Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

ASTRO FOR DERIVATIVE-BASED STOCHASTIC OPTIMIZATION: ALGORITHM DESCRIPTION & NUMERICAL EXPERIMENTS

Daniel Vasquez
Raghu Pasupathy

Department of Statistics
Purdue University
250 University St.
West Lafayette, IN 47907, USA

Sara Shashaani

Department of Industrial and Systems Engineering
North Carolina State University
111 Lampe Dr.
Raleigh, NC 27601, USA

ABSTRACT

Adaptive Sampling Trust-Region Optimization (ASTRO) is a class of derivative-based stochastic trust-region algorithms developed to solve stochastic unconstrained optimization problems where the objective function and its gradient are observable only through a noisy oracle or using a large dataset. ASTRO incorporates adaptively sampled function and gradient estimates within a trust-region framework to generate iterates that are guaranteed to converge almost surely to a first-order or a second-order critical point of the objective function. Efficiency in ASTRO stems from two key aspects: (i) adaptive sampling to ensure that the objective function and its gradient are sampled only to the extent needed, so that small sample sizes result when iterates are far from a critical point and large sample sizes result when iterates are near a critical point; and (ii) quasi-Newton Hessian updates using BFGS. We describe ASTRO in detail, give a sense of its theoretical guarantees, and report extensive numerical results.

1 INTRODUCTION

We consider the stochastic unconstrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x}) = \mathbb{E}[F(\mathbf{x},\xi)] = \int_\Omega F(\mathbf{x},\xi)dP(\xi), \tag{1}$$

where $f:\mathbb{R}^d\to\mathbb{R}$ is a smooth function and ξ is a random object with distribution P. The iterative algorithms we consider will assume the existence of a filtered probability space $(\Omega,\mathcal{F},(\mathcal{F}_k)_{k\geq 1},P)$ and access to an unbiased *first-order oracle*, that is, estimators $F:\mathbb{R}^d\times\Omega\to\mathbb{R}$ and $G:\mathbb{R}^d\times\Omega\to\mathbb{R}^d$ such that for each $\mathbf{x}\in\mathbb{R}^d$,

$$F(\mathbf{x},\xi_k) = f(\mathbf{x}) + \varepsilon_k(\mathbf{x},\xi_k) \text{ and } G(\mathbf{x},\xi_k) = \nabla f(\mathbf{x}) + \tilde{\varepsilon}_k(\mathbf{x},\xi_k),$$

where $\varepsilon_k(\cdot,\xi_k),\tilde{\varepsilon}_k(\cdot,\xi_k)$, $k=1,2,\ldots$ are independent and identically distributed (iid) $\mathcal{F}_k$-measurable random functions such that, almost surely (a.s.),

$$\mathbb{E}\left[\varepsilon_k(\mathbf{x},\xi_k)\,|\,\mathcal{F}_{k-1}\right]=0; \quad \mathbb{E}\left[\varepsilon_k(\mathbf{x},\xi_k)^2\,|\,\mathcal{F}_{k-1}\right]=\sigma^2(\mathbf{x});$$
$$\mathbb{E}\left[\tilde{\varepsilon}_k(\mathbf{x},\xi_k)\,|\,\mathcal{F}_{k-1}\right]=\mathbf{0}; \quad \mathbb{E}\left[\|\tilde{\varepsilon}_k(\mathbf{x},\xi_k)\|_2^2\,|\,\mathcal{F}_{k-1}\right]=\tilde{\sigma}^2(\mathbf{x}).$$

The notion of a first-order oracle is best interpreted generally. For instance, the estimators F and G could be outputs from a simulation oracle, or Monte Carlo observations from a large dataset. For the purposes of this paper solving problem (1) means identifying a sequence of stochastic iterates $(\mathbf{X}_k)_{k\geq 1}$ such that $\|\nabla_x f(\mathbf{X}_k)\|\to 0$ almost surely.

Problem (1) seems to have been posed first in the early 1950s (Robbins and Monro 1951; Kiefer and Wolfowitz 1952) and enjoyed intermittent attention throughout the ensuing decades (Kushner and Yin 2003). Interestingly, over the last decade, problem (1) has gained in prominence with the recognition that virtually all optimization problems involving large data sets in machine learning, e.g., regression, classification, clustering, sensing, matrix completion, are all conveniently posed as problem (1). See Bubeck (2015) for further details.

The "workhorse" algorithm to solve problem (1) has been Stochastic Gradient Descent (SGD), given by the recursion

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \alpha_k H_k^{-1} G(\mathbf{X}_k, \xi_k). \tag{2}$$

In (2), H_k^{-1} is an approximation to the inverse Hessian $\nabla_{xx} f(\mathbf{X}_k)^{-1}$, and $(\alpha_k)_{k\geq 1}$ is the sequence of step sizes chosen so that $\alpha_k \geq 0$, $\sum_{k=1}^{\infty} \alpha_k = \infty$, $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$. We do not go into further detail but note that the SGD iteration's prominence is principally due to its simplicity and the fact that when, for instance, f is smooth and strongly convex with unique minimizer $\mathbf{x}^*$ and $\alpha_k H_k^{-1} := k^{-1} \nabla_{xx} f(\mathbf{x}^*)^{-1}$, the resulting iteration satisfies

$$\sqrt{k}\,(\mathbf{X}_k - \mathbf{x}^*) \xrightarrow{\mu} \mathcal{N}\left(0, \nabla_{xx} f(\mathbf{x}^*)^{-1}\, \Sigma(\mathbf{x}^*)\, (\nabla_{xx} f(\mathbf{x}^*)^{-1})^T\right), \tag{3}$$

where $\xrightarrow{\mu}$ denotes convergence in distribution and $\Sigma(\mathbf{x}^*)$ is the covariance matrix associated with $G(\mathbf{x}^*, \xi)$. Importantly, the limit in (3) signifies the attainment of the Cramér-Rao lower bound, implying that SGD has reached theoretical limits of asymptotic algorithm performance (Nesterov 2003; Toulis et al. 2016).

Notwithstanding the attractive theoretical properties, a number of criticisms have been raised about SGD over the years. Chiefly, notice that the optimal parameter choice within SGD depends on unknown curvature constants. Moreover, it has been shown (Nemirovskii et al. 2009) compellingly that mis-estimation of the optimal step choice, specifically, the inverse of the Hessian at $\mathbf{x}^*$, can result in a rapid deterioration of the convergence rate. This last observation is far from just being theoretical; instead, it is generally believed that "playing with the step size" α_k is needed to get SGD to work well in practice.

1.1 Trust-Region Methods

Our investigation in this paper is motivated by the worldview that problem (1) is best solved by appropriately "stochasticizing" the best performing algorithms in the deterministic context. We use the word *stochasticizing* in a loose sense, to mean adding and modifying elements of a deterministic solver in such a way as to facilitate tackling problem (1). This worldview is strongly supported by practice, where our experience has been that the best performing algorithms for solving problem (1) tend to be well established deterministic solvers that are appropriately modified to account for the sampling variability of the estimators F and G.

