Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

TOWARDS GREENER STOCHASTIC DERIVATIVE-FREE OPTIMIZATION WITH TRUST REGIONS AND ADAPTIVE SAMPLING

Yunsoo Ha
Sara Shashaani

Edward P. Fitts Department of Industrial and Systems Engineering
North Carolina State University
915 Partners Way,
Raleigh, NC 27607, USA

ABSTRACT

Adaptive sampling-based trust-region optimization has emerged as an efficient solver for nonlinear and nonconvex problems in noisy derivative-free environments. This class of algorithms proceeds by iteratively constructing local models on objective function estimates that use a carefully chosen number of calls to the stochastic oracle. In this paper, we introduce a refined version of this class of algorithms that reuses the information from previous iterations. The advantage of this approach is reducing computational burden without sacrificing consistency or work complexity to attain the same level of optimality, which we demonstrate through numerical results using the SimOpt library.

1 INTRODUCTION

We pursue solving unconstrained stochastic optimization (SO) of a nonconvex, smooth, bounded-below function $f : \mathbb{R}^d \to \mathbb{R}$ defined in $\mathbb{R}^d$. The problem is of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ f(\boldsymbol{x}) := \mathbb{E}[F(\boldsymbol{x},\xi)] = \int_{\Xi} F(\boldsymbol{x},\xi) dP \right\}, \tag{1}$$

where $F : \mathbb{R}^d \times \Xi \to \mathbb{R}$ is a function defined on a probability space $(\Xi, \mathcal{F}, P)$. To estimate $f(\boldsymbol{x})$, we generate independent and identically distributed copies of the random variable $F(\boldsymbol{x},\xi)$ using a Monte Carlo simulation to obtain the sample average $\bar{F}(\boldsymbol{x},n) = n^{-1}\sum_{i=1}^{n} F(\boldsymbol{x},\xi_i)$. We also estimate the variance of the function value at $\boldsymbol{x}$ with $\hat{\sigma}_F^2(\boldsymbol{x},n) = (n-1)^{-1}\sum_{i=1}^{n}(F(\boldsymbol{x},\xi_i) - \bar{F}(\boldsymbol{x},n))^2$. We assume access to zeroth-order stochastic oracles, meaning that direct derivative information from the Monte Carlo simulation is unattainable. Consequently, to solve (1) using a model-based method such as trust-region optimization (TRO) (Conn et al. 2000), we must implicitly approximate the gradient with a local model.

TRO has gained widespread popularity as a solver for nonlinear and nonconvex problems, especially in settings with stochastic noise (Sun and Nocedal 2023; Cao et al. 2022; Chen et al. 2018; Shashaani et al. 2018; Chang et al. 2013). The random sequence of iterates $\{\boldsymbol{X}_k\}$ recommended by a single run of TRO in a stochastic setting, as described for Problem (1), leverages local approximations of the function and their minimizers within neighborhoods of dynamic sizes. In a derivative-free setting, the approximation is often done with interpolation or regression using estimated function values at adjacent points around the incumbent solution. A less explored feature of TRO is that the information from previous iterations can be reused for parsimonious model and iterate updating. In particular, design points and the observations at those points evaluated in the previous iterations can be reused to save computation. We explore the effect of intuitive "reuse" strategies within a class of adaptive sampling based TRO algorithms for derivative-free settings, called ASTRO-DF. ASTRO-DF has the ability to operate with dependence between model and function values albeit with strong consistency and convergence rates (Shashaani et al. 2018; Ha and Shashaani 2023).

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

Overview of this algorithm and standing definitions are laid out in Section 2. Reusing previous points and their queried oracle values can lead to decrease in the overall function evaluations required for each iteration; this is particularly useful in cases where evaluating the function is computationally expensive. However, reusing points may not always result in a faster convergence within the constraints of a limited budget; we will discuss this further in Section 3. In fact, there are examples to suggest that aggressive reuse of information can hinder the algorithm's consistency instead of improving its performance. The particular setting for effective reuse in ASTRO-DF discussed in Section 4 will safeguard practical consistency and efficiency with promising numerical investigations summarized in Section 5.

