Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Gradient-Based Algorithms for Convex Discrete Optimization via Simulation

Haixiang Zhang, Zeyu Zheng, Javad Lavaei

To cite this article:
Haixiang Zhang, Zeyu Zheng, Javad Lavaei (2022) Gradient-Based Algorithms for Convex Discrete Optimization via Simulation. Operations Research

Published online in Articles in Advance 28 Apr 2022

. https://doi.org/10.1287/opre.2022.2295

http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–20
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

Gradient-Based Algorithms for Convex Discrete Optimization via Simulation

Haixiang Zhang,[a] **Zeyu Zheng,**[b,*] **Javad Lavaei**[b]

[a] Department of Mathematics, University of California, Berkeley, Berkeley, California 94720; [b] Department of Industrial Engineering and Operations Research, University of California, Berkeley, Berkeley, California 94720
* Corresponding author

Contact: haixiang_zhang@berkeley.edu, https://orcid.org/0000-0001-5386-1208 (HZ); zyzheng@berkeley.edu, https://orcid.org/0000-0001-5653-152X (ZZ); lavaei@berkeley.edu, https://orcid.org/0000-0003-4294-1338 (JL)

Received: October 30, 2020
Revised: July 3, 2021; December 11, 2021; February 8, 2022
Accepted: February 11, 2022
Published Online in Articles in Advance: April 28, 2022

Area of Review: Simulation

https://doi.org/10.1287/opre.2022.2295

Abstract. We propose new sequential simulation–optimization algorithms for general convex optimization via simulation problems with high-dimensional discrete decision space. The performance of each choice of discrete decision variables is evaluated via stochastic simulation replications. If an upper bound on the overall level of uncertainties is known, our proposed simulation–optimization algorithms utilize the discrete convex structure and are guaranteed with high probability to find a solution that is close to the best within any given user-specified precision level. The proposed algorithms work for any general convex problem, and the efficiency is demonstrated by proven upper bounds on simulation costs. The upper bounds demonstrate a polynomial dependence on the dimension and scale of the decision space. For some discrete optimization via simulation problems, a gradient estimator may be available at low costs along with a single simulation replication. By integrating gradient estimators, which are possibly biased, we propose simulation–optimization algorithms to achieve optimality guarantees with a reduced dependence on the dimension under moderate assumptions on the bias.

Funding: H. Zhang and J. Lavaei were funded by grants from Air Force Office of Scientific Research (AFOSR), Army Research Office (ARO), Office of Naval Research (ONR), National Science Foundation (NSF) and C3.ai Digital Transformation Institute.
Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2022.2295.

Keywords: discrete optimization via simulation • discrete convex functions • sequential simulation–optimization algorithms • simulation costs • biased gradient estimators

1. Introduction

Many decision-making problems in operations research and management science involve large-scale complex stochastic systems. The objective function in the decision-making problems often involves expected system performances that need to be evaluated by discrete event simulation or general stochastic simulation. The decision variables in many of these problems are naturally discrete-valued and multidimensional. This class of problems is called discrete optimization via simulation; see Hong et al. (2015). Typically, for discrete optimization via simulation problems, continuous approximations are either not naturally available or may incur additional errors that are themselves difficult to accurately quantify; see Nelson (2010). This work is centered around designing and proving theoretical guarantees for simulation–optimization algorithms to solve discrete optimization via simulation problems with multidimensional decision space.

In large-scale complex stochastic systems, one replication of simulation to evaluate the performance of a single decision can be computationally costly. An accurate evaluation of the expected performance associated with a single decision needs many independent replications of simulation. Running simulations for all feasible choices of decision variables in a high-dimensional discrete space to find the optimal is computationally prohibitive. The use of parallel computing (e.g., Luo et al. 2015) may alleviate the computation burden, but to find the best decision in high-dimensional problems can still be challenging. Fortunately, for a number of applications, the objective function exhibits convexity in the discrete decision variables, or the problem can be transformed into a convex one. One such example with convex structure comes from a bike-sharing system (Singhvi et al. 2015, Jian et al. 2016, Freund et al. 2017). This problem involves around 750 stations and 25,000 docks. The goal is to find the optimal allocation

of bikes and docks, which are naturally discrete decision variables. The performance of each allocation is evaluated by the dissatisfaction function, which is defined as the total number of failures to rent or return a bike in a whole day. In the presence of nonstationary exogenous random demands and travel patterns, the evaluation of the dissatisfaction function for a given allocation needs to be done by simulation. This simulation is costly as it may need to simulate the full operation of the system over the entire day. In Freund et al. (2017), the expected dissatisfaction function is proved to be "convex" under a linear transformation if the stochastic arrival processes are exogenous. For this problem, running stochastic simulations for the entire discrete and high-dimensional decision space is computationally prohibitive. It is, therefore, of interest to explore how the convexity structure of the objective function may help solve the simulation–optimization problem. In fact, many performance functions in the operations research and management science domain exhibit convexity in discrete decision variables. For example, the expected customer waiting time in a multiserver queueing network is proved to be convex in the routing policy and staffing decisions; see Altman et al. (2003) and Wolff and Wang (2002). Shaked and Shanthikumar (1988) discuss a wide range of stochastic systems, including queueing, reliability, and branching systems and show the convexity of key expected performance measures as a function of the associated decision variable. In addition, a large variety of problems in economics, computer vision, and network flow optimization exhibit convexity with discrete decision variables (Murota 2003).

Even in the presence of convexity, the nominal task in discrete optimization via simulation—correctly finding the best decision with high enough probability, which is often referred to as the *probability of correct selection* (PCS) guarantee—can still be computationally prohibitive. For a convex problem without convenient assumptions, such as strong and strict convexity, there may be a large number of choices of decision variables that render very close objective value compared with the optimal. In this case, the simulation efforts to identify the exact optimal choice of decision variables can be huge and practically unnecessary. Our focus, alternatively, is to find a good choice of decision variables that is assured to render ϵ-close objective value compared with the optimal with high probability, where ϵ is any arbitrarily small user-specified precision level. This guarantee is also called the *probability of good selection* (PGS) or *probably approximately correct* (PAC) in the literature. This paper adopts the notion of PGS as a guarantee for simulation–optimization algorithm design. We refer to Eckman and Henderson (2021, 2018) for thorough discussions on settings when the use of PGS is preferable compared with the use of PCS. In this work, we propose simulation–optimization algorithms that achieve the PGS guarantee for general discrete convex problems without knowing any further information, such as strong convexity, etc. Knowing strong convexity or a specific parametric function form of the objective function, of course, further enhances the simulation–optimization algorithms. However, such fine structural information may not be available a priori for large-scale simulation optimization problems. The design of our simulation–optimization algorithms utilizes the convex structure, and the intuition is that the convex structure of optimization landscapes can provide *global information* through *local evaluations*. Global information helps the algorithm avoid evaluating all feasible choices of decision variables, which, therefore, avoids spending simulation efforts that are proportional to the number of choices of decision variables and are exponentially dependent on the dimension in general. Our proposed simulation–optimization algorithms are based on stochastic gradient methods and discrete steepest descent methods, which need to be designed as fundamentally different from continuous optimization algorithms. For high-dimensional problems, gradient-based methods are preferred compared with strongly polynomial methods, such as cutting-plane methods, because the simulation costs of gradient-based methods usually have a slower growth rate when the dimension increases.

In order to compare algorithms that all return a solution that achieves the PGS optimality guarantee, we use the metric of expected simulation cost. Intuitively here, but with exact definition to follow in the main body of this work, the expected simulation cost is described by the expected number of simulation replications that are run over the decision space in order to achieve a solution with the PGS guarantee. We prove upper bounds on the expected simulation cost for our proposed simulation–optimization algorithms that achieve the PGS guarantee. The proven upper bounds show a low-order polynomial dependence on the decision space dimension d. Note that the upper bounds hold for any arbitrary convex problem. As a comparison, if the convex structure is not present or utilized, the expected simulation cost to achieve the PGS guarantee can easily be exponential in the dimension d. We also provide lower bounds on the expected simulation costs that are needed for any possible simulation–optimization algorithm. The lower and upper bounds of expected simulation costs imply the limit of algorithm performance and provide directions to improving existing simulation–optimization algorithms. In general, we refer readers to Ma and Henderson (2019) and Zhong and Hong (2021) for more detailed discussions on the use of simulation costs and upper/lower bounds on the order of simulation costs to analyze and compare algorithms.

1.1. Main Results and Contributions

We design gradient-based simulation–optimization algorithms that achieve the PGS guarantee for high-

dimensional and large-scale discrete convex problems with a known upper bound on the level of overall uncertainties. We consider the decision space to be $\{(x_1, x_2 \ldots, x_d) : x_i \in \{1, 2, \ldots, N\}, i \in \{1, 2, \ldots, d\}\}$ that has in total N^d possible choices of decision variables. The discrete convexity in high dimension that preserves the midpoint convexity (namely, the midpoint has an objective value smaller than the average of objective values at the two endpoints) is called $L^\natural$-convexity (Murota 2003). From the optimization perspective, our work addresses the stochastic version of discrete convex analysis in Murota (2003). From the simulation–optimization perspective, this work provides simulation–optimization algorithms with optimality guarantee and polynomial dependence of simulation costs on dimension for high-dimensional discrete convex simulation optimization problems.

We categorize our simulation–optimization algorithms to two classes. One class is the *zeroth order algorithm*, for which the simulation is a black box, and one run of the simulation can only provide an evaluation of a single decision. The other class is the *first order algorithm*, for which the neighboring choices of decision variables can be simultaneously evaluated (possibly results in a biased finite difference gradient estimator) within a single simulation run for a given choice of decision variables. We develop simulation–optimization algorithms with the PGS guarantee as a major focus, but we also provide algorithms with the PCS-IZ guarantee for cases when the indifference zone (IZ) parameter is known. See Hong et al. (2021) for detailed discussions on the PCS-IZ guarantee. We summarize our results in Table 1, in which algorithm performance is demonstrated by the expected simulation cost. In this table, we omit terms in the expected simulation cost that do not depend on the failing probability δ, that is, the probability that the solution does not satisfy the specified precision. Therefore, when δ is very small, the dominating term in the expected computation cost is what we list in Table 1. This comparison scheme is also considered in Kaufmann et al. (2016). That being said, we provide all terms in the upper bounds for expected simulation costs in corresponding theorems.

For zeroth order algorithms, the Lovász (1983) extension is introduced to define a convex linear interpolation of the original discrete function. Using properties of the Lovász extension (Fujishige 2005), it is equivalent to optimize the interpolated continuous function. Therefore, the projected stochastic subgradient descent method can be used to find PGS solutions. Moreover, the truncation of stochastic subgradients is essential in reducing the expected simulation costs, and we prove that the dependence on the dimension d is reduced from $O(d^3)$ to $O(d^2)$ using truncation. In the stochastic optimization literature, it is common to assume the stochastic subgradient is bounded when deriving high-probability bounds, and we also provide a theoretical guarantee under the boundedness assumption. When the boundedness assumption can be verified, the dependence on dimension can be further reduced to $O(d)$. When the indifference zone parameter c is known, an accelerated algorithm is proposed and is proved to reduce the dependence on the scale N from $O(N^2)$ to $O(\log(N))$. Finally, an information-theoretical lower bound is derived to show the limit of simulation–optimization algorithms.