Accordingly, our focus in this paper is a stochasticized version of the *trust-region method* (Conn et al. 2000), which is arguably amongst the most successful iterative techniques to solve deterministic versions of problem (1).

The (deterministic) trust-region method is easily stated and understood. During each iteration k, a first-order or a second-order local approximation of the objective function f is constructed around an incumbent solution $\mathbf{x}_k$. The approximating function is then *imprecisely* minimized within a region called the *trust region*, which is usually a ball centered on the incumbent $\mathbf{x}_k$. The resulting imprecise minimizer $\tilde{\mathbf{x}}_k$ of this trust-region subproblem is then accepted as a new incumbent if a simply computed number called the *success ratio* is large enough. The success ratio also determines how to update the trust-region, e.g., expand or contract, for use during the subsequent iteration. The procedure is then repeated to generate a sequence of iterates that converges to a first-order critical point under mild conditions on the constructed function approximation and if f is twice continuously differentiable.

The trust-region method seems to have been developed in parallel within the optimization and the statistics communities, with the latter under the name *ridge analysis*. The method was originally conceived in the early 1940s as a way to stabilize Newton's method for solving nonlinear least squares problems. See Trosset (2011) for an interesting commentary on this history, especially on parallel development within

the statistics community. Since the 1970s, due to the work of Powell (1970) and others (Trosset 2011), the trust-region method has become remarkably popular and evolved to be amongst the most stable techniques for solving deterministic versions of problem (1).

1.2 Stochastic Trust-Region Methods

The challenge in creating stochastic versions of the trust-region method stems from the fact that the objective function $f(\cdot)$ and its gradient $\nabla f(\cdot)$ can no longer be observed without error in the context of problem (1). Sampling to construct estimators of f and ∇f is the obvious remedy to this impasse. However, sampling is usually computationally burdensome, leading one to ask how much sampling is really adequate? Too much sampling will lead to computational inefficiency, while too little might lead to highly spurious estimates that result in non-convergence. Hence, effective sampling within trust-region methods should estimate the accuracy of the obtained function and gradient estimates, and accordingly make decisions on the adequacy of sampling.

There have been a number of recent attempts at constructing stochastic versions of the trust-region method. For example, the Basic Trust-Region with Dynamic Accuracy algorithm (Bastin et al. 2006) employs variable sample sizes for estimation within a trust region framework, with an upper finite limit on the sample size. The method is shown to generate iterates that converge almost surely to a first-order and a second-order critical point under stringent conditions. STRONG or Stochastic Trust-Region Response-Surface Method (Chang et al. 2013) is a more general trust-region framework that generates iterates attaining almost surely convergence by (i) assuming a normally distributed error on the function evaluations, and (ii) two hypotheses tests designed to accept or reject a new iterate. STRONG-X (Chang and Wan 2009) relaxes the parametric assumption in STRONG but assumes that the errors are additive with bounded variance. STORM or Stochastic Trust-Region Method with Random Models (Chen et al. 2018) is a more recent algorithm that is based on assumed access to a random model of specified accuracy within the trust-region framework. Blanchet et al. (2019) provide the complexity results for STORM, identifying the expected number of iterations required to reduce the norm of the gradient below $\varepsilon \in (0,1)$, yielding $\mathcal{O}(\varepsilon^{-2})$ and $\mathcal{O}(\varepsilon^{-3})$ for first-order and second-order Taylor models, respectively.

1.3 Notation and Convention

We now list important notation that we will heavily use in the ensuing sections.

1. We let $f(\mathbf{x},n)$ denote the sample mean of n iid function estimates at $\mathbf{x}$, that is, $f(\mathbf{x},n) := n^{-1}\sum_{j=1}^{n} F(\mathbf{x},\xi_j)$. Likewise, we let $g(\mathbf{x},n)$ denote the sample mean of n iid gradient estimates at $\mathbf{x}$, that is, $g(\mathbf{x},n) := n^{-1}\sum_{j=1}^{n} G(\mathbf{x},\xi_j)$.
2. Throughout the paper, we use bold font for vectors, lowercase font for real numbers, and uppercase font for random variables. Hence $\{\mathbf{X}_k\}$ denotes a sequence of random vectors in $\mathbb{R}^d$ and $\mathbf{x} = (x_1,\ldots,x_d)$ denotes a vector in $\mathbb{R}^d$. We let $\|\cdot\|_2$ denote the ℓ_2-norm. So, for a vector $\mathbf{x} = (x_1,x_2,\ldots,x_d) \in \mathbb{R}^d$, $\|\mathbf{x}\|_2 := (\sum_{j=1}^{d} x_i^2)^{\frac{1}{2}}$. For a $d \times d$ matrix A, $\|A\| = \sigma_{\max}(A)$ is the square root of the largest eigenvalue of $A^T A$.
3. $\mathbf{x}^*$ is called a *first-order critical point* of a differentiable function f if $\|\nabla f(\mathbf{x}^*)\| = 0$; $\mathbf{x}^*$ is called a *second-order critical point* of a twice differentiable function f if $\|\nabla f(\mathbf{x}^*)\| = 0$ and the $d \times d$ matrix $\nabla_{xx} f(\mathbf{x}^*)$ is positive definite.

2 ASTRO ALGORITHM

We now present the Adaptive Sampling Trust-Region Optimization (ASTRO) — a stochastic trust-region algorithm that employs *adaptive sampling*, that is, "on the fly" function and gradient estimation for model construction and related operations within the trust-region framework to ensure efficiency. ASTRO closely follows its derivative-free counterpart — ASTRO-Derivative Free (Shashaani et al. 2018).

Broadly speaking, ASTRO generates its incumbent iterates $(\mathbf{X}_k)_{k\geq 1}$ as follows. During iteration k, ASTRO *constructs* a local (convex) quadratic approximation $M_k(\cdot)$ of the objective function f centered at $\mathbf{X}_k$. The quadratic approximation is then minimized *approximately*, within a *trust region* $\mathcal{B}_k$ centered on $\mathbf{X}_k$ and having radius Δ_k, to obtain a trial point $\tilde{\mathbf{X}}_k$. The trial point $\tilde{\mathbf{X}}_k$ is either accepted or rejected based on a *success ratio* $\hat{\rho}_k$ that in a sense quantifies how well $M_k(\cdot)$ approximates $f(\cdot)$. If the success ratio is larger than a specified constant η_1, the trial point $\tilde{\mathbf{X}}_k$ is accepted, and it becomes the next incumbent $\mathbf{X}_{k+1}$; furthermore, if the trial point is accepted, the trust region radius Δ_k is expanded by a factor (as a vote of confidence for the model) if the iteration is deemed *very successful*, that is, the success ratio is larger than another specified constant η_2. On the other hand, if the success ratio is not large enough, the trial point is rejected, and the incumbent $\mathbf{X}_k$ is not updated. In such a case, the trust region radius is shrunk by a factor to reflect the idea that the model is not working adequately well and it needs to be constructed in a smaller region. This concludes the $k+1$-th iteration, yielding a new incumbent $\mathbf{X}_{k+1}$ and a new trust region $\mathcal{B}_{k+1}$ having the updated radius Δ_{k+1}. The process is then repeated.

