$

We note that the value of the Lovász extension does not rely on the choice of the consistent permutation. We list several well-known properties of the Lovász extension and refer their proofs to Lovász (1983); Fujishige (2005).

**Lemma 2.** *The following properties hold for  $\tilde{f}(x)$ :*

- (i) For any  $x \in \mathcal{X}$ , it holds that  $\tilde{f}(x) = f(x)$ .
- (ii) The minimizers of  $\tilde{f}(x)$  satisfy  $\arg \min_{x \in [0,1]^d} \tilde{f}(x) = \arg \min_{x \in \mathcal{X}} f(x)$ .
- (iii) The function  $\tilde{f}(x)$  is a convex function on  $[0, 1]^d$ .
- (iv) A subgradient  $g \in \partial \tilde{f}(x)$  is given by

$$g_{\alpha_x(i)} := f(S^{x,i}) - f(S^{x,i-1}), \quad \forall i \in [d]. \quad (5)$$

Using the expression (5), we construct a subgradient estimator at point  $x$  as

$$\hat{g}_{\alpha_x(i)} := F(S^{x,i}, \xi_i^1) - F(S^{x,i-1}, \xi_{i-1}^2), \quad \forall i \in [d], \quad (6)$$

where  $\xi_i^j$  are mutually independent for  $i \in [d]$  and  $j \in [2]$ . By definition, we know the components of  $\hat{g}$  are mutually independent and the computational complexity of each  $\hat{g}$  is  $2d$ . Using the subgradient defined in (5), we have

$$\mathbb{E}[\hat{g}_{\alpha_x(i)}] = \mathbb{E}[F(S^{x,i}, \xi_i) - F(S^{x,i-1}, \xi_{i-1})] = f(S^{x,i}) - f(S^{x,i-1}) = g_{\alpha_x(i)}, \quad \forall i \in [d],$$

which means that  $\hat{g}$  is an unbiased estimator of  $g$ .

Then, we show that the Lovász extension in the neighborhood of each point can be pieced together to form a convex function on  $\text{conv}(\mathcal{X}) = [1, N]^d$ . We define the local neighborhood of each point  $y \in [1, N - 1]^d$  as the cube

$$\mathcal{C}_y := y + [0, 1]^d,$$

where the Minkowski sum of a point  $y \in \mathbb{R}^d$  and a set  $\mathcal{C} \subset \mathbb{R}^d$  is defined as

$$y + \mathcal{C} := \{y + x \mid x \in \mathcal{C}\}.$$

We denote the objective function  $f(x)$  restricted to  $\mathcal{C}_y \cap \mathcal{X}$  as  $f_y(x)$ , which is submodular by the translation submodularity of  $f(x)$ . For point  $x \in \mathcal{C}_y$ , we denote  $\alpha_x$  as a consistent permutation of  $x - y$  in  $\{0, 1\}^d$  and, for each  $i \in \{0, 1, \dots, d\}$ , the corresponding  $i$ -th neighboring point of  $x$  is defined as

$$S^{x,i} := y + \sum_{j=1}^i e_{\alpha_x(j)}.$$

Then, the Lovász extension of  $f_y(x)$  in  $\mathcal{C}_y$  can be calculated as

$$\tilde{f}_y(x) := f(S^{x,0}) + \sum_{i=1}^d [f(S^{x,i}) - f(S^{x,i-1})] x_{\alpha_x(i)}.$$

Now, we piece together the Lovász extension in each cube by defining

$$\tilde{f}(x) := \tilde{f}_y(x) \quad \forall x \in [1, N]^d, \quad y \in [N-1]^d, \quad \text{s.t. } x \in \mathcal{C}_y. \quad (7)$$

Using the results in Murota (2003), we can prove that  $\tilde{f}(x)$  is well-defined and is a convex function.

**Theorem 8.** *The function  $\tilde{f}(x)$  in (7) is well-defined and is convex on  $\mathcal{X}$ .*

Utilizing properties (i) and (ii) of Lemma 2, we know that problem (1) is equivalent to the relaxed problem

$$f^* := \min_{x \in [1, N]^d} \tilde{f}(x).$$

Moreover, the subgradient (5) and stochastic subgradient (6) are also valid for the convex extension  $\tilde{f}(x)$ . Similarly, (stochastic) subgradients can be computed in the neighboring cube of each point with  $2d$  evaluations to  $F(\cdot, \cdot)$  and it does not matter which cube is chosen for points belonging to multiple cubes.

## B.1 Illustrations of the Lovász extension

In this subsection, we show the Lovász extension of a two-dimensional function on  $[3]^2 = \{1, 2, 3\}^2$ . We consider the quadratic function

$$f(x) := x^T \begin{bmatrix} 0.101 & -0.068 \\ -0.068 & 0.146 \end{bmatrix} x, \quad \forall x \in \mathbb{R}^2.$$

By the results in Murota (2003, Section 7.3), we know the function  $f(\cdot)$  is an  $L^1$ -convex function. We compare the landscapes of the original objective and the Lovász extension in Figure 1. We can see that the Lovász extension is a piecewise linear and convex function, which is consistent with the results in Murota (2003).

## C Proofs in Section 3

### C.1 Proof of Theorem 2

*Proof of Theorem 2.* To gain some intuition, we first consider the deterministic case. Suppose that we already have an  $\epsilon$ -optimal solution to problem (2), i.e., a point  $\bar{x}$  in  $[1, N]^d$  such that  $\tilde{f}(\bar{x}) \leq f^* + \epsilon$ . Then, we rewrite the Lovász extension in (4) as

$$\tilde{f}(\bar{x}) = [1 - \bar{x}_{\alpha_{\bar{x}}(1)}] f(S^{\bar{x},0}) + \sum_{i=1}^{d-1} [\bar{x}_{\alpha_{\bar{x}}(i)} - \bar{x}_{\alpha_{\bar{x}}(i+1)}] f(S^{\bar{x},i}) + \bar{x}_{\alpha_{\bar{x}}(d)} f(S^{\bar{x},d}), \quad (8)$$

which is a convex combination of  $f(S^{\bar{x},0}), \dots, f(S^{\bar{x},d})$ . Hence, there exists an  $\epsilon$ -optimal solution among the neighboring points of  $\bar{x}$ . This means that we can solve a sub-problem with  $d+1$  points to get the  $\epsilon$ -optimal solution among neighboring points.

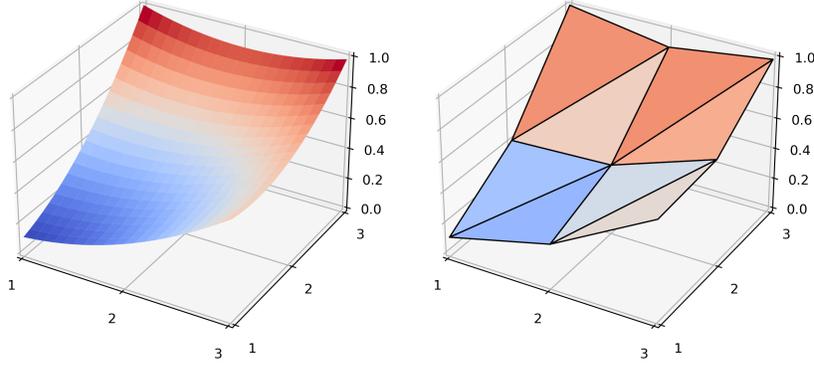


Figure 1: The landscapes of the objective function and its Lovász extension. **Left:** The original objective function. **Right:** The Lovász extension of the objective function.

The proof for the stochastic case is similar. We denote the optimal value of  $f(x)$  as  $f^*$ . Since point  $\bar{x}$  satisfies the  $(\epsilon/2, \delta/2)$ -PAC guarantee, we have

$$\tilde{f}(\bar{x}) - f^* \leq \epsilon/2$$

holds with probability at least  $1 - \delta/2$ . We assume this event happens in the following of this proof. Let  $S^0, S^1, \dots, S^d$  be the neighboring points of  $\bar{x}$ . Using the expression of the Lovász extension in (8), we know there exists an  $\epsilon/2$ -optimal solution among  $S^0, S^1, \dots, S^d$ . We denote the  $\epsilon/2$ -optimal solution and the solution returned by Algorithm 1 as  $S^*$  and  $\hat{S}$ , respectively. By the definition of confidence intervals, we know

$$\left| \hat{F}_n(S_i) - f(S_i) \right| \leq \epsilon/4 \quad \forall i \in \{0, \dots, d\}, \quad \left| \hat{F}_n(\hat{S}) - f(\hat{S}) \right| \leq \epsilon/4$$

holds uniformly with probability at least  $1 - \delta/2$ . Under this event, we know

$$f(\hat{S}) - f^* \leq \hat{F}_n(\hat{S}) - f^* + \epsilon/4 \leq \hat{F}_n(S^*) - f^* + \epsilon/4 \leq f(S^*) - f^* + \epsilon/2 \leq \epsilon,$$

which implies that  $x^* \in \mathcal{X}$  is an  $\epsilon$ -optimal solution and the probability is at least  $1 - \delta/2 - \delta/2 = 1 - \delta$ . Hence, we know  $x^*$  is an  $(\epsilon, \delta)$ -PAC solution to problem (1).

Now, we estimate the computational complexity of Algorithm 1. By the Hoeffding bound, simulating

$$\frac{32}{\epsilon^2} \log \left( \frac{8d}{\delta} \right)$$

times on each neighboring point is enough to achieve  $1 - \delta/(4d)$  confidence half-width  $\epsilon/4$ . Hence, the computational complexity of Algorithm 1 is at most

$$\frac{32(d+1)}{\epsilon^2} \log \left( \frac{8d}{\delta} \right) = O \left[ \frac{d}{\epsilon^2} \log \left( \frac{d}{\delta} \right) \right] = \tilde{O} \left[ \frac{d}{\epsilon^2} \log \left( \frac{1}{\delta} \right) \right].$$

□

## C.2 Proof of Theorem 3

The analysis of Algorithm 2 fits into the classical convex optimization framework. The following Azuma's inequality for martingales with sub-Gaussian tails plays as a major role for deriving high-probability bounds, i.e., the number of required samples to ensure the algorithms succeed with high probability.

**Lemma 3** (Azuma's inequality for sub-Gaussian tails (Shamir, 2011)). *Let  $X_0, \dots, X_{T-1}$  be a martingale difference sequence. Suppose there exist constants  $b_1 \geq 1, b_2 > 0$  such that, for any  $t \in \{0, \dots, T-1\}$ ,*

$$\mathbb{P}(|X_t| \geq a \mid X_1, \dots, X_{t-1}) \leq 2b_1 \exp(-b_2 a^2), \quad \forall a \geq 0. \quad (9)$$

Then for any  $\delta > 0$ , it holds with probability at least  $1 - \delta$  that

$$\frac{1}{T} \sum_{t=0}^{T-1} X_t \leq \sqrt{\frac{28b_1}{b_2T} \log\left(\frac{1}{\delta}\right)}.$$

Since the stochastic subgradient  $\hat{g}^t$  is truncated, the stochastic subgradient used for updating, namely  $\tilde{g}^t$ , is not unbiased. We define the bias at each step as

$$b_t := \mathbb{E} [\tilde{g}^t \mid x^0, x^1, \dots, x^t] - g^t, \quad \forall t \in \{0, 1, \dots, T-1\}.$$

First, we bound the  $\ell_1$ -norm of the bias.

