Stochastic Localization Methods for Discrete Convex Simulation Optimization

Haixiang Zhang
Department of Mathematics, University of California, Berkeley, CA 94720, haixiang_zhang@berkeley.edu

Zeyu Zheng
Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, zyzheng@berkeley.edu

Javad Lavaei
Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, lavaei@berkeley.edu

This Version: December 3, 2020

We propose a set of new algorithms based on stochastic localization methods for large-scale discrete simulation optimization problems with convexity structure. All proposed algorithms, with the general idea of "localizing" potential good solutions to an adaptively shrinking subset, are guaranteed with high probability to identify a solution that is close enough to the optimal given any precision level. Specifically, for one-dimensional large-scale problems, we propose an enhanced adaptive algorithm with an expected simulation cost asymptotically independent of the problem scale, which is proved to attain the best achievable performance. For multi-dimensional large-scale problems, we propose statistically guaranteed stochastic cutting-plane algorithms, the simulation costs of which have no dependence on model parameters such as the Lipschitz parameter, as well as low polynomial order of dependence on the problem scale and dimension. Numerical experiments are implemented to support our theoretical findings. The theory results, joint the numerical experiments, provide insights and recommendations on which algorithm to use in different real application settings.

Key words: Discrete simulation optimization, convex optimization, enhanced adaptive algorithm, best achievable performance, stochastic cutting-plane methods

1. Introduction

In the area of operations research and management science, many decision making problems involve complex stochastic systems and discrete decision variables. In presence of stochastic uncertainties, many replications of stochastic simulation are often needed to accurately evaluate the objective function associated with a discrete decision variable. Such problems are sometimes referred to as *Discrete Simulation Optimization* or *Discrete Optimization via Simulation* (see Nelson (2010), Hong et al. (2015)). For complex stochastic systems, even one replication of simulation can be time consuming or costly. When the decision space is large, it is often computationally impractical to run simulations for all the decision variables, creating a challenge in finding the optimal or near-optimal decision variable. To circumvent this challenge, problem structure such as convexity or local convexity of the objective function may need to be exploited to save costs and improve efficiency.

In this paper, we consider discrete simulation optimization problems with a large number of discrete decision variables and a convex objective function. Optimization problems with such features naturally arise in many operations research and management science applications, including queueing networks, supply chain networks, sharing economy operations, financial markets, etc.; see Shaked and Shanthikumar (1988), Wolff and Wang (2002), Altman et al. (2003), Singhvi et al. (2015), Jian et al. (2016), Freund et al. (2017) for examples and discussions. In these applications where simulation is heavily used to evaluate system performance, the convexity of expected performance as the objective function is either theoretically proved or empirically verified. Even in presence of convexity, there may be a large number of decision variables that render very close objective value to the optimal, and information such as the gap between the optimal and the sub-optimal is hard to exactly know a priori. Our goal in this paper is to develop provably efficient simulation optimization algorithms that are guaranteed with arbitrarily high probability $1-\delta$ to find a good decision variable that renders an objective value ϵ-close compared to the optimal, where ϵ is any arbitrary small user-specified precision level. This criterion is called (ϵ,δ)-*Probability of Good Selection* ((ϵ,δ)-PGS) in the literature; see Ma and Henderson (2017) and Hong et al. (2020).

The major methodology in algorithm design in this paper can be classified as stochastic localization methods, in the sense that we 'localize' potential good solutions in a subset and adaptively shrink the subset at each step. Zhang and Zheng (2020) also considered the problem of discrete convex simulation optimization, but their methodology is based on gradient-based search in the discrete decision space, so no decision variable is screened out during the optimization procedure. Different from both the stochastic localization methods in this paper and the gradient-based search methods in Zhang and Zheng (2020), in the simulation optimization literature, the method of metamodeling has also been used to exploit the structure of objective function so that not all decision variables are needed; see Ankenman et al. (2010), Sun et al. (2014), Barton (2015), Kleijnen (2017), Salemi et al. (2019) and references within. We describe our contributions as follows.

First, we consider large-scale one-dimensional problems with the decision space as $\{1, 2, \ldots, N\}$, in which N is a large number that represents the problem scale. This problem setting is mathematically equivalent to the problem of *ranking and selection*; see Hong et al. (2020) for a review. The objective function is assumed to be discrete convex on the decision space, but no other structure information such as strong convexity or a minimal gap between the optimal and suboptimal is known. To quantify the computational cost, we take the view that the simulation cost is the dominant contributor to the computational cost; see also Ma and Henderson (2019). The simulation cost of an algorithm is measured as the total number of simulation replications run at all decision variables visited by the algorithm until it stops. When designing algorithms to solve large-scale discrete simulation optimization problems, the dependence of the simulation cost on the problem scale N (or, the number of alternatives/solutions/systems in the area of ranking and selection) is crucial to understand; see also discussions in Zhong and Hong (2019). We design an adaptive sampling (AS) algorithm that is guaranteed to return a decision variable that satisfies the (ϵ, δ)-PGS criterion. The upper bound on the asymptotic simulation cost for the AS algorithm for any convex problem can be proved as $O(\log(N)\epsilon^{-2}\log(1/\delta))$ when ϵ and δ is small, which represents a $\log(N)$ dependence on the scale of the problem. Note that when the convexity structure is not exploited,

the dependence on N can be linear. We then design an enhanced adaptive sampling (EAS) algorithm that is proved to enjoy an upper bound on the asymptotic simulation cost as $O(\epsilon^{-2}\log(1/\delta))$. This upper bound does not depend on the problem scale N when ϵ and δ is small. This EAS algorithm essentially hits the best achievable performance (i.e., lower bound on simulation cost) and therefore provides a matching upper and lower bound on simulation costs for large-scale ranking and selection problems with general convex structure. This theoretical superiority of the EAS algorithm has also been verified by numerical experiments. On the contrary, gradient-based algorithms cannot achieve the performance of the EAS algorithm for large-scale one-dimensional problems.

Next, we turn to the settings of large-scale multi-dimensional problems with a d-dimensional discrete decision space as $\{1,2,\ldots,N\}\times\{1,2,\ldots,N\}\times\ldots\times\{1,2,\ldots,N\}$. We note that the scale N can easily be relaxed to be different in each dimension in our algorithm design, but we unify the use of N in each dimension in the analysis, so as to clearly demonstrate the impact of the scale N. A natural definition of discrete convexity on the multi-dimensional decision space is the $L^\natural$-convexity (Murota 2003), which guarantees that a local optimum is global optimal; see Dyer and Proll (1977), Freund et al. (2017), Zhang and Zheng (2020) for example. We first observe that even though the AS algorithm and the EAS algorithm in one-dimensional can be extended to the multi-dimensional case, the dependence of their simulation cost on the dimension d is too large (Agarwal et al. 2011, Liang et al. 2014), even up to an exponential order of dependence, which may prohibit their practical use in high-dimensional problems. This motivates us to develop new stochastic localization algorithms that have low dependence on the dimension d. Such algorithms take advantage of the subgradient information constructed by taking simulation samples, which plays a a crucial role in reducing the dependence of simulation cost on the dimension d. Specifically we develop stochastic cutting-plane (SCP) algorithms based on deterministic cutting-plane algorithms (Vaidya 1996, Bertsimas and Vempala 2004, Lee et al. 2015, Jiang et al. 2020), with the goal to achieve the PGS criterion. When evaluating the algorithm performance in terms of asymptotic simulation cost when the precision level ϵ and failing probability δ are small, in additional to analyzing the

dependence on dimension d and scale N, it is sometimes of interest to analyze the dependence on the Lipschitz parameter L. We develop a weakly polynomial SCP algorithm that has a $O(d^3)$ dependence on the dimension and a logarithm dependence on L. We then develop a strongly polynomial SCP algorithm that has a $O(d^4)$ dependence on the dimension but no dependence on L. This is the first algorithm that is dependence-free of the Lipschitz parameter L to solve general large-scale multi-dimensional discrete stochastic convex problems. In contrast, the gradient-based search algorithms developed in Zhang and Zheng (2020) has a quadratic dependence on L, on top of its lower $O(d^2)$ dependence on the dimension compared to stochastic SCP algorithms. Our developed SCP algorithms may particularly be preferable when the Lipschitz parameter L for a given problem is large or unknown. Numerically, it is shown that SCP algorithms have comparable performance to gradient-based algorithms in Zhang and Zheng (2020) and that SCP algorithms are suitable for the cases when the information about model parameters such as Lipschitz parameter is unknown a priori. In terms of dependence on the scale N, we show that the gradient-search algorithm and the strongly and weakly stochastic SCP algorithms all present an $O(N^2)$ dependence on the scale N for their simulation costs. On the other hand, the EAS algorithm, when extended to high-dimensional, still present no dependence on N, but however incurs an exponential dependence on d. These analyses can assist practitioners to choose which algorithm to use depending on the knowledge or partial knowledge on d, N and L in the specific problems.

We remark that algorithms that satisfy PGS criterion are our major focus. If, in addition, for scenarios when extra information on the indifference zone parameter $c > 0$ is available, i.e., the objective function value gap between the best decision and the second best is known, our algorithms can naturally be extended to identify the exact best decision variable with high probability $1 - \delta$. This criterion is referred to as *Probability of Correct Selection with Indifference Zone Parameter* (PCS-IZ). We also provide performance analysis for our proposed algorithms when they are used to achieve the PCS-IZ criterion.

We summarize the proved algorithm performances in Table 1, where the algorithm performance is demonstrated by the order of asymptotic simulation cost, when the failing probability δ and

the precision level ϵ are small and the problem scale N is large. A detailed numerical comparison of all algorithms that satisfy the PGS criterion are provided in Section 5 to show the consistency between our theoretical analysis and practical performance.

Two contemporaneous papers Zhang and Zheng (2020) and Eckman et al. (2020) also discuss the use of the convexity structure in simulation. Zhang and Zheng (2020) proposes gradient-based algorithms with PGS guarantee for general convex discrete simulation optimization problems with high-dimension decision space (i.e., d is large). The adaptive search algorithm, enhanced adaptive search algorithm and general stochastic cutting-plane algorithms introduced in our paper are based on a completely different idea that adaptively screens out inferior decision variables. Moreover, for scenarios when the problem scale N is large and the dimension $d = 1$, the enhanced adaptive search algorithm attains the best achievable performance and demonstrates an asymptotic simulation cost independent of N. One of our proposed stochastic cutting-plane algorithm completely does not depend on the Lipschitz parameter of the problem, while the gradient-based algorithms has a quadratic dependence on the Lipschitz parameter. In general, it appears that the gradient-based algorithms tend to be preferable for large d, while the stochastic cutting-plane algorithms tend to be preferable for large N and large Lipschitz parameter L, even though both sets of algorithms are designed to be generally applicable. The other contemporaneous paper Eckman et al. (2020) discusses the idea of using functional structure such as convexity to screen out inferior decision variables based on given simulations on other decision variables. Their focus is different. They do not discuss simulation optimization algorithms that can find the best or a statistically guaranteed good decision variable. Nor do they have analysis on simulation costs and the dependence on problem scale and dimension.

The remainder of the paper is outlined as follows. Section 1.1 summarizes the notation. Section 2 introduces the setup of model, framework, optimality criterion, and simulation costs. Section 3 discuss the algorithms and performance analysis developed for one-dimensional large-scale problems. Section 4 discuss the algorithms and performance analysis developed for multi-dimensional large-scale problems. Section 5 provide numerical experiments that support the theoretical findings. Section 6 concludes.

Table 1 Upper bounds on the expected simulation cost for algorithms that achieve the PGS or PCS-IZ guarantee. Here, M is an absolute constant. Constants except for $d, N, \epsilon, \delta, c, \gamma, M$ are omitted in the $O(\cdot)$ notation. In comparison, the expected simulation cost without convexity is $O(N^d \epsilon^{-2} \log(1/\delta))$.

Algorithms	**PGS**	**PCS-IZ**
Adaptive Sampling (One-dim)	$O(\log(N)\epsilon^{-2}\log(1/\delta))$	$O(c^{-2}\log(1/\delta))$
Enhanced Adaptive (One-dim)	$O(\epsilon^{-2}\log(1/\delta))$ (best achievable performance)	$O(c^{-2}\log(1/\delta))$ (best achievable performance)
Multi-dim Gradient-based (Pseudo-poly)	$O(d^2N^2\epsilon^{-2}\log(1/\delta))$ (Zhang and Zheng 2020)	$O(d^2\log(N)c^{-2}\log(1/\delta))$ (Zhang and Zheng 2020)
Stochastic Cutting-plane (Weakly-poly)	$O(d^3N^2\epsilon^{-2}\log(dNL/\epsilon)\log(1/\delta))$	$O(d^3\log(N)\epsilon^{-2}\log(dNL/\epsilon)\log(1/\delta))$
Stochastic Cutting-plane (Strongly-poly)	$O(d^3N^2(d+\log(N))\epsilon^{-2}\log(1/\delta))$	$O(d^3\log(N)(d+\log(N))\epsilon^{-2}\log(1/\delta))$
Enhanced Adaptive (Multi-dim)	$O(M^d\epsilon^{-2}\log(1/\delta))$	$O(M^dc^{-2}\log(1/\delta))$

1.1. Notation

We adopt the same set of notation as in (Zhang and Zheng 2020). For a stochastic system labeled by its decision variable x, we denote ξ_x as the random object associated with the system. We write $\xi_{x,1}, \xi_{x,2}, \ldots, \xi_{x,n}$ as independent and identically distributed (iid) copies of ξ_x. The empirical mean of the n independent evaluations for system labeled by x is denoted as $\hat{F}_n(x) := \frac{1}{n}\sum_{j=1}^{n} F(x, \xi_{x,j})$. The indices set $[N] := \{1, 2, \ldots, N\}$ is defined for every positive integer N. For any set S and positive integer d, we define the product set S^d as $\{(x_1, x_2 \ldots, x_d) : x_i \in \mathcal{S}, i \in [d]\}$. For two vectors $x, y \in \mathbb{R}^d$, the maximum operation $x \vee y$, minimum operation $x \wedge y$, the ceiling function $\lceil x \rceil$ and the flooring function $\lfloor x \rfloor$ are all considered as component-wise operations. To compare simulation

costs, we omit terms that are independent of $d, N, \epsilon, \delta, c$ in $O(\cdot)$ and omit terms independent of δ in $\tilde{O}(\cdot)$. To be more concrete, the notation $f = O(g)$ means that there exist constants $c_1, c_2 > 0$ independent of $N, d, \epsilon, \delta, c$ such that $f \leq c_1 g + c_2$. Similarly, the notation $f = \tilde{O}(g)$ means that there exist constants $c_1 > 0$ independent of $N, d, \epsilon, \delta, c$ and constant $c_2 > 0$ independent of δ such that $f \leq c_1 g + c_2$. In other words,

2. Model and Framework

We consider a complex stochastic system that involves discrete decision variables in a d-dimensional subspace $\mathcal{X} = [N_1] \times [N_2] \times \cdots \times [N_d]$ in which the N_i's are positive integers. The objective function $f(x)$ for $x \in \mathcal{X}$ is given by

$$f(x) \triangleq \mathbb{E}[F(x, \xi_x)], \tag{1}$$

in which ξ_x is a random object supported on a proper space $(\mathsf{Y}, \mathcal{B}_\mathsf{Y})$ and $F : \mathcal{X} \times \mathsf{Y} \to \mathbb{R}$ is a performance function. Specifically, the function F captures the full operations logic in the stochastic system. We consider scenarios when the objective function $f(x)$ is not in closed-form and needs to be evaluated by averaging over simulation replications of $F(x, \xi_x)$. The random objects ξ_x's can be different for different decision variables.

Assumption 1. *The objective function $f(x)$ is a convex function on the discrete set $\mathcal{X}$.*

For the exact definition of discrete convexity, we describe in details in Section 3 for the one-dimensional cases and Section 4 for the multi-dimensional cases.

2.1. Optimality Guarantees and Classes of Algorithms

Our general goal is to design algorithms that guarantees the selection of a good decision variable that has close-to-optimal performance with high probability. Formally, this criterion is defined as *Probability of Good Selection*.

- (ϵ, δ)**-Probability of good selection (PGS).** The solution x returned by an algorithm has objective value at most ϵ larger than the optimal objective value with probability at least $1 - \delta$.

This PGS guarantee is also referred to as the probably approximately correct selection (PAC) guarantee in the literature (Even-Dar et al. 2002, Kaufmann et al. 2016, Ma and Henderson 2017). While our major focus is to design algorithms that satisfy the PGS optimality guarantee, we also consider the optimality guarantee of *Probability of Correct Selection with Indifference Zone Parameter* as a comparison.

- **Probability of correct selection with indifference zone parameter (PCS-IZ).** (See Hong et al. (2020)) The problem is assumed to have a unique solution that renders the optimal objective value. The optimal value is assumed to be at least $c > 0$ smaller than other objective values. The gap width c is called the **indifference zone parameter** in Bechhofer (1954). The PCS-IZ guarantee requires that the solution returned by an algorithm is the optimal solution with probability at least $1 - \delta$.

In general, by choosing $\epsilon < c$, algorithms satisfying the PGS guarantee can be readily applied to satisfy the PCS-IZ guarantee. On the other hand, however, algorithms satisfying the PCS-IZ guarantee may fail to satisfy the PGS guarantee; see Eckman and Henderson (2018) and Hong et al. (2020). The failing probability δ in either PGS or PCS-IZ is usually chosen to be small to ensure a high probability result. We assume in this paper that δ is small.

ASSUMPTION 2. *The failing probability δ is sufficiently small.*

In addition, we assume that the probability distribution for the stochastic simulation output $F(x, \xi_x)$ is Gaussian or sub-Gaussian.

