OPTIMIZATION VIA STRATEGIC LAW OF LARGE NUMBERS

By

Xiaohong Chen, Zengjing Chen, Wayne Yuan Gao,

Xiaodong Yan and Guodong Zhang

October 2025

COWLES FOUNDATION DISCUSSION PAPER NO. 2471

COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

YALE UNIVERSITY
Box 208281
New Haven, Connecticut 06520-8281

http://cowles.yale.edu/

Optimization via Strategic Law of Large Numbers

Xiaohong Chen
Cowles Foundation for Research in Economics, Yale University

Zengjing Chen
Zhongtai Securities Institute for Financial Studies, Shandong University

Wayne Yuan Gao
Department of Economics, University of Pennsylvania

Xiaodong Yan
School of Mathematics and Statistics, Xi'an Jiaotong University

Guodong Zhang
School of Statistics and Mathematics, Shandong University of Finance and Economics

October 31, 2025

Abstract

This paper proposes a novel framework for the global optimization of a continuous function in a bounded rectangular domain. Specifically, we show that: (1) global optimization is equivalent to optimal strategy formation in a two-armed decision problem with known distributions, based on the Strategic Law of Large Numbers we establish; and (2) a sign-based strategy based on the solution of a parabolic PDE is asymptotically optimal. Motivated by this result, we propose a class of **S**trategic **M**onte **C**arlo **O**ptimization (SMCO) algorithms, which uses a simple strategy that makes coordinate-wise two-armed decisions based on the signs of the partial gradient (or practically the first difference) of the objective function, without the need of solving PDEs. While this simple strategy is not generally optimal, it is sufficient for our SMCO algorithm to converge to a local optimizer from a single starting point, and to a global optimizer under a growing set of starting points. Numerical studies demonstrate the suitability of our SMCO algorithms for global optimization well beyond the theoretical guarantees established herein. For a wide range of test functions with challenging landscapes (multi-modal, non-differentiable and discontinuous), our SMCO algorithms perform robustly well, even in high-dimensional ($d = 200 \sim 1000$) settings. In fact, our algorithms outperform many state-of-the-art global optimizers, as well as local algorithms augmented with the same set of starting points as ours.

Keywords: Black box global optimization, multi-modal criterion, high dimension, two-armed machine, strategic sampling, strategic limit theorem, HJB equation, stochastic control

1 Introduction

It has been a notoriously difficult task to develop fast algorithms for global optimization of high-dimensional and potentially multi-modal continuous functions, which is known as the "curse of optimality" phenomenon. In this paper we propose a novel approach to solving the following optimization problem

$$\max_{x\in\Gamma} f(x), \tag{1.1}$$

where $\Gamma := [\underline{\mu}_1, \overline{\mu}_1] \times \cdots \times [\underline{\mu}_d, \overline{\mu}_d] \subset \mathbb{R}^d$, and $f : \Gamma \to \mathbb{R}$ is a continuous objective function.

Problem (1.1) is well-defined in the sense that there exists a global maximizer of any continuous deterministic function f over a compact region $\Gamma \subset \mathbb{R}^d$. Under mild regularity conditions, it is known that a continuous random function also has a global maximizer on a compact region Γ almost surely without assuming f being a global concave function (see, for example, Cox (2020)). However, when f is a multi-modal and high-dimensional continuous function, such as Rastrigin function with $d = 200$ or 1000, it is very difficult to develop a fast algorithm to solve Problem (1.1) and to mathematically prove its convergence to a global maximizer. Various Stochastic Gradient Descent (SGD) algorithms rely on accurate estimation of gradients (some also rely on Hessians) for iterative updates, and are the key behind recent success of deep learning (neural networks). The gradient based algorithms are only shown to converge to a global maximum when the objective function f is strictly concave, however. There are also many popular meta-heuristic global optimization algorithms for Problem (1.1), such as Simulated Annealing (SA) of Kirkpatrick et al. Kirkpatrick et al. (1983) and the Particle Swarm Optimization (PSO) of Eberhart and Kennedy Eberhart and Kennedy (1995). SA has been shown to converge weakly to global optimal points (see, for example, Geman and Hwang Geman and Hwang (1986)) but its convergence rate is known to be slow and it performs poorly in practice. Although a popular tool in applied work, the PSO has not been mathematically proved to converge to a global maximum for a general multi-modal continuous function yet.

We propose a novel framework for the global optimization problem (1.1). Our core theoretical results, Theorems 2.1 and 2.2, establish that solving problem (1.1) is equivalent to the search for an optimal (sampling) strategy in a two-armed decision model with known distributions, based on the Strategic Law of Large Numbers (StLLN). Precisely, the StLLN provides a new theoretical framework that translates the global optimization problem (1.1) into an equivalent problem of finding an approximately optimal strategy in a two-armed bandit model with known distributions. Both StLLN and the classical LLN concern the asymptotic behavior of the sample mean, but StLLN

considers strategic sampling from two distributions, while the classical LLN focuses on (nonstrategic) sampling from a single distribution. Hence, the classical LLN can be viewed as a special case of StLLN in which the two distributions coincide.

There are in general many approximately optimal strategies to solve problem (1.1). Our Theorem 2.3 establishes that one approximately optimal strategy is to use the signs of partial gradient of the unique solution to a PDE. For dimension $d = 1, 2$, one could apply finite-difference method to numerically solve for PDE, and then design a two-armed sampling algorithm for maximizing f according to the sign of the partial gradient of the numerical solution of the low-dimensional PDE. Unfortunately, solving a high-dimensional PDE numerically is in general a very difficult problem. Remark 2.2 points out that in one-dimensional case, under some extra symmetry and smoothness condition of f, the sign of gradient of the solution to the PDE coincides with the sign of the gradient of the original f itself.

Motivated by the above, we propose a feasible optimization algorithm based on a simple tractable strategy for the two-armed decision model, using only the sign information of the partial derivative of the original function f (instead of the PDE solution). The running sample mean under iterative application of this strategy is then returned as the candidate optimizer by our algorithm. We call this the "**S**trategic **M**onte **C**arlo **O**ptimization" (SMCO) algorithm. Even though the simple sign-based strategy we adopt is generally not optimal, we establish in Theorem 2.4 that the SMCO algorithm is nevertheless guaranteed to converge to *local* optimizers. We show in Theorem 2.5 that the SMCO algorithm, when augmented with a growing set of starting points, converges to a global optimizer. For practical applications, we implement the SMCO algorithms with multiple randomly generated starting points, and use step-adaptive finite differences in place of the true gradients. We also produce several versions of the SMCO algorithms with bookkeeping of running maximum, as well as some additional meta-heuristic and programming enhancements. The source code of SMCO, written in R, is publicly available on GitHub.

We conduct a wide range of numerical experiments to analyze the performance of our SMCO algorithms in solving complex global optimization problems in comparison with existing popular optimizers in R. We consider various deterministic test functions widely used as benchmarks in the global optimization literature, such as Rastrigin, Ackely, Michalewicz, Griewank, and more, of dimensions ranging from 1 to 1000. In addition, we also consider a variety of random test functions with challenging optimization landscapes (some non-differentiable and even discontinuous), such as sample criterion function for maximum score estimation, empirical welfare function for optimal

treatment assignment, conditional moment inequalities in dynamic binary choice models, in-class approximation and out-of-sample prediction based on ReLU neural networks in regression and classification settings, and Cauchy likelihood functions. We compare our SMCO algorithms with a range of widely used state-of-the-art optimization algorithms/packages implemented in R, including six local algorithms (GD, SignGD, ADAM, SPSA, L-BFGS, and BOBYQA) and six global ones (GenSA, SA, DEoptim, STOGO, GA, and PSO).

Our SMCO algorithms, designed using the simple sign-based strategy, perform remarkably well in complex global optimization problems. Specifically, our SMCO algorithms are consistently able to solve almost all test optimization problems to a reasonable degree of accuracy, using only a very limited number of randomly generated starting points for dimensions up to $d = 1000$. In contrast, all other algorithms considered here, with the sole exception of GenSA, suffer from some drastic errors in at least a number of test cases, while our algorithms and GenSA perform robustly well. In particular, the six local optimizers perform significantly worse than ours in many test cases, even though they use the same set of starting points as our SMCO algorithms. This provides evidence that the performance of our SMCO algorithms in global optimization cannot be explained by the use of multiple starting points alone. Moreover, our SMCO algorithms also compare favorably to most of the global optimizers, which is again remarkable given the small number of starting points we use, especially in higher dimensions. The performance of our SMCO algorithms thus appears unexpectedly strong, especially given that many of these optimizers have been widely tested and improved over many years, and that there are likely to be many aspects of our SMCO algorithms and code that can be significantly improved (such as coding in C++).

The numerical experiments demonstrate the promise of our SMCO algorithms, and more generally the optimization framework we proposed based on StLLN with two-armed decision strategy. It should be noted that our SMCO algorithm is based on a single simple sign-based strategy, which is not generally optimal. This suggests an array of future research questions under our two-armed StLLN framework: to analyze the performances of other strategies under this framework; to search for and/or implement the (approximately) optimal strategy in a tractable way; or to provide additional theoretical results that articulate the observed performance of our SMCO algorithms in solving global optimization problems.

Related Literature:

(i) **Gradient Descent and its variants**: Numerous optimization algorithms are based on gra-

dient descent (GD) and stochastic gradient descent (SGD), including sign gradient descent (SignGD) Bernstein et al. (2018); Moulay et al. (2019); Safaryan and Richtárik (2021) and adaptive gradient Kingma and Ba (2014); Hinton et al. (2012); Zeiler (2012); Reddi et al. (2019). Unlike traditional GD, which adjusts based on gradient magnitude, SignGD focuses solely on the gradient's sign for updates. Li et al. Li et al. (2023) showed that SignGD can improve stability and convergence in some cases, but its performance, like traditional GD, remains highly sensitive to the choices of step sizes and initial points. Guo and Fu Guo and Fu (2022) proposed to apply multi-arm bandit to select initial starting points for SGD and demonstrated its convergence advantages numerically. Adaptive GD methods automatically adjust step sizes using past stochastic gradients, which potentially lead to faster convergence and mitigate gradient explosion. However, these methods still face challenges, particularly in high-dimensional settings, where step size determination is complex. Additionally, they are prone to gradient explosion in early training phases Bengio et al. (1994); Pascanu et al. (2013), causing slow convergence or divergence. Our SMCO algorithm is most closely related to SignGD, but differs from SignGD in that our sample mean formulation based on StLLN automatically adapts step sizes with respect to both the iteration counter and the current optimizer position, with randomness incorporated in the mean-zero distribution around the upper/lower bounds. We demonstrate in simulations how our algorithm compares favorably with GD, SignGD, and its variants in terms of optimization performances, especially in optimization problems with challenging landscapes.

(ii) **Decision model-driven optimization:** Recently, bandit process, a simplified decision model of reinforcement learning, has shown significant success in optimization by employing random strategy with exploration, because this exploration expands the search space, enhances flexibility, and supports learning through trial and error. See Gittins et al. (2011); Li et al. (2018); Lattimore and Szepesvári (2020); Guo and Fu (2022) and the references therein for various applications of bandit algorithms with unknown arm distributions in optimization problems. Nevertheless, to the best of our knowledge, our work is the first to formally establish the equivalence of global optimization problem (1.1) to the optimal policy/strategy formation in a two-armed decision model with known arm distributions. We further propose a simple yet novel SMCO algorithm with multi-start, and rigorously establish its convergence to the global maximizer(s). The StLLN for the constructed decision model (Theorem 2.1) is our theoretical foundation for recasting the global optimization of a general continuous function as a decision rule problem, which is inspired by the LLN under the nonlinear expectation of Peng (2019).

(iii) **Two-armed bandit problems**: Our optimization framework is reminiscent of, but in fact distinct from, the classical two-armed bandit problems. While traditional bandit problems focus on maximizing cumulative rewards under uncertainty about arm distributions Robbins (1952); Lai and Robbins (1985); Agrawal (1995); Bubeck and Cesa-Bianchi (2012), we search for global maximizer(s) of a possibly non-concave continuous function f over Γ according to a two-armed decision model with known arm distributions. Precisely, SMCO selects between two fixed arms (or $2 \times d$ for d-dimensional problems) based on the sign changes of the objective function f's coordinate-wise partial gradient, ensuring quicker convergence to maximum points while reverting to standard Monte Carlo sampling when the arms coincide. While the standard Gradient Descent method is sensitive to the choice of starting points, the updating step-size, and the accuracy of gradient estimates for iterative updates, our SMCO relies on two-armed decision rule-generated samples, in which the selection of arms is determined by the signs of coordinate-wise partial gradient, while the update step size is automatically adapted through the sample mean process.

Outline of the rest of the paper:

Section 2 presents the Strategic Law of Large Numbers, the corresponding optimization framework, our SMCO algorithm design, and theoretical guarantees on the convergence of the SMCO algorithm. Section 3 conducts a wide range of numerical comparisons with the existing popular optimization algorithms. We conclude with Section 4. The SI Appendix contains supplemental materials, consisting of all the proofs, technical lemmas, the details on the practical implementations of the SMCO algorithms, and additional numerical experiments.

2 Theory and Algorithms

2.1 New Framework: Optimization via Strategic Law of Large Numbers

We use a simple decision model to illustrate our new approach to the optimization problem (1.1). Suppose that there are d two-armed decision models: these d decision models are independent of each other and differ only in the mean value corresponding to their arms. For $j = 1, \cdots, d$, machine j has two arms with populations X_j and Y_j and different known means $\overline{\mu}_j$ and $\underline{\mu}_j$ (without loss of generality, we assume $\overline{\mu}_j > \underline{\mu}_j$). One can draw one d-dimensional sample from these d two-armed decision models at each round; that is, the d components of the sample come from the d two-armed decision models, and decision model j can be drawn from either arm X_j or arm Y_j, but not both.

If X_j and Y_j are identical, the decision model j has a single arm and it becomes the classical Monte Carlo draws.

We will focus on the global optimization of a continuous function f on a bounded rectangular region Γ, $\Gamma := [\underline{\mu}_1, \overline{\mu}_1] \times [\underline{\mu}_2, \overline{\mu}_2] \times \cdots \times [\underline{\mu}_d, \overline{\mu}_d] \subset \mathbb{R}^d$. For a small positive δ, we assume that the function f is well-defined on the δ-extension region Γ_δ of Γ defined by $\Gamma_\delta := [\underline{\mu}_1 - \delta, \overline{\mu}_1 + \delta] \times [\underline{\mu}_2 - \delta, \overline{\mu}_2 + \delta] \times \cdots \times [\underline{\mu}_d - \delta, \overline{\mu}_d + \delta]$. In this paper all random variables are defined on a given probability space $(\Omega, \mathcal{F}, P)$. Consider d two-armed decision models, for $1 \le j \le d$, let (X_j, Y_j) denote the random reward of one of these d two-armed decision models. For simplicity, we assume that $\{X_j : 1 \le j \le d\}$ and $\{Y_j : 1 \le j \le d\}$ satisfy

$$X_j = \overline{\mu}_j + \xi_j, \quad Y_j = \underline{\mu}_j + \xi_j', \tag{2.1}$$

where ξ_j is a bounded random variable with mean 0, variance σ^2 and bounded by $\delta > 0$ (that is, $|\xi_j| \le \delta$), ξ_j' is an independent copy of ξ_j. For all $1 \le j \le d$, let $\{X_{i,j} : i \ge 1\}$ and $\{Y_{i,j} : i \ge 1\}$ respectively denote the outcomes from the arm X_j and arm Y_j at the ith round. We define a sampling strategy $\boldsymbol{\theta}$ as a sequence of (binary) random vectors $\boldsymbol{\theta} := \{\boldsymbol{\vartheta}_1, \cdots, \boldsymbol{\vartheta}_n, \cdots\}$, where $\boldsymbol{\vartheta}_i := (\vartheta_{i,1}, \cdots, \vartheta_{i,d}) \in \{0,1\}^d$ and $\vartheta_{i,j} = 1$ (respectively, $\vartheta_{i,j} = 0$) means in the ith round we pull the arm X_j (respectively, Y_j) of the jth slot machine. Let $\{\boldsymbol{Z}_i^{\boldsymbol{\theta}} := (Z_{i,1}^{\boldsymbol{\theta}}, \cdots, Z_{i,d}^{\boldsymbol{\theta}}) : i \ge 1\}$ be a sequence of d-dimensional random vectors denoting the ith reward obtained from these d slot machines related to the strategy $\boldsymbol{\theta}$. That is, for $j = 1, \cdots, d$,

$$Z_{i,j}^{\boldsymbol{\theta}} := \vartheta_{i,j} X_{i,j} + (1 - \vartheta_{i,j}) Y_{i,j} = \begin{cases} X_{i,j}, & \text{if } \vartheta_{i,j} = 1, \\ Y_{i,j}, & \text{if } \vartheta_{i,j} = 0. \end{cases} \tag{2.2}$$

We call a sampling strategy $\boldsymbol{\theta} = \{\boldsymbol{\vartheta}_1, \cdots, \boldsymbol{\vartheta}_n, \cdots\}$ *admissible* if $\boldsymbol{\vartheta}_n$ is $\mathcal{H}_{n-1}^{\boldsymbol{\theta}}$-measurable for all $n \ge 1$, where

$$\mathcal{H}_{n-1}^{\theta} = \sigma\{\eta, \boldsymbol{Z}_1^{\boldsymbol{\theta}}, \cdots, \boldsymbol{Z}_{n-1}^{\boldsymbol{\theta}}, \boldsymbol{\vartheta}_1, ..., \boldsymbol{\vartheta}_{n-1}\} \text{ for } n > 1, \text{ and } \mathcal{H}_0^{\boldsymbol{\theta}} = \sigma\{\eta\},$$

where $\eta := (\eta_1, \cdots, \eta_d) \in \Gamma$ denotes the initial point, which could be non-random or randomly drawn from a known distribution (such as uniform distribution on the domain or a deterministic grid distribution on the hyperdiagonal of Γ). Let Θ denote the collection of all admissible sampling strategies. For any strategy $\boldsymbol{\theta} \in \Theta$, we let

$$S_0^{\boldsymbol{\theta}} = \eta \ , \quad \text{and for all } n \ge 1, \ S_n^{\boldsymbol{\theta}} = S_{n-1}^{\boldsymbol{\theta}} + \boldsymbol{Z}_n^{\boldsymbol{\theta}}. \tag{2.3}$$

Theorem 2.1 (**Strategic Law of Large Numbers and a Two-sided Framework for Optimization**). *Let the d two-armed decision models $\{(X_j, Y_j) : 1 \le j \le d\}$ satisfy* (2.1). *Let $\boldsymbol{Z}_n^{\boldsymbol{\theta}}$ and $S_n^{\boldsymbol{\theta}}$ be defined in* (2.2) *and* (2.3). *Let f be any continuous function defined on Γ_δ. Then: the global optimization problem* (1.1) *is equivalent to the problem of asymptotically maximizing the "reward" with f*

$$\max_{x\in\Gamma} f(x) = \lim_{n\to\infty} \sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]. \tag{2.4}$$

Furthermore, we have,

$$\sup_{\boldsymbol{\theta}\in\Theta} P\left(\limsup_{n\to\infty} f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right) = \max_{x\in\Gamma} f(x)\right) = 1 \tag{2.5}$$

According to Theorem 2.1, the global optimization (1.1) can be transformed into two-armed decision model problem, which can be equivalently expressed as

$$\lim_{n\to\infty} \inf_{\boldsymbol{\theta}\in\Theta} R_f^{\boldsymbol{\theta}}(n) = 0, \tag{2.6}$$

where $R_f^{\boldsymbol{\theta}}(n) := \max_{x\in\Gamma} f(x) - E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]$ denote the "averaged regret" with f under $\boldsymbol{\theta}$. Therefore, to solve the global optimization problem (1.1), one just needs to find an asymptotically *optimal (sampling) strategy* $\boldsymbol{\theta}^o \in \Theta$ such that $\lim_{n\to\infty} R_f^{\boldsymbol{\theta}^o}(n) = 0$. In addition, if the objective function f is Lipschitz continuous, we can obtain the following speed of convergence result.

Theorem 2.2 (**Convergence Rate**). *Under conditions of Theorem 2.1, if f is Lipschitz continuous on Γ_δ (with a Lipschitz constant L_f), then:*

$$\left|\sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right] - \max_{x\in\Gamma} f(x)\right| \le L_f\left(\frac{\sigma\sqrt{d} + \sum_{j=1}^d \frac{|\overline{\mu}_j - \underline{\mu}_j|}{2}}{\sqrt{n}} + \frac{\sqrt{E[\|\eta\|^2]}}{n}\right). \tag{2.7}$$

Remark 2.1. *In our decision model, when these two arms (populations) coincide, our strategic laws of large numbers (StLLN) (Theorems 2.1 and 2.2) degenerates into the classical LLN, which serves as the theoretical foundation for the classical Monte Carlo simulation method. Consequently, we have the following viewpoints: repeatedly operating a single-armed decision model corresponds to the conclusions drawn from the classical laws of large numbers, and can thus be utilized for simulating the integration of a function. However, strategically operating a two-armed decision model aligns with our proposed strategic laws of large numbers, enabling its application in simulating the extremum of functions.*

Theorems 2.1 and 2.2 establish that any optimization problem (1.1) can be solved by finding some optimal (sampling) strategy of the two-armed decision model.[1] Both results are significant and broadly applicable, and motivate us to consider a class of optimization algorithms that compute sample means over draws from one of two pre-specified distributions around the two boundary points (in each dimension), along with a potentially optimal strategy to choose from the two distributions.

2.2 A Concrete Asymptotically Optimal Strategy

In the next theorem, we will suggest an asymptotically optimal (sampling) strategy using the signs of the partial gradients of the solutions to some parabolic PDEs associated with the Lipschitz continuous function f. This PDE approach enables us to find the global optima of function f on region Γ via a two-armed decision model.

For any small enough $\epsilon > 0$, consider the following parabolic PDE

$$\begin{cases} \partial_t u_\epsilon(t,x) + \sup\limits_{p\in\Gamma}\left[D_x u_\epsilon(t,x)\cdot p\right] + \dfrac{\epsilon^2}{2}\mathrm{tr}\left[D_x^2 u_\epsilon(t,x)\right] = 0, \quad (t,x)\in[0,1+\epsilon)\times\mathbb{R}^d, \\ u_\epsilon(1+\epsilon,x) = \hat{f}(x), \end{cases} \tag{2.8}$$

where $\hat{f}$ is an extension of f defined on $\mathbb{R}^d$, $D_x u_\epsilon$ and $D_x^2 u_\epsilon$ denote the gradient vector and the Hessian matrix of the function $x \to u_\epsilon(t,x)$ respectively. Under the bounded Lipschitz continuity assumption on $\hat{f}$, this PDE (2.8) is actually the HJB equation, and has a unique solution u_ϵ (see Yong and Zhou (Yong and Zhou, 1999, Theorem 5.2, Ch. 4)). Moreover, the solution $u_\epsilon(t,x)$ is continuously differentiable in x (see Peng (Peng, 2019, Theorem C.4.5), also see Appendix A for details). We can establish the following refinement of Theorem 2.2.

Theorem 2.3. *Let f be a Lipschitz continuous function on Γ_δ with Lipschitz constant L_f, and $\hat{f}$ be a bounded Lipschitz continuous (with Lipschitz constant L_f) extension of f defined on $\mathbb{R}^d$. For any small $\epsilon > 0$ let u_ϵ be the unique solution of PDE (2.8). For any $n \in \mathbb{N}$, define the strategy $\hat{\boldsymbol{\theta}}^{n,\epsilon} = (\hat{\boldsymbol{\vartheta}}_1^{n,\epsilon}, \cdots, \hat{\boldsymbol{\vartheta}}_i^{n,\epsilon}, \cdots)$ with $\hat{\boldsymbol{\vartheta}}_i^{n,\epsilon} = (\hat{\vartheta}_{i,1}^{n,\epsilon}, \cdots, \hat{\vartheta}_{i,d}^{n,\epsilon})$ as follows, for $j = 1, ..., d$,*

$$\hat{\vartheta}_{i,j}^{n,\epsilon} = \begin{cases} 1, & \text{if } \partial_{x_j} u_\epsilon\left(\dfrac{i}{n}, \dfrac{S_{i-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) > 0, \\ 0, & \text{otherwise.} \end{cases}, \text{ for } 1 \le i \le n, \text{ and } \hat{\vartheta}_{i,j}^{n,\epsilon} = 1 \text{ for } i > n. \tag{2.9}$$

[1] For other recent studies on nonlinear and strategic limit theory for decision model problems, the reader is referred to Chen et al. (2023a, 2024) and references therein.

Then: $\{\hat{\boldsymbol{\theta}}^{n,\epsilon}\}$ *is* ***asymptotically*** ϵ***-optimal*** *in the sense that*

$$\left|E\left[f\left(\frac{S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right]-\max_{x\in\Gamma}f(x)\right|\leq L_f\epsilon(2C_\Gamma+\sqrt{\epsilon}+\sqrt{1+\epsilon})+\frac{C_1E[\|\eta\|]}{n}$$
$$+C\left(\frac{1}{n}\sum_{i=1}^{d}\left(\sigma^2+\frac{(\overline{\mu}_i-\underline{\mu}_i)^2}{4}\right)+\frac{\sum_{i=1}^{d}|\overline{\mu}_i-\underline{\mu}_i|}{2n}+2\epsilon\sqrt{\frac{d}{n}}+\epsilon^2\right), \tag{2.10}$$

where $C_\Gamma>0$ *is a finite constant depending on the boundary of* Γ*, and* $C,C_1>0$ *are finite constants depending on* f*.*

According to Theorem 2.3, we can use the sign of the gradient vector D_xu_ϵ to construct a sequence of strategies $\hat{\boldsymbol{\theta}}^{n,\epsilon}$ such that $f(S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}/n)$ approximates $\max_{x\in\Gamma}f(x)$.

Remark 2.2. *To construct a sequence of asymptotically* ϵ*-optimal strategies, we only need to know the sign of each component of the gradient vector* D_xu_ϵ*. When* f *and its extension* $\hat{f}$ *satisfy some assumptions, the sign of the gradient vector* D_xu_ϵ *can be denoted by the sign of the gradient* f*. For example,*

- *When* $\hat{f}\in C^3(\mathbb{R})$*, assume that for any* $t\in[0,1]$*, the nodal set of* $\partial_xu_\epsilon(t,x)$ *is denoted by* $D=\{d_k,k=0,\pm1,\pm2,\cdots\}$*. Moreover, if* $\hat{f}$ *is strictly monotone on* $[d_k,d_{k+1}]$*, and the derivatives* $\hat{f}^{(k)}$ $(k=0,1,2,3)$ *have at most polynomial growth, then (see more details in Chen et al. (2023b); Liu and Sun (2025)):*

$$\left\{\partial_xu_\epsilon\left(\frac{i}{n},\frac{S_{i-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)>0\right\}\ \textit{is equivalent to}\ \left\{\partial_x\hat{f}\left(\frac{S_{i-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)>0\right\}.$$

  Hence the optimal strategy defined in (2.9) *can be replaced by the sign of the gradient vector* $D_x\hat{f}$*. A sufficient condition for the above assumption to hold is that* $\hat{f}\in C^3(\mathbb{R})$ *is a symmetric function with symmetric points* $D=\{d_k,k=0,\pm1,\pm2,\cdots\}$ *and strictly monotone on* $[d_k,d_{k+1}]$*.*

It is well-known that obtaining an analytical solution for PDE (2.8) is generally intractable, often necessitating numerical methods to approximate the sign of its solution's gradient. Consequently, the operational feasibility of locating the global optimum of function f via the aforementioned approach faces limitations. However, Remark 2.2 shows that for certain specific functions f, the sign of the gradient of the PDE solution coincides with that of the gradient of f itself. This motivates

us to investigate whether we can directly leverage the sign of function f's gradient to construct a class of asymptotically optimal strategies. Such strategies would locate the optimum of f within the two-armed decision-making framework. To address this, we present a simple but novel algorithm, which we call "Strategic Monte Carlo Optimization" in the next subsection.

2.3 Strategic Monte Carlo Optimization (SMCO) Algorithm

Motivated by the theoretical results in previous subsections, we now introduce a simple strategy $\boldsymbol{\theta}^*$ that just uses the sign information of the coordinate-wise partial derivative of f.

At round $m \geq 1$, we construct $\boldsymbol{\vartheta}_m^* = (\vartheta_{m,1}^*, \cdots, \vartheta_{m,d}^*)$ as follows:[2]

$$\vartheta_{m,j}^* = \begin{cases} 1, & \text{if } \partial_{x_j} f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right) \geq 0; \\ 0, & \text{otherwise,} \end{cases} \qquad j = 1, \cdots, d. \tag{2.11}$$

Intuitively, suppose the current sample mean is in a region where the function f is increasing, then we aim to maximize f by selecting a sample from the arm with a higher population mean in the next round; if the sample mean is in a region where the function f is decreasing, then we select the next sample from the arm with a lower (population) mean. Figure 2(c) shows the different iterative mechanisms for attaining the maximum value by the GD method (blue arrow line) and our SMCO (red arrow line) respectively. It also demonstrates that under the simple sign-based strategy $\boldsymbol{\theta}^*$ given in (2.11), the sample average $S_n^{\boldsymbol{\theta}^*}/n$ converges to the one of the local maxima of f. This result constitutes the Strategic Law of Large Numbers for the decision model problem (Theorem 2.4 below). In the one-dimensional setting, our proposed two-armed decision model process devises strategies that combine the two arms with known distributions to identify the maximum value of a non-concave function f. This approach differs from the classical two-armed decision model, which explores optimal arms to achieve the maximum average reward for each arm under a monotonic function f with unknown distributions.

The following theorem establishes the StLLN according to the sign-based strategy $\boldsymbol{\theta}^*$ described in Algorithm 1, which ensures that under $\boldsymbol{\theta}^*$, the sample mean $S_n^{\boldsymbol{\theta}^*}/n$ converges to one of the local maxima of f on Γ.

Theorem 2.4 (Local Optimization under the Sign-Based Strategy)**.** *Let f be a continuously differentiable function on Γ_δ, and satisfy the following condition: For any $k = 1, \cdots, d$ and a fixed*

[2]When $m = 1$, $S_0^{\boldsymbol{\theta}^*}/0$ can be defined as η, which is uniform distributed on Γ.