**Lemma 4.** *Suppose Assumptions 1-4 hold. If we have*

$$M \geq 2\sigma \cdot \sqrt{\log\left(\frac{8\sigma dT}{\epsilon}\right)} = \Theta \left[ \sqrt{\log\left(\frac{dT}{\epsilon}\right)} \right], \quad T \geq \frac{2\epsilon}{\sigma},$$

then it holds

$$\|b^t\|_1 \leq \frac{\epsilon}{2T}, \quad \forall t \in \{0, 1, \dots, T-1\}.$$

*Proof.* Let  $\alpha_t$  be a consistent permutation of  $x^t$  and  $S^{t,i}$  be the corresponding  $i$ -th neighboring points in the neighbourhood of  $x^t$ . We only need to prove

$$\left| b_{\alpha_t(i)}^t \right| \leq \frac{\epsilon}{2dT}, \quad \forall i \in [d].$$

We define two random variables

$$Y_1 := F(S^{t,i}, \xi_i^1) - f(S^{t,i}), \quad Y_2 := F(S^{t,i-1}, \xi_{i-1}^2) - f(S^{t,i-1}).$$

By Assumption 1, both  $Y_1$  and  $Y_2$  are independent and sub-Gaussian with parameter  $\sigma^2$ . Hence, we know

$$\begin{aligned} b_{\alpha_t(i)}^t &= \mathbb{E} \left[ \tilde{g}_{\alpha_t(i)}^t - g_{\alpha_t(i)}^t \right] = \mathbb{E} [(Y_1 + Y_2) \cdot \mathbf{1}_{-M \leq Y_1 + Y_2 \leq M}] + \mathbb{E} [M \cdot \mathbf{1}_{Y_1 + Y_2 > M}] + \mathbb{E} [-M \cdot \mathbf{1}_{Y_1 + Y_2 < -M}] \\ &= \mathbb{E} [(M - Y_1 - Y_2) \cdot \mathbf{1}_{Y_1 + Y_2 > M}] + \mathbb{E} [-(M + Y_1 + Y_2) \cdot \mathbf{1}_{Y_1 + Y_2 < -M}], \end{aligned}$$

where the second step is from  $\mathbb{E}[Y_1] = \mathbb{E}[Y_2] = 0$ . Taking the absolute value on both sides, we get

$$\begin{aligned} \left| b_{\alpha_t(i)}^t \right| &\leq \mathbb{E} [(Y_1 + Y_2 - M) \cdot \mathbf{1}_{Y_1 + Y_2 > M}] + \mathbb{E} [-(M + Y_1 + Y_2) \cdot \mathbf{1}_{Y_1 + Y_2 < -M}] \quad (10) \\ &= \mathbb{E} [(Y - M) \cdot \mathbf{1}_{Y > M}] + \mathbb{E} [-(Y + M) \cdot \mathbf{1}_{Y < -M}], \end{aligned}$$

where we define the random variable  $Y := Y_1 + Y_2$ . Since  $Y_1, Y_2$  are independent, random variable  $Y$  is sub-Gaussian with parameter  $2\sigma^2$ . Let  $F(y) := \mathbb{P}[Y \leq y]$  be the distribution function of  $Y$ . Then, we have

$$\mathbb{E} [(Y - M) \cdot \mathbf{1}_{Y > M}] = \int_M^\infty (y - M) dF(y) = \int_M^\infty (1 - F(y)) dy. \quad (11)$$

By the Hoeffding bound, we know

$$1 - F(y) = \mathbb{P}[Y > y] \leq \exp(-y^2/4\sigma^2), \quad \forall y \geq 0.$$

Using the upper bound for  $Q$ -function in Borjesson & Sundberg (1979), it holds that

$$\int_M^\infty 1 - F(y) dy \leq \int_M^\infty \exp(-y^2/4\sigma^2) dy \leq \frac{2\sigma^2}{M} \exp\left(-\frac{M^2}{4\sigma^2}\right).$$

By the choice of  $M$ , we know

$$M \geq 2\sigma \sqrt{\log(8d)} \geq 2\sigma \quad \text{and} \quad \sigma \exp(-M^2/4\sigma^2) \leq \frac{\epsilon}{4dT}.$$

which implies that

$$\int_M^\infty 1 - F(y) dy \leq \frac{2\sigma^2}{M} \exp(-M^2/4\sigma^2) \leq \frac{\epsilon}{4dT}.$$

Substituting the above inequality into (11), we have

$$\mathbb{E}[(Y - M) \cdot \mathbf{1}_{Y > M}] \leq \frac{\epsilon}{4dT}.$$

Considering  $-Y$  in the same way, we can prove

$$\mathbb{E}[-(Y + M) \cdot \mathbf{1}_{Y < -M}] \leq \frac{\epsilon}{4dT}.$$

Substituting the last two estimates into inequality (10), we know

$$\left| b_{\alpha_t(i)}^t \right| \leq \frac{\epsilon}{2dT}.$$

□

Next, we show that  $\langle g^t + b^t - \tilde{g}^t, x^t - x^* \rangle$  forms a martingale sequence and use Azuma's inequality to bound the deviation, where  $x^*$  is a minimizer of  $f(x)$ .

**Lemma 5.** *Suppose Assumptions 1-4 hold and let  $x^*$  be a minimizer of  $f(x)$ . The sequence*

$$X_t := \langle g^t + b^t - \tilde{g}^t, x^t - x^* \rangle \quad t = 0, 1, \dots, T-1$$

*forms a martingale difference sequence. Furthermore, if we have*

$$M = \max \left\{ L, 2\sigma \cdot \sqrt{\log\left(\frac{4\sigma dNT}{\epsilon}\right)} \right\} = \tilde{\Theta} \left[ \sqrt{\log\left(\frac{dNT}{\epsilon}\right)} \right], \quad T \geq \frac{2\epsilon}{\sigma},$$

*then it holds*

$$\frac{1}{T} \sum_{t=0}^{T-1} X_t \leq \sqrt{\frac{224dN^2\sigma^2}{T} \log\left(\frac{1}{\delta}\right)}$$

*with probability at least  $1 - \delta$ .*

*Proof.* Let  $\mathcal{F}_t$  be the filtration generated by  $x_0, x_1, \dots, x_t$ . By the definition of  $b^t$ , we know

$$\mathbb{E}[g^t + b^t - \tilde{g}^t \mid \mathcal{F}_t] = 0,$$

which implies that

$$\mathbb{E}[X_t \mid \mathcal{F}_t] = \langle \mathbb{E}[g^t + b^t - \tilde{g}^t \mid \mathcal{F}_t], x^t - x^* \rangle = 0.$$

Hence, the sequence  $\{X_t\}$  is a martingale difference sequence. Next, we estimate the probability  $\mathbb{P}[|X_t| \geq a \mid \mathcal{F}_t]$ . We have the bound

$$\begin{aligned} |X_t| &= |\langle g^t + b^t - \tilde{g}^t, x^t - x^* \rangle| \leq \|g^t + b^t - \tilde{g}^t\|_1 \|x^t - x^*\|_\infty \\ &\leq N \|g^t + b^t - \tilde{g}^t\|_1 \leq N \|g^t - \tilde{g}^t\|_1 + N \|b^t\|_1. \end{aligned}$$

Since  $M$  satisfies the condition in Lemma 4, we know  $\|b^t\|_1 \leq \epsilon/2T$ . Recalling Assumption 4, we get  $|g_i^t| \leq L$  for all  $i \in [d]$ . By the truncation rule and the assumption  $M \geq L$ , we have

$$|\tilde{g}_i^t - g_i^t| = |(\hat{g}_i^t \wedge M) \vee (-M) - g_i^t| \leq |\hat{g}_i^t - g_i^t|, \quad \forall i \in [d].$$

Hence, we get

$$|X_t| \leq \frac{N\epsilon}{2T} + N \|\hat{g}^t - g^t\|_1. \quad (12)$$

Define random variables  $Y_i := |\hat{g}_i^t - g_i^t|$  for all  $i \in [d]$ . By Assumption 1,  $Y_i$  is sub-Gaussian with parameter  $\sigma^2$ . Hence, we have

$$Y := \|\hat{g}^t - g^t\|_1 = \sum_{i=1}^d Y_i$$

is sub-Gaussian with parameter  $d\sigma^2$ . First, we consider the case when  $a/N \geq \epsilon/T$ . Using inequality (12), it follows that

$$\mathbb{P}[|X_t| \geq a \mid \mathcal{F}_t] \leq \mathbb{P}\left[\frac{\epsilon}{2T} + Y \geq \frac{a}{N}\right] \leq \mathbb{P}\left[Y \geq \frac{a}{N} - \frac{\epsilon}{2T}\right]$$

$$\leq \mathbb{P} \left[ Y \geq \frac{a}{2N} \right] \leq 2 \exp \left( -\frac{a^2}{8dN^2\sigma^2} \right), \quad (13)$$

where the last inequality is from Hoeffding bound. In this case, we know condition (9) holds with

$$b_1 = 1, \quad b_2 = \frac{1}{8dN^2\sigma^2}.$$

Now, we consider the case when  $a/N < \epsilon/T$ . In this case, by the assumption that  $T \geq 2\epsilon/\sigma$ , we have

$$2b_1 \exp(-b_2 a^2) > 2 \exp \left( -\frac{1}{8dN^2\sigma^2} \cdot \frac{\epsilon^2}{T^2} \right) \geq 2 \exp \left( -\frac{1}{32d} \right) \geq 2 \exp \left( -\frac{1}{32} \right) > 1.$$

Hence, it holds

$$\mathbb{P} [|X_t| \geq a \mid \mathcal{F}_\sigma] \leq 1 < 2b_1 \exp(-b_2 a^2).$$

Combining with inequality (13), we know condition (9) holds with  $b$  and  $c$  defined above. Using Lemma 3, we know

$$\frac{1}{T} \sum_{t=0}^{T-1} X_t \leq \sqrt{\frac{224dN^2\sigma^2}{T} \log\left(\frac{1}{\delta}\right)}$$

holds with probability at least  $1 - \delta$ .  $\square$

Then, we prove a lemma similar to the Lemma in Zinkevich (2003).

**Lemma 6.** *Suppose Assumptions 1-4 hold and let  $x^*$  be a minimizer of  $f(x)$ . If we choose*

$$\eta = \frac{N}{M\sqrt{T}},$$

then we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \langle \tilde{g}^t, x^t - x^* \rangle \leq \frac{dNM}{\sqrt{T}}.$$

*Proof.* We define  $\tilde{x}^{t+1} := x^t - \eta \tilde{g}^t$  as the next point before the projection onto  $[1, N]^d$ . Recalling the non-expansion property of orthogonal projection, we get

$$\begin{aligned} \|x^{t+1} - x^*\|_2^2 &= \|\mathcal{P}(\tilde{x}^{t+1} - x^*)\|_2^2 \leq \|\tilde{x}^{t+1} - x^*\|_2^2 = \|x^t - x^* - \eta \tilde{g}^t\|_2^2 \\ &= \|x^t - x^*\|_2^2 + \eta^2 \|\tilde{g}^t\|_2^2 - 2\eta \langle \tilde{g}^t, x^t - x^* \rangle, \end{aligned}$$

and equivalently,

$$\langle \tilde{g}^t, x^t - x^* \rangle = \frac{1}{2\eta} \left[ \|x^t - x^*\|_2^2 - \|x^{t+1} - x^*\|_2^2 \right] + \frac{\eta}{2} \cdot \|\tilde{g}^t\|_2^2.$$

Summing over  $t = 0, 1, \dots, T-1$ , we have

$$\begin{aligned} \sum_{t=0}^{T-1} \langle \tilde{g}^t, x^t - x^* \rangle &= \frac{\|x^0 - x^*\|_2^2 - \|x^T - x^*\|_2^2}{2\eta} + \frac{\eta}{2} \sum_{t=0}^{T-1} \|\tilde{g}^t\|_2^2 \\ &\leq \frac{d\|x^0 - x^*\|_\infty^2}{2\eta} + \frac{\eta}{2} \sum_{t=0}^{T-1} \|\tilde{g}^t\|_2^2 \leq \frac{dN^2}{2\eta} + \frac{\eta}{2} \sum_{t=0}^{T-1} \|\tilde{g}^t\|_2^2. \end{aligned}$$

By the definition of truncation, it follows that  $\|\tilde{g}^t\|_2^2 \leq dM^2$ . Choosing

$$\eta := \frac{N}{M\sqrt{T}},$$

it follows that

$$\sum_{t=0}^{T-1} \langle \tilde{g}^t, x^t - x^* \rangle \leq \frac{dN^2}{2\eta} + \frac{\eta}{2} \sum_{t=0}^{T-1} \|\tilde{g}^t\|_2^2 \leq \frac{dN^2}{2\eta} + \frac{\eta T d M^2}{2} = dNM\sqrt{T}.$$

$\square$

Finally, using Lemmas 4, 5 and 6, we can finish the proof of Theorem 3.