For first order algorithms, we have available gradient information at a cost as a constant multiplying the cost of one simulation run for which the constant does not depend on the dimension. This gradient information is regarded as a subgradient estimator. In practice, the subgradient estimator can be biased, and there is no convergence guarantee for any optimization algorithm in general. However, under a moderate assumption on the bias, we are still able to develop simulation–optimization algorithms that achieve the PGS guarantee through a stochastic version of the steepest descent method. The associated simulation cost does not scale up with d, but the memory cost and the number of arithmetic operations can be much larger than those of simulation–optimization algorithms designed for the unbiased gradient estimators.

Table 1. Upper and Lower Bounds on Expected Simulation Cost for the Proposed Simulation–Optimization Algorithms That Achieve the PGS and the PCS-IZ Guarantees

Algorithms	PGS	PCS-IZ (known IZ parameter c)
Zeroth order algorithm (Gaussian noise)	$\tilde{O}(d^2N^2\epsilon^{-2}\log(1/\delta))$ (Lower bound: $\tilde{O}(d\epsilon^{-2}\log(1/\delta))$)	$\tilde{O}(d^2\log(N)c^{-2}\log(1/\delta))$
Zeroth order algorithm (Assumption EC.1 in the supplementary material)	$\tilde{O}(dN^2\epsilon^{-2}\log(1/\delta))$	$\tilde{O}(d\log(N)c^{-2}\log(1/\delta))$
Lower bound	$\tilde{O}(d\epsilon^{-2}\log(1/\delta))$	$\tilde{O}(dc^{-2}\log(1/\delta))$
Biased first order algorithm (Assumption 5)	$\tilde{O}(N^3\epsilon^{-2}\log(1/\delta))$ (requires additional memory cost)	$\tilde{O}(Nc^{-2}\log(1/\delta))$

Notes. Constants and terms that do not depend on δ are omitted in the $\tilde{O}(\cdot)$ notation. In comparison, the expected simulation cost without $L^\natural$-convexity is $\tilde{O}(N^d\epsilon^{-2}\log(1/\delta))$. Here, d and N are the problem dimension and scale, the feasible set is $\{1, \ldots, N\}^d$, constants ϵ and δ are the precision and failing probability of algorithms.

Finally, utilizing the indifference zone, the expected simulation cost can be reduced from $O(N^3)$ to $O(N)$ in terms of dependence on N.

1.2. Literature Review

The problem of selecting the best or a good choice of decision variables through simulation is widely studied in the simulation literature. The problem is often called ranking-and-selection (R&S). We refer to Hong et al. (2021) for a recent review of this literature. There are two approaches to categorize the R&S literature. One approach is differentiating the frequentist and the Bayesian views when describing the probability models and procedures in R&S; see Kim and Nelson (2006) and Chick (2006). The other approach differentiates fixed confidence and fixed budget procedures; see Hunter and Nelson (2017) and Hong et al. (2021). In particular, the PCS of the best choice of decision variables is a widely used guarantee for both types of procedures. Generally, in R&S problems, there is no structural information such as convexity that is considered.

A large number of R&S procedures based on the PCS guarantee adopt the indifference zone formulation called PCS-IZ. The PCS-IZ guarantee is built upon the assumption that the expected performance of the best choice of decision variables is at least $c > 0$ better than all other choices of decision variables. This IZ parameter c is typically assumed to be known, whereas Fan et al. (2016), as a notable exception, provides selection guarantees without the knowledge of the indifference zone parameter. In practice, for some problem settings, this IZ parameter may be unknown a priori. When many choices of decision variables have close performance compared with the best, it is practically inefficient to select the exact best. In this case, choices of decision variables that are close enough to the best are referred to as "good choices," and any one of them can be satisfying. This naturally gives rise to a notion of PGS. Eckman and Henderson (2021, 2018) thoroughly discuss settings in which the use of PGS is preferable to the use of PCS-IZ.

Discussions on discrete optimization via simulation can be found in Fu (2002), Nelson (2010), Sun et al. (2014), Park et al. (2014), Park and Kim (2015), Hong et al. (2015), and Chen et al. (2018), among others. Hu et al. (2007, 2008) discuss model reference adaptive search algorithms in order to ensure global convergence. Hong and Nelson (2006), Hong et al. (2010), and Xu et al. (2010) propose and study algorithms based on the convergent optimization via most promising area stochastic search that can be used to solve general simulation–optimization problems with discrete decision variables. The proposed algorithms are computationally efficient and are proven to converge with probability one to optimal points. Lim (2012) studies simulation–optimization problems over multidimensional discrete sets in which the objective function adopts multimodularity, which is equivalent to the submodularity under a linear transform; see two equivalent definitions of multimodular functions in Altman et al. (2000) and Murota (2003). They propose algorithms that converge almost surely to the global optimal. Wang et al. (2013) discuss stochastic optimization problems with integer-ordered decision variables. Eckman et al. (2022) discuss a statistically guaranteed screening to rule out decisions based on initial simulation experiments utilizing the convex structure.

When a simulation problem involves a response surface to estimate or optimize over, gradient information may be constructed and used to enhance simulation. Chen et al. (2013) constructs a gradient estimator to enhance simulation metamodeling. Qu and Fu (2014) propose a new approach called gradient extrapolated stochastic kriging that exploits the extrapolation structure. Fu and Qu (2014) discuss the use of Monte Carlo gradient estimators to enhance regression. See also L'Ecuyer (1990) for a review of Monte Carlo gradient estimators. Eckman and Henderson (2020) discuss the use of possibly biased gradient estimators in continuous stochastic optimization by assuming that the bias is uniformly bounded. Wang et al. (2021) consider a setting in which the response surface is a quadratic function and gradient information is available and discuss optimal budget allocation to maximize the probability of correct selection. In general simulation–optimization problems, when the decision variables are discrete, the gradient with respect to the decision variable may not be appropriately defined. Instead, the difference of performance between two neighboring choices of decision variables contains gradient-like information. Jian (2017) uses this information to guide the search for the optimal choices of decision variables.

Discrete optimization via simulation is also formulated as the best-arm identification problem or the pure exploration multiarmed bandit problem. The best-arm identification literature usually does not consider the problem structure or the high-dimensional nature of an arm. More recent works focus on general distribution families and utilize techniques from information theory. Informational upper bounds and lower bounds for exponential bandit models are established by the change-of-measure technique in Kaufmann and Kalyanakrishnan (2013) and Kaufmann et al. (2016). In Garivier and Kaufmann (2016), a transportation inequality is proved and a general nonasymptotic lower bound can be formulated thorough the solution of a max-min optimization problem. Agrawal et al. (2020) show that restrictions on the distribution family are necessary and generalize the algorithm to models with milder restriction than an exponential family.

Discrete optimization via simulation problems fall into the more general class of problems called discrete

stochastic optimization. In contrast to continuous optimization, most works on discrete stochastic optimization (Futschik and Pflug 1995, 1997; Gutjahr and Pflug 1996; Kleywegt et al. 2002; Semelhago et al. 2020) do not consider the convex structure. The main obstacle to the development of discrete convex optimization lies in the lack of a suitable definition of the discrete convex structure. A natural definition of the discrete convex functions is functions that are extensible to continuous convex functions. However, for that class of functions, the local optimality does not imply the global optimality, and therefore, it is not suitable for the purpose of optimization. An example with spurious local minima is given in Section 2.3. Later, Favati (1990) proposes a stronger condition, named integral convexity, that ensures the local optimality is equivalent to the global optimality. On the other hand, after Lovász (1983) shows the equivalence between the submodularity of a function and the convexity of its Lovász extension, submodular functions are viewed as the discrete analogy of convex functions in the field of combinatorial optimization. The Fenchel-type min-max duality theorem (Fujishige 1984) and the subgradient (Fujishige 2005) of submodular functions provide a good framework of applying the gradient-based method to the submodular function minimization (SFM) problem. The SFM problem has wide applications in computer vision, economics, and game theory and is well-studied in literature (Lee et al. 2015, Axelrod et al. 2020, Zhang et al. 2020). In contrast, the stochastic SFM problem is less understood, and Ito (2019) gives the only result on the stochastic SFM problem, in which they provide upper and lower bounds for finding solutions with a small error bound in expectation. In Murota (2003), a generalization of submodular functions, called the $L^\natural$-convex functions, are defined through the translation submodularity. The $L^\natural$-convex functions are equivalent to functions that are both submodular and integrally convex on the integer lattice. In addition, the $L^\natural$-convex function has a convex extension that shares similar properties as the Lovász extension, and therefore, gradient-based methods are also applicable for $L^\natural$-convex function minimization.

1.3. Notation

For $N \in \mathbb{N}$, we define $[N] := \{1, 2, \ldots, N\}$. For a given set $\mathcal{S}$ and an integer $d \in \mathbb{N}$, the product set $\mathcal{S}^d$ is defined as $\{(x_1, x_2 \ldots, x_d) : x_i \in \mathcal{S}, i \in [d]\}$ in which $[d] = \{1, 2, \ldots, d\}$. For example, if $\mathcal{S} = [N]$, then $\mathcal{S}^d = \{(x_1, x_2 \ldots, x_d) : x_i \in [N], i \in [d]\}$. For two vectors $x, y \in \mathbb{R}^d$, we use $(x \wedge y)_i := \min\{x_i, y_i\}$ and $(x \vee y)_i := \max\{x_i, y_i\}$ to denote the component-wise minimum and maximum. Similarly, the ceiling function $\lceil \cdot \rceil$ and the flooring function $\lfloor \cdot \rfloor$ round each component to an integer when applied to vectors. We denote ξ_x as the random object associated with the stochastic system labeled by the choice of decision variables x. The failing probability of simulation–optimization algorithms is denoted as δ. The notation $f = O(g)$ (respectively, $f = \Theta(g)$) means that there exist absolute constants $c_1, c_2 > 0$ such that $f \le c_1 g$ ($c_1 g \le f \le c_2 g$). Similarly, the notation $f = \tilde{O}(g)$ ($f = \tilde{\Theta}(g)$) means that there exist absolute constants $c_1, c_2 > 0$ and constant $c_3 > 0$ independent of δ such that $f \le c_1 g + c_3$ ($c_1 g \le f \le c_2 g + c_3$).