Algorithm 1 Strategic Monte Carlo Optimization (SMCO)

Input: Function f, tolerance level ε, initial point $\hat{x}_0$, with $S_0^{\boldsymbol{\theta}^*} := \hat{x}_0$.

for $n = 0$ **to** ∞ **do**
  for $j = 1$ **to** d **do**
    if $\partial_{x_j} f(\hat{x}_n) \geq 0$ **then**
      $\vartheta^*_{n+1,j} = 1$, draw $Z^{\boldsymbol{\theta}^*}_{n+1,j} = X_{n+1,j}$ from arm X_j (larger mean in [2.1]);
    else
      $\vartheta^*_{n+1,j} = 0$, draw $Z^{\boldsymbol{\theta}^*}_{n+1,j} = Y_{n+1,j}$ from arm Y_j (smaller mean in [2.1]).
    end
  end
  Update $S^{\boldsymbol{\theta}^*}_{n+1} := S^{\boldsymbol{\theta}^*}_n + \boldsymbol{Z}^{\boldsymbol{\theta}^*}_{n+1}$, with $\boldsymbol{Z}^{\boldsymbol{\theta}^*}_{n+1} = (Z^{\boldsymbol{\theta}^*}_{n+1,1}, \cdots, Z^{\boldsymbol{\theta}^*}_{n+1,d})$.
  Update $\hat{x}_{n+1} = S^{\theta^*}_{n+1}/(n+1)$.
  if $|f(\hat{x}_{n+1}) - f(\hat{x}_n)| \leq \varepsilon$ **then**
    break
  end
end

Output: Final point $\hat{x}^* := \hat{x}_{n+1}$ and final function value $\hat{f}^* := f(\hat{x}^*)$.

integer $K_k \in \mathbb{N}$, there exist two sets of points $\underline{\mu}_k \leq c_k^1 < \cdots < c_k^{K_k} \leq \overline{\mu}_k$ and $\underline{\mu}_k - \delta = b_k^0 < b_k^1 < \cdots < b_k^{K_k - 1} < b_k^{K_k} = \overline{\mu}_k + \delta$, such that for any fixed $x = (x_1, \cdots, x_d) \in \Gamma_\delta$, the function $f(x_1, .., x_{k-1}, \cdot, x_{k+1}, .., x_d)$ is strictly increasing on (b_k^{j-1}, c_k^j) and strictly decreasing on $[c_k^j, b_k^j)$ for $j = 1, \cdots, K_k$.

Let $\mathcal{M}$ denote all local maximum points of f on Γ, and for any $\varepsilon > 0$, $\mathcal{M}_\varepsilon = \bigcup_{c \in \mathcal{M}} B(c, \epsilon)$, where $B(c, \varepsilon) = \{x \in \Gamma_\delta : \|x - c\|_\infty < \varepsilon\}$. Let $\boldsymbol{\theta}^* = (\boldsymbol{\vartheta}^*_1, \cdots, \boldsymbol{\vartheta}^*_m, \cdots)$, with $\boldsymbol{\vartheta}^*_m = (\vartheta^*_{m,1}, \cdots, \vartheta^*_{m,d})$, be the strategy defined in (2.11)*. Then:*

$$\lim_{n \to \infty} P\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n} \in \mathcal{M}_\varepsilon\right) = 1. \tag{2.12}$$

Remark 2.3. *In Theorem 2.4, we make a sufficient condition that f is continuously differentiable. In fact, the above conclusion can be obtained similarly for continuous function f that is not differentiable. For a non-differentiable function, we can approximate it by a smooth function, for example, $f_\varepsilon(x) = \int f(x + \varepsilon y)\phi(y)dy \to f(x)$, as $\varepsilon \to 0$, where ϕ is the probability density function of a standard normal distribution. Then, one can simulate the extremes of f by applying the method to f_ε for a small enough ε.*

Remark 2.4. *The coordinate separability condition essentially ensures that (local) coordinate-wise updates are sufficient to increase the value of the function in each iteration. The condition plays a role that is akin to the coordinate-wise smoothness condition (Lipcchitz property of gradients)*

imposed for sign gradient descent Bernstein et al. (2018) and coordinate gradient descent Richtárik and Takáč (2016). Note that we do not impose additional smoothness conditions beyond the stated monotonicity requirements.

Next, we show that SMCO, when augmented with a growing set of multiple starting points, attains the global maximum for functions with multiple local maxima. Specifically, we show theoretically that, given a starting point in a local neighborhood of the global maximizer, the SMCO algorithm modified with a "boosted" starting iteration index is guaranteed to converge to the global maximizer.

Theorem 2.5. *Let f be a continuously differentiable function on Γ_δ. For any global maximum x^* of f on Γ, assume that there exists $\delta_0 > 0$, such that for any $y \in B(x^*, \delta_0)$, $sgn(\partial_{x_i} f(y)) = sgn(x_i^* - y_i), i = 1, \cdots, d$. Then there exists an integer $N_0 > 0$ such that for any $m \geq N_0$, whenever $S_m^{\boldsymbol{\theta}}/m \in B(x^*, \delta_0)$, it holds that for any $\varepsilon > 0$,*

$$\lim_{n\to\infty} P\left(\left\| \frac{S_n^{\boldsymbol{\theta}'}}{n} - x^* \right\|_\infty < \varepsilon\right) = 1, \tag{2.13}$$

where $B(x^*, \delta_0) = \{x \in \Gamma_\delta : \|x - x^*\|_\infty < \delta_0\}$ and $\boldsymbol{\theta}' = (\boldsymbol{\vartheta}_1, \cdots, \boldsymbol{\vartheta}_m, \boldsymbol{\vartheta}^*_{m+1}, \boldsymbol{\vartheta}^*_{m+2} \ldots)$.

Consequently, given a growing (deterministically, randomly, or quasi-randomly generated) set of starting points $\mathcal{X}_m$ such that $\#(\mathcal{X}_m) = m$ and $P(B(x^*, \delta_0) \cap \mathcal{X}_m \neq \emptyset) \to 1$ as $m \to \infty$, the SMCO algorithm run from all points in $\mathcal{X}_m$ with a "boosted" starting iteration index[3] $n_0 = N_0$ is guaranteed to attain a global maximizer x^* asymptotically as $n \to \infty$ and $m \to \infty$.

Finally, we present a rate of convergence of our SMCO algorithm 1 when f has an unique local maximum point $x^* \in \Gamma$, which could be on the boundary of Γ.

Theorem 2.6 (**Convergence Rate II**). *Let f be a continuous twice-differential function on Γ_δ, and assume that f has an unique local maximum point $x^* = (x_1^*, \cdots, x_d^*)$ on Γ, and further the ith component of f is strictly increasing on $(\underline{\mu}_i - \delta, x_i^*)$ and strictly decreasing on $[x_i^*, \overline{\mu}_i + \delta)$. Then we*

[3]Formally, running SMCO with a boosted iteration index n_0 means the following: given a starting point x_0, set the initial sum $S_0 := n_0 x_0$, and in the n-th iteration, update the running sum as in Algorithm 1 but compute the sample mean $x_{(n+1)} := S_{(n+1)}/(n+n_0)$. Effectively, this is as if the algorithm starts with the iteration counter $n = n_0$ instead of $n = 0$.

have,

$$\left|E\left[f\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right]-f(x^*)\right| \leq \max\left\{L_f\left(\frac{\sigma\sqrt{d}}{\sqrt{n}}+\frac{\sqrt{E[\|\eta\|^2]}}{n}\right), \frac{K_f\left(d\sigma^2+\sum_{j=1}^d(\overline{\mu}_j-\underline{\mu}_j)^2\right)+L_f E[\|\eta\|]}{n}\right\}, \tag{2.14}$$

where L_f is the Lipschitz constant of f, $K_f = \max_{i,j}\sup_{x\in\Gamma_\delta}|\partial^2_{x_ix_j}f(x)|$ and σ^2 is the uniform variances of $\{X_j\}$ and $\{Y_j\}$ as defined in (2.1)*. Moreover,*

$$E\left[\left\|\frac{S_n^{\boldsymbol{\theta}^*}}{n}-x^*\right\|^2\right] \leq \frac{d\sigma^2+\sum_{j=1}^d(\overline{\mu}_j-\underline{\mu}_j)^2}{n}+\frac{E[\|\eta\|^2]}{n^2}. \tag{2.15}$$

3 Numerical Experiments

3.1 Three Implemented Versions of SMCO algorithms

We implement the following three versions of our SMCO algorithms, which we use in the numerical experiments in Section 3. See Appendix B for some additional details .

(1) SMCO: Our baseline algorithm that implements Algorithm 1 with the adaptive-size finite difference as described above. We set the random variable ξ_j in Equation [2.1] to be $U[-\delta_j,\delta_j]$ with $\delta_j := \delta\left(\overline{\mu}_j-\underline{\mu}_j\right)$. We set the bound expansion parameter $\delta = 0.05$, the convergence tolerance at 1e-6, and the maximum number of iterations to be 500.

(2) SMCO-R: A refined version of SMCO with two programming/meta-heuristic enhancements. First, SMCO-R records and exploits the running maximum of all test function evaluations: This is purely bookkeeping and does not add any additional evaluations of the test functions beyond those for finite-difference calculations and convergence checks. From a programming perspective, the running maximum generated in past iterations is "free information" that requires essentially no additional computation and memory storage. Second, SMCO-R adopts a two-stage procedure: it starts by running SMCO for at most 50% iterations of the designated maximum number of iterations, and then runs the rest 50% iterations with a more local search (and smaller step sizes). Specifically, the end point of the initial stage is used as the start point of the second stage, namely $x_0^{(2)}$, and the second stage SMCO with the starting iteration counter set to $n = 1000$. Specifically, the partial sum is initialized at $S_0^{(2)} := 1000x_0^{(2)}$, the next point is updated according to $x_1^{(2)} := \left(S_0^{(2)}+\boldsymbol{Z}_1^{(2)}\right)/1001$, and etc. Other hyperparameters are set to be the same as in SMCO.

(3) SMCO-BR: A "boosted" version of SMCO-R that runs two passes of SMCO-R with: (i) the first pass is exactly the same as SMCO-R, (ii) the second pass runs a more local version of SMCO-R with the starting iteration counter set to $n = 100$, as motivated by Section 2.3 and Theorem 2.5. Specifically, given any starting point x_0, the second pass initializes the running sum as $S_0 := 100x_0$, computes the first update as $x_1 = (S_0 + \boldsymbol{Z}_1)/101$, and so on and so forth. Since SMCO-BR effectively runs two separate passes of SMCO-R, to make the comparison fair, we set the maximum number of iterations to be half of that for SMCO/SMCO-R, so that the running times of the three versions are similar. Other hyperparameters are set to be the same as in SMCO.

3.2 Alternative Optimization Algorithms in R for Comparison

Below we list two groups of state-of-the-art optimization algorithms (implemented in R) to which we compare our SMCO/-R/-BR algorithms in terms of performance accuracy (in root mean squared error) and computation speed (in second). See Appendix C for more information on these algorithms and their hyperparameter configurations.

Algorithm Group I: Six Local Optimizers This set consists of six gradient-based or derivative-free optimization algorithms that are designed to converge to a local optimum starting from a given starting point: (1) GD, (2) L-BFGS, (3) SignGD, (4) SPSA, (5) ADAM, (6) BOBYQA. To make the comparison as fair as possible, we pass the exact sets of starting points used for our SMCO algorithms to the following algorithms as well: for single-start comparisons, our SMCO algorithms and the Group I algorithms are all run from a common randomly drawn starting point; for multi-start comparisons, they are all run from the same set of starting points.

Algorithm Group II: Six Global Optimizers This set consists of six R packages that implement meta-heuristic global optimization algorithms: (1) GenSA (2) SA (3) DEoptim (4) STOGO (5) GA (6) PSO. We use the default hyperparameters set in these packages. Since these algorithms are explicitly designed for global optimization (and many already implement multiple starting points from within), we do not pass any generated starting points to these algorithms whenever applicable.[4]

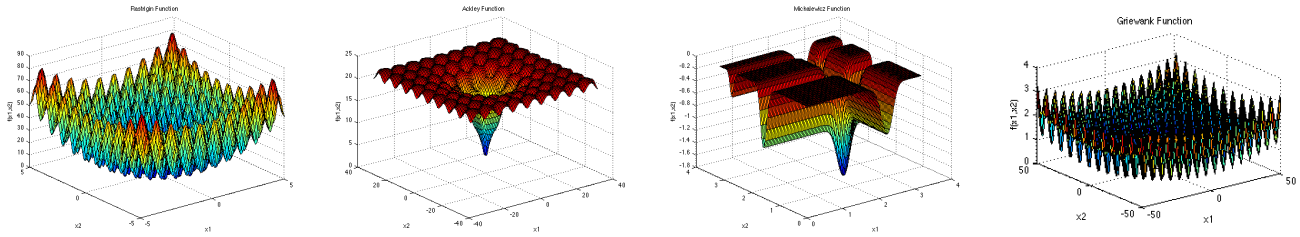
3.3 Optimization of Deterministic Test Functions

We first run the optimization algorithms on the following set of deterministic test functions, which have been widely used to benchmark the performances of global optimization algorithms. The func-

[4] A few of these global algorithms nevertheless require a user-designated starting point, in which case we pass a start point randomly chosen from the set of multiple starting points generated for our SMCO and the Group I algorithms.

Figure 1: Visualization of Test Functions

(a) Rastrigin Function (b) Ackley Function (c) Michalewicz Function (d) Griewank Function



Retrieved from: https://www.sfu.ca/~ssurjano/optimization.html

tion definitions (and hyperparameter values, if applicable) follow those given in the optimization section of the Virtual Library of Simulated Experiments: Test Functions and Datasets[5] Surjanovic and Bingham (2013). We have tested a wide array of test functions from different categories with various dimensionalities. Due to page limit, we only report results in the main text on the following four test functions: (1) **Rastrigin** function on $[-5.12, 5.12]^d$. This function has a unique global minimum of value 0 at the origin and multiple global maxima at $x \in \{\pm 4.52299\}^d$. (2) **Ackley** function on $[-32.768, 32.768]^d$. This function has a unique global minimum of value 0 at the origin and multiple global maxima of unknown values. (3) **Griewank** function on $[-600, 600]^d$. This function has a unique global minimum of value 0 at the origin and multiple global maxima of unknown values. (4) **Michalewicz** function on $[0, \pi]^d$. This function has a unique global minimum (of values reported up to $d = 75$) and multiple global maxima of the value 0 at the vertices $\{0, \pi\}^d$. These functions are constructed to have challenging optimization landscapes with many local optima, boundary optima, (near) flat regions/directions, steep drops/ridges, or a combination of these. Figure 1 contains visualizations (retrieved from the Virtual Library of Simulated Experiments) of these test functions in 2-dimensional forms.

One notable feature of several of the test functions above lies in their symmetric forms, i.e., they are constructed with symmetric domains and its global minimum attained at the origin (or, the center point of the domain). Such a kind of symmetry is unlikely to be found in realistic optimization problems. This would make the optimization problem drastically easier for any algorithm that explicitly or implicitly searches the center point of the domain, making the comparison between algorithms less clear and fair. Hence, we randomly rotate, shift, and asymmetrize the default domains of the test functions above, which we formally describe in the following. Specifically, for a given test function f on its default domain $\prod_{j=1}^{d}[\underline{\mu}_j, \overline{\mu}_j]$, write $\Delta\mu_j := \overline{\mu}_j - \underline{\mu}_j$ be the length

[5] Available online at https://www.sfu.ca/~ssurjano/optimization.html

of the j-th coordinate domain. Then, we shift and asymmetrize the domain in each coordinate independently as follows: for each $j = 1, ..., d$, we draw $r_j \sim Bernoulli\,(0.5)$, $v_j \sim N(0,1)$, and $\nu_j \sim U[0,1]$, and set

$$\underline{\mu}_j^{mod} := \underline{\mu}_j + [v_j + r_j\,(0.2 + 0.1\nu_j) - (1 - r_j)\,(0.4 + 0.2\nu_j)]\,\Delta\mu_j,$$
$$\overline{\mu}_j^{mod} := \overline{\mu}_j + [v_j + r_j\,(0.4 + 0.2\nu_j) - (1 - r_j)\,(0.2 + 0.1\nu_j)]\,\Delta\mu_j.$$

Essentially, the above functions constitute the following operations in each coordinate. (i) Randomly shift the center point of the original domain by $v_j \cdot \Delta\mu_j$; (ii) Randomly decide whether to asymmetrize the domain rightwards ($r_j = 1$) or leftwards ($r_j = 0$). If asymmetrizing rightwards, i.e., $r_j = 1$, then push the lower bound $\underline{\mu}_j$ rightwards by $25\% \pm 5\%$ of $\Delta\mu_j$, and push the upper bound $\overline{\mu}_j$ rightwards by $50\% \pm 10\%$ of $\Delta\mu_j$. If asymmetrizing leftwards, i.e., $r_j = 0$, then push the lower bound $\underline{\mu}_j$ leftwards by $50\% \pm 10\%$ of $\Delta\mu_j$, and push the upper bound $\overline{\mu}_j$ leftwards by $25\% \pm 5\%$ of $\Delta\mu_j$. Note that the domain of each coordinate is perturbed independently, so the "asymmetrization directions" generally differ across coordinates. After shifting and asymmetrizing the domain into $\mathcal{X}^{mod} := \prod_{j=1}^{d} \left[\underline{\mu}_j^{mod}, \overline{\mu}_j^{mod}\right]$ as described above, we further randomly rotate the domain by randomly drawing a rotation (orthonormal) matrix Q of dimension d, and define the final test function by

$$f^{mod}(x) \;:=\; f\left(Q\left(x - v \odot \Delta\mu\right)\right),$$

where $\odot$ represents the coordinate-wise (Hadamard) product.

For each of the test functions (with rotation and domain modification), we test the optimization algorithms both for minimization and maximization. Due to the randomness in the random starting point generation as well as the stochastic nature of many optimization algorithms used, we run multiple replications of all optimization algorithms and report summary statistics for their performances across the replications. Specifically, we report the following performance measures: "Time", "RMSE", and "AExx", whose definitions we now explain. "Time" reports the average time (in second) it takes for a particular algorithm to return the final function value, averaged across all the replications. "RMSE" and "AExx" are relative accuracy measures defined as follows. Since the true maximum and minimum values may be different on the rotated, enlarged, and asymmetrized domain from those on the default domains, we first define "Best Value" as the best value found by all algorithms (ours and those in Groups I & II) over all replications. We then define "RMSE" as

the square root of the mean squared error between the value returned by a specific algorithm and the "Best Value" found, and define "AExx" as the xx-th percentile of the absolute error between the value returned by a specific algorithm and the "Best Value".

Table 1 summarizes how our SMCO algorithms **Group I (local)** algorithms in maximizing and minimizing all four test functions of **dimension $d = 2$**, when run from a **single** uniformly generated starting point, based on 500 replications. First, our SMCO algorithms tend to produce significantly more accurate results than all Group I algorithms in most of the test configurations reported in Table 1. The RMSE and AE percentiles of our SMCO algorithms are often orders of magnitude smaller than those of other algorithms in most configurations. Second, only our SMCO algorithms display robust performance across all eight test configurations in comparison with all other algorithm. Finally, our SMCO algorithms display remarkably stable performance in terms of the error percentiles: the differences between the 50th, 95th, and 99th error percentiles of our SMCO algorithms tend to be substantially smaller than those of all other Group I algorithms, demonstrating that the performance of our SMCO algorithms is much *less sensitive* to the random starting point relative to others.

In Appendix D, **Tables 5-8** report comparisons of our SMCO algorithms with both Group I and II algorithms for **dimensions $d = 2$**, **10**, **20**, **50**, in which our SMCO algorithms and all Group I algorithms are run with $10\sqrt{d} \approx 14$, 32, 45, and 71 **uniformly drawn starting points**. The general pattern there shows that: First, our SMCO algorithms perform robustly well across most settings, both in terms of RMSE and AE99. Second, Group I (local) algorithms perform significantly worse than SMCO and Group II (global) algorithms, especially when the dimension increases. Consequently, due to the space limit, we only compare our SMCO algorithms with Group II algorithms in higher-dimensional settings in the main text.

We now report results on how our SMCO algorithms compare with Group II (global) algorithms in a **high-dimensional settings** ($d = 200$). Here our SMCO algorithms are run from $\sqrt{d} \approx 14$ uniformly drawn starting points in a parallelized manner. The Group II (global) algorithms are run without any specified starting point (when the specification of a starting point is not required by the algorithm) or one randomly drawn starting point (when the specification of a starting point is required by the algorithm), again at their default settings. Again, our (multi-start) SMCO algorithms demonstrate robust performance across almost all test configurations, with speed and accuracy levels that compare competitively, and in fact often favorably, to Group II (global) algorithms. Among Group II (global) algorithms, GenSA appears to be the only one that shows

Table 1: Deterministic Functions, Dimension $d = 2$: Comparison with Group I Algorithms

Performance		RMSE	AE50	AE95	AE99	Time	RMSE	AE50	AE95	AE99	Time
Min	**Function: Best Value**	**Rastrigin: 0**					**Griewank: 0**				
	SMCO	**7.8**	**8.0**	**13.0**	**13.0**	**0.016**	**9.9**	**9.8**	**11.1**	**11.6**	**0.017**
	SMCO-R	**2.8**	**2.0**	**5.0**	**8.0**	**0.016**	**0.04**	**0.03**	**0.08**	**0.12**	**0.017**
	SMCO-BR	**2.7**	**2.0**	**5.0**	**8.0**	**0.016**	**0.06**	**0.03**	**0.11**	**0.16**	**0.018**
	GD	36.7	17.0	80.9	104.7	0.010	196.2	141.4	345.6	437.4	0.032
	SignGD	62.3	42.2	115.4	156.2	0.035	208.4	141.4	403.3	492.5	0.026
	SPSA	39.6	16.9	84.6	135.4	0.002	137.9	72.6	277.6	330.7	0.003
	ADAM	15.4	8.3	30.0	48.5	0.008	211.5	142.1	418.3	506.2	0.014
	L-BFGS	43.8	15.9	97.5	115.5	0.0004	141.9	78.8	289.6	333.8	0.0005
	BOBYQA	42.9	9.9	88.5	135.4	0.002	18.9	0.2	0.8	15.9	0.002
	Function: Best Value	**Ackley: 4.44e-16**					**Michalewicz: -1.801303**				
	SMCO	**3.1**	**0.4**	**0.9**	**19.8**	**0.018**	**1.25**	**1.15**	**1.64**	**1.64**	**0.018**
	SMCO-R	**2.3**	**0.01**	**0.01**	**19.5**	**0.019**	**1.11**	**1.14**	**1.37**	**1.43**	**0.018**
	SMCO-BR	**2.6**	**0.01**	**0.01**	**19.4**	**0.018**	**1.09**	**1.14**	**1.15**	**1.37**	**0.018**
	GD	19.0	19.9	20.0	20.3	0.021	1.74	1.84	1.99	1.99	0.012
	SignGD	19.0	19.9	20.0	20.0	0.034	1.55	1.63	1.99	1.99	0.023
	SPSA	19.0	19.9	20.0	20.0	0.002	1.45	1.34	1.99	1.99	0.004
	ADAM	19.1	19.9	20.0	20.7	0.001	1.84	1.99	2.00	2.00	0.006
	L-BFGS	18.8	19.9	20.0	20.4	0.0005	1.65	1.63	1.99	1.99	0.001
	BOBYQA	15.1	18.0	20.0	20.2	0.002	1.47	1.37	1.99	1.99	0.003
Max	**Function: Best Value**	**Rastrigin: 213.5824**					**Griewank: 622.3434**				
	SMCO	**17.4**	**10.3**	**10.4**	**88.7**	**0.015**	**2.5**	**1.0**	**5.5**	**6.7**	**0.016**
	SMCO-R	**13.4**	**10.2**	**10.2**	**58.5**	**0.015**	**1.1**	**3e-13**	**2.1**	**3.6**	**0.015**
	SMCO-BR	**15.6**	**10.2**	**10.2**	**80.6**	**0.015**	**2.2**	**0.1**	**4.8**	**8.5**	**0.016**
	GD	85.3	82.7	173.3	192.9	0.004	469.6	477.0	609.1	618.5	0.033
	SignGD	130.2	130.9	171.1	173.1	0.035	471.1	477.0	609.1	618.5	0.027
	SPSA	85.1	80.6	167.2	176.0	0.004	472.0	477.6	609.1	618.9	0.013
	L-BFGS	82.8	81.9	165.0	171.1	0.0004	447.6	459.9	608.7	618.5	0.0005
	BOBYQA	73.6	20.3	167.0	171.1	0.002	204.4	0.0	293.4	575.1	0.002
	Function: Best Value	**Ackley: 22.35029**					**Michalewicz: 1.21406**				
	SMCO	**0.20**	**0.18**	**0.29**	**0.37**	**0.017**	**0.25**	**0.21**	**0.41**	**0.41**	**0.017**
	SMCO-R	**0.03**	**0.02**	**0.06**	**0.08**	**0.018**	**0.17**	**6.5e-7**	**0.41**	**0.41**	**0.016**
	SMCO-BR	**0.02**	**0.01**	**0.05**	**0.08**	**0.018**	**0.15**	**7.6e-7**	**0.41**	**0.41**	**0.016**
	GD	1.90	0.08	4.44	8.83	0.017	0.85	0.72	1.21	1.21	0.011
	SignGD	2.20	0.08	4.75	9.55	0.032	0.65	0.41	1.21	1.21	0.024
	SPSA	2.19	0.08	4.75	9.55	0.002	0.94	1.21	1.21	1.21	0.002
	L-BFGS	1.66	0.08	3.64	7.71	0.000	0.77	0.41	1.21	1.21	0.001
	BOBYQA	0.25	0.02	0.61	1.19	0.002	0.58	0.41	1.21	1.21	0.002

Based on 500 reps. In each rep, all algorithms run from a **single uniformly drawn starting point**. Computed using Posit Cloud. SMCO/-R/-BR run with 1e-6 tolerance and 500 maximum iterations.

overall better performance than our SMCO algorithms in Table 2. That said, it still yields noticeably larger errors (in terms of RMSE and AE99) than ours, say, in the minimization of Rastrigin and Ackley functions. Additional results in higher dimensions $d = 400, 500$ and 1000, as well as various sensitivity checks with respect to key hyperparameters of SMCO/-R/-BR, are available in Appendix D.

3.4 Optimization of Random Functions

In this section, we investigate two randomly generated test functions that arise from two important economic contexts. The first test function is the sample criterion for maximum score estimation, as introduced by Manski Manski (1975, 1985). The maximum score estimator can be used to estimate model parameters in discrete choice models under weak assumptions on the error terms, and since Manski (1975, 1985) there has been much subsequent work in economics that either extends the applicability of the maximum score estimation approach or analyzes its intriguing properties. Specifically, the sample criterion function for maximum score estimator is defined as $f_{MS}(\beta) := \frac{1}{N}\sum_{i=1}^{N} Y_i \mathbb{1}\left\{X_{i1} + X_{i,-1}'\beta \geq 0\right\}$ where $(X_i, Y_i)_{i=1}^{N}$ is a random sample with $X_i = (X_{i,1}, X_{i,-1}) \in \mathbb{R}^{d+1}$ being a vector of covariates and Y_i being a binary outcome variable generated by the model $Y_i = \mathbb{1}\left\{X_{i1} + X_{i,-1}'\beta_0 + \epsilon_i \geq 0\right\}$ under the conditional median restriction $\text{med}(\epsilon_i | X_i) = 0$ at the true parameter β_0. The maximum score estimator is then defined as any solution to the maximization problem $\max_\beta f_{MS}(\beta)$. Note that the parameter to be optimized over, β, enters through the discontinuous indicator function, and furthermore the outcome variable Y_i is also a binary variable itself. Due to the discreteness issues, the maximization of the maximum score criterion function is well-known to be computationally challenging, especially when the dimension of parameters becomes (moderately) high. We investigate the performance of our SMCO algorithms in optimizing the maximum score criterion function. Specifically, we build the test function f_{MS} using simulated data generated as follows. We simulate $X_{i,1} \sim \mathcal{N}(0, d)$ and $X_{i,2} \sim ... \sim X_{i,d+1} \sim \mathcal{N}(0, 1)$. We set $\beta_0 \in \mathbb{R}^d$ through random draws of $\beta_{0,j} \sim \mathcal{N}(0, 1)$. We simulate $\epsilon_i \sim \mathcal{N}(0, d)$ and set $N = 500$. The variances of $X_{i,1}$ and ϵ_i are set equal to d so that the effects from $X_{i,1}$ and ϵ_i will not be dominated by those from $X_{i,-1} \in \mathbb{R}^d$ as the dimension d increases. We consider $d = 5, 10, 20$ and 50, and constrain the parameter search in the rectangle $[-20, 20]^d$.