*Proof of Theorem 3.* Denote  $f^*$  as the optimal value of  $\tilde{f}(x)$ . Using the convexity of  $\tilde{f}(x)$ , we know

$$\begin{aligned}\tilde{f}(\bar{x}) - f^* &\leq \frac{1}{T} \sum_{t=0}^{T-1} [\tilde{f}(x^t) - f^*] \leq \frac{1}{T} \sum_{t=0}^{T-1} \langle g^t, x^t - x^* \rangle \\ &= \frac{1}{T} \sum_{t=0}^{T-1} [\langle g^t + b^t - \tilde{g}^t, x^t - x^* \rangle + \langle \tilde{g}^t, x^t - x^* \rangle - \langle b^t, x^t - x^* \rangle].\end{aligned}\quad (14)$$

We choose

$$T := \frac{3584dN^2\sigma^2}{\epsilon^2} \log\left(\frac{2}{\delta}\right) = \Theta\left[\frac{dN^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right].$$

Recalling Assumption 1, we know  $\delta$  is small enough and therefore we have the following estimates:

$$L^2 \leq M^2 = \tilde{\Theta}\left[\log\left(\frac{dT}{\epsilon}\right)\right] = \tilde{O}\left[\log\left(\frac{d^2N^2}{\epsilon^3}\right) + \log\log\left(\frac{1}{\delta}\right)\right] \leq \frac{\epsilon^2 T}{64d^2}, \quad T \geq \max\left\{\frac{2\epsilon}{\sigma}, 4\right\}.$$

Hence, the conditions in Lemmas 4 and 5 are satisfied. By Lemma 4, we know

$$-\frac{1}{T} \sum_{t=0}^{T-1} \langle b^t, x^t - x^* \rangle \leq \frac{1}{T} \sum_{t=0}^{T-1} \|b^t\|_1 \|x^t - x^*\|_\infty \leq \frac{\epsilon}{2T} \leq \frac{\epsilon}{8}.\quad (15)$$

By Lemma 5, it holds

$$\frac{1}{T} \sum_{t=0}^{T-1} \langle g^t + b^t - \tilde{g}^t, x^t - x^* \rangle \leq \sqrt{\frac{224dN^2\sigma^2}{T} \log\left(\frac{2}{\delta}\right)} \leq \frac{\epsilon}{4}\quad (16)$$

with probability at least  $1 - \delta$ , where the last inequality is from our choice of  $T$ . By Lemma 6, we know

$$\frac{1}{T} \sum_{t=0}^{T-1} \langle \tilde{g}^t, x^t - x^* \rangle \leq \frac{dNM}{\sqrt{T}} \leq \frac{\epsilon}{8}.\quad (17)$$

Substituting inequalities (15), (16) and (17) into inequality (14), we get

$$\tilde{f}(\bar{x}) - f^* \leq \frac{\epsilon}{2}$$

holds with probability at least  $1 - \delta/2$ . By the results of Theorem 2, we know Algorithm 2 returns an  $(\epsilon, \delta)$ -PAC solution.

Finally, we estimate the computational complexity of Algorithm 2. For each iteration, we need to generate a stochastic subgradient using (6) and the computational complexity is  $2d$ . Hence, the total computational complexity of all iterations is

$$2d \cdot T = \tilde{\Theta}\left[\frac{d^2N^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right].$$

By Theorem 2, the computational complexity of rounding process is at most

$$\tilde{O}\left[\frac{d}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right].$$

Thus, we know the total computational complexity of Algorithm 2 is at most

$$\tilde{O}\left[\frac{d^2N^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right)\right].$$

□

### C.3 Details of examples where Assumption 5 holds

**Example 3.** We consider the case when the randomness of each choice of decision variables shares the same measure space, i.e., there exists a measure space  $(Z, \mathcal{B}_Z)$  such that  $\xi_x$  can be any element in the measure space for all  $x \in \mathcal{X}$ . Moreover, for any fixed  $\xi \in \mathcal{B}$ , the function  $F(\cdot, \xi)$  is also  $L^\sharp$ -convex (or submodular when  $N = 2$ ) and has  $\ell_\infty$ -Lipschitz constant  $\tilde{L}$ . Then, we consider the subgradient estimator

$$\hat{g}_{\alpha_x(i)} := F(S^{x,i}, \xi) - F(S^{x,i-1}, \xi) \quad \forall i \in [d]. \quad (18)$$

The computational complexity of estimator (18) is  $d + 1$ . In addition, property (v) of Lemma 2 gives

$$\|\hat{g}\|_1 \leq 3\tilde{L}/2.$$

Therefore, in this situation, the Assumption 5 holds with  $G = 3\tilde{L}/2$  and  $\beta = d + 1$ .

When the distribution at each choice of decision variables is bounded almost surely, we show that Assumption 5 also holds.

**Example 4.** We consider the case when the distribution at each point  $x \in \mathcal{X}$  is bounded, namely, there exists a constant  $C \geq 0$  such that

$$\mathbb{P}[|F(x, \xi_x)| \leq C] = 1, \quad \forall x \in \mathcal{X}.$$

We note that a bounded distribution is a special case of sub-Gaussian distributions. In this case, the  $\ell_\infty$ -Lipschitz constant is  $2C$  and property (v) in Lemma 2 gives  $\|g\|_1 \leq 3C$  for any subgradient  $g$ . We consider the subgradient estimator (6). At point  $x$ , if index  $i$  is chosen, then we know that

$$\|\hat{g}\|_1 = d \cdot |F(S^{x,i}, \xi_i^1) - F(S^{x,i-1}, \xi_{i-1}^2)| \leq 2dC.$$

Hence, Assumption 5 holds with  $G = 2dC$  and  $\beta = 2$ .

### C.4 Proof of Theorem 4

Since the stochastic gradient is bounded, we apply the Azuma's inequality for martingale difference sequences with bounded tails.

**Lemma 7** (Azuma's inequality with bounded tails). *Let  $X_0, \dots, X_{T-1}$  be a martingale difference sequence. Suppose there exists a constant  $b$  such that for any  $t \in \{0, \dots, T-1\}$ ,*

$$\mathbb{P}(|X_t| \leq b) = 1.$$

*Then for any  $\delta > 0$ , it holds with probability at least  $1 - \delta$  that*

$$\frac{1}{T} \sum_{t=0}^{T-1} X_t \leq b \sqrt{\frac{2}{T} \log\left(\frac{1}{\delta}\right)}. \quad (19)$$

The proof follows a similar way as Theorem 3. We first bound the noise term by Azuma's inequality.

**Lemma 8.** *Suppose Assumptions 1-5 hold and let  $x^*$  be a minimizer of  $f(x)$ . Then, it holds*

$$\frac{1}{T} \sum_{t=0}^{T-1} \langle g^t - \hat{g}^t, x^t - x^* \rangle \leq N \left( \frac{3L}{2} + G \right) \sqrt{\frac{2}{T} \log\left(\frac{1}{\delta}\right)}$$

*with probability at least  $1 - \delta$ .*

*Proof.* Same as the proof of Lemma 5, the fact that  $\hat{g}^t$  is unbiased implies that

$$X_t := \langle g^t - \hat{g}^t, x^t - x^* \rangle \quad t = 0, 1, \dots, T-1$$

is a martingale difference sequence. By Assumption 5 and property (v) in Lemma 2, we know

$$|X_t| = |\langle g^t - \hat{g}^t, x^t - x^* \rangle| \leq \|g^t - \hat{g}^t\|_1 \|x^t - x^*\|_\infty \leq N \|g^t - \hat{g}^t\|_1 \leq N(3L/2 + G),$$

which implies that the condition (19) holds with  $b = N(3L/2 + G)$ . Using Lemma 7, we get the conclusion of this lemma.  $\square$

The following lemma bounds the error of the algorithm and is similar to Theorem 3.2.2 in Nesterov (2018).

**Lemma 9.** *Suppose Assumptions 1-5 hold and let  $x^*$  be a minimizer of  $f(x)$ . If we choose*

$$\eta = \sqrt{\frac{dN^2}{TG^2}},$$

then we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \langle \hat{g}^t, x^t - x^* \rangle \leq \sqrt{\frac{dN^2 G^2}{T}}.$$

*Proof.* We define  $\tilde{x}^{t+1} := x^t - \eta \hat{g}^t$  as the next point before the projection onto  $[1, N]^d$ . Recalling the non-expansion property of orthogonal projection, we get

$$\begin{aligned} \|x^{t+1} - x^*\|_2^2 &= \|\mathcal{P}(\tilde{x}^{t+1} - x^*)\|_2^2 \leq \|\tilde{x}^{t+1} - x^*\|_2^2 = \|x^t - x^* - \eta \hat{g}^t\|_2^2 \\ &= \|x^t - x^*\|_2^2 + \eta^2 \|\hat{g}^t\|_2^2 - 2\eta \langle \hat{g}^t, x^t - x^* \rangle, \end{aligned}$$

and equivalently,

$$\langle \hat{g}^t, x^t - x^* \rangle = \frac{1}{2\eta} \left[ \|x^t - x^*\|_2^2 - \|x^{t+1} - x^*\|_2^2 \right] + \frac{\eta}{2} \cdot \|\hat{g}^t\|_2^2.$$

Using Assumption 5, we know  $\|\hat{g}^t\|_2^2 \leq \|\hat{g}^t\|_1^2 \leq G^2$  and therefore

$$\langle \hat{g}^t, x^t - x^* \rangle = \frac{1}{2\eta} \left[ \|x^t - x^*\|_2^2 - \|x^{t+1} - x^*\|_2^2 \right] + \frac{\eta G^2}{2}.$$

Summing over  $t = 0, 1, \dots, T-1$ , we have

$$\begin{aligned} \sum_{t=0}^{T-1} \langle \hat{g}^t, x^t - x^* \rangle &= \frac{\|x^0 - x^*\|_2^2 - \|x^T - x^*\|_2^2}{2\eta} + T \cdot \frac{\eta G^2}{2} \\ &\leq \frac{d \|x^0 - x^*\|_\infty^2}{2\eta} + \frac{\eta T G^2}{2} \leq \frac{dN^2}{2\eta} + \frac{\eta T G^2}{2}. \end{aligned}$$

Choosing

$$\eta := \sqrt{\frac{dN^2}{TG^2}},$$

it follows that

$$\sum_{t=0}^{T-1} \langle \hat{g}^t, x^t - x^* \rangle \leq NG\sqrt{dT}.$$

□

Now, we prove Theorem 4 using Lemmas 8 and 9.

*Proof of Theorem 4.* According to the proof of Theorem 3, we have

$$\begin{aligned} \tilde{f}(\bar{x}) - f^* &\leq \frac{1}{T} \sum_{t=0}^{T-1} [\tilde{f}(x^t) - f^*] \leq \frac{1}{T} \sum_{t=0}^{T-1} \langle g^t, x^t - x^* \rangle \\ &= \frac{1}{T} \sum_{t=0}^{T-1} \langle \hat{g}^t, x^t - x^* \rangle + \frac{1}{T} \sum_{t=0}^{T-1} \langle g^t - \hat{g}^t, x^t - x^* \rangle. \end{aligned} \tag{20}$$

By Lemmas 8 and 9, it holds

$$\frac{1}{T} \sum_{t=0}^{T-1} \langle \hat{g}^t, x^t - x^* \rangle \leq N \left( \frac{3L}{2} + G \right) \sqrt{\frac{2}{T} \log\left(\frac{2}{\delta}\right)}, \quad \frac{1}{T} \sum_{t=0}^{T-1} \langle g^t - \hat{g}^t, x^t - x^* \rangle \leq \sqrt{\frac{dN^2 G^2}{T}}$$

with probability at least  $1 - \delta/2$ . Choosing

$$T = N^2 \left( \frac{3L}{2} + G \right)^2 \cdot \frac{32}{\epsilon^2} \log\left(\frac{2}{\delta}\right) = \Theta \left[ \frac{N^2(L+G)^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right],$$

we know

$$T \geq \frac{16dN^2G^2}{\epsilon^2}$$

when  $\delta$  is small enough. Hence, we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \langle \hat{g}^t, x^t - x^* \rangle \leq \frac{\epsilon}{4}, \quad \frac{1}{T} \sum_{t=0}^{T-1} \langle g^t - \hat{g}^t, x^t - x^* \rangle \leq \frac{\epsilon}{4}$$

holds with probability at least  $1 - \delta/2$ . Substituting into inequality (20), we have

$$\tilde{f}(\bar{x}) - f^* \leq \frac{\epsilon}{2}$$

holds with probability at least  $1 - \delta/2$ . By the results of Theorem 2, we know Algorithm 2 returns an  $(\epsilon, \delta)$ -PAC solution.

Finally, we estimate the computational complexity of Algorithm 2. For each iteration, the computational complexity is decided by the generation of a stochastic subgradient, which is at most  $\beta$  by Assumption 5. Hence, the total computational complexity of all iterations is

$$O[\beta T] = \tilde{O} \left[ \frac{\beta N^2(L+G)^2}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

By Theorem 2, the computational complexity of rounding process is at most

$$\tilde{O} \left[ \frac{d}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

Thus, we know the total computational complexity of Algorithm 2 is at most

$$\tilde{O} \left[ \frac{\beta N^2(L+G)^2 + d}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

□

### C.5 Algorithms for the known sub-optimality gap case

In this section, we consider the case when the global optimum is unique and the sub-optimality is known to be at least  $c > 0$ . In this case, it suffices to find a  $(c, \delta)$ -PAC solution. We first prove that the existence of sub-optimality is equivalent to the so-called weak sharp minima condition of the convex extension. In addition, we use the  $\ell_\infty$  norm in place of the  $\ell_2$  norm since the feasible set is a hypercube.