ASSUMPTION 3. *The distribution of $F(x, \xi_x)$ is Gaussian or sub-Gaussian with known upper bound σ^2 on the variance for any $x \in \mathcal{X}$.*

The triad of the decision set $\mathcal{X}$, the space of randomness $(\Upsilon, \mathcal{B}_\Upsilon)$ and the function $F(\cdot, \cdot)$ is called the **model** of problem (1). We define the set of all models such that function $f(\cdot)$ is convex on set $\mathcal{X}$ as $\mathcal{MC}(\mathcal{X})$, or simply $\mathcal{MC}$. The set $\mathcal{MC}_c$ includes all convex models with IZ parameter c. Next, we define the class of simulation algorithms that are proved to find solutions satisfying certain optimality guarantee for a given set of models.

DEFINITION 1. Suppose the optimality guarantee $\mathcal{O}$ and the set of models $\mathcal{M}$ is given. A simulation algorithm is called an $(\mathcal{O}, \mathcal{M})$-algorithm, if for any model $M \in \mathcal{M}$, the algorithm returns a solution to M that satisfies the optimality guarantee $\mathcal{O}$.

For example, the class of $(\text{PGS}, \mathcal{MC})$-algorithms guarantees a PGS solution for any convex model.

2.2. Simulation Costs

For simulation optimization optimization problems, the view that the simulation cost of generating replications of $F(x, \xi_x)$ is the dominant contributor to the computational cost is widely hold; see Luo et al. (2015), Ni et al. (2017), Ma and Henderson (2019). Therefore, for the purpose of comparing different simulation algorithms that satisfy certain optimality guarantee, the performance of different algorithms is quantified as the total number of evaluations of $F(x, \xi_x)$ at different x. The number of evaluations during a solving process is called the simulation cost. Besides providing a measurement to compare different algorithms, simulation costs can provide insights on how the computational cost depends on the scale and dimension of the problem. Moreover, understanding the simulation costs can provide information to facilitate the setup of parallel procedures for large-scale problems. The main focus of this paper is to develop provable simulation algorithms for a certain optimality guarantee and provide an upper bound on the simulation cost to achieve that guarantee. We note that our proposed algorithms do not require additional structures of the selection problem in addition to convexity. Now, we give the rigorous definition of the expected simulation cost for a given set of models $\mathcal{M}$ and given optimality guarantee $\mathcal{O}$.

DEFINITION 2. Given the optimality guarantee $\mathcal{O}$ and a set of models $\mathcal{M}$, the **expected simulation cost** is defined as

$$T(\mathcal{O}, \mathcal{M}) := \inf_{\mathbf{A} \text{ is } (\mathcal{O}, \mathcal{M})} \sup_{M \in \mathcal{M}} \mathbb{E}[\tau],$$

where τ is the number of simulation evaluations of $F(\cdot, \cdot)$ for each algorithm implementation.

The notion of simulation cost in this paper is largely focused on

$$T(\epsilon, \delta, \mathcal{MC}) := T((\epsilon, \delta)\text{-}PGS, \mathcal{MC}) \quad T(\delta, \mathcal{MC}_c) := T((c, \delta)\text{-}PCS\text{-}IZ, \mathcal{MC}_c).$$

We mention that upper bounds derived in this paper also hold almost surely, while lower bounds only hold in expectation.

To better present the dependence of expected simulation cost on the scale and dimension of the problem, we assume that $N_1 = N_2 = \cdots = N_d$.

ASSUMPTION 4. *The feasible set of decision variables is $\mathcal{X} = [N]^d$, where $N \geq 2$ and $d \geq 1$.*

With Assumption 4 in hand, we will present the dependence of the expected simulation on N and d when at least one of them is large. We note that the results in this work can be naturally extended to the case when each dimension has different number of feasible systems. Furthermore, if the objective function f is defined on the cube $[N_1] \times [N_2] \times \cdots \times [N_d]$ while the feasible set $\mathcal{X}$ characterized by a polytope $\{x \in \mathbb{Z}^d : Ax \leq b\}$, the algorithms proposed in this paper can be directly extended with few modification.

3. Simulation Algorithms and Complexity Analysis: One-dimensional Case

We first consider the case when the discrete decision variable is one-dimensional, i.e., $\mathcal{X} = [N] = \{1, 2, \ldots, N\}$. This setting is mathematically equivalent to the problem of ranking and selection with convexity. We have in mind cases where the number of systems N is large. In the one-dimensional case, the discrete convexity for a function f can be defined similarly as the ordinary continuous convexity through the discrete midpoint convexity property, namely,

$$f(i+1) + f(i-1) \geq 2f(i) \quad \forall i \in \{2, \ldots, N-1\}.$$

If the function $f(x)$ is convex on $\mathcal{X}$, it has a convex linear interpolation on $[1, N]$, defined as

$$\tilde{f}(x) := [f(x_{i+1}) - f(x_i)] \cdot (x - x_i) + f(x_i), \quad \forall x \in [x_i, x_{i+1}],\ \forall i \in [N-1]. \tag{2}$$

The definition of discrete convexity in multi-dimensional decision space is called $L^\natural$-convexity (Murota 2003). We defer the discussion of $L^\natural$-convex functions for multi-dimensional case to section 4.

In this section, we propose simulation algorithms that are guaranteed to find solutions that satisfy the PGS criterion, when the objective function has a convex structure. The presumption is that there is no further structure information available to the algorithm. For every proposed simulation algorithm, we provide an upper bound on the expected simulation cost to achieve the PGS guarantee. We demonstrate the dependence of the upper bound on the precision level ϵ, the failing probability allowance δ, and the number of systems N. We also provide a lower bound on the expected simulation cost that reflects the best achievable performance for any algorithm. We show that one of our proposed algorithms can attain the best achievable performance.

3.1. Adaptive Sampling Algorithm and Upper Bound on Expected Simulation Cost

We first propose an adaptive sampling algorithm for the PGS guarantee. The idea of adaptive sampling algorithm is from the classical bi-section method. Agarwal et al. (2011) propose a stochastic bi-section method for stochastic continuous convex optimization, which is different from ours. The pseudo-code is listed in Algorithm 1. We discuss the details right afterwards.

Algorithm 1 Adaptive sampling algorithm for PGS guarantee

Input: Model $\mathcal{X}=[N], \mathcal{B}_x, F(x,\xi_x)$, optimality guarantee parameters ϵ, δ.

Output: An (ϵ,δ)-PGS solution x^* to problem (1).

1: Set upper and lower bound of the current interval $L \leftarrow 1, U \leftarrow N$.

2: Set maximal number of comparisons $T_{max} \leftarrow \log_{1.5}(N)+2$.

3: **while** $U-L>2$ **do** ▷ Iterate until there are at most 3 points.

4: Compute 3-quantiles $N_{1/3} \leftarrow \lfloor 2L/3+U/3 \rfloor$ and $N_{2/3} \leftarrow \lceil L/3+2U/3 \rceil$.

5: **repeat** simulate an independent copy of $F(N_{1/3},\xi_{1/3})$ and an independent copy of $F(N_{2/3},\xi_{2/3})$

6: Compute the empirical mean (using all the simulated samples) $\hat{F}_n(N_{1/3}), \hat{F}_n(N_{2/3})$.

7: Compute $1-\delta/(2T_{max})$ confidence intervals at each quantile:

$$\left[\hat{F}_n(N_{1/3})-h_{1/3}, \hat{F}_n(N_{1/3})+h_{1/3}\right] \quad \text{and} \quad \left[\hat{F}_n(N_{2/3})-h_{2/3}, \hat{F}_n(N_{2/3})+h_{2/3}\right].$$

until the first time that one of the following three conditions holds:

$$(i) \quad \hat{F}_n(N_{1/3}) - h_{1/3} \geq \hat{F}_n(N_{2/3}) + h_{2/3},$$

$$(ii) \quad \hat{F}_n(N_{1/3}) + h_{1/3} \leq \hat{F}_n(N_{2/3}) - h_{2/3},$$

$$(iii) \quad h_{1/3} \leq \epsilon/8 \quad \text{and} \quad h_{2/3} \leq \epsilon/8.$$

if $\hat{F}_n(N_{1/3}) - h_{1/3} \geq \hat{F}_n(N_{2/3}) + h_{2/3}$ **then**

    Update $L \leftarrow N_{1/3}$.

else if $\hat{F}_n(N_{1/3}) + h_{1/3} \leq \hat{F}_n(N_{2/3}) - h_{2/3}$ **then**

    Update $U \leftarrow N_{2/3}$.

else if $h_{1/3} \leq \epsilon/8$ and $h_{2/3} \leq \epsilon/8$ **then**

    Update $L \leftarrow N_{1/3}$ and $U \leftarrow N_{2/3}$.

end if

end while

Simulate $F(x, \xi_x)$ for $x \in \{L, \ldots, U\}$ until the $1 - \delta/(2T_{max})$ confidence widths are smaller than $\epsilon/2$.

▷ Now $U - L \leq 2$.

Return the point in $\{L, \ldots, U\}$ with minimal empirical mean.

In the procedure of Algorithm 1, one step is to compute confidence intervals that satisfy certain confidence guarantees. We now provide one feasible approach to construct such confidence intervals, which is based on Hoeffding's inequality for Gaussian and sub-Gaussian random variables. Define

$$h(n, \sigma, \alpha) := \sqrt{\frac{2\sigma^2}{n} \cdot \log(2/\alpha)}.$$

Recall that σ^2 is the upper bound of the variances of all systems. With this function $h(\cdot)$ in hand, whenever n independent simulations of system x is available, one can construct a $1 - \alpha$ confidence interval for $f(x)$ as

$$\left[\hat{F}_n(x) - h(n, \sigma, \alpha), \hat{F}_n(x) + h(n, \sigma, \alpha)\right].$$

If the variance σ_x^2 of a single system x is known, the confidence interval may be sharpened by replacing σ with σ_x.

Intuitively, the algorithm iteratively shrinks the size of the set containing a potential good decision. Specifically, the algorithm shrinks the length of current interval by at least 1/3 for each iteration. Thus, the total number of iterations is at most $O(\log_{1.5}(N))$ to shrink the set containing a potential good decision with at least 3 points. Then, the algorithm solves a sub-problem with at most 3 points. We can prove that Algorithm 1 achieves PGS guarantee for any given convex problem without knowing further structure information, i.e., Algorithm 1 is a $[(\epsilon,\delta)\text{-PGS}, \mathcal{MC}]$-algorithm. By estimating the simulation cost of the algorithm, an upper bound on the expected simulation cost to achieve the PGS guarantee follows.

THEOREM 1. *Suppose Assumptions 1-4 hold. Algorithm 1 is a* $[(\epsilon,\delta)$*-PGS,* $\mathcal{MC}]$*-algorithm. Furthermore, we have*

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

Proof of Theorem 1. The proof of Theorem 1 is given in EC.2.1. □

We provide an explanation on the additional hidden term of smaller order of $\log(1/\delta)$. In fact, the upper bound on the expected simulation cost is given by a constant multiplied by the sum $\log(N)\epsilon^{-2}\log(1/\delta)+\log(N)$. We note that in practice, the number of simulation samples taken in each iteration must be an integer, while in analysis, the simulation cost is treated as a real number. Hence, the practical simulation cost of each iteration should be the smallest integer larger than the the theoretical simulation cost and an extra $O(1)$ term is introduced. Then, the total expected simulation cost of Algorithm 1 should contain an extra $O(\log(N))$ term, which is not related to δ and is relatively small compared to the main term when δ is small.

We next discuss an extension to Algorithm 1 under application scenarios when the simulation of two very close systems can be simultaneously done at the cost of simulating one system. Specifically, when a copy of simulation for system x is generated, with its outcome denoted as $F(x,\xi_{x,1})$, a

copy of simulation for system $x+1$ (or $x-1$) can be generated at the same time with a negligible additional cost, with its outcome denoted as $F(x+1,\xi_{x+1,1})$. It is usually the case that the two outcomes $(F(x,\xi_{x,1}),F(x+1,\xi_{x+1,1}))$ are positively correlated. Therefore, throughout the procedure of implementing Algorithm 1, whenever n independent copies of simulation are done for system x and an empirical mean estimator $\hat{F}_n(x)$ is obtained, we also have available an empirical mean estimator $\hat{F}_n^{\text{free}}(x+1)$ for system $x+1$ that is positively correlated with $\hat{F}_n(x)$. Intuitively, with the additional free information on system $x+1$, Algorithm 1 can be terminated early if the estimated performances for the two neighboring systems are close enough. Rigorously, Algorithm 1 can be terminated early if

$$\hat{F}_n(x)+h_x \le \hat{F}_n^{\text{free}}(x+1)-h_{x+1}^{\text{free}}+\epsilon/N \quad \text{and} \quad \hat{F}_n^{\text{free}}(x+1)+h_{x+1}^{\text{free}} \le \hat{F}_n(x)-h_x+\epsilon/N,$$

where h_x and h_{x+1}^{free} are half-widths of confidence intervals with at least $1-\delta/2$ confidence. One convenient choice of h_x and h_{x+1}^{free} is to set $h_x = h_{x+1}^{\text{free}} = h(n,\sigma,\delta/2)$. If the positive correlation between $\hat{F}_n(x)$ and $\hat{F}_n^{\text{free}}(x)$ is known, the half-widths can be sharpened. The legitimacy of this termination rule is because of the convex structure. For any system $x \in \mathcal{X}$, the local and global expected performances satisfy the relation that

$$|f(x)-f(x+1)| < \frac{\epsilon}{N} \implies |f(x)-f(x^*)| < \epsilon,$$

where x^* is the label of the unknown best system. On one hand, Algorithm 1 is clearly enhanced by adding this termination rule. On the other hand, we note that for the termination rule to be triggered, it is required that $h(n,\sigma,\delta/2) < \epsilon/N$, or equivalently

$$n > 2\sigma^2\epsilon^{-2}N^2\log(4/\delta),$$

which means that a large number of simulation copies are needed to trigger this termination rule, with a quadratic dependence on the total number of systems N. Recall that the upper bound for the expected simulation cost of Algorithm 1 has a logarithm dependence on N. Therefore, this

additional termination rule benefits the algorithm, but likely will not improve the order of upper bound for the expected simulation cost when the number of systems N is large.

Next, we consider the PCS-IZ guarantee. When the priori information about indifference zone parameter c is available, we can modify the adaptive sampling algorithm to achieve better simulation cost. The modified algorithm also consists of two parts: the shrinkage of intervals and a sub-problem with at most 3 points. The improvement is achieved by a weaker condition for the comparison of objective values at two 3-quantiles. We give the modified algorithm in EC.1.1. The following theorem gives the correctness and the expected simulation cost of the modified adaptive sampling algorithm.

THEOREM 2. *Suppose Assumptions 1-4 hold. The modified adaptive sampling algorithm is a* $[(c,\delta)$*-PCS-IZ,*$\mathcal{MC}_c]$*-algorithm. Furthermore, we have*

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

Proof of Theorem 2. The proof of Theorem 2 is given in EC.2.2. □

By Theorem 2, the expected simulation cost for the PCS-IZ guarantee is independent of the number of points N. In comparison, the cost of bi-section method on deterministic convex problem has dependency $O(\log_2(N))$ on N. Hence, stochastic problems have different properties compared with deterministic problems. Optimal algorithms for deterministic problems cannot be naively adjusted to stochastic problems to achieve the optimal expected simulation cost. This intuition is also observed in Zhang and Zheng (2020), where fundamental differences between simulation optimization algorithms and deterministic optimization algorithms are observed.

3.2. Enhanced Adaptive Sampling Algorithm and Upper Bound on Expected Simulation Cost

We have shown that the expected simulation cost of adaptive sampling algorithm for the PGS guarantee has a $\log(N)$ dependence on N for large N. Then, one may naturally ask: is there any algorithm for the PGS guarantee that has simulation cost independent of N? The answer is affirmative. In this subsection, an enhanced adaptive sampling algorithm for the PGS guarantee

is proposed and is proved to have asymptotic simulation cost independent of N. Similar to the adaptive sampling algorithm, the enhanced adaptive sampling algorithm maintains a set of active points and shrinks the set in each iteration until there are at most 2 points. However, instead of only sampling at 3-quantiles points of the current interval, the enhanced adaptive sampling algorithm samples all points in the current active set. We give the pseudo-code in Algorithm 2.

Algorithm 2 Enhanced adaptive sampling algorithm for PGS guarantee

Input: Model $\mathcal{X} = [N], \mathcal{B}_x, F(x, \xi_x)$, optimality guarantee parameters ϵ, δ.

Output: An (ϵ, δ)-PGS solution x^* to problem (1).

Set the active set $\mathcal{S} \leftarrow \mathcal{X}$.

Set the step size $d \leftarrow 1$, maximal number of comparisons $T_{max} \leftarrow N$.

while the size of $\mathcal{S}$ is at least 3 **do** ▷ Iterate until $\mathcal{S}$ has at most 2 points.

  repeat simulate an independent copy of $F(x, \xi_x)$ for all $x \in \mathcal{S}$

    Compute the empirical mean (using all the available simulated samples) $\hat{F}_n(x)$ for all $x \in \mathcal{S}$.

    Compute $1 - \delta/(2T_{max})$ confidence intervals at each quantile:

$$[\hat{F}_{n_x}(x) - h_x, \hat{F}_{n_x}(x) + h_x] \quad \forall x \in S.$$

    if the confidence width $h_x \leq |\mathcal{S}| \cdot \epsilon/80$ for some $x \in S$ **then**

      Stop sampling point x.

    end if

  until at least one of the following conditions holds

$$(i) \quad \exists x, y \in \mathcal{S} \quad \text{s.t.} \quad \hat{F}_{n_x}(x) + h_x \leq \hat{F}_{n_y}(y) - h_y$$

$$(ii) \quad \forall x \in \mathcal{S} \qquad h_x \leq |\mathcal{S}| \cdot \epsilon/80.$$

  if $\hat{F}_{n_x}(x) + h_x \leq \hat{F}_{n_y}(y) - h_y$ for some $x, y \in \mathcal{S}$ **then** ▷ **Type-I Operation**

    if $x < y$ **then**

      Remove all points $z \in \mathcal{S}$ such that $z \geq y$ from $\mathcal{S}$.