The above description of ASTRO encapsulates the following four repeating steps in Algorithm 1.

1. Construct local model $M_k(\cdot)$: Step 1 and Step 2 in Algorithm 1.
2. Approximately optimize local model in a trust region to obtain a trial point $\tilde{\mathbf{X}}_k$: Step 3 in Algorithm 1.
3. Calculate success ratio $\hat{\rho}_k$ and update incumbent: Step 4 in Algorithm 1.
4. Update trust region radius: Step 5 in Algorithm 1.

The latter three of the above steps are identical to well-established theory from deterministic trust-region methods, as detailed beautifully in Chapter 6 of Conn et al. (2000). For example, *approximately* optimizing $M_k(\cdot)$ in $\mathcal{B}_k$ means finding an approximate Cauchy point $\mathbf{X}_k + \mathbf{S}_k$ that satisfies the approximate Cauchy condition

$$M_k(\mathbf{X}_k) - M_k(\mathbf{X}_k + \mathbf{S}_k) \geq \kappa_{mdc} \|g(\mathbf{X}_k, N_k)\| \min\left\{ \frac{\|g(\mathbf{X}_k, N_k)\|}{1 + \|B_k\|}, \Delta_k \right\},$$

appearing in Step 4 of Algorithm 1. Other ideas of solving the trust region sub-problem are possible but we do not pursue them here — again, see Chapter 7 of Conn et al. (2000) for more. Similarly, Step 4 and Step 5 in Algorithm 1 are identical to the basic deterministic trust-region algorithm.

Much of the complication involving the construction of a stochastic trust-region analogue thus lies in Steps 1 and 2 of Algorithm 1. Unlike the deterministic context where the function oracle gives exact values $f(\mathbf{x}), \nabla f(\mathbf{x})$ at any requested point $\mathbf{x} \in \mathbb{R}^d$, much of the challenge to efficiency in the stochastic context depends on the appropriate choice of the sample size $N(\mathbf{X}_k)$. Informally, the sample size N_k should be small when the incumbent $\mathbf{X}_k$ is such that $\|\nabla f(\mathbf{X}_k)\|$ is much larger than zero; and large when $\mathbf{X}_k$ is such that $\|\nabla f(\mathbf{X}_k)\|$ is close to zero. Of course the notions of "large" and "close to" in the previous sentence are ill-defined and the sample sizing rule

$$N_k := \min\left\{ n \geq \lambda_k : \frac{\max\{\hat{\sigma}_n(\mathbf{X}_k), \delta\}}{\sqrt{n}} \leq \theta \|g(\mathbf{X}_k, n)\| \right\};$$

$$\hat{\sigma}_n^2(\mathbf{X}_k) := \mathrm{Tr}\left(\frac{1}{n-1} \sum_{i=1}^{n} (G(\mathbf{X}_k, \xi_i) - g(\mathbf{X}_k, n))(G(\mathbf{X}_k, \xi_i) - g(\mathbf{X}_k, n))^T \right),$$

appearing in (4) makes this idea precise. Specifically, N_k is chosen as the smallest sample size $n \geq \lambda_k$ such that the ratio $\sqrt{n}\|g(\mathbf{X}_k, n)\| / \max\{\hat{\sigma}_n(\mathbf{X}_k), \delta\}$ exceeds the constant θ^{-1}. The ratio $\sqrt{n}\|g(\mathbf{X}_k, n)\| / \max\{\hat{\sigma}_n(\mathbf{X}_k), \delta\}$ should be reminiscent of the Student's T ratio (modulo the "max" operation), and is intended to keep the sampling variability, as connoted by $\hat{\sigma}_n(\mathbf{X}_k)/\sqrt{n}$, in "lock step" with $\|g(\mathbf{X}_k, n)\|$ which measures the proximity of $\mathbf{X}_k$ to a first-order critical point. The lower bound sequence λ_k is to protect against the sample size spuriously becoming too small. Our choice of N_k in intended to produce small sample sizes during early iterations when the incumbent point $\mathbf{X}_k$ is likely to have a large gradient norm, and large sample sizes during later iterations when $\mathbf{X}_k$ is likely to have a small gradient norm.

Algorithm 1 ASTRO Main Algorithm

Require: Initial point $\mathbf{X}_0 \in \mathbb{R}^d$, initial and upper bound of trust-region radii $\Delta_0 > 0$, $\Delta_{\max} > 0$, initial sample size n_0, acceptance ratio constants $0 < \eta_1 \leq \eta_2 < 1$, expansion and contraction coefficients $0 < \gamma_1 < 1 < \gamma_2$, lower bound sequence λ_k with $k^{1+\varepsilon} = \mathcal{O}(\lambda_k)$ for some $\varepsilon > 0$, $\theta \in (0,1)$, $\kappa_{mdc} \in (0,1/2)$, $\delta > 0$ and the Hessian approximation at the initial point B_0.

for $k = 0, 1, 2, \ldots$**:**

1: *Incumbent Solution Estimation:* Observe $f(\mathbf{X}_k, N_k)$ and $g(\mathbf{X}_k, N_k)$ where

$$N_k := N(\mathbf{X}_k) = \min\left\{ n \geq \lambda_k : \frac{\max\{\hat{\sigma}_n(\mathbf{X}_k), \delta\}}{\sqrt{n}} \leq \theta \|g(\mathbf{X}_k, n)\| \right\}. \tag{4}$$

2: *Model Construction:* Compute the approximated Hessian B_k and form the quadratic model

$$M_k(\mathbf{X}_k + \mathbf{S}) = f(\mathbf{X}_k, N_k) + g(\mathbf{X}_k, N_k)^T \mathbf{S} + \frac{1}{2}\mathbf{S}^T B_k \mathbf{S}.$$

3: *Step Calculation:* Compute the step $\mathbf{S}_k$ such that $\|\mathbf{S}_k\| \leq \Delta_k$ and

$$M_k(\mathbf{X}_k) - M_k(\mathbf{X}_k + \mathbf{S}_k) \geq \kappa_{mdc} \|g(\mathbf{X}_k, N_k)\| \min\left\{ \frac{\|g(\mathbf{X}_k, N_k)\|}{1 + \|B_k\|}, \Delta_k \right\}.$$

4: *Trial Point Acceptance:* Observe $f(\tilde{\mathbf{X}}_k, N_k)$ where $\tilde{\mathbf{X}}_k := \mathbf{X}_k + \mathbf{S}_k$ and define

$$\hat{\rho}_k = \frac{f(\mathbf{X}_k, N_k) - f(\tilde{\mathbf{X}}_k, N_k)}{M_k(\mathbf{X}_k) - M_k(\tilde{\mathbf{X}}_k)}.$$

If $\hat{\rho}_k \geq \eta_1$, then $\mathbf{X}_{k+1} := \tilde{\mathbf{X}}_k$; otherwise $\mathbf{X}_{k+1} := \mathbf{X}_k$.