The second randomly generated test function we consider arises from empirical welfare maximization as introduced in Kitagawa and Tetenov (2018), which has inspired a very active line of related

Table 2: Deterministic Functions, Dimension $d = 200$: Comparison with Group II Algorithms

Test Function		**Rastrigin**			**Griewank**			**Ackley**			**Michalewicz**		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
	Best Value	**1596.819**			**2.35e-13**			**19.85938**			**-62.77543**		
	SMCOms	**285**	**375**	**34.9**	**1.07**	**1.07**	**34.4**	**0.101**	**0.106**	**31.9**	**42.22**	**44.08**	**42.7**
	SMCOms-R	**262**	**308**	**33.4**	**0.95**	**0.95**	**37.2**	**0.009**	**0.014**	**32.3**	**28.42**	**29.41**	**42.5**
	SMCOms-BR	**272**	**324**	**32.6**	**0.96**	**0.96**	**36.5**	**0.010**	**0.015**	**32.6**	**30.33**	**32.21**	**43.0**
Min	GenSA	1718	2011	268.3	3.05e-11	6.10e-11	302.2	0.107	0.114	271.7	6.27	12.03	813.4
	SA	3947	4338	0.2	11626	14002	0.1	1.651	1.679	0.3	52.76	54.53	0.36
	DEoptim	3813	3968	31.5	5856	6086	33.4	1.545	1.563	32.2	47.41	48.91	43.5
	STOGO	1293	1293	285.6	4.38e-09	4.38e-09	307.1	0	0	295.4	43.48	43.48	404.4
	GA	2496	2623	0.5	7372	7920	0.5	1.450	1.518	0.5	49.28	52.30	0.61
	PSO	590	812	4.9	81	143	5.1	1.546	1.590	5.1	43.13	45.36	6.19
	Best Value	**18239.57**			**72331.4**			**22.33939**			**69.53472**		
	SMCOms	**7328**	**7648**	**34.7**	**192**	**199**	**38.0**	**0.119**	**0.125**	**31.6**	**36.67**	**38.25**	**42.48**
	SMCOms-R	**4419**	**4939**	**33.3**	**278**	**287**	**36.2**	**0.105**	**0.110**	**30.8**	**20.91**	**22.16**	**42.47**
	SMCOms-BR	**4512**	**5043**	**34.7**	**601**	**627**	**37.4**	**0.030**	**0.034**	**31.7**	**25.43**	**26.87**	**42.49**
Max	GenSA	2605	2975	275.9	0	0	312.7	0.004	0.007	273.5	4.65	8.98	814.1
	SA	5653	7180	0.3	32221	36635	0.3	0.488	0.505	0.3	55.48	57.14	0.37
	DEoptim	7978	8191	31.8	36012	36465	33.7	0.421	0.432	32.6	49.83	50.98	43.8
	STOGO	10987	10987	290.0	9908	9908	310.0	0.120	0.120	299.9	45.26	45.26	407.8
	GA	8977	9310	0.5	45978	46953	0.5	0.450	0.489	0.5	49.86	51.20	0.61
	PSO	998	1719	5.0	20792	23940	5.2	0.429	0.448	5.2	46.50	48.88	6.23

Based on 10 reps. SMCO/-R/-BR run from $\sqrt{d} \approx 14$ **uniformly drawn starting points** in parallel. Run on high performance cluster using 15 cores on 1 node.

research in economics on optimal treatment assignment and policy evaluation. Specifically, the empirical welfare function, say, under linear treatment assignment, is defined in Kitagawa and Tetenov (2018) as $f_{EW}(\beta) := \frac{1}{N}\sum_{i=1}^{N}\left(\frac{D_i}{p(X_i)} - \frac{1-D_i}{1-p(X_i)}\right) Y_i \mathbb{1}\left\{\beta_1 + X_i'\beta_{-1} \geq 0\right\}$ where $(D_i, X_i, Y_i)_{i=1}^N$ is a random sample with D_i being a binary treatment status variable with (known/chosen) propensity scores $p(X_i) := \mathbb{E}[D_i | X_i]$, $X_i \in \mathbb{R}^{d-1}$ being a vector of covariates, and Y_i is a scalar-valued outcome variable. The optimal linear treatment assignment rule is then obtained by solving the optimization problem $\max_\beta f_{EW}(\beta)$. Similarly to the maximum score criterion function, f_{EW} also features the discontinuous indicator function that makes the optimization problem challenging. We also investigated the performance of the SMCO algorithms in solving empirical welfare maximization problems. Specifically, we build the test function f_{MS} based on a combination of real and simulated data. The real data are constructed based on the replication package of Kitagawa and Tetenov (2018), which is constructed from the Job Training Partnership Act (JTPA) data on job training programs. The specific data set we use contains $N = 9223$ observations on the binary treatment status variable D_i, the outcome variable Y_i (post-program earning), and two covariates: pre-program earning (X_{i1}) and education (X_{i2}). The propensity score is known to be constant at $p(X_i) = 2/3$. Following Kitagawa and Tetenov (2018), we set $X_{i3} = X_{i2}^2$ and $X_{i4} = X_{i2}^3$ to be quadratic and cubic transformation of education. We then standardize $X_{i1}, ..., X_{i4}$ by demeaning it and rescaling its standard deviation to 1. We also augment the data by simulating additional covariates $X_{i5}, ..., X_{i,d-1} \sim \mathcal{N}(0,1)$ for higher dimensions $d \geq 6$. Again, we consider $d = 5, 10, 20$ and 50, and constrain the parameter space to $[-20, 20]^d$.

Table 3 reports the performance of our SMCO algorithms, along with the competing algorithms, in maximizing the two randomly generated test functions f_{MS} and F_{EW} as described above. First, all gradient-based local algorithms perform significantly worse than our SMCO algorithms and the global algorithms: this is expected given that the gradients of f_{MS} and F_{EW} are both degenerate almost everywhere given that the parameter enters through the indicator function. Second, our SMCO algorithms (especially SMCO-R/BR) perform robustly well across almost all the configruations in Table 3: in fact, SMCO-R/BR yields the best results among all algorithms except for one configuration (f_{EW} with $d = 5$).

Appendix E contains supplemental numerical results on maximum score estimation and empirical welfare maximization up to $d = 100$, including replication of Table 3 using 10 replications only, as well as supplemental robustness checks of maximum score estimation with Cauchy errors and different bound buffer parameters of SMCO/-R/-BR.

Table 3: Maximization of Random Functions, Dimension $d = 5, 10, 20$ & 50

Dimension		$d = 5$			$d = 10$			$d = 20$			$d = 50$		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Maximum Score Estimation: $\max_\beta f_{MS}(\beta)$													
Best Value			**0.42**			**0.448**			**0.462**			**0.424**	
(I)	GD	0.135	0.196	0.06	0.146	0.190	0.03	0.167	0.198	0.08	0.157	0.174	0.13
	SignGD	0.135	0.196	0.08	0.146	0.190	0.03	0.167	0.198	0.09	0.157	0.174	0.11
	ADAM	0.135	0.196	0.07	0.146	0.190	0.06	0.167	0.198	0.08	0.157	0.174	0.11
	SPSA	0.135	0.196	0.11	0.146	0.190	0.08	0.167	0.198	0.11	0.157	0.174	0.12
	L-BFGS	0.135	0.196	0.07	0.146	0.190	0.07	0.167	0.198	0.09	0.157	0.174	0.13
	BOBYQA	0.084	0.116	0.07	0.100	0.141	0.08	0.130	0.153	0.09	0.134	0.155	0.22
	SMCO	**0.008**	**0.012**	**0.20**	**0.009**	**0.012**	**0.33**	**0.013**	**0.019**	**0.76**	**0.056**	**0.068**	**5.09**
	SMCO-R	**0.002**	**0.004**	**0.21**	**0.005**	**0.006**	**0.33**	**0.006**	**0.008**	**0.77**	**0.010**	**0.016**	**5.19**
	SMCO-BR	**0.003**	**0.008**	**0.21**	**0.007**	**0.010**	**0.35**	**0.010**	**0.015**	**0.81**	**0.010**	**0.017**	**5.41**
(II)	GenSA	0.020	0.054	1.71	0.031	0.034	4.35	0.032	0.038	15.69	0.010	0.018	89.93
	SA	0.069	0.076	0.06	0.073	0.085	0.07	0.108	0.124	0.14	0.114	0.124	0.34
	DEoptim	0.013	0.054	0.30	0.036	0.038	0.76	0.045	0.050	2.78	0.056	0.064	13.18
	STOGO	0.052	0.052	3.49	0.068	0.068	8.18	0.080	0.080	28.43	0.082	0.082	182.28
	GA	0.050	0.058	0.22	0.036	0.049	0.25	0.054	0.074	0.38	0.059	0.070	0.65
	PSO	0.040	0.056	0.65	0.034	0.038	0.90	0.038	0.046	1.64	0.017	0.026	4.32
Empirical Welfare Maximization: $\max_\beta f_{EW}(\beta)$													
Best Value			**1541.032**			**2061.409**			**2649.911**			**3825.49**	
(I)	GD	752	1448	0.12	1288	1666	0.16	1827	2058	0.24	2934	3067	0.60
	SignGD	752	1448	0.13	1288	1666	0.17	1827	2058	0.25	2934	3067	0.60
	ADAM	752	1448	0.13	1288	1666	0.16	1827	2058	0.25	2934	3067	0.62
	SPSA	752	1448	0.11	1305	1711	0.14	1848	2114	0.21	2920	3063	0.69
	L-BFGS	721	1448	0.12	1199	1621	0.18	1799	2050	0.28	2927	3067	0.99
	BOBYQA	398	955	0.13	846	1267	0.20	1365	1622	0.40	2338	2599	1.51
	SMCO	**207**	**301**	**1.50**	**195**	**310**	**3.41**	**271**	**379**	**9.49**	**428**	**498**	**78.37**
	SMCO-R	**128**	**230**	**1.51**	**69**	**141**	**3.41**	**98**	**171**	**9.52**	**158**	**233**	**78.59**
	SMCO-BR	**140**	**229**	**1.55**	**85**	**198**	**3.46**	**144**	**252**	**9.57**	**274**	**361**	**78.71**
(II)	GenSA	104	150	13.37	220	368	32.37	639	918	91.25	1586	1762	455.24
	SA	214	318	0.41	431	638	0.56	881	1268	0.85	1939	2398	1.57
	DEoptim	116	161	2.37	233	279	6.11	640	723	17.54	1722	1805	87.03
	STOGO	134	134	27.94	470	470	66.33	851	851	181.67	2096	2096	891.62
	GA	160	238	1.08	288	453	1.36	689	973	1.90	1685	1831	3.69
	PSO	108	192	3.60	138	272	5.26	297	433	8.26	848	1051	21.52

Based on $500/d = 100, 50, 25, 10$ reps. SMCO/-R/-BR and Group (I) algorithms run from $\sqrt{d} \approx 2, 3, 4, 7$ **uniformly drawn starting points** in parallel. Run on high performance cluster with 8 cores on 1 node.

3.5 Additional Numerical Experiments

We have conducted many more numerical experiments to check the wide applicability of our SMCO algorithms. Some of which are reported in Appendices F and G. Specifically, see: Appendix F for results on two additional random test functions: one based on conditional moment inequalities in a dynamic binary choice model from economic contexts, the other based on mean squared in-class approximation error for ReLU neural networks; Appendix G for results on additional random test functions in statistics, such as Cauchy log-likelihood, ReLU neural network out of sample predictions, ReLU neural network out of sample misclassification problems. This section contains statistical performance measurements such as "out-of-sample" MSEs and the "out-of-sample" misclassification error rate. These are different from but complementary to the optimization performance measurements we focus on in the main text. We have also tested many other deterministic test functions, such as Bukin No.6, Damavandi, Dropwave, Easom, EggHolder, JennrichSampson, McCormick, Mishra No.6, Qing, Rosenbrock, Schwefel, Shubert, Six-Hump Camel, Trefethen, and Zakharov functions, with dimensions up to 200 (when applicable). We do not report them here due to the lack of space. Such results are available upon request, and can also be easily replicated using our SMCO source code on GitHub. The main findings from all these exercises are that, our SMCO algorithms, especially SMCO-R and SMCO-BR, perform very well across all these different configurations, essentially without any change in the hyper-parameters other than the dimension-adaptive number of starting points used. In fact, this observation suggests that the suitability of the SMCO algorithms for global optimization goes well beyond the theoretical guarantees we provide in Section 2.

4 Conclusion

Based on the Strategic Law of Large Numbers we establish, this paper proposes a unified two-armed decision model for black-box optimization. We transform the difficult problem of global optimization over a bounded rectangle region to optimal strategy formation of multiple two-armed decision models in infinite policy sets. We propose a novel class of Strategic Monte Carlo Optimization (SMCO) algorithms, for which we establish local and global convergence results. We also show via an extensive array of numerical experiments that our proposed SMCO algorithms perform consistently well across different complex multi-modal test functions and different dimensions (up to $d = 1000$), often outperforming a portfolio of state-of-the-art existing algorithms, even though our SMCO

algorithms are currently not coded in C++. Our work opens the door to a promising area of future research that explores and improves the theoretical properties as well as the practical implementation of optimization algorithms under our StLLN-based framework.

Data, Materials, and Software Availability

The Job Training Partnership Act (JTPA) data used in Section 3.4 can be constructed based on the replication package of Kitagawa and Tetenov (2018), which is publicly available at: https://doi.org/10.3982/ECTA13288. The R code used to generate the numerical studies in Section 3 and the Appendix are freely available for academic use on Github at https://github.com/wayne-y-gao/SMCO and https://github.com/yanxiaodong128/SMCO

Acknowledgments

We thank Michael Fu, Jiaqiao Hu, Faming Liang, Shige Peng, Andre Wibisono and Yinyu Ye for helpful discussions. This work was supported in part by the National Key R&D Program of China (Grant No. 2023YFA1008701), the Key Project of the National Natural Science Foundation of China (No. 12431017). the National Natural Science Foundation of China (Grant No. 12371292, 11901352, 12371148). Xiaohong Chen's research is partially supported by Cowles Foundation for Research in Economics, Yale University.

References

Agrawal, R. (1995): "Sample mean based index policies with $O(\log n)$ regret for the multi-armed bandit problem," *Advances in Applied Mathematics*, 27, 1054–1078.

Bengio, Y., P. Simard, and P. Frasconi (1994): "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, 5, 157–166.

Bernstein, J., Y. Wang, K. Azizzadenesheli, and A. Anandkumar (2018): "SignSGD: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*, 560–569.

Brooks, S. P. and B. J. T. Morgan (1995): "Optimization using simulated annealing," *Journal of the Royal Statistical Society. Series D: The Statistician*, 44, 241–257.

Bubeck, S. and N. Cesa-Bianchi (2012): "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, 5, 1–122.

Chen, Z. (2016): "Strong laws of large numbers for sub-linear expectations," *Science China Mathematics*, 59, 945–954.

Chen, Z., L. Epstein, and G. Zhang (2023a): "A central limit theorem, loss aversion and multi-armed bandits," *Journal of Economic Theory*, 209, 105645.

——— (2024): "Approximate optimality and the risk/reward tradeoff given repeated gambles," *Economic Theory*, https://doi.org/10.1007/s00199-024-01614-4.

Chen, Z., X. Feng, S. Liu, and W. Zhang (2023b): "Bang–bang control for a class of optimal stochastic control problems with symmetric cost functional," *Automatica*, 149, 110849.

Cox, G. (2020): "Almost sure uniqueness of a global minimum without convexity," *The Annals of Statistics*, 48, 584–606.

Eberhart, R. and J. Kennedy (1995): "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micromachine and Human Science, Nagoya, Japan, March 13-16*, IEEE, 39–43.

Feng, O. Y., Y. C. Kao, M. Xu, and R. J. Samworth (2024): "Optimal convex M-estimation via score matching," *arXiv preprint arXiv:2403.16688*.

Gao, W. Y. and R. Wang (2025): "Identification of Nonlinear Dynamic Panels under Partial Stationarity," *arXiv preprint arXiv:2401.00264*.

Geman, S. and C. R. Hwang (1986): "Diffusions for global optimization," *SIAM Journal on Control and Optimization*, 24, 1031–1043.

Gittins, J., K. Glazebrook, and R. Weber (2011): *Multi-Armed Bandit Allocation Indices*, John Wiley & Sons, second ed.

Guo, P. and M. C. Fu (2022): "Bandit-based multi-Start strategies for global continuous optimization," in *2022 Winter Simulation Conference (WSC)*, IEEE, 3194–3205.

Hinton, G., N. Srivastava, and K. Swersky (2012): "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

Kingma, D. P. and J. Ba (2014): "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*.

Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi (1983): "Optimization by simulated annealing," *Science*, 220, 671–680.

Kitagawa, T. and A. Tetenov (2018): "Who should be treated? empirical welfare maximization methods for treatment choice," *Econometrica*, 86, 591–616.

Krylov, N. V. (1987): *Nonlinear elliptic and parabolic equations of the second order*, Dordrecht, Holland: D. Reidel Publishing Company.

Lai, T. L. and H. Robbins (1985): "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, 6, 4–22.

Lattimore, T. and C. Szepesvári (2020): *Bandit Algorithms*, Cambridge University Press.

Li, L., K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar (2018): "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research*, 18, 1–52.

Li, X., K. Y. Lin, L. Li, Y. Hong, and J. Chen (2023): "On faster convergence of scaled sign gradient descent," *IEEE Transactions on Industrial Informatics*, 20, 1732–1741.

Liu, D. and J. Nocedal (1989): "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, 45, 503–528.

Liu, S. and D. Sun (2025): "The sign and the co-monotonicity of Z for a class of decoupled FBSDEs: Theory and applications," *Systems & Control Letters*, 203, 106160.

Manski, C. F. (1975): "Maximum score estimation of the stochastic utility model of choice," *Journal of econometrics*, 3, 205–228.

——— (1985): "Semiparametric analysis of discrete response: Asymptotic properties of the maximum score estimator," *Journal of econometrics*, 27, 313–333.

Moulay, E., V. Léchappé, and F. Plestan (2019): "Properties of the sign gradient descent algorithms," *Information Sciences*, 492, 29–39.

Pascanu, R., T. Mikolov, and Y. Bengio (2013): "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, PMLR, 1310–1318.

Peng, S. (2019): *Nonlinear Expectations and Stochastic Calculus under Uncertainty: with Robust CLT and G-Brownian Motion*, Springer Nature.

Powell, M. (2009): "The BOBYQA algorithm for bound constrained optimization without derivatives," Tech. Rep. NA06, DAMTP, University of Cambridge.

Reddi, S. J., S. Kale, and S. Kumar (2019): "On the convergence of Adam and beyond," *arXiv preprint arXiv:1904.09237*.

Richtárik, P. and M. Takáč (2016): "Distributed coordinate descent method for learning with big data," *Journal of Machine Learning Research*, 17, 1–25.

Robbins, H. (1952): "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, 58, 527–535.

Safaryan, M. and P. Richtárik (2021): "Stochastic sign descent methods: New algorithms and better theory," in *International Conference on Machine Learning*, PMLR, 9224–9234.

Spall, J. (1992): "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, 37, 332–341.

——— (1998): "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Transactions on Aerospace and Electronic Systems*, 34, 817–823.

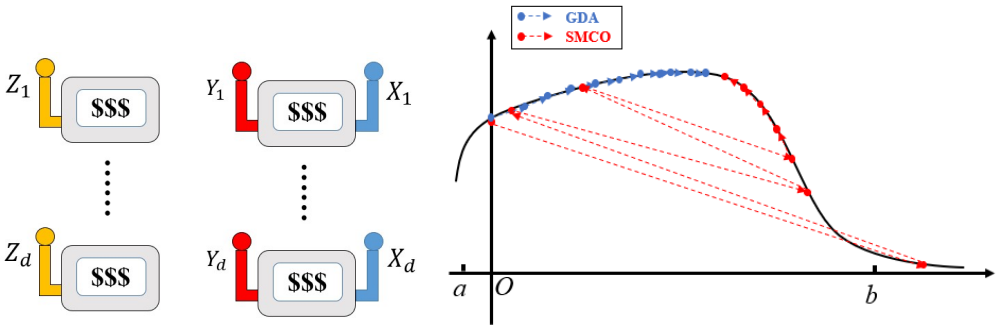
Surjanovic, S. and D. Bingham (2013): "Virtual library of simulation experiments: test functions and datasets," `https://www.sfu.ca/~ssurjano/optimization.html`.

Yong, J. and X. Zhou (1999): *Stochastic Controls: Hamiltonian Systems and HJB Equations*, Springer Science & Business Media.

Zeiler, M. D. (2012): "AdaDelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*.

Appendix

We first present a graphic illustration of our two-armed decision model as well as the difference between our novel SMCO algorithm against the standard gradient descent algorithm in one dimension:



(a) Classical Monte Carlo (b) Strategic Monte Carlo (c) Iterative Mechanism by GDA and SMCO

Figure 2: (a). Classic Monte Carlo sampling from a sequence of one-armed slot machines, with arm population $\{Z_1, \cdots, Z_d\}$; (b). Strategic Monte Carlo sampling from a sequence of two-armed slot machines, with arms populations $\{X_1, \cdots, X_d\}$ and $\{Y_1, \cdots, Y_d\}$; (c) The iterative mechanism for attaining the extreme value by GD (blue arrow line) and SMCO (red arrow line) under $d = 1$, where red points denote the iterative value $S_n^{\boldsymbol{\theta}^*}/n$ for SMCO given in Algorithm 1.

The Appendix contains several sections. Appendix A contains additional lemmas and proofs of the mathematical results in Section 2 of the main paper. Appendix B provides more details on the practical implementations of the SMCO algorithms. Appendix C shows the details on comparsion-group algorithms. Appendix D presents numerical experiments results that supplement those in Section 3.3 on deterministic test functions. Appendix E provides results that supplement those in Section 3.4 on randomly generated test functions. Appendix F presents results on two additional randomly generated functions constructed from ReLU neural networks. Appendix G presents comparisons using a variety of statistical random objective functions.

A Lemmas and Proofs

For convenience, we assume the rectangular domain $\Gamma = \prod_{i=1}^{d}[\underline{\mu}_i, \overline{\mu}_i]$ is symmetric about the origin (i.e. $\underline{\mu}_i = -\overline{\mu}_i$ for all i). If not satisfied, apply the translation $T(x) = x - t$ where

$t = \left(\frac{\underline{\mu}_1+\overline{\mu}_1}{2}, \ldots, \frac{\underline{\mu}_d+\overline{\mu}_d}{2}\right)$. This transformation yields a shifted domain $\widetilde{\Gamma} = \prod_{i=1}^d [\widetilde{\underline{\mu}}_i, \widetilde{\overline{\mu}}_i]$, where $\widetilde{\underline{\mu}}_i = \underline{\mu}_i - t_i, \widetilde{\overline{\mu}}_i = \overline{\mu}_i - t_i$, satisfying the symmetry condition, and a shifted function $\tilde{f}(x) = f(x+t)$. Consequently, for $\widetilde{\Gamma}$ we have $\widetilde{\underline{\mu}}_i + \widetilde{\overline{\mu}}_i = 0$ and $\widetilde{\overline{\mu}}_i = -\widetilde{\underline{\mu}}_i = \frac{\widetilde{\overline{\mu}}_i - \widetilde{\underline{\mu}}_i}{2}$ for all i, ensuring origin symmetry.

We first present two lemmas that will be used in the rest of the proofs. Let $\psi \in C_{b,Lip}(\mathbb{R}^d)$ be a bounded Lipschitz continuous function on $\mathbb{R}^d$. For any $\epsilon > 0$ and $t \in [0, 1+\epsilon]$, define

$$H_t^\epsilon(x) := u_\epsilon(t,x) := \sup_{a \in \mathcal{A}(t,1+\epsilon)} E\left[\psi\left(x + \int_t^{1+\epsilon} a_s ds + \int_t^{1+\epsilon} \epsilon \boldsymbol{I}_d dB_s\right)\right] \tag{A.1}$$

where $\boldsymbol{I}_d$ is a d-dimensional identity matrix and (B_t) is a d-dimensional Brownian motion and $\mathcal{F}_t$ is the natural filtration generated by (B_t). Let $\mathcal{A}(t,T)$ denote the set of all $\{\mathcal{F}_s\}$-adapted measurable processes valued in Γ.

Lemma A.1. *(a)* $\max_{x\in\Gamma} \psi(x) = \sup_{a\in\mathcal{A}(0,1)} E\left[\psi\left(\int_0^1 a_s ds\right)\right]$*; for any* $t \in [0,1], x \in \mathbb{R}^d$*,*

$$u(t,x) := \sup_{a\in\mathcal{A}(t,1)} E\left[\psi\left(x + \int_t^1 a_s ds\right)\right] = \max_{y\in\Gamma} \psi(x + (1-t)y).$$

(b) There are finite constants $c_1, c_2 > 0$ *such that*

$$\sup_{x\in\mathbb{R}^d} |u_\epsilon(t,x) - u(t,x)| \le \epsilon \times [c_1 + c_2(d(\epsilon + 1 - t))^{1/2}] \quad \textit{uniformly over } t \in [0,1].$$

(c) For any fixed $\epsilon > 0$*, the functions* $\{H_t^\epsilon(x)\}_{t\in[0,1]}$ *satisfy the following properties:*

(1) $H_t^\epsilon \in C_b^2(\mathbb{R}^d)$ *and each element of the gradient vector and Hessian matrix of* H_t^ϵ *are uniformly bounded for all* $t \in [0,1]$*.*

(2) *Dynamic programming principle: For any* $\gamma \in [0, 1+\epsilon-t]$*,*

$$H_t^\epsilon(x) = \sup_{a\in\mathcal{A}(t,t+\gamma)} E\left[H_{t+\gamma}^\epsilon\left(x + \int_t^{t+\gamma} a_s ds + \int_t^{t+\gamma} \epsilon \boldsymbol{I}_d dB_s\right)\right],\ x \in \mathbb{R}^d.$$

(3) *There exists a constant* C*, depends only on the uniform bound of* $D_x^2 H_t^\epsilon$*, such that for any* $n \in \mathbb{N}$*,*

$$\sum_{m=1}^n \sup_{x\in\mathbb{R}^d} \left| H^\epsilon_{\frac{m-1}{n}}(x) - H^\epsilon_{\frac{m}{n}}(x) - \frac{1}{n} \sup_{p\in\Gamma} \left[D_x H^\epsilon_{\frac{m}{n}}(x) \cdot p\right]\right| \le C\left(\frac{\sum_{i=1}^d |\overline{\mu}_i - \underline{\mu}_i|}{2n} + 2\epsilon\sqrt{\frac{d}{n}} + \epsilon^2\right).$$

Proof of Lemma A.1: For Result (a), we note that for any $x \in \Gamma$, let $a_s^x = x$ for $s \in [0,1]$, we have $a_s^x \in \mathcal{A}(0,1)$, then

$$\sup_{a\in\mathcal{A}(0,1)} E\left[\psi\left(\int_0^1 a_s ds\right)\right] \geq \sup_{x\in\Gamma} E\left[\psi\left(\int_0^1 a_s^x ds\right)\right] = \sup_{x\in\Gamma} E\left[\psi\left(x\right)\right] = \max_{x\in\Gamma}\psi(x).$$

In addition, since any $a \in \mathcal{A}(0,1)$ takes values in Γ, we have

$$\sup_{a\in\mathcal{A}(0,1)} \psi\left(\int_0^1 a_s(\omega) ds\right) \leq \max_{x\in\Gamma}\psi(x), \quad \omega \in \Omega.$$

Therefore,

$$\sup_{a\in\mathcal{A}(0,1)} E\left[\psi\left(\int_0^1 a_s ds\right)\right] \leq \sup_{a\in\mathcal{A}(0,1)} E\left[\sup_{a\in\mathcal{A}(0,1)} \psi\left(\int_0^1 a_s ds\right)\right] \leq \max_{x\in\Gamma}\psi(x).$$

Given the definition of $u(t,x)$, the relation $u(t,x) = \max_{y\in\Gamma}\psi(x+(1-t)y)$ can be established similarly.

For Result (b), follows from Result (a), we have uniformly over $t \in [0,1],\ x \in \mathbb{R}^d$,

$$\begin{aligned}
&|u_\epsilon(t,x) - u(t,x)| \\
=&\left|\sup_{a\in\mathcal{A}(t,1+\epsilon)} E\left[\psi\left(x+\int_t^{1+\epsilon} a_s ds + \int_t^{1+\epsilon}\epsilon \boldsymbol{I}_d dB_s\right)\right] - \sup_{a\in\mathcal{A}(t,1)} E\left[\psi\left(x+\int_t^1 a_s ds\right)\right]\right| \\
\leq& \sup_{a\in\mathcal{A}(t,1+\epsilon)} E\left[\left|\psi\left(x+\int_t^{1+\epsilon} a_s ds + \int_t^{1+\epsilon}\epsilon \boldsymbol{I}_d dB_s\right) - \psi\left(x+\int_t^1 a_s ds\right)\right|\right] \\
\leq& L_\psi \sup_{a\in\mathcal{A}(t,1+\epsilon)} E\left[\left\|\int_1^{1+\epsilon} a_s ds\right\| + \left\|\int_t^{1+\epsilon}\epsilon \boldsymbol{I}_d dB_s\right\|\right] \\
\leq& L_\psi \epsilon\left(\sqrt{\sum_{i=1}^d \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{4}} + \sqrt{d(1+\epsilon-t)}\right).
\end{aligned}$$

For Result (c) Part (1), for any $(t,x) \in [0,1+\epsilon]\times\mathbb{R}^d$, let $u_\epsilon(t,x) = H_t^\epsilon(x)$, it can be checked that u_ϵ is the solution of the PDE (A.2) (HJB-equation, see Yong and Zhou (Yong and Zhou, 1999, Theorem 5.2, Ch. 4)).

$$\begin{cases}
\partial_t u_\epsilon(t,x) + \sup_{p\in\Gamma}\left[D_x u_\epsilon(t,x)\cdot p\right] + \dfrac{\epsilon^2}{2}\mathrm{tr}\left[D_x^2 u_\epsilon(t,x)\right] = 0, \quad (t,x) \in [0,1+\epsilon)\times\mathbb{R}^d \\
u_\epsilon(1+\epsilon,x) = \psi(x).
\end{cases} \tag{A.2}$$

By Theorem C.4.5 in Peng Peng (2019), $\exists \beta \in (0,1)$ such that

$$\|u_\epsilon\|_{C^{1+\beta/2,2+\beta}([0,1]\times\mathbb{R}^d)} < \infty.$$

Here $\|\cdot\|_{C^{1+\beta/2,2+\beta}([0,1]\times\mathbb{R}^d)}$ is the Krylov Krylov (1987) norm on $C^{1+\beta/2,2+\beta}([0,1]\times\mathbb{R}^d)$, the set of continuous and suitably differentiable functions on $[0,1]\times\mathbb{R}^d$. This proves Part (1).

Result (c) Part (2) follows directly from the dynamic programming principle (see Yong and Zhou (Yong and Zhou, 1999, Theorem 3.3, Ch. 4)).