14: **else**

15: Remove all points $z \in \mathcal{S}$ such that $z \leq y$ from $\mathcal{S}$.

16: **end if**

17: **else if** $h_x \leq |\mathcal{S}| \cdot \epsilon/80$ for all $x \in \mathcal{S}$ **then** ▷ **Type-II Operation**

18: Update the step size $d \leftarrow 2d$.

19: Update $\mathcal{S} \leftarrow \{x_{min}, x_{min}+d, \ldots, x_{min}+kd\}$, where $x_{min} = \min_{x\in\mathcal{S}} x$ and $k = \lceil |\mathcal{S}|/2 \rceil - 1$.

20: **end if**

21: **end while** ▷ Now $\mathcal{S}$ has at most 2 points.

22: Simulate $F(x, \xi_x)$ for $x \in \mathcal{S}$ until the $1 - \delta/(2T_{max})$ confidence widths are smaller than $\epsilon/4$.

23: Return the point in $\mathcal{S}$ with minimal empirical mean.

There are two kinds of shrinkage operations in Algorithm 2, which we denote as Type-I and Type-II Operations. Intuitively, Type-I Operations are implemented when we can compare and differentiate the function values of two points with high probability, and Type-II Operations are implemented when all points have similar function values. In the latter case, we prove that there exists a neighboring point to the optimum that has function value at most $\epsilon/2$ larger than the optimum. Hence, we can discard every other point in $\mathcal{S}$ (the set in the algorithm that contains a potential good selection) with at least one $\epsilon/2$-optimal point remained in the active set. We give a rough estimate to the expected simulation cost of Algorithm 2. We assign an order to points in $\mathcal{X}$ by the time they are discarded from $\mathcal{S}$. Points discarded in the same iteration are ordered randomly. Then, for the last k-th discarded point x_k, there are at least k points in $\mathcal{S}$ when x_k is discarded. By the second termination condition in line 11, the confidence width at x_k is at least $k\epsilon/80$. If Hoeffding bound is used, simulating $\tilde{O}(\epsilon^{-2}k^{-2}\log(1/\delta))$ times is enough to achieve the confidence width. Recalling the fact that $\sum_k k^{-2} < \pi^2/6 < \infty$, if we sum the simulation cost over $k \in [N]$, the total expected simulation cost is bounded by $\tilde{O}(\epsilon^{-2}\log(1/\delta))$ and is independent of N. The following theorem proves that Algorithm 2 indeed achieves PGS guarantee for any convex problem and provides a rigorous upper bound on the expected simulation cost $T(\epsilon, \delta, \mathcal{MC})$.

THEOREM 3. *Suppose Assumptions 1-4 hold. Algorithm 2 is a zeroth-order* $[(\epsilon,\delta)$*-PGS*$,\mathcal{MC})]$*-algorithm. Furthermore, we have*

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

Proof of Theorem 3. The proof of Theorem 3 is given in EC.2.3. □

We can see that the expected simulation cost of the enhanced adaptive sampling algorithm for the PGS guarantee is independent of N no matter how large N is. If used for the PCS-IZ guarantee, the enhanced adaptive sampling algorithm achieves $\tilde{O}(c^{-2}\log(1/\delta))$ expected simulation cost by setting parameter $\epsilon = c/2$. This is because all sub-optimal solutions have function value at least c larger than the optimal solution and the solution satisfies $(c/2,\delta)$-PGS guarantee also satisfies the (c,δ)-PCS-IZ guarantee. Hence, for both PGS and PCS-IZ guarantees, we get upper bounds for expected simulation cost that are independent of N.

3.3. Lower Bound on Expected Simulation Cost

In this subsection, we give lower bounds to the expected simulation costs for all simulation algorithms that satisfy certain optimality guarantee. Zhang and Zheng (2020) proved a lower bound for the PGS guarantee and therefore we focus on lower bounds for the PCS-IZ guarantee in this work. The lower bounds suggest the limits for all simulation algorithms for general selection problems with convex structure. In the one-dimensional case, the derived lower bound for the PCS-IZ guarantee also holds for the PGS guarantee by choosing $c = 2\epsilon$. By comparing with upper bounds established for specific simulation algorithms, we can conclude that the enhanced adaptive sampling algorithm is optimal up to a constant factor. Similar to the method used in Zhang and Zheng (2020), we construct two convex models that have similar distributions at each point but have distinct optimal solutions. Then, the information-theoretical inequality in Kaufmann et al. (2016) can be used to provide a lower bound of zeroth-order algorithms.

We present the results in Kaufmann et al. (2016) for the completeness. 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 sampled when

the algorithm terminates, where τ is the stopping time of the algorithm. Then, it follows from the definition that

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

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 is proved in Kaufmann et al. (2016) and is the major tool for deriving lower bounds in this paper.

LEMMA 1. *For any two models M_1, M_2 and any event $\mathcal{E} \in \mathcal{F}_\tau$, we have*

$$\sum_{x \in \mathcal{X}} \mathbb{E}_{M_1}\left[N_x(\tau)\right] \mathrm{KL}(\nu_{1,x}, \nu_{2,x}) \geq d(\mathbb{P}_{M_1}(\mathcal{E}), \mathbb{P}_{M_2}(\mathcal{E})), \tag{3}$$

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

We first give a lower bound for the PCS-IZ guarantee.

THEOREM 4. *Suppose Assumptions 1-4 hold. We have*

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

Proof of Theorem 4. The proof of Theorem 4 is given in EC.2.4. □

The lower bound of $T(\epsilon, \delta, \mathcal{MC})$, i.e., the expected simulation cost for achieving the PGS guarantee, can be derived in the same way by substituting c with 2ϵ in the construction of two models.

COROLLARY 1. *Suppose Assumptions 1-4 hold. We have*

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

Combining with the upper bounds we derived in Sections 3.1 and 3.2, we know the adaptive sampling algorithm is optimal up to a constant for the PCS-IZ guarantee, while having a $\log(N)$ order gap for the PGS-IZ guarantee. On the other hand, the enhanced adaptive sampling algorithm is optimal for both guarantees up to a constant. However, the space complexities of the adaptive

sampling algorithm and the enhanced adaptive sampling algorithms are $O(\log(N))$ and $O(N)$, respectively. This observation implies that the adaptive sampling algorithm is preferred for the PCS-IZ guarantee, while for the PGS guarantee, we need to consider the trade-off between the simulation cost and the space complexity when choosing algorithms. Before concluding this section, we note that our simulation algorithms are designed to ensure PGS/PCS-IZ guarantees for large-scale convex problems whose detailed structure is hard to detect a priori. If a specific simulation problem has more refined structure, the simulation algorithms may be adjusted to achieve better expected simulation costs under the same guarantees.

4. Simulation Algorithms and Complexity Analysis: Multi-dimensional Case

In this section, we propose stochastic cutting-plane algorithms to achieve the PGS guarantee for discrete convex simulation optimization problems with multi-dimensional decision variables. The decision space is considered as $\mathcal{X} = [N]^d$. We have in mind cases where both the problem scale N and the dimension d are large. In the multi-dimensional case, the discrete convexity of f is defined by the so-called $L^\natural$-convexity, which is defined on the mid-point convexity for discrete variables. The exact definition will be given shortly in Section 4.1. The $L^\natural$-convexity can lead to a property that the discrete convex function has a convex extension along with an explicit subgradient defined on the convex hull $[1, N]^d$. We next discuss a brief outline of results in this section before discussing the details. A first natural idea is to extend the adaptive sampling algorithm developed in Section 3 to the multi-dimensional case. A direct generalization of the adaptive sampling algorithm results in the zeroth-order stochastic ellipsoid method (Agarwal et al. 2011) and the zeroth-order random walk method (Liang et al. 2014), whose computational complexities have $O(d^{33})$ and $O(d^{14})$ dependence on the dimension, respectively. On the other hand, we show that the enhanced adaptive sampling method can be naturally extended to the multi-dimensional case. The multi-dimensional enhanced adaptive sampling algorithm also has expected simulation cost independent of scale N when N is large. However, the expected simulation cost has exponential dependence on the dimension d and therefore the enhanced adaptive sampling algorithm is only suitable for low-dimensional

problems. The high order dependence on d by simply extending the one-dimensional algorithms to multi-dimensional problems may prohibit the practical use even for moderate-dimension problems. We therefore develop new stochastic cutting-plane algorithms, which utilize properties of $L^\natural$-convex functions and the Lovász extension, to evaluate unbiased stochastic subgradients at each point via finite difference. In contrast to the one-dimensional case, the stochastic subgradients information is proved to be useful in reducing the dependence of simulation cost on d. More specifically, our proposed stochastic cutting-plane algorithms are proved to have $O(d^3)$ dependence on d. In addition, the stochastic cutting-plane methods have only logarithm dependence on the Lipschitz constant L, while the gradient-based method in Zhang and Zheng (2020) has quadratic dependence on L. Furthermore, the discrete nature of problem (1) makes it possible to devise strongly polynomial algorithms, i.e., algorithms that do not require the knowledge of Lipschitz constant.

4.1. Discrete Convex Functions in Multi-dimensional Space

Similar to the one-dimensional case, discrete convex functions in multi-dimensional space are characterized by the discrete midpoint convexity property and have convex piecewise linear extension. In Murota (2003), this collection of functions is named as $L^\natural$-convex functions and is proved to have the property that local optimality implies global optimality. The exact definition of $L^\natural$-convex functions is given by

DEFINITION 3. A function $f(x) : \mathcal{X} \mapsto \mathbb{R}$ is called a $L^\natural$-convex function, if the discrete midpoint convexity holds:

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

The set of models such that $f(x)$ is $L^\natural$-convex on $\mathcal{X}$ is denoted as $\mathcal{MC}$. The set of models such that $f(x)$ is $L^\natural$-convex with indifference zone parameter c is denoted as $\mathcal{MC}_c$.

REMARK 1. As noted in Zhang and Zheng (2020), the feasible set $\mathcal{X} = [N]^d$ is a $L^\natural$-convex set and we only need the discrete midpoint convexity property to define $L^\natural$-convex functions on $\mathcal{X}$. In the case when $d = 1$, the $L^\natural$-convexity is equivalent to the discrete convexity defined in Section 3. Hence, the definitions of $\mathcal{MC}$ and $\mathcal{MC}_c$ are consistent with Section 3.

The following property shows that $L^\natural$-convex functions can be viewed as a generalization of submodular functions.

LEMMA 2 **(Murota (2003))**. *Suppose the function $f(x):\mathcal{X}\mapsto\mathbb{R}$ is $L^\natural$-convex. Then, the translation submodularity holds:*

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

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). Therefore, we can get 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 provides a good framework for extending continuous convex optimization theory to the discrete case. In the remaining of this subsection, we specify this intuition of $L^\natural$-convex function 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 4. Suppose function $f(x):\{0,1\}^d\mapsto\mathbb{R}$ is a submodular function. For any $x\in[0,1]^d$, we say a permutation $\alpha_x:[d]\mapsto[d]$ is a **consistent permutation** of x, if

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

For each $i\in\{0,1,\ldots,d\}$, 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

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

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

LEMMA 3. *Suppose Assumptions 1-4 hold. Then, the following properties of $\tilde{f}(x)$ hold.*

(i) For any $x \in \mathcal{X}$, it holds $\tilde{f}(x) = f(x)$.

(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) Function $\tilde{f}(x)$ is a convex function on $[0,1]^d$.

(iv) A subgradient $g \in \partial\tilde{f}(x)$ is given by

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

It is proved in Zhang and Zheng (2020) that, with $2d$ simulation runs, we can generate a stochastic subgradient at point x by

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

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

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

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 α_x as a consistent permutation of $x - y$ in $\{0,1\}^d$ and, for each $i \in \{0, 1, \ldots, d\}$, 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

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

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

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

It is proved in Murota (2003), Zhang and Zheng (2020) that $\tilde{f}(x)$ is well-defined and is a convex function.

THEOREM 5. *The function $\tilde{f}(x)$ in (7) is well-defined and is convex on $\mathcal{X}$.*

Utilizing properties (i) and (ii) of Lemma 3, problem (1) is equivalent to the relaxed problem

$$f^* := \min_{x \in [1,N]^d} \tilde{f}(x). \tag{8}$$

Moreover, the subgradient (5) and stochastic subgradient (6) are valid for the convex extension $\tilde{f}(x)$. Similarly, (stochastic) subgradients can be computed in the neighboring cube of each point and it does not matter which cube is chosen for points belonging to multiple cubes. Finally, the linear-time rounding process proposed in Zhang and Zheng (2020) reduces the problem of finding PGS solutions of the original problem to that of the relaxed problem.

Algorithm 3 Rounding process to a feasible solution

Input: Model $\mathcal{X}, \mathcal{B}_x, F(x, \xi_x)$, optimality guarantee parameters ϵ, δ, $(\epsilon/2, \delta/2)$-PGS solution $\bar{x}$ to problem (8).

Output: An (ϵ, δ)-PGS solution x^* to problem (1).

1: Compute a consistent permutation of $\bar{x}$, denoted as α.

2: Compute the neighbouring points of $\bar{x}$, denoted as $S^0, \ldots, S^d$.

3: Simulate $F(S^0, \xi_0), \ldots, F(S^d, \xi_d)$ until the $1 - \delta/4$ confidence width is smaller than $\epsilon/4$.

4: Return the point x^* with minimal empirical mean.

The following theorem verifies the correctness and estimates the simulation cost of Algorithm 3.

THEOREM 6. *Suppose Assumptions 1-4 hold. The solution returned by Algorithm 3 satisfies the (ϵ, δ)-PGS guarantee and the simulation cost of Algorithm 3 is at most*

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

The rounding process for the (c, δ)-PCS-IZ guarantee follows by choosing $\epsilon = c/2$.

4.2. Stochastic Cutting-plane Method: Weakly Polynomial Algorithm

With the definitions and tools introduced in Section 4.1, we have a suitable framework for extending deterministic cutting-plane methods to the stochastic case. In this subsection, we develop weakly polynomial stochastic cutting-plane methods for discrete convex simulation optimization. Weakly polynomial algorithms guarantee an (ϵ, δ)-PGS solution within $\text{poly}(d, N, \epsilon, \delta, \log(L))$ times of simulations and arithmetic operations, where L is the Lipschitz constant and is assumed known a priori.

ASSUMPTION 5. *The ℓ_∞-Lipschitz constant of $f(x)$ is L. Namely, we have*

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

Algorithms that guarantee an (ϵ, δ)-PGS solution within $\text{poly}(d, N, \epsilon, \delta)$ times of simulations and arithmetic operations are called strongly polynomial algorithms and are proposed in Section 4.3. In comparison, gradient-based methods proposed in Zhang and Zheng (2020) are pseudo-polynomial algorithms, since they require $\text{poly}(d, N, \epsilon, \delta, L)$ simulation runs and arithmetic operations. Generally, cutting-plane methods based on separation oracles (Vaidya 1996, Bertsimas and Vempala 2004, Lee et al. 2015) have lower-order dependence on the problem dimension than those based on membership oracles (Agarwal et al. 2011, Liang et al. 2014). Compared to gradient-based methods, cutting-plane methods are more sensitive to noise. Hence, more simulation runs are needed to generate robust separation oracles and therefore the simulation cost has a higher-order dependence on the problem dimension than gradient-based methods.

Now, we show that the cutting-plane method in Vaidya (1996) can be extended to find PGS solutions in the stochastic case. Vaidya's cutting-plane method maintains a polytope that contains the optimal points and iteratively reduces the volume of polytope by generating a separation oracle at the approximate volumetric center. When the volume is small enough, the Lipschitz continuity implies the all points in the polytope have close objective values to the optimum. Vaidya's method is able to achieve the optimal number of separation oracles evaluations up to a constant in the

deterministic case. We note that other cutting-plane methods based on reducing the volume of polytope can also be extended to our case. We consider Vaidya's method for its simplicity. As a counterpart of separation oracles, we introduce the stochastic separation oracle, namely the (ϵ,δ)-separation oracle, to characterize the accuracy of separation oracles in the stochastic case.

DEFINITION 5. An **(ϵ,δ)-separation oracle ((ϵ,δ)-$\mathcal{SO}$)** is a function on $[1,N]^d$ that for any input $x\in[1,N]^d$, it outputs a vector $\hat{g}\in\mathbb{R}^d$ such that for any $y\in[1,N]^d\cap H^c$,

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

holds with probability at least $1-\delta$, where the half space $H:=\{z:\langle\hat{g},z-x\rangle\leq 0\}$.

Before we state the algorithm, we give a concrete example of (ϵ,δ)-$\mathcal{SO}$ oracle and provide an upper bound of the expected simulation cost of evaluating each oracle. We define the averaged subgradient estimator as

$$\hat{g}^n_{\alpha_x(i)}:=\hat{F}_n\left(S^{x,i}\right)-\hat{F}_n\left(S^{x,i-1}\right)\quad\forall i\in[d],\tag{9}$$

where α_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 of n such that $\hat{g}^n$ is an (ϵ,δ)-$\mathcal{SO}$ oracle.

LEMMA 4. *Suppose Assumptions 1-4 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 (ϵ,δ)-$\mathcal{SO}$ oracle. Moreover, the expected simulation cost of generating an (ϵ,δ)-$\mathcal{SO}$ oracle is at most

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

Proof of Lemma 4. The proof of Lemma 4 is given in EC.3.1. □

We prove that, by substituting the separation oracles with stochastic separation oracles, the cutting-plane method in Vaidya (1996) can be used to find PGS solutions. The pseudo-code of the stochastic Vaidya's method is given in Algorithm 4.

Algorithm 4 Stochastic Vaidya's method for PGS guarantee

Input: Model $\mathcal{X}, \mathcal{B}_x, F(x, \xi_x)$, optimality guarantee parameters ϵ, δ, Lipschitz constant L, (ϵ, δ)-$\mathcal{SO}$ oracle $\hat{g}$.

Output: An (ϵ, δ)-PGS solution x^* to problem (1).