2 ASTRO-DF: ADAPTIVE SAMPLING TRUST REGION OPTIMIZATION DERIVATIVE-FREE

In the realm of stochastic derivative-free optimization methods, ASTRO-DF is one with proven almost sure convergence to a first-order critical point (Shashaani et al. 2018). ASTRO-DF additionally enjoys $\mathcal{O}(\epsilon^{-2})$ ϵ-optimal expected iteration complexity, defined as $T_\epsilon := \min\{k : \|\nabla f(\boldsymbol{X}_k)\| \leq \epsilon\}$ (Ha and Shashaani 2023), and $\widetilde{\mathcal{O}}(\epsilon^{-4})$ ϵ-optimal expected work complexity, defined as $W_\epsilon := \sum_{k=1}^{T_\epsilon} W_k$, with W_k tracking the total calls to the stochastic oracle during iteration k (Ha et al. 2023). Importantly, because the consistency and complexity results of ASTRO-DF do not require the independence of models, they are ideal for previous iterations' information reuse.

ASTRO-DF operates akin to its deterministic TRO counterpart. It evaluates the function at a set of design points $\mathcal{X}_k$ around the incumbent $\boldsymbol{X}_k$, fits a model $M_k(\cdot)$ on those evaluated values, and find its minimizer $\widetilde{\boldsymbol{X}}_{k+1}$ inside a trust region of size Δ_k, i.e., $\mathcal{B}(\boldsymbol{X}_k;\Delta_k)$. Different from the deterministic setting, ASTRO-DF obtains efficiency by introducing an adaptive sampling scheme that determines the appropriate number of oracle calls at each visited point. The crux of this scheme lies in allocating the computational effort based on a measure of optimality gap such as $\|\nabla f(\boldsymbol{X}_k)\|$, which ASTRO-DF tracks with the trust region radius Δ_k almost surely. As a result, more effort is dedicated to points closer to first-order critical regions. We will proved more details on the adaptive sampling rule in Section 2.2 but before that we review important definitions in Section 2.1 to facilitate the understanding of the components within ASTRO-DF.

2.1 Notation and Definition

Throughout the paper, we will use capital letters for random objects, bold font for vectors, script font for sets and sigma-fields, and san serif font for matrices.

Definition 1 (stochastic interpolation models). Let $\Phi(\boldsymbol{x}) = (\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x}), \ldots, \phi_q(\boldsymbol{x}))$ be a polynomial basis on $\mathbb{R}^d$. With $p = q = d(d+3)/2$, $\boldsymbol{X}_k^{(0)} := \boldsymbol{X}_k$, and the design set $\mathcal{X}_k := \{\boldsymbol{X}^{(i)}\}_{i=1}^p \subset \mathcal{B}(\boldsymbol{X}_k;\Delta_k)$, we find $\boldsymbol{\beta}_k = (\beta_{k,i},\ i = 1,2,\ldots,p)$ such that

$$\mathsf{M}(\Phi,\mathcal{X}_k)\boldsymbol{\beta}_k = \bar{F}(\mathcal{X}_k, N(\mathcal{X}_k)), \tag{2}$$

where

$$\mathsf{M}(\Phi,\mathcal{X}_k) = \begin{bmatrix} \phi_1(\boldsymbol{X}_k^{(0)}) & \phi_2(\boldsymbol{X}_k^{(0)}) & \cdots & \phi_q(\boldsymbol{X}_k^{(0)}) \\ \phi_1(\boldsymbol{X}_k^{(1)}) & \phi_2(\boldsymbol{X}_k^{(1)}) & \cdots & \phi_q(\boldsymbol{X}_k^{(1)}) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1(\boldsymbol{X}_k^{(p)}) & \phi_2(\boldsymbol{X}_k^{(p)}) & \cdots & \phi_q(\boldsymbol{X}_k^{(p)}) \end{bmatrix}, \bar{F}(\mathcal{X}_k, N(\mathcal{X}_k)) = \begin{bmatrix} \bar{F}(\boldsymbol{X}_k^{(0)}, N(\boldsymbol{X}_k^{(0)})) \\ \bar{F}(\boldsymbol{X}_k^{(1)}, N(\boldsymbol{X}_k^{(1)})) \\ \vdots \\ \bar{F}(\boldsymbol{X}_k^{(p)}, N(\boldsymbol{X}_k^{(p)})) \end{bmatrix}.$$