For Result (c) Part (3), by Results Parts (1) and (2), and Ito's formula, we have:

$$\begin{aligned}
&\sum_{m=1}^{n} \sup_{x\in\mathbb{R}^d} \left| H^\epsilon_{\frac{m-1}{n}}(x) - H^\epsilon_{\frac{m}{n}}(x) - \frac{1}{n}\sup_{p\in\Gamma}\left[D_x H^\epsilon_{\frac{m}{n}}(x)\cdot p\right]\right| \\
=&\sum_{m=1}^{n} \sup_{x\in\mathbb{R}^d} \left| \sup_{a\in\mathcal{A}(\frac{m-1}{n},\frac{m}{n})} E\left[H^\epsilon_{\frac{m}{n}}\left(x+\int_{\frac{m-1}{n}}^{\frac{m}{n}} a_s ds + \int_{\frac{m-1}{n}}^{\frac{m}{n}} \epsilon \boldsymbol{I}_d dB_s\right)\right] - H^\epsilon_{\frac{m}{n}}(x) - \frac{1}{n}\sup_{p\in\Gamma}\left[D_x H^\epsilon_{\frac{m}{n}}(x)\cdot p\right]\right| \\
=&\sum_{m=1}^{n} \sup_{x\in\mathbb{R}^d} \left| \sup_{a\in\mathcal{A}(\frac{m-1}{n},\frac{m}{n})} E\left[\int_{\frac{m-1}{n}}^{\frac{m}{n}} D_x H^\epsilon_{\frac{m}{n}}\left(x+\int_{\frac{m-1}{n}}^{s} a_s ds + \int_{\frac{m-1}{n}}^{s} \epsilon \boldsymbol{I}_d dB_s\right)\cdot a_s ds\right.\right. \\
&\qquad\qquad \left.\left. + \frac{\epsilon^2}{2}\int_{\frac{m-1}{n}}^{\frac{m}{n}} \operatorname{tr}\left[D_x^2 H^\epsilon_{\frac{m}{n}}\left(x+\int_{\frac{m-1}{n}}^{s} a_s ds + \int_{\frac{m-1}{n}}^{s} \epsilon \boldsymbol{I}_d dB_s\right)\right] ds\right] - \frac{1}{n}\sup_{p\in\Gamma}\left[D_x H^\epsilon_{\frac{m}{n}}(x)\cdot p\right]\right| \\
\le&\frac{C}{n}\sum_{m=1}^{n}\left| \sup_{a\in\mathcal{A}(\frac{m-1}{n},\frac{m}{n})} E\left[\sup_{s\in[\frac{m-1}{n},\frac{m}{n}]}\left(\left\|\int_{\frac{m-1}{n}}^{s} a_s ds\right\| + \left\|\int_{\frac{m-1}{n}}^{s} \epsilon \boldsymbol{I}_d dB_s\right\|\right)\right]\right| + C\epsilon^2 \\
\le&C\left(\frac{\sum_{i=1}^d |\overline{\mu}_i - \underline{\mu}_i|}{2n} + 2\epsilon\sqrt{\frac{d}{n}} + \epsilon^2\right),
\end{aligned}$$

where C is a constant that depends only on the uniform bound of $D_x^2 H_t^\epsilon$. ■

Lemma A.2. *For any $1 \le m \le n$, define a family of functions $\{L_{m,n}\}_{m=1}^n$ by*

$$L_{m,n}(x) := H^\epsilon_{\frac{m}{n}}(x) + \frac{1}{n}\sup_{p\in\Gamma}\left[D_x H^\epsilon_{\frac{m}{n}}(x)\cdot p\right]. \tag{A.3}$$

Let $S_n^{\boldsymbol{\theta}} = S_{n-1}^{\boldsymbol{\theta}} + \boldsymbol{Z}_n^{\boldsymbol{\theta}}$ with $\boldsymbol{Z}_n^{\boldsymbol{\theta}}$ defined in (2.2)*. Then we have:*

$$\lim_{n\to\infty}\sum_{m=1}^{n}\left\{\sup_{\boldsymbol{\theta}\in\Theta} E\left[H^\epsilon_{\frac{m}{n}}\left(\frac{S_m^{\boldsymbol{\theta}}}{n}\right)\right] - \sup_{\boldsymbol{\theta}\in\Theta} E\left[L_{m,n}\left(\frac{S_{m-1}^{\boldsymbol{\theta}}}{n}\right)\right]\right\} = 0. \tag{A.4}$$

Proof of Lemma A.2: It is sufficient to prove the following two results:

$$\lim_{n\to\infty}\sum_{m=1}^{n}\sup_{\boldsymbol{\theta}\in\Theta}E\left[\left|H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_m}{n}\right)-H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)-D_x\,H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\cdot\frac{\boldsymbol{Z}^{\boldsymbol{\theta}}_m}{n}\right|\right]=0, \tag{A.5}$$

$$\sup_{\boldsymbol{\theta}\in\Theta}E\left[H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)+D_x\,H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\cdot\frac{\boldsymbol{Z}^{\boldsymbol{\theta}}_m}{n}\right]=\sup_{\boldsymbol{\theta}\in\Theta}E\left[L_{m,n}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\right]. \tag{A.6}$$

For Result (A.5), by Result (1) of Lemma A.1, there exists a constant $C>0$ such that

$$\sup_{t\in[0,1]}\sup_{x\in\mathbb{R}^d}\sum_{j,k=1}^{d}|\partial^2_{x_jx_k}H^{\epsilon}_t(x)|\le C.$$

It follows from Taylor's expansion that for any $x,y\in\mathbb{R}^d$, and $t\in[0,1]$,

$$|H^{\epsilon}_t(x+y)-H^{\epsilon}_t(x)-D_x\,H^{\epsilon}_t(x)\cdot y|\le C\|y\|^2. \tag{A.7}$$

For any $1\le m\le n$, taking $x=\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}, y=\frac{\boldsymbol{Z}^{\boldsymbol{\theta}}_m}{n}$ in (A.7), we obtain

$$\begin{aligned}&\sum_{m=1}^{n}\sup_{\boldsymbol{\theta}\in\Theta}E\left[\left|H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_m}{n}\right)-H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)-D_x\,H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\cdot\frac{\boldsymbol{Z}^{\boldsymbol{\theta}}_m}{n}\right|\right]\\ \le&\frac{C}{n^2}\sum_{m=1}^{n}\sup_{\boldsymbol{\theta}\in\Theta}E[\|\boldsymbol{Z}^{\boldsymbol{\theta}}_m\|^2]\to 0,\ \text{as } n\to\infty.\end{aligned}$$

Next, for Result (A.6), we have for any $1\le m\le n$,

$$\begin{aligned}&\sup_{\boldsymbol{\theta}\in\Theta}E\left[H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)+D_x\,H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\cdot\frac{\boldsymbol{Z}^{\boldsymbol{\theta}}_m}{n}\right]\\ =&\sup_{\boldsymbol{\theta}\in\Theta}E\left[H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)+\operatorname*{ess\,sup}_{\boldsymbol{\theta}\in\Theta}E\left[D_x\,H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\cdot\frac{\boldsymbol{Z}^{\boldsymbol{\theta}}_m}{n}\middle|\mathcal{H}^{\boldsymbol{\theta}}_{m-1}\right]\right]\\ =&\sup_{\boldsymbol{\theta}\in\Theta}E\left[H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)+\operatorname*{ess\,sup}_{\boldsymbol{\theta}\in\Theta}\frac{1}{n}\sum_{j=1}^{d}\left[\partial_{x_j}H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)E\left[\vartheta_{m,j}X_{m,j}+(1-\vartheta_{m,j})Y_{m,j}|\mathcal{H}^{\boldsymbol{\theta}}_{m-1}\right]\right]\right]\\ =&\sup_{\boldsymbol{\theta}\in\Theta}E\left[H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)+\sum_{j=1}^{d}\left[\frac{\overline{\mu}_j}{n}\partial_{x_j}H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)I\left\{\partial_{x_j}H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\ge 0\right\}\right.\right.\\ &\qquad\left.\left.+\frac{\underline{\mu}_j}{n}\partial_{x_j}H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)I\left\{\partial_{x_j}H^{\epsilon}_{\frac{m}{n}}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)<0\right\}\right]\right]\\ =&\sup_{\boldsymbol{\theta}\in\Theta}E\left[L_{m,n}\left(\frac{S^{\boldsymbol{\theta}}_{m-1}}{n}\right)\right].\end{aligned}$$

Thus we obtain Result (A.6). ■

Proof of Theorem 2.1: Let $\hat{f}$ be a bounded continuous extension of f defined on $\mathbb{R}^d$ (we can assume that $\hat{f}$ is constant outside a compact set $\Gamma' \supset \Gamma_\delta$). For any $\varepsilon > 0$, there exists $\psi \in C_{b,Lip}(\mathbb{R}^d)$ such that $\sup_{x\in\mathbb{R}^d} |\hat{f}(x) - \psi(x)| \leq \varepsilon$.

For any $\epsilon > 0$, let $\{H_t^\epsilon(x)\}_{t\in[0,1]}$ be the functions defined by (A.1) via ψ. It can be checked that

$$\begin{aligned}
&\sup_{\boldsymbol{\theta}\in\Theta} E\left[H_1^\epsilon\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right] - E\left[H_0^\epsilon\left(\frac{\eta}{n}\right)\right] \\
=&\sum_{m=1}^{n}\left\{\sup_{\boldsymbol{\theta}\in\Theta} E\left[H_{\frac{m}{n}}^\epsilon\left(\frac{S_m^{\boldsymbol{\theta}}}{n}\right)\right] - \sup_{\boldsymbol{\theta}\in\Theta} E\left[H_{\frac{m-1}{n}}^\epsilon\left(\frac{S_{m-1}^{\boldsymbol{\theta}}}{n}\right)\right]\right\} \\
=&\sum_{m=1}^{n}\left\{\sup_{\boldsymbol{\theta}\in\Theta} E\left[H_{\frac{m}{n}}^\epsilon\left(\frac{S_m^{\boldsymbol{\theta}}}{n}\right)\right] - \sup_{\boldsymbol{\theta}\in\Theta} E\left[L_{m,n}\left(\frac{S_{m-1}^{\boldsymbol{\theta}}}{n}\right)\right]\right\} \\
&+\sum_{m=1}^{n}\left\{\sup_{\boldsymbol{\theta}\in\Theta} E\left[L_{m,n}\left(\frac{S_{m-1}^{\boldsymbol{\theta}}}{n}\right)\right] - \sup_{\boldsymbol{\theta}\in\Theta} E\left[H_{\frac{m-1}{n}}^\epsilon\left(\frac{S_{m-1}^{\boldsymbol{\theta}}}{n}\right)\right]\right\} \\
=:&\Delta_{n,1} + \Delta_{n,2}.
\end{aligned}$$

By Lemma A.2 we have $|\Delta_{n,1}| \to 0$ as $n \to \infty$. Applying Lemma A.1(3) we have $\limsup_{n\to\infty} |\Delta_{n,2}| \leq C\epsilon^2$, where C is a constant depends only on the uniform bound of $D_x^2 H_t^\epsilon$. Therefore, we have

$$\limsup_{n\to\infty}\left|\sup_{\boldsymbol{\theta}\in\Theta} E\left[H_1^\epsilon\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right] - E\left[H_0^\epsilon\left(\frac{\eta}{n}\right)\right]\right| \leq C\epsilon^2. \tag{A.8}$$

By Result (a) we have:

$$\begin{aligned}
&\left|\max_{x\in\Gamma}\psi(x) - E\left[H_0^\epsilon\left(\frac{\eta}{n}\right)\right]\right| \\
\leq&\left|\max_{x\in\Gamma}\psi(x) - H_0^\epsilon(0)\right| + \left|H_0^\epsilon(0) - E\left[H_0^\epsilon\left(\frac{\eta}{n}\right)\right]\right| \\
=&\left|\sup_{a\in\mathcal{A}(0,1)} E\left[\psi\left(\int_0^1 a_s ds\right)\right] - \sup_{a\in\mathcal{A}(0,1+\epsilon)} E\left[\psi\left(\int_0^{1+\epsilon} a_s ds + \epsilon\int_0^{1+\epsilon} \boldsymbol{I}_d dB_s\right)\right]\right| + \left|H_0^\epsilon(0) - E\left[H_0^\epsilon\left(\frac{\eta}{n}\right)\right]\right| \\
\leq&L_\psi\epsilon(C_\Gamma + \sqrt{1+\epsilon}) + \frac{C_1 E[\|\eta\|]}{n},
\end{aligned}$$

where L_ψ is the Lipschitz constant of ψ, C_1 is a finite constant depending on the uniform bound of

$D_x H_t^\epsilon$ and C_Γ is a finite constant depending on the boundary of Γ. Combine with (A.8), we have

$$\begin{aligned}
&\limsup_{n\to\infty}\left|\sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]-\max_{x\in\Gamma} f(x)\right| \\
\leq&\limsup_{n\to\infty}\left|\sup_{\boldsymbol{\theta}\in\Theta} E\left[\hat{f}\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]-\sup_{\boldsymbol{\theta}\in\Theta} E\left[\psi\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]\right| \\
&+\limsup_{n\to\infty}\left|\sup_{\boldsymbol{\theta}\in\Theta} E\left[\psi\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]-\sup_{\boldsymbol{\theta}\in\Theta} E\left[H_1^\epsilon\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]\right| \\
&+\limsup_{n\to\infty}\left|\sup_{\boldsymbol{\theta}\in\Theta} E\left[H_1^\epsilon\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right]-E\left[H_0^\epsilon\left(\frac{\eta}{n}\right)\right]\right| \\
&+\limsup_{n\to\infty}\left|E\left[H_0^\epsilon\left(\frac{\eta}{n}\right)\right]-\max_{x\in\Gamma}\psi(x)\right| \\
&+\left|\max_{x\in\Gamma}\psi(x)-\max_{x\in\Gamma}\hat{f}(x)\right|+\left|\max_{x\in\Gamma}\hat{f}(x)-\max_{x\in\Gamma} f(x)\right| \\
\leq&2\varepsilon+L_\psi\epsilon(2C_\Gamma+\sqrt{\epsilon}+\sqrt{1+\epsilon})+C\epsilon^2.
\end{aligned}$$

Due to the arbitrariness of ϵ and ε, we have (2.4).

Now we prove (2.5). Recall that $S_n^{\boldsymbol{\theta}}=(S_{n,1}^{\boldsymbol{\theta}},\cdots,S_{n,d}^{\boldsymbol{\theta}})$, and $\Gamma=[\underline{\mu}_1,\overline{\mu}_1]\times\cdots\times[\underline{\mu}_d,\overline{\mu}_d]$. By Theorem 3.1 of Chen Chen (2016), we have the following results (A) and (B):

$$(A)\quad \inf_{\boldsymbol{\theta}\in\Theta} P\left(\underline{\mu}_i\leq\liminf_{n\to\infty}\frac{S_{n,i}^{\boldsymbol{\theta}}}{n}\leq\limsup_{n\to\infty}\frac{S_{n,i}^{\boldsymbol{\theta}}}{n}\leq\overline{\mu}_i, i=1,\cdots,d\right)=1$$

(B) Let $\mathcal{C}\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)$ denote the cluster set of $\frac{S_n^{\boldsymbol{\theta}}}{n}$. For any $\mu\in\Gamma$, we have:

$$\sup_{\boldsymbol{\theta}\in\Theta} P\left(\mu\in\mathcal{C}\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right)=1.$$

By result (A) and the continuity of f, we have

$$\inf_{\boldsymbol{\theta}\in\Theta} P\left(\limsup_{n\to\infty} f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\leq\max_{x\in\Gamma} f(x)\right)=1. \tag{A.9}$$

On the other hand, there exists $x^*\in\Gamma$ such that $f(x^*)=\max_{x\in\Gamma} f(x)$. Combined with result (B), we have:

$$\sup_{\boldsymbol{\theta}\in\Theta} P\left(f(x^*)\in\mathcal{C}\left(f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right)\right)=1 \tag{A.10}$$

then

$$\sup_{\boldsymbol{\theta}\in\Theta} P\left(f(x^*) \leq \limsup_{n\to\infty} f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right) = 1 \tag{A.11}$$

Relations (A.9) and (A.11) together imply result (2.5). ■

Proof of Theorem 2.2: For any $\boldsymbol{\theta} \in \Theta$, set $\tilde{S}_n^{\boldsymbol{\theta}} = \sum_{i=1}^n E\left[\boldsymbol{Z}_i^{\boldsymbol{\theta}} | \mathcal{H}_{i-1}^{\boldsymbol{\theta}}\right]$, then we have $\frac{\tilde{S}_n^{\boldsymbol{\theta}}}{n} \in \Gamma$, P-a.s., which implies that

$$\sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right] - \max_{x\in\Gamma} f(x) \leq \sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right) - f\left(\frac{\tilde{S}_n^{\boldsymbol{\theta}}}{n}\right)\right] \leq \frac{L_f}{n}\left(\sup_{\boldsymbol{\theta}\in\Theta} E\left[\|S_n^{\boldsymbol{\theta}} - \tilde{S}_n^{\boldsymbol{\theta}}\|^2\right]\right)^{\frac{1}{2}}.$$

It can be checked that, for any $\boldsymbol{\theta} \in \Theta$

$$\begin{aligned} E\left[\|S_n^{\boldsymbol{\theta}} - \tilde{S}_n^{\boldsymbol{\theta}}\|^2\right] \leq & E[\|\eta\|^2] + \sum_{j=1}^d \sum_{i=1}^n E\left[(Z_{i,j}^{\boldsymbol{\theta}} - E[Z_{i,j}^{\boldsymbol{\theta}} | \mathcal{H}_{i-1}^{\boldsymbol{\theta}}])^2\right] \\ \leq & E[\|\eta\|^2] + \sum_{j=1}^d \sum_{i=1}^n \max\{Var(X_j), Var(Y_j)\} \\ = & nd\sigma^2 + E[\|\eta\|^2]. \end{aligned}$$

Therefore, we have

$$\sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right] - \max_{x\in\Gamma} f(x) \leq L_f\left(\frac{\sigma\sqrt{d}}{\sqrt{n}} + \frac{\sqrt{E[\|\eta\|^2]}}{n}\right).$$

On the other hand, there exists $x^* = (x_1^*, \cdots, x_d^*) \in \Gamma$ such that $f(x^*) = \max_{x\in\Gamma} f(x)$. Then there exists a strategy $\hat{\boldsymbol{\theta}}$ such that

$$P\left(\hat{\vartheta}_{ij} = 1\right) = \frac{x_j^* - \underline{\mu}_j}{\overline{\mu}_j - \underline{\mu}_j}, \quad P\left(\hat{\vartheta}_{ij} = 0\right) = \frac{\overline{\mu}_j - x_j^*}{\overline{\mu}_j - \underline{\mu}_j}.$$

Therefore,

$$E[Z_{ij}^{\hat{\boldsymbol{\theta}}}] = E\left[E[Z_{ij}^{\hat{\boldsymbol{\theta}}} | \mathcal{H}_{i-1}^{\hat{\boldsymbol{\theta}}}]\right] = E\left[\overline{\mu}_j I\left\{\hat{\vartheta}_{i,j} = 1\right\} + \underline{\mu}_j I\left\{\hat{\vartheta}_{i,j} = 0\right\}\right] = x_j^*$$

which leads to

$$\sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right] - \max_{x\in\Gamma} f(x) \geq E\left[f\left(\frac{S_n^{\hat{\boldsymbol{\theta}}}}{n}\right)\right] - f(x^*) \geq -\frac{L_f}{n}\left(E[\|S_n^{\hat{\boldsymbol{\theta}}} - nx^*\|^2]\right)^{\frac{1}{2}},$$

and

$$\begin{aligned}
E\left[\|S_n^{\hat{\boldsymbol{\theta}}} - nx^*\|^2\right] \leq & E[\|\eta\|^2] + \sum_{j=1}^{d}\sum_{i=1}^{n} E\left[(Z_{i,j}^{\hat{\boldsymbol{\theta}}} - E[Z_{i,j}^{\hat{\boldsymbol{\theta}}}])^2\right] \\
\leq & E[\|\eta\|^2] + \sum_{j=1}^{d}\sum_{i=1}^{n}\left\{\max\{Var(X_j), Var(Y_j)\} + E\left[(E[Z_{i,j}^{\hat{\boldsymbol{\theta}}}|\mathcal{H}_{i-1}^{\hat{\boldsymbol{\theta}}}] - E[Z_{ij}^{\hat{\boldsymbol{\theta}}}])^2\right]\right\} \\
= & nd\sigma^2 + n\sum_{j=1}^{d}\frac{(\overline{\mu}_j - \underline{\mu}_j)^2}{4} + E[\|\eta\|^2].
\end{aligned}$$

Finally, we have

$$\left|\sup_{\boldsymbol{\theta}\in\Theta} E\left[f\left(\frac{S_n^{\boldsymbol{\theta}}}{n}\right)\right] - \max_{x\in\Gamma} f(x)\right| \leq L_f\left(\frac{\sigma\sqrt{d} + \sum_{j=1}^{d}\frac{|\overline{\mu}_j - \underline{\mu}_j|}{2}}{\sqrt{n}} + \frac{\sqrt{E[\|\eta\|^2]}}{n}\right).$$

■

Next, we give the proof of Theorem 2.3. For any $\epsilon > 0$ and $(t,x) \in [0, 1+\epsilon] \times \mathbb{R}^d$, define $u_\epsilon(t,x) = H_t^\epsilon(x)$, where H_t^ϵ is defined by (A.1) with ψ replaced by $\hat{f}$, which is a bounded Lipschitz continuous (with Lipschitz constant L_f) extension of f from Γ_δ to $\mathbb{R}^d$. Then $u_\epsilon(t,x) = H_t^\epsilon(x)$ is the unique solution of the HJB-equation [2.8] and satisfies Lemma A.1.

Lemma A.3. *For any $1 \leq m \leq n$, let functions $\{L_{m,n}\}_{m=1}^n$ be defined by* (A.3)*, which can be equivalently denoted by*

$$L_{m,n}(x) = u_\epsilon\left(\tfrac{m}{n}, x\right) + \tfrac{1}{n}\sup_{p\in\Gamma}\left[D_x u_\epsilon\left(\tfrac{m}{n}, x\right)\cdot p\right].$$

Let the strategies $\{\hat{\boldsymbol{\theta}}^{n,\epsilon}\}_{n\geq 1}$ be defined by (2.9)*. Then we have,*

$$\sum_{m=1}^{n}\left\{E\left[u_\epsilon\left(\frac{m}{n}, \frac{S_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[L_{m,n}\left(\frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right]\right\} \leq \frac{C}{n}\sum_{i=1}^{d}\left(\sigma^2 + \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{4}\right), \tag{A.12}$$

where C is a constant only depending on the uniform bound of $D_x^2 u_\epsilon(t,\cdot)$.

Proof: It is sufficient to prove

$$\begin{aligned}
&\sum_{m=1}^{n} E\left[\left|u_\epsilon\left(\frac{m}{n}, \frac{S_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) - u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) - D_x\, u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) \cdot \frac{\boldsymbol{Z}_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right|\right] \\
\leq &\frac{C}{n}\sum_{i=1}^{d}\left(\sigma^2 + \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{4}\right), \qquad \text{(A.13)} \\
\text{and } &\sum_{m=1}^{n} E\left[u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) + D_x\, u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) \cdot \frac{\boldsymbol{Z}_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n} - L_{m,n}\left(\frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] = 0. \qquad \text{(A.14)}
\end{aligned}$$

In fact, by Lemma A.1 Part (1) with $H_t^\epsilon(x) = u_\epsilon(t,x)$, there exists a constant $C > 0$ such that

$$\sup_{t\in[0,1]}\sup_{x\in\mathbb{R}^d}\sum_{j,k=1}^{d}|\partial^2_{x_j x_k} u_\epsilon(t,x)| \leq C.$$

It follows from Taylor's expansion that for any $x, y \in \mathbb{R}^d$, and $t \in [0,1]$,

$$|u_\epsilon(t, x+y) - u_\epsilon(t,x) - D_x\, u_\epsilon(t,x)\cdot y| \leq C\|y\|^2. \qquad \text{(A.15)}$$

For any $1 \leq m \leq n$, taking $x = \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}, y = \frac{\boldsymbol{Z}_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}$ in (A.15), we obtain (A.13) as follows

$$\begin{aligned}
&\sum_{m=1}^{n} E\left[\left|u_\epsilon\left(\frac{m}{n}, \frac{S_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) - u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) - D_x\, u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) \cdot \frac{\boldsymbol{Z}_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right|\right] \\
\leq &\frac{C}{n^2}\sum_{m=1}^{n} E[\|\boldsymbol{Z}_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}\|^2] \leq \frac{C}{n^2}\sum_{m=1}^{n}\sum_{i=1}^{d} E\left[|\xi_i + \overline{\mu}_i|^2\right] \vee E\left[|\xi_i + \underline{\mu}_i|^2\right] \\
\leq &\frac{C}{n}\sum_{i=1}^{d}\left(\sigma^2 + \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{4}\right).
\end{aligned}$$

On the other hand, we have

$$
\begin{aligned}
&\sum_{m=1}^{n} E\left[u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) + D_x\, u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) \cdot \frac{\boldsymbol{Z}_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right] \\
=&\sum_{m=1}^{n} E\left[u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) + \sum_{j=1}^{d}\left[\partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) E\left[\frac{X_{m,j}}{n} | \mathcal{H}_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}\right] I\left\{\partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) \geq 0\right\}\right.\right. \\
&\qquad\left.\left. + \partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) E\left[\frac{Y_{m,j}}{n} | \mathcal{H}_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}\right] I\left\{\partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) < 0\right\}\right]\right] \\
=&\sum_{m=1}^{n} E\left[u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) + \sum_{j=1}^{d}\left[\frac{\overline{\mu}_j}{n} \partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) I\left\{\partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) \geq 0\right\}\right.\right. \\
&\qquad\left.\left. + \frac{\underline{\mu}_j}{n} \partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) I\left\{\partial_{x_j} u_\epsilon\left(\frac{m}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right) < 0\right\}\right]\right] \\
=&\sum_{m=1}^{n} E\left[L_{m,n}\left(\frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right].
\end{aligned}
$$

Then we have (A.14). The proof is completed. ■

Proof of Theorem 2.3: It can be checked that

$$
\begin{aligned}
&E\left[u_\epsilon\left(1, \frac{S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[u_\epsilon\left(0, \frac{\eta}{n}\right)\right] \\
=&\sum_{m=1}^{n}\left\{E\left[u_\epsilon\left(\frac{m}{n}, \frac{S_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[u_\epsilon\left(\frac{m-1}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right]\right\} \\
=&\sum_{m=1}^{n}\left\{E\left[u_\epsilon\left(\frac{m}{n}, \frac{S_m^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[L_{m,n}\left(\frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right]\right\} + \sum_{m=1}^{n}\left\{E\left[L_{m,n}\left(\frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[u_\epsilon\left(\frac{m-1}{n}, \frac{S_{m-1}^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right]\right\} \\
=:&\hat{\Delta}_n^1 + \hat{\Delta}_n^2.
\end{aligned}
$$

By Lemma A.3 and Lemma A.1 Part (3) with $H_t^\epsilon(x) = u_\epsilon(t,x)$, we have

$$
\left|E\left[u_\epsilon\left(1, \frac{S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[u_\epsilon\left(0, \frac{\eta}{n}\right)\right]\right| \leq \frac{C}{n}\sum_{i=1}^{d}\left(\sigma^2 + \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{4}\right) + C\left(\frac{\sum_{i=1}^{d}|\overline{\mu}_i - \underline{\mu}_i|}{2n} + 2\epsilon\sqrt{\frac{d}{n}} + \epsilon^2\right), \tag{A.16}
$$

where C is a finite constant depending on f.

Next, by Lemma A.1 Result (a) with ψ replaced by $\hat{f}$ or f on Γ we have:

$$\max_{x\in\Gamma} f(x) = \max_{x\in\Gamma} \hat{f}(x) = \sup_{a\in\mathcal{A}(0,1)} E\left[\hat{f}\left(\int_0^1 a_s ds\right)\right].$$

Then we have,

$$\begin{aligned}
&\left|\max_{x\in\Gamma} f(x) - E\left[u_\epsilon\left(0,\frac{\eta}{n}\right)\right]\right| \\
\leq &\left|\max_{x\in\Gamma} f(x) - u_\epsilon(0,0)\right| + \left|E\left[u_\epsilon\left(0,\frac{\eta}{n}\right)\right] - u_\epsilon(0,0)\right| \\
= &\left|\sup_{a\in\mathcal{A}(0,1)} E\left[\hat{f}\left(\int_0^1 a_s ds\right)\right] - \sup_{a\in\mathcal{A}(0,1+\epsilon)} E\left[\hat{f}\left(\int_0^{1+\epsilon} a_s ds + \epsilon\int_0^{1+\epsilon} \boldsymbol{I}_d dB_s\right)\right]\right| + \left|E\left[u_\epsilon\left(0,\frac{\eta}{n}\right)\right] - u_\epsilon(0,0)\right| \\
\leq &L_f\epsilon(C_\Gamma + \sqrt{1+\epsilon}) + \frac{C_1 E[\|\eta\|]}{n},
\end{aligned}$$

where L_f is the Lipschitz constant of $\hat{f}$, C_1 is a finite constant depending on the uniform bound of $D_x u_\epsilon$ and C_Γ is a finite constant depending on the boundary of Γ. Combine with (A.16), we have

$$\begin{aligned}
&\left|E\left[f\left(\frac{S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - \max_{x\in\Gamma} f(x)\right| \\
\leq &\left|E\left[\hat{f}\left(\frac{S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[u_\epsilon\left(1,\frac{S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right]\right| + \left|E\left[u_\epsilon\left(1,\frac{S_n^{\hat{\boldsymbol{\theta}}^{n,\epsilon}}}{n}\right)\right] - E\left[u_\epsilon\left(0,\frac{\eta}{n}\right)\right]\right| + \left|E\left[u_\epsilon\left(0,\frac{\eta}{n}\right)\right] - \max_{x\in\Gamma} f(x)\right| \\
\leq &L_f\epsilon(2C_\Gamma + \sqrt{\epsilon} + \sqrt{1+\epsilon}) + \frac{C_1 E[\|\eta\|]}{n} + C\left(\frac{1}{n}\sum_{i=1}^d\left(\sigma^2 + \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{4}\right) + \frac{\sum_{i=1}^d |\overline{\mu}_i - \underline{\mu}_i|}{2n} + 2\epsilon\sqrt{\frac{d}{n}} + \epsilon^2\right).
\end{aligned}$$

The proof of Theorem 2.3 is completed. ■

Finally, we present the proofs of Theorems 2.4, 2.6 and 2.5. The following lemma can be regarded as a weak form of strategic law for large numbers.