**Definition 3.** We say that a function  $f(x) : \mathcal{X} \mapsto \mathbb{R}$  satisfies the **Weak Sharp Minimum (WSM)** condition, if the function  $f(x)$  has a unique minimizer  $x^*$  and there exists a constant  $\kappa > 0$  such that

$$\|x - x^*\|_\infty \leq \kappa (f(x) - f^*), \quad \forall x \in \mathcal{X},$$

where  $f^* := f(x^*)$ .

The WSM condition was first defined in Burke & Ferris (1993), and is also called the polyhedral error bound condition in recent literature (Yang & Lin, 2018). In addition, the WSM condition is a special case of the global growth condition in Xu et al. (2016) with  $\theta = 1$ . The WSM condition can be used to leverage the distance between intermediate solutions and  $(c, \delta)$ -PAC solutions. Given a constant  $c > 0$ , we use  $\mathcal{MC}_c$  to denote the set of  $L^{\natural}$ -convex models with sub-optimality gap at least  $c$ . The next theorem verifies that the WSM condition is equivalent to the existence of sub-optimality gap  $c$ .

**Theorem 9.** Suppose that function  $f(x) : \mathcal{X} \mapsto \mathbb{R}$  is an  $L^{\natural}$ -convex function and  $\tilde{f}(x)$  is its Lovász extension on  $[1, N]^d$ . For any constant  $c > 0$ , function  $f(x) \in \mathcal{MC}_c$  if and only if  $\tilde{f}(x)$  satisfies the WSM condition with  $\kappa = c^{-1}$ .

*Proof.* We first prove the sufficiency part and then consider the necessity part.

**Sufficiency.** Suppose there exists a constant  $\kappa > 0$  such that the function  $\tilde{f}(x)$  satisfies the WSM condition with  $\kappa$ . Considering any point  $x \in \mathcal{X} \setminus \{x^*\}$ , we know  $\|x - x^*\|_\infty \geq 1$  and, by the WSM condition,

$$f(x) - f^* = \tilde{f}(x) - f^* \geq \kappa^{-1} \|x - x^*\|_\infty \geq \kappa^{-1}.$$

Thus, we know the sub-optimality gap for  $f(x)$  is at least  $\kappa^{-1}$  and  $f(x) \in \mathcal{MC}_{\kappa^{-1}}$ .

**Necessity.** Suppose there exists a constant  $c > 0$  such that

$$f(x) - f^* \geq c \quad \forall x \in \mathcal{X} \setminus \{x^*\}.$$

We first consider point  $x \in [1, N]^d$  such that  $\|x - x^*\|_\infty \leq 1$ . In this case, we know there exists a hypercube  $\mathcal{C}_y$  containing both  $x$  and  $x^*$ . By the definition of Lovász extension, we know that

$$\tilde{f}(x) = [1 - x_{\alpha_x(1)}]f(S^{x,0}) + \sum_{i=1}^{d-1} [x_{\alpha_x(i)} - x_{\alpha_x(i+1)}]f(S^{x,i}) + x_{\alpha_x(d)}f(S^{x,d}) = \sum_{i=0}^d \lambda_i f(S^{x,i}),$$

where we define

$$\lambda_i := x_{\alpha_x(i)} - x_{\alpha_x(i+1)} \quad \forall i \in [d-1], \quad \lambda_0 := 1 - x_{\alpha_x(1)}, \quad \lambda_d := x_{\alpha_x(d)}.$$

Recalling the definition of consistent permutation, we get

$$\sum_{i=0}^d \lambda_i = 1, \quad \lambda_i \geq 0, \quad \forall i \in \{0, \dots, d\}$$

and  $\tilde{f}(x)$  is a convex combination of  $f(S^{x,0}), \dots, f(S^{x,d})$ . In addition, we can calculate that

$$\left( \sum_{i=0}^d \lambda_i S^{x,i} \right)_{\alpha_x(k)} = \sum_{i=0}^d \lambda_i \cdot S_{\alpha_x(k)}^{x,i} = \sum_{i=0}^d \lambda_i \cdot \mathbf{1}(i \leq k) = \sum_{i=k}^d \lambda_i = x_{\alpha_x(k)},$$

which implies that

$$x = \sum_{i=0}^d \lambda_i S^{x,i}.$$

If  $x^* \notin \{S^{x,0}, \dots, S^{x,d}\}$ , the assumption that the sub-optimality gap is at least  $c$  gives

$$\tilde{f}(x) - f^* = \sum_{i=0}^d \lambda_i [f(S^{x,i}) - f^*] \geq \sum_{i=0}^d \lambda_i \cdot c = c.$$

Combining with  $\|x - x^*\|_\infty \leq 1$ , we have

$$\|x - x^*\|_\infty \leq c^{-1} \cdot [\tilde{f}(x) - f^*].$$

Otherwise if  $x^* = S^{x,i}$  for some  $i \in \{0, \dots, d\}$ . Then, we know

$$\tilde{f}(x) - f^* = \sum_{i=0}^d \lambda_i [f(S^{x,i}) - f^*] \geq \sum_{i \neq k} \lambda_i \cdot c = (1 - \lambda_k)c$$

and

$$\begin{aligned} \|x - x^*\|_\infty &= \left\| \sum_{i=0}^d \lambda_i S^{x,i} - x^* \right\|_\infty = \left\| \sum_{i=0}^d \lambda_i (S^{x,i} - x^*) \right\|_\infty = \left\| \sum_{i \neq k} \lambda_i (S^{x,i} - x^*) \right\|_\infty \\ &\leq \sum_{i \neq k} \lambda_i \|S^{x,i} - x^*\|_\infty \leq \sum_{i \neq k} \lambda_i = 1 - \lambda_k, \end{aligned}$$

where the last inequality is because  $S^{x,i}$  and  $x^*$  are in the same hypercube  $\mathcal{C}_y$ . Combining the above two inequalities, it follows that

$$\|x - x^*\|_2 \leq c^{-1} \cdot [\tilde{f}(x) - f^*],$$

---

**Algorithm 3** Adaptive truncated SSGD method for the PAC guarantee
 

---

**Input:** Model  $\mathcal{X}, \mathcal{B}_{\mathcal{Y}}, F(x, \xi_x)$ , optimality guarantee parameter  $\delta$ , indifference zone parameter  $c$ .

**Output:** An  $(c, \delta)$ -PAC solution  $x^*$  to problem (1).

- 1: Set the initial guarantee  $\epsilon_0 \leftarrow cN/4$ .
  - 2: Set the number of epochs  $E \leftarrow \lceil \log_2(N) \rceil + 1$ .
  - 3: Set the initial search space  $\mathcal{Y}_0 \leftarrow [1, N]^d$ .
  - 4: **for**  $e = 0, \dots, E - 1$  **do**
  - 5:     Use Algorithm 2 to get an  $(\epsilon_e, \delta/(2E))$ -PAC solution  $x_e$  in  $\mathcal{Y}_e$ .
  - 6:     Update guarantee  $\epsilon_{e+1} \leftarrow \epsilon_e/2$ .
  - 7:     Update the search space  $\mathcal{Y}_{e+1} \leftarrow \mathcal{N}(x_e, 2^{-e-2}N)$ .
  - 8: **end for**
  - 9: Round  $x_{E-1}$  to an integral point satisfying the  $(c, \delta)$ -PAC guarantee by Algorithm 1.
- 

which means that the WSM condition holds with  $\kappa = c^{-1}$ . Now we consider point  $x \in [1, N]^d$  such that  $\|x - x^*\|_{\infty} \geq 1$ . We define

$$\tilde{x} := x^* + \frac{x - x^*}{\|x - x^*\|_{\infty}}$$

to be the point on the segment  $\overline{xx^*}$  such that  $\|\tilde{x} - x^*\|_{\infty} = 1$ . By the convexity of  $\tilde{f}(x)$  and the WSM condition for point  $\tilde{x}$ , we know

$$\tilde{f}(x) - f^* \geq \frac{\|x - x^*\|_{\infty}}{\|\tilde{x} - x^*\|_{\infty}} [\tilde{f}(\tilde{x}) - f^*] = \frac{\tilde{f}(\tilde{x}) - f^*}{\|\tilde{x} - x^*\|_{\infty}} \cdot \|x - x^*\|_{\infty} \geq c^{-1} \cdot \|x - x^*\|_{\infty},$$

which shows that the WSM condition holds with  $\kappa = c^{-1}$ . Hence, the WSM condition holds for all points in  $[1, N]^d$  with  $\kappa = c^{-1}$ .  $\square$

Using the WSM condition, we can accelerate Algorithm 2 by dynamically shrinking the search space. To describe the shrinkage of search space, we define the  $\ell_{\infty}$ -neighbourhood of point  $x$  as

$$\mathcal{N}(x, a) := \{y \in [1, N]^d : \|y - x\|_{\infty} \leq a\}$$

and the orthogonal projection onto  $\mathcal{N}(x, a)$  as

$$\mathcal{P}_{x,a}(y) := (y \wedge (x + a)\mathbf{1}) \vee (x - a)\mathbf{1}, \quad \forall x \in \mathbb{R}^d.$$

Now we give the adaptive truncated SSGD algorithm for the PAC guarantee. Basically, the algorithm finds a  $(c, \delta)$ -PAC solution and, with the assumption that the sub-optimality gap is at least  $c$ , the solution is the global optimum with probability at least  $1 - \delta$ . We prove that the computational complexity of Algorithm 3 has only  $O(\log(N))$  dependence on  $N$ .

**Theorem 10.** *Suppose that Assumptions 1-4 hold. Then, Algorithm 3 returns a  $(c, \delta)$ -PAC solution. Furthermore, we have*

$$\begin{aligned} T(\delta, \mathcal{MC}_c) &= O \left[ \frac{d^2 \log(N)}{c^2} \log\left(\frac{1}{\delta}\right) + \frac{d^3 \log(N)}{c^2} \log\left(\frac{d^2 N}{\epsilon^3}\right) + \frac{d^3 \log(N) L^2}{c^2} \right] \\ &= \tilde{O} \left[ \frac{d^2 \log(N)}{c^2} \log\left(\frac{1}{\delta}\right) \right]. \end{aligned}$$

*Proof.* We first prove the correctness of Algorithm 3. Let  $x^*$  be the minimizer of  $f(x)$  and  $f^* := f(x^*)$ . We use the induction method to prove that, for each epoch  $e$ , it holds

$$\tilde{f}(x_e) - f^* \leq \epsilon_e$$

with probability at least  $1 - (e + 1)\delta/(2E)$ . For epoch 0, the solution  $x_0$  is  $(\epsilon_0, \delta/(2E))$ -PAC and we know

$$\tilde{f}(x_0) - f^* \leq \epsilon_0$$

holds with probability at least  $1 - \delta/(2E)$ . We assume that the above event happens for the  $(e - 1)$ -th epoch with probability at least  $1 - e \cdot \delta/(2E)$  and consider the case when this event happens. By

Theorem 9, function  $\tilde{f}(x)$  satisfies the WSM condition with  $\kappa = c^{-1}$ . Hence, the intermediate solution  $x_{e-1}$  satisfies

$$\|x_{e-1} - x^*\|_\infty \leq c^{-1} \left[ \tilde{f}(x_{e-1}) - f^* \right] \leq c^{-1} \epsilon_{e-1} = c^{-1} \cdot 2^{-e+1} \epsilon_0 = 2^{-e-1} N,$$

which implies that  $x^* \in \mathcal{N}(x_{e-1}, 2^{-e-1} N) = \mathcal{N}_e$  and therefore  $x^* \in \mathcal{N}_e$ . For the epoch  $e$ , it holds

$$\tilde{f}(x_e) - f^* = \tilde{f}(x_e) - \min_{x \in \mathcal{N}_e} \tilde{f}(x) \leq \epsilon_e$$

with probability at least  $1 - \delta/(2E)$ . Hence, the above event happens with probability at least  $1 - \delta/(2E) - e \cdot \delta/(2E) = 1 - (e+1)\delta/(2E)$  for epoch  $e$ . By the induction method, we know the claim holds for all epochs. Considering the last epoch, we know

$$\tilde{f}(x_{E-1}) - f^* \leq \epsilon_{E-1} = 2^{-E+1} \epsilon_0 = 2^{-\lceil \log_2(N) \rceil - 2} \cdot cN \leq 2^{-\log_2(N) - 2} \cdot cN = c/4$$

holds with probability at least  $1 - \delta/2$ . Thus, we know  $x_{E-1}$  satisfies the  $(c/4, \delta/2)$ -PAC guarantee. By Theorem 2, the integral solution returned by Algorithm 3 satisfies the  $(c/2, \delta)$ -PAC guarantee. Since the indifference zone parameter is  $c$ , the solution satisfying the  $(c/2, \delta)$ -PAC guarantee must satisfy the  $(c, \delta)$ -PAC guarantee.