The matrix $\mathsf{M}(\Phi,\mathcal{X}_k)$ is nonsingular if the set $\mathcal{X}_k$ is poised in $\mathcal{B}(\boldsymbol{X}_k;\Delta_k)$. A set $\mathcal{X}_k$ is $\Lambda-$poised in $\mathcal{B}(\boldsymbol{X}_k;\Delta_k)$ if $\Lambda \geq \max_{i=0,\ldots,p}\max_{\boldsymbol{z}\in\mathcal{B}(\boldsymbol{X}_k;\Delta_k)}|l_i(\boldsymbol{z})|$,where $l_i(\boldsymbol{z})$ are the Lagrange polynomials. If there exists a solution to (2), then the function $M_k : \mathcal{B}(\boldsymbol{X}_k;\Delta_k) \to \mathbb{R}$, defined as $M_k(\boldsymbol{x}) = \sum_{j=0}^p \beta_{k,j}\phi_j(\boldsymbol{x})$ is a stochastic polynomial interpolation of estimated values of f on $\mathcal{B}(\boldsymbol{X}_k;\Delta_k)$. In particular, if

$\boldsymbol{G}_k := \begin{bmatrix}\beta_{k,1} & \beta_{k,2} & \cdots & \beta_{k,d}\end{bmatrix}^\intercal$ and H_k is a symmetric $d \times d$ matrix with elements uniquely defined by $(\beta_{k,d+1}, \beta_{k,d+2}, \ldots, \beta_{k,p})$, then we can define the stochastic quadratic model $M_k : \mathcal{B}(\boldsymbol{X}_k; \Delta_k) \to \mathbb{R}$, as

$$M_k(\boldsymbol{x}) = \beta_{k,0} + (\boldsymbol{x} - \boldsymbol{X}_k)^\intercal \boldsymbol{G}_k + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{X}_k)^\intercal \mathsf{H}_k (\boldsymbol{x} - \boldsymbol{X}_k). \tag{3}$$

Definition 2 (stochastic quadratic models with diagonal Hessians) A special case of (3) is when the Hessian has only diagonal values, i.e., $\mathsf{H}_k = \text{diag}(H_{k,1}, H_{k,2}, \ldots, H_{k,d}) \in \mathbb{R}^{d\times d}$. In the stochastic quadratic interpolation model with diagonal Hessian, $p = 2d$, the model (3) contains $2d + 1$ unknowns, and $2d + 1$ function value estimations are needed to uniquely determine $\boldsymbol{G}_k$ and H_k with interpolation set $\mathcal{X}_k^{cb} = \{\boldsymbol{X}_k, \boldsymbol{X}_k + \boldsymbol{e}_1\Delta_k, \ldots, \boldsymbol{X}_k + \boldsymbol{e}_d\Delta_k, \boldsymbol{X}_k - \boldsymbol{e}_1\Delta_k, \ldots, \boldsymbol{X}_k - \boldsymbol{e}_d\Delta_k\}$ contained in $\mathcal{B}(\boldsymbol{X}_k; \Delta_k)$, with $\boldsymbol{e}_i$ as the i-th unit vector and $\Phi(\boldsymbol{x}) := (1, x_1, x_2, \ldots, x_d, x_1^2, x_2^2, \ldots, x_d^2)$. With $\mathcal{X}_k^{cb}$ as the design set, $\boldsymbol{\beta}_k$ is guaranteed to exist. Hence, $H_{k,i} = \beta_{k,d+i} < \infty$ for all $i = 1, 2, \ldots, d$. In this case, $M_k(\boldsymbol{x})$ is said to be a stochastic quadratic model of f on $\mathcal{B}(\boldsymbol{X}_k; \Delta_k)$ with a diagonal Hessian.