Set the initial polytope $P \leftarrow [1, N]^d$.
Set the number of iterations $T_{max} \leftarrow O[d \log(dLN/\epsilon)]$.
Initialize the set of points used to query separation oracles $\mathcal{S} \leftarrow \emptyset$.
Initialize the volumetric center $z \leftarrow (N+1)/2 \cdot (1, 1, \ldots, 1)^T$.
for $T = 1, 2, \ldots, T_{max}$ **do**
    Decide adding or removing a cutting plane by Vaidya's method.
    if add a cutting plane **then**
        Evaluate the $(\epsilon/8, \delta/4)$-$\mathcal{SO}$ oracle $\hat{g}$ at z.
        if $\hat{g} = 0$ **then**
            Round z to an integral solution by Algorithm 3 and return the rounded solution.
        end if
        Add the current point z to $\mathcal{S}$.
    else if remove a cutting plane **then**
        Remove corresponding point z from $\mathcal{S}$.
    end if
    Update the approximate volumetric center z by Newton-type method.
end for ▷ There are at most $O(d)$ points in $\mathcal{S}$ by Vaidya's method.
Find an $(\epsilon/4, \delta/4)$-PGS solution $\hat{x}$ of problem $\min_{x \in \mathcal{S}} f(x)$.
Round $\hat{x}$ to an integral solution by Algorithm 3.

We note that if the approximate volumetric center z is not in $[1, N]^d$, then we choose a violated constraint $x_i \geq 1$ or $x_i \leq N$ and return e_i or $-e_i$ as the separating vector, respectively. For arithmetic operations, each iteration of Algorithm 4 requires $O(d)$ inversions and multiplications of $d \times d$

matrices. Each inversion and multiplication can be finished within $O(d^\omega)$ arithmetic operations, where $\omega < 2.373$ is the matrix exponent (Alman and Williams 2020). Hence, Algorithm 4 needs $O(d^{\omega+1})$ arithmetic operations for each iteration. The correctness and the expected simulation cost of Algorithm 4 are provided in the following theorem.

THEOREM 7. *Suppose Assumptions 1-5 hold. Algorithm 4 returns an (ϵ, δ)-PGS solution and we have*

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

Proof of Theorem 7. The proof of Theorem 7 is given in EC.3.2. □

REMARK 2. We note that the random walk method in Bertsimas and Vempala (2004) can achieve a better expected simulation cost

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

at the expense of $O[d^6 + \log^2(1/\delta)]$ arithmetic operations in each iteration. Here, the $O[\log^2(1/\delta)]$ factor is required to ensure the high-probability approximation to the centroid. Also, we note that the fast implementation of Vaidya's method in Jiang et al. (2020) reduces number of arithmetic operations in each iteration to $O(d^2)$.

REMARK 3. Stochastic Vaidya's method can also be applied to problems that defined on $[N]^d$ with linear constraints $\{x \in \mathbb{Z}^d : Ax \le b\}$, since we can choose the initial polytope to be $[1, N]^d \cap \{Ax \le b\}$.

Using the same adaptive acceleration scheme as in Zhang and Zheng (2020), the indifference zone parameter can help reducing the dependence of simulation cost on problem scale N. We give the pseudo-code of the accelerated Vaidya's method in EC.1.2. Similar to Zhang and Zheng (2020), we can prove the correctness and estimate the expected simulation cost of the accelerated algorithm.

THEOREM 8. *Suppose Assumptions 1-5 hold. The accelerated stochastic Vaidya's method returns a (c, δ)-PCS-IZ solution and we have*

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

4.3. Stochastic Cutting-plane Method: Strongly Polynomial Algorithm

In this subsection, we develop strongly polynomial algorithms for the PGS guarantee, i.e., algorithms that do not require the knowledge of the Lipschitz constant. The design of strongly polynomial algorithms is based on the following observation: if a convex body $P \subset \mathbb{R}^d$ has 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 $d+1$ integral points $x_0, \ldots, x_d \in P$ that are not on the same hyperplane, then the convex body P contains the polytope $\text{conv}\{x_0, \ldots, x_d\}$, which has volume

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

which is a contradiction. Hence, we may use Vaidya's method or the random walk method to reduce the volume of search polytope P to $O[\exp(-d\log(d))]$ and then reduce the problem dimension by restricting to the hyperplane. After $d-1$ dimension reductions, we have an one-dimensional convex problem and algorithms in Section 3 can be applied. We give the pseudo-code in Algorithm 5.

Algorithm 5 Dimension reduction method for PGS guarantee

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

Output: An (ϵ, δ)-PGS 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** $d' = d, d-1, \ldots, 2$ **do**

4: Apply Vaidya's method or the Random Walk method with $(\epsilon/4, \delta/4)$-$\mathcal{SO}$ oracles.

5: Add points where separation oracles are called to $\mathcal{S}$.

6: Stop when the volume of P is small enough.

7: Find the hyperplane H that contains all integral points in P.

▷ If P contains no integral points, then an arbitrary hyperplane works.

8: Update P to a polytope on the hyperplane H. ▷ Reduce the dimension by 1.

9: **end for**

10: Find an $(\epsilon/4, \delta/4)$-PGS solution of the last one-dim problem and add the solution to $\mathcal{S}$.

11: Find the $(\epsilon/4, \delta/4)$-PGS solution $\hat{x}$ of problem $\min_{x\in\mathcal{S}}\ f(x)$.

12: Round $\hat{x}$ to an integral solution by Algorithm 3.

To make Algorithm 5 practical, we need to consider the following two problems:

- How large is the volume of projected polytope on the hyperplane?
- How many arithmetic operations are required to identify the hyperplane given the volume of P is small enough?

The answer to the first problem is crucial to the number of separation oracles needed in each iteration. Although the polytope P has small volume in the original space, it may have much larger volume after projected onto a hyperplane. Hence, the number of separation oracle evaluations in each iteration is decided by the upper bound of the volume. On the other hand, the number of arithmetic operations in each iteration is mainly determined by the answer to the second problem, since other parts of the algorithm requires polynomially many arithmetic operations. Recently, the two problems are solved in Jiang (2020). For the first problem, the author has designed an alternative potential function that includes both the volume of search polytope and the density of integral points in the current subspace. The increase of the potential function after each projection is better bounded than that of the volume and is decreased at a constant rate after adding a cutting plane. For the second problem, the author shows that the LLL algorithm Lenstra et al. (1982) can be applied to find LLL-reduced basis, which contains the normal vector of the hyperplane when the volume of search set is small enough. More specifically, Jiang (2020) proves that $O(d^2 + d\log(N))$ separation oracles and $\text{poly}(d, \log(N))$ arithmetic operations are enough for finding an optimal solution in the deterministic case. We show that the results can be naturally extended to the stochastic case to achieve the PGS guarantee.

THEOREM 9. *Suppose Assumptions 1-4 hold. Algorithm 5 returns an (ϵ, δ)-PGS solution and we have*

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

Proof of Theorem 9. The proof of Theorem 9 is given in EC.3.3. □

It is noted in Jiang (2020) that with exponentially many arithmetic operations, we can reduce the number of separation oracles to $O(d^2)$. Hence, with exponentially many arithmetic operations, the expected simulation cost can be reduced to $\tilde{O}[d^4N^2\epsilon^{-2}\log(1/\delta)]$. Similar to the stochastic Vaidya's method, the acceleration scheme can reduce the number of required simulation runs if we substitute Algorithm 4 with Algorithm 5. We give the reduced expected simulation cost for achieving the PCS-IZ guarantee and omit the proof.

THEOREM 10. *Suppose Assumptions 1-4 hold. The accelerated dimension reduction method returns an (c,δ)-PCS-IZ solution and we have*

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

4.4. Multi-dimensional Enhanced Adaptive Sampling Algorithm

In this subsection, we give the multi-dimensional version of the enhanced adaptive sampling algorithm designed in Section 3.2. Similar to the one-dimensional case, the asymptotic simulation cost of the multi-dimensional algorithm does not depend on the problem scale N and Lipschitz constant L. Hence, the multi-dimensional algorithm provides a matching simulation cost to the one-dimensional case up to a constant. However, the expected simulation cost is exponentially dependent on the dimension d. Therefore, the enhanced adaptive sampling algorithm is only suitable for low-dimensional problems.

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 6. Suppose a constant $C>0$ is given. We say 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 $|\hat{f}^*-f^*|\le\epsilon$ with probability at least $1-\delta$, using at most

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

simulation runs.

For example, Theorem 3 shows that the enhanced adaptive sampling algorithm (Algorithm 2) returns an $(\epsilon/2, \delta/2)$-PGS solution with $\tilde{O}[\epsilon^{-2}\log(1/\delta)]$ simulations. Then, we can simulate the function value at the solution for $\tilde{O}[\epsilon^{-2}\log(1/\delta)]$ times such that the $1-\delta/4$ confidence width is smaller than $\epsilon/4$. Then, the empirical mean of function values at the solution is at most ϵ far from f^* with probability at least $1-\delta$. Hence, we know that Algorithm 2 is sub-Gaussian with dimension 1. We note that, if we treat algorithms as estimators, estimators are generally "biased". This fact implies that the empirical mean of several (ϵ, δ)-PGS solutions does not give 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 multi-dimensional problems. We first define the marginal objective function as

$$f^{d-1}(x) := \min_{y\in[N]^{d-1}} f(y,x). \tag{10}$$

We can see 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

$$\min_{x\in[N]} f^{d-1}(x) = \min_{x\in[N]} \min_{y\in[N]^{d-1}} f(y,x) = \min_{x\in\mathcal{X}} f(x) \tag{11}$$

Moreover, we can prove that problem (11) is also a convex problem.

LEMMA 5. *If function $f(x)$ is $L^\natural$-convex, then function $f^{d-1}(x)$ is $L^\natural$-convex on $[N]$.*

Proof of Lemma 5. The proof of Lemma 5 is given in EC.3.4. □

Based on the observations above, we can use sub-Gaussian algorithms for $(d-1)$-dimensional problems and Algorithm 2 to construct sub-Gaussian algorithms for d-dimensional problems. We give the pseudo-code in Algorithm 6.

Algorithm 6 Multi-dimensional enhanced adaptive sampling algorithm

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

Output: An (ϵ, δ)-PGS solution x^* to problem (1).

Set the active set $\mathcal{S} \leftarrow [N]$.

Set the step size $d \leftarrow 1$ and the maximal number of comparisons $T_{max} \leftarrow N$.

Set $N_{cur} \leftarrow +\infty$.

while the size of $\mathcal{S}$ is at least 3 **do** ▷ Iterate until $\mathcal{S}$ has at most 2 points.

  if $|\mathcal{S}| \leq N_{cur}/2$ **then** ▷ Update the confidence interval.

    Record current active set size $N_{cur} \leftarrow |\mathcal{S}|$.

    Set the confidence width $h \leftarrow N_{cur} \cdot \epsilon/160$.

    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 $1 - \delta/(2T_{max})$.

  end if

  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**

    if $x < y$ **then**

      Remove all points $z \in \mathcal{S}$ such that $z \geq y$ from $\mathcal{S}$.

    else

      Remove all points $z \in \mathcal{S}$ such that $z \leq y$ from $\mathcal{S}$.

    end if

  else ▷ **Type-II Operation**

    Update the step size $d \leftarrow 2d$.

    Update $\mathcal{S} \leftarrow \{x_{min}, x_{min} + d, \ldots, x_{min} + kd\}$, where $x_{min} = \min_{x \in \mathcal{S}} x$ and $k = \lceil |\mathcal{S}|/2 \rceil - 1$.

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

We prove that Algorithm 6 is sub-Gaussian with dimension d and estimate its parameter.

THEOREM 11. *Suppose Assumptions 1-4 hold. Also, suppose Algorithm $\mathcal{A}$ is sub-Gaussian with dimension $d-1$ and parameter C. Then, Algorithm 6 is a sub-Gaussian algorithm with dimension d and parameter MC, where $M > 0$ is an absolute constant.*

Proof of Theorem 11. The proof of Theorem 11 is given in EC.3.5. □

Using the results of above theorem and the fact that Algorithm 2 is sub-Gaussian with dimension 1, we can inductively construct sub-Gaussian algorithms with any dimension d.

THEOREM 12. *Suppose Assumptions 1-4 hold. There exists an $[(\epsilon,\delta)$-$PGS, \mathcal{MC}]$-algorithm that is sub-Gaussian with parameter $M^{d-1}C$, where M is the constant in Theorem 11 and C is the parameter of Algorithm 2. Hence, we have*

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

Furthermore, by choosing $\epsilon = c/2$, we get

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

If we treat $F(x,\xi_x)$ as a sub-Gaussian algorithm with dimension 0 and parameter σ^2, then Theorem 11 implies that there exists a sub-Gaussian algorithm with dimension 1 and parameter $\sigma^2 M$. However, the parameter C of Algorithm 2 is usually smaller than $\sigma^2 M$ and therefore Algorithm 2 is preferred in the one-dimensional case.

5. Numerical Experiments

In this section, we implement our proposed simulation algorithms that are guaranteed to find high-confidence high-precision PGS solutions. First, we consider the problem of finding the optimal allocation of a total number of N staffs to two queues so that the average waiting time for all the arrivals from the two queues is minimized. Given the optimality parameters ϵ, δ, we show that the adaptive sampling algorithm and the enhanced adaptive sampling algorithm have respectively $O(\log N)$ and $O(1)$ dependence on the scale N, which supports our theory results. Second, we construct a high-dimensional stochastic function, whose expectation is a separable convex function, i.e., functions with the form $f(x) = \sum_{i=1}^{d} f^i(x_i)$ for convex functions $f^1(x), \ldots, f^d(x)$, to test and compare gradient-based methods with the stochastic cutting-plane methods proposed in this work for different scale N and dimension d.

5.1. Staffing Two Queues under Resource Constraints

Consider a service system that operates over a time horizon $[0, T]$ with two streams of customers arriving at the system. One example is that the system receives service requests from both online app-based customers and offline walk-in customers, and each stream needs dedicated servers assigned. The first stream of customers arrive according to a doubly stochastic non-homogeneous Poisson process $N_1 = (N_1(t) : t \in [0, T])$, and each customer requests a service with service time independent and identically distributed according to a distribution S_1. The second stream of customers obeys the same model with process $N_2 = (N_2(t) : t \in [0, T])$ and distribution S_2. The two streams of customers form two separate queues and their arrival processes can be correlated. Suppose that the decision maker needs to staff the two queues separately. There are in total a number $N + 1$ of homogeneous servers that work independently in parallel. Each server can handle the service requested by customers from either stream, one at a time. Suppose that no change on the staffing plan can be made once the system starts working. 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 function is select the staffing level $x \in [N]$ for the first queue and the staffing level $N+1-x$ for the second queue, in order to minimize the expected average waiting time for all customers from the two streams over $[0,T]$. In the numerical example, we consider $N \in \{10,20,\ldots,150\}$, $T=2$. The arrival processes N_1 and N_2 are non-homogeneous processes with random intensity functions $\Gamma_1 \cdot \lambda_1(t)$ and $\Gamma_2 \cdot \lambda_2(t)$, in which

$$\lambda_1(t) := 75 + 25\sin(0.3t), \quad \lambda_2(t) := 80 + 40\sin(0.2t).$$

Positive-valued random variables Γ_1, Γ_2 are defined as

$$\Gamma_1 := X + Z, \quad \Gamma_2 := Y - Z,$$

where X, Y are independent uniform random variables on $[0.75, 1.25]$ and Z is an independent uniform random variable on $[-0.5, 0.5]$. The service time distribution S_1 is log-normal distributed with mean 0.75 and variance 0.1. The service time distribution S_2 is gamma distributed with mean 0.65 and variance 0.1. Figure 1 plots an empirical average waiting time as a function in the discrete decision variable x. Shown by the plot, the landscape around the optimum is extremely flat and such property may cause challenges for algorithms that desire the selection of the exact best solution (i.e., PCS criterion). Instead, algorithms that are designed for the (ϵ, δ)-PGS criterion do not suffer from the extremely flat landscape around the global optimum.

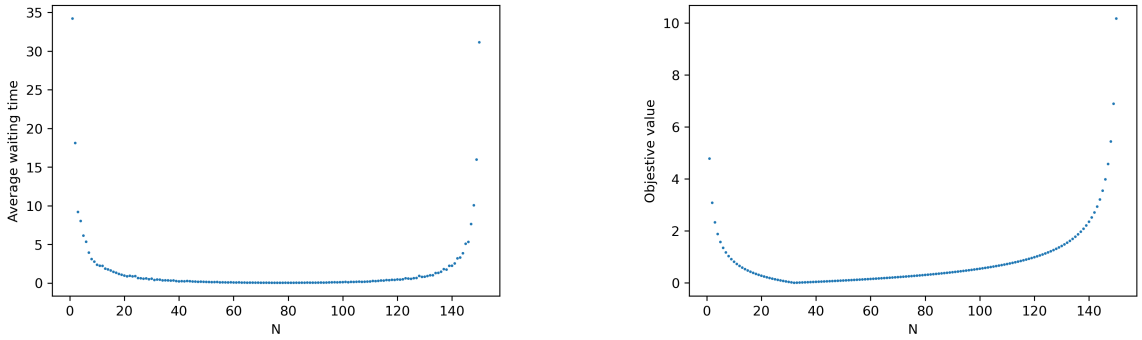
5.2. Separable Convex Function Minimization

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

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

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

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



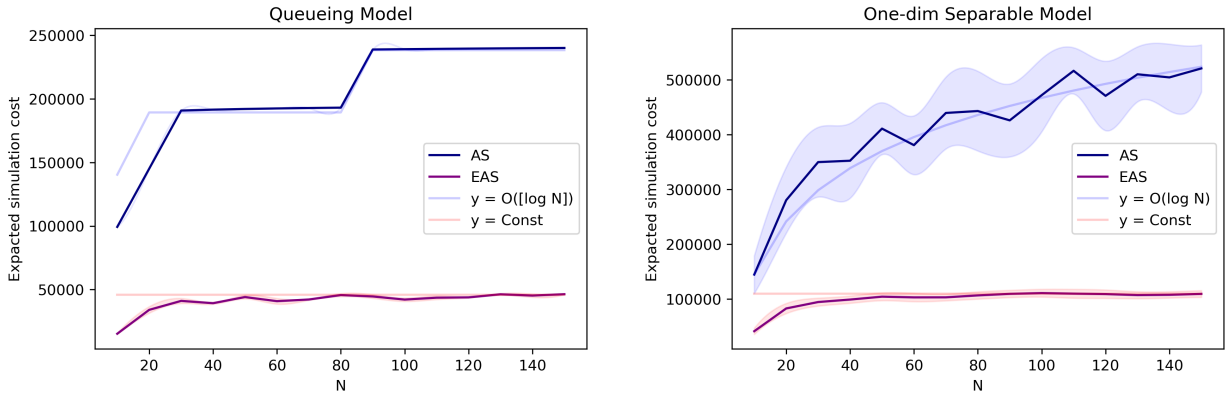
(a) (b)

Figure 1 The landscapes of objective functions in the one-dimensional case. **(a)** The empirical average waiting time with $N = 150$. **(b)** The landscape of the separable convex function with $d = 1, N = 150$ and optimum $x^* = 31$.