2. Model and Framework

The model in consideration contains a complex stochastic system whose performance depends on discrete decision variables that belong to a discrete feasible set $\mathcal{X} \subset \mathbb{Z}^d$. From a modeling perspective, in a stochastic system, the system performance may depend on three elements: the decision variable $x \in \mathbb{Z}^d$, a random object ξ_x supported on a proper space $(\Upsilon, \mathcal{B}_\Upsilon)$ that summarizes all the associated random quantities and processes involved in the system when the decision x is taken, and a deterministic function $F : \mathcal{X} \times \Upsilon \to \mathbb{R}$ that takes the value of decision variables and a realization of the randomness as inputs and outputs the associated system performance. Specifically, the deterministic function F captures the full operations logic of the stochastic system, which can be complicated. The objective function with decision variable x is given by

$$f(x) := \mathbb{E}[F(x, \xi_x)].$$

We consider scenarios when $f(x)$ does not adopt a closed-form representation and can only be evaluated by averaging over simulation samples of $F(x, \xi_x)$. More specifically, we write $\xi_{x,1}, \xi_{x,2}, \ldots, \xi_{x,n}$ as independent and identically distributed copies of ξ_x. We use $\hat{F}_n(x) := \frac{1}{n}\sum_{j=1}^n F(x, \xi_{x,j})$ to denote the empirical mean of the n independent evaluations for the choice of decision variables x. The selection of the optimal choice of decision variables is through the selection of a choice of decision variable x that renders the best objective value $f(x)$. Denote x^* as any choice of decision variable that renders the optimal objective value, such that

$$f(x^*) = \min_{x \in \mathcal{X}} f(x). \tag{1}$$

Note that we fix the use of minimum operation to represent the optimal. Our general goal is to develop simulation–optimization algorithms that select a good choice of decision variable x, such that

$$f(x) - f(x^*) \le \epsilon,$$

where $\epsilon > 0$ is the given user-specified precision level. In this paper, we consider this selection problem in a large decision space with high dimension.

Because f does not have a closed-form representation and has to be evaluated by simulation, we take the view that no further structure information is

available in addition to the convex structure. For instance, for a real-world model, f may have a very flat landscape around the minimum, which may not be known a priori. In this case, there may be a number of choices of decision variables that render objective value that is at most ϵ apart from the optimal. This also motivates our goal to select a good choice of decision variables instead of the best because too much computational resource may be needed to identify exactly the best when the landscape around the minimum is flat. Therefore, our general goal is to develop simulation–optimization algorithms that are expected to robustly work for any convex model without knowing further specific structure.

Because the precision level ϵ cannot be delivered almost surely with a finite computational budget for simulation, we consider a selection optimality guarantee called the probability of good selection; see Eckman and Henderson (2021, 2018) and Hong et al. (2021).

- PGS: With probability at least $1-\delta$, the solution x returned by an algorithm has objective value at most ϵ larger than the optimal objective value.

This PGS guarantee is also called the PAC guarantee in the literature (Even-Dar et al. 2002, Kaufmann et al. 2016, Ma and Henderson 2017). Whereas our focus is to design algorithms that satisfy the PGS optimality guarantee, we also consider the optimality guarantee of the probability of correct selection with indifference zone as a comparison.

- PCS-IZ: 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, with probability at least $1-\delta$, the solution x returned by an algorithm is the unique optimal solution.

By choosing $\epsilon < c$, algorithms satisfying the PGS guarantee can be directly applied to satisfy the PCS-IZ guarantee. On the other hand, counterexamples in Eckman and Henderson (2021) show that algorithms satisfying the PCS-IZ guarantee may fail to satisfy the PGS guarantee. This phenomenon is further explained from the hypothesis-testing perspective in Hong et al. (2021). The failing probability δ in either PGS or PCS-IZ is typically chosen to be very small to ensure a high-probability result. Hence, we assume in the following that δ is small enough and focus on the asymptotic expected simulation cost.

To facilitate the construction of simulation–optimization algorithms that can deliver the PGS guarantee for general convex problems, we specify the composition of simulation–optimization algorithms in the next section. In addition, we assume that the probability distribution for the simulation output $F(x,\xi_x)$ is sub-Gaussian.

Assumption 1. *The distribution of $F(x,\xi_x)$ is sub-Gaussian with known parameter σ^2 for any $x \in \mathcal{X}$.*

The sub-Gaussian distributional assumption part in Assumption 1 is standard in the simulation–optimization literature; see, for example, the discussions in Zhong and Hong (2018). One special case is that the probability distribution for the simulation output at a choice of decision variables x is Gaussian with variance σ_x^2. However, it is indeed possible that these variances for different x's are unknown in advance, therefore posing a challenge. In that regard, one may consider using the system structure to provide a generic upper bound $\sigma^2 \geq \max_{x\in\mathcal{X}}\sigma_x^2$, particularly when the maximum possible level of uncertainties associated with a system is available. In practice, if the decision maker knows in advance what specific extreme choices of decision variables lead to the highest achievable variance of the system, that would be significantly valuable to find the upper bound. In general, when the variances are not known in advance, such a generic upper bound can sometimes be loose and, therefore, is conservative. In this work, we take the view that an upper bound (maybe a loose one) is known in advance and focus on the algorithm design to search for a good solution that has light dependence on the dimension. Note that our analysis under Assumption 1 can be naturally extended to models whose randomness distribution satisfies certain concentration inequalities. For example, when the randomness is subexponential (which may have heavier tails than Gaussian), one can apply the Hoeffding–Azuma inequality for subexponential tailed martingales to achieve provably efficient algorithms.

2.1. Simulation–Optimization Algorithms

In this section, we define different classes of simulation–optimization algorithms. We hope to design simulation–optimization algorithms that can deliver a certain optimality guarantee, say, PGS, for any convex model without knowing further structure. A broad range of sequential simulation–optimization algorithms consists of three parts.

- The sampling rule determines which choice of decision variables to simulate next based on the history of simulation observations up to the current time.
- The stopping rule controls the end of the simulation phase and is a stopping time according to the filtration up to the current time. We assume that the stopping time is finite almost surely.
- The recommendation rule selects the choice of decision variables that satisfies the optimality guarantee based on the history of simulation observations.

The model of Problem (1) consists of the decision set $\mathcal{X}$, the space of randomness $(\Upsilon,\mathcal{B}_\Upsilon)$, and the function $F(\cdot,\cdot)$.

Next, we define the class of simulation–optimization algorithms that can deliver solutions satisfying a 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–optimization 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 $\mathcal{O}$.

We define the set of all models such that the objective function $f(\cdot)$ is convex (defined in the next section) on the discrete set $\mathcal{X}$ as $\mathcal{MC}(\mathcal{X})$ or, simply, $\mathcal{MC}$. Using this definition, a $(\text{PGS},\mathcal{MC})$-algorithm is one that guarantees the finding of a solution that satisfies the PGS guarantee for any convex model without knowing further structure.

2.2. Simulation Costs

In the development of simulation–optimization algorithms that satisfy a certain optimality guarantee, especially for large-scale problems, the performance of different algorithms can be compared based on their computational costs to achieve the same optimality guarantee. We take the view that the simulation cost of generating replications of $F(x,\xi_x)$ is the dominant contributor to the computational cost associated with a simulation–optimization algorithm. See also Luo et al. (2015), Ni et al. (2017), and Ma and Henderson (2019). Therefore, we quantify the computational cost as the total number of evaluations of $F(x,\xi_x)$ for all $x \in \mathcal{X}$. In some simulation problems, but not all, we may also have access to noisy and possibly biased estimates of $f(\cdot)$ near point x along with an evaluation of $F(x,\xi_x)$. The simulation cost in this case is discussed in Section 6. For all simulation–optimization algorithms proposed in this paper, we provide upper bounds on the expected simulation cost to achieve a certain optimality guarantee. Note that these upper bounds do not rely on the specific structure of the problem in addition to convexity. The expected simulation cost serves as a measurement to compare different algorithms and provide insights on how the computational cost depends on the scale and dimension of the problem.

Now, we define 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 $\mathbf{A}$ is a simulation–optimization algorithm and τ is the stopping time of the algorithm $\mathbf{A}$, which is also the number of simulation evaluations of $F(\cdot,\cdot)$.

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

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

Note that the (ϵ,δ) *-PGS* refers to the PGS optimality guarantee with user-specified precision level $\epsilon > 0$ and confidence level $1-\delta$. The notion (c,δ)*-PCS-IZ* refers to the PCS-IZ optimality guarantee with confidence level $1-\delta$ and IZ parameter c. The class of models $\mathcal{MC}$ include all convex models, whereas $\mathcal{MC}_c$ include all convex models with IZ parameter c. In addition, we mention that all upper bounds derived in this paper are actually almost sure bounds of the simulation cost, whereas lower bounds only hold in expectation.

2.3. Discrete Convex Functions in Multidimensional Space

In contrast to the continuous case, the discrete convexity has various definitions, for example, convex extensible functions and submodular functions. Although these concepts coincide for the one-dimensional case, they have essential differences in the multidimensional case. In this work, we consider $L^\natural$-convex functions (Murota 2003), which are defined by the midpoint convexity (defined later in this section) for discrete variables. Considerably many discrete optimization via simulation problems have a $L^\natural$-convex structure. For example, the expected customer waiting time in a multiserver queueing network is proved to be a separated convex function (Wolff and Wang 2002, Altman et al. 2003) and, therefore, is $L^\natural$-convex. In addition, the dissatisfaction function of a bike-sharing system is shown to be multimodular in Freund et al. (2017), which is $L^\natural$-convex under a linear transformation. More examples of $L^\natural$-convex functions are given in Murota (2003). On the other hand, the minimization of an $L^\natural$-convex function is equivalent to the minimization of its linear interpolation, which is continuous and convex. Combined with the closed-form subgradient, $L^\natural$-convex functions provide a good framework for studying discrete convex simulation optimization problems.

Before we give the definition of $L^\natural$-convexity, we first show that it is not suitable to define discrete convex functions just as functions that have a convex extension. The main problem of this definition based on extension is that the "local optimality" may not be equivalent to the global optimality, which is one of the important properties used in convex optimization. In the discrete case, we say a point $\bar{x}$ is a *local minimum* of $f(\cdot)$ if $f(\bar{x}) \le f(x)$ for all feasible x such that $\|x-\bar{x}\|_\infty \le 1$. Without this property, algorithms may get stuck at spurious local minima and fail to satisfy the optimality guarantee. We give an example to illustrate the failure.

Example 1. We consider the case when $n = 4$ and $d = 2$. The objective function is given as

$$f(x, y) := 4|2x + y - 8| + |x - 2y + 6|.$$

The function $f(x, y)$ is a convex function on the set $[1, 4]^2$, and the unique global minimizer is (2, 4). When restricted to the integer lattice $\{1, 2, 3, 4\}^2$, the global minimizer is still (2, 4). We consider the point (3, 2) with objective value $f(3, 2) = 5$. In the local neighborhood $\{2, 3, 4\} \times \{1, 2, 3\}$, which contains points that have ℓ_∞-distance at most 1 from (3, 2), the objective values are

$$f(2, 1) = 18,\ f(3, 1) = 11,\ f(4, 1) = 12,\ f(2, 2) = 12,$$
$$f(4, 2) = 14,\ f(2, 3) = 6,\ f(3, 3) = 7,\ f(4, 3) = 16.$$

Thus, the point (3, 2) is a spurious local minimizer of the discrete function. This shows that local optimality cannot imply global optimality.

On the other hand, the $L^\natural$-convexity ensures that local optimality implies global optimality. Similar to the continuous case, $L^\natural$-convex functions can be characterized by the midpoint convexity property.