5: *Trust-Region Radius Update:* Set

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_2 \Delta_k, \Delta_{\max}\}, & \text{if} \quad \hat{\rho}_k \geq \eta_2, & \text{(very successful)} \\ \Delta_k, & \text{if} \quad \hat{\rho}_k \in [\eta_1, \eta_2), & \text{(successful)} \\ \gamma_1 \Delta_k, & \text{if} \quad \hat{\rho}_k < \eta_1. & \text{(unsuccessful)} \end{cases}$$

end for.

It is worth making the important but obvious observation that in Algorithm 1, other choices of model form $M_k(\cdot)$, and the sample size expression N_k are possible in Steps 1 and 2 of Algorithm 1. For example, one might choose a linear or a cubic Taylor polynomial approximation as the choice of $M_k(\cdot)$. Similarly, one might introduce the Hessian approximation to f at $\mathbf{X}_k$ in the expression for N_k. Of course, such choices will have consequences on the nature of convergence, e.g., with a linear model form for $M_k(\cdot)$ one cannot hope to guarantee convergence to a second-order critical point. Likewise, sample sizing rules that do not diverge with iteration will in general not produce iterates that converge. While these are important issues, we do not go into further detail here.

2.1 Analysis of ASTRO's Behavior

To understand the sort of guarantees that ASTRO could provide, it is instructive to analyze corresponding guarantees provided by the analogous algorithm in the deterministic context. Consider the "deterministic context" where the oracle on hand provides exact function and gradient values at any requested point. We thus set $N_k = 1$ and notice that the iterates $\{\mathbf{x}_k\}$ resulting from the application of ASTRO forms a deterministic sequence in $\mathbb{R}^d$. For this deterministic setting, the following result can be proved with some effort.

Theorem 1 Let the following assumptions hold.

A.1 f is twice continuously differentiable in $\mathbb{R}^d$;
A.2 f is bounded from below, that is, $\inf_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) > -\infty$;
A.3 the ℓ_2-norm of the Hessian of f is bounded, that is, $\sup_{\mathbf{x}\in\mathbb{R}^d} \|\nabla_{xx} f(\mathbf{x})\|_2 < \infty$;
A.4 the model Hessian B_k is chosen so that $\|B_k\|_2 \leq \kappa_{\text{umh}} < \infty$, where κ_{umh} is an unknown constant.

Then,

$$\lim_{k\to\infty} \|\nabla_x f(\mathbf{x}_k)\| = 0. \tag{5}$$

Let $\{\mathbf{x}_k\}$ be a sequence of iterates converging to a first-order critical point $\mathbf{x}^*$ and that $\nabla_{xx} f(\mathbf{x}^*)$ is positive definite. If in addition to the assumptions *A*.1–*A*.4, suppose that

A.5 $\lim_{k\to\infty} \|B_k - \nabla_{xx} f(\mathbf{x}_k)\| = 0$ whenever $\lim_{k\to\infty} \|\nabla_x f(\mathbf{x}_k)\| = 0$.

Then $\{\mathbf{x}_k\} \to \mathbf{x}^*$, all iterations are eventually very successful, and the trust region radius Δ_k is bounded away from zero.

Analogous to the assertion (5) in Theorem 1 we have been able to demonstrate that the stochastic sequence $\{\mathbf{X}_k\}$ of iterates generated by ASTRO satisfies

$$\lim_{k\to\infty} \|\nabla_x f(\mathbf{X}_k)\| = 0 \text{ a.s..}$$

This result makes intuitive sense since $N_k \to \infty$ a.s.. implying that the function and gradient estimates converge to their population counterparts almost surely.

What is more interesting, however, is the convergence rate behavior of ASTRO's iterates. Specifically, note that the total sampling workload incurred by ASTRO after k iterations is $W_k = \sum_{j=1}^{k} N_j$. So, it is pertinent to investigate the limiting behavior of the quantities $W_k^{-1}\|\nabla_x f(\mathbf{X}_k)\|$ and $W_k^{-1}\|\nabla_x f(\mathbf{X}_k)\|_2^2$. It seems likely that under a local strong convexity assumptions on the first-order critical point $\mathbf{x}^*$ to which the iterates converge, the quantity $W_k^{-1}\|\nabla_x f(\mathbf{X}_k)\|_2^2$ can be shown to converge to an appropriate limit. Identifying the latter limit will also establish clues as to whether ASTRO attains Cramér-Rao lower bound for stochastic optimization.

3 NUMERICAL EXPERIMENTS

In this section we report the finite-time performance of ASTRO and the SGD recursion given in (2) on a suite of unconstrained stochastic minimization problems adapted from problems compiled by Moré et al. (1981). The problems are 2 to 100 dimensional least squares of the form

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i^2(\mathbf{x}),$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is smooth, and most of the functions f_i are non-convex.

We incorporate the following noise structure to construct the function and gradient estimators.

1. Additive static structure, that is, $\varepsilon(\cdot,\xi) \sim \mathcal{N}(0,\sigma^2)$;
2. Additive structure proportional to the gradient norm, that is, $\varepsilon(\cdot,\xi) \sim \mathcal{N}(0,\sigma^2(1+\|g(\cdot)\|^2))$;
3. Additive structure inversely proportional to the gradient norm, that is, $\varepsilon(\cdot,\xi) \sim \mathcal{N}(0,\sigma^2(1+\|g(\cdot)\|^2)^{-1})$.
4. Multiplicative structure, that is, $\varepsilon(\cdot,\xi) \sim \mathcal{N}(1,\sigma^2)$.

The first three of the above settings are additive in the sense that the function estimator $F(\mathbf{x},\xi) = f(\mathbf{x}) + \varepsilon(\mathbf{x},\xi)$. In the fourth setting above, $F(\mathbf{x},\xi) = f(\mathbf{x})\varepsilon(\mathbf{x},\xi)$. In the second and third structure we investigate cases where the noise decreases and increases when approaching the first-order critical point,

respectively. Note that the variance changes in the interval $[\sigma^2, \infty)$ and $(0, \sigma^2]$ for the second and third structure, respectively. In the fourth case, the noise variability is fixed but the magnitude changes with the true function value. The same noise structures are added in vector form to the gradients to create noisy gradient values.

3.1 Algorithm Parameters and Implementation

For ASTRO we choose the following initializations for the general trust-region parameters: $\gamma_1 = 0.5$, $\gamma_2 = 2$, $\eta_1 = 0.25$, $\eta_2 = 0.75$, $\Delta_0 = 0.1$, and $\Delta_{\max} = 10^5$. This choice is based on the guidelines that appear in (Nocedal and Wright 2006). We also choose the adaptive sampling parameters in (4) as $\delta = 10^{-3}$, $\theta = 0.90$, and $\lambda_k = k^{(1+10^{-4})}$. For SGD, an adaptive learning rate of $\alpha/(1+k), \alpha = 0.001$ for $k = 0, 1, 2, \ldots$ was used. As noted in the introduction, SGD's performance tends to be sensitive to the choice of α and the choice $\alpha = 0.001$ was based on some experimentation.