We can see that the function $f_{c,x^*}(x)$ is the sum of separable convex function and therefore is $L^\natural$-convex. Also, function $f_{c,x^*}(x)$ has optimum x^* and optimal value 0. For stochastic evaluations, we add the Gaussian noise with mean 0 variance 1. We can see from Figure 1 that the landscape of $g(x^*; x)$ is similar to the average waiting time of the queueing model. One purpose of designing this numerical example is that the expected objective function has closed form, and we are able to exactly compute the optimality gap of the decision variable returned by the proposed algorithms.

5.3. Numerical Results: Adaptive Sampling Algorithm and Enhanced Adaptive Sampling Algorithm

We first compare the performance of the adaptive sampling (AS) algorithm (Section 3.1) and the enhanced adaptive sampling (EAS) algorithm (Section 3.2) on both problems. We consider problems with $d = 1$ and scale $N \in \{10, 20, \ldots, 150\}$. The expected simulation cost is computed by averaging 400 independent solving processes. For the queueing optimal allocation problem, we set the optimality parameters for the PGS criterion as $\epsilon = 1, \delta = 10^{-6}$. An upper bound on the variance is given as $\sigma^2 = 10$. For the separable convex function minimization, we generate each c_i from the uniform distribution on $[0.75, 1.25]$ and x_i^* from the discrete uniform distribution on $\{1, 2, \ldots, \lfloor 0.3N \rfloor\}$. The optimality parameters are chosen as $\epsilon = 0.2, \delta = 10^{-6}$ and the variance $\sigma^2 = 1$.



(a) (b)

Figure 2 The expected simulation cost of adaptive sampling (AS) and enhanced adaptive sampling (EAS) algorithms in the one-dimensional case. **(a)** The optimal allocation problem. **(b)** The one-dimensional separable convex function minimization.

The algorithms are run to satisfy the given PGS criterion. We then plot the expected simulation costs in Figure 2. For the queueing optimal allocation problem, the expected simulation cost of AS and EAS algorithms have $O(\lceil \log N \rceil)$ and $O(1)$ dependence on the scale N, respectively. The expected simulation cost of the enhanced adaptive sampling algorithm is almost independent of N and this verifies our theoretical analysis. For the separable convex function problem, same as the queueing example, the expected simulation cost of the AS algorithm and EAS algorithm have $O(\log N)$ and $O(1)$ dependence on N, respectively. This again verifies our theoretical analysis.

5.4. Numerical Results: Gradient-based and Stochastic Cutting-plane Methods

We next compare the performance of gradient-based and stochastic cutting-plane algorithms (Section 4) on multi-dimensional problems. To facilitate comparison, we consider the separable convex function minimization problem, where the optimal solution and optimal objective value are exact. Algorithm-wise, we consider the truncated subgradient descent method in Zhang and Zheng (2020), Vaidya's method, the random walk method and the dimension reduction method. 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 $\epsilon = (d!)^{1/d}/5, \delta = 10^{-6}$, respectively. The choice of ϵ ensures that the ϵ-sub-level set of objective function covers a "flat region" of the landscape. Same

Table 2 Simulation cost and coverage rate of different algorithms on separable convex functions.

		Gradient Methods		**Cutting-plane Methods**					
Params.		Subgradient		Vaidya's Method		Random Walk		Dim Reduction	
d	N	Cost(10^7)	Rate(%)	Cost(10^7)	Rate(%)	Cost(10^7)	Rate(%)	Cost(10^7)	Rate(%)
2	50	0.39	100.0	2.03	100.0	0.57	100.0	0.19	100.0
2	100	0.44	100.0	2.05	100.0	0.59	100.0	0.23	100.0
2	150	0.51	100.0	2.05	100.0	0.58	100.0	0.26	100.0
6	50	0.96	100.0	4.53	100.0	1.59	100.0	1.71	100.0
6	100	1.07	100.0	4.61	100.0	1.68	100.0	1.94	100.0
6	150	1.25	100.0	4.64	100.0	1.72	100.0	2.04	100.0
10	50	1.27	100.0	5.95	100.0	2.14	100.0	3.20	100.0
10	100	1.42	100.0	6.14	100.0	2.34	100.0	3.53	100.0
10	150	1.65	100.0	6.21	100.0	2.44	100.0	3.65	100.0
15	50	1.52	100.0	8.56	100.0	3.71	100.0	4.14	100.0
20	50	1.69	100.0	10.3	100.0	4.14	96.25	4.28	99.00
50	50	2.25	100.0	21.6	100.0	TLE	-	TLE	-

as the one-dimensional case, we compute the average simulation cost of 400 independently generated models to approximate the expected simulation cost. Moreover, early stopping conditions are designed to terminate algorithms early when little progress is achieved for every 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 Vaidya's method and the random walk method, we terminate the algorithm if the empirical mean of the last 3 iterations does not decrease by $2\epsilon/(d\sqrt{N})$. For the dimension reduction method, we terminate the algorithm if the polytope is empty. Also, we have observed that using $(N\epsilon/4, \delta/4)$-$\mathcal{SO}$ oracles is enough for producing high-probability guarantee within the range $N \leq 150$.

We summarize the results in Table 2, where the coverage rate refers to the percentage of implementations that produce an ϵ-optimal solution and "TLE" (Time Limit Exceed) means a single implementation does not finish in 30 minutes. Since the coverage rates of all algorithms are close to 100%, we claim that the PGS guarantee is satisfied by all algorithms. When the dimension d is not large, the dimension reduction method has the best performance and has the advantage of not requiring the knowledge about Lipschitz constant; when the dimension d is large, the higher order dependence on d makes stochastic cutting-plane methods inferior to the stochastic subgradient method. Moreover, Vaidya's method requires more simulation runs compared to the random walk method, which is consistent with our remark after Theorem 7. However, the $O(d^7)$ arithmetic operations of the random walk method prohibit its application to higher-dimension problems. In summary, based on the theory and the supportive results from the numerical experiments, we recommend the use of adaptive sampling algorithms and stochastic cutting-plane algorithms when (i) N is large while d is moderate, or (ii) the Lipschitz constant is not known or hard to estimate.

6. Conclusion

New stochastic localization algorithms are proposed for large-scale discrete convex simulation optimization problems. All simulation algorithms are theoretically guaranteed to identify a solution that has close objective values with the optimal up the the given precision with high probability. Also, the efficiency of proposed algorithms is characterized by the upper bound of the expected simulation cost. Specifically, in the one-dimensional case, we propose an enhanced adaptive sampling method, which has an expected simulation cost asymptotically independent of the problem scale and attains the best achievable performance. For the multi-dimensional case, stochastic cutting-plane methods are designed by extending deterministic cutting-plane methods. The expected simulation cost is proved to have only logarithmic or no dependence on the Lipschitz constant. In addition, numerical results on both artificial and real-world models verify our theoretical findings and provide insights on which algorithm to use based on real-application settings.

References

Agarwal A, Foster DP, Hsu DJ, Kakade SM, Rakhlin A (2011) Stochastic convex optimization with bandit feedback. Advances in Neural Information Processing Systems, 1035–1043.

Alman J, Williams VV (2020) A refined laser method and faster matrix multiplication. arXiv preprint arXiv:2010.05846 .

Altman E, Gaujal B, Hordijk A (2003) Discrete-event control of stochastic networks: Multimodularity and regularity (springer).

Ankenman B, Nelson BL, Staum J (2010) Stochastic kriging for simulation metamodeling. Operations Research 58(2):371–382.

Barton RR (2015) Tutorial: simulation metamodeling. 2015 Winter Simulation Conference (WSC), 1765–1779 (IEEE).

Bechhofer RE (1954) A single-sample multiple decision procedure for ranking means of normal populations with known variances. The Annals of Mathematical Statistics 16–39.

Bertsimas D, Vempala S (2004) Solving convex programs by random walks. Journal of the ACM (JACM) 51(4):540–556.

Dyer M, Proll L (1977) Note—on the validity of marginal analysis for allocating servers in m/m/c queues. Management Science 23(9):1019–1022.

Eckman DJ, Henderson SG (2018) Fixed-confidence, fixed-tolerance guarantees for selection-of-the-best procedures. Technical report, Working paper, Cornell University, School of Operations Research and ....

Eckman DJ, Plumlee M, Nelson BL (2020) Plausible screening using functional properties for simulations with large solution spaces, working paper.

Even-Dar E, Mannor S, Mansour Y (2002) Pac bounds for multi-armed bandit and markov decision processes. International Conference on Computational Learning Theory, 255–270 (Springer).

Freund D, Henderson SG, Shmoys DB (2017) Minimizing multimodular functions and allocating capacity in bike-sharing systems. International Conference on Integer Programming and Combinatorial Optimization, 186–198 (Springer).

Fujishige S (2005) Submodular functions and optimization (Elsevier).

Hong LJ, Fan W, Luo J (2020) Review on ranking and selection: A new perspective. arXiv preprint arXiv:2008.00249 .

Hong LJ, Nelson BL, Xu J (2015) Discrete optimization via simulation. Handbook of simulation optimization, 9–44 (Springer).

Jian N, Freund D, Wiberg HM, Henderson SG (2016) Simulation optimization for a large-scale bike-sharing system. 2016 Winter Simulation Conference (WSC), 602–613 (IEEE).

Jiang H (2020) Minimizing convex functions with integral minimizers. arXiv preprint arXiv:2007.01445 .

Jiang H, Lee YT, Song Z, Wong SCw (2020) An improved cutting plane method for convex optimization, convex-concave games, and its applications. Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, 944–953.

Kaufmann E, Cappé O, Garivier A (2016) On the complexity of best-arm identification in multi-armed bandit models. The Journal of Machine Learning Research 17(1):1–42.

Kleijnen JP (2017) Regression and kriging metamodels with their experimental designs in simulation: a review. European Journal of Operational Research 256(1):1–16.

Lee YT, Sidford A, Wong SCw (2015) A faster cutting plane method and its implications for combinatorial and convex optimization. 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, 1049–1065 (IEEE).

Lenstra AK, Lenstra HW, Lovász L (1982) Factoring polynomials with rational coefficients. Mathematische annalen 261(ARTICLE):515–534.

Liang T, Narayanan H, Rakhlin A (2014) On zeroth-order stochastic convex optimization via random walks. arXiv preprint arXiv:1402.2667 .

Lovász L (1983) Submodular functions and convexity. Mathematical programming the state of the art, 235–257 (Springer).

Luo J, Hong LJ, Nelson BL, Wu Y (2015) Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. Operations Research 63(5):1177–1194.

Ma S, Henderson SG (2017) An efficient fully sequential selection procedure guaranteeing probably approximately correct selection. 2017 Winter Simulation Conference (WSC), 2225–2236 (IEEE).

Ma S, Henderson SG (2019) Predicting the simulation budget in ranking and selection procedures. ACM Transactions on Modeling and Computer Simulation 29(3):Article 14, 1–25.

Murota K (2003) Discrete convex analysis. Society for Industrial and Applied Mathematics (Citeseer).

Nelson BL (2010) Optimization via simulation over discrete decision variables. Risk and Optimization in an Uncertain World, 193–207 (Informs).

Ni EC, Ciocan DF, Henderson SG, Hunter SR (2017) Efficient ranking and selection in high performance computing environments. Operations Research 65(3):821–836.

Salemi PL, Song E, Nelson BL, Staum J (2019) Gaussian markov random fields for discrete optimization via simulation: Framework and algorithms. Operations Research 67(1):250–266.

Shaked M, Shanthikumar JG (1988) Stochastic convexity and its applications. Advances in Applied Probability 20(2):427–446.

Singhvi D, Singhvi S, Frazier PI, Henderson SG, O'Mahony E, Shmoys DB, Woodard DB (2015) Predicting bike usage for new york city's bike sharing system. AAAI Workshop: Computational Sustainability (Citeseer).

Sun L, Hong LJ, Hu Z (2014) Balancing exploitation and exploration in discrete optimization via simulation through a gaussian process-based search. Operations Research 62(6):1416–1438.

Vaidya PM (1996) A new algorithm for minimizing convex functions over convex sets. Mathematical programming 73(3):291–341.

Wolff RW, Wang CL (2002) On the convexity of loss probabilities. Journal of applied probability 402–406.

Zhang H, Zheng Z (2020) Discrete convex simulation optimization. arXiv preprint arXiv:2010.16250 .

Zhong Y, Hong LJ (2019) Knockout-tournament procedures for large-scale ranking and selection in parallel computing environments. accepted .

Proofs of Statements

EC.1. Omitted PCS-IZ Algorithms

EC.1.1. Modified Adaptive Sampling Algorithm

We first give the modified adaptive sampling algorithm and omit lines that are the same as Algorithm 1.

Algorithm 7 Adaptive sampling algorithm for PCS-IZ guarantee

Input: Model $\mathcal{X}=[N], \mathcal{B}_x, F(x,\xi_x)$, optimality guarantee parameter δ, indifference zone parameter c.

Output: An (c,δ)-PCS-IZ solution x^* to problem (1).

1: Set upper and lower bound of current interval $L \leftarrow 1, U \leftarrow N$.

2: Set maximal number of comparisons $T_{max} \leftarrow \log_{1.5}(N)+2$.

3: **while** $U-L>2$ **do** ▷ Iterate until there are at most 3 systems.

...

8: Stop sampling if one of the following conditions holds:

$$(i)\quad \hat{F}_n(N_{1/3})-h_{1/3} \geq \hat{F}_n(N_{2/3})+h_{2/3},$$

$$(ii)\quad \hat{F}_n(N_{1/3})+h_{1/3} \leq \hat{F}_n(N_{2/3})-h_{2/3},$$

$$(iii)\quad h_{1/3} \leq (N_{2/3}-N_{1/3})\cdot c/5 \text{ and } h_{2/3} \leq (N_{2/3}-N_{1/3})\cdot c/5.$$

...

13: **if** $h_{1/3} \leq (N_{2/3}-N_{1/3})\cdot c/5$ and $h_{2/3} \leq (N_{2/3}-N_{1/3})\cdot c/5$ **then**

14: Update $L \leftarrow N_{1/3}$ and $U \leftarrow N_{2/3}$.

15: **end if**

16: **end while**

17: Simulate $F(x,\xi_x)$ for all $x \in \{L,\ldots,U\}$ until the $1-\delta/(2T_{max})$ confidence widths are smaller than $c/3$.

▷ Now $U - L \leq 2$.

18: Return the point in $\{L, \ldots, U\}$ with minimal empirical mean.

EC.1.2. Stochastic Vaidya's Method for The PCS-IZ Guarantee

Then, we give the stochastic Vaidya's method for PCS-IZ guarantee. We note that by substituting Algorithm 4 with Algorithm 5, we get an accelerated strongly polynomial algorithm for the PCS-IZ guarantee.

Algorithm 8 Stochastic Vaidya's method for PCS-IZ guarantee

Input: Model $\mathcal{X}, \mathcal{B}_x, F(x, \xi_x)$, optimality guarantee parameter δ, indifference zone parameter c, Lipschitz constant L, (ϵ, δ)-$\mathcal{SO}$ oracle $\hat{g}$.

Output: An (c, δ)-PCS-IZ 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 searching space $\mathcal{Y}_0 \leftarrow [1, N]^d$.

4: **for** $e = 0, \ldots, E-1$ **do**

5: Use Algorithm 4 to get an $(\epsilon_e, \delta/(2E))$-PGS solution x_e in $\mathcal{Y}_e$.

6: Update guarantee $\epsilon_{e+1} \leftarrow \epsilon_e/2$.

7: Update the searching 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 by Algorithm 3.

EC.2. Proofs in section 3

EC.2.1. Proof of Theorem 1

We first estimate the simulation cost of Algorithm 1. The following lemma gives an upper bound on the total number of iterations.

LEMMA EC.1. *Suppose Assumptions 1-4 hold. The number of iterations of Algorithm 1 is at most* $\log_{1.5}(N) + 1$.