Definition 3. A set $\mathcal{S} \subset \mathbb{Z}^d$ is called a $L^\natural$-convex set if it holds that

$$x, y \in \mathcal{S} \Rightarrow \lfloor (x + y)/2 \rfloor, \lceil (x + y)/2 \rceil \in \mathcal{S}.$$

A function $f(x) : \mathcal{X} \mapsto \mathbb{R}$ is called a $L^\natural$-convex function if $\mathcal{X}$ is a $L^\natural$-convex set and 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 $\mathcal{MC}(\mathcal{X})$ or, simply, $\mathcal{MC}$. The set of models such that $f(x)$ is $L^\natural$-convex with indifference zone parameter c is denoted as $\mathcal{MC}_c(\mathcal{X})$ or, simply, $\mathcal{MC}_c$.

We assume that the objective function is $L^\natural$-convex in the remainder of this work.

Assumption 2. *The objective function $f(x)$ is an $L^\natural$-convex function on the $L^\natural$-convex set $\mathcal{X}$.*

Before proceeding to the properties, we provide a few examples of $L^\natural$-convex sets and $L^\natural$-convex functions.

Example 2. Examples of $L^\natural$-convex sets include the whole space $\mathbb{Z}^d$ and the hypercube $[N_1] \times [N_2] \times \ldots \times [N_d]$, where d and N_i are positive integers for all $i \in [d]$. Another important example of $L^\natural$-convex sets is the linearly transformed capacity-constrained hypercube; see the derivation in Section 7. Specifically, for positive integers d, N, and $M \leq N$, the following set is $L^\natural$-convex:

$$\{x \in \mathbb{Z}^d \mid x_1 \in [N],\ x_{i+1} - x_i \in [N],\ \ \forall i \in [d-1],\ x_d \leq M\}.$$

Examples of $L^\natural$-convex functions include the indicator function of any $L^\natural$-convex set, linear functions, and separably convex functions, namely, functions having the form

$$f(x) = \sum_{i=1}^{d} f^i(x_i),$$

where $f^i(\cdot)$ is a convex function for all $i \in [d]$. See Murota (2003) for more examples.

In the following lemma, we list several properties of $L^\natural$-convex functions.

Lemma 1. *Suppose that the function $f(x) : \mathcal{X} \mapsto \mathbb{R}$ is $L^\natural$-convex. The following properties hold:*

- *There exists a convex function $\tilde{f}(x)$ on the convex hull $\mathrm{conv}(\mathcal{X})$ such that $\tilde{f}(x) = f(x)$ for all $x \in \mathcal{X}$.*
- *Local optimality is equivalent to global optimality:*

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

- *Translation submodularity holds:*

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

The $L^\natural$-convexity can be viewed as a combination of submodularity and integral convexity (Murota 2003, theorem 7.20). Intuitively, the submodularity ensures the existence of a piecewise linear convex interpolation in the local neighborhood of each point, whereas the integral convexity ensures that the piecewise linear convex interpolations can be pieced together to form a convex function on $[1, N]^d$. In addition, we can calculate a subgradient of the convex extension with $O(d)$ function value evaluations. Hence, $L^\natural$-convex functions provide a good framework for extending continuous convex optimization theory to the discrete case.

3. Simulation–Optimization Algorithms and Expected Simulation Costs for a Special Case

In this and the following section, we propose simulation–optimization algorithms that achieve the PGS guarantee for any simulation optimization problem with a $L^\natural$-convex objective function. We prove upper bounds on the expected simulation costs. To better present the dependence of expected simulation costs on the scale and dimension of the problem, we assume that the feasible set is the hypercube $[N]^d$ in complexity analysis.

Assumption 3. *The feasible set of decision variables is $\mathcal{X} = [N]^d$, where $N \geq 2$ and $d \geq 1$.*

In large-scale simulation problems, either N or d or both N and d can be large. We note that, if the feasible set $\mathcal{X}$ is a general $L^\natural$-convex set, the construction of the

convex extension and the analysis are still valid by replacing N with $\max_{x,x' \in \mathcal{X}} \|x - x'\|_\infty$. Moreover, our algorithms are directly applicable to the case in which $\mathcal{X}$ is a general $L^\natural$-convex set, which is also the minimal requirement on the feasible set for the definition of $L^\natural$-convexity. In this section, we start with a special case in which the decision space is $\{0,1\}^d$ for a large d. We defer the discussions for general N to Section 4. The simulator may have a general complex and discontinuous structure that no unbiased gradient estimator is available within the replication of simulation. For scenarios in which a single replication of simulation can also generate gradient information at very low costs, we propose and analyze simulation–optimization algorithms in Section 6.

The general idea of designing simulation–optimization algorithms in the multidimensional case is to construct subgradients of the convex extension with $O(d)$ function value evaluations on the neighboring choices of a decision. Hence, the stochastic subgradient descent (SSGD) method can be used to solve Problem (1). Compared with the bisection method and general cutting-plane methods, gradient-based methods have two advantages in our case. First, as pseudo-polynomial algorithms, gradient-based methods usually have lighter dependence on the problem dimension d compared with strongly or weakly polynomial algorithms. For example, the deterministic integer-valued submodular function minimization (SFM) problem can be solved with $\tilde{O}(d), \tilde{O}(d^2), \tilde{O}(d^3)$ function value evaluations using pseudo-polynomial (Axelrod et al. 2020), weakly polynomial and strongly polynomial (Lee et al. 2015) algorithms, respectively. Usually, gradient-based methods have extra polynomial dependence on the Lipschitz constant of the objective function, in exchange for the reduced dependence on d. However, for a large group of problems, the Lipschitz constant may be estimated a priori. Moreover, we can design algorithms whose expected simulation cost does not critically rely on the Lipschitz constant, in the sense that the Lipschitz constant only appears in a smaller order term in the expected simulation cost. Hence, gradient-based methods are preferred for high-dimensional problems. On the other hand, ordinary cutting plane methods are not robust to noise and problem-specific stabilization techniques should be designed for stochastic problems (Sen and Higle 2001), or complicated robust scheme should be constructed (Nemirovskij and Yudin 1983, Agarwal et al. 2011). Considering these two advantages of gradient-based methods, we focus on the SSGD method in designing our simulation-optimization algorithms and make the assumption that an upper bound of the ℓ_∞-Lipschitz constant is known a priori.

Assumption 4. *An upper bound on the ℓ_∞-Lipschitz constant of $f(x)$ is known to be L a priori. Namely, we know beforehand that*

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

We remark that this constant L, in the general decision-making contexts, reflects the impact on the objective function by a small change in the value of the high-dimensional decision variable. For example, in bike-sharing applications, this L may reflect the impact of allocating one more bike to a station. Whether the objective function being revenue or number of dissatisfied customers, the upper bound on the impact of allocating one more bike can be quantified. The estimation of L usually relies on the domain knowledge about the problem. For example, the user dissatisfaction function in the bike-sharing application takes values in $\{0,1,\ldots,M\}$, where M is the expected number of users each day. Then, an estimate of the Lipschitz constant is $L \le M$.

When the decision space is $\mathcal{X} = \{0,1\}^d$, $L^\natural$-convex functions are equivalent to submodular functions, and therefore, Problem (1) is equivalent to the stochastic submodular function minimization (stochastic SFM) problem. To prepare the design of simulation algorithms, we first define the Lovász extension of submodular functions and give an explicit subgradient of the Lovász extension at each point.

Definition 4. Suppose that function $f(x) : \{0,1\}^d \mapsto \mathbb{R}$ is a submodular function, that is, it holds that

$$f(x) + f(y) \ge f(x \wedge y) + f(x \vee y), \quad \forall x, y \in \{0,1\}^d.$$

For any $x \in [0,1]^d$, we say a permutation $\alpha_x : [d] \mapsto [d]$ is a consistent permutation of x if

$$x_{\alpha_x(1)} \ge x_{\alpha_x(2)} \ge \ldots \ge x_{\alpha_x(d)}.$$

We define $S^{x,0} := (0,\ldots,0)$. For each $i \in \{1,\ldots,d\}$, the ith neighboring point 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 kth unit vector of $\mathbb{R}^d$. We define the Lovász extension $\tilde{f}(x) : [0,1]^d \mapsto \mathbb{R}$ as

$$\tilde{f}(x) := f(S^{x,0}) + \sum_{i=1}^{d} [f(S^{x,i}) - f(S^{x,i-1})] x_{\alpha_x(i)}. \tag{2}$$

We note that the value of the Lovász extension does not rely on the consistent permutation we choose. A numerical illustration of the Lovász extension is provided in the online appendix. We list several well-known properties of the Lovász extension and refer their proofs to Lovász (1983) and Fujishige (2005). We note that the subdifferential at point $x \in [0,1]^d$ is defined as the set

$$\partial \tilde{f}(x) = \{g \in \mathbb{R}^d : \langle g, x - y \rangle \ge \tilde{f}(x) - \tilde{f}(y), \ \ \forall y \in [0,1]^d\}.$$

Lemma 2. *Suppose that Assumptions 1–4 hold. Then, the following properties of $\tilde{f}(x)$ hold.*

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

ii. *The minimizers of $\tilde{f}(x)$ satisfy* $\arg\min_{x\in[0,1]^d} \tilde{f}(x) = \arg\min_{x\in\{0,1\}^d} f(x)$.

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*

$$g_{\alpha_x(i)} := f(S^{x,i}) - f(S^{x,i-1}), \quad \forall i \in [d]. \tag{3}$$

v. *Subgradients of $\tilde{f}(x)$ satisfy*

$$\|g\|_1 \le 3L/2, \quad \forall g \in \partial\tilde{f}(x),\ x \in [0,1]^d.$$

To apply the SSGD method to design simulation–optimization algorithms for Problem (1), we need to resolve the following two questions:

- How to design an unbiased subgradient estimator.
- How to round an approximate solution in $[0,1]^d$ to an approximate solution in $\mathcal{X} = \{0,1\}^d$.

For the first question, we consider the subgradient estimator at point x as

$$\hat{g}_{\alpha_x(i)} := F(S^{x,i}, \xi_i^1) - F(S^{x,i-1}, \xi_{i-1}^2), \quad \forall i \in [d], \tag{4}$$

where ξ_i^j are mutually independent for $i \in [d]$ and $j \in [2]$. By definition, we know the components of $\hat{g}$ are mutually independent and the simulation cost of each $\hat{g}$ is $2d$. Using the subgradient defined in (3), we have

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

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

Next, we consider the second question. We define the relaxed problem as

$$f^* := \min_{x\in[0,1]^d} \tilde{f}(x). \tag{5}$$

Properties (i) and (ii) of Lemma 2 imply that the original Problem (1) is equivalent to the relaxed Problem (5). In the deterministic case, suppose we already have an ϵ-optimal solution to Problem (5), that is, a point $\bar{x}$ in $[0,1]^d$ such that $\tilde{f}(\bar{x}) \le f^* + \epsilon$. Then, we rewrite the Lovász extension in (2) as

$$\begin{aligned}\tilde{f}(\bar{x}) = [1 - \bar{x}_{\alpha_{\bar{x}}(1)}] f(S^{\bar{x},0}) + \sum_{i=1}^{d-1} [\bar{x}_{\alpha_{\bar{x}}(i)} - \bar{x}_{\alpha_{\bar{x}}(i+1)}] f(S^{\bar{x},i}) \\ + \bar{x}_{\alpha_{\bar{x}}(d)} f(S^{\bar{x},d}),\end{aligned} \tag{6}$$

which is a convex combination of $f(S^{\bar{x},0}), \ldots, f(S^{\bar{x},d})$. Hence, there exists an ϵ-optimal solution among the neighboring points of $\bar{x}$. This means that we can solve a subproblem with $d + 1$ points to get the ϵ-optimal solution among neighboring points. For the stochastic case, a similar rounding process can be designed, and we give the pseudo-code in Algorithm 1. The rounding process for the (c,δ)-PCS-IZ guarantee follows by choosing $\epsilon = c/2$.