An important step in ASTRO is the use and update of the Hessian approximation B_k during the model construction and step calculation of Algorithm 1. For executing the latter step, we use the BFGS (Nocedal and Wright 2006) update. Define $\mathbf{Y}_{k-1} = g(\mathbf{X}_k, N_k) - g(\mathbf{X}_{k-1}, N_{k-1})$ as the change in the estimated gradient when moving from $\mathbf{X}_{k-1}$ to $\mathbf{X}_k$, when $k-1$ is a successful iteration, and let $\mathbf{S}_{k-1}$ be the step size computed at iteration $k-1$. The BFGS update is then given by

$$B_k = B_{k-1} - \frac{B_{k-1}\mathbf{S}_{k-1}\mathbf{S}_{k-1}^T B_{k-1}}{\mathbf{S}_{k-1}^T B_{k-1}\mathbf{S}_{k-1}} + \frac{\mathbf{Y}_{k-1}\mathbf{Y}_{k-1}^T}{\mathbf{Y}_{k-1}^T\mathbf{S}_{k-1}}, \tag{6}$$

$$B_0 = \frac{\mathbf{Y}_0^T\mathbf{Y}_0}{\mathbf{Y}_0^T\mathbf{S}_0} I_d.$$

It should be noted that the above update provides a positive definite matrix only under the curvature condition $\mathbf{S}_{k-1}^T\mathbf{Y}_{k-1} > 0$. However, this condition is not guaranteed to be satisfied in the computation of the step $\mathbf{S}_{k-1}$ in Step 3 of Algorithm 1. Thus, we skip the update formula (6) when $\mathbf{S}_{k-1}^T\mathbf{Y}_{k-1} < \varepsilon = 10^{-3}$.

ASTRO and SGD are executed until a specified simulation budget is exhausted. Then the performance is recorded in terms of the optimality gap $f(\mathbf{X}_{k_{max}(i)}) - f(\mathbf{x}^*)$ where $\mathbf{X}_{k_{max}(i)}, i = 1, 2, \ldots, m$, denotes the solution returned by the i-th execution of both algorithms, and the true gradient norm $\|g(\mathbf{X}_{k_{max}(i)})\|$. We then report the median value and the interquartile range of the $m = 20$ values for each measure at every specified simulation budget.

3.2 Results

We perform the comparison test for 12 problems as listed in Table 1 and Table 2 (2 separate problems in each of the dimensions 2, 3, 4, 6, 8, 100), with all the noise structures explained in Section 3. We first investigate the sensitivity of both algorithms to different levels of variability in the noise, by changing σ^2. We observe that results for all problems and all structures of noise were consistent for $\sigma^2 = 0.1$, 1, and 10. Since the higher dimensional problems tend to magnify any effect in the performance, Figure 1 shows the effect of different values of σ on the 100-dimensional trigonometric problem up to 3,000 simulation calls, with additive static noise.

From Figure 1 we decide to choose $\sigma^2 = 1$ throughout the rest of our experiments. Next we are interested in the effect of static or dynamic noise structures. Figure 2 demonstrates the median and interquartile range (IQR) for the 100-dimensional trigonometric problem for simulations calls up to 20,000. We observe that in the smaller noise structures that are the static additive and the additive inversely proportional to the gradient norm, ASTRO outperforms SGD in optimality gap. When the noise variance is fixed but the magnitude varies (the multiplicative case), we observe that the closer we get to the optimal solution, the better the performance of ASTRO; although in the first 10,000 oracle calls SGD outperforms ASTRO and beyond that it stalls for a while. In the case of highly variable noise, that is the additive proportional to

the gradient norm, ASTRO underperforms SGD. Figures 3 and 4 show the same effect on the true gradient norm of the 100-dimensional trigonometric function.

The last two observations (in the additive proportional to the gradient norm and the multiplicative case) were much more visible in the 100-dimensional trigonometric problem. In the lower dimensional problems ASTRO was more competitive with SGD even for the largely variable cases. To demonstrate this we next list both the optimality gap and the true gradient norm of all the problems for the bottom two structures in Table 1 and Table 2, respectively.

In these tables the initial values for the associated measures are also listed, that are the same for both of the algorithms. Since the progress in both the algorithms significantly slows down after the first few hundred runs, we list the measures at $n = 500$ and $n = 20,000$ noting that the changes in the last $10,000$ is stable and small consistently for all the problems. Each cell of the tables includes the median and the interquartile range or IQR in parentheses. For problems with dimensions higher than 4, we observe that in the first 500 oracle calls ASTRO does not reach a solution as good as SGD in both performance measures. This shows a somewhat faster convergence of SGD in the first iterations. However the converges significantly slows down in SGD and we can compare the performance of all cases but one (trigonometric problem with additive noise proportional to gradient norm – as shown in the figures before) ASTRO reaches a better incumbent solution, reasonably close to the optimal value for most instances.

Table 1: The estimated median and interquartile range of the true optimality gap at a (random) returned solution of ASTRO and SGD, as a function of the total simulation budget. The statistics were computed based on 20 independent runs of ASTRO and SGD on each problem.

Dim	Problem	Initial Optimality Gap	Algorithm	Additive Noise Variance $1+\|g(\cdot)\|^2$		Multiplicative Noise	
				$n = 500$	$n = 20,000$	$n = 500$	$n = 20,000$
2	ROSENBROCK	201.77	ASTRO	1.40 (5.16)	0.05 (0.04)	0.61 (1.44)	0.04 (0.13)
			SGD	2.12 (2.03)	1.43 (2.08)	2.23 (2.70)	1.17 (2.54)
	FREUDENSTEIN & ROTH	889.70	ASTRO	59.38 (25.38)	6.28 (12.58)	55.01 (14.47)	3.66 (6.96)
			SGD	54.63 (13.99)	54.19 (13.88)	51.00 (14.24)	49.60 (12.07)
3	HELICAL	511.10	ASTRO	4.73 (257.42)	0.03 (0.06)	1.94 (3.18)	0.04 (0.20)
			SGD	14.27 (80.88)	5.87 (17.27)	6.83 (18.80)	2.84 (3.95)
	BARD	1,258.30	ASTRO	56.38 (61.09)	19.95 (21.63)	30.02 (39.87)	0.90 (21.25)
			SGD	127.14 (350.16)	106.73 (269.89)	44.79 (70.21)	37.21 (63.05)
4	WOOD	579.01	ASTRO	1.83 (9.23)	0.01 (0.01)	11.56 (16.95)	2.45 (3.29)
			SGD	12.26 (13.03)	5.39 (5.21)	20.88 (22.84)	15.45 (19.72)
	KOWALIK & OSBORNE	8.86	ASTRO	0.26 (1.10)	0.03 (0.09)	0.07 (0.11)	0.03 (0.10)
			SGD	1.43 (2.64)	1.26 (2.35)	2.44 (3.93)	2.01 (3.70)
6	BIGGS EXP6	125.43	ASTRO	12.87 (30.90)	0.27 (0.04)	1.39 (12.98)	0.30 (0.11)
			SGD	5.61 (7.90)	4.07 (6.14)	7.13 (9.14)	6.09 (7.09)
	WATSON	398.40	ASTRO	134.10 (199.54)	0.02 (0.06)	8.68 (35.47)	0.03 (0.11)
			SGD	16.51 (14.62)	12.28 (9.80)	13.82 (20.53)	8.15 (13.83)
8	EXTENDED POWELL	425.33	ASTRO	226.29 (441.22)	0.01 (0.01)	8.51 (15.15)	0.03 (0.06)
			SGD	53.33 (52.41)	25.14 (26.52)	32.38 (23.59)	14.71 (7.51)
	PENALTY II	961.60	ASTRO	325.92 (594.24)	0.01 (0.01)	1.98 (76.35)	0.02 (0.06)
			SGD	19.59 (17.61)	10.11 (7.89)	9.88 (4.78)	4.06 (3.83)
100	TRIGONOMETRIC	3,774,634.00	ASTRO	349,382.40 (149,783.30)	185,692.10 (112,289.00)	64,280.76 (68,859.61)	2.27 (4.57)
			SGD	1,380,964.00 (750,351.10)	45.30 (35.78)	29.38 (30.42)	7.08 (4.03)
	DISCRETE INTEGRAL EQ.	161.66	ASTRO	102.10 (27.28)	50.17 (33.41)	21.22 (10.73)	0.29 (0.31)
			SGD	104.40 (26.04)	102.45 (24.97)	105.45 (18.24)	103.26 (17.14)