Next, we estimate the computational complexity of Algorithm 3. By Theorem 3, the computational complexity of epoch  $e$  is at most

$$\tilde{O} \left[ \frac{d^2 (2^{-e} N)^2}{\epsilon_e^2} \log\left(\frac{E}{\delta}\right) \right] = \tilde{O} \left[ \frac{d^2 (2^{-e} N)^2}{(2^{-e-2} \cdot cN)^2} \log\left(\frac{E}{\delta}\right) \right] = \tilde{O} \left[ \frac{d^2}{c^2} \log\left(\frac{1}{\delta}\right) \right].$$

Summing over  $e = 0, 1, \dots, E-1$ , we know the total computational complexity of  $E$  epochs is at most

$$\tilde{O} \left[ E \cdot \frac{d^2}{c^2} \log\left(\frac{1}{\delta}\right) \right] = \tilde{O} \left[ \frac{d^2 \log(N)}{c^2} \log\left(\frac{1}{\delta}\right) \right].$$

By Theorem 2, the computational complexity of the rounding process is at most

$$\tilde{O} \left[ \frac{d}{c^2} \log\left(\frac{1}{\delta}\right) \right].$$

Combining the two parts, we know the computational complexity of Algorithm 3 is at most

$$\tilde{O} \left[ \frac{d^2 \log(N)}{c^2} \log\left(\frac{1}{\delta}\right) \right].$$

□

Similarly, we can estimate the computational complexity under Assumption 5 and we omit the proof.

**Theorem 11.** *Suppose that Assumptions 1-5 hold. Then, Algorithm 3 returns a  $(c, \delta)$ -PAC solution. Furthermore, we have*

$$T(\delta, \mathcal{MC}_c) = \tilde{O} \left[ \frac{\beta(L+G)^2 \log(N) + d}{c^2} \log\left(\frac{1}{\delta}\right) \right].$$

## D Proofs in Section 4

### D.1 Proof of Theorem 5

The main idea of the generalization to multi-dimensional problems is to view optimization algorithms as estimators to the optimal value. This intuition is elaborated in the following definition.

**Definition 4.** Given a constant  $C > 0$ , we say that an algorithm is **sub-Gaussian with dimension  $d$  and parameter  $C$**  if for any  $d$ -dimensional  $L^\natural$ -convex problem, any  $\epsilon > 0$  and small enough  $\delta > 0$ , the algorithm returns an estimate to  $f^*$  satisfying  $|\hat{f}^* - f^*| \leq \epsilon$  with probability at least  $1 - \delta$  using at most

$$T(\epsilon, \delta) = \tilde{O} \left[ \frac{2C}{\epsilon^2} \log\left(\frac{2}{\delta}\right) \right]$$

objective function value evaluations.

We note that if we treat algorithms as estimators, estimators are generally “biased”. This fact implies that the empirical mean of several  $(\epsilon, \delta)$ -PAC solutions does not produce a better optimality guarantee, while the empirical mean of several unbiased estimators usually has a tighter deviation bound.

Now, we inductively construct sub-Gaussian algorithms for problem (1). We first define the marginal objective function as

$$f^{d-1}(x) := \min_{y \in [N]^{d-1}} f(y, x). \quad (21)$$

Observe that each evaluation of  $f^{d-1}(x)$  requires solving a  $(d-1)$ -dimensional  $L^\natural$ -convex sub-problem. Hence, if we have an algorithm for  $(d-1)$ -dimensional  $L^\natural$ -convex problems, we only need to solve the last one-dimensional problem

$$\min_{x \in [N]} f^{d-1}(x) = \min_{x \in [N]} \min_{y \in [N]^{d-1}} f(y, x) = \min_{x \in \mathcal{X}} f(x) \quad (22)$$

Moreover, we can prove that problem (22) is also a convex problem.

**Lemma 10.** *If function  $f(x)$  is  $L^\natural$ -convex, then function  $f^{d-1}(x)$  is  $L^\natural$ -convex on  $[N]$ .*

*Proof.* Let  $k \in \{2, 3, \dots, N-1\}$ . By the definition of  $f^{d-1}(x)$ , there exists vectors  $y_{k-1}, y_{k+1} \in [N]^{d-1}$  such that

$$f^{d-1}(k-1) = f(y_{k-1}, k-1), \quad f^{d-1}(k+1) = f(y_{k+1}, k+1).$$

By the  $L^\natural$ -convexity of  $f(x)$ , we have

$$\begin{aligned} f^{d-1}(k-1) + f^{d-1}(k+1) &= f(y_{k-1}, k-1) + f(y_{k+1}, k+1) \\ &\geq f\left(\left\lfloor \frac{y_{k-1} + y_{k+1}}{2} \right\rfloor, k\right) + f\left(\left\lfloor \frac{y_{k-1} + y_{k+1}}{2} \right\rfloor, k\right) \\ &\geq 2 \min_{y \in [N]^{d-1}} f(y, k) = 2f^{d-1}(k), \end{aligned}$$

which means the discrete midpoint convexity holds at point  $k$ . Since we can choose  $k$  arbitrarily, we know function  $f^{d-1}(x)$  is convex on  $[N]$ .  $\square$

Based on the observations above, we can use sub-Gaussian algorithms for  $(d-1)$ -dimensional problems and the uniform sampling algorithm to construct sub-Gaussian algorithms for  $d$ -dimensional problems. We give the pseudo-code in Algorithm 4. We prove that Algorithm 4 is sub-Gaussian with dimension  $d$  and estimate its parameter.

**Theorem 12.** *Suppose that Assumptions 1-3 hold, and that Algorithm  $\mathcal{A}$  is sub-Gaussian with dimension  $d-1$  and parameter  $C$ . Then, Algorithm 4 is a sub-Gaussian algorithm with dimension  $d$  and parameter  $MC$ , where  $M > 0$  is an absolute constant.*

*Proof.* We first verify the correctness of Algorithm 4. The algorithm is the same as the uniform sampling algorithm in Zhang et al. (2021) except the condition for implementing Type-II Operations. Hence, if we can prove that, when Type-II Operations are implemented, it holds

$$h \leq |\mathcal{S}| \cdot \epsilon/80, \quad (23)$$

then the proof of Theorem 2 in Zhang et al. (2021) can be directly applied to this case. If the confidence interval is updated at the beginning of current iteration, then we have

$$h = |\mathcal{S}| \cdot \epsilon/160 < |\mathcal{S}| \cdot \epsilon/80.$$

Otherwise, if the confidence interval is not updated in the current iteration. Then, we have  $|\mathcal{S}| > N_{cur}/2$  and therefore

$$h = N_{cur} \cdot \epsilon/160 < 2|\mathcal{S}| \cdot \epsilon/160 = |\mathcal{S}| \cdot \epsilon/80.$$

Combining the two cases, we have inequality (23) and the correctness of Algorithm 4.

Next, we estimate the computational complexity of Algorithm 4. Denote the active sets when we update the confidence interval as  $\mathcal{S}_1, \dots, \mathcal{S}_m$ , where  $m \geq 1$  is the number of times when the

---

**Algorithm 4** Multi-dimensional uniform sampling algorithm
 

---

**Input:** Model  $\mathcal{X}, \mathcal{B}_x, F(x, \xi_x)$ , optimality guarantee parameters  $\epsilon$  and  $\delta$ , sub-Gaussian algorithm  $\mathcal{A}$  with dimension  $d - 1$ .

**Output:** An  $(\epsilon, \delta)$ -PAC solution  $x^*$  to problem (1).

- 1: Set the active set  $\mathcal{S} \leftarrow [N]$ .
- 2: Set the step size  $d \leftarrow 1$  and the maximal number of comparisons  $T_{max} \leftarrow N$ .
- 3: Set  $N_{cur} \leftarrow +\infty$ .
- 4: **while** the size of  $\mathcal{S}$  is at least 3 **do** ▷ Iterate until  $\mathcal{S}$  has at most 2 points.
- 5:   **if**  $|\mathcal{S}| \leq N_{cur}/2$  **then** ▷ Update the confidence interval.
- 6:     Record current active set size  $N_{cur} \leftarrow |\mathcal{S}|$ .
- 7:     Set the confidence width  $h \leftarrow N_{cur} \cdot \epsilon/160$ .
- 8:     For each  $x \in \mathcal{S}$ , use algorithm  $\mathcal{A}$  to get an estimate to  $f^{d-1}(x)$  such that

$$\left| \hat{f}^{d-1}(x) - f^{d-1}(x) \right| \leq h$$

holds with probability at least  $1 - \delta/(2T_{max})$ .

- 9:   **end if**
- 10:   **if**  $\hat{f}^{d-1}(x) + h \leq \hat{f}^{d-1}(y) - h$  for some  $x, y \in \mathcal{S}$  **then** ▷ **Type-I Operation**
- 11:     **if**  $x < y$  **then**
- 12:       Remove all points  $z \in \mathcal{S}$  with the property  $z \geq y$  from  $\mathcal{S}$ .
- 13:     **else**
- 14:       Remove all points  $z \in \mathcal{S}$  with the property  $z \leq y$  from  $\mathcal{S}$ .
- 15:     **end if**
- 16:   **else** ▷ **Type-II Operation**
- 17:     Update the step size  $d \leftarrow 2d$ .
- 18:     Update  $\mathcal{S} \leftarrow \{x_{min}, x_{min} + d, \dots, x_{min} + kd\}$ , where  $x_{min} = \min_{x \in \mathcal{S}} x$  and  $k = \lceil |\mathcal{S}|/2 \rceil - 1$ .
- 19:   **end if**
- 20: **end while** ▷ Now  $\mathcal{S}$  has at most 2 points.
- 21: For each  $x \in \mathcal{S}$ , use Algorithm  $\mathcal{A}$  to obtain an estimate to  $f^{d-1}(x)$  such that

$$\left| \hat{f}^{d-1}(x) - f^{d-1}(x) \right| \leq \epsilon/4$$

holds with probability at least  $1 - \delta/(2T_{max})$ .

- 22: Return  $x^* \leftarrow \arg \min_{x \in \mathcal{S}} \hat{f}^{d-1}(x)$ .
- 

confidence interval is updated. Then, we know  $|\mathcal{S}_1| = N$  and  $|\mathcal{S}_m| \geq 3$ . By the condition for updating the confidence interval, it holds

$$|\mathcal{S}_{k+1}| \leq |\mathcal{S}_k| \quad \forall k \in [m-1],$$

which implies

$$|\mathcal{S}_k| \geq 2^{m-k} |\mathcal{S}_m| \geq 3 \cdot 2^{m-k} \quad \forall k \in [m].$$

Since the algorithm  $\mathcal{A}$  is sub-Gaussian with parameter  $C$ , for each  $x \in \mathcal{S}_k$ , the computational complexity for generating  $\hat{f}^{d-1}(x)$  is at most

$$\frac{2C}{h^2} \log\left(\frac{2T_{max}}{\delta}\right) = \frac{2C}{160^{-2} |\mathcal{S}_k|^2 \epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right) = |\mathcal{S}_k|^{-2} \cdot \frac{51200C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right).$$

Hence, the total computational complexity for the  $k$ -th update of confidence intervals is at most

$$|\mathcal{S}_k| \cdot |\mathcal{S}_k|^{-2} \cdot \frac{51200C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right) = |\mathcal{S}_k|^{-1} \cdot \frac{51200C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right) \leq 2^{k-m}/3 \cdot \frac{51200C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right).$$

Summing over all iterations, we have the computational complexity of all iterations of Algorithm 4 is at most

$$\sum_{k=1}^m 2^{k-m}/3 \cdot \frac{51200C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right) = (2 - 2^{1-m}) \cdot \frac{51200C}{3\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right) < \frac{102400C}{3\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right).$$

Now we consider the computational complexity of the last subproblem. Since the algorithm 4 is sub-Gaussian with parameter  $C$ , the computational complexity of the subproblem is at most

$$2 \cdot \frac{2C}{(\epsilon/4)^2} \log\left(\frac{2T_{max}}{\delta}\right) = \frac{64C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right).$$

Hence, the total computational complexity of Algorithm 4 is at most

$$\frac{102400C}{3\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right) + \frac{64C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right) < 17099 \cdot \frac{2C}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right).$$

When  $\delta$  is small enough, we can choose  $M = 17100$  and the computational complexity of Algorithm 4 is at most

$$\frac{2MC}{\epsilon^2} \log\left(\frac{2T_{max}}{\delta}\right),$$

which implies that Algorithm 4 is sub-Gaussian with dimension  $d$  and parameter  $MC$ .  $\square$

If we treat the noisy evaluation oracle  $F(x, \xi_x)$  as a sub-Gaussian algorithm with dimension 0 and parameter  $\sigma^2$ , then Theorem 12 implies that there exists a sub-Gaussian algorithm with dimension 1 and parameter  $\sigma^2 M$ . However, the parameter  $C$  of the uniform sampling algorithm is usually smaller than  $\sigma^2 M$  and therefore the uniform sampling algorithm is preferred in the one-dimensional case. Using the results of Theorem 12 and the fact that the uniform sampling algorithm is sub-Gaussian with dimension 1, we can inductively construct sub-Gaussian algorithms with any dimension  $d$  and this finishes the proof of Theorem 5.