Algorithm 1 (Rounding Process to a Feasible Solution)

Input: Model $\mathcal{X}, \mathcal{B}_Y, F(x,\xi_x)$; optimality guarantee parameters ϵ, δ, $(\epsilon/2, \delta/2)$-PGS solution $\bar{x}$ to Problem (5).

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

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

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

3: Simulate at S^i until the $1 - \delta/(4d)$ confidence half-width of $\hat{F}_n(S^i)$ is smaller than $\epsilon/4$ for all i.

4: Return the optimal point $x^* \leftarrow \arg\min_{S\in\{S^0,\ldots,S^d\}} \hat{F}_n(S)$.

The following theorem proves the correctness and estimates the simulation cost of Algorithm 1. Note that all the upper bound results on simulation costs in this paper are proved to hold both almost surely and in expectation. We do not differentiate the use of *simulation costs* and *expected simulation costs* in upper bound results.

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

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

Proof of Theorem 1. The proof of Theorem 1 is given in the supplementary material. □

We note that the simulation cost in the $\tilde{O}$ notation gives the asymptotic simulation cost when δ is small enough. After resolving these two problems, we can first use the SSGD method to find an approximate solution to Problem (5) and then round the solution to get an approximate solution to Problem (1). Hence, the focus of the remainder of this section is to provide upper bounds of simulation cost to the SSGD method. The main difficulty of giving sharp upper bounds lies in the fact that the Lovász extension is neither smooth nor strongly convex. This property of the Lovász extension prohibits the application of Nesterov acceleration and common variance reduction techniques.

Now, we propose the projected and truncated SSGD method for the (ϵ,δ)-PGS guarantee. The orthogonal projection onto the convex hull $\mathrm{conv}(\mathcal{X})$, which is defined as

$$\mathcal{P}_{\mathcal{X}}(x) := \arg\min_{y\in\mathrm{conv}(\mathcal{X})} \|y - x\|_2, \quad \forall x \in \mathbb{R}^d,$$

is applied after each iteration to ensure the feasibility of iteration point. Because the convex hull is a convex set, the projection is well-defined. For the case in which the feasible set is $\{0,1\}^d$, the projection

is given by

$$\mathcal{P}_{\mathcal{X}}(x) := (x \wedge \mathbf{1}) \vee \mathbf{0}, \quad \forall x \in \mathbb{R}^d.$$

In addition to the projection, componentwise truncation of stochastic subgradient is critical in reducing expected simulation costs. The truncation operator with threshold $M > 0$ is defined as

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

The pseudo-code of the projected and truncated SSGD method is listed in Algorithm 2.

Algorithm 2 (Projected and Truncated SSGD Method for the PGS Guarantee)

Input: Model $\mathcal{X}, \mathcal{B}_Y, F(x, \xi_x)$; optimality guarantee parameters ϵ, δ; number of iterations T; step size η; truncation threshold M.

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

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

The analysis of Algorithm 2 fits into the classical convex optimization framework. With a suitable choice of step size, truncation threshold, and number of iterations, Algorithm 2 returns an (ϵ, δ)-PGS solution, and the expected simulation cost has $O(d^2)$ dependence on the dimension.

Theorem 2. *Suppose that Assumptions 1–4 hold and the subgradient estimator in (4) is used. If we choose*

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

then Algorithm 2 returns a (ϵ, δ)-PGS solution. Furthermore, we have

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

Proof of Theorem 2. The proof of Theorem 2 is given in the supplementary material. □

Although independent of δ, we note that the last two terms in the expected simulation cost may be comparable to the first term when δ is not that small. We can prove that, without the truncation step (i.e., $M = \infty$), the expected simulation becomes

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

Hence, the truncation of the stochastic subgradient is necessary for reducing the asymptotic expected simulation cost. In addition, we note that the Lipschitz constant L is required in determining the truncation threshold M; see Lemma EC.3 for more details. Whereas the error of the normal SSGD method only contains the optimization residual and the variance terms, the residual of the truncated SSGD method has an extra bias term. We note that the bias term can be made arbitrarily small with high probability by choosing large enough M and utilizing the tail bound for sub-Gaussian random variables, and therefore, the total error can be controlled similarly as with the normal SSGD method. By choosing $\epsilon = c/2$, Algorithm 2 returns a (c, δ)-PCS-IZ solution, and the expected simulation cost for the PCS-IZ guarantee is

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

We note that the expected simulation cost for both guarantees does not critically depend on the Lipschitz constant L. As an alternative to Estimator (4), one may consider generating a stochastic subgradient by randomly choosing a subset of components and only estimating the chosen components of subgradients. However, using this estimator, we cannot achieve better simulation cost, and the expected simulation cost may be critically dependent on L.

Before finishing the discussion of the stochastic SFM problem, we note that the expected simulation cost in Theorem 2 may be improved if we further assume the stochastic subgradient is bounded almost surely. We provide a detailed analysis in the online appendix.

4. Simulation–Optimization Algorithms and Expected Simulation Costs for the General Case

In this section, we extend to the general $L^\natural$-convex function minimization problem with decision space $[N]^d$ for general large N and d. We design simulation–optimization algorithms that achieve the PGS guarantee and prove upper bounds on the simulation costs.

As an extension to the methodology in Section 3, we first show that the Lovász extension in the neighborhood of each point can be pieced together to form a convex function on $\mathrm{conv}(\mathcal{X}) = [1, N]^d$. We define the local neighborhood of each point $y \in [N-1]^d$ as the hypercube

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

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

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

We denote the objective function $f(x)$ restricted to $\mathcal{C}_y \cap \mathcal{X}$ as $f_y(x)$. 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 ith neighboring point of x is defined as

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

By the translation submodularity property of $L^\natural$-convex functions, we know function $f_y(x)$ is a submodular function on $y + \{0,1\}^d$, and its Lovász extension in $\mathcal{C}_y$ can be calculated as

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

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

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

The next theorem verifies the well definedness and the convexity of $\tilde{f}$.

Theorem 3. *The function $\tilde{f}(x)$ in (7) is well-defined and is convex on $\mathcal{X}$.*

Proof of Theorem 3. The proof of Theorem 3 is given in the supplementary material. □

A numerical verification of the results of Theorem 3 is provided in the online appendix. Properties of the Lovász extension in Lemma 2 can be naturally extended to the convex extension $\tilde{f}(x)$.

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

- *For any $x \in \mathcal{X}$, it holds that $\tilde{f}(x) = f(x)$.*
- *The minimizers of $\tilde{f}$ satisfy $\arg\min_{y \in [1,N]^d} \tilde{f}(y) = \arg\min_{y \in [N]^d} f(y)$.*
- *For a point $x \in \mathcal{C}_y$, a subgradient $g \in \partial \tilde{f}(x)$ is given by*

$$g_{\alpha_x(i)} := f(S^{x,i}) - f(S^{x,i-1}), \quad \forall i \in [d]. \tag{8}$$

- *Subgradients of function $\tilde{f}(x)$ satisfy*

$$\|g\|_1 \le 3L/2, \quad \forall g \in \partial \tilde{f}(x),\ x \in \mathcal{X}.$$

Similar to the proof of Theorem 3, the subgradient given in (8) does not depend on the hypercube and the consistent permutation we choose. The subgradient estimator defined in (4) is still valid in the general case. Thus, changing the orthogonal projection to be

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

Algorithm 2 can be applied to the general case, and we get the counterpart to Theorem 2.

Theorem 4. *Suppose that Assumptions 1–4 hold and the subgradient estimator in (4) is used. If we choose*

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

then Algorithm 2 returns an (ϵ,δ)-PGS solution. Furthermore, we have

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

Proof of Theorem 4. The proof of Theorem 4 is given in the supplementary material. □

We reiterate that the results also apply to the general $L^\natural$-convex set case by replacing the scale N with $\max_{x,x' \in \mathcal{X}} \|x - x'\|_\infty$. Similarly, the expected simulation costs in Theorem 4 can be improved under the bounded stochastic subgradient assumption, and we defer the discussion to the online appendix. For the PCS-IZ guarantee, we can choose $\epsilon = c/2$, and Algorithm 2 returns a (c,δ)-PCS-IZ solution. Hence, these asymptotic simulation costs also hold for the PCS-IZ guarantee. However, with the a priori knowledge about the indifference zone parameter, we can design an acceleration scheme similar to Xu et al. (2016), which is based on the weak sharp minimum condition. The acceleration scheme reduces the dependence on N from $O(N^2)$ to $O(\log(N))$, and we provide details in the online appendix.

5. Lower Bound on Expected Simulation Cost

We derive lower bounds on the expected simulation cost for any simulation–optimization algorithm that can achieve the PGS guarantee. In this section, we prove that the expected simulation cost is lower bounded by $O(d\epsilon^{-2}\log(1/\delta))$. We acknowledge that the lower bound may not be tight, but the proven lower bound results suggest the limits for all simulation–optimization algorithms to achieve the PGS guarantee for general simulation optimization problems with convex structure.

To prove lower bounds, basically, we construct several convex models that are "similar" to each other, but they have distinct optimal solutions, in which the difference between two models is characterized by the

Kullback–Leibler (KL) divergence between their distributions. Hence, any simulation–optimization algorithms need a large number of simulation runs to differentiate these models. More rigorously, the information-theoretical inequality in Kaufmann et al. (2016) provides a systematic way to prove lower bounds of zeroth order algorithms. 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

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

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

Lemma 4. *For any two models M_1, M_2 and any event $\mathcal{E}\in\mathcal{F}_\tau$, we have*

$$\sum_{x\in\mathcal{X}} \mathbb{E}_{M_1}[N_x(\tau)]\mathrm{KL}(\nu_{1,x},\nu_{2,x}) \geq d(\mathbb{P}_{M_1}(\mathcal{E}),\mathbb{P}_{M_2}(\mathcal{E})), \quad (9)$$

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 $k = 1, 2$.

The information-theoretical Inequality (9) is our major tool for deriving lower bounds. We first reduce the construction of $L^\natural$-convex functions to the construction of submodular functions. Then, using the family of submodular functions defined in Graur et al. (2020), we can construct $d+1$ submodular functions that have different optimal solutions and have the same value except on $d+1$ potential solutions. Hence, the algorithm has to simulate enough samples on the $d+1$ potential solutions to decide the optimal solution, and the simulation cost is proportional to d.

Theorem 5. *Suppose that Assumptions 1–3 hold. We have*

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

Proof of Theorem 5. The proof of Theorem 5 is given in the supplementary material. □

We note that this lower bound is also true when Assumption 4 holds with $L \geq \epsilon/N$. In addition, a similar construction to Theorem 5 leads to a lower bound on the expected simulation cost for the PCS-IZ guarantee.