Lemma A.4. *Let $\varphi \in C_b^3(\mathbb{R}^d)$, the space of bounded and three-times continuously differentiable functions with bounded derivatives of all orders less than or equal to three, satisfy the assumptions in Theorem 2.4 with f replaced by φ. Furthermore, we assume that there exists a constant $L > 0$ such that for any $j = 1, \cdots, d$, $|X_j| < L$ and $|Y_j| < L$. Let $N > 0$ be the smallest n such that*

$$\frac{2L}{n+1} < \min_{1\leq i\leq d} \min_{1\leq j\leq c_i^{k_i}} \{b_i^j - c_i^j, c_i^j - b_i^{j-1}\}$$

Consider the strategy $\boldsymbol{\theta}^c = (\boldsymbol{\vartheta}_1^c, \cdots, \boldsymbol{\vartheta}_m^c, \cdots)$*, where* $\boldsymbol{\vartheta}_m^c = (\vartheta_{m1}^c, \cdots, \vartheta_{md}^c)$ *is defined by*

$$\vartheta_{mi}^c = \begin{cases} 1, & \text{if } \partial_{x_i}\varphi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}\right) \geq 0; \\ 0, & \text{otherwise}, \end{cases} \qquad i = 1, \cdots, d. \tag{A.17}$$

where $S_0^{\boldsymbol{\theta}^c}/0$ *is set to be* η*. Then we have*

$$\liminf_{n\to\infty} \left\{ E\left[\varphi\left(\frac{S_n^{\boldsymbol{\theta}^c}}{n}\right)\right] - E\left[\sum_{j_1=1}^{K_1} \cdots \sum_{j_d=1}^{K_d} \varphi\left(c_1^{j_1}, c_2^{j_2}, \cdots, c_d^{j_d}\right) I\left\{\frac{S_N^{\boldsymbol{\theta}^c}}{N} \in \Gamma(j_1, \cdots, j_d)\right\}\right] \right\} \geq 0, \tag{A.18}$$

where $\Gamma(j_1, \cdots, j_d) = (b_1^{j_1-1}, b_1^{j_1}] \times \cdots \times (b_d^{j_d-1}, b_d^{j_d}]$.

Proof of Lemma A.4: For convenience, we only give the proof for the case $d = 2$, it can be proved similarly for the general case.

For any large enough $n > N$, it can be checked that

$$\begin{aligned}
&E\left[\varphi\left(\frac{S_n^{\boldsymbol{\theta}^c}}{n}\right)\right] - E\left[\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2} \varphi\left(c_1^{j_1}, c_2^{j_2}\right) I\left\{\frac{S_N^{\boldsymbol{\theta}^c}}{N} \in \Gamma(j_1, j_2)\right\}\right] \\
&= \sum_{m=1}^{n}\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2} E\left[\left[\varphi\left(\frac{S_m^{\boldsymbol{\theta}^c}}{n} + \frac{n-m}{n}\left(c_1^{j_1}, c_2^{j_2}\right)\right) - \varphi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^c}}{n} + \frac{n-m+1}{n}\left(c_1^{j_1}, c_2^{j_2}\right)\right)\right] I\left\{\frac{S_N^{\boldsymbol{\theta}^c}}{N} \in \Gamma(j_1, j_2)\right\}\right] \\
&=: \sum_{m=1}^{n}\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2} E\left[D_x\,\varphi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^c}}{n} + \frac{n-m+1}{n}\left(c_1^{j_1}, c_2^{j_2}\right)\right) \cdot \frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^c} - \left(c_1^{j_1}, c_2^{j_2}\right)}{n} I\left\{\frac{S_N^{\boldsymbol{\theta}^c}}{N} \in \Gamma(j_1, j_2)\right\}\right] + R_n,
\end{aligned}$$

where $D_x\varphi = (\partial_{x_1}\varphi, \partial_{x_2}\varphi)$ denotes the gradient vector of φ. It follows from Taylor's expansion that for any $x, y \in \mathbb{R}^d$,

$$|\varphi(x+y) - \varphi(x) - D_x\,\varphi(x) \cdot y| \leq C\|y\|^2, \tag{A.19}$$

where C is a constant depend only on the bound of the second derivatives of φ. For any $1 \leq m \leq n$, taking $x = \frac{S_{m-1}^{\boldsymbol{\theta}^c}}{n} + \frac{n-m+1}{n}\left(c_1^{j_1}, c_2^{j_2}\right), y = \frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^c} - \left(c_1^{j_1}, c_2^{j_2}\right)}{n}$ in [A.19], we obtain

$$|R_n| \leq \frac{C_\varphi}{n} E[\|\eta\|] + \frac{C}{n^2}\sum_{m=1}^{n}\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2} E\left[\|\boldsymbol{Z}_m^{\boldsymbol{\theta}^c}\|^2 + \|\left(c_1^{j_1}, c_2^{j_2}\right)\|^2\right] \to 0, \text{ as } n \to \infty,$$

where C_φ is a finite constant depending on the bound of $D_x\varphi$.

On the other hand, for any $N < m \leq n$, if $\frac{S_N^{\boldsymbol{\theta}^c}}{N} \in \Gamma(j_1, j_2)$, then one have $\frac{S_m^{\boldsymbol{\theta}^c}}{m} \in \Gamma(j_1, j_2)$. Set

$\lambda_n^m = \frac{m-1}{n}$, then

$$
\begin{aligned}
&E\left[D_x\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{n}+\tfrac{n-m+1}{n}\left(c_1^{j_1},c_2^{j_2}\right)\right)\cdot\tfrac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^c}-\left(c_1^{j_1},c_2^{j_2}\right)}{n}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\\
=&\sum_{i=1}^{2}E\left[\partial_{x_i}\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{n}+\tfrac{n-m+1}{n}\left(c_1^{j_1},c_2^{j_2}\right)\right)\tfrac{Z_{mi}^{\boldsymbol{\theta}^c}-c_i^{j_i}}{n}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\\
=&\sum_{i=1}^{2}E\left[\partial_{x_i}\varphi\left(\lambda_n^m\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}+(1-\lambda_n^m)\left(c_1^{j_1},c_2^{j_2}\right)\right)\tfrac{X_{m,i}-c_i^{j_i}}{n}I\left\{\partial_{x_i}\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}\right)\geq 0\right\}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\\
&+\sum_{i=1}^{2}E\left[\partial_{x_i}\varphi\left(\lambda_n^m\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}+(1-\lambda_n^m)\left(c_1^{j_1},c_2^{j_2}\right)\right)\tfrac{Y_{m,i}-c_i^{j_i}}{n}I\left\{\partial_{x_i}\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}\right)<0\right\}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\\
=&\sum_{i=1}^{2}E\left[\partial_{x_i}\varphi\left(\lambda_n^m\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}+(1-\lambda_n^m)\left(c_1^{j_1},c_2^{j_2}\right)\right)\tfrac{\overline{\mu}_i-c_i^{j_i}}{n}I\left\{\partial_{x_i}\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}\right)\geq 0\right\}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\\
&+\sum_{i=1}^{2}E\left[\partial_{x_i}\varphi\left(\lambda_n^m\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}+(1-\lambda_n^m)\left(c_1^{j_1},c_2^{j_2}\right)\right)\tfrac{\underline{\mu}_i-c_i^{j_i}}{n}I\left\{\partial_{x_i}\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}\right)<0\right\}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right].
\end{aligned}
$$

Since

$$\left\{\partial_{x_i}\varphi(x)\geq 0, x\in\Gamma(j_1,j_2)\right\}=\left\{x=(x_1,x_2)\in\Gamma(j_1,j_2):x_i\leq c_i^{j_i}\right\}=:M_i^{j_i}$$

Then for any $y\in M_i^{j_i}$ and any $\lambda\in(0,1)$, we have $\lambda y+(1-\lambda)\left(c_1^{j_1},c_2^{j_2}\right)\in M_i^{j_i}$. Therefore we have, for any $j=1,2$,

$$\partial_{x_i}\varphi\left(\lambda_n^m\frac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}+(1-\lambda_n^m)\left(c_1^{j_1},c_2^{j_2}\right)\right)I\left\{\partial_{x_i}\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}\right)\geq 0\right\}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\geq 0$$

and similarly,

$$\partial_{x_i}\varphi\left(\lambda_n^m\frac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}+(1-\lambda_n^m)\left(c_1^{j_1},c_2^{j_2}\right)\right)I\left\{\partial_{x_i}\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{m-1}\right)<0\right\}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\leq 0.$$

And these will lead to, for $N<m\leq n$,

$$E\left[D_x\varphi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^c}}{n}+\frac{n-m+1}{n}\left(c_1^{j_1},c_2^{j_2}\right)\right)\cdot\frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^c}-\left(c_1^{j_1},c_2^{j_2}\right)}{n}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\geq 0.$$

That is,

$$\begin{aligned}
&\liminf_{n\to\infty}\left\{E\left[\varphi\left(\tfrac{S_n^{\boldsymbol{\theta}^c}}{n}\right)\right]-E\left[\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2}\varphi\left(c_1^{j_1},c_2^{j_2}\right)I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\right\}\\
\geq&\liminf_{n\to\infty}\sum_{m=1}^{n}\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2}E\left[D_x\,\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{n}+\tfrac{n-m+1}{n}\left(c_1^{j_1},c_2^{j_2}\right)\right)\cdot\tfrac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^c}-\left(c_1^{j_1},c_2^{j_2}\right)}{n}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]-\lim_{n\to\infty}|R_n|\\
\geq&\liminf_{n\to\infty}\sum_{m=1}^{N}\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2}E\left[D_x\,\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{n}+\tfrac{n-m+1}{n}\left(c_1^{j_1},c_2^{j_2}\right)\right)\cdot\tfrac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^c}-\left(c_1^{j_1},c_2^{j_2}\right)}{n}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]\\
&+\liminf_{n\to\infty}\sum_{m=N+1}^{n}\sum_{j_1=1}^{K_1}\sum_{j_2=1}^{K_2}E\left[D_x\,\varphi\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^c}}{n}+\tfrac{n-m+1}{n}\left(c_1^{j_1},c_2^{j_2}\right)\right)\cdot\tfrac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^c}-\left(c_1^{j_1},c_2^{j_2}\right)}{n}I\left\{\tfrac{S_N^{\boldsymbol{\theta}^c}}{N}\in\Gamma(j_1,j_2)\right\}\right]-\lim_{n\to\infty}|R_n|\\
\geq&0.
\end{aligned}$$

The proof is completed. ■

Proof of Theorem 2.4: Let $\psi\in C_b^3(\mathbb{R}^d)$ satisfy the assumption in Theorem 2.4. Furthermore suppose that for $x\notin\mathcal{M}_\varepsilon$, $\psi(x)\leq 0$, for $x\in\mathcal{M}_\varepsilon$, $0\leq\psi(x)\leq 1$ and $\psi(c_1^{j_1},\cdots,c_d^{j_d})=1$. Then the strategy $\boldsymbol{\theta}^*$ given in [2.11] is same as the strategy $\boldsymbol{\theta}^c$, which is defined by [A.17] via ψ, and we have

$$\begin{aligned}
&\lim_{n\to\infty}P\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\in\mathcal{M}_\varepsilon\right)\\
=&\lim_{n\to\infty}E\left[I\left\{\frac{S_n^{\boldsymbol{\theta}^*}}{n}\in\mathcal{M}_\varepsilon\right\}\right]\\
\geq&\lim_{n\to\infty}E\left[\psi\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right]-1+1\\
=&\liminf_{n\to\infty}\left\{E\left[\psi\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right]-E\left[\sum_{j_1=1}^{K_1}\cdots\sum_{j_d=1}^{K_d}\psi\left(c_1^{j_1},c_2^{j_2},\cdots,c_d^{j_d}\right)I\left\{\frac{S_N^{\boldsymbol{\theta}^*}}{N}\in\Gamma(j_1,\cdots,j_d)\right\}\right]\right\}+1\\
\geq&1,
\end{aligned}$$

where the last inequality is due to Lemma A.4. Then we have $\lim_{n\to\infty}P\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\in\mathcal{M}_\varepsilon\right)=1$. ■

Proof of Theorem 2.5: Suppose x^* is any global maximum point of $f(x)$ on Γ. By the definition of the arms in (2.1), since ξ_j and ξ_j' are bounded random variables, assume that $\max_{1\leq j\leq d}\{|X_j|,|Y_j|\}<L$.

For strategy $\boldsymbol{\theta}$, we have:

$$\begin{aligned}\left\|\frac{S_{n+1}^{\boldsymbol{\theta}}}{n+1}-\frac{S_n^{\boldsymbol{\theta}}}{n}\right\|_\infty &= \frac{1}{n+1}\left\|S_{n+1}^{\boldsymbol{\theta}}-\frac{(n+1)S_n^{\boldsymbol{\theta}}}{n}\right\|_\infty \\ &\le \frac{1}{n+1}\left(\left\|S_{n+1}^{\boldsymbol{\theta}}-S_n^{\boldsymbol{\theta}}\right\|_\infty+\left\|\frac{S_n^{\boldsymbol{\theta}}}{n}\right\|_\infty\right)\le\frac{2L}{n+1}.\end{aligned}$$

This indicates that for each $n\ge m>N=\left\lceil\frac{4L}{\delta_0}\right\rceil$, it follows $\left\|\frac{S_{n+1}^{\boldsymbol{\theta}}}{n+1}-\frac{S_n^{\boldsymbol{\theta}}}{n}\right\|_\infty<\delta_0/2$, which means that when $S_m^{\boldsymbol{\theta}}/m\in B(x^*,\delta_0)$, $S_n^{\boldsymbol{\theta}'}/n$ will not jump out of $B(x^*,\delta_0)$.

Since $f(x)$ has only one maximum point x^* in $B(x^*,\delta_0)$, then according to Theorem 2.4, it follows for any $\varepsilon>0$:

$$\lim_{n\to\infty}P\left(\left\|\frac{S_n^{\boldsymbol{\theta}'}}{n}-x^*\right\|_\infty<\varepsilon\right)=1. \tag{A.20}$$

■

Proof of Theorem 2.6: For any $\boldsymbol{\theta}\in\Theta$, set $\tilde{S}_n^{\boldsymbol{\theta}}=\sum_{i=1}^n E\left[\boldsymbol{Z}_i^{\boldsymbol{\theta}}|\mathcal{H}_{i-1}^{\boldsymbol{\theta}}\right]$, then we have $\frac{\tilde{S}_n^{\boldsymbol{\theta}}}{n}\in\Gamma$, P-a.s., which implies that

$$E\left[f\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right]-\max_{x\in\Gamma}f(x)\le E\left[f\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)-f\left(\frac{\tilde{S}_n^{\boldsymbol{\theta}^*}}{n}\right)\right]\le\frac{L_f}{n}\left(E\left[\|S_n^{\boldsymbol{\theta}^*}-\tilde{S}_n^{\boldsymbol{\theta}^*}\|^2\right]\right)^{\frac{1}{2}}.$$

It can be checked that,

$$\begin{aligned}E\left[\|S_n^{\boldsymbol{\theta}^*}-\tilde{S}_n^{\boldsymbol{\theta}^*}\|^2\right]\le&\sum_{j=1}^d\sum_{i=1}^n E\left[(Z_{ij}^{\boldsymbol{\theta}^*}-E[Z_{ij}^{\boldsymbol{\theta}^*}|\mathcal{H}_{i-1}^{\boldsymbol{\theta}^*}])^2\right]+E[\|\eta\|^2]\\ \le&\sum_{j=1}^d\sum_{i=1}^n\max\{Var(X_j),Var(Y_j)\}+E[\|\eta\|^2]\\ =&nd\sigma^2+E[\|\eta\|^2].\end{aligned}$$

Therefore, we have

$$E\left[f\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right]-f(x^*)\le L_f\left(\frac{\sigma\sqrt{d}}{\sqrt{n}}+\frac{\sqrt{E[\|\eta\|^2]}}{n}\right).$$

On the other hand,

$$
\begin{aligned}
&E\left[f\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right]-f(x^*)\\
=&\sum_{m=1}^{n}E\left[f\left(\frac{S_m^{\boldsymbol{\theta}^*}}{n}+\frac{(n-m)x^*}{n}\right)-f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{(n-m+1)x^*}{n}\right)\right]+E\left[f\left(\frac{\eta}{n}+x^*\right)-f\left(x^*\right)\right]\\
\geq&\sum_{m=1}^{n}E\left[\frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^*}-x^*}{n}\cdot D_xf\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{(n-m+1)x^*}{n}\right)\right]-\frac{K_f}{n^2}\sum_{m=1}^{n}E\left[\left\|\boldsymbol{Z}_m^{\boldsymbol{\theta}^*}-x^*\right\|^2\right]-\frac{L_f}{n}E[\|\eta\|]\\
\geq&-\frac{K_f\left[d\sigma^2+\sum_{j=1}^{d}(\overline{\mu}_j-\underline{\mu}_j)^2\right]}{n}-\frac{L_f}{n}E[\|\eta\|],
\end{aligned}
$$

where $K_f=\max_{i,j}\sup_{x\in\Gamma_\delta}|\partial^2_{x_ix_j}f(x)|$. The last inequality is due to, for any $1\leq m\leq n$, set $\lambda_n^m=\frac{m-1}{n}$, we have

$$
\begin{aligned}
&E\left[D_xf\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{n-m+1}{n}x^*\right)\cdot\frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^*}-x^*}{n}\right]\\
=&\sum_{j=1}^{d}E\left[\partial_{x_j}f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{n-m+1}{n}x^*\right)\frac{Z_{mj}^{\boldsymbol{\theta}^*}-x_j^*}{n}\right]\\
=&\sum_{j=1}^{d}E\left[\partial_{x_j}f\left(\lambda_n^m\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}+(1-\lambda_n^m)x^*\right)\frac{X_{m,j}-x_j^*}{n}I\left\{\partial_{x_j}f\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)\geq 0\right\}\right]\\
&+\sum_{j=1}^{d}E\left[\partial_{x_j}f\left(\lambda_n^m\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}+(1-\lambda_n^m)x^*\right)\frac{Y_{m,j}-x_j^*}{n}I\left\{\partial_{x_j}f\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)<0\right\}\right]\\
=&\sum_{j=1}^{d}E\left[\partial_{x_j}f\left(\lambda_n^m\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}+(1-\lambda_n^m)x^*\right)\frac{\overline{\mu}_j-x_j^*}{n}I\left\{\partial_{x_j}f\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)\geq 0\right\}\right]\\
&+\sum_{j=1}^{d}E\left[\partial_{x_j}f\left(\lambda_n^m\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}+(1-\lambda_n^m)x^*\right)\frac{\underline{\mu}_j-x_j^*}{n}I\left\{\partial_{x_j}f\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)<0\right\}\right].
\end{aligned}
$$

Since

$$
\left\{\partial_{x_j}f\left(x\right)\geq 0\right\}=\left\{x=(x_1,\cdots,x_d)\in\Gamma_\delta:x_j\leq x_j^*\right\}=:M_j
$$

Then for any $y\in M_j$ and any $\lambda\in(0,1)$, we have $\lambda y+(1-\lambda)x^*\in M_j$. Therefore we have, for any $j=1,\cdots,d$,

$$
\partial_{x_j}f\left(\lambda_n^m\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}+(1-\lambda_n^m)x^*\right)I\left\{\partial_{x_j}f\left(\tfrac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)\geq 0\right\}\geq 0
$$

and similarly,

$$\partial_{x_j} f\left(\lambda_n^m \frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1} + (1-\lambda_n^m)x^*\right) I\left\{\partial_{x_j} f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right) < 0\right\} \leq 0.$$

And these will lead to

$$E\left[D_x f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n} + \frac{n-m+1}{n}x^*\right) \cdot \frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^*} - x^*}{n}\right] \geq 0.$$

Therefore, we have

$$\left|E\left[f\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right] - f(x^*)\right| \leq \max\left\{L_f\left(\frac{\sigma\sqrt{d}}{\sqrt{n}} + \frac{\sqrt{E[\|\eta\|^2]}}{n}\right), \frac{K_f\left(d\sigma^2 + \sum_{j=1}^d (\overline{\mu}_j - \underline{\mu}_j)^2\right) + L_f E[\|\eta\|]}{n}\right\}$$

For any $x \in \mathbb{R}^d$, define $\phi(x) = \|x - x^*\|^2$, therefore we have $\phi(x^*) = 0$ and by Taylor's formula

$$\begin{aligned}
&E\left[\left\|\frac{S_n^{\boldsymbol{\theta}^*}}{n} - x^*\right\|^2\right] \\
=&E\left[\phi\left(\frac{S_n^{\boldsymbol{\theta}^*}}{n}\right)\right] - \phi(x^*) \\
=&\sum_{m=1}^n E\left[\phi\left(\frac{S_m^{\boldsymbol{\theta}^*}}{n} + \frac{n-m}{n}x^*\right) - \phi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n} + \frac{n-m+1}{n}x^*\right)\right] + E\left[\phi\left(\frac{\eta}{n} + x^*\right) - \phi\left(x^*\right)\right] \\
=&\sum_{m=1}^n E\left[\frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^*} - x^*}{n} \cdot D_x\phi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n} + \frac{(n-m+1)x^*}{n}\right)\right] + \frac{1}{n^2}\sum_{m=1}^n E\left[\left\|\boldsymbol{Z}_m^{\boldsymbol{\theta}^*} - x^*\right\|^2\right] + \frac{1}{n^2}E[\|\eta\|^2]
\end{aligned}$$

It can be checked that,

$$
\begin{aligned}
& E\left[D_x\,\phi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{n-m+1}{n}x^*\right)\cdot\frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^*}-x^*}{n}\right] \\
=&\sum_{j=1}^{d}E\left[\partial_{x_j}\phi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{n-m+1}{n}x^*\right)\frac{Z_{mj}^{\boldsymbol{\theta}^*}-x_j^*}{n}\right] \\
=&\sum_{j=1}^{d}E\left[2\left(\frac{\sum_{i=1}^{m-1}Z_{ij}^{\boldsymbol{\theta}^*}}{n}-\frac{m-1}{n}x_j^*\right)\frac{Z_{mj}^{\boldsymbol{\theta}^*}-x_j^*}{n}\right] \\
=&\sum_{j=1}^{d}E\left[2\left(\frac{\sum_{i=1}^{m-1}Z_{ij}^{\boldsymbol{\theta}^*}}{n}-\frac{m-1}{n}x_j^*\right)\frac{X_{m,j}-x_j^*}{n}I\left\{\partial_{x_j}f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)\geq 0\right\}\right] \\
&+\sum_{j=1}^{d}E\left[2\left(\frac{\sum_{i=1}^{m-1}Z_{ij}^{\boldsymbol{\theta}^*}}{n}-\frac{m-1}{n}x_j^*\right)\frac{Y_{m,j}-x_j^*}{n}I\left\{\partial_{x_j}f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)<0\right\}\right] \\
=&\sum_{j=1}^{d}E\left[\frac{2(m-1)}{n}\left(\frac{\sum_{i=1}^{m-1}Z_{ij}^{\boldsymbol{\theta}^*}}{m-1}-x_j^*\right)\frac{\overline{\mu}_j-x_j^*}{n}I\left\{\partial_{x_j}f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)\geq 0\right\}\right] \\
&+\sum_{j=1}^{d}E\left[\frac{2(m-1)}{n}\left(\frac{\sum_{i=1}^{m-1}Z_{ij}^{\boldsymbol{\theta}^*}}{m-1}-x_j^*\right)\frac{\underline{\mu}_j-x_j^*}{n}I\left\{\partial_{x_j}f\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{m-1}\right)<0\right\}\right]
\end{aligned}
$$

Since $\underline{\mu}_j\leq x_j^*\leq\overline{\mu}_j$ and

$$
\left\{\partial_{x_j}f\left(x\right)\geq 0\right\}=\left\{x=(x_1,\cdots,x_d)\in\Gamma_\delta:x_j\leq x_j^*\right\},
$$

we have

$$
E\left[D_x\,\phi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{n-m+1}{n}x^*\right)\cdot\frac{\boldsymbol{Z}_m^{\boldsymbol{\theta}^*}-x^*}{n}\right]\leq 0,
$$

and then

$$
\begin{aligned}
E\left[\left\|\frac{S_n^{\boldsymbol{\theta}^*}}{n}-x^*\right\|^2\right]=&\sum_{m=1}^{n}E\left[\frac{Z_m^{\boldsymbol{\theta}^*}-x^*}{n}\cdot D_x\phi\left(\frac{S_{m-1}^{\boldsymbol{\theta}^*}}{n}+\frac{(n-m+1)x^*}{n}\right)\right]+\frac{1}{n^2}\sum_{m=1}^{n}E\left[\left\|Z_m^{\boldsymbol{\theta}^*}-x^*\right\|^2\right] \\
\leq&\frac{d\sigma^2+\sum_{j=1}^{d}(\overline{\mu}_j-\underline{\mu}_j)^2}{n}+\frac{E[\|\eta\|^2]}{n^2}.
\end{aligned}
$$

■

B Practical Implementation of the SMCO Algorithm

In this subsection, we describe in more details how we practically implement Algorithm 1 in R, and introduce three versions of our algorithms, namely "SMCO", "SMCO-R", and "SMCO-BR", which we use for the numerical experiments in Section 3.

For the practical implementation of Algorithm 1, which requires the evaluation of the gradient of the function, we compute the finite differences place of the gradients. We adopt this approach for two reasons. First, analytical gradient information is usually hard, or even impossible, to obtain in complex and/or black-box models. Second, we observe that many standard optimization packages in R, even if gradient based, do not require user specification of the gradient, and we follow this practice to: (i) make the comparisons with other optimization algorithms in R relatively fair, and (ii) maximize user friendliness of our algorithm as other algorithms do.

Motivated by the Strategic Law of Large Numbers, we adopt the following step-adaptive approach to calculate the finite differences. Specifically, at iteration n, we define the step size in the j-th coordinate $h_{n,j} := (\overline{\mu}_j - \underline{\mu}_j)/(n+1)$ and compute the following finite difference $\Delta_{x_j} f(\hat{x}_n) := f(\hat{x}_n + h_{n,j} \cdot e_j) - f(\hat{x}_n - h_{n,j} \cdot e_j)$, where e_j denotes the j-th elementary vector. We then use the sign information $\Delta_{x_j} f(\hat{x}_n) \geq 0$ in place of $\partial_{x_j} f(\hat{x}_n) \geq 0$ in Algorithm 1. Observe that $\Delta_{x_j} f(x) \geq 0$ is equivalent to $\Delta_{x_j} f(x)/(2h_{n,j}) \geq 0$, and that $\Delta_{x_j} f(x)/(2h_{n,j}) \to \partial_{x_j} f(x)$ as $n \to \infty$. Hence, the finite difference calculation above is a valid approximation of the gradient, establishing the coherence of our practical implementation with the gradient-based theoretical algorithm.

We now provide a heuristic explanation for our particular choice of step size h_n, which is motivated by our Strategic Law of Large Numbers. Specifically, under the Strategic Law of Large Numbers, $\hat{x}_n$ takes the form of a running average, and the update from $\hat{x}_n$ to $\hat{x}_{n+1}$ can be rewritten as $\hat{x}_{n+1} = \frac{n}{n+1}\hat{x}_n + \frac{1}{n+1}\boldsymbol{Z}_{n+1}^{\theta^*}$. Recall that $\boldsymbol{Z}_{n+1}^{\theta^*}$ is drawn from two fixed distributions concentrated near the upper/lower bounds $\overline{\mu}_j$ or $\underline{\mu}_j$ in each coordinate, and thus the step size of the update from $\hat{x}_n$ to $\hat{x}_{n+1}$ is of the order $(n+1)^{-1}$. Since the updating of $\hat{x}_n$ is guided by the (sign) information in the finite difference calculation, it is natural to set the step size h_n for the finite difference calculation to be of the same rate $(n+1)^{-1}$ as the step size for the updating of $\hat{x}_n$.

We now describe the default choice of hyperparameters for our SMCO algorithms. For the distributions of $X_j = \overline{\mu}_j + \xi_j$, $Y_j = \underline{\mu}_j + \xi'_j$ as introduced in (2.1), we set $\xi_j, \xi'_j \sim U[-\delta^*, \delta^*] \cdot (\overline{\mu}_j - \underline{\mu}_j)$ with default $\delta^* = 0.05$, i.e., 5% of the length of the domain in each coordinate. We set the default

tolerance level of our algorithms to be $1e-6$, the maximum number of iterations to be 500. By default, we generate the starting point(s) for our SMCO algorithms $\hat{x}_0 \sim \eta$ with η being the uniform distribution over the domain. See Appendix D and E for sensitivity checks with respect to these hyperparameter configurations.

C Details on Comparison-Group Algorithms

Algorithm Group I: Six Local Optimizers

This set consists of six gradient-based or derivative-free optimization algorithms that are designed to converge to a local optimum starting from a given starting point.

- GD: Gradient Descent. We implement this algorithm in R with learning rate 0.1, momentum 0.9, max number of iterations 1000 and tolerance 1e-6.
- L-BFGS: As implemented in R command `optim` through method "L-BFGS-B" in R package `stats` (version 4.4.2), with default hyperparameter values.
- SignGD: Sign Gradient Descent by Bernstein et al. (2018). We implement this in R with learning rate 0.1, learning decay rate 0.995, max number of iterations 1000 and tolerance 1e-6.
- SPSA: Simultaneous Perturbation Stochastic Approximation by Spall (1992). The tuning parameters of SPSA are set following Spall (1998) and Guo and Fu (2022): $A = 50, a = 0.1, \alpha = 0.602, \gamma = 0.101$,$c$ =1e-3, and tolerance 1e-7.
- ADAM: Adaptive Moment Estimation method by Kingma and Ba (2014), as implemented in the R package `optimg` (version 0.1.2), with default hyperparameter values.
- BOBYQA: Bound Optimization by Quadratic Approximation by Powell (2009), as implemented in the R package `nloptr` (version 2.1.1), with default hyperparameter values.