## D.2 Proof of Theorem 6

As a counterpart of separation oracles, we introduce the stochastic separation oracle, namely the  $(\epsilon, \delta)$ -separation oracle, to characterize the accuracy of separation oracles in the stochastic case.

**Definition 5.** An  $(\epsilon, \delta)$ -separation oracle ( $(\epsilon, \delta)$ - $\mathcal{SO}$ ) is a function on  $[1, N]^d$  with the property that for any input  $x \in [1, N]^d$ , it outputs a vector  $\hat{g} \in \mathbb{R}^d$  such that the inequality

$$f(y) \geq f(x) - \epsilon$$

holds with probability at least  $1 - \delta$  for any  $y \in [1, N]^d \cap H$ , where the half space  $H$  is defined as  $\{z : \langle \hat{g}, z - x \rangle \geq 0\}$ .

We give a concrete example of  $(\epsilon, \delta)$ - $\mathcal{SO}$  oracle and provide an upper bound on the computational complexity of evaluating each oracle. We define the averaged subgradient estimator as

$$\hat{g}_{\alpha_x}^n := \hat{F}_n(S^{x,i}) - \hat{F}_n(S^{x,i-1}), \quad \forall i \in [d], \quad (24)$$

where  $\alpha_x$  is a consistent permutation of  $x$ ,  $n \geq 1$  is the number of samples, and  $\hat{F}_n$  is the empirical mean of  $n$  independent evaluations of  $F$ . The following lemma gives a lower bound on  $n$  to guarantee that  $\hat{g}^n$  is an  $(\epsilon, \delta)$ - $\mathcal{SO}$  oracle.

**Lemma 11.** *Suppose that Assumptions 1-3 hold. If we choose*

$$n = \tilde{O} \left[ \frac{dN^2}{\epsilon^2} \log \left( \frac{1}{\delta} \right) \right],$$

*then  $\hat{g}^n$  is an  $(\epsilon, \delta)$ - $\mathcal{SO}$  oracle. Moreover, the expected computational complexity of generating an  $(\epsilon, \delta)$ - $\mathcal{SO}$  oracle is at most*

$$O \left[ \frac{d^2 N^2}{\epsilon^2} \log \left( \frac{1}{\delta} \right) + d \right] = \tilde{O} \left[ \frac{d^2 N^2}{\epsilon^2} \log \left( \frac{1}{\delta} \right) \right].$$

*Proof.* By the assumption that  $F(x, \xi_x) - f(x)$  is sub-Gaussian with parameter  $\sigma^2$  for any  $x$ , we know that  $\hat{g}_{\alpha_x}^n - g_{\alpha_x}^n$  is the difference of two independent sub-Gaussian random variables and therefore

$$\hat{g}_{\alpha_x}^n - g_{\alpha_x}^n \sim \text{subGaussian}(2\sigma^2) \quad \forall i \in [d],$$

where  $g$  is the subgradient of  $f(x)$  defined in (5). Then, using the properties of sub-Gaussian random variables, it holds that

$$\hat{g}_{\alpha_x(i)}^n - g_{\alpha_x(i)} \sim \text{subGaussian} \left( \frac{2\sigma^2}{n} \right) \quad \forall i \in [d].$$

Recalling that components of  $\hat{g}^n$  are mutually independent, we know

$$\langle \hat{g}^n - g, y - x \rangle = \sum_i (\hat{g}_{\alpha_x(i)}^n - g_{\alpha_x(i)}) \cdot (y - x)_{\alpha_x(i)} \sim \text{subGaussian} \left( \frac{2\sigma^2}{n} \cdot \|y - x\|_2^2 \right).$$

Since  $\|y - x\|_2^2 \leq dN^2$ , we know

$$\langle \hat{g}^n - g, y - x \rangle \sim \text{subGaussian} \left( \frac{2dN^2\sigma^2}{n} \right).$$

By the Hoeffding bound, it holds

$$|\langle \hat{g}^n - g, y - x \rangle| \leq \sqrt{\frac{4dN^2\sigma^2}{n} \log \left( \frac{2}{\delta} \right)}$$

with probability at least  $1 - \delta$ . If we choose

$$n = \left\lceil \frac{4dN^2\sigma^2}{\epsilon^2} \log \left( \frac{2}{\delta} \right) \right\rceil \leq \frac{4dN^2\sigma^2}{\epsilon^2} \log \left( \frac{2}{\delta} \right) + 1,$$

it follows that

$$|\langle \hat{g}^n - g, y - x \rangle| \leq \epsilon. \quad (25)$$

Since  $f(x)$  is a convex function and  $g$  is a subgradient at point  $x$ , we have  $f(y) \geq f(x) + \langle g, y - x \rangle$  for all  $y \in [1, N]^d$ . Combining with inequality (25) gives

$$f(y) \geq f(x) + \langle \hat{g}^n, y - x \rangle + \langle g - \hat{g}^n, y - x \rangle \geq f(x) + \langle \hat{g}^n, y - x \rangle - \epsilon \quad \forall y \in [1, N]^d$$

holds with probability at least  $1 - \delta$ . Then, considering the half space  $H = \{y : \langle \hat{g}^n, y - x \rangle \leq 0\}$ , it holds

$$f(y) \geq f(x) + \langle \hat{g}^n, y - x \rangle - \epsilon \geq f(x) - \epsilon \quad \forall y \in [1, N]^d \cap H^c$$

with the same probability. Taking the minimum over  $[1, N]^d \cap H^c$ , it follows that the averaged stochastic subgradient provides an  $(\epsilon, \delta)$ - $\mathcal{SO}$  oracle. Finally, the expected computational complexity of each oracle evaluation is at most

$$d \cdot n \leq \frac{4d^2N^2\sigma^2}{\epsilon^2} \log \left( \frac{2}{\delta} \right) + d = \tilde{O} \left[ \frac{d^2N^2\sigma^2}{\epsilon^2} \log \left( \frac{1}{\delta} \right) \right].$$

□

Using the above results, the cutting-plane methods in Vaidya (1996); Bertsimas & Vempala (2004) can be used to shrink the search polytope without excluding all  $\epsilon$ -approximate solutions with high probability. We give the pseudo-code of the dimension reduction algorithm in Algorithm 5.

*Proof of Theorem 6.* We first verify the correctness of Algorithm 5. If the optimal solution has been removed during the dimension reduction process, we claim that the optimal solutions are removed from the search set by some cutting plane. This is because the dimension reduction steps will not remove integral points from the current search set (Jiang, 2020). This implies that  $x$  is not in the half space

$$H := \{y : \hat{g}^T y \leq \hat{g}^T z\},$$

where  $\hat{g}$  is an  $(\epsilon/8, \delta/4)$ - $\mathcal{SO}$  oracle. Then, by the definition of  $(\epsilon/8, \delta/4)$ - $\mathcal{SO}$  oracle and the claim that  $x \in [1, N]^d \cap H^c$ , it holds

$$\min_{x \in \mathcal{S}} f(x) \leq \min_{x \in \mathcal{X}} f(x) + \epsilon/4 \quad (26)$$

with probability at least  $1 - \delta/4$ .

---

**Algorithm 5** Dimension reduction method for PAC guarantee
 

---

**Input:** Model  $\mathcal{X}, \mathcal{B}_x, F(x, \xi_x)$ , optimality guarantee parameters  $\epsilon$  and  $\delta$ ,  $(\epsilon, \delta)$ - $\mathcal{SO}$  oracle  $\hat{g}$ .

**Output:** An  $(\epsilon, \delta)$ -PAC solution  $x^*$  to problem (1).

- 1: Set the initial polytope  $P \leftarrow [1, N]^d$ .
  - 2: Initialize the set of points used to query separation oracles  $\mathcal{S} \leftarrow \emptyset$ .
  - 3: **for**  $d' = d, d-1, \dots, 2$  **do**
  - 4:   Apply Vaidya's method or the random walk-based cutting-plane method with  $(\epsilon/4, \delta/4)$ - $\mathcal{SO}$  oracles.
  - 5:   Add points to  $\mathcal{S}$  whenever separation oracles are called.
  - 6:   Stop when the volume of  $P$  is small enough.
  - 7:   Find the hyperplane  $H$  that contains all integral points in  $P$ .  
        $\triangleright$  If  $P$  contains no integral points, then an arbitrary hyperplane works.
  - 8:   Update  $P$  to a polytope on the hyperplane  $H$ .                    $\triangleright$  Reduce the dimension by 1.
  - 9: **end for**
  - 10: Find an  $(\epsilon/4, \delta/4)$ -PAC solution of the last one-dim problem and add the solution to  $\mathcal{S}$ .
  - 11: Find the  $(\epsilon/4, \delta/4)$ -PAC solution  $\hat{x}$  of problem  $\min_{x \in \mathcal{S}} f(x)$ .
  - 12: Round  $\hat{x}$  to an integral solution by Algorithm 1.
- 

Otherwise if the optimal solution has not been removed from the search set throughout the dimension reduction process, we know the last one-dimensional problem contains the optimal solution. Hence, the  $(\epsilon/4, \delta/4)$ -PAC solution to the one-dimensional problem is also an  $(\epsilon/4, \delta/4)$ -PAC solution to the original problem. Since the PAC solution is also added to the set  $\mathcal{S}$ , we also have relation (26) holds with probability at least  $1 - \delta/4$ . Then, the  $(\epsilon/4, \delta/4)$ -PAC solution  $\bar{x}$  to problem  $\min_{x \in \mathcal{S}} f(x)$  satisfies

$$f(\bar{x}) \leq \min_{x \in \mathcal{X}} f(x) + \epsilon/2$$

with probability at least  $1 - \delta/2$ , or equivalently  $\bar{x}$  is an  $(\epsilon/2, \delta/2)$ -PAC solution to problem (1). Using the results of Theorem 2, the solution returned by Algorithm 5 is an  $(\epsilon, \delta)$ -PAC solution.