Theorem 6. *Suppose that Assumptions 1–3 hold. We have*

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

Proof of Theorem 6. The proof of Theorem 6 is given in the supplementary material. □

6. Simulation–Optimization Algorithms with Biased Gradient Information

In large-scale discrete optimization via simulation, during a simulation run for performance evaluation at a given value of the d-dimensional decision variable x, it is sometimes possible that the neighboring values of decision variables (those very close to x) can be evaluated simultaneously within the same simulation run for x at marginal costs. See Jian et al. (2016) and Jian (2017) for a bike-sharing discrete optimization via simulation problem that adopts this feature. When the decision variable x is in continuous space, this simultaneous simulation approach is called the *infinitesimal perturbation analysis* or *forward/backward automatic differentiation*, in which a gradient estimator at x can be obtained within the same simulation run for evaluation of x. In the continuous decision space, such gradient estimators can be unbiased under Lipschitz continuity regularity conditions though no general guarantees on unbiasedness exist when continuity fails. In contrast, for discrete optimization via simulation problems, in particular for those for which discrete decision variables do not easily relax to continuous variables, the difference of function value on x and function value on the neighboring points of x can be viewed as an approximate directional derivative. This approximate gradient information (i.e., the difference of objective function values) is very difficult, if not impossible, to estimate without bias using only a single simulation run. In general, the system dynamics and logic are different for two different discrete decision variables even when they differ in only one coordinate. Therefore, in the simulation run for some choice of the decision variable x, the simultaneous evaluation for neighboring choices of the decision variable may incur a bias. See chapter 4 of Jian (2017) for a detailed discussion on bike-sharing optimization as an example. Despite the bias, the availability of such gradient information can potentially be beneficial when d is large because only one simulation run is needed to evaluate a biased version of a d-dimension gradient estimator. The gradient estimator can be usually obtained at a marginal cost that does not depend on the dimension d, which is much lower than the cost of constructing a finite difference gradient estimator.

In this section, we provide simulation–optimization algorithms to achieve the PGS guarantee for discrete

convex simulation optimization problems when the gradient information is available (but possibly biased) within a simulation run at a cost that does not depend on dimension. We call this class of simulation–optimization algorithms, which utilize the available gradient information, first order algorithms. We show how the use of the gradient information reduces the expected simulation cost and how the bias existing in the gradient information affects the results. We first rigorously define the gradient information that can be obtained in simulation with different choices of decision variables. The gradient information that can be obtained within one simulation run is generally biased and has correlated components. The existence of correlation may increase the difficulty of analyzing the performance of simulation–optimization algorithms. Moreover, the correlation could contribute to a larger overall variance of the norm of the subgradient estimator, which may adversely affect the simulation–optimization algorithm.

On the bias side, if the bias in the subgradient estimator can be arbitrarily large, the sign of a subgradient estimator can even be flipped (see an example in Eckman and Henderson 2020). In those cases, there is, in general, no guarantee for gradient-based algorithms even for convex problems. Examples in Ajalloeian and Stich (2020) also show that the biased gradient-based methods may not converge to the optimum or even dramatically diverge. To circumvent this challenge, some existing works on biased gradient-based methods require the objective function to be smooth and have additional benign geometrical properties, for example, strong convexity or the Polyak–Łojasiewicz condition (Devolder et al. 2014, Chen and Luss 2018, Ajalloeian and Stich 2020, Hu et al. 2020). Because the convex extension of a general $L^\natural$-convex function is a piecewise linear function and is neither smooth nor strongly convex, these methods that require benign structure cannot be applied to our case.

In the special case when the biased subgradient estimator of $f(x)$ is the unbiased subgradient estimator of another function $h(x)$, we can view $h(x)$ as a perturbed version of $f(x)$. We define the Lovász extension of $h(x)$ in the same way and equivalently minimize the Lovász extension via the SSGD method. However, because function $h(x)$ may not be $L^\natural$-convex, its Lovász extension is a nonsmooth and nonconvex function, and there is no guarantee on the complexity of the SSGD method (Daniilidis and Drusvyatskiy 2020, Davis et al. 2020). In Zhang et al. (2020), the authors propose a stochastic normalized subgradient descent method with sample complexity $O(\epsilon^{-4})$ for finding a point with a subgradient with norm smaller than ϵ. Under the assumption of weak convexity, algorithms with sample complexity of $O(\epsilon^{-2})$ are proved in Davis and Drusvyatskiy (2019), Zhang and He (2018), and Mai and Johansson (2020). On the other hand, to achieve the same sample complexity as convex optimization, it is proved that the perturbation $h(x)-f(x)$ should have order $O(1/d)$ for all feasible x (Belloni et al. 2015, Jin et al. 2018, Mangoubi and Vishnoi 2018). However, the existence of the perturbed function $h(x)$ does not always hold, and therefore, we may not use these methods.

This discussion shows that some regularity assumptions on the bias are necessary for the applicability of gradient information to achieve the PGS guarantee. Now, we describe a formal definition of a biased subgradient estimator along with the assumption on bias. The key in the assumption is to regulate the relative magnitude of the bias so that, in expectation, the bias does not flip the sign of any components of the true subgradient at any choices of decision variables, that is, the magnitude of any component of the bias is bounded by the magnitude of this component of the true subgradient. The use of common random variables whenever available, in general, can contribute to the validity of this assumption. As a comparison, Eckman and Henderson (2020) regulate the norm of the bias to provide guarantees for continuous stochastic optimization problems. To prepare notation, the set of neighboring choices of decision variable $x \in \mathcal{X}$ is defined as

$$\mathcal{N}_x := \{x \pm e_{\mathcal{S}} : \mathcal{S} \subset [d]\} \cap \mathcal{X}.$$

Here, e_i is the ith unit vector of $\mathbb{R}^d$ and $e_{\mathcal{S}}$ is the indicator vector $\sum_{i\in\mathcal{S}} e_i$. The following assumption describes the case that allows the gradient information to have bias and correlation among different directions.

Assumption 5 (Subgradient Estimator with Bias and Correlation). *Given the bias ratio $a \in [0,1)$, for any point $x \in \mathcal{X}$, there exists a deterministic function $H_x(y,\eta_y): \mathcal{N}_x \times \mathrm{Z} \mapsto \mathbb{R}$ such that*

$$|\mathbb{E}[H_x(y,\eta_y)] - [f(y)-f(x)]| \le a \cdot |f(y)-f(x)|, \quad \forall y \in \mathcal{N}_x, \tag{10}$$

where $\mathcal{N}_x$ is the set of neighboring points of x and $(\mathrm{Z},\mathcal{B}_\mathrm{Z})$ is a proper space that summarizes the randomness of $G(x,\eta_x)$. Moreover, the marginal distribution for each $H_x(y,\eta_y)$ is sub-Gaussian with parameter $\tilde{\sigma}^2$, and the simulation cost of evaluating $H_x(y,\eta_y)$ for all $y \in \mathcal{N}_x$ is at most γ, multiplying the simulation cost of evaluating $F(x,\xi_x)$.

Under Assumption 5, $\mathbb{E}[H_x(y,\eta_y)]$ has the same sign as $f(y)-f(x)$ and, using theorem 7.14 in Murota (2003), point $x \in \mathcal{X}$ is a minimizer of $f(x)$ if and only if

$$\mathbb{E}[H_x(y,\eta_y)] \ge 0, \quad \forall y \in \mathcal{N}_x.$$

Therefore, it is still possible to check the global optimality by merely comparing the differences with neighboring points. A similar optimality condition can be established for the PGS guarantee. Using this observation, we give an

algorithm for the PGS guarantee using the biased subgradient estimator $H_x(y,\eta_y)$. The algorithm can be seen as a stochastic version of the steepest descent method in Murota (2003) and is listed in Algorithm 3.

Algorithm 3 (Adaptive Stochastic Steepest Descent Method for the PGS Guarantee)

Input: Model $\mathcal{X},\mathcal{B}_Y,F(x,\xi_x)$; optimality guarantee parameters ϵ,δ; biased subgradient estimator $H_x(y,\eta_y)$; bias ratio a.

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

1: Choose the initial point $x^{0,0} \leftarrow (N/2,\ldots,N/2)^T$.
2: Set the initial confidence half-width threshold $h_0 \leftarrow (1-a)L/12$.
3: Set maximal number of epochs $E \leftarrow \lceil \log_2(NL/\epsilon)\rceil$.
4: Set maximal number of iterations $T \leftarrow (1+a)/(1-a)\cdot 6N$.
5: **for** $e=0,1,\ldots,E-1$ **do**
6: **for** $t=0,1,\ldots,T-1$ **do**
7: **repeat** simulate $H_{x^{e,t}}(y,\eta_y)$ for all $y\in\mathcal{N}_{x^{e,t}}$
8: Compute the empirical mean $\hat{H}_{x^{e,t}}(y)$ using all simulated samples for all $y\in\mathcal{N}_{x^{e,t}}$.
9: Compute the $1-\delta/(ET)$ one-sided confidence interval

$$[\hat{H}_{x^{e,t}}(y)-h_y,\infty),\quad \forall y\in\mathcal{N}_{x^{e,t}}.$$

10: **until** the confidence half-width $h_y\le h_e$ for all $y\in\mathcal{N}_{x^{e,t}}$.
11: **if** $\hat{H}_{x^{e,t}}(y)\le -2h_e$ for some $y\in\mathcal{N}_{x^{e,t}}$ **then**
▷ This takes 2^{d+1} arithmetic operations.
12: Update $x^{e,t+1}\leftarrow y$.
13: **else if** $\hat{H}_{x^{e,t}}(y)>-2h_e$ for all $y\in\mathcal{N}_{x^{e,t}}$ **then**
14: **break**
15: **end if**
16: **end for**
17: Set $x^{e+1,0}\leftarrow x^{e,t}$ and $h_{e+1}\leftarrow h_e/2$.
18: **end for**
19: Return $x^{E,0}$.

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

Theorem 7. *Suppose that Assumptions 1–5 hold. Algorithm 3 returns an (ϵ,δ)-PGS solution, and we have*

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

Proof of Theorem 7. The proof of Theorem 7 is given in the supplementary material. □

We note that Algorithm 3 requires 2^{d+1} arithmetic operations for each iteration. Even though they share the same simulation logic, the memory cost may not be negligible, which may also incur additional computational cost of keeping track of large-scale vectors. There is then a trade-off between simulation costs and memory in general, which we do not exactly model in this work. To avoid exponentially many arithmetic operations and memory occupation in the steepest descent method, the comparison-based zeroth order method in Agarwal et al. (2011) can be extended to our case and reduce the number of arithmetic operations to a polynomial in d. In addition, we may consider using the following stochastic coordinate steepest descent method as a simple and fast implementation of Algorithms 3 and Algorithm 5, which is defined in the supplementary material. Let x^t be the current iteration point, and we update by two steps.

1. Simulate $H_{x^t}(y,\eta_y)$ for all $y\in\{x^t\pm e_i,\ i\in[d]\}$ until the confidence interval is small enough.

2. If, for some $y\in\{x^t\pm e_i,\ i\in[d]\}$, we know $f(y)<f(x^t)$ holds with high probability, then update $x^{t+1}=y$; otherwise, if $f(y)\ge f(x^t)-O(\epsilon)$ holds for all $y\in\{x^t\pm e_i,\ i\in[d]\}$ with high probability, then we terminate the iteration and return x^t as the solution.