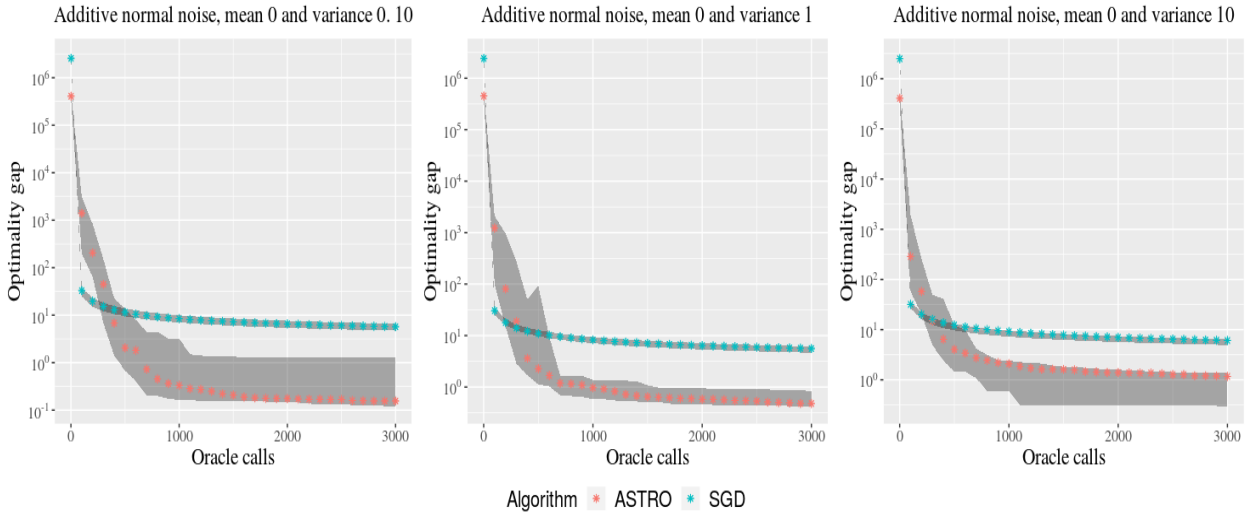

Figure 1: The median and interquartile interval of the optimality gaps of 20 independent runs for ASTRO and SGD on a 100-dimensional problem.

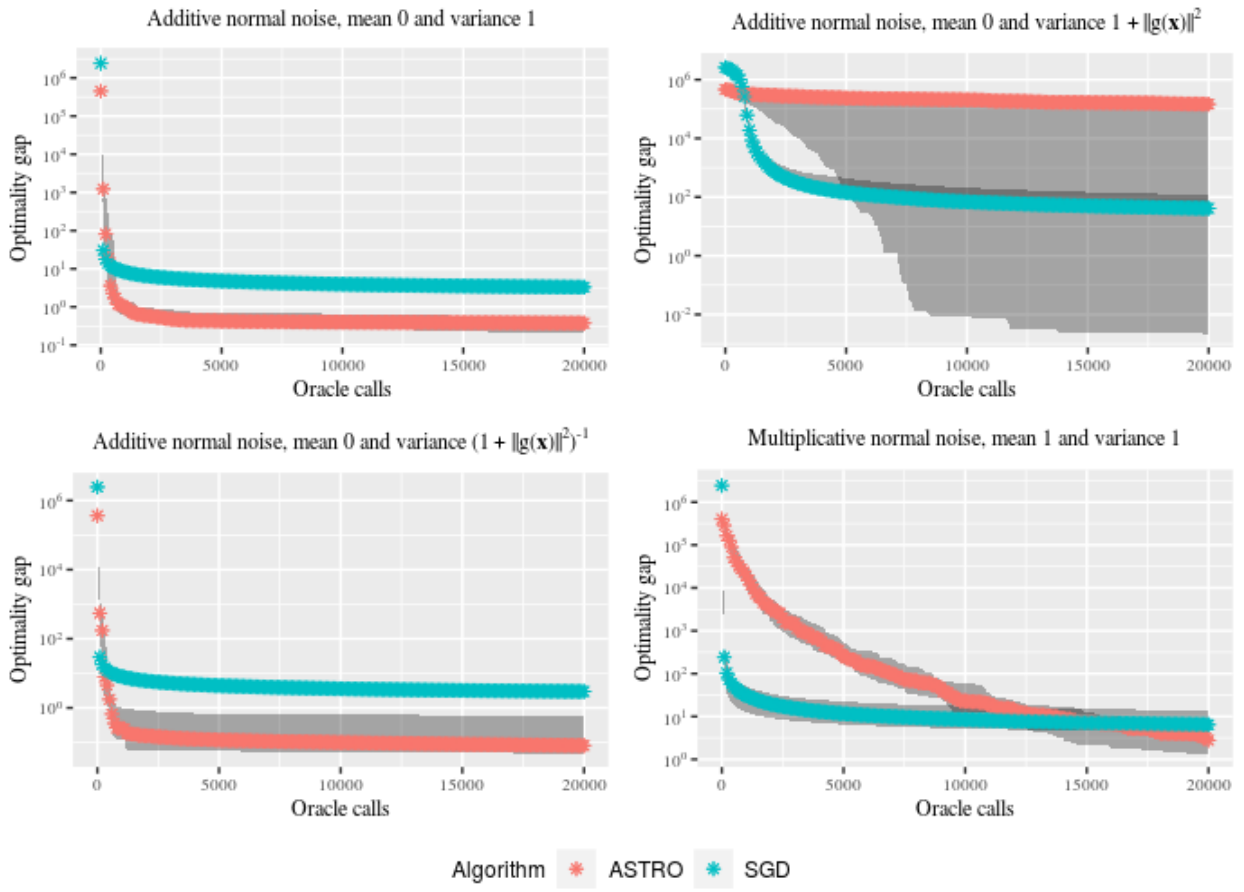

Figure 2: The median and interquartile interval of the optimality gaps of 20 independent runs of ASTRO and SGD on a 100-dimensional problem.