Next, we estimate the expected computational complexity of Algorithm 5. By the results in Jiang (2020),  $(\epsilon/4, \delta/4)$ - $\mathcal{SO}$  oracles are called at most  $O[d(d + \log(N))]$  times. Hence, the size of  $\mathcal{S}$  is at most  $O[d(d + \log(N))]$ . By the estimates in Lemma 11, the total computational complexity of the dimension reduction process is at most

$$O \left[ \frac{d^3 N^2 (d + \log(N))}{\epsilon^2} \log\left(\frac{1}{\delta}\right) + d^2 (d + \log(N)) \right] = \tilde{O} \left[ \frac{d^3 N^2 (d + \log(N))}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

Moreover, the one-dimensional convex problem has at most  $N$  feasible points and Theorem 2 in Zhang et al. (2021) implies that the expected computational complexity for this problem is at most

$$\tilde{O} \left[ \frac{1}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

Since the size of  $\mathcal{S}$  is at most  $O[d(d + \log(N))]$ , the sub-problem for the set  $\mathcal{S}$  takes at most

$$O \left[ \frac{d^2 (d + \log(N))}{\epsilon^2} \log\left(\frac{1}{\delta}\right) + d^2 (d + \log(N)) \right] = \tilde{O} \left[ \frac{d^2 (d + \log(N))}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right]$$

objective function value evaluations. Finally, Theorem 2 shows that the expected computational complexity of the rounding process is at most

$$\tilde{O} \left[ \frac{d}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

In summary, the total expected computational complexity of Algorithm 5 is at most

$$\tilde{O} \left[ \frac{d^3 N^2 (d + \log(N))}{\epsilon^2} \log\left(\frac{1}{\delta}\right) \right].$$

□

## E Proofs in Section 5

### E.1 Proof of Theorem 7

The information-theoretical inequality in Kaufmann et al. (2016) provides a systematic way to prove lower bounds of zeroth-order algorithms. Given an algorithm and a model  $M$ , we denote  $N_x(\tau)$  as the number of times that  $F(x, \xi_x)$  is evaluated when the algorithm terminates, where  $\tau$  is the stopping time of the algorithm. Then, it follows from the definition that

$$\mathbb{E}_M[\tau] = \sum_{x \in \mathcal{X}} \mathbb{E}_M [N_x(\tau)],$$

where  $\mathbb{E}_M$  is the expectation when the model  $M$  is given. Similarly, we can define  $\mathbb{P}_M$  as the probability when the model  $M$  is given. The following lemma was proved in Kaufmann et al. (2016) and is the major tool for deriving lower bounds in this paper.

**Lemma 12.** *For any two models  $M_1, M_2$  and any event  $\mathcal{E} \in \mathcal{F}_\tau$ , we have*

$$\sum_{x \in \mathcal{X}} \mathbb{E}_{M_1} [N_x(\tau)] \text{KL}(\nu_{1,x}, \nu_{2,x}) \geq d(\mathbb{P}_{M_1}(\mathcal{E}), \mathbb{P}_{M_2}(\mathcal{E})), \quad (27)$$

where  $d(x, y) := x \log(x/y) + (1-x) \log((1-x)/(1-y))$ ,  $\text{KL}(\cdot, \cdot)$  is the KL divergence and  $\nu_{k,x}$  is the distribution of model  $M_k$  at point  $x$  for  $k = 1, 2$ .

The information-theoretical inequality (27) is our major tool for deriving lower bounds. We first reduce the construction of  $L^{\natural}$ -convex functions to the construction of submodular functions. Then, using the family of submodular functions defined in Graur et al. (2020), we can construct  $d + 1$  submodular functions that have different optimal solutions and have the same value except on  $d + 1$  potential solutions. Hence, the algorithm has to simulate enough samples on the  $d + 1$  potential solutions to decide the optimal solution and the computational complexity is proportional to  $d$ .

*Proof of Theorem 7.* In this proof, we change the feasible set to  $\mathcal{X} = \{0, 1, \dots, N\}^d$ , where  $N \geq 1$ . We split the proof into three steps.

**Step 1.** We first show that the construction of  $L^{\natural}$ -convex functions can be reduced to the construction of submodular functions. Equivalently, we show that any submodular function defined on  $\{0, 1\}^d$  can be extended to an  $L^{\natural}$ -convex function on  $\mathcal{X}$  with the same convex extension after scaling. Let  $g(x)$  be a submodular function defined on  $\{0, 1\}^d$  and  $\tilde{g}(x)$  be the Lovász extension of  $g(x)$ . We first extend the domain of the Lovász extension to  $[0, N]^d$  by scaling, i.e.,

$$\tilde{f}(x) := \tilde{g}(x/N) \quad \forall x \in [0, N]^d.$$

Then, we define the discretization of  $\tilde{f}(x)$  by restricting to the integer lattice

$$f(x) := \tilde{f}(x) \quad \forall x \in \mathcal{X}.$$

We prove that  $f(x)$  is an  $L^{\natural}$ -convex function. By Proposition 7.25 in Murota (2003), we know the Lovász extension  $\tilde{g}(x)$  is a polyhedral  $L$ -convex function. Since the scaling operation does not change the  $L$ -convexity, we know  $\tilde{f}(x)$  is also polyhedral  $L$ -convex. Hence, by Theorem 7.29 in Murota (2003), the function  $\tilde{f}(x)$  satisfies the  $\text{SBF}^{\natural}[\mathbb{R}]$  property, namely,

$$\tilde{f}(p) + \tilde{f}(q) \geq \tilde{f}[(p - \alpha \mathbf{1}) \vee q] + \tilde{f}(p \wedge (q + \alpha \mathbf{1})) \quad \forall p, q \in [0, N]^d, \alpha \geq 0.$$

Restricting to the integer lattice, we know the  $\text{SBF}^{\natural}[\mathbb{Z}]$  property holds for  $f(x)$ , namely,

$$f(p) + f(q) \geq f[(p - \alpha \mathbf{1}) \vee q] + f(p \wedge (q + \alpha \mathbf{1})) \quad \forall p, q \in \{0, \dots, N\}^d, \alpha \in \mathbb{N}.$$

Finally, Theorem 7.7 in Murota (2003) shows that the  $L^{\natural}$ -convexity is equivalent to the  $\text{SBF}^{\natural}[\mathbb{Z}]$  property and therefore we know that  $f(x)$  is an  $L^{\natural}$ -convex function.

**Step 2.** Next, we construct  $d + 1$  submodular functions on  $\{0, 1\}^d$  and extend them to  $\mathcal{X}$  by the process defined in Step 1. The construction is based on the family of submodular functions defined in Graur et al. (2020). We denote  $\mathcal{I} := \{0\} \cup [d]$ . For each  $i \in \mathcal{I}$ , we define point  $x^i \in \{0, 1\}^d$  as

$$x^i := \sum_{j=1}^i e_j,$$

where  $e_j$  is the  $j$ -th unit vector of  $\mathbb{R}^d$ . Index  $j(x)$  is defined as the maximal index  $j$  such that

$$x_i = 1 \quad \forall i \in [j].$$

If  $x_1 = 0$ , then we define  $j(x) = 0$ . Given  $c : \mathcal{I} \mapsto \mathbb{R}$ , we define a function on  $\{0, 1\}^d$  as

$$g^c(x) := \begin{cases} -c(i) & \text{if } x = x^i \text{ for some } i \in \mathcal{I} \\ (\|x\|_1 - j(x)) \cdot (d + 2 - \|x\|_1) & \text{otherwise.} \end{cases}$$

By Lemma 6 in Graur et al. (2020), the function  $g^c(x)$  is submodular if  $c(\mathcal{I}) \subset \{0, 1\}$ . Using the fact that convex combinations of submodular functions are still submodular, we know that  $g^c(x)$  is submodular for any  $c$  such that  $c(\mathcal{I}) \subset [0, 1]$ . Then, for each  $i \in \mathcal{I}$ , we construct

$$c^i(0) := \frac{1}{2}, \quad c^i(j) := \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases} \quad \forall j \in [d].$$

We denote  $g^i(x) := g^{c^i}(x)$  and let  $f^i(x)$  be the extension of  $6\epsilon \cdot g^i(x)$  on  $\mathcal{X}$  by the process in Step 1. By the result in Step 1, we know that  $f^i(x)$  is  $L^1$ -convex.

Next, we prove that  $f^0(x)$  has disjoint set of  $\epsilon$ -optimal solutions with  $f^i(x)$  for any  $i \in [d]$ . For each  $f^i(x)$ , we define the set of  $\epsilon$ -optimal solutions as

$$\mathcal{X}_\epsilon^i := \{x \in \mathcal{X} : f^i(x) - \min_y f^i(y) \leq \epsilon\}.$$

We first consider  $\mathcal{X}_\epsilon^0$ . By the definition of  $g^0(x)$ , we know that

$$f^0(x^0) = g^0(x^0) = -3\epsilon, \quad f^0(x) = g^0(x/N) \geq 0 \quad \forall x \in \{0, N\}^d \setminus \{x^0\}, \quad (28)$$

which implies that

$$\mathcal{X}_\epsilon^0 = \{x \in \mathcal{X} : f^0(x) \leq -2\epsilon\}.$$

Since  $f^0(x)$  is defined by the scaled Lovász extension of  $g^0(x)$ , we have

$$f^0(x) = N^{-1} \cdot \left[ (N - x_{\alpha(1)})f^0(S^0) + \sum_{i=1}^{d-1} (x_{\alpha(i)} - x_{\alpha(i+1)})f^0(S^i) + x_{\alpha(d)}f^0(S^d) \right], \quad (29)$$

where  $\alpha$  is a consistent permutation of  $x/N$  and  $S^i := N \cdot S^{x/N, i} \in \{0, N\}^d$  is the  $i$ -th neighbouring points of  $x$  in the hypercube  $\{0, N\}^d$ . Using the relation in (28) and the fact  $S^0 = x^0$ , we get

$$f^0(x) \geq N^{-1} \cdot (N - x_{\alpha(1)})f(S_0) = N^{-1} \cdot (N - x_{\alpha(1)})f(x^0) = -3\epsilon N^{-1} \cdot (N - x_{\alpha(1)}).$$

Hence, for any point  $x \in \mathcal{X}_\epsilon^0$ , we have  $N - x_{\alpha(1)} = N - \max_i x_i \geq 2N/3$  and therefore

$$\mathcal{X}_\epsilon^0 \subset \{x \in \mathcal{X} : N - \max_i x_i \geq 2N/3\} = \{x \in \mathcal{X} : \max_i x_i \leq N/3\}. \quad (30)$$

Next, we consider  $\mathcal{X}_\epsilon^i$  with  $i \in [d]$ . By the definition of  $g^i(x)$ , we have

$$f^i(x^0) = g^i(x^0) = -3\epsilon, \quad f^i(x) = g^i(x) \geq -6\epsilon \quad \forall x \in \{0, N\}^d \setminus \{x^0\},$$

which implies that

$$\mathcal{X}_\epsilon^i = \{x \in \mathcal{X} : f^i(x) \leq -5\epsilon\}.$$

Since the consistent permutation and neighboring points only depend on the coordinate of  $x$ , we know

$$f^i(x) = N^{-1} \cdot \left[ (N - x_{\alpha(1)})f^i(S^0) + \sum_{i=1}^{d-1} (x_{\alpha(i)} - x_{\alpha(i+1)})f^i(S^i) + x_{\alpha(d)}f^i(S^d) \right] \quad (31)$$

$$\begin{aligned}
&\geq N^{-1} \cdot \left[ -3\epsilon(N - x_{\alpha(1)}) - 6\epsilon \sum_{i=1}^{d-1} (x_{\alpha(i)} - x_{\alpha(i+1)}) - 6\epsilon \cdot x_{\alpha(d)} \right] \\
&= N^{-1} \cdot [-3\epsilon(N - x_{\alpha(1)}) - 6\epsilon \cdot x_{\alpha(1)}] = -3\epsilon N^{-1} \cdot (N + x_{\alpha(1)}).
\end{aligned}$$

Hence, the set  $\mathcal{X}_\epsilon^i$  satisfies

$$\mathcal{X}_\epsilon^i \subset \{x \in \mathcal{X} : N + \max_i x_i \geq 5N/3\} = \{x \in \mathcal{X} : \max_i x_i \geq 2N/3\}. \quad (32)$$

Combining the relations (30) and (32), we know  $\mathcal{X}_\epsilon^0 \cap \mathcal{X}_\epsilon^i = \emptyset$  for all  $i \in [d]$ .