Algorithm Group II: Six Global Optimizers

This set consists of six meta-heuristic global optimization algorithms. Since these algorithms are explicitly designed for global optimization (and many already implement multiple starting points from within), we do not pass any generated starting points to these algorithms whenever applicable.[6]

[6] A few of these global algorithms nevertheless require a user-designated starting point, in which case we pass

- GenSA: Generalized Simulated Annealing implemented in the R package `GenSA` (version 1.1.14.1), with default hyperparameter values.[7]
- SA: Simulated Annealing implement in the R package `SA` (version 1.0-9), with default hyperparameter values.
- DEoptim: Differential Evolution implemented in the R package `DEoptim` (version 2.2-8), with default hyperparameter values.
- STOGO: Stochastic Global Optimization as implemented in the R package `nloptr` (version 2.1.1), with default hyperparameter values.
- GA: Genetic Algorithms as implemented in the R package `GA` (version 3.2.4), with default hyperparameter values.
- PSO: Particle Swarm Optimizer as implemented in the R package `PSO` (version 1.0.4, with default hyperparameter values.[8]

Remark C.1 (About C++)**.** *Our algorithms SMCO/SMCO-R/SMCO-BR are fully written in R without (explicit) use of C++. Translating the key functions in our algorithms into C++ and calling these functions in R through the* ***`Rcpp`*** *package are likely to significantly increase the running speed of our algorithms. In fact, the documentation of some optimization algorithms, such as* ***`nloptr`*** *and* ***`GenSA`****, explicitly state that their core functions/algorithms are coded in C++ for faster execution. Hence, one should keep this difference in mind when comparing the running speed between our algorithms and those algorithms having exploited C++.*

D Supplemental Numerical Results on Deterministic Test Functions

This section contains additional numerical results on the four deterministic test functions (Rastrigin, Griewank, Ackley, Michaelwicz) considered in the main text, including robustness checks across an array of test configurations and hyperparameter settings.

a start point randomly chosen from the set of multiple starting points generated for our SMCO and the Group I algorithms.

[7] Another R implementation for simulated annealing is availble through the `optim` command via method "`SANN`". However, initial tests of GenSA versus optim-SANN reveals that `optim-SANN` performs much worse than our algorithms and GenSA, and is thus removed from our comparison.

[8] Another R package for particle swarm optimization, namely `hydroPSO`, was removed from CRAN and no longer available on R 4.4.2.

Tables 4 replicates the results in Table 1 for dimension $\boldsymbol{d = 1}$, further confirming that our SMCO algorithms perform drastically better than all Group I algorithms, when all are run from a single common randomly generated starting point.

Tables 5-8 report results for the four test functions for dimensions $\boldsymbol{d = 2}$, **10**, **20**, **and 50**, in which our SMCO algorithms and all Group I algorithms are run with ($10\sqrt{d} \approx 14$, 32, 45, and 71) uniformly drawn starting points, respectively. [9]. Again, our (multi-start) SMCO algorithms demonstrate robust performance across almost all test configurations, with speed and accuracy levels that compare competitively, and in fact often favorably, to Group II (global) algorithms. Not very surprisingly, all Group I (local) algorithms, even with the use of multiple starting points, tend to suffer from very large errors in most test configurations, given the challenging landscapes of the four test functions in the moderate dimension (say, $d = 10$). This demonstrates that the use of multiple starting points alone cannot explain the good and robust performance of our SMCO algorithms. Among Group II (global) algorithms, GenSA appears to be the only one that shows similar level of robustness in its performance across the configurations. That said, it still yields noticeably larger errors (in terms of RMSE and AE99) than ours, say, in the minimization of Rastrigin and Ackley functions. It is worth noting again that our SMCO algorithms are particularly stable in terms of the 99th error percentile: certain global optimization algorithms may outperform ours in terms of RMSE under a particular specification, but its 99th error percentile becomes notably larger than ours, showing that our algorithm is less sensitive to the randomness inherent in the starting points and the optimization algorithms.

Table 9 replicates Table 2 on optimizaiton in the high-dimensional setting $d = 200$, but uses **diagonal starting points** rather than uniformly drawn ones. Specifically, under the multiple starting points are generated deterministically as an equally spaced grid of points on the hyper-diagonal of the rectangular parameter space. **Table 10** further compares our SMCO algorithms with Group II algorithms in **500-dimensional** settings. Observe that our SMCO-R/-BR algorithms perform consistently well across most of the test cases, with the maximization of the Rastrigin function being the only exception where our algorithms yield significant errors.

Table 11 analyzes the sensitivity of our results with respect to the magnitude of the asymmetrization transformation. In the main text, we randomly push out the support by 25% ±5% on one side, and by 50% ±10% on the other side. We consider two additional variants: Variant 1

[9]These results were generated in a previous using 200 maximum iterations and 1e-8 tolerance levels for our SMCO/-R/-BR algorithms.

pushes out both sides by 25% ±5%, resulting in a smaller and more symmetric domain; Variant 2 pushes out one side by 25% ±5% but 75% ±10% on the other, resulting in a larger and more asymmetric domain. Note that larger domains make the optimization problem harder for all optimization algorithms. **Table 12** analyzes the sensitivity of our results with respect to the type of staring points (uniform versus diagonal) and the maximum iterations (for our SMCO algorithms only). **Table 13** analyzes the sensitivity of our results with respect to the tolerance level. Overall, the pattern of the performance comparison of our SMCO algorithms with other algorithms is quite stable.

Table 14 reports results on the comparison of our SMCO algorithms with three best-performing global optimization algorithms in a very high-dimensional setting ($d = 1000$). In particular, the rows labeled GenSA* and PSO* contain results from GenSA and PSO with their hyperparameter adjusted so that the running time of GenSA* and PSO* are on the same order of magnitude as those of our SMCO algorithms (with parallelization). Since the running time of DEoptim under the default setting is about twice of SMCO algorithms, we view it as being on the same order of magnitude and did not tune DEoptim hyperparameters. The overall conclusion of this table is that our SMCO algorithms are able to deliver better results than GenSA*, PSO* and DEoptim when their running times are similar.

E Supplemental Numerical Results on Maximum Score Estimation and Empirical Welfare Maximization

Table 15 replicates Table 3 on the two random test functions (maximum score estimation and empirical welfare maximization) **using 10 replications only** (instead of $500/d = 100, 50, 25, 10$ replications as in Table 3 for $d = 5, 10, 20, 50$). This replication serves to show that the results in Table 15 obtained using only 10 replications are very similar to those in Table 3 based on more replications. Due to constraints on computation time and resources, we report most results on higher dimensions in the main text and the appendix based on 10 replications. While more replications generally yield more accurate performance measures given the randomness in the optimization algorithms, we hope that the similarity of Table 3 and Table 15 suggests that our numerical results are fairly stable with respect to the number of replications used.

Table 16 replicates Table 3 using diagonal starting points instead of uniform ones for SMCO/-

Table 4: Dimension $d = 1$, Uniform Single-Start

Test Function		**Rastrigin**					**Griewank**				
Performance		RMSE	AE50	AE95	AE99	Time	RMSE	AE50	AE95	AE99	Time
Max	**Best Value**	**130.8169**					**427.1695**				
	GD	61.42	2.94	121.06	121.06	1.2	332.20	367.38	424.88	425.11	19
	SignGD	83.58	90.46	110.57	110.57	18	332.20	367.38	424.88	425.11	15
	SPSA	82.41	90.46	110.57	110.57	1.6	332.31	367.61	424.93	425.16	14
	L-BFGS	58.60	2.94	121.06	121.06	0.19	330.66	365.86	424.88	425.11	0.26
	BOBYQA	48.85	2.94	110.57	121.06	1.3	170.70	5.68E-14	409.13	409.77	1.4
	SMCO	0.302	0.135	0.659	0.716	3.8	3.67	3.21	7.49	8.49	4.3
	SMCO-R	0.0188	3.96e-7	1.66e-6	0.0714	4.1	0.185	0.012	0.085	0.150	4.6
	SMCO-BR	0.438	7.12e-7	5.86e-6	2.94	4.0	0.489	0.024	0.117	0.178	4.5
Min	**Best Value**	**0**					**0**				
	GD	14.43	5.61	35.82	52.44	4.2	174.83	73.81	367.63	414.60	19
	SignGD	52.15	15.92	100.54	120.38	19	174.83	73.81	367.63	414.60	15
	ADAM	14.95	0.995	24.87	80.59	2.1	174.83	73.81	367.63	414.60	1.4
	SPSA	42.72	15.92	80.59	120.38	2.0	174.91	73.83	367.92	414.60	16
	L-BFGS	39.11	3.98	99.49	120.38	0.25	166.63	64.72	356.30	414.60	0.26
	BOBYQA	33.65	3.98	99.49	120.38	1.3	26.76	0.0395	0.799	3.29	1.4
	SMCO	1.73	1.08	4.06	4.27	4.1	8.11	7.92	10.44	11.55	4.4
	SMCO-R	0.594	1.43e-6	0.995	0.995	4.3	0.0159	0.00986	0.0395	0.0400	4.9
	SMCO-BR	0.580	2.89e-6	0.995	0.995	4.3	0.0203	0.00987	0.0395	0.0888	4.7

Based on 500 replications. "Time" is reported in 10^{-3} second. SMCO: max iter = 200, tol = 1e-8.

Table 5: Dimension $d = 2$, Uniform Multi-Start

Test Function		**Rastrigin**			**Griewank**			**Ackley**			**Michalewicz**		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Max	**Best**	**213.5824**			**622.3434**			**22.35029**			**1.21406**		
	GD	1.99e-13	1.99e-13	0.048	192.52	370.71	0.55	2.56e-3	8.79e-3	0.11	0.220	0.413	0.18
	SignGD	58.70	100.71	0.44	215.45	370.71	0.44	2.46e-3	8.48e-3	0.12	0.180	0.413	0.39
	SPSA	1.44	1.99e-13	0.058	221.90	370.80	0.24	2.98e-3	0.0111	0.12	0.263	0.413	0.027
	L-BFGS	6.48	16.50	0.003	117.35	303.68	0.007	2.73e-3	0.0112	0.28	0.156	0.413	0.008
	BOBYQA	1.73e-13	1.99e-13	0.016	2.87e-13	3.41e-13	0.020	1.36e-3	5.01e-3	0.51	0.110	0.413	0.037
	SMCO	3.31	10.34	0.076	0.754	2.21	0.11	0.148	0.240	0.11	8.73e-4	2.85e-3	0.12
	SMCO-R	0.737	1.87	0.077	3.41e-13	3.41e-13	0.11	2.35e-3	6.70e-3	0.12	1.84e-7	4.82e-7	0.12
	SMCO-BR	0.867	4.24	0.079	4.00e-3	3.41e-13	0.12	1.73e-3	4.82e-3	0.12	3.24e-7	9.48e-7	0.12
	GenSA	1.99e-13	1.99e-13	0.24	3.41e-13	3.41e-13	0.26	1.66e-4	5.07e-4	0.35	2.21e-16	2.22e-16	0.50
	SA	2.76	10.29	0.010	0.0465	0.0684	0.015	0.00612	0.0141	0.01	0.0580	0.186	0.015
	DEoptim	8.19	20.30	0.024	0.0343	0.0529	0.034	0.00154	4.57e-3	0.04	0.0309	0.214	0.036
	CMAES	61.51	168.10	0.050	369.77	616.53	0.074	0.390	1.67	0.17	0.340	1.19	0.14
	STOGO	10.25	10.25	0.32	4.34	4.34	0.47	8.51e-5	8.51e-5	0.48	6.17e-10	6.17e-10	0.49
	GA	13.38	32.36	0.11	17.80	33.72	0.15	0.00607	0.0120	0.14	0.0754	0.413	0.14
	PSO	5.48	1.99e-13	0.25	3.41e-13	3.41e-13	0.35	7.10e-4	2.49e-3	0.34	0.119	0.315	0.34
Min	**Best**	**0**			**0**			**4.440892e-16**			**-1.995552**		
	GD	5.42	12.70	0.14	16.19	45.56	0.41	14.00	19.54	0.26	1.11	1.51	0.16
	SignGD	5.31	18.92	0.45	16.20	45.56	0.34	14.69	19.51	0.44	1.02	1.15	0.30
	ADAM	2.03	4.97	0.024	7.53	25.89	0.034	14.55	19.51	0.027	0.72	1.15	0.053
	SPSA	1.07	3.98	0.095	16.43	45.82	0.18	14.70	19.53	0.006	1.18	1.63	0.074
	L-BFGS	4.42	12.93	0.004	9.63	35.71	0.005	12.06	19.51	0.005	1.02	1.15	0.006
	BOBYQA	1.42	4.49	0.016	0.0151	0.0531	0.017	1.15	4.50	0.017	1.01	1.15	0.031
	SMCO	2.92	5.03	0.080	0.560	0.909	0.090	0.322	0.872	0.092	1.14	1.16	0.10
	SMCO-R	0.563	1.00	0.084	0.00784	0.0219	0.093	0.00308	0.00670	0.097	1.00	1.00	0.10
	SMCO-BR	0.482	1.00	0.085	0.0108	0.0300	0.095	0.00436	0.00883	0.098	1.00	1.00	0.094
	GenSA	0.350	0.995	0.23	0.0109	0.0296	0.24	0	0	0.60	1.00	1.00	0.30
	SA	1.77	4.27	0.007	0.526	1.18	0.008	3.95	6.79	0.008	1.00	1.01	0.012
	DEoptim	0.192	0.995	0.022	0.00847	0.0219	0.025	0	0	0.026	1.00	1.00	0.027
	CMAES	11.66	24.87	0.10	78.72	240.00	0.14	17.10	20.70	0.13	1.26	2.00	0.085
	STOGO	3.75e-5	3.75e-5	0.30	3.41e-10	3.41e-10	0.34	0.00376	0.00376	0.35	1.00	1.00	0.37
	GA	0.808	1.99	0.11	0.0724	0.241	0.11	0.311	1.86	0.11	1.00	1.01	0.11
	PSO	0.623	1.50	0.25	0.00377	0.00986	0.26	0	0	0.26	1.00	1.00	0.26

Based on 250 replications. 14 starting points. SMCO: max iter = 200, tol = 1e-8.

Table 6: Dimension $d = 10$, Uniform Multi-Start

Test Function		**Rastrigin**			**Griewank**			**Ackley**			**Michalewicz**		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
	Best Value	**1232.878**			**3771.88**			**22.3502**			**8.051224**		
Max	GD	89.17	184.95	1.63	966.68	1,459.17	4.0	2.51e-3	5.09e-3	2.9	4.81	5.81	2.8
	SignGD	504.36	596.61	3.49	1918.83	2,314.83	4.0	2.31e-3	5.15e-3	3.1	4.51	5.57	0.45
	SPSA	108.40	159.60	0.52	1988.77	2,372.31	0.19	0.0257	0.0491	0.09	5.61	6.81	0.11
	L-BFGS	144.81	273.17	0.057	72.10	311.97	0.14	2.22e-3	4.77e-3	0.04	3.14	4.20	0.28
	BOBYQA	135.30	278.02	0.16	63.74	306.40	0.04	2.03e-3	4.73e-3	0.18	2.97	3.98	0.42
	SMCO	84.15	167.17	0.63	12.93	18.28	0.72	0.150	0.179	0.76	4.38	4.88	0.81
	SMCO-R	52.58	92.77	0.64	16.91	23.49	0.74	7.17e-3	0.0163	0.76	3.26	4.05	0.83
	SMCO-BR	57.10	110.34	0.65	37.52	53.17	0.75	5.51e-3	9.52e-3	0.77	3.15	3.95	0.84
	GenSA	33.81	79.10	0.91	4.55e-13	4.55e-13	0.84	2.39e-3	5.42e-3	0.93	2.16	3.44	1.65
	SA	70.63	204.01	0.016	79.76	385.30	0.02	0.150	0.205	0.02	4.97	5.58	0.02
	DEoptim	18.34	29.17	0.11	1.48	2.18	0.13	0.085	0.118	0.13	3.54	4.07	0.14
	CMAES	117.68	417.88	1.93	593.59	1434.37	0.16	0.274	0.551	1.28	3.67	6.08	1.93
	STOGO	320.71	320.71	1.12	Error	Error	1.23	0	0	1.30	2.88	2.88	1.40
	GA	301.36	420.60	0.21	677.92	890.64	0.21	0.122	0.189	0.22	4.28	5.25	0.22
	PSO	78.35	215.34	0.37	297.61	748.59	0.37	0.070	0.111	0.38	2.37	3.86	0.39
	Best Value	**1.989918**			**0**			**4.440892e-16**			**-8.003921**		
Min	GD	55.16	95.33	2.06	142.89	240.02	3.8	19.58	19.92	2.2	4.42	5.45	2.78
	SignGD	109.19	184.16	3.35	354.16	611.64	3.8	19.58	19.92	3.4	4.49	5.50	0.47
	ADAM	90.30	149.75	0.31	2.84e-13	4.73e-13	0.95	19.58	19.92	0.27	3.30	4.39	1.2
	SPSA	82.54	103.80	0.49	382.48	645.54	0.18	19.61	19.94	0.08	5.56	6.56	0.11
	L-BFGS	73.12	119.39	0.056	1.92e-03	7.40e-3	0.15	19.58	19.92	0.05	3.06	4.02	0.26
	BOBYQA	69.45	130.30	0.16	7.16e-10	1.97e-9	0.22	18.79	19.91	0.18	2.89	3.94	0.42
	SMCO	37.54	46.59	0.68	0.968	1.03	0.78	3.08	3.53	0.80	3.88	4.43	0.87
	SMCO-R	15.28	29.86	0.69	0.175	0.235	0.79	0.0955	0.0850	0.81	3.05	3.95	0.89
	SMCO-BR	16.86	29.64	0.70	0.203	0.267	0.80	0.919	1.67	0.82	3.02	3.81	0.90
	GenSA	23.48	45.28	0.83	5.44e-3	0.0246	1.3	1.66	3.22	1.0	2.14	3.35	1.6
	SA	106.34	149.25	0.009	55.55	118.28	0.01	18.07	20.68	0.01	4.88	5.58	0.02
	DEoptim	25.32	33.57	0.11	0.367	0.548	0.12	3.60	6.31	0.13	3.48	4.15	0.14
	CMAES	45.91	116.10	1.73	23.41	91.16	1.7	20.33	21.60	1.30	3.87	6.32	2.0
	STOGO	68.65	68.65	1.07	6.29e-4	6.29e-4	1.2	18.54	18.54	1.2	3.13	3.13	1.4
	GA	49.94	81.75	0.22	12.41	24.85	0.22	14.34	20.54	0.22	4.29	5.50	0.23
	PSO	13.68	33.15	0.36	0.132	0.487	0.37	2.20	13.02	0.37	2.31	3.65	0.40

Based on 250 replications. 32 starting points. SMCO: max iter = 200, tol = 1e-8.

Table 7: Dimension $d = 20$, Uniform Multi-Start

Test Function		**Rastrigin**			**Griewank**			**Ackley**			**Michalewicz**		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
	Best	**2417.617**			**7240.209**			**22.34938**			**14.37489**		
Max	GD	253.35	399.29	6.7	2276.85	2,858.55	12	3.54e-3	6.57e-3	7.3	8.67	10.08	10.7
	SignGD	1,113.38	1,239.69	10	4,056.59	4,472.41	12	3.51e-3	6.44e-3	11.5	8.48	9.67	1.6
	SPSA	638.42	767.70	0.86	4,184.55	4,596.13	0.32	0.0908	0.130	0.15	11.70	12.71	1.0
	L-BFGS	445.62	614.87	0.23	279.33	647.21	1.1	3.47e-3	6.44e-3	0.14	6.57	7.88	1.6
	BOBYQA	747.83	987.66	1.1	277.67	627.07	0.086	3.46e-3	6.58e-3	1.2	6.25	7.66	1.3
	SMCO	632.59	731.91	1.9	30.34	37.95	2.3	0.174	0.196	2.4	8.45	9.06	2.6
	SMCO-R	353.79	433.02	1.91	48.23	58.73	2.4	8.89e-3	0.0159	2.4	6.89	7.68	2.7
	SMCO-BR	367.18	446.75	1.9	101.57	121.98	2.4	8.68e-3	0.0137	2.4	6.72	7.58	2.7
	GenSA	219.27	410.31	2.1	9.09e-13	9.09e-13	2.2	4.77e-3	9.91e-3	2.3	4.96	6.83	4.8
	SA	341.55	540.56	0.02	394.09	935.70	0.022	0.252	0.307	0.023	10.19	11.11	0.028
	DEoptim	252.20	299.05	0.26	138.09	168.08	0.31	0.174	0.208	0.31	8.30	9.02	0.35
	CMAES	230.09	636.56	9.7	758.94	1,863.78	0.64	0.360	0.595	4.2	6.38	10.80	8.9
	STOGO	1,385.39	1,385.39	2.3	ERROR	ERROR	2.8	6.19e-3	6.19e-3	2.9	7.65	7.65	3.1
	GA	988.53	1,218.89	0.24	2,134.93	2,567.87	0.25	0.205	0.291	0.26	8.70	10.23	0.25
	PSO	266.43	550.05	0.46	829.96	1,877.95	0.49	0.162	0.217	0.52	6.92	8.57	0.53
	Best	**10.94455**			**2.220446e-16**			**2.586065e-11**			**-14.48265**		
Min	GD	134.67	206.36	6.0	377.08	492.81	11	19.82	19.93	5.4	8.36	9.77	9.9
	SignGD	274.76	370.22	9.3	951.60	1,253.15	11	19.82	19.93	11	8.55	9.93	1.4
	ADAM	244.08	331.84	0.53	6.09e-13	8.84e-13	2.6	19.82	19.93	0.51	7.14	8.62	6.2
	SPSA	248.98	304.39	0.79	1,017.81	1330.50	0.30	19.87	19.93	0.20	11.75	12.85	1.0
	L-BFGS	190.98	260.00	0.18	1.27e-10	2.68e-10	0.36	19.82	19.93	0.15	6.31	8.00	1.4
	BOBYQA	236.82	324.35	1.1	7.67e-9	2.20e-8	1.2	19.81	19.93	1.1	6.27	7.59	1.2
	SMCO	85.32	101.94	1.9	0.992	1.02	3.5	5.82	19.92	3.5	7.16	7.61	3.9
	SMCO-R	63.52	98.04	1.9	0.340	0.423	3.5	2.24	2.34	3.5	6.53	7.04	4.0
	SMCO-BR	60.01	92.56	2.0	0.414	0.510	3.5	1.83	2.34	3.5	6.30	7.03	4.0
	GenSA	74.49	130.24	1.9	9.31e-13	2.02e-12	2.8	11.53	19.93	2.1	4.88	7.00	4.1
	SA	314.88	399.62	0.01	274.79	498.07	0.009	20.88	21.09	0.021	10.32	11.13	0.026
	DEoptim	108.35	127.56	0.24	1.05	1.07	0.28	14.93	18.40	0.28	8.38	9.14	0.32
	CMAES	90.65	233.08	5.4	48.11	147.13	4.6	20.48	21.64	4.2	6.52	11.20	8.5
	STOGO	174.12	174.12	2.1	2.94e-6	2.94e-6	2.6	19.34	19.34	2.6	8.46	8.46	2.9
	GA	171.04	236.88	0.24	90.23	139.16	0.25	20.05	20.90	0.25	8.81	10.32	0.25
	PSO	55.44	104.09	0.43	0.018	0.055	0.45	3.63	20.77	0.46	7.27	8.79	0.50

Based on 250 replications. SMCO & Group I: 45 starting points. SMCO: max iter = 200, tol = 1e-8.

Table 8: Dimension $d = 50$, Uniform Multi-Start

Test Function		**Rastrigin**			**Griewank**			**Ackley**			**Michalewicz**		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Max	**Best**	**5860.922**			**18061.87**			**22.34741**			**25.11029**		
	GD	615.25	868.45	37	6,867.83	7634.46	59	4.13e-3	7.06e-3	22	12.83	15.05	56
	SignGD	2,879.52	3059.61	49	10,992.09	11,651.92	59	4.18e-3	7.10e-3	56	17.50	18.76	7.7
	SPSA	2,533.16	2758.01	1.9	11,286.52	11,936.42	0.95	0.236	0.282	0.28	20.91	22.53	2.7
	L-BFGS	1,288.07	1547.96	1.7	1,280.51	1,816.09	12	4.13e-3	7.06e-3	0.70	10.43	11.98	9.2
	BOBYQA	2,718.94	2914.27	8.1	1,280.51	1,816.09	0.38	7.35e-3	0.0111	8.2	9.91	11.52	8.0
	SMCO	2,307.95	2,422.66	9.3	89.09	100.55	12	0.191	0.204	11	15.09	15.92	14
	SMCO-R	1,424.85	1,581.45	9.5	152.51	168.25	12	0.0102	0.0150	11	10.28	12.25	14
	SMCO-BR	1,482.92	1,646.48	9.6	314.89	342.48	12	0.0118	0.0163	11	9.65	11.39	14
	GenSA	1,167.86	1,772.72	7.9	0	0	8.8	6.61e-3	0.0109	8.2	6.23	8.80	18
	SA	1,556.56	1,917.50	0.045	3,539.77	4688.62	0.035	0.363	0.413	0.039	19.14	20.34	0.052
	DEoptim	1,805.04	1,937.33	0.87	3,292.97	3484.97	1.0	0.284	0.312	1.0	16.23	17.20	1.3
	CMAES	418.84	868.91	136	672.77	1,614.30	8.4	0.428	0.580	28	11.84	19.75	74
	STOGO	3,837.20	3,837.20	7.3	ERROR	ERROR	8.5	0.113	0.113	8.4	15.58	15.58	11
	GA	3,113.40	3,361.89	0.28	8,637.59	9,412.69	0.29	0.308	0.389	0.28	16.60	19.08	0.29
	PSO	827.38	1,475.95	0.74	3,097.68	5,063.57	0.76	0.290	0.328	0.8	14.39	16.16	0.90
Min	**Best**	**69.80496**			**7.320811e-13**			**1.179612**			**-25.24325**		
	GD	350.64	462.75	32	1,186.21	1439.17	56	18.74	18.77	34	13.19	15.06	56
	SignGD	853.17	1,011.03	49	3,015.98	3685.10	56	18.74	18.77	53	17.60	18.93	7.3
	ADAM	792.90	937.19	1.4	1.97e-13	3.00e-13	9.5	18.74	18.77	1.4	10.85	12.59	28
	SPSA	966.96	1,077.18	1.9	3204.76	3,890.44	0.93	18.91	19.10	0.44	21.09	22.54	2.6
	L-BFGS	599.33	711.08	1.0	4.15e-10	5.99e-10	1.0	18.74	18.77	0.86	10.12	11.84	9.0
	BOBYQA	813.91	943.09	8.3	1.70e-3	5.12e-3	8.6	18.76	18.79	7.9	9.10	10.87	7.9
	SMCO	264.32	296.12	10	1.08	1.09	12	19.06	19.18	12	16.51	17.38	15
	SMCO-R	261.10	289.54	10	0.680	0.755	12	18.37	18.63	12	8.33	10.24	15
	SMCO-BR	258.94	294.48	10	0.970	1.01	12	18.27	18.63	12	9.13	10.68	15
	GenSA	348.10	483.46	7.7	7.51e-12	1.11e-11	9.7	18.74	18.79	8.0	5.54	8.34	17
	SA	1,099.59	1,261.13	0.03	1,403.02	2,217.09	0.01	20.10	20.18	0.039	19.09	20.36	0.055
	DEoptim	539.37	591.06	0.87	53.19	62.06	0.98	19.93	20.01	0.96	16.12	17.18	1.2
	CMAES	195.48	569.49	42	140.47	410.28	25	19.94	20.46	26	12.74	20.27	84
	STOGO	579.89	579.89	7.2	3.85e-9	3.85e-9	8.2	18.66	18.66	8.1	15.99	15.99	11
	GA	649.13	757.55	0.29	756.14	946.54	0.28	19.81	20.01	0.29	16.29	19.23	0.30
	PSO	229.28	338.40	0.77	0.0175	0.05	0.75	10.88	20.03	0.79	14.47	16.20	0.91

Based on 100 replications. SMCO & Group I: 71 starting points. SMCO: max iter = 200, tol = 1e-8.

Table 9: Dimension $d = 200$, Diagonal Multi-Start

Test Function		**Rastrigin**			**Griewank**			**Ackley**			**Michalewicz**		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Min	**Best**	**1596.819**			**2.35e-13**			**19.85761**			**-62.77543**		
	SMCO	**285**	**365**	23.2	**1.08**	**1.09**	23.6	**0.103**	**0.112**	23.9	**42.51**	**45.05**	31.5
	SMCO-R	**318**	**376**	23.3	**0.95**	**0.98**	24.0	**0.011**	**0.015**	24.1	**28.00**	**29.09**	31.6
	SMCO-BR	**287**	**340**	23.4	**0.97**	**0.99**	24.7	**0.012**	**0.018**	24.2	**30.84**	**32.49**	31.6
	GenSA	1718	2011	268.3	3.05e-11	6.10e-11	302.2	0.109	0.114	271.7	6.27	12.03	813.4
	SA	3947	4338	0.2	11626	14002	0.1	1.653	1.678	0.3	52.76	54.53	0.36
	DEoptim	3813	3968	31.5	5856	6086	33.4	1.548	1.564	32.2	47.41	48.91	43.5
	STOGO	1293	1293	285.6	4.38e-09	4.38e-09	307.1	0.002	0.002	295.4	43.48	43.48	404.4
	GA	2496	2623	0.5	7372	7920	0.5	1.452	1.502	0.5	49.28	52.30	0.61
	PSO	590	812	4.9	81	143	5.1	1.548	1.587	5.1	43.13	45.36	6.19
Max	**Best**	**18239.57**			**72331.4**			**22.33939**			**69.53472**		
	SMCO	**7219**	**7596**	22.9	**189**	**200**	24.6	**0.121**	**0.124**	23.7	**36.91**	**38.65**	31.3
	SMCO-R	**3918**	**4061**	23.3	**272**	**283**	24.9	**0.106**	**0.110**	23.5	**20.64**	**21.82**	31.4
	SMCO-BR	**4050**	**4095**	23.3	**587**	**605**	25.0	**0.012**	**0.012**	23.9	**25.22**	**26.92**	31.6
	GenSA	2605	2975	275.9	0	0	312.7	0.004	0.007	273.5	4.65	8.98	814.1
	SA	5653	7180	0.3	32221	36635	0.3	0.488	0.505	0.3	55.48	57.14	0.37
	DEoptim	7978	8191	31.8	36012	36465	33.7	0.421	0.432	32.6	49.83	50.98	43.8
	STOGO	10987	10987	290.0	9908	9908	310.0	0.120	0.120	299.9	45.26	45.26	407.8
	GA	8977	9310	0.5	45978	46953	0.5	0.450	0.489	0.5	49.86	51.20	0.61
	PSO	998	1719	5.0	20792	23940	5.2	0.429	0.448	5.2	46.50	48.88	6.23

Based on: 10 reps. SMCO/-R/-BR run from 14 diagonal start points in parallel.