Proof. If the total number of points N is at most 3, then there is no iteration. In the following proof, we assume $N \geq 4$. We first calculate the shrinkage of interval length after each iteration. We denote the upper and the lower bound at the beginning of the k-th iteration as U_k and L_k, respectively. Then, we know there are $n_k := U_k - L_k + 1$ points in the k-th iteration and the algorithm starts with $L_1 = 1, U_1 = N$. We define the 3-quantiles $N_{1/3} := \lfloor 2L_k/3 + U_k/3 \rfloor$ and $N_{2/3} := \lceil L_k/3 + 2U_k/3 \rceil$. By the updating rule, the next interval is

$$[L_k, N_{2/3}] \quad \text{or} \quad [N_{1/3}, U_k] \quad \text{or} \quad [N_{1/3}, N_{2/3}].$$

By discussing three cases when $n_k \in 3\mathbb{Z}$, $n_k \in 3\mathbb{Z}+1$ and $n_k \in 3\mathbb{Z}+2$, we know the next interval has at most $2n_k/3 + 1$ points, i.e.,

$$n_{k+1} \leq 2n_k/3 + 1. \tag{EC.1}$$

Rewriting the inequality, we get the relation $n_{k+1} - 3 \leq 2(n_k - 3)/3$. Combining with the fact that $n_1 = N$, it follows that

$$n_k \leq \left(\frac{2}{3}\right)^{k-1} (N-3) + 3.$$

Suppose Algorithm 1 terminates after T iterations. Then, it holds that $n_T \geq 4$ and $n_{T+1} \leq 3$. Hence, we know

$$4 \leq n_T \leq \left(\frac{2}{3}\right)^{T-1} (N-3) + 3,$$

which implies

$$T \leq \log_{1.5}(N-3) + 1 \leq \log_{1.5}(N) + 1.$$

□

In the next lemma, we estimate the simulation cost of each iteration.

LEMMA EC.2. *Suppose Assumptions 1-4 hold. The simulation cost of each iteration of Algorithm 1 is at most* $256\sigma^2\epsilon^{-2}\log(4T_{max}/\delta)$*, where* $T_{max} := \log_{1.5}(N) + 2$*. The simulation cost of the sub-problem is at most* $24\sigma^2\epsilon^{-2}\log(4T_{max}/\delta)$*.*

Proof. We first calculate the confidence interval at each point $x \in \mathcal{X}$. By Assumption 3, the distribution of $F(x, \xi_x)$ is sub-Gaussian with parameter σ^2. Hence, the distribution of the empirical mean $\hat{F}_n(x)$ is sub-Gaussian with parameter σ^2/n and the Hoeffding bound gives

$$\mathbb{P}\left[|\hat{F}_n(x) - f(x)| \geq t\right] \leq 2\exp\left(-\frac{nt^2}{2\sigma^2}\right) \quad \forall t \geq 0.$$

Next, we estimate the simulation cost of each iteration. Choosing

$$t_\epsilon = \frac{\epsilon}{8}, \quad n_{\epsilon,\delta} = \frac{128\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right),$$

we get the deviation bound

$$\mathbb{P}\left[|\hat{F}_{n_{\epsilon,\delta}}(x) - f(x)| \geq \frac{\epsilon}{8}\right] \leq \frac{\delta}{2T_{max}}.$$

If we simulate $F(N_{1/3}, \xi_{1/3})$ and $F(N_{2/3}, \xi_{2/3})$ for $n_{\epsilon,\delta}$ times, the confidence width is $\epsilon/4$ and the third condition in Line 8 is satisfied. Hence, the simulation cost of each iteration is at most $2n_{\epsilon,\delta}$. Finally, for the sub-problem, we choose

$$t_\epsilon = \frac{\epsilon}{2}, \quad \tilde{n}_{\epsilon,\delta} = \frac{8\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)$$

and it follows that

$$\mathbb{P}\left[|\hat{F}_{\tilde{n}_{\epsilon,\delta}}(x) - f(x)| \geq \frac{\epsilon}{2}\right] \leq \frac{\delta}{2T_{max}}.$$

Hence, simulating $\tilde{n}_{\epsilon,\delta}$ times on each point are sufficient and the simulation cost is at most $3\tilde{n}_{\epsilon,\delta}$.

□

Combining Lemmas EC.1 and EC.2, we get the total simulation cost of Algorithm 1.

LEMMA EC.3. *Suppose Assumptions 1-4 hold. The expected simulation cost of Algorithm 1 is at most*

$$\frac{256T_{max}\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right) = O\left[\frac{\log(N)}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right],$$

where $T_{max} := \log_{1.5}(N) + 2$.

Proof. By Lemmas EC.1 and EC.2, the total simulation cost of the first part is at most

$$[\log_{1.5}(N)+1]\cdot\frac{256\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)\leq[T_{max}-1]\cdot\frac{256\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)$$

and the simulation cost of the second part is at most

$$\frac{24\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right).$$

Combining two parts, we know the total simulation cost is at most

$$[T_{max}-1]\cdot\frac{256\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)+\frac{24\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)\leq\frac{256T_{max}\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right).$$

□

Finally, we verify the correctness of Algorithm 1 and get an upper bound on $T_0(\epsilon,\delta\mathcal{MC})$.

Proof of Theorem 1. We denote $T_{max}:=\log_{1.5}(N)+2$. We also denote the upper and the lower bound at the beginning of the k-th iteration as U_k and L_k, respectively. We use the induction method to prove that, for the k-th iteration, at least one of the following two events happens with probability at least $1-(k-1)\cdot\delta/T_{max}$:

- **Event-I.** A solution to problem (1) is in $\{L_k,\ldots,U_k\}$,
- **Event-II.** For any $x\in\{L_k,\ldots,U_k\}$, it holds $f(x)\leq\min_{y\in\mathcal{X}}f(y)+\epsilon$.

When $k=1$, all solutions to problem (1) are in $\mathcal{X}=\{L_1,\ldots,U_1\}$ and Event-I happens with probability 1. Suppose the claim is true for the first $k-1$ iterations. We consider the k-th iteration. For the $(k-1)$-th iteration, if Event-II happens with probability at least $1-(k-2)\cdot\delta/T_{max}$, then Event-II happens for the k-th iteration with the same probability. This is because the interval $\{L_k,\ldots,U_k\}$ is a subset of $\{L_{k-1},\ldots,U_{k-1}\}$ and all points in the new interval satisfy the condition of Event-II.

Hence, we only need to consider the case when only Event-I for the $(k-1)$-th iteration happens with probability at least $1-(k-2)\delta/T_{max}$. We assume Event-I happens and consider conditional probabilities in the following of the proof. We denote

$$N_{1/3}:=\lfloor 2L_{k-1}/3+U_{k-1}/3\rfloor,\quad N_{2/3}:=\lfloor L_{k-1}/3+2U_{k-1}/3\rfloor$$

and discuss by two different cases.

Case I. Suppose one of the first two conditions in line 8 holds when the sampling process terminates. Since the two conditions are symmetrical, we assume without loss of generality that the first condition holds. Then, the new interval is $[N_{1/3}, U_{k-1}]$ and, by the definition of confidence interval, we know

$$f(N_{1/3}) \geq f(N_{2/3})$$

holds with probability at least $1-\delta/T_{max}$. We assume the above event and Event-I for the $(k-1)$-th iteration both happen, which has joint probability at least $1-\delta/T_{max}-(k-2)\delta/T_{max} = 1-(k-1)\delta/T_{max}$. By the convexity of $f(x)$, it holds that

$$f(x) \geq f(N_{1/3}) + \frac{x-N_{1/3}}{N_{1/3}-N_{2/3}}\left[f(N_{1/3})-f(N_{2/3})\right] \geq f(N_{1/3}) \quad \forall x \in \{L_{k-1},\ldots,N_{1/3}\}.$$

Hence, the minimum of $f(x)$ in $\{L_{k-1},\ldots,U_{k-1}\}$ is attained by a point in $\{N_{1/3},\ldots,U_{k-1}\}$. Combining with the assumption that there exists a solution to problem (1) in $\{U_{k-1},\ldots,L_{k-1}\}$, we know that there exists a solution to problem (1) in $\{N_{1/3},\ldots,U_{k-1}\}$. Thus, Event-I for the k-th iteration happens with probability at least $1-(k-1)\delta/T_{max}$.

Case II. Suppose only the last condition in line 8 holds when the sampling process terminates. Since the first two conditions in line 8 do not hold, we have

$$\left|\hat{F}_n(N_{1/3}) - \hat{F}_n(N_{2/3})\right| \leq \epsilon/4. \tag{EC.2}$$

In addition, by the definition of confidence interval, it holds

$$\left|f(N_{1/3}) - \hat{F}_n(N_{1/3})\right| \leq \epsilon/8, \quad \left|f(N_{2/3}) - \hat{F}_n(N_{2/3})\right| \leq \epsilon/8$$

with probability at least $1-\delta/T_{max}$. Combining with inequality (EC.2), we know

$$\left|f(N_{1/3}) - f(N_{2/3})\right| \leq \epsilon/2 \tag{EC.3}$$

holds with probability at least $1-\delta/T_{max}$. We assume that the above event and Event-I for the $(k-1)$-th iteration both happen, which has joint probability at least $1-\delta/T_{max}-(k-2)\delta/T_{max} = 1-(k-1)\delta/T_{max}$. We prove that if Event-I for the k-th iteration does not happen, then Event-II for

the k-th iteration happens. Under the condition that Event-I does not happen, we assume without loss of generality that solutions to problem (1) are in $\{L_{k-1}, \ldots, N_{1/3} - 1\}$. Using the convexity of function $f(x)$, we know

$$f(x) \geq f(N_{1/3}) - \frac{N_{1/3} - x}{N_{2/3} - N_{1/3}} \left[f(N_{1/3}) - f(N_{2/3}) \right] \quad \forall x \in \{L_{k-1}, \ldots, N_{1/3}\}.$$

Choosing

$$x \in \left(\arg\min_{y \in \mathcal{X}} \ f(y) \right) \cap \{L_{k-1}, \ldots, U_{k-1}\} \neq \emptyset,$$

we get

$$\begin{aligned} \min_{y \in \mathcal{X}} \ f(y) &\geq f(X_{1/3}) - \frac{N_{1/3} - x}{N_{2/3} - N_{1/3}} \left[f(N_{1/3}) - f(N_{2/3}) \right] \\ &\geq f(X_{1/3}) - \frac{N_{1/3} - L_{k-1}}{N_{2/3} - N_{1/3}} \cdot \epsilon/2 \geq f(X_{1/3}) - \epsilon/2, \end{aligned}$$

where the last inequality is from the definition of 3-quantiles. Combining with inequality (EC.3), we get

$$\min_{y \in \mathcal{X}} \ f(y) \geq f(N_{2/3}) - \epsilon.$$

By the convexity of $f(x)$, it holds that

$$\max_{x \in \{N_{1/3}, \ldots, N_{2/3}\}} \ f(x) = \max\{f(N_{1/3}), f(N_{2/3})\} \leq \min_{y \in \mathcal{X}} \ f(y) + \epsilon,$$

which means Event-II for the k-th iteration happens.

Combining the two cases, we know the claim holds for the k-th iteration. Suppose there are T iterations in Algorithm 1. By Lemma EC.1, we have $T \leq T_{max} - 1$. By the induction method, the last interval $\{L_{T+1}, \ldots, U_{T+1}\}$ satisfies the condition in Event-I or Event-II with probability at least $1 - T \cdot \delta / T_{max} \geq 1 - \delta + \delta / T_{max}$. If Event-II happens with probability at least $1 - \delta + \delta / T_{max}$, then regardless of the point chosen in the sub-problem, the solution returned by the algorithm has value at most ϵ larger than the optimal value with probability at least $1 - \delta + \delta / T_{max} \geq 1 - \delta$. Hence, the solution satisfies the (ϵ, δ)-PGS guarantee. Otherwise, we assume Event-I happens with probability at least $1 - \delta + \delta / T_{max}$. Then, a solution to problem (1) is in $\{L_{T+1}, \ldots, U_{T+1}\}$. We choose

$$x^* \in \left(\arg\min_{x \in \mathcal{X}} \ f(x) \right) \cap \{L_{T+1}, \ldots, U_{T+1}\}$$

and suppose the algorithm returns

$$x^{**} \in \underset{x\in\{L_{T+1},\dots,U_{T+1}\}}{\arg\min} \hat{F}_n(x).$$

By the definition of confidence interval, it holds

$$f(x^{**}) \le \hat{F}_n(x^{**}) + \epsilon/2, \quad f(x^*) \ge \hat{F}_n(x^*) - \epsilon/2$$

with probability at least $1-\delta/T_{max}$. Under the above event, we get

$$f(x^{**}) \le \hat{F}_n(x^{**}) + \epsilon/2 \le \hat{F}_n(x^*) + \epsilon/2 \le f(x^*) + \epsilon.$$

Recalling that Event-I happens with probability at least $1-\delta+\delta/T_{max}$, the point x^{**} satisfies the above relation with probability at least $1-\delta$ and therefore satisfies the (ϵ,δ)-PGS guarantee. Combining with the first case, we know Algorithm 1 is an $[(\epsilon,\delta)\text{-PGS}, \mathcal{MC}]$-algorithm. □

EC.2.2. Proof of Theorem 2

We first estimate the simulation cost of each iteration and the sub-problem.

LEMMA EC.4. *Suppose Assumptions 1-4 hold. The simulation cost for each iteration of Algorithm 7 is at most* $100\sigma^2 c^{-2}(N_{2/3}-N_{1/3})^{-2}\log[4T_{max}/\delta]$, *where* $T_{max} := \log_{1.5}(N)+2$. *The simulation cost of the sub-problem is at most* $54\sigma^2 c^{-2}\log[4T_{max}/\delta]$.

Proof. The proof is similar to the proof of Lemma EC.2 and we only give a sketch of the proof. By the Hoeffding bound, simulating

$$\frac{50\sigma^2}{(N_{2/3}-N_{1/3})^2 c^2}\log\left(\frac{4T_{max}}{\delta}\right)$$

times on quantiles $N_{1/3}$ and $N_{2/3}$ is enough to ensure that the confidence width is at most $(N_{2/3}-N_{1/3})\cdot c/5$. It implies that the last condition in line 8 is satisfied and the simulation cost of each iteration is at most $100\sigma^2 c^{-2}(N_{2/3}-N_{1/3})^{-2}\log[4T_{max}/\delta]$. For the sub-problem, Hoeffding bound gives that simulating

$$\frac{18\sigma^2}{c^2}\log\left(\frac{4T_{max}}{\delta}\right)$$

times for each point are enough to ensure that the confidence width is at most $c/3$. Since there are at most 3 points in the sub-problem, the simulation cost for the sub-problem is at most $54\sigma^2 c^{-2}\log[4T_{max}/\delta]$. □

Using Lemma EC.4, we can estimate the total simulation cost of Algorithm 7.

LEMMA EC.5. *Suppose Assumptions 1-4 hold. The expected simulation cost of Algorithm 7 is bounded by*

$$\frac{459\sigma^2}{c^2}\log\left(\frac{4T_{max}}{\delta}\right) = O\left[\frac{1}{c^2}\log\left(\frac{1}{\delta}\right)\right],$$

where $T_{max} := \log_{1.5}(N) + 2$.

Proof. We denote the upper and the lower bound at the beginning of the k-th iteration as U_k and L_k, respectively. By Lemma EC.4, the simulation cost for the k-th iteration is at most $100\sigma^2 c^{-2}(N^k_{2/3} - N^k_{1/3})^{-2}\log[4T_{max}/\delta]$, where $N^k_{1/3}$ and $N^k_{2/3}$ are the 3-quantiles for the k-th iteration. By the definition of 3-quantiles, it follows that $N^k_{2/3} - N^k_{1/3} \geq (U_k - L_k)/3$ and therefore

$$\frac{100\sigma^2}{(N^k_{2/3} - N^k_{1/3})^2 c^2}\log\left(\frac{4T_{max}}{\delta}\right) \leq \frac{900\sigma^2}{(U_k - L_k)^2 c^2}\log\left(\frac{4T_{max}}{\delta}\right). \tag{EC.4}$$

Hence, we only need to bound the sum $\sum_{k=1}^{T}(U_k - L_k)^{-2}$, where T is the number of iterations of Algorithm 7. By inequality (EC.1), we know

$$U_k - L_k \geq \frac{3}{2}(U_{k+1} - L_{k+1}) - 1 \quad \forall k \in \{1, 2, \ldots, T-1\}.$$

We can rewrite the above inequality as $U_k - L_k - 2 \geq 3/2 \cdot (U_{k+1} - L_{k+1} - 2)$. Since T is the last iteration, it holds that $U_T - L_T \geq 4$ and therefore

$$U_k - L_k - 2 \geq \left(\frac{3}{2}\right)^{T-k}(U_T - L_T - 2) \geq 2 \cdot \left(\frac{3}{2}\right)^{T-k}.$$

Summing over $k = 1, 2, \ldots, T$, we get the bound

$$\sum_{k=1}^{T}(U_k - L_k)^{-2} \leq \sum_{k=1}^{T}\left(2 \cdot \left(\frac{3}{2}\right)^{T-k} + 2\right)^{-2} \leq \sum_{k=1}^{T}\frac{1}{4} \cdot \left(\frac{3}{2}\right)^{-2(T-k)} = \frac{9}{20}\left[1 - \left(\frac{4}{9}\right)^{T}\right] \leq \frac{9}{20}.$$

Combining with inequality (EC.4), the simulation cost for T iterations is at most

$$\frac{900\sigma^2}{c^2}\log\left(\frac{4T_{max}}{\delta}\right) \cdot \sum_{k=1}^{T}(U_k - L_k)^{-2} \leq \frac{405\sigma^2}{c^2}\log\left(\frac{4T_{max}}{\delta}\right).$$

Considering the simulation cost of the sub-problem, the total simulation cost of Algorithm 7 is at most

$$\frac{405\sigma^2}{c^2}\log\left(\frac{4T_{max}}{\delta}\right)+\frac{54\sigma^2}{c^2}\log\left(\frac{4T_{max}}{\delta}\right)=\frac{459\sigma^2}{c^2}\log\left(\frac{4T_{max}}{\delta}\right).$$

□

Finally, we verify the correctness of Algorithm 7 and get an upper bound on $T_0(\delta,\mathcal{MC}_c)$.