**Step 3.** Finally, we give a lower bound of  $T_0(\epsilon, \delta, \mathcal{MC})$ . For each  $i \in \mathcal{I}$ , we define  $M_i$  as the model such that the objective function is  $f^i(x)$  and the distribution at each point is Gaussian with variance  $\sigma^2$ . Same as the one-dimensional case, given a zeroth-order algorithm and a model  $M$ , we denote  $N_x(\tau)$  as the number of times that  $F(x, \xi_x)$  is simulated when the algorithm terminates. By definition, we have

$$\mathbb{E}_M[\tau] = \sum_{x \in \mathcal{X}} \mathbb{E}_M[N_x(\tau)],$$

where  $\mathbb{E}_M$  is the expectation when the model  $M$  is given. Similarly, we can define  $\mathbb{P}_M$  as the probability when the model  $M$  is given. Suppose  $\mathcal{A}$  is an  $[(\epsilon, \delta)$ -PAC,  $\mathcal{MC}]$ -algorithm and let  $\mathcal{E}$  be the event that the solution returned by  $\mathcal{A}$  is in the set  $\mathcal{X}_\epsilon^0$ . Since  $\mathcal{X}_\epsilon^0 \cap \mathcal{X}_\epsilon^i = \emptyset$  for all  $i \in [d]$ , we know

$$\mathbb{P}_{M_0}[\mathcal{E}] \geq 1 - \delta, \quad \mathbb{P}_{M_i}[\mathcal{E}] \leq \delta \quad \forall i \in [d].$$

Using the information-theoretical inequality (27), it holds

$$\sum_{x \in \mathcal{X}} \mathbb{E}_{M_0}[N_x(\tau)] \text{KL}(\nu_{0,x}, \nu_{i,x}) \geq d(\mathbb{P}_{M_0}(\mathcal{E}), \mathbb{P}_{M_i}(\mathcal{E})) \geq d(1 - \delta, \delta) \geq \log\left(\frac{1}{2.4\delta}\right), \quad (33)$$

where  $d(x, y) := x \log(x/y) + (1-x) \log((1-x)/(1-y))$ ,  $\text{KL}(\cdot, \cdot)$  is the KL divergence and  $\nu_{i,x}$  is the distribution of  $F^i(x, \xi_x)$ . Since the distributions  $\nu_{i,x}$  are Gaussian with variance  $\sigma^2$ , the KL divergence can be calculated as

$$\text{KL}(\nu_{0,x}, \nu_{i,x}) = 2\sigma^{-2} (f^0(x) - f^i(x))^2.$$

Now we estimate  $f^0(x) - f^i(x)$  for all  $i \in [d]$ . By equations (29) and (31), we get

$$\begin{aligned}
f^0(x) - f^i(x) &= N^{-1} \left[ (N - x_{\alpha(1)}) (f^0(S^0) - f^i(S^0)) \right. \\
&\quad \left. + \sum_{j=1}^{d-1} (x_{\alpha(j)} - x_{\alpha(j+1)}) (f^0(S^j) - f^i(S^j)) + x_{\alpha(d)} (f^0(S^d) - f^i(S^d)) \right], \quad (34)
\end{aligned}$$

where  $\alpha$  is a consistent permutation of  $x/N$  and  $S^i$  is the  $i$ -th neighboring point of  $x$  in hypercube  $\{0, N\}^d$ . By the definition of  $f^0(x)$  and  $f^i(x)$ , we have

$$f^0(x) - f^i(x) = \begin{cases} 6\epsilon & \text{if } x = x^i \\ 0 & \text{otherwise.} \end{cases}$$

Since  $\|x^i\|_1 = i$  and  $\|S^j\|_1 = j$  for all  $j \in \mathcal{I}$ , we know

$$f^0(S^i) - f^i(S^i) \leq 6\epsilon, \quad f^0(S^j) - f^i(S^j) = 0 \quad \forall j \in \mathcal{I} \setminus \{i\}.$$

Substituting into equation (34), it follows that

$$f^0(x) - f^i(x) \leq \begin{cases} (6\epsilon \cdot (x_{\alpha(i)} - x_{\alpha(i+1)}))^2 & \text{if } i \in [d-1] \\ (6\epsilon \cdot x_{\alpha(d)})^2 & \text{if } i = d. \end{cases}$$

Hence, the KL divergence is bounded by

$$\text{KL}(\nu_{0,x}, \nu_{i,x}) = 2\sigma^{-2} (f^0(x) - f^i(x))^2 \leq \begin{cases} 72\sigma^{-2} N^{-2} \epsilon^2 ((x_{\alpha(i)} - x_{\alpha(i+1)}))^2 & \text{if } i \in [d-1] \\ 72\sigma^{-2} N^{-2} \epsilon^2 x_{\alpha(d)}^2 & \text{if } i = d. \end{cases}$$

Substituting the KL divergence into inequality (33) and summing over  $i = 1, \dots, d$ , we get

$$\sum_{x \in \mathcal{X}} \mathbb{E}_{M_0} [N_x(\tau)] \cdot 72\sigma^{-2} N^{-2} \epsilon^2 \left[ \sum_{i=1}^{d-1} (x_{\alpha(i)} - x_{\alpha(i+1)})^2 + x_{\alpha(d)}^2 \right] \geq d \log \left( \frac{1}{2.4\delta} \right). \quad (35)$$

Since  $\alpha$  is the consistent permutation of  $x$ , we know

$$0 \leq x_{\alpha(i)} - x_{\alpha(i+1)} \leq N \quad \forall i \in [d-1]$$

and therefore

$$\sum_{i=1}^{d-1} (x_{\alpha(i)} - x_{\alpha(i+1)})^2 + x_{\alpha(d)}^2 \leq N \cdot \left( \sum_{i=1}^{d-1} (x_{\alpha(i)} - x_{\alpha(i+1)}) + x_{\alpha(d)} \right) = N \cdot x_{\alpha(1)} \leq N^2.$$

Combining with inequality (35), we get

$$\sum_{x \in \mathcal{X}} \mathbb{E}_{M_0} [N_x(\tau)] \cdot 72\epsilon^2 \sigma^{-2} \geq d \log \left( \frac{1}{2.4\delta} \right),$$

which implies that

$$\mathbb{E}_{M_0}[\tau] = \sum_{x \in \mathcal{X}} \mathbb{E}_{M_0} [N_x(\tau)] \geq \frac{d\sigma^2}{72\epsilon^2} \log \left( \frac{1}{2.4\delta} \right).$$

□

## F More numerical experiments

### F.1 Separable convex function minimization

We consider the problem of minimizing a stochastic function whose expectation is a separable  $L^{\flat}$ -convex function of the form

$$f_{c,x^*}(x) := \sum_{i=1}^d c_i g(x_i^*; x_i),$$

where  $c_i \in [0.75, 1.25]$ ,  $x_i^* \in \{1, \dots, \lfloor 0.3N \rfloor\}$  for all  $i \in [d]$  and

$$g(y^*; y) := \begin{cases} \sqrt{\frac{y^*}{y}} - 1 & \text{if } y \leq y^* \\ \sqrt{\frac{N+1-y^*}{N+1-y}} - 1 & \text{if } y > y^* \end{cases} \quad \forall y, y^* \in [N].$$

It can be observed that the function  $f_{c,x^*}(x)$  is the sum of separable convex functions and therefore is  $L^{\flat}$ -convex. Moreover, the function  $f_{c,x^*}(x)$  has the optimum  $x^*$  associated with the optimal value 0. For stochastic evaluations, we add Gaussian noise with mean 0 and variance 1. Figure 1 in Zhang et al. (2021) shows that the landscape of  $g(x^*; x)$  is very flat. The advantage of this numerical example is that the expected objective function has a closed form, and we are able to exactly compute the optimality gap of the solutions returned by the proposed algorithms.

The dimension and scale of the separable convex model are chosen as  $d \in \{2, 6, 10\}$  and  $N \in \{50, 100, 150\}$ . The optimality guarantee parameters are chosen as  $\epsilon = (d!)^{1/d}/5$  and  $\delta = 10^{-6}$ , respectively. The choice of  $\epsilon$  ensures that the  $\epsilon$ -sub-level set of objective function covers a “flat region” of the landscape. Same as the one-dimensional case, we compute the average computational complexity of 400 independently generated models to approximate the expected computational complexity. Moreover, early stopping conditions are designed to terminate algorithms early when little progress is made at any iteration. For the truncated subgradient descent method, we maintain the empirical mean of stochastic objective function values up to the current iteration and terminate the algorithm if the empirical mean does not decrease by  $\epsilon/\sqrt{N}$  after  $O(d\epsilon^{-2} \log(1/\delta))$  consecutive iterations. For the dimension reduction method, we terminate the algorithm if the polytope is empty. Furthermore, we have observed that using  $(N\epsilon/4, \delta/4)$ - $\mathcal{SO}$  oracles is enough for producing high-probability guarantees within the range  $1 \leq N \leq 150$ .

## F.2 Optimal allocation problem of a service system

In this subsection, we implement our proposed algorithms on the optimal allocation problem of a queueing system. We consider the 24-hour operation of a service system with a single stream of incoming customers. The customers arrive according to a doubly stochastic non-homogeneous Poisson process with intensity function

$$\Lambda(t) := 0.5\lambda N \cdot (1 - |t - 12|/12) \quad \forall t \in [0, 24],$$

where  $\lambda$  is a positive constant and  $N$  is a positive integer. Each customer requests a service with service time independent and identically distributed according to the log-normal distribution with mean  $1/\lambda$  and variance 0.1. We divide the 24-hours operation into  $d$  time slots with length  $24/d$  for some positive integer  $d$ . For the  $i$ -th time slot, there are  $x_i \in [N]$  of homogeneous servers that work independently in parallel and the number of servers cannot be changed during the slot. Assume that the system operates based on a first-come-first-serve routine, with unlimited waiting room in each queue, and that customers never abandon.

The decision maker's objective is to select the staffing level  $x := (x_1, \dots, x_d)$  such that the total waiting time of all customers is minimized. Namely, letting  $f(x)$  be the expected total waiting time under the staffing plan  $x$ , then the optimization problem can be written as

$$\min_{x \in [N]^d} f(x). \quad (36)$$

It has been proved in Altman et al. (2003) that the function  $f(\cdot)$  is multimodular. We define the linear transformation

$$g(y) := (y_1, y_2 - y_1, \dots, y_d - y_{d-1}) \quad \forall y \in \mathbb{R}^d.$$

Then, Murota (2003) has proved that

$$h(y) := f \circ g(y) = f(y_1, y_2 - y_1, \dots, y_d - y_{d-1})$$

is an  $L^{\natural}$ -convex function on the  $L^{\natural}$ -convex set

$$\mathcal{Y} := \{y \in [Nd]^d \mid y_1 \in [N], y_{i+1} - y_i \in [N], i = 1, \dots, N-1\}.$$

The optimization problem (36) has the trivial solution  $x_1 = \dots = x_d = N$ . However, in reality, it is also necessary to keeping the staffing cost low. There are two different approaches to achieve this goal. Firstly, we can constrain the total number of servers  $\sum_{i=1}^d x_i$  to be at most  $K$ , where  $K \leq Nd$  is a positive integer and the optimization problem can be written as

$$\min_{y \in \mathcal{Y}} h(y) \quad \text{s.t. } y_d \leq K. \quad (37)$$

On the other hand, we can add a regularization term  $R(x_1, \dots, x_d) := c/d \cdot \sum_{i=1}^d x_i = c/d \cdot y_d$  to the objective function, where  $c$  is a positive constant. The optimization problem can be written as

$$\min_{y \in \mathcal{Y}} h(y) + c/d \cdot y_d. \quad (38)$$

We refer problems (37) and (38) as the constrained and the regularized problems, respectively. Our algorithms can be extended to this case by considering the Lovász extension  $\tilde{h}(y)$  on the set

$$\tilde{\mathcal{Y}} := \{y \in [1, Nd]^d \mid y_1 \in [1, N], y_{i+1} - y_i \in [1, N], i = 1, \dots, N-1\}.$$

We compare the performance of the projected SSGD method (Algorithm 2) with truncation ( $M < \infty$ ) and without truncation ( $M = \infty$ ) on both problems. In the truncation-free case, the step size is chosen to be  $\eta = O(N\sqrt{d/T})$ . We first fix the dimension (number of time slots) to be  $d = 4$  and compare the performance when the scale  $N \in \{10, 20, 30, 40, 50\}$ , and we then fix the scale to be  $N = 10$  and compare the performance when the dimension  $d \in \{4, 8, 12, 16, 20, 24\}$ . The parameters of the problem are chosen as  $\lambda = 4$ ,  $c = 50$  and  $K = \lfloor Nd/3 \rfloor$ , and the optimality guarantee parameters are  $\epsilon = N/2$  and  $\delta = 10^{-6}$ . For each problem setup, we average the computational complexities of 10 independent implementations to estimate the expected computational complexity. Moreover, early stopping is used to terminate algorithms early when little progress is made after some iterations. More concretely, we maintain the empirical mean of stochastic objective function values up to the current iteration and terminate the algorithm if the empirical mean does not decrease by  $\epsilon/\sqrt{N}$  after  $O(d\epsilon^{-2} \log(1/\delta))$  consecutive iterations.

We first implement both algorithms on the trivial problem (36) for 10 times. Since the optimal solution is known, it is possible to verify whether the solutions returned by algorithms are at most  $\epsilon$  worse than the optimum. It turns out that both algorithms succeed in implementations with any parameters  $d$  and  $N$ . Next, we consider the performance of algorithms on problems (37) and (38), and the results are summarized in Table 3.