Table 10: Dimension $d = 500$, Diagonal Multi-Start

		Rastrigin			Michalewicz			Griewank			Ackley		
		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Max	Best		**41635.04**			**123.50**			**181936.18**			**22.35**	
	SMCO	**16141.12**	**16305.43**	**207.61**	**109.90**	**113.51**	**223.86**	**778.66**	**805.66**	**212.92**	**0.16**	**0.17**	**206.97**
	SMCO-R	**5939.57**	**5945.41**	**696.47**	**72.93**	**75.79**	**228.85**	**201.05**	**214.89**	**724.20**	**0.12**	**0.12**	**211.83**
	SMCO-BR	**5683.68**	**5768.69**	**206.89**	**78.84**	**80.96**	**227.13**	**1958.97**	**2008.52**	**212.96**	**0.01**	**0.01**	**213.54**
	GenSA	2007.79	3382.04	2088.85	6.81	10.37	3941.08	0.00	0.00	2734.51	0.01	0.01	2085.72
	DEoptim	18596.52	18955.17	338.83	98.52	99.83	364.38	111051.66	111240.77	350.08	0.49	0.50	329.34
	PSO	3370.14	6057.01	22.12	91.60	95.21	24.76	67941.28	72225.00	23.92	0.49	0.50	23.35
Min	Best		**4953.99**			**-140.11**			**0.00**			**19.86**	
	SMCO	**188.23**	**302.66**	**206.59**	**113.11**	**115.00**	**236.19**	**1.49**	**1.51**	**215.68**	**0.29**	**0.29**	**209.97**
	SMCO-R	**163.41**	**248.16**	**206.33**	**78.85**	**80.81**	**226.41**	**0.94**	**0.97**	**217.47**	**0.02**	**0.02**	**211.02**
	SMCO-BR	**187.19**	**281.60**	**214.01**	**84.48**	**86.89**	**222.84**	**1.02**	**1.04**	**211.94**	**0.02**	**0.02**	**209.58**
	GenSA	4994.21	5541.87	2057.16	5.88	6.56	3941.37	0.00	0.00	2286.15	0.12	0.13	2076.07
	DEoptim	11294.15	11294.15	354.66	109.41	109.41	390.09	29094.97	29094.97	367.80	1.66	1.66	355.15
	PSO	2495.80	3101.78	23.59	101.60	105.62	25.41	2277.66	2582.27	23.57	1.63	1.65	23.15

Based on 10 replications, tolerance =1e-06, maximum iteration=300 and the number of multi-start is 30 for SMCO method.

Table 11: Sensitivity: Asymmetrization/Enlargement Transformation

Minimization of Rastrigin Function ($d = 500$)						
Algorithm	Time (sec)	RMSE	Performance Metrics			Best Value
			AE 50%	AE 95%	AE 99%	
Main-Text Configuration (25% ±5%, 50% ±10%)						
SMCO	188.889	116.510	71.163	177.069	186.483	4978.24
SMCO-BR	198.900	134.120	151.389	171.082	172.832	5010.15
SMCO-R	200.381	154.574	178.063	197.746	199.496	5017.27
GenSA	1371.450	5019.651	4893.066	5252.740	5284.711	9907.08
DEoptim	166.636	10954.476	10917.423	11095.872	11111.734	15845.09
PSO	10.535	2842.682	2855.954	2954.504	2963.264	7731.87
Variant1 Configuration (25% ±5%, 25% ±5%)						
SMCO	188.687	148.780	152.367	158.668	159.229	2705.21
SMCO-BR	202.054	71.965	51.707	107.246	112.183	2611.88
SMCO-R	200.051	116.283	127.235	136.212	137.010	2669.82
GenSA	1371.725	3001.373	3095.112	3136.194	3139.845	5553.23
DEoptim	166.118	7767.578	7736.598	7868.279	7879.984	10323.95
PSO	10.397	4261.991	4238.920	4307.672	4313.784	6818.66
Variant2 Configuration (25% ±5% , 75% ±10%)						
SMCO	189.525	490.762	519.815	547.085	549.509	8016.38
SMCO-BR	193.758	457.677	447.492	475.280	477.750	7988.20
SMCO-R	193.144	483.404	521.899	537.952	539.378	8008.17
GenSA	1373.213	6903.926	6854.474	7107.022	7129.471	14432.48
DEoptim	166.342	15421.691	15338.121	15611.723	15636.043	22951.64
PSO	10.510	414.609	36.133	649.106	703.593	7781.88

Minimization of Rastrigin function with d=500 based on 10 replications. SMCO/-R/-BR run from 30 diagonal starting points in parallel, with 300 max iterations.

Table 12: Sensitivity: Number of Maximum Iterations and Type of Starting Points

Test Function		Rastrigin, dimension = 500					
Start Point		Single Uniform Start Point		30 Uniform Start Points		30 Diagonal Start Points	
Max Iteration		300	500	300	500	300	500
Max	**SMCO**	**324.71**	**268.91**	**232.39**	**179.41**	**29.74**	**10.77**
	SMCO-R	**288.16**	**268.23**	**176.98**	**176.82**	**2.13**	**1.89**
	SMCO-BR	**346.76**	**289.74**	**230.53**	**177.47**	**2.49**	**1.86**
	GD	5712.24	5713.03	5703.85	5703.85	1151.57	1151.57
	SignGD	6000.50	6103.89	5733.55	5733.55	3.64e-12	3.64e-12
	SPSA	5714.29	5714.29	5714.29	5714.29	5714.29	5467.90
	BOBYQA	10961.36	11036.42	10538.14	10538.14	1151.57	1151.57
Min	**SMCO**	**14.07**	**7.99**	**10.44**	**4.00**	**9.44**	**3.73**
	SMCO-R	**7.51**	**5.05**	**1.66**	**1.29**	**1.19**	**1.26**
	SMCO-BR	**4.81**	**4.77**	**1.73**	**1.52**	**0.61**	**0.60**
	GD	7118.46	7082.94	6266.04	6351.35	646.38	646.38
	SignGD	4482.44	4488.21	4024.81	4014.86	1.21e-09	1.21e-09
	SPSA	14462.36	14462.36	11190.31	12832.44	11840.35	11881.44
	BOBYQA	9314.75	9159.82	8777.84	8713.90	1686.01	1686.01

Based on 10 reps. On original (untransformed) domain.

Table 13: Sensitivity: Tolerance Level

Test Function		Rastrigin					
Setting		$d = 500$, Max Iter = 300			$d = 500$, Max Iter = 500		
Tolerance		1e-04	1e-06	1e-08	1e-04	1e-06	1e-08
Max	**SMCO**	**233.20**	**222.78**	**206.89**	**202.98**	**226.73**	**195.98**
	SMCO-R	**183.33**	**197.71**	**188.02**	**192.89**	**192.06**	**206.55**
	SMCO-BR	**251.21**	**252.67**	**253.95**	**185.63**	**202.86**	**195.82**
	GD	5700.08	5700.91	5700.91	5700.08	5700.91	5700.91
	SignGD	5720.35	5741.50	5741.50	5720.35	5741.50	5741.50
	SPSA	5714.29	5714.29	5714.29	5714.29	5714.29	5714.29
	L-BFGS	6790.24	4037.30	4037.30	6984.23	4042.53	4042.53
	BOBYQA	10471.71	10497.07	10497.07	10471.71	10497.07	10497.07
Min	**SMCO**	**10.26**	**10.50**	**10.35**	**4.52**	**4.63**	**4.20**
	SMCO-R	**1.76**	**1.75**	**1.36**	**1.42**	**1.46**	**1.70**
	SMCO-BR	**1.60**	**1.60**	**1.76**	**1.57**	**1.20**	**1.76**
	GD	6328.91	6328.91	6328.91	6328.91	6317.07	6328.91
	SignGD	3993.21	3993.21	3993.21	3993.21	4022.50	3993.21
	ADAM	2077.66	987.57	987.57	1961.95	981.28	981.27
	SPSA	10987.23	10987.23	10987.23	10987.23	12860.98	10987.23
	L-BFGS	2164.18	1074.12	1074.12	2048.47	1067.83	1067.83
	BOBYQA	8775.96	8775.96	8775.96	8775.96	8852.37	8775.96

Based on 10 reps. On original (untransformed) domain. All algorithms use 30 diagonal starting points.

Table 14: Sensitivity: Running Time of Selected Competing Algorithms, Original Rastrigin with $d = 1000$

Algorithm	**Minimization (Target: 0)**				**Maximization (Target: 40353.29)**			
	Time (sec)	RMSE	AE50	AE95	Time (sec)	RMSE	AE50	AE95
Original Rastrigin Domain: $[-5.12, 5.12]^{1000}$.								
SMCO	**56.14**	**19.62**	**19.47**	**20.75**	**54.94**	**59.65**	**59.63**	**61.11**
SMCO-R	**56.19**	**0.00**	**0.00**	**0.00**	**54.98**	**0.00**	**0.00**	**0.00**
SMCO-BR	**56.70**	**0.00**	**0.00**	**0.00**	**55.44**	**0.00**	**0.00**	**0.00**
GenSA	1867.35	0.00	0.00	0.00	2232.68	0.00	0.00	0.00
GenSA*	57.78	387.64	379.08	432.56	61.14	50.22	27.14	85.64
DEoptim	113.85	15847.66	15859.08	15915.79	115.16	19385.83	19387.06	19491.50
PSO	5.64	5417.89	6048.48	6549.47	6.03	10959.24	11064.37	11418.06
PSO*	58.36	5409.06	5503.365	6537.48	61.72	9459.40	10552.61	11444.92

Results based on 10 replications with diagonal draw starting points. SMCO configuration: 35 starting points, 300 max iterations, parallel computing enabled with tolerance level 1e-06. RMSE: root mean square error; AE: absolute error percentiles. SMCO-R, SMCO-BR and GenSA achieve perfect convergence (0.00) for both optimization directions, but GenSA needs a great deal of time. GenSA and PSO use the defalt settings, relatively faster GenSA* uses the same defalt hyperparameters of GenSA except for max.call= 1e6 (defalt max.call=1e7) and PSO* choose the same hyperparameters of PSO except for s=700 (defalt s=10+2$\sqrt{d}$).

R/-BR and Group I (local) algorithms. While Group I algorithms display sensitivity with respect to the type of starting points, the performance of our SMCO algorithms are very stable, still outperforming both Group I and II algorithms.

Table 17 reports results on maximum score estimation for $d = 50, 100$, in which the error term ϵ_i is drawn from the Cauchy distribution instead of the normal distribution in the main text. **Table 18** further reports the sensitivity checks of SMCO performance in maximum score estimation in $d = 100$ with respect to the SMCO bound buffer parameter δ^*. Overall our SMCO algorithms perform well across these test configurations.

Table 19 reports results on empirical welfare maximization in $d = 100$, as well as the minimization of empirical welfare function as an additional test function (without economic meanings). The results further demonstrate the best performance of our SMCO algorithsm among all Group I and II algorithms in higher dimensions, as well as the robustness of our SMCO algorithms wrt the SMCO bound buffer parameter δ^*.

Table 15: Maximization of $f_{MS}(\theta)$ and $f_{EW}(\theta)$

Dimension		$d=5$			$d=10$			$d=20$			$d=50$		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Maximum Score Estimation: $\max_\beta f_{MS}(\beta)$													
Best		**0.42**			**0.448**			**0.462**			**0.424**		
(I)	GD	0.137	0.169	0.06	0.154	0.183	0.03	0.172	0.197	0.08	0.157	0.174	0.13
	SignGD	0.137	0.169	0.08	0.154	0.183	0.03	0.172	0.197	0.09	0.157	0.174	0.11
	ADAM	0.137	0.169	0.07	0.154	0.183	0.07	0.172	0.197	0.08	0.157	0.174	0.11
	SPSA	0.137	0.169	0.11	0.154	0.183	0.08	0.172	0.197	0.12	0.157	0.174	0.12
	L-BFGS	0.137	0.169	0.07	0.154	0.183	0.08	0.172	0.197	0.10	0.157	0.174	0.13
	BOBYQA	0.090	0.105	0.07	0.110	0.136	0.09	0.131	0.146	0.11	0.134	0.155	0.22
	SMCO	0.008	0.010	0.26	0.009	0.012	0.40	0.013	0.016	0.86	0.056	0.068	5.09
	SMCO-R	0.002	0.004	0.22	0.004	0.006	0.37	0.004	0.006	0.86	0.010	0.016	5.19
	SMCO-BR	0.003	0.004	0.24	0.007	0.008	0.40	0.010	0.012	0.89	0.010	0.017	5.41
(II)	GenSA	0.017	0.054	1.96	0.032	0.034	5.01	0.032	0.036	16.74	0.010	0.018	89.93
	SA	0.071	0.076	0.07	0.073	0.087	0.08	0.106	0.116	0.14	0.114	0.124	0.34
	DEoptim	0.004	0.006	0.34	0.036	0.038	0.82	0.045	0.048	2.92	0.056	0.064	13.18
	STOGO	0.052	0.052	3.88	0.068	0.068	8.84	0.080	0.080	29.02	0.082	0.082	182.28
	GA	0.053	0.058	0.27	0.035	0.040	0.29	0.053	0.062	0.43	0.059	0.070	0.65
	PSO	0.048	0.054	0.84	0.034	0.038	1.11	0.039	0.044	1.80	0.017	0.026	4.32
Empirical Welfare Maximization: $\max_b f_{EW}(\beta)$													
Best		**1541.032**			**2061.409**			**2634.523**			**3825.49**		
(I)	GD	1058	1442	0.11	1351	1646	0.14	1767	1942	0.22	2934	3067	0.60
	SignGD	1058	1442	0.14	1351	1646	0.14	1767	1942	0.21	2934	3067	0.60
	ADAM	1058	1442	0.13	1351	1646	0.15	1767	1942	0.21	2934	3067	0.62
	SPSA	991	1442	0.12	1344	1721	0.13	1788	1942	0.17	2920	3063	0.69
	L-BFGS	886	1437	0.12	1259	1448	0.16	1766	1942	0.23	2927	3067	0.99
	BOBYQA	465	921	0.17	930	1235	0.20	1272	1503	0.35	2338	2599	1.51
	SMCO	214	270	1.81	191	309	3.88	249	312	10.41	428	498	78.37
	SMCO-R	139	218	1.73	69	131	3.85	82	159	10.46	158	233	78.59
	SMCO-BR	134	208	1.75	52	87	3.88	124	193	10.52	274	361	78.71
(II)	GenSA	105	145	15.15	191	245	35.40	636	903	102.96	1586	1762	455.24
	SA	199	247	0.51	416	527	0.63	860	1227	0.94	1939	2398	1.57
	DEoptim	136	155	2.69	232	272	6.61	611	681	19.74	1722	1805	87.03
	STOGO	134	134	31.85	470	470	71.97	835	835	204.88	2096	2096	891.62
	GA	155	206	1.21	277	441	1.48	657	779	2.17	1685	1831	3.69
	PSO	102	134	4.13	165	274	5.72	285	384	9.46	848	1051	21.52

SMCO/-R/-BR and Group (I) algos use $\sqrt{d} \approx 2, 3, 4, 7$ uniform starting points.
All based on 10 replications.

Table 16: Maximum Score Estimation and Empirical Welfare Maximization: Diagonal Multi-Start

Dimension		$d = 5$			$d = 10$			$d = 20$			$d = 50$		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Maximum Score Estimation: $\max_\beta f_{MS}(\beta)$													
Best		**0.42**			**0.45**			**0.46**			**0.424**		
(I)	GD	0.060	0.060	0.08	0.090	0.090	0.04	0.152	0.152	0.09	0.098	0.098	0.13
	SignGD	0.060	0.060	0.07	0.090	0.090	0.04	0.152	0.152	0.08	0.098	0.098	0.11
	ADAM	0.060	0.060	0.07	0.090	0.090	0.07	0.152	0.152	0.08	0.098	0.098	0.11
	SPSA	0.060	0.060	0.11	0.090	0.090	0.08	0.152	0.152	0.12	0.098	0.098	0.12
	L-BFGS	0.060	0.060	0.07	0.090	0.090	0.07	0.152	0.152	0.09	0.098	0.098	0.11
	BOBYQA	0.022	0.022	0.07	0.020	0.020	0.08	0.066	0.066	0.10	0.022	0.022	0.24
	SMCO	0.008	0.010	0.29	0.010	0.014	0.41	0.012	0.018	0.94	0.052	0.058	4.92
	SMCO-R	0.003	0.004	0.25	0.007	0.010	0.38	0.003	0.006	0.92	0.008	0.012	4.99
	SMCO-BR	0.002	0.004	0.28	0.007	0.010	0.41	0.008	0.012	0.97	0.009	0.014	5.16
(II)	GenSA	0.017	0.054	1.96	0.034	0.036	5.01	0.030	0.034	16.74	0.010	0.018	89.93
	SA	0.071	0.076	0.07	0.075	0.089	0.08	0.104	0.114	0.14	0.114	0.124	0.34
	DEoptim	0.004	0.006	0.34	0.038	0.040	0.82	0.043	0.046	2.92	0.056	0.064	13.18
	STOGO	0.052	0.052	3.88	0.070	0.070	8.84	0.078	0.078	29.02	0.082	0.082	182.28
	GA	0.053	0.058	0.27	0.037	0.042	0.29	0.051	0.060	0.43	0.059	0.070	0.65
	PSO	0.048	0.054	0.84	0.036	0.040	1.11	0.037	0.042	1.80	0.017	0.026	4.32
Empirical Welfare Maximization: $\max_b f_{EW}(\beta)$													
Best		**1541.032**			**2050.312**			**2659.651**			**3827.822**		
(I)	GD	356	356	0.15	898	898	0.19	2122	2122	0.24	2477	2477	0.70
	SignGD	356	356	0.13	898	898	0.16	2122	2122	0.22	2477	2477	0.67
	ADAM	484	484	0.45	993	993	0.56	2122	2122	0.21	2771	2771	1.41
	SPSA	622	857	0.14	1329	1489	0.17	2094	2122	0.17	2974	3153	0.73
	L-BFGS	236	236	0.43	797	797	0.56	2122	2122	0.20	1775	1775	8.83
	BOBYQA	386	386	0.19	656	656	0.25	1085	1085	0.36	1163	1163	1.50
	SMCO	200	268	1.89	164	200	4.27	318	386	10.32	409	547	76.46
	SMCO-R	136	153	1.84	63	109	4.24	106	173	10.33	172	251	76.56
	SMCO-BR	146	177	1.91	66	120	4.27	118	157	10.42	217	316	76.68
(II)	GenSA	105	145	15.15	181	234	35.40	660	928	102.96	1589	1765	455.24
	SA	199	247	0.51	405	516	0.63	885	1252	0.94	1942	2401	1.57
	DEoptim	136	155	2.69	221	261	6.61	637	706	19.74	1724	1807	87.03
	STOGO	134	134	31.85	459	459	71.97	860	860	204.88	2098	2098	891.62
	GA	155	206	1.21	267	430	1.48	682	804	2.17	1687	1833	3.69
	PSO	102	134	4.13	155	262	5.72	309	409	9.46	851	1054	21.52

Based on 10 replications. SMCO/-R/-BR and Group (I) algos use $\sqrt{d} \approx 2, 3, 4, 7$ diagonal starting points.

Table 17: Maximum Score Estimation with Cauchy Errors

	$d = 50$			$d = 100$		
Algorithm	RMSE	AE99	Time	RMSE	AE99	Time
Best		**0.4900**			**0.4540**	
SMCO	**0.1118**	**0.1176**	19.8548	**0.1410**	**0.1458**	70.1807
SMCO-R	**0.0210**	**0.0276**	19.2028	**0.0341**	**0.0398**	74.7200
SMCO-BR	**0.0046**	**0.0060**	19.6607	**0.0143**	**0.0180**	72.0697
GD	0.1420	0.1420	0.3721	0.1020	0.1020	1.9571
SignGD	0.1420	0.1420	0.3554	0.1020	0.1020	1.8590
SPSA	0.1420	0.1420	0.0506	0.1020	0.1020	0.0853
optimLBFGS	0.1420	0.1420	0.3520	0.1020	0.1020	2.8355
BOBYQA	0.0420	0.0420	2.2335	0.0240	0.0240	16.2511
GenSA	0.0344	0.0380	63.6950	0.0152	0.0218	286.4138
SA	0.1520	0.1600	0.2773	0.1429	0.1553	0.5721
DEoptim	0.0841	0.0898	10.7332	0.0922	0.0998	44.2958
CMAES	0.0581	0.0789	3.4109	0.0315	0.0478	13.1697
stogo	0.1060	0.1060	136.1108	0.0880	0.0880	541.6599
GA	0.0874	0.1206	0.6455	0.0872	0.1095	1.3692
PSO	0.0452	0.0589	3.7318	0.0357	0.0526	9.0746

Based on 10 replications. SMCO/-R/-BR and Group I algorithms run from $4\sqrt{d}$ diagonal starting points. SMCO: iter max = 1000, bounds buffer = 0.02, parallelization across start points.

Table 18: Maximum Score Estimation with Cauchy Errors and $d = 100$: Sensitivity wrt Bound Buffer δ^*

Bound Buffer	$\delta^* = 0.02$			$\delta^* = 0.05$		
Algorithm	RMSE	AE99	Time (s)	RMSE	AE99	Time (s)
Best		**0.4640**			**0.458**	
SMCO	**0.1216**	**0.1280**	46.6113	**0.1154**	**0.1240**	45.4532
SMCO-R	**0.0457**	**0.0514**	45.1869	**0.0388**	**0.0468**	44.4921
SMCO-BR	**0.0302**	**0.0354**	48.1126	**0.0236**	**0.0314**	46.9663
GD	0.1120	0.1120	1.8369	0.1060	0.1060	1.8720
SignGD	0.1120	0.1120	1.8657	0.1060	0.1060	1.8777
SPSA	0.1120	0.1120	0.0865	0.1060	0.1060	0.0931
optimLBFGS	0.1120	0.1120	3.0625	0.1060	0.1060	3.3938
BOBYQA	0.0340	0.0340	16.3390	0.0280	0.0280	16.6256
GenSA	0.0255	0.0314	281.1997	0.0185	0.0308	308.4736
SA	0.1521	0.1660	0.5504	0.1476	0.1580	0.6533
DEoptim	0.1042	0.1100	45.6777	0.0976	0.1054	47.3026
CMAES	0.0373	0.0588	14.2327	0.0374	0.0771	13.0083
stogo	0.0980	0.0980	565.0344	0.0920	0.0920	616.7450
GA	0.0903	0.1063	1.2363	0.0882	0.1100	1.3138
PSO	0.0420	0.0574	8.9114	0.0382	0.0588	9.6422

Based on 30 replications. SMCO and Group I algos run from diagonal start $4\sqrt{d}$.
SMCO run with max iter = 500, parallelization across start points.

Table 19: Empirical Welfare Maximization & Minimization: $d = 100$

# Starting Points		Max			Min		
Performance		RMSE	AE99	Time	RMSE	AE99	Time
	Best	**4536.174**			**-3513.749**		
(I)	GD	3589	3815	1.5	3732	3887	1.6
	SignGD	3589	3815	1.3	3732	3887	1.5
	ADAM	3589	3815	1.5	3732	3887	1.8
	SPSA	3616	3823	0.6	3709	3891	0.6
	L-BFGS	3589	3815	3.6	3701	3887	4.6
	BOBYQA	3062	3279	4.8	3177	3361	6.1
$\delta^* = 0.05$	**SMCO**	**328**	**406**	**362**	**328**	**455**	**461**
	SMCO-R	**92**	**174**	**352**	**99**	**137**	**475**
	SMCO-BR	**267**	**324**	**351**	**311**	**382**	**477**
$\delta^* = 0.02$	**SMCO**	**349**	**404**	**413**	**386**	**485**	**421**
	SMCO-R	**70**	**142**	**421**	**132**	**207**	**420**
	SMCO-BR	**294**	**343**	**411**	**311**	**421**	**424**
(II)	GenSA	2079	2463	1478	2137	2330	2001
	SA	2517	2028	2.7	3195 2	3403	1.4
	DEoptim	2485	2052	282	487	2588	381
	GA	2400	2142	5.9	2364	2617	7.9
	PSO	1043	3509	43	1123	1340	59

Based on 10 replications.
SMCO/-R/-BR and Group (I) algos use $\sqrt{d} = 10$ uniform starting points (in parallel).
δ^* refers to bound buffer parameter.
STOGO excluded due to overly long computation time.

F Additional Random Test Functions

F.1 Conditional Moment Inequalities in Dynamic Binary Choice Models

We now consider another random test function from economics, which summarizes identifying restrictions from a class of conditional moment inequalities in a dynamic (endogenous) binary choice model as studied in a recent paper by Gao and Wang (2025). Specifically, consider the following binary choice model:

$$Y_{it} = \mathbb{1}\left\{Z_{it} + \gamma_0 X_{it} + \alpha_i + \epsilon_{it} \geq 0\right\}, \quad t = 1, ..., T = 3$$

where Z_{it} and X_{it} are both taken to be scalar valued observed covariates, with Z_{it} being the exogenous covariate while X_{it} being the endogenous covariates. We normalize the coefficient on Z_{it} to 1 and focus on the identification of the coefficient γ_0. Here, α_i is an unobserved fixed effect, while ϵ_{it} is an unobserved time-varying error term whose distribution is left nonparametric but assumed to satisfy a partial stationarity assumption across time given the exogenous covariates Z_i. Gao and Wang (2025) shows that under appropriate conditions, the identified set for γ_0 is given by

$$\Gamma_I := \{\gamma : \ Q(\gamma) \leq 0\}.$$

where the criterion function is defined as a parametrized maximum function $Q(\gamma) := \max_{c \in \mathcal{C}, z \in \mathcal{Z}} Q(\gamma, c, z)$ with

$$Q(\gamma, c, z) := \max_t \mathbb{P}\left(Y_{it} = 1, \ z_t + \gamma_0 X_{it} \leq c \middle| \, z\right) + \max_t \mathbb{P}\left(Y_{it} = 0, \ z_t + \gamma_0 X_{it} \geq c \middle| \, z\right) - 1$$

Gao and Wang (2025) numerically computes $Q(\gamma)$ under a specific data generating process, and we replicate this exercise here. Specifically, conditional on $Z_i = z \in \mathcal{Z} := [-10, 10]^T$, we set the error term $\epsilon_{it} \sim_{i.i.d.} N(0, 1)$, and the fixed effect $\alpha_i = \rho_\alpha \cdot \frac{1}{T}\sum_{t=1}^{T} z_{t,1} + \sqrt{1 - \rho_\alpha^2}\xi_i$ with $\rho_\alpha = 0.1$ and $\xi_i \sim_{i.i.d.} \mathcal{N}(0, 1)$. We consider the continuous specification where $X_{it} := 5\left(2\left(Y_{i,t-1} - 0.5\right) + \eta_{it}\right)$ with $\eta_{it} \sim_{i.i.d.} U(-1, 1)$, and $\gamma_0 = 1$, so that $X_{it}^{'}\gamma_0$ is again of the similar order of magnitude as z. The initial condition $Y_{i,0} \sim_{i.i.d} Bernoulli(0.5)$. We then numerically compute $\hat{Q}(\gamma, c, z)$ as

numerical approximation of $Q(\gamma, c, z)$ via simulations as follows

$$\hat{Q}(\gamma, c, z) := \max_{t=1,\ldots,T} \frac{1}{B} \sum_{b=1}^{B} Y_{bt} \mathbb{1}\{z_t + \gamma X_{bt} \leq c\} + \max_{t=1,\ldots,T} \frac{1}{B} \sum_{b=1}^{B} (1 - Y_{bt}) \mathbb{1}\{z_t + \gamma X_{bt} \geq c\} - 1$$

based on $B = 500$ simulations of (Y_{bt}, X_{bt}) based on the DGP described above, conditional on each value of z. We then optimize $\hat{Q}(\gamma, c, z)$ over $(c, z) \in \mathcal{C} \times \mathcal{Z}$, with $\mathcal{C} := [-30, 30]$, to obtain $\hat{Q}(\gamma)$ as a numerical approximation of $Q(\gamma)$, using our SMCO algorithms along with other competing optimization algorithms. We then plot $\hat{Q}(\gamma)$ as a function of γ in the figure below, on a grid of 40 points for $\gamma \in [-20, 20]$.

Note that the optimization of $\hat{Q}(\gamma, c, z)$ w.r.t. (c, z) is challenging for several reasons. To start, the presence of indicator function $\mathbb{1}\{z_t + \gamma X_{bt} \leq c\}$ and $\mathbb{1}\{z_t + \gamma X_{bt} \geq c\}$ in $\hat{Q}(\gamma, c, z)$ makes the function nonsmooth and discontinuous in the parameters (c, z) to be optimized over. Moreover, the indicator function also implies that the gradient of $\hat{Q}$ is almost everywhere zero. The discreteness in $\hat{Q}$ is further compounded by the fact that Y_{bt} is also a binary variable. Finally, there is little a priori information on the structure of $\hat{Q}$ w.r.t (c, z): the function $\hat{Q}$ needs not be convex or monotone.

Figure 3 contains plots of $\hat{Q}(\gamma)$ computed from SMCO_BR along with other competing optimization algorithms, while Figure 4 contains results from the same exercise but focuses on the comparison of SMCO_BR with GenSA and DEoptim only. Observe that SMCO_BR performs robustly at the top across the whole range of γ under consideration, and performs on par with GenSA and DEoptim while beating all other algorithms. It is also clear from Figure 4 that SMCO_BR, GenSA and DEoptim yield very similar curves, though our SMCO_BR finds a noticeably larger maximum than GenSA and DEoptim at $\gamma = 0$.