Proof of Theorem 2. Similar to the proof of Theorem 1, we use the induction method to prove that Event-I happens for the k-th iteration with probability at least $1-(k-1)\delta/T_{max}$. For the first iteration, the solution to problem (1) is in $\mathcal{X}=\{1,2,\ldots,N\}$ with probability 1. We assume the claim is true for the first $k-1$ iterations and consider the k-th iteration. If one of the first two conditions in line 8 holds when the sampling process terminates, then, by the same analysis as the proof of Theorem 1, we know Event-I happens for the k-th iteration with probability at least $1-(k-1)\delta/T_{max}$. Hence, we only need consider the case when only the last condition in line 8 holds when the sampling process terminates. Since the first two conditions in line 8 do not hold, we know

$$\left|\hat{F}_n(N_{1/3})-\hat{F}_n(N_{2/3})\right|\le(N_{2/3}-N_{1/3})\cdot c/5. \tag{EC.5}$$

In addition, it holds

$$\left|f(N_{1/3})-\hat{F}_n(N_{1/3})\right|\le(N_{2/3}-N_{1/3})\cdot c/5,\quad \left|f(N_{2/3})-\hat{F}_n(N_{2/3})\right|\le(N_{2/3}-N_{1/3})\cdot c/5$$

with probability at least $1-\delta/T_{max}$. Combining with inequality (EC.5), we know that

$$\left|f(N_{1/3})-f(N_{2/3})\right|\le(N_{2/3}-N_{1/3})\cdot 4c/5<(N_{2/3}-N_{1/3})\cdot c \tag{EC.6}$$

holds with probability at least $1-\delta/T_{max}$. We assume the above event and Event-I for the $(k-1)$-th iteration both hold, which has joint probability at least $1-\delta/T_{max}-(k-2)\delta/T_{max}=1-(k-1)\delta/T_{max}$. If the solution to problem (1) is not in $\{N_{1/3},\ldots,N_{2/3}\}$, then function $f(x)$ is monotone in $\{N_{1/3},\ldots,N_{2/3}\}$ and

$$\left|f(N_{1/3})-f(N_{2/3})\right|=\sum_{x=N_{1/3}}^{N_{2/3}-1}\left|f(x)-f(x+1)\right|.$$

Since the indifference zone parameter is c and the function $f(x)$ is convex, we know the function value difference between any two neighbouring points is at least c, which implies that

$$\sum_{x=N_{1/3}}^{N_{2/3}-1} |f(x)-f(x+1)| \geq (N_{2/3})-N_{1/3})\cdot c.$$

However, the above inequality contradicts with inequality (EC.6) and thus we know the solution to problem (1) is in $\{N_{1/3},\ldots,N_{2/3}\}$. Hence, Event-I happens for the k-th iteration with probability at least $1-(k-1)\delta/T_{max}$.

Suppose there are T iterations in Algorithm 7. Since the updating rule of intervals is not changed, Lemma EC.1 gives $T \leq T_{max}-1$. By the induction method, the solution to problem (1) is in $\{L_{T+1},\ldots,U_{k+1}\}$ with probability at least $1-T\cdot\delta/T_{max} \geq 1-\delta+\delta/T_{max}$. Using the same analysis as Theorem 1, the point returned by the sub-problem is at most $2c/3$ larger than the optimal value with probability at least $1-\delta$. By the assumption that the indifference zone parameter is c, all feasible points have function values at least c larger than the optimal value. This implies that the solution returned by Algorithm 7 is optimal with probability at least $1-\delta+\delta/T_{max}-\delta/T_{max} \geq 1-\delta$ and Algorithm 7 is a $[(c,\delta)$-PCS-IZ$,\mathcal{MC}_c]$-algorithm. □

EC.2.3. Proof of Theorem 3

We first estimate the simulation cost of Algorithm 2.

LEMMA EC.6. *Suppose Assumptions 1-4 hold. The expected simulation cost for Algorithm 2 is at most*

$$\frac{25600\sigma^2}{\epsilon^2}\log\left[\frac{4N}{\delta}\right] = O\left[\frac{1}{\epsilon^2}\log\left(\frac{1}{\delta}\right)\right].$$

Proof. Denote $T_{max} := N$. Suppose there are T iterations in Algorithm 2. We denote $\mathcal{S}_k$ as the active set at the beginning of the k-th iteration. Since each iteration reduces the size of $\mathcal{S}_k$ by at least 1, it follows that

$$|\mathcal{S}_k| \geq |\mathcal{S}_{T+1}| + T + 1 - k \quad \forall k \in [T+1].$$

By the same analysis as Lemma EC.2, we know that for the k-th iteration, simulating

$$n(|\mathcal{S}_k|) := \frac{12800\sigma^2}{|\mathcal{S}_k|^2\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)$$

times is sufficient to achieve $1-\delta/(2T_{max})$ confidence width $|\mathcal{S}_k|/80\cdot\epsilon$. Considering the condition on line 8, each point discarded during the k-th iteration is simulated at most $n(|\mathcal{S}_k|)$ times. Hence, the total number of simulations on points discarded during the k-th iteration is at most

$$\begin{aligned}(|\mathcal{S}_k|-|\mathcal{S}_{k+1}|)\cdot n(|\mathcal{S}_k|) &= \frac{|\mathcal{S}_k|-|\mathcal{S}_{k+1}|}{|\mathcal{S}_k|^2}\cdot\frac{12800\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)\\ &\le \left(\frac{1}{|\mathcal{S}_{k+1}|}-\frac{1}{|\mathcal{S}_k|}\right)\cdot\frac{12800\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right),\end{aligned}$$

where the inequality is because of $|\mathcal{S}_k|\ge|\mathcal{S}_{k+1}|$. Summing over $k=1,2,\ldots,T$, we get the number of simulations on all discarded points during iterations is at most

$$\begin{aligned}\sum_{k=1}^{T}\left(\frac{1}{|\mathcal{S}_{k+1}|}-\frac{1}{|\mathcal{S}_k|}\right)\cdot\frac{12800\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right) &= \left(\frac{1}{|\mathcal{S}_{T+1}|}-\frac{1}{|\mathcal{S}_1|}\right)\cdot\frac{12800\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)\\ &\le \left(1-\frac{1}{N}\right)\cdot\frac{12800\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)\le\frac{12800\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right).\end{aligned}$$

For points in the last active set $\mathcal{S}_{T+1}$, the number of simulations is bounded by

$$|\mathcal{S}_{T+1}|\cdot n(|\mathcal{S}_{T+1}|)=\frac{12800\sigma^2}{|\mathcal{S}_{T+1}|\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right)\le\frac{12800\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right).$$

Considering two parts, we know that the simulation cost of Algorithm 2 is at most

$$\frac{25600\sigma^2}{\epsilon^2}\log\left(\frac{4T_{max}}{\delta}\right).$$

□

Then, we prove the correctness of Algorithm 2. The following lemma plays a critical role in verifying the correctness of Type-II Operations.

LEMMA EC.7. *Suppose function $h(x)$ is convex on $[1,M+a]$, where integer $M\ge3$ and constant $a\in[0,1]$. Then, the restriction of function $h(x)$ to $[M]$, which we denote as $\tilde{h}(x)$, is also convex. Furthermore, given a constant $\epsilon>0$, if it holds that*

$$\max_{x\in[M]}\ \tilde{h}(x)-\min_{x\in[M]}\ \tilde{h}(x)\le M/20\cdot\epsilon, \tag{EC.7}$$

then we know

$$\min_{y\in[M']}\ \tilde{h}(2y-1)-\min_{x\in[1,M+a]}\ h(x)\le\epsilon/2.$$

where we define $M':=\lceil M/2\rceil$.

Proof. Since the midpoint convexity of $h(x)$ implies the discrete midpoint convexity of $\tilde{h}$, we know $\tilde{h}(x)$ is also convex. We prove the second claim in three steps.

Step 1. We first prove that

$$\min_{x\in[1,M]} h(x) - \min_{x\in[1,M+a]} h(x) \le \epsilon/10. \tag{EC.8}$$

Suppose x^* is a minimizer of $h(x)$ on $[1, M+a]$. If $x^* \in [1, M]$, then inequality (EC.8) holds trivially. We assume that $x^* \in (M, M+a]$. By the convexity of $h(x)$, we have

$$h(M) - h(x^*) \le \frac{x^* - M}{M-1} \cdot [h(M) - h(1)] \le \frac{1}{M-1} \cdot [h(M) - h(1)] \le \frac{M/20 \cdot \epsilon}{M/2} = \epsilon/10.$$

Hence, we know

$$\min_{x\in[1,M]} h(x) - \min_{x\in[1,M+a]} h(x) = \min_{x\in[1,M]} h(x) - h(x^*) \le h(M) - h(x^*) \le \epsilon/10.$$

Step 2. Next, we prove that

$$\min_{x\in[M]} \tilde{h}(x) - \min_{x\in[1,M]} h(x) \le \epsilon/5. \tag{EC.9}$$

Let x^* be a minimizer of $\tilde{h}(x)$ on $[M]$. By inequality (EC.7), we know

$$\max_{x\in[M]} \tilde{h}(x) = \max\{\tilde{h}(1), \tilde{h}(M)\} \le \tilde{h}(x^*) + M/20 \cdot \epsilon.$$

By the convexity of $h(x)$, there exists a minimizer $x^{**} \in [1, M]$ of $h(x)$ in $(x^* - 1, x^* + 1)$. If $x^{**} = x^*$, then $\min \tilde{h}(x) = \min h(x)$ and inequality (EC.9) holds. Hence, we assume that $x^{**} \neq x^*$ and, without loss of generality, $x^{**} \in (x^*, x^* + 1)$. Since $(M - x^* - 1) + (x^* - 1) = M - 2$, we know $\max\{M - x^* - 1, x^* - 1\} \ge \lceil (M-2)/2 \rceil$. We first consider the case when

$$M - x^* - 1 \ge \lceil (M-2)/2 \rceil.$$

By the convexity of $h(x)$, we have

$$\begin{aligned} h(x^* + 1) - h(x^{**}) &\le \frac{x^* + 1 - x^{**}}{M - x^* - 1} \cdot [h(M) - h(x^* + 1)] \\ &\le \frac{1}{\lceil (M-2)/2 \rceil} \cdot [h(M) - h(x^*)] \le \frac{M/20 \cdot \epsilon}{\lceil (M-2)/2 \rceil}. \end{aligned}$$

By simple calculations, we get $M/4 \leq \lceil (M-2)/2 \rceil$ for all $M \geq 3$ and therefore

$$h(x^*) - h(x^{**}) \leq h(x^*+1) - h(x^{**}) \leq \epsilon/5,$$

which means inequality (EC.9) holds. Now we consider the case when

$$x^* - 1 \geq \lceil (M-2)/2 \rceil.$$

Similarly, by the convexity of $h(x)$, we have

$$h(x^*) - h(x^{**}) \leq \frac{x^{**} - x^*}{x^* - 1} \cdot [h(x^*) - h(1)] \leq \frac{1}{\lceil (M-2)/2 \rceil} \cdot [h(x^*) - h(1)] \leq \frac{M/20 \cdot \epsilon}{\lceil (M-2)/2 \rceil} \leq \epsilon/5.$$

Combining the two cases, we know inequality (EC.9) holds.

Step 3. Finally, we prove that

$$\min_{y \in [M']} \tilde{h}(2y-1) - \min_{x \in [M]} \tilde{h}(x) \leq \epsilon/5. \tag{EC.10}$$

Let x^* be a minimizer of $\tilde{h}(x)$. If x^* is an odd number, then $\min_{y \in [M']} \tilde{h}(2y-1) = \min_{x \in [M]} \tilde{h}(x)$ and inequality (EC.10) holds. Otherwise, we assume $x^* = 2y^*$ is an even number. Then, by the convexity of $\tilde{h}(x)$, there exists a minimizer of $\tilde{h}(2y-1)$ in $\{y^*, y^*+1\}$. Without loss of generality, we assume y^*+1 is a minimizer of $\tilde{h}(2y-1)$. Since $(M-x^*)+(x^*-1) = M-1$, we have $\max\{M-x^*, x^*-1\} \geq \lceil (M-1)/2 \rceil$. We first consider the case when

$$M - x^* \geq \lceil (M-1)/2 \rceil.$$

By the convexity of $\tilde{h}(x)$, we have

$$\tilde{h}(2y^*+1) - \tilde{h}(2y^*) \leq \frac{1}{M - 2y^*} \cdot \left[\tilde{h}(M) - \tilde{h}(2y^*)\right] \leq \frac{M/20 \cdot \epsilon}{\lceil (M-1)/2 \rceil}.$$

We can verify that $M/4 \leq \lceil (M-1)/2 \rceil$ for all $M \geq 3$. Hence, it holds that

$$\tilde{h}(2y^*+1) - \tilde{h}(2y^*) \leq \epsilon/5.$$

Then, we consider the case when

$$x^* - 1 \geq \lceil (M-1)/2 \rceil.$$

Similarly, using the convexity of $\tilde{h}(x)$, we have

$$\tilde{h}(2y^*-1)-\tilde{h}(2y^*)\le\frac{1}{2y^*-1}\cdot\left[\tilde{h}(2y^*)-\tilde{h}(1)\right]\le\frac{M/20\cdot\epsilon}{\lceil(M-1)/2\rceil}\le\epsilon/5,$$

which implies that

$$\tilde{h}(2y^*+1)-\tilde{h}(2y^*)\le\tilde{h}(2y^*-1)-\tilde{h}(2y^*)\le\epsilon/5.$$

Combining the two cases, we know inequality (EC.10) holds.

By inequalities (EC.8), (EC.9) and (EC.10), we have

$$\min_{y\in[M']}\ \tilde{h}(2y-1)-\min_{x\in[1,M]}\ h(x)\le\epsilon/10+\epsilon/5+\epsilon/5=\epsilon/2.$$

□

We denote $\mathcal{S}_k$ and d_k as the active set and the step size at the beginning of the k-th iteration, respectively. We define the upper bound and the lower bound for the k-th iteration as

$$L_1:=1,\quad L_{k+1}:=\begin{cases}y+d_k & \text{if the second case of Type-I Operation happens}\\ L_k & \text{otherwise,}\end{cases}$$

$$U_1:=N,\quad U_{k+1}:=\begin{cases}y-d_k & \text{if the first case of Type-I Operation happens}\\ U_k & \text{otherwise.}\end{cases}$$

Although not explicitly defined in the algorithm, the interval $\{L_k,\dots,U_k\}$ plays a similar role as in the adaptive sampling algorithm and characterizes the set of possible solutions. In the following lemma, we prove that the active set $\mathcal{S}_k$ is a good approximation to the interval $\{L_k,\dots,U_k\}$. We note that the following lemma is deterministic.

LEMMA EC.8. *For any iteration k, we have*

$$L_k=\min\ \mathcal{S}_k\quad \textit{and}\quad U_k\le\max\ \mathcal{S}_k+d_k. \tag{EC.11}$$

Proof. We use the induction method to prove the result. When $k=1$, we know $L_1=1,U_1=N,\mathcal{S}=[N]$ and $d_1=1$. Hence, the relations in (EC.11) hold. We assume these relations hold for the first $k-1$ iterations. We discuss by two different cases.

Case I. Type-I Operation is implemented during the $(k-1)$-th iteration. If the first case of Type-I Operation happens, then we know $L_k = L_{k-1}$ and $U_k = y - d_{k-1}$. By the updating rule, the step size d_{k-1} is not changed and all points in $\mathcal{S}_{k-1}$ that are at least y are discarded from $\mathcal{S}_{k-1}$. Hence, it follows that $\max\ \mathcal{S}_k = U_k$ and the inequality $U_k \leq \max\ \mathcal{S}_k + d_k$ holds. Moreover, since both L_k and $\min\ \mathcal{S}_{k-1}$ are not changed, the equality $L_k = \min\ \mathcal{S}_k$ still holds.

Otherwise if the second case of Type-I Operation happens, then we know $L_k = y + d_{k-1}$ and $U_k = U_{k-1}$. Similarly, we can prove that $L_k = \min\ \mathcal{S}_k$. Moreover, since $d_k = d_{k-1}$ and $\max\ \mathcal{S}_{k-1} + d_{k-1} = \max\ \mathcal{S}_k$, it holds

$$U_k = U_{k-1} \leq \max\ \mathcal{S}_{k-1} + d_{k-1} = \max\ \mathcal{S}_k + d_k.$$

Case II. Type-II Operation is implemented during the $(k-1)$-th iteration. In this case, bounds L_{k-1} and U_{k-1} are not changed. By the update rule, we know the step size $d_k = 2d_{k-1}$ and

$$\min\ \mathcal{S}_k = \min\ \mathcal{S}_{k-1}, \quad \max\ \mathcal{S}_k \in \{\max\ \mathcal{S}_{k-1} - d_{k-1}, \max\ \mathcal{S}_{k-1}\}. \tag{EC.12}$$

Thus, the equality $L_k = \min\ \mathcal{S}_k$ still holds. By the induction assumption, we know that

$$U_k = U_{k-1} \leq \max\ \mathcal{S}_{k-1} + d_{k-1}.$$

Combining with the latter relation in (EC.12), we get

$$U_k \leq \max\ \mathcal{S}_{k-1} + d_{k-1} \leq \max\ \mathcal{S}_k + 2d_{k-1} = \max\ \mathcal{S}_k + d_k.$$

Combining the two cases, we know the relations in (EC.11) hold for the k-th iteration. By the induction method, the relations hold for all iterations. □

Finally, utilizing Lemmas EC.7 and EC.8, we can prove the correctness of Algorithm 2 and get a better upper bound on $T_0(\epsilon, \delta, \mathcal{MC})$.

Proof of Theorem 3. Denote $T_{max} := N$. We use the induction method to prove that, for any iteration k, the two events

- $\min_{x \in \mathcal{S}_k}\ f(x) \leq \min_{x \in \mathcal{X}}\ f(x) + \epsilon/2$.
- $\min_{x \in \{L_k, L_k+1, \ldots, U_k\}}\ f(x) = \min_{x \in \mathcal{X}}\ f(x)$.

happen jointly with probability at least $1-(k-1)\delta/T_{max}$. When $k=1$, we know $\mathcal{S}_1=\mathcal{X}$ and $L_1=1, U_1=N$. Hence, the two events happen with probability 1. Suppose the claim is true for the first $k-1$ iterations. We assume the two events happen for the $(k-1)$-th iteration and consider conditional probabilities in the following proof. We discuss by two different cases.