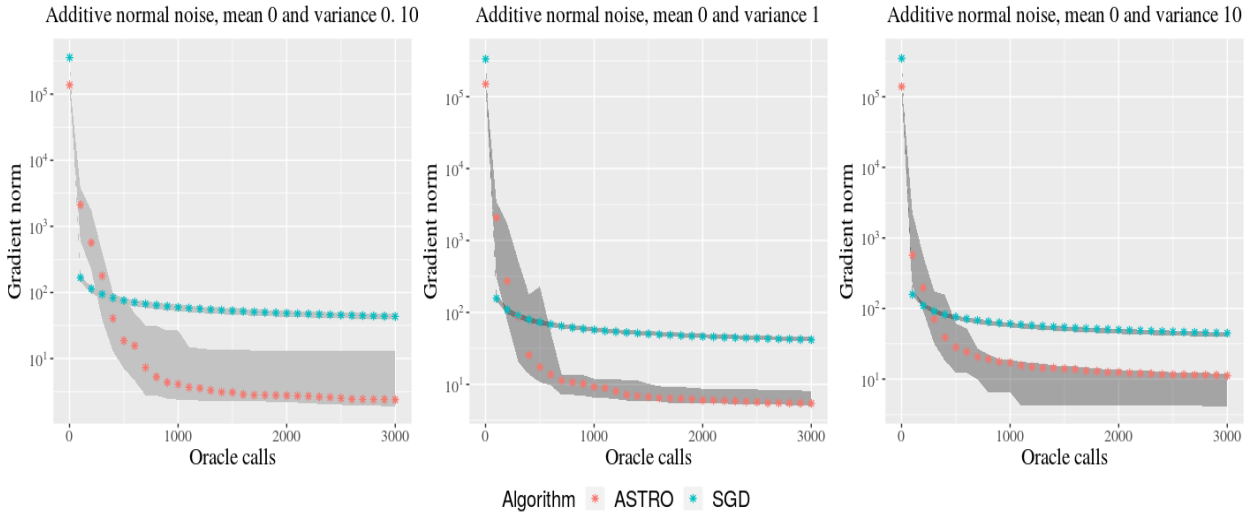

Figure 3: The median and interquartile interval of the true gradient norms of 20 independent runs for ASTRO and SGD on a 100-dimensional problem.

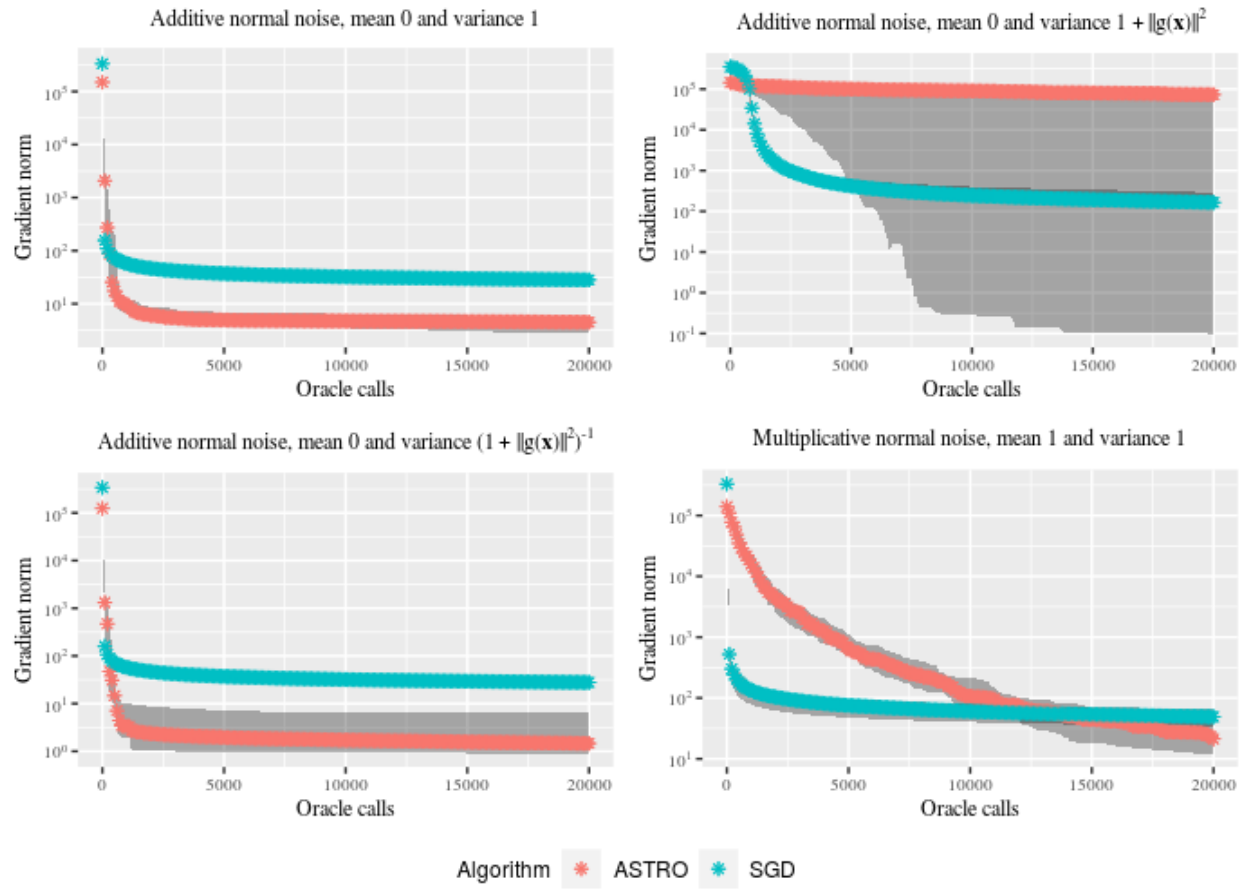

Figure 4: The median and interquartile interval of the true gradient norms of 20 independent runs of ASTRO and SGD on a 100-dimensional problem.

Table 2: The estimated median and interquartile range of the true gradient norm at a (random) returned solution of ASTRO and SGD, as a function of the total simulation budget. The statistics were computed based on 20 independent runs of ASTRO and SGD on each problem.