F.2 Square Loss Function for ReLU Neural Networks

We also test the optimization algorithms on randomly generated functions constructed from ReLU neural networks, which are well known to have complicated and challenging optimization landscapes. Specifically, we start with the following single-hidden-layer ReLU neural networks with d_{in} input nodes and d_{hi} hidden nodes:

$$g(x, z) = \sum_{k=1}^{d_{hi}} w_{2k} \left[\sum_{j=1}^{d_{in}} w_{1kj} z_j + b_{1k} \right]_+ + b_2,$$

Figure 3: Dynamic Binary Choice: All

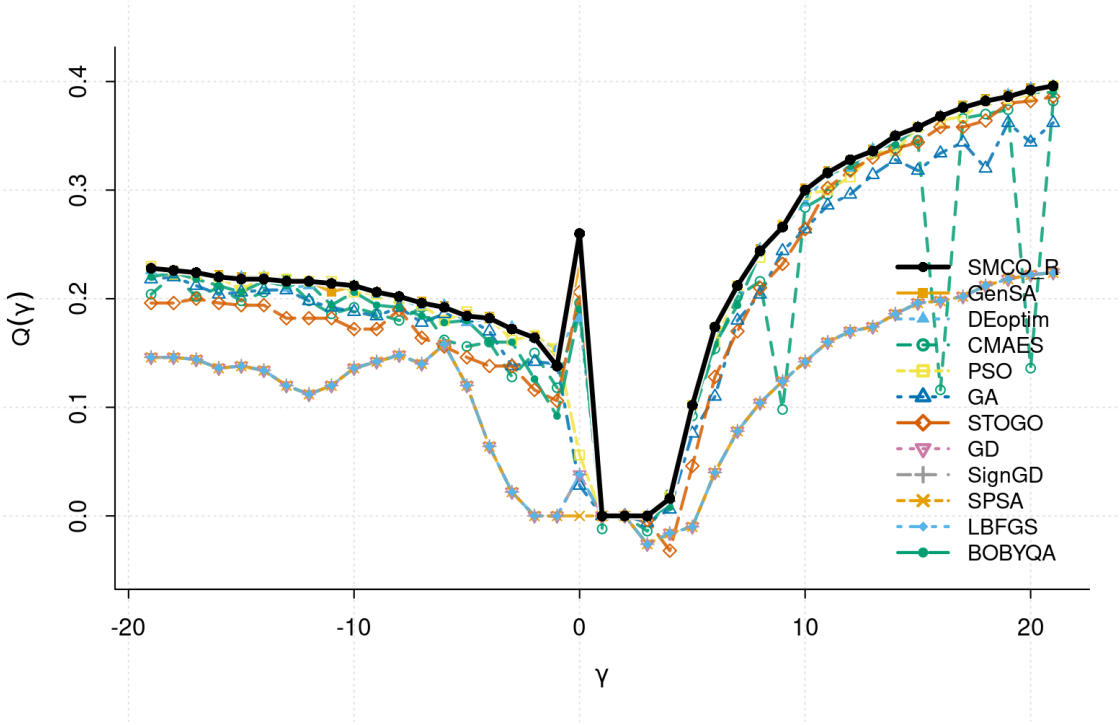
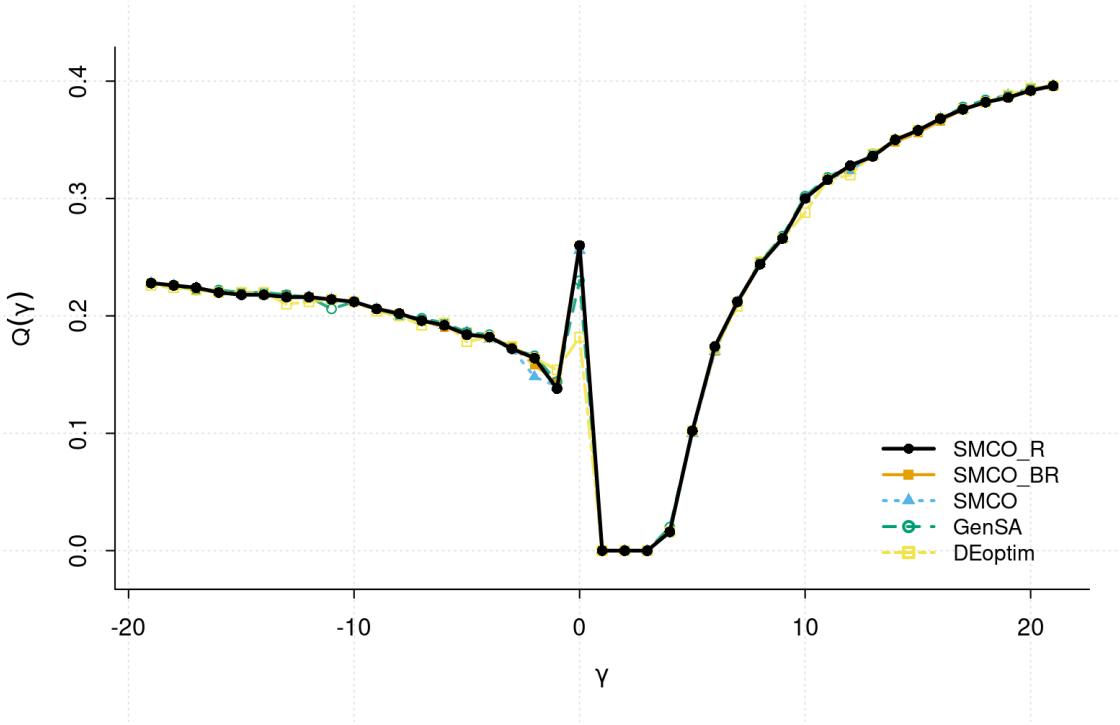


Figure 4: Dynamic Binary Choice: SMCO/-R/-BR, GenSA and DEoptim



where $z \in \mathbb{R}^{d_{in}}$ is a vector of input features and

$$x := \left((w_{1kj})_{j=1,k=1}^{j=d_{in},k=d_{hi}}, (w_{2k})_{k=1}^{d_{hi}}, (b_{1k})_{k=1}^{d_{hi}}, b_2 \right) \in \mathbb{R}^{d_x}$$

collects all the neural-network parameters (weights and biases). Note that the dimension of x is $d_x := d_{in}d_{hi} + 2d_{hi} + 1$. We construct two test functions based on the neural network described above.

We construct the test function based on the neural network using the square loss function, which widely used for regression purposes in statistics and machine learning. Essentially, we test how well the optimization algorithms can solve "in-class" regression problems under the neural network structure described above.

Specifically, we simulate M copies of network parameters x by sampling $w_{1kj} \sim_{i.i.d.} w_{2k} \sim_{i.i.d.} b_2 \sim_{i.i.d.} U[-4,4]$ and $b_{1k} \sim_{i.i.d.} U[0,8]$. The positive support of b_{1k} is designed to make hidden nodes more likely to be activated, creating relatively more challenging optimization landscapes. For each simulated network parameter $x^{(m)}$, we construct an optimization test function as

$$f_m(x) := \frac{1}{H} \sum_{h=1}^{H} \left(g\left(x, z^{(h)}\right) - g\left(x^{(m)}, z^{(h)}\right) \right)^2$$

where $\left\{ z^{(h)} : h = 1, ..., H \right\}$ are $H = 1000$ input points, each of which is randomly sampled according to $z_k^{(h)} \sim_{i.i.d.} U[-4,4]$. By construction, f_m has a global minimum of value 0 at $x^{(m)}$, but there might be multiple global minima (and almost certainly many local optima).

We then run the optimization algorithms to minimize f_m on $[-10,10]^{d_x}$, which contains the support of the simulated $x^{(m)}$ as a strict subset. The minimization of f_m can be interpreted as computing an approximation for the true neural-network regression function $g\left(x^{(m)}, \cdot\right)$ through the least-square method evaluated at the $H = 1000$ input points. We choose this "noise-free" setup to make sure that the true global minimum is known by design, so that the comparison of the optimization algorithms below can be cleanly interpreted as a comparison of optimization performances on complex regression functions induced by the neural networks.[10]

[10] An alternative exercise would be to simulate a noisy outcome $y^{(m,h)} := g\left(x^{(m)}, z^{(h)}\right) + \epsilon^{(h)}$ and minimize $\tilde{f}_m(x) = \frac{1}{H} \sum_{h=1}^{H} \left(g\left(x, z^{(h)}\right) - y^{(m,h)} \right)^2$, with $\lim_{H \to \infty} \tilde{f}_m(x)$ being the (asymptotic) out-of-sample error, whose global minimum becomes $\text{Var}\left(\epsilon^{(h)}\right)$. We could then assess the out-of-sample MSE using an additional sample of (z, y), obtained by the optimization algorithms, which resembles a real-world use of neural networks more. We note, however, such an exercise can be misleading as a measure of optimization performance, since it contaminates the optimization problem with statistical complications. For example, it could be the case that a better optimization

Table 20 contains our comparison using the following configurations: (i) We set the input dimension of the neural networks to be $d_{in} = 3$ and the number of nodes in the hidden layer to be $d_{hi} = 5$, with a total of $d = 26$ parameters. (ii) We set $d_{in} = 3$, $d_{hi} = 10$ with a total of $d = 51$ parameters. (iii) We set $d_{in} = 6$, $d_{hi} = 15$ with a total of $d = 121$ parameters. We run our SMCO/-R/-BR algorithms with the default hyperparameters as well as another pass with the maximum iterations set to 1000 (the default is 500). We report results in minimizing the square loss test function with parameters randomly generated over $M = 10$ replications, for each of which we run the optimization algorithms once. We then compute the average performance measures across the $M = 10$ replications. It is clear from Table 20 that our SMCO algorithms still perform reasonably well, with RMSE competitive with or better than most of the included algorithms (with GenSA being an notable exception). Again, it is worth noting the remarkable stability of our algorithms in terms of AE99 (say, in comparison with PSO).

G Additional Random Statistical Test Functions

All numerical comparisons reported in this Appendix are for statistical estimation and prediction problems, and are implemented in `R` 4.4.1 on a Windows server with an NVIDIA GeForce RTX 4080 GPU. We use the `sgd` for SGD, `GenSA` solver for GenSA, and the `psoptim` solver in package `pso` for PSO available in `R`.

We have implemented the following versions of the SMCO algorithms in this Appendix: SMCO (i.e., step-adaptive SMCO with a single starting point as described in Appendix B); SMCOfs (i.e., the original SMCO Algorithm 1 with a fixed single starting point); SMCOfs_XXX (i.e., the existing XXX algorithm in `R` package using the SMCOfs's 20th iterative point $\hat{x}_{20} = \frac{S^{\boldsymbol{\theta}^*}_{20}}{20}$ as its starting value); and SMCO-BR (i.e., the "boosted" version of SMCO as described in Appendix B).

Examples 1 and 2 below are about correctly specified statistical models with Cauchy density, which is a leading example of a density with fat-tails. Cauchy likelihood is known to be difficult to optimize Brooks and Morgan (1995). Examples 3 and 4 are about using neural networks to approximate true unknown functions for regression and classification problems, in which the random objective functions are not globally concave/convex with respect to the neural network parameters.

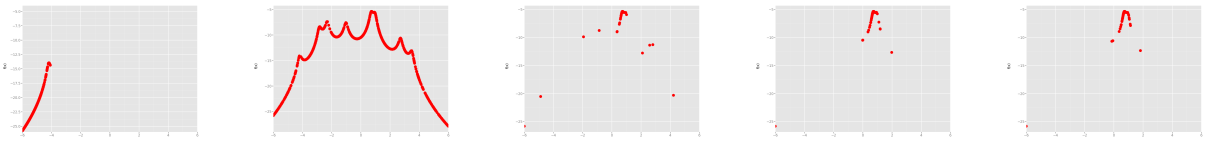
Example 1 (Cauchy log-likelihood function with a small sample size N)**.** *Let X be a Cauchy random*

algorithm is able to "overfit" the model, generates smaller in-sample errors but larger out-of-sample errors in the test sample. However, this should not be interpreted as evidence against the said optimization algorithm, whose sole purpose is to minimize functions.

Table 20: Square Loss Minimization for ReLU Neural Networks

Dimension		$d = 26$			$d = 51$			$d = 121$		
Layer Dimensions		$d_{in} = 3$, $d_{hi} = 5$			$d_{in} = 3$, $d_{hi} = 10$			$d_{in} = 6$, $d_{hi} = 15$		
Performance		RMSE	AE99	Time	RMSE	AE99	Time	RMSE	AE99	Time
Best		**1.13e-8**			**0.01778**			**0.008116**		
(I)	GD	547.8	1273	10.0	420.7	570.5	45.7	1225	2141	154.9
	SignGD	14.4	25.6	10.1	16.7	21.2	46.8	130.2	156.4	160.4
	ADAM	3.7	8.4	7.6	6.7	10.1	18.3	43.6	53.2	32.3
	SPSA	102576	146316	0.8	139017	204410	1.6	809811	1125694	2.4
	L-BFGS	5.3	11.4	1.3	6.3	8.9	5.4	68.7	85.5	17.8
	BOBYQA	22.9	40.3	0.4	43.7	56.1	0.9	907.1	1131	2.3
maxit = 500	**SMCO**	**2.7**	**4.0**	**5.2**	**6.0**	**6.6**	**23.2**	**50.7**	**54.9**	**79.9**
	SMCO-R	**1.5**	**2.5**	**5.2**	**4.4**	**4.8**	**23.3**	**37.3**	**42.5**	**80.4**
	SMCO-BR	**2.7**	**3.3**	**5.3**	**6.1**	**7.0**	**23.6**	**55.1**	**60.2**	**80.4**
maxit = 1000	**SMCO**	**0.9**	**1.9**	**26.9**	**4.0**	**4.8**	**60.7**	**35.8**	**38.4**	**153.7**
	SMCO-R	**0.6**	**1.3**	**26.9**	**3.5**	**4.2**	**60.8**	**31.1**	**34.3**	**153.9**
	SMCO-BR	**0.8**	**1.3**	**27.2**	**4.4**	**5.0**	**61.2**	**34.4**	**39.6**	**153.4**
(II)	GenSA	0.2	0.3	167.8	0.4	1.1	749.0	3.1	6.4	2769
	SA	2530	6747	0.2	2639	3936	0.3	36899	71465	0.1
	DEoptim	68.8	93.8	8.9	293.4	383.3	27.9	2755	3194	96.3
	STOGO	2.9	2.9	94.9	2181	2181	246.0	2315	2315	813.1
	GA	132.7	191.6	0.9	145.3	197.4	1.3	823.6	1182	1.8
	PSO	6.9	13.2	4.1	13.3	19.3	7.4	106.3	130.8	14.1

Based on 10 replications. SMCO/-R/-BR and Group (I) algos use $\sqrt{d} \approx 5, 7, 11$ uniform starting points (in parallel).



(a) SGD (b) GenSA (c) PSO (d) SMCO (e) SMCOfs

Figure 5: The 1000 iterative point-specific function values of Cauchy log-likelihood in Example 1 under SGD, GenSA, PSO, SMCO and SMCOfs. SGD attains a local solution $\tilde{x} = -4.2$ with $f(\tilde{x}) = -14.02$, while GenSA, PSO, SMCO and SMCOfs attain the global maximum $x^* = 0.73$ with $f(x^*) = -5.36$.

variable. The log-likelihood surface of a Cauchy density function $g(X) = c/\{\pi(c^2 + (X - x)^2)\}$ *is known to be difficult to optimize when estimating* x *under some known value* $c > 0$. *Let a random sample* $\{X_i\}_{i=1}^N$ *with* $N = 8$ *be generated from the Cauchy density* $g(X)$ *with* $c = 0.1$, *which is* $X_i \in \{$*-4.20, -2.85, -2.30, -1.02, 0.70, 0.98, 2.72, 3.50*$\}$. *The joint log-likelihood is* $N \log c - \sum_{i=1}^N \log\left\{c^2 + (X_i - x)^2\right\} - N \log \pi$ *with* $N = 8$. *The maximum likelihood estimate for* x *is the global maximizer* $x^* = 0.73$ *of*

$$f(x) = -\sum_{i=1}^{8} \log\left\{c^2 + (X_i - x)^2\right\}, \quad \textit{with } f(x^*) = -5.36.$$

For Example 1, Figure 5 depicts one thousand iterative points of Cauchy log-likelihood $f(x)$ under the five optimization algorithms: SGD, GenSA, PSO, SMCO and SMCOfs. All algorithms have the same initial starting point value -6. The SGD has the learning rate 0.01 and batch size 1 observation. The SMCOfs has uniform distributions Unif (-6.5, -5.5) and Unif (5.5, 6.5) as its left and right arm distributions respectively. The depicted scatterplot of the Cauchy log-likelihood reveals the following: (1) the SGD attains the local maximum $(-4.2, -14.02)$; while GenSA, PSO, SMCO and SMCOfs attain the global maximum $(0.73, -5.36)$. (2) Our SMCO delivers faster exploration to reach the global maximum in terms of more accumulated points around the global maximum under the accuracy level $\varepsilon = 10^{-7}$, while our SMCOfs performs very similar to SMCO for this one dimensional example. (3) In terms of running time, SGD takes 41 milliseconds but attains a local maximum point only; GenSA takes 205.2 milliseconds, PSO needs 3250.8 milliseconds, whereas our SMCO and SMCOfs takes 110.5 and 128.2 milliseconds respectively.

Example 2 (Regression model with Cauchy error)**.** *Recently Feng et al. (2024) propose a new robust antitonic score matching (asm) method for linear regression models with heavy tailed error distributions such as Cauchy error. We follow their numerical experiment design by generating data*

Table S18: Median (over 200 replicates) of the parameter root mean squared error (RMSE) $\|\hat{x} - x^*\|_2/\sqrt{6}$ and standard deviation (SD) of the parameter RMSE, with $\hat{x}$ computed using SMCO-R, SMCO-BR, `nlm`, and `asm` in Example 2.

Method	Cauchy c	RMSE	SD	Time
SMCO-R	1	0.0700	0.0008	1.6446
	2	0.0688	0.0011	2.0276
SMCO-BR	1	0.0703	0.0008	2.6876
	2	0.0695	0.0011	2.1741
`nlm`	1	0.0731	0.0009	0.0022
	2	0.0719	0.0012	0.0037
`asm`	1	0.0763	0.0010	0.1906
	2	0.0738	0.0013	0.2789

from a linear regression model with a Cauchy error:

$$Y_i = Z_i^\top x + \epsilon_i, i = 1, \cdots, N, \quad \textit{with } N = 600,$$

where $Z_i = (1, Z_{i1}, \cdots, Z_{i5})^\top$ *and* Z_{ij} *is independently generated from normal distribution with* $E(Z_{ij}) = \mathrm{var}(Z_{ij}) = 1$*, and the true unknown values* $x^* = (x_1^*, ..., x_6^*)'$ *with* $x_1^* = 1$ *and* $\|x_{-1}^*\|_2 = 3$*. The error term* ϵ_i *is generated from the Cauchy density function* $g(\epsilon) = \frac{c}{\pi(c^2+\epsilon^2)}$ *under two cases* $c = 1,\ 2$*.*

For Example 2, we estimate/calculate $\hat{x}$ using three optimization algorithms: our SMCO-BR, the R function `nlm`, and the antitonic score matching (asm) method of Feng et al. Feng et al. (2024) in the R package `asm`. We implement the `nlm` and `asm` packages using zero vector as the initial values and set the initial value of `asm` minus and plus 2 to be the lower and upper bounds respectively for SMCO-BR in each replicate. `nlm` and SMCO-BR use Cauchy density-based likelihood as the objective functions. Table S18 shows the slightly better estimates performance of our SMCO-BR in terms of both smaller median and standard deviation of parameter root mean squared error (RMSE) $\sqrt{\sum_{j=1}^6 (\hat{x}_j - x_j^*)^2/6}$ after 200 replicates.

Example 3 (Regression using neural network approximation)**.** *We observe five-dimensional covariates* $Z = (Z_1, \cdots, Z_5)^\top$ *following the five-dimensional normal distribution with* $E(Z) = (0)_5^\top$ *and the covariance element* $0.01^{l-l'}$*, but the response generation only uses three dimensions with* $Y = \sin(Z_1) + 0.5\cos(Z_2) + 0.3Z_3^2 + \epsilon$*, where* ϵ *follows a normal distribution with zero mean and standard deviation 0.01. We use a single-hidden layer ReLU neural network* $g(Z, x)$ *to approximate*

the true unknown conditional mean function $E[Y|Z]$*:*

$$g(Z_i, x) = \sum_{k=1}^{3} \omega_{2k} \max(0, \sum_{j=1}^{5} \omega_{1kj} Z_{ij} + b_{1k}) + b_2,$$

where its unknown neural network parameter vector x *consists of 22 weights and biases* $\{\omega_1, b_1, \omega_2, b_2\}$ *with* $\omega_{1,5\times 3}, b_{1,3\times 1}, \omega_{2,3\times 1}, b_{2,1\times 1}$ *for* $k = 1, 2, 3$*. We obtain an estimator* $\hat{x}$ *by nonlinear least squares using a training data set of size* N*, that is,* $\hat{x} = \arg\max_x f(x)$ *with* $f(x) = -\frac{1}{N}\sum_{i=1}^{N} (Y_i - g(Z_i, x))^2$*.*

For Example 3, we compute $\hat{x}$ using three optimization algorithms: our SMCO-BR, the L-BFGS Liu and Nocedal (1989), the SMCOfs_L-BFGS and R package `nnet`. L-BFGS is an improved variant of SGD with batch updates based on the backpropagation algorithm, which is used to calculate the gradient of the loss function with respect to the weights and biases of the network. We implement L-BFGS using uniform distribution Unif(0,1) to generate the initial values of the elements of ω_1 and ω_2 and zero as the the initial values of b_1 and b_2 in each replicate, which is denoted as L-BFGS. We set L-BFGS's initial value minus and plus 1 to be the lower and upper bounds for SMCO-BR and SMCOfs. SMCO-BR considers the number of starting points is set to be 50 and the maximum number of iterations to be 300 with boosting iteration number 100.

We consider three training sizes with $N = 1000, 3000$ and $N_{\text{test}} = 500$ testing dataset. Table 22 shows the averaged (over 500 replicates) mean square errors (MSE) using the test sample of size $N_{\text{test}} = 500$ with MSE $= \frac{1}{N_{\text{test}}} \sum_{i\in\mathcal{D}_{\text{test}}} \{Y_i - g(Z_i; \hat{x})\}^2$ across different accuracy levels ε to stop the algorithms. Table 22 shows the better prediction performance of our SMCO-BR over L-BFGS in terms of a smaller MSE and standard deviation using training and testing datasets for different accuracy levels. Interestingly, the L-BFGS algorithm under the initial SMCO-based value after 20 samplings, denoted SMCOfs_L-BFGS, generates better prediction performance in the test data than that with L-BFGS. Table 22 again shows the advantage of combining SMCOfs with existing advanced optimization algorithms to improve performance. The R package `nnet` is also considered for fitting the single hidden layer ReLU neural network but generates larger averaged MSE values for training and testing data sets, respectively.

Example 4 (Classification using neural network approximation)**.** *We generate binary response* Y *based on five-dimensional covariates* $Z = (Z_1, \cdots, Z_5)^\top$*. The true data generating process is:* $Y = 1$ *when* $\{1/[1 + \exp(2.5 - 3Z_1Z_2 - 1.5\,|Z_3| - |Z_4Z_5|]\} > 0.5$*; and* $Y = 0$ *otherwise, where* Z *follows the five-dimensional normal distribution with* $E(Z) = (1)_5^\top$ *for half of the data sets and the other half*

Table 22: Averaged (over 500 replicates) MSE (Training | Testing), standard deviation (Training | Testing) and running time (second), with $\hat{x}$ computed using SMCO-BR (without parallel), SMCOfs_L-BFGS, L-BFGS, and R package `nnet` in Example 3.

	MSE$\times 10^2$	SD$\times 10^2$	Time	MSE$\times 10^2$	SD$\times 10^2$	Time
	$N = 1000 \mid N_{\text{test}} = 500$			$N = 3000 \mid N_{\text{test}} = 500$		
training accuracy level $\varepsilon = 10^{-5}$						
SMCO-BR	**9.36 | 10.17**	**1.47 | 1.86**	51.02	**9.62 | 9.81**	**1.06 | 1.73**	118.27
SMCOfs_L-BFGS	14.78 | 15.61	4.86 | 5.44	0.07	15.15 | 15.43	4.85 | 5.36	0.14
L-BFGS	46.41 | 46.83	28.10 | 28.34	0.03	48.67 | 48.24	29.51 | 28.90	0.06
`nnet`	29.64 | 30.17	7.95 | 8.56	0.28	32.30 | 32.44	5.81 | 6.94	0.49
training accuracy level $\varepsilon = 10^{-8}$						
SMCO-BR	**9.08 | 9.62**	**1.20 | 1.76**	61.25	**9.21| 9.35**	**0.93 | 1.84**	133.84
SMCOfs_L-BFGS	13.83 | 14.67	4.90 | 5.22	0.20	14.38 | 14.66	4.69 | 4.88	0.37
L-BFGS	47.75 | 48.34	33.54 | 33.19	0.05	46.58 |46.59	28.99 |28.81	0.10
`nnet`	30.15 | 30.69	9.20 | 9.84	0.28	32.72 | 32.97	8.25 | 9.16	0.49

of the data sets using $E(Z) = (0)_5^\top$ *with common covariance matrix* $0.5^{l-l'}$.

We generate a random sample of 1500 observations from the distribution of (Y, Z)*, with* $N = 1000$ *as the training sample and the remaining* $N_{test} = 500$ *as the test sample. We consider a single-layer neural network* $g(Z, x)$ *with three hidden units and one output to approximate the true unknown* $E[Y \mid Z]$*. Then the parameter vector* x *consists of 22 weights and biases* $\{\omega_1, b_1, \omega_2, b_2\}$ *with* $\omega_{1,5\times 3}, b_{1,3\times 1}, \omega_{2,3\times 1}, b_{2,1\times 1}$ *and* $o_k = \max\left(0, \sum_{j=1}^5 \omega_{1kj} Z_j + b_{1k}\right)$ *for* $k = 1, 2, 3$ *and output is* $g\left(Z_i; x\right) = \text{sigmoid}\left(\sum_{k=1}^3 \omega_{2k} o_k + b_2\right)$*. We obtain the estimator* $\hat{x}$ *by maximizing the following negative quadratic loss function:*

$$f(x) = -\frac{1}{N}\sum_{i=1}^{N}\left\{Y_i - g\left(Z_i, x\right)\right\}^2, g\left(Z_i, x\right) = \text{sigmoid}\left(\sum_{k=1}^{3}\omega_{2k}\max\left(0, \sum_{j=1}^{5}\omega_{1kj}Z_{ij} + b_{1k}\right) + b_2\right).$$

For Example 4, we calculate $\hat{x}$ using three optimization algorithms: our SMCO-BR, the L-BFGS in the R package `nnet`, and the SMCOfs_`nnet`. We implement the L-BFGS in R package `nnet` using uniform distribution Unif(0,1) to generate the initial values of the elements of ω_1 and ω_2 and zero as the initial values of b_1 and b_2 and set the initial value of L-BFGS minus and plus 3 to be the lower and upper bounds respectively for SMCO-BR and SMCOfs in each replicate. SMCO-BR also considers that the number of starting points is set to 50 and the maximum number of iterations to be 300 with boosting iteration number 100. The SMCOfs_`nnet` is implemented the same way as the `nnet` except using our SMCOfs' 20th sampling point as its warm starting point. Table 23 shows the misclassification rate (MR) values on the test sample of size $N_{\text{test}} = 500$ with

Table 23: Averaged (over 200 replicates) out-of-sample mis-classification rate (MR), standard deviation (SD) and running time (seconds) with $\hat{x}$ computed using SMCO-BR (without parallel), `nnet` and SMCOfs_`nnet` in Example 4.

Method	Training Accuracy ε	MR	SD	Time
SMCO-BR	10^{-3}	**0.121**	**0.016**	19.47
	10^{-5}	**0.112**	**0.013**	75.82
	10^{-7}	**0.112**	**0.012**	146.33
SMCOfs_`nnet`	10^{-3}	0.129	0.029	0.205
	10^{-5}	0.128	0.037	0.217
	10^{-7}	0.127	0.049	0.373
`nnet`	10^{-3}	0.142	0.078	0.166
	10^{-5}	0.134	0.049	0.178
	10^{-7}	0.133	0.061	0.333

$\text{MR} = 1 - \frac{1}{N_{\text{test}}} \sum_{i \in \mathcal{D}_{\text{test}}} I\left\{Y_i = I\left[g\left(Z_i; \hat{x}\right) > 0.5\right]\right\}$ across different accuracy levels ε to stop the algorithms and take an average of MR's after 200 replicates at each accuracy level. Table 23 shows the best classification prediction performance of our SMCO-BR. Since `nnet` adequately accounts for the scalability of memory efficiency with storage of a small number of vectors and also takes advantage of the Hessian approximation for SGD iterates, its running speed is faster than that of our SMCO-BR on average. Interestingly, the SMCOfs_`nnet`, the `nnet` algorithm under SMCOfs-based initial value after 20 samplings, generates better classification performance on the test data than that of `nnet`. Table 23 again shows the advantage of combining SMCOfs (or even better SMCO) with existing advanced optimization algorithms to improve performance.

Table 24: Averaged (over 200 replicates) in-sample (training) and out-of-sample (testing) misclassification rate (MR), standard deviation (SD) and running time (seconds) with $\hat{x}$ computed in-sample (with Training Accuracy $\varepsilon = 10^{-6}$) using SMCO-R/-BR (with parallel), `nnet` and SMCOfs_`nnet` in Example 4. SMCO-R/-BR run with 50 diagonal multi-starting points, maximum iterations = 300.

Method	Dataset	MR	SD	Time (s)
SMCO-R	Test	0.1132	0.0154	25.413
	Train	0.0943	0.0085	
SMCO-BR	Test	0.1130	0.0156	30.422
	Train	0.0943	0.0084	
nnet	Test	0.1281	0.0450	0.172
	Train	0.0985	0.0460	
SMCOfs_nnet	Test	0.1266	0.0413	0.168
	Train	0.0990	0.0409	