Case I. Type-I Operation is implemented in the $(k-1)$-th iteration. In this case, there exists $x,y\in\mathcal{S}_{k-1}$ such that $\hat{F}_{n_x}(x)+h_x\le\hat{F}_{n_y}(y)-h_y$. By the definition of confidence intervals, we know $f(x)\le f(y)$ holds with probability at least $1-\delta/T_{max}$. We assume event $f(x)\le f(y)$ happens jointly with the claim for the $(k-1)$-th iteration, which has probability at least $1-(k-2)\delta/T_{max}-\delta/T_{max}=1-(k-1)\delta/T_{max}$. If $x<y$, then using the convexity of $f(x)$, we know

$$f(z)\ge f(y)\ge f(x)\quad\forall z\in[N]\quad\text{s.t. } z\ge y,$$

which means all discarded points have function values at least $f(x)$. Hence, the minimums in the claim are not changed, i.e., we have

$$\min_{x\in\mathcal{S}_k}\ f(x)=\min_{x\in\mathcal{S}_{k-1}}\ f(x)\le\min_{x\in\mathcal{X}}\ f(x)+\epsilon/2$$

and

$$\min_{x\in\{L_k,L_k+1,\ldots,U_k\}}\ f(x)=\min_{x\in\{L_{k-1},L_{k-1}+1,\ldots,U_{k-1}\}}\ f(x)=\min_{x\in\mathcal{X}}\ f(x).$$

The two events happen with probability at least $1-(k-1)\delta/T_{max}$. If $y<x$, the proof is the same and therefore the claim holds for the k-th iteration.

Case II. Type-II Operation is implemented in the $(k-1)$-th iteration. Since L_{k-1} and U_{k-1} are not changed, the first event happens for the k-th iteration. Hence, we only need to verify that the second event happens with high probability. Let x^* and x^{**} be a minimizer and a maximizer of $f(x)$ on $\mathcal{S}_{k-1}$, respectively. By the condition of Type-II Operations, we know

$$|\hat{F}_n\ (x^*)-\hat{F}_n(x^{**})|\le h_{x^*}+h_{x^{**}}$$

and

$$h_x\le|\mathcal{S}_{k-1}|/80\cdot\epsilon\quad\forall x\in\mathcal{S}_{k-1}.$$

By the definition of confidence intervals, it holds

$$|f(x^*) - \hat{F}_n\ (x^*)| \le h_{x^*}, \quad |f(x^{**}) - \hat{F}_n\ (x^{**})| \le h_{x^{**}}$$

with probability at least $1 - \delta/T_{max}$. Under the above event, we have

$$\begin{aligned} |f(x^*) - f(x^{**})| &\le |f(x^*) - \hat{F}_n\ (x^*)| + |\hat{F}_n\ (x^*) - \hat{F}_n(x^{**})| + |f(x^{**}) - \hat{F}_n\ (x^{**})| \\ &\le 2(h_{x^*}) + h_{x^{**}} \le M/20 \cdot \epsilon. \end{aligned} \tag{EC.13}$$

We assume the above event happens jointly with with the claim for the $(k-1)$ -th iteration, which has probability at least $1-(k-1)\delta/T_{max}$. By the induction assumption, the original problem (1) is equivalent to

$$\min_{x \in \{L_{k-1}, L_{k-1}+1, \ldots, U_{k-1}\}}\ f(x).$$

Moreover, if we denote $\tilde{f}$ as the linear interpolation of $f(x)$ defined in (2), then the above problem is equivalent to

$$\min_{x \in \{L_{k-1}, L_{k-1}+1, \ldots, U_{k-1}\}}\ f(x) = \min_{x \in [L_{k-1}, U_{k-1}]}\ \tilde{f}(x). \tag{EC.14}$$

We define the constant $\tilde{M} := (U_{k-1} - L_{k-1})/d_{k-1} + 1$ and the linear transformation

$$T(x) := L_{k-1} + d_{k-1}(x-1) \quad \forall x \in [1, \tilde{M}].$$

The inverse image $T^{-1}\left([L_{k-1}, U_{k-1}]\right)$ is $[1, \tilde{M}]$. Defining the composite function

$$\tilde{g}(x) := \tilde{f}(T(x)) \quad \forall x \in [1, \tilde{M}],$$

we know that the problem (EC.14) is equivalent to

$$\min_{x \in [1, \tilde{M}]}\ \tilde{g}(x). \tag{EC.15}$$

The inverse image $T^{-1}(\mathcal{S}_{k-1})$ is $[M]$, where $M := |\mathcal{S}_{k-1}|$ is the number of points in $\mathcal{S}_{k-1}$. Lemma EC.8 implies that $a := \tilde{M} - M \in [0, 1]$. Recalling inequality (EC.13), we get

$$\max_{x \in [M]}\ \tilde{g}(x) - \min_{x \in [M]}\ \tilde{g}(x) \le M/20 \cdot \epsilon.$$

Now we can apply Lemma EC.7 to get

$$\min_{y\in[M']} \tilde{g}(2y-1) - \min_{x\in[1,M+a]} \tilde{g}(x) = \min_{y\in[M']} \tilde{g}(2y-1) - \min_{x\in[1,\tilde{M}]} \tilde{g}(x) \le \epsilon/2,$$

where $M' := \lceil M/2 \rceil$. Since problem (EC.15) is equivalent to problem (EC.14) and further equivalent to problem (1), it holds that

$$\min_{y\in[M']} \tilde{g}(2y-1) - \min_{x\in[1,\tilde{M}]} \tilde{g}(x) = \min_{y\in[M']} \tilde{g}(2y-1) - \min_{x\in\mathcal{X}} f(x) \le \epsilon/2.$$

By the definition of $\mathcal{S}_k$, we know $T(2[M']-1)$ is $\mathcal{S}_k$ and therefore

$$\min_{y\in[M']} \tilde{g}(2y-1) - \min_{x\in\mathcal{X}} f(x) = \min_{x\in\mathcal{S}_k} f(x) - \min_{x\in\mathcal{X}} f(x) \le \epsilon/2,$$

which implies that the second case happens for the k-th iteration with probability at least $1-(k-1)\delta/T_{max}$.

Combining the above two cases, we know the claim holds for all iterations. Suppose there are T iterations in Algorithm 2. Since each iteration will decrease the active set $\mathcal{S}$ by at least 1, we get $T \le N-1$. Then after the T iterations, we have

$$\min_{x\in\mathcal{S}_{T+1}} f(x) \le \min_{x\in\mathcal{X}} f(x) + \epsilon/2 \tag{EC.16}$$

holds with probability at least $1 - T\cdot\delta/T_{max} \ge 1-\delta+\delta/T_{max}$. For the sub-problem, using the same analysis as Theorem 1, the point returned by Algorithm 2 satisfies the $(\epsilon/2, \delta/T_{max})$-PGS guarantee. Combining with the relation (EC.16), we know the algorithm returns a solution satisfying the (ϵ,δ)-PGS guarantee. □

EC.2.4. Proof of Theorem 4

Proof of Theorem 4. We construct the two models $M_1, M_2 \in \mathcal{MC}$ as

$$\nu_{1,x} := \mathcal{N}\left[cx, \sigma^2\right], \quad \nu_{2,x} := \mathcal{N}\left[c\left(|x-2|+2\right), \sigma^2\right] \quad \forall x \in \mathcal{X}.$$

Given a $[(c,\delta)$-PCS-IZ$, \mathcal{MC}_c]$-algorithm, the algorithm returns point 1 with probability at least $1-\delta$ when applied to model M_1, and returns point 2 with probability at least $1-\delta$ when applied

to model M_2. We choose $\mathcal{E}$ as the event that the algorithm returns point 1 as the solution. Then, we know

$$\mathbb{P}_{M_1}(\mathcal{E}) \geq 1-\delta, \quad \mathbb{P}_{M_2}(\mathcal{E}) \leq \delta.$$

Using the monotonicity of function $d(x,y)$, we get

$$d(\mathbb{P}_{M_1}(\mathcal{E}), \mathbb{P}_{M_2}(\mathcal{E})) \geq d(1-\delta,\delta) \geq \log(1/2.4\delta). \tag{EC.17}$$

Since the distributions $\nu_{1,x}$ and $\nu_{2,x}$ are Gaussian with variance σ^2, the KL divergence can be calculated as

$$\mathrm{KL}(\nu_{1,x},\nu_{2,x}) = \frac{[cx - c(|x-2|+2)]^2}{2\sigma^2} = \begin{cases} 2c^2\sigma^{-2} & \text{if } x=1 \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the summation can be calculated as

$$\sum_{x\in\mathcal{X}} \mathbb{E}_{M_1}[N_x(\tau)]\,\mathrm{KL}(\nu_{1,x},\nu_{2,x}) = \frac{2c^2}{\sigma^2}\cdot \mathbb{E}_{M_1}[N_1(\tau)]. \tag{EC.18}$$

Substituting (EC.17) and (EC.18) into inequality (3), we know

$$\frac{2c^2}{\sigma^2}\cdot \mathbb{E}_{M_1}[N_1(\tau)] \geq \log(1/2.4\delta),$$

which implies that

$$\mathbb{E}_{M_1}[\tau] \geq \mathbb{E}_{M_1}[N_1(\tau)] \geq \frac{\sigma^2}{2c^2}\cdot\log(\frac{1}{2.4\delta}) = O\left[\frac{1}{c^2}\log\left(\frac{1}{\delta}\right)\right].$$

□

EC.3. Proofs in Section 4

EC.3.1. Proof of Lemma 4

Proof of Lemma 4. By the assumption that $F(x,\xi_x) - f(x)$ is sub-Gaussian with parameter σ^2 for any x, we know that $\hat{g}_{\alpha_x(i)} - g_{\alpha_x(i)}$ is the difference of two independent sub-Gaussian random variables and therefore

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

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

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

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

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

Since $\|y-x\|_2^2 \le dN^2$, 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

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

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

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

it follows that

$$|\langle \hat{g}^n - g, y - x\rangle| \le \epsilon. \tag{EC.19}$$

Since $f(x)$ is a convex function and g is a subgradient at point x, we have $f(y) \ge f(x) + \langle g, y-x\rangle$ for all $y \in [1,N]^d$. Combining with inequality (EC.19) gives

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

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

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

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

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

□

EC.3.2. Proof of Theorem 7

Proof of Theorem 7. We first prove the correctness of Algorithm 4. If $\hat{g}=0$ for some iteration, the half space $H=\mathbb{R}^d$ and the definition of $(\epsilon/8,\delta/4)$-$\mathcal{SO}$ implies that

$$f(y)\geq f(z)-\epsilon/8 \quad \forall y\in[1,N]^d$$

holds with probability at least $1-\delta/4$, where z is the point that the separation oracle is called. Hence, we know z is an $(\epsilon/8,\delta/4)$-PGS solution and obviously satisfies the $(\epsilon/2,\delta/2)$-PGS guarantee. Then, by Theorem 6, the integral solution after the round process is an (ϵ,δ)-PGS solution.

In the following of the proof, we assume $\hat{g}\neq 0$ for all iterations. Let $x^*\in\mathcal{X}$ be a minimizer of problem (1). We consider the set

$$Q:=\left(x^*+\left[-\frac{\epsilon}{8L},\frac{\epsilon}{8L}\right]^d\right)\cap[1,N]^d.$$

We can verify that set Q is not empty and has volume at least $(\epsilon/8L)^d$. Also, for any $x\in Q$, it holds

$$f(x)\leq f(x^*)+L\|x-x^*\|_\infty\leq f(x^*)+\frac{\epsilon}{8}.$$

By the analysis in Vaidya (1996), the volume of the polytope P is smaller than $(\epsilon/8L)^d$ after

$$T_{max}:=O\left[d\log\left(\frac{8dLN}{\epsilon}\right)\right]$$

iterations. Hence, after T_{max} iterations, the volume of P is smaller than the volume of Q and it must hold $Q\backslash P\neq\emptyset$. Since $Q\subset[1,N]^d$, the constraint $1\leq x_i\leq N$ is not violated for all $i\in[d]$. Thus, if we choose $x\in Q\backslash P$, there exists a cutting plane $-\hat{g}^Ty\geq\beta$ in P such that

$$-\hat{g}^Tx<\beta\leq-\hat{g}^Tz,$$

where z is the point that the $(\epsilon/8,\delta/4)$-$\mathcal{SO}$ oracle $\hat{g}$ is evaluated and β is the value chosen by Vaidya's method. This implies that x is not in the half space

$$H:=\{y:\hat{g}^Ty\leq\hat{g}^Tz\}.$$

Then, by the definition of $(\epsilon/8, \delta/4)$-$\mathcal{SO}$ oracle and the claim that $x \in [1, N]^d \cap H^c$, we know

$$f(x) \geq f(z) - \epsilon/8$$

holds with probability at least $1 - \delta/4$. On the other hand, the condition $x \in P$ leads to

$$f(x) \leq f(x^*) + \epsilon/8.$$

Combining the last two inequalities gives that

$$\min_{y \in \mathcal{S}} \ f(y) \leq f(z) \leq f(x^*) + \epsilon/4$$

holds with probability at least $1 - \delta/4$. Hence, the $(\epsilon/4, \delta/4)$-PGS solution $\hat{x}$ of problem $\min_{y \in \mathcal{S}} \ f(y)$ satisfies

$$f(\hat{x}) \leq f(x^*) + \epsilon/2$$

with probability at least $1 - \delta/2$. Equivalently, the solution $\hat{x}$ is an $(\epsilon/2, \delta/2)$-PGS solution. Using Theorem 6, the integral solution returned by Algorithm 4 is an (ϵ, δ)-PGS solution.

Now, we estimate the expected simulation cost of Algorithm 4. By Lemma 4, the simulation cost of each $(\epsilon/8, \delta/4)$-$\mathcal{SO}$ oracle is at most

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

Since at most one separation oracle is evaluated in each iteration, the total simulation cost of T_{max} iterations is at most

$$O\left[\left(\frac{d^2 N^2}{\epsilon^2} \log(\frac{1}{\delta}) + d\right) \cdot d \log\left(\frac{8dLN}{\epsilon}\right)\right] = \tilde{O}\left[\frac{d^3 N^2}{\epsilon^2} \log(\frac{dLN}{\epsilon}) \log(\frac{1}{\delta})\right].$$

By the property of Vaidya's method, there are $O(d)$ cutting planes in the polytope P. Then, using the same analysis as Zhang and Zheng (2020), the expected simulation cost of finding an $(\epsilon/4, \delta/4)$-PGS solution of the sub-problem $\min_{y \in \mathcal{S}} \ f(y)$ is at most

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

We note that the evaluation of $(\epsilon/8, \delta/4)$-$\mathcal{SO}$ oracles at points in $\mathcal{S}$ provides enough simulations for the sub-problem and therefore the simulation cost of this part can be avoided. Finally, the expected simulation cost of the rounding process is bounded by

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

Combining the three parts, the total expected simulation cost of Algorithm 4 is at most

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

□

EC.3.3. Proof of Theorem 9

Proof of Theorem 9. 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). Then, by the same proof as Theorem 7, it holds

$$\min_{x\in\mathcal{S}} \ f(x) \le \min_{x\in\mathcal{X}} \ f(x) + \epsilon/4 \tag{EC.20}$$

with probability at least $1-\delta/4$. 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 $(\epsilon/4, \delta/4)$-PGS solution to the one-dimensional problem is also an $(\epsilon/4, \delta/4)$-PGS solution to the original problem. Since the PGS solution is also added to the set $\mathcal{S}$, we also have relation (EC.20) holds with probability at least $1-\delta/4$. Then, the $(\epsilon/4, \delta/4)$-PGS solution $\bar{x}$ to problem $\min_{x\in\mathcal{S}} f(x)$ satisfies

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

with probability at least $1-\delta/2$, or equivalently $\bar{x}$ is an $(\epsilon/2, \delta/2)$-PGS solution to problem (1). Using the results of Theorem 6, the solution returned by Algorithm 5 is an (ϵ, δ)-PGS solution.

Next, we estimate the expected simulation cost 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 4, the total simulation cost of the dimension reduction process is at most

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

Also, the one-dimensional convex problem has at most N feasible points and Theorem 3 implies that the expected simulation cost for this problem is at most

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

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

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

simulation runs. Finally, Theorem 6 shows that the expected simulation cost of the rounding process is at most

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

In summary, the total expected simulation cost of Algorithm 5 is at most

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

□

EC.3.4. Proof of Lemma 5

Proof of Lemma 5. Let $k\in\{2,3,\ldots,N-1\}$. 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

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

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 $[N]$. □

EC.3.5. Proof of Theorem 11

Proof of Theorem 11. We first verify the correctness of Algorithm 6. The algorithm is the same as Algorithm 2 except the condition for implementing Type-II Operations. Hence, if we can prove that, when Type-II Operations are implemented, it holds

$$h \leq |\mathcal{S}|\cdot\epsilon/80, \tag{EC.21}$$

then the proof of Theorem 3 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 (EC.21) and the correctness of Algorithm 6.

Next, we estimate the simulation cost of Algorithm 6. Denote the active sets when we update the confidence interval as $\mathcal{S}_1,\ldots,\mathcal{S}_m$, where $m\geq 1$ is the number of times when the 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 simulation cost for generating $\hat{f}^{d-1}(x)$ is at most

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

Hence, the total simulation cost for the k-th update of confidence intervals is at most

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

Summing over all iterations, we have the simulation cost of all iterations of Algorithm 6 is at most

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

Now we consider the simulation cost of the last subproblem. Since the algorithm 6 is sub-Gaussian with parameter C, the simulation cost of the subproblem is at most

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

Hence, the total simulation cost of Algorithm 6 is at most

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

When δ is small enough, we can choose $M = 17100$ and the asymptotic simulation of Algorithm 6 is at most

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

which implies that Algorithm 6 is sub-Gaussian with dimension d and parameter MC. □