We can see that the number of arithmetic operations for each iteration is $O(d)$. Moreover, an analogous method utilizing $O(d)$ neighboring points in the constructing gradient is shown to have good empirical performance in Jian (2017). However, theoretically, without extra assumptions on the problem structure, the stopping criterion $f(y)\ge f(x)-O(\epsilon)$ for all $y\in\{x^t\pm e_i,\ i\in[d]\}$ cannot ensure the approximate optimality of solution x. We give a counterexample to show that $f(y)\ge f(x)$ for all $y\in\{x^t\pm e_i,\ i\in[d]\}$ cannot ensure the optimality of solution x.

Example 3. We consider the case when $d=2$ and $n=3$. Define the objective function as

$$f(x,y) := 2|x-y|-|x+y-2|,\quad \forall (x,y)\in\{1,2,3\}^2.$$

We can verify that $f(x,y)$ is a $L^\natural$-convex function and its minimizer is (3, 3) with optimal value –4. Considering point (2, 2), we can calculate that

$$f(2,2)=-2, f(1,2)=1, f(3,2)=-1, f(2,1)=1, f(2,3)=-1.$$

Hence, the guarantee is satisfied at (2, 2), but the point is not a minimizer of $f(x)$.

Finally, in the case when the indifference zone parameter c is known, we can prove that choosing $\epsilon = Nc$ is enough for the (c,δ)-PCS-IZ guarantee. We provide the algorithm and its complexity analysis in the online appendix.

7. Numerical Experiments

In this section, we implement our proposed simulation–optimization algorithms that are guaranteed to find high-confidence, high-precision PGS

solutions. We first consider the optimal allocation problem of a queueing system, in which we show the advantage of using the truncation step. Next, we consider an artificially constructed $L^\natural$-convex function, in which more details about the objective function landscape are available for the evaluation of the performance.

7.1. Optimal Allocation Problem

In the optimal allocation problem, we consider the 24-hour operation of a service system with a single stream of incoming customers. The customers arrive according to a doubly stochastic nonhomogeneous Poisson process with intensity function

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

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

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

$$\min_{x \in [N]^d} f(x). \tag{11}$$

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

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

Then, Murota (2003) proves that

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

is a $L^\natural$-convex function on the $L^\natural$-convex set

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

The optimization Problem (11) has the trivial solution $x_1 = \ldots = x_d = N$. However, in reality, it is also necessary to keeping the staffing cost low. There are two different approaches to achieve this goal. First, we can constrain the total number of servers $\sum_{i=1}^d x_i$ to be at most K, where $K \le Nd$ is a positive integer, and the optimization problem can be written as

$$\min_{y \in \mathcal{Y}} h(y) \quad \text{s.t. } y_d \le K. \tag{12}$$

On the other hand, we can add a regularization term $R(x_1, \ldots, x_d) := C/d \cdot \sum_{i=1}^d x_i = C/d \cdot y_d$ to the objective function, where $C > 0$ is a constant. The optimization problem can be written as

$$\min_{y \in \mathcal{Y}} h(y) + C/d \cdot y_d. \tag{13}$$

We refer to Problems (12) and (13) as the constrained and regularized problems, respectively. Our algorithms can be extended to this case by considering the Lovász extension $\tilde{h}(y)$ on the set

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

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

We first implement both algorithms on the trivial Problem (11) for 10 times. Because the optimal solution is known, it is possible to verify whether the solutions returned by algorithms are at most ϵ worse than the optimum at a confidence that is larger than $1 - \delta$. In the experiment, we run a sufficiently large number of simulation replications to verify the ϵ-optimality at the selected solution with confidence higher than $1 - \delta'$, where $\delta' \ll \delta$.

Next, we consider the performance of algorithms on Problems (12) and (13). We summarize the simulation costs and the objective values in Table 2. We can see that both algorithms return a similar objective value, and the simulation cost grows when d becomes larger. The growth rate is approximately quadratic. The simulation cost becomes smaller when N gets larger because we allow a larger suboptimality gap ($N/2$) when N is larger. We note that the feasible set of both problems is not a hypercube, and thus, the dependence of simulation costs on d and N is not

Table 2. Simulation Costs and Objective Function Values of Algorithm 2 on the Optimal Allocation Problem

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

exactly quadratic as indicated by our theory. In addition, we can see that the truncation plays an important role in reducing the simulation cost, especially when the dimension is high.

7.2. Separable Convex Function Minimization

We consider the problem of minimizing a stochastic $L^\natural$-convex function whose expectation is a separable convex function parameterized by a vector $c \in \mathbb{R}^d$ and the optimal solution $x^* \in \mathbb{R}^d$:

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

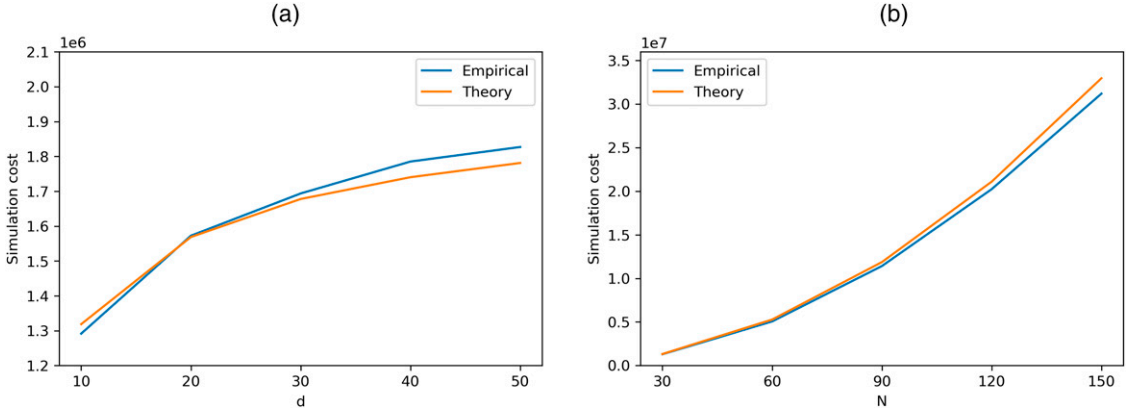
where $c_i \in [0.75, 1.25]$, $x_i^* \in \{1, \ldots, \lfloor 0.3N \rfloor\}$ for all $i \in [d]$ and

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

It is observed that the function $f_{c,x^*}(x)$ is a separable convex function and, therefore, is $L^\natural$-convex. Moreover, the function $f_{c,x^*}(x)$ has the optimum x^* associated with the optimal value zero. For stochastic evaluations, we add Gaussian noise with mean zero and variance one to each point $x \in \mathcal{X}$. Because of the $O[(y^*)^{-3/2}]$ growth rate, the landscape of $g(y^*; y)$ is flat around x^*. The advantage of this numerical example is that the expected objective function has a closed form, and we are able to verify the ϵ-optimality of the solutions returned by the proposed algorithms.

To analyze the effect of the dimension and the scale on the expected simulation cost, we first fix $d = 10$ and compare the performance when $N = 30, 60, 90, 120, 150$; then, we fix $n = 30$ and compare the performance when $d = 10, 20, 30, 40, 50$. The optimality guarantee parameters are chosen as $\epsilon = (d!)^{1/d}/5$ and $\delta = 10^{-6}$. In the one-dimensional case, this choice of ϵ ensures that the ϵ-sublevel set of the objective function approximately covers $N/4$ choices of decisions. We note that this choice of ϵ is only for comparisons between different (d, N), and our results can be extended to other choices of ϵ. We compute the average simulation cost of 100 independently generated models to estimate the expected simulation cost. Similar early stopping criteria are also applied.

Figure 1. (Color online) The Expected Simulation Costs of the Separable Convex Minimization Problem



Notes. (a) Expected simulation costs with $n = 30$. (b) Expected simulation costs with $d = 10$.

Figure 1 shows the results of fixed d and N. Because the choice of ϵ is dependent on d, the relation between the simulation costs and d is not clear. Therefore, we compare the simulation costs to the theoretical bound (up to a constant)

$$T(d,N) := N^2 d^2 \epsilon^{-2} \log 1/\delta.$$

More specifically, we compare the simulation costs to $0.87T(d,N)$ in this experiment, which corresponds to the "theory" curve in the figure. We can observe from the plotting that the growth of simulation costs matches our theory very well. This implies that our estimation on the performance of the truncated SSGD algorithm is tight on this example. Moreover, the optimality gap between the returned solutions and the optimal solution is smaller than ϵ for all experiments, which implies that the algorithm succeeds with high probability.

8. Conclusion

We propose computationally efficient simulation–optimization algorithms for large-scale simulation optimization problems that have high-dimensional discrete decision space in the presence of a convex structure. For a user-specified precision level, the proposed simulation–optimization algorithms are guaranteed to find a choice of decision variables that is close to the optimal within the precision level with desired high probability. We provide upper bounds on simulation costs for the proposed simulation–optimization algorithms. In this work, we mainly focus on algorithm design and theoretical guarantees. In future work, we seek to design better simulation–optimization algorithms that provide simulation costs with matching upper and lower bounds.

Acknowledgments

The authors are grateful to the anonymous reviewers, the associate editor, Jeff Hong, and Shane Henderson for very helpful comments and suggestions.

References

Agarwal A, Foster DP, Hsu DJ, Kakade SM, Rakhlin A (2011) Stochastic convex optimization with bandit feedback. *Adv. Neural Inform. Processing Systems* (Curran Associates, Inc., Red Hook, NJ) 24:1035–1043.

Agrawal S, Juneja S, Glynn P (2020) Optimal δ-correct best-arm selection for heavy-tailed distributions. *Proc. 31st Internat. Conf. Algorithmic Learn. Theory* (PMLR, Cambridge, MA) 117:61–110.

Ajalloeian A, Stich SU (2020) On the convergence of SGD with biased gradients. *Workshop Beyond First Order Methods ML Systems Internat. Conf. Machine Learn* (PMLR, Cambridge, MA).

Altman E, Gaujal B, Hordijk A (2000) Multimodularity, convexity, and optimization properties. *Math. Oper. Res.* 25(2):324–347.

Altman E, Gaujal B, Hordijk A (2003) *Discrete-Event Control of Stochastic Networks: Multimodularity and Regularity* (Springer, Berlin).

Axelrod B, Liu YP, Sidford A (2020) Near-optimal approximate discrete and continuous submodular function minimization. *Proc. 14th Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 837–853.

Bechhofer RE (1954) A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Ann. Math. Statist.* 25(1):16–39.

Belloni A, Liang T, Narayanan H, Rakhlin A (2015) Escaping the local minima via simulated annealing: Optimization of approximately convex functions. *Conf. Learn. Theory* (PMLR, Cambridge, MA) 240–265.

Chen J, Luss R (2018) Stochastic gradient descent with biased but consistent gradient estimators. Preprint, submitted July 31, https://arxiv.org/abs/1807.11880.

Chen X, Ankenman BE, Nelson BL (2013) Enhancing stochastic kriging metamodels with gradient estimators. *Oper. Res.* 61(2):512–528.