Dim	Problem	Initial $\|g(\cdot)\|$	Algorithm	Additive Noise Variance $1+\|g(\cdot)\|^2$		Multiplicative Noise	
				$n=500$	$n=20000$	$n=500$	$n=20000$
2	ROSENBROCK	333.06	ASTRO	2.17 (12.85)	0.30 (0.12)	2.46 (13.83)	0.32 (0.34)
			SGD	18.33 (27.76)	3.89 (9.85)	7.30 (20.34)	2.99 (3.20)
	FREUDENSTEIN & ROTH	827.20	ASTRO	19.29 (306.90)	5.21 (12.22)	14.81 (15.18)	3.08 (3.69)
			SGD	10.96 (2.36)	9.55 (2.72)	10.76 (2.11)	10.05 (1.62)
3	HELICAL	1,087.88	ASTRO	40.67 (742.40)	0.50 (0.52)	4.42 (99.62)	0.30 (0.50)
			SGD	51.87 (114.47)	19.68 (35.22)	66.26 (140.57)	26.47 (37.91)
	BARD	3,362.24	ASTRO	469.05 (2,010.12)	0.34 (0.82)	4.17 (342.30)	0.25 (0.25)
			SGD	87.84 (1,490.06)	71.89 (1,338.30)	38.66 (32.60)	29.66 (23.10)
4	WOOD	299.22	ASTRO	32.19 (104.04)	0.09 (0.10)	28.78 (39.11)	0.86 (1.38)
			SGD	58.92 (32.22)	35.17 (14.53)	41.14 (30.45)	25.27 (7.75)
	KOWALIK & OSBORNE	11.16	ASTRO	1.36 (158.02)	0.06 (0.04)	0.20 (0.35)	0.01 (0.03)
			SGD	3.36 (34.64)	3.33 (26.07)	5.29 (4.71)	4.93 (4.08)
6	BIGGS EXP6	67.55	ASTRO	18.91 (84.33)	0.15 (0.17)	7.06 (11.82)	0.09 (0.19)
			SGD	18.02 (17.96)	14.12 (14.3)	31.89 (41.09)	18.81 (21.12)
	WATSON	679.67	ASTRO	327.40 (531.90)	0.15 (0.15)	22.79 (113.21)	0.12 (0.09)
			SGD	47.77 (44.29)	32.02 (21.63)	58.73 (35.79)	36.13 (24.39)
8	EXTENDED POWELL	540.69	ASTRO	310.49 (300.88)	0.20 (0.13)	16.43 (23.47)	0.10 (0.26)
			SGD	88.08 (57.29)	47.11 (30.10)	97.31 (53.01)	52.16 (23.78)
	PENALTY II	1,320.45	ASTRO	692.82 (825.69)	0.09 (0.05)	78.88 (189.26)	0.31 (0.43)
			SGD	69.91 (31.24)	39.39 (17.54)	58.40 (27.71)	29.99 (13.99)
100	TRIGONOMETRIC	450,279.30	ASTRO	124,662.80 (22,823.18)	70,628.48 (26,998.55)	41,525.05 (26,483.16)	18.14 (27.64)
			SGD	252,336.50 (69,898.79)	180.78 (55.75)	134.56 (66.08)	44.50 (15.29)
	DISCRETE INTEGRAL EQ.	25.83	ASTRO	24.39 (6.36)	14.99 (7.01)	9.38 (2.48)	1.08 (0.50)
			SGD	24.44 (4.66)	24.06 (4.55)	23.98 (3.99)	23.61 (3.81)

REFERENCES

Bastin, F., C. Cirillo, and P. L. Toint. 2006. "An Adaptive Monte Carlo Algorithm for Computing Mixed Logit Estimators". *Computational Management Science* 3(1):55–79.

Blanchet, J., C. Cartis, M. Menickelly, and K. Scheinberg. 2019. "Convergence Rate Analysis of a Stochastic Trust-Region Method Via Supermartingales". *INFORMS Journal on Optimization* 1(2):92–119.

Bubeck, S. 2015. "Convex Optimization: Algorithms and Complexity". *Foundations and Trends in Machine Learning* 8(3–4):231–358.

Chang, K., L. Hong, and H. Wan. 2013. "Stochastic Trust-Region Response-Surface Method STRONG—A New Response-Surface Framework for Simulation Optimization". *INFORMS Journal on Computing* 25(2):230–243.

Chang, K., and H. Wan. 2009. "Stochastic Trust Region Response Surface Convergent Method for generally-distributed response surface". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 563–573. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Chen, R., M. Menickelly, and K. Scheinberg. 2018. "Stochastic Optimization Using a Trust-Region Method and Random Models". *Mathematical Programming* 169(2):447–487.

Conn, A., N. Gould, and P. Toint. 2000. *Trust-Region Methods*. MPS-SIAM Series on Optimization. Philadelphia: Society for Industrial and Applied Mathematics.

Kiefer, J., and J. Wolfowitz. 1952. "Stochastic Estimation of the Maximum of a Regression Function". *Annals of Mathematical Statistics* 23(3):462–466.

Kushner, H. J., and G. G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. 2nd ed. New York: Springer-Verlag.

Moré, J. J., B. S. Garbow, and K. E. Hillstrom. 1981. "Testing Unconstrained Optimization Software". Technical report, Argonne National Laboratories, Lemont, Illinois.

Nemirovskii, A., A. Juditsky, G. Lan, and A. Shapiro. 2009. "Robust Stochastic Approximation Approach to Stochastic Programming". *SIAM Journal on Optimization* 19(4):1574–1609.

Nesterov, Y. 2003. *Introductory Lectures on Convex Optimization: A Basic Course*, Volume 87 of *Applied Optimization*. New York: Springer Science & Business Media.

Nocedal, J., and S. J. Wright. 2006. *Numerical Optimization*. 2nd ed. New York: Springer.

Powell, M. 1970. "A New Algorithm for Unconstrained Optimization". In *Nonlinear Programming*, edited by J. Rosen, O. Mangasarian, and K. Ritter, 31 – 65. New York: Academic Press.

Robbins, H., and S. Monro. 1951. "A Stochastic Approximation Method". *Annals of Mathematical Statistics* 22:400–407.

Shashaani, S., F. S. Hashemi, and R. Pasupathy. 2018. "ASTRO-DF: A Class of Adaptive Sampling Trust-Region Algorithms for Derivative-Free Simulation Optimization.". *SIAM Journal on Optimization* 28(4):3145–3176.

Toulis, P., D. Tran, and E. Airoldi. 2016. "Towards Stability and Optimality in Stochastic Gradient Descent". In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, edited by A. Gretton and C. C. Robert, Volume 51 of *Proceedings of Machine Learning Research*, 1290–1298.

Trosset, M. 2011. "Trust Regions and Ridge Analysis". https://pdfs.semanticscholar.org/1f0e/3598b516000beb584e3b3a2a65122740e243.pdf. accessed 30th April, 2019.

AUTHOR BIOGRAPHIES

DANIEL VASQUEZ is a Ph.D. candidate in the Department of Statistics at Purdue University. His doctoral dissertation focuses on stochastic trust-region methods. His email address is dvasque@purdue.edu.

SARA SHASHAANI is an Assistant Professor of Industrial and Systems Engineering at North Carolina State University. Her research interests include Monte Carlo simulation, applied probability, optimization in machine learning, and their applications in the risk assessment of high-impact events. Her email address is sshasha2@ncsu.edu.

RAGHU PASUPATHY is an Associate Professor in the Department of Statistics at Purdue University. His research interests lie broadly in Monte Carlo methods with a specific focus on stochastic optimization. His email address is pasupath@purdue.edu. More information, including downloadable papers can be obtained through his website at https://web.ics.purdue.edu/ ~pasupath.