Chen X, Zhou E, Hu J (2018) Discrete optimization via gradient-based adaptive stochastic search methods. *IISE Trans.* 50(9): 789–805.

Chick SE (2006) Subjective probability and Bayesian methodology. Henderson SG, Nelson BL, eds. *Handbooks in Operations Research and Management Science*, vol. 13 (Elsevier, New York), 225–257.

Daniilidis A, Drusvyatskiy D (2020) Pathological subgradient dynamics. *SIAM J. Optim.* 30(2):1327–1338.

Davis D, Drusvyatskiy D (2019) Stochastic model-based minimization of weakly convex functions. *SIAM J. Optim.* 29(1): 207–239.

Davis D, Drusvyatskiy D, Kakade S, Lee JD (2020) Stochastic subgradient method converges on tame functions. *Foundations Comput. Math.* 20(1):119–154.

Devolder O, Glineur F, Nesterov Y (2014) First-order methods of smooth convex optimization with inexact oracle. *Math. Programming* 146(1–2):37–75.

Eckman DJ, Henderson SG (2018) Guarantees on the probability of good selection. *2018 Winter Simulation Conf.* (IEEE, Piscataway, NJ), 351–365.

Eckman DJ, Henderson SG (2020) Biased gradient estimators in simulation optimization. Bae KH, Feng B, Kim S, Lazarova-Molnar S, Zheng Z, Roeder T, Thiesing R, eds. *Proc. 2020 Winter Simulation Conf.* (IEEE, Piscataway, NJ).

Eckman DJ, Henderson SG (2021) Fixed-confidence, fixed-tolerance guarantees for ranking-and-selection procedures. *ACM Trans. Model. Comput. Simulation* 31(2):1–33.

Eckman DJ, Plumlee M, Nelson BL (2022) Plausible screening using functional properties for simulations with large solution spaces. *Oper. Res.* Forthcoming.

Even-Dar E, Mannor S, Mansour Y (2002) PAC bounds for multi-armed bandit and markov decision processes. *Internat. Conf. Comput. Learn. Theory* (Springer, Berlin), 255–270.

Fan W, Hong LJ, Nelson BL (2016) Indifference-zone-free selection of the best. *Oper. Res.* 64(6):1499–1514.

Favati P (1990) Convexity in nonlinear integer programming. *Ricerca Operativa* 53:3–44.

Freund D, Henderson SG, Shmoys DB (2017) Minimizing multimodular functions and allocating capacity in bike-sharing systems. *Internat. Conf. Integer Programming Combin. Optim.* (Springer), 186–198.

Fu MC (2002) Optimization for simulation: Theory vs. practice. *INFORMS J. Comput.* 14(3):192–215.

Fu MC, Qu H (2014) Regression models augmented with direct stochastic gradient estimators. *INFORMS J. Comput.* 26(3):484–499.

Fujishige S (1984) Theory of submodular programs: A Fenchel-type min-max theorem and subgradients of submodular functions. *Math. Programming* 29(2):142–155.

Fujishige S (2005) *Submodular Functions and Optimization* (Elsevier, Amsterdam).

Futschik A, Pflug G (1995) Confidence sets for discrete stochastic optimization. *Ann. Oper. Res.* 56(1):95–108.

Futschik A, Pflug GC (1997) Optimal allocation of simulation experiments in discrete stochastic optimization and approximative algorithms. *Eur. J. Oper. Res.* 101(2):245–260.

Garivier A, Kaufmann E (2016) Optimal best arm identification with fixed confidence. *Conf. Learn. Theory*, (PMLR, Cambridge, MA) 998–1027.

Graur A, Pollner T, Ramaswamy V, Weinberg SM (2020) New query lower bounds for submodular function minimization. *11th Innovations Theoretical Comput. Sci. Conf.* (Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Saarbrücken/Wadern, Germany).

Gutjahr WJ, Pflug GC (1996) Simulated annealing for noisy cost functions. *J. Global Optim.* 8(1):1–13.

Hong LJ, Nelson BL (2006) Discrete optimization via simulation using compass. *Oper. Res.* 54(1):115–129.

Hong LJ, Fan W, Luo J (2021) Review on ranking and selection: A new perspective. *Frontiers Engrg. Management* 8(3): 321–343.

Hong LJ, Nelson BL, Xu J (2010) Speeding up COMPASS for high-dimensional discrete optimization via simulation. *Oper. Res. Lett.* 38(6):550–555.

Hong LJ, Nelson BL, Xu J (2015) Discrete optimization via simulation. Fu MC, ed. *Handbook of Simulation Optimization* (Springer, New York), 9–44.

Hu J, Fu MC, Marcus SI (2007) A model reference adaptive search method for global optimization. *Oper. Res.* 55(3):549–568.

Hu J, Fu MC, Marcus SI (2008) A model reference adaptive search method for stochastic global optimization. *Comm. Inform. Systems* 8(3):245–276.

Hu Y, Zhang S, Chen X, He N (2020) Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning. *Adv. Neural Inform. Processing Systems* 33: 2759–2770.

Hunter SR, Nelson BL (2017) Parallel ranking and selection. Tolk A, Fowler J, Shao G, Yucesan E, eds. *Advances in Modeling and Simulation* (Springer, New York), 249–275.

Ito S (2019) Submodular function minimization with noisy evaluation oracle. *Adv. Neural Inform. Processing Systems* 32:12103–12113.

Jian N (2017) Exploring and exploiting structure in large scale simulation optimization. Unpublished PhD thesis, Cornell University, Ithaca, New York.

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

Jin C, Liu LT, Ge R, Jordan MI (2018) On the local minima of the empirical risk. *Proc. 32nd Internat. Conf. Neural Inform. Processing Systems*, 4901–4910.

Kaufmann E, Kalyanakrishnan S (2013) Information complexity in bandit subset selection. *Conf. Learn. Theory* (PMLR, Cambridge, MA), 228–251.

Kaufmann E, Cappé O, Garivier A (2016) On the complexity of best-arm identification in multi-armed bandit models. *J. Machine Learn. Res.* 17(1):1–42.

Kim SH, Nelson BL (2006) Selecting the best system. Henderson SG, Nelson BL, eds. *Handbooks in Operations Research and Management Science*, vol. 13 (Elsevier, New York), 501–534.

Kleywegt AJ, Shapiro A, Homem-de Mello T (2002) The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* 12(2):479–502.

L'Ecuyer P (1990) A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Sci.* 36(11):1364–1383.

Lee YT, Sidford A, Wong SCW (2015) A faster cutting plane method and its implications for combinatorial and convex optimization. *2015 IEEE 56th Annual Sympos. Foundations Comput. Sci.* (IEEE, Piscataway, NJ), 1049–1065.

Lim E (2012) Stochastic approximation over multidimensional discrete sets with applications to inventory systems and admission control of queueing networks. *ACM Trans. Model. Comput. Simulation* 22(4):1–23.

Lovász L (1983) Submodular functions and convexity. Bachem A, Grotschel M, Korte B, eds. *Mathematical Programming the State of the Art* (Springer, Berlin), 235–257.

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

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

Ma S, Henderson SG (2019) Predicting the simulation budget in ranking and selection procedures. *ACM Trans. Modeling Comput. Simulation* 29(3):1–25.

Mai V, Johansson M (2020) Convergence of a stochastic gradient method with momentum for non-smooth non-convex optimization. *Internat. Conf. Machine Learn.* (PMLR, Cambridge, MA), 6630–6639.

Mangoubi O, Vishnoi NK (2018) Convex optimization with unbounded nonconvex oracles using simulated annealing. *Conference Learn. Theory* (PMLR), 1086–1124.

Murota K (2003) *Discrete Convex Analysis (*Society for Industrial and Applied Mathematics, Philadelphia).

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

Nemirovskij AS, Yudin DB (1983) *Problem Complexity and Method Efficiency in Optimization* (Wiley-Interscience, New York).

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

Park C, Kim SH (2015) Penalty function with memory for discrete optimization via simulation with stochastic constraints. *Oper. Res.* 63(5):1195–1212.

Park C, Telci IT, Kim SH, Aral MM (2014) Designing an optimal water quality monitoring network for river systems using constrained discrete optimization via simulation. *Engrg. Optim.* 46(1):107–129.

Qu H, Fu MC (2014) Gradient extrapolated stochastic kriging. *ACM Trans. Model. Comput. Simulation* 24(4):1–25.

Semelhago M, Nelson BL, Song E, Wächter A (2020) Rapid discrete optimization via simulation with Gaussian Markov random fields. *INFORMS J. Comput.* 33(3):915–930.

Sen S, Higle JL (2001) Stabilization of cutting plane algorithms for stochastic linear programming problems. Floudas C, Pardalos P, eds. *Encyclopedia of Optimization* (Springer, Boston), 2434–2440.

Shaked M, Shanthikumar JG (1988) Stochastic convexity and its applications. *Adv. Appl. Probab.* 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: Comput. Sustainability*.

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

Wang H, Pasupathy R, Schmeiser BW (2013) Integer-ordered simulation optimization using r-spline: Retrospective search with piecewise-linear interpolation and neighborhood enumeration. *ACM Trans. Model. Comput. Simulation* 23(3):1–24.

Wang T, Xu J, Hu JQ, Chen CH (2021) Optimal computing budget allocation for regression with gradient information. *Automatica*, 134: 109927.

Wolff RW, Wang CL (2002) On the convexity of loss probabilities. *J. Appl. Probab.* 39(2):402–406.

Xu Y, Lin Q, Yang T (2016) Accelerated stochastic subgradient methods under local error bound condition. Preprint, submitted July 4, https://arxiv.org/abs/1607.01027.

Xu J, Nelson BL, Hong JL (2010) Industrial strength COMPASS: A comprehensive algorithm and software for optimization

via simulation. *ACM Trans. Model. Comput. Simulation* 20(1): 1–29.

Zhang S, He N (2018) On the convergence rate of stochastic mirror descent for nonsmooth nonconvex optimization. Preprint, submitted June 12, https://arxiv.org/abs/1806.04781.

Zhang J, Lin H, Jegelka S, Sra S, Jadbabaie A (2020) Complexity of finding stationary points of nonconvex nonsmooth functions. *Proc. 37th Internat. Conf. Machine Learn.*, vol. 119 (PMLR, Cambridge, MA), 11173–11182.

Zhong Y, Hong LJ (2018) Fully sequential ranking and selection procedures with PAC guarantee. *2018 Winter Simulation Conf.* (IEEE, Piscataway, NJ), 1898–1908.

Zhong Y, Hong LJ (2021) Knockout-tournament procedures for large-scale ranking and selection in parallel computing environments. *Oper. Res.* 70(1):432–453.

Haixiang Zhang is currently a PhD student in the department of mathematics at the University of California, Berkeley. His research interests focus on simulation optimization and nonconvex optimization.

Zeyu Zheng is an assistant professor in the department of industrial engineering and operations research, University of California, Berkeley. His has done research in simulation and nonstationary stochastic modeling and decision making.

Javad Lavaei is an associate professor in the department of industrial engineering and operations research, University of California, Berkeley. His research interests are in various interdisciplinary problems in control theory, optimization theory, power systems, and machine learning.