Definition 3 (stochastic fully linear models) Given $\boldsymbol{X}_k \in \mathbb{R}^d$ and $\Delta_k > 0$, a function $M_k : \mathcal{B}(\boldsymbol{X}_k; \Delta) \to \mathbb{R}$ obtained following Definition 2 is the stochastic fully linear model of f if ∇f is Lipschitz continuous with constant κ_L, and there exist constants $\kappa_{eg} > 0, \kappa_{ef} > 0$ dependent on κ_L but independent of $\boldsymbol{X}_k$ and Δ_k such that $\mathbb{P}\left\{\|\nabla f(\boldsymbol{x}) - \nabla M(\boldsymbol{x})\| \le \kappa_{eg}\Delta, \text{ and } |f(\boldsymbol{x}) - M(\boldsymbol{x})| \le \kappa_{ef}\Delta^2 \ \forall \boldsymbol{x} \in \mathcal{B}(\boldsymbol{X}_k; \Delta_k)\right\} = 1$.

Definition 4 (filtration and stopping time). A filtration $\{\mathcal{F}_k\}_{k\ge1}$ over a probability space $(\Omega, \mathbb{P}, \mathcal{F})$ is defined as an increasing family of σ-algebras of $\mathcal{F}$, i.e., $\mathcal{F}_k \subset \mathcal{F}_{k+1} \subset \mathcal{F}$ for all k. We interpret $\mathcal{F}_k$ as "all the information available at time k." A filtered space $(\Omega, \mathbb{P}, \{\mathcal{F}_k\}_{k\ge1}, \mathcal{F})$ is a probability space equipped with a filtration. A map $N : \Omega \to \{0, 1, 2, \ldots, \infty\}$ is called a stopping time with respect to $\mathcal{F}_k$ if the event $\{N = n\} := \{\omega : N(\omega) = n\} \in \mathcal{F}_k$ for all $n \le \infty$.

2.2 ASTRO-DF Refinements

Recent developments of the ASTRO-DF algorithm have succeeded in enhancing its efficiency (Ha and Shashaani 2023) with two refinements: (1) the adoption of a quadratic model with a diagonal Hessian using the coordinate basis (see Definition 2), and (2) the incorporation of the direct search when possible. The first refinement yields a more accurate gradient estimate at $\boldsymbol{X}_k$ besides partial curvature information with $\mathcal{O}(d)$ number of design points instead of $\mathcal{O}(d^2)$ in the original version. This lower order of dependency on the problem dimension is crucial for all derivative-free solvers. Furthermore, $\mathcal{X}^{cb}$ is recognized as optimally poised for design sets of any size ranging from $d + 2$ to $2d + 1$ (Ragonneau and Zhang 2023).

The second refinement increases the likelihood of finding a better solution, which we refer to as the probability of success, without requiring an increase in the allotted budget. In cases where $\widetilde{\boldsymbol{X}}_{k+1}$ (recommended by the local model minimization) does not lead to a sufficient reduction in the estimated function values, the original strategy would declare the iteration as unsuccessful and move on to contracting the next trust region around the $\boldsymbol{X}_k$, even if the other already visited design points that have helped with the model construction may offer a good next incumbent. Considering a design point that was used for approximation as the next incumbent point is what we refer to as direct search. In principle, direct search is also an attempt to reuse information. Fewer unsuccessful iterations due to the direct search feature mean a slower rate of decay in the trust-region radius Δ_k, which is advantageous for allowing forthcoming moves with larger steps in addition to keeping the required oracles calls from growing too quickly.

ASTRO-DF with these two refinements has the following logic. At each iteration k, the model is constructed using function value estimates obtained from the incumbent solution $\boldsymbol{X}_k$ and $2d$ design points chosen based on a coordinate basis in Δ_k distance from $\boldsymbol{X}_k$. The adaptive number of Monte Carlo oracle calls at each point is a stopping time determined by striking a balance between the standard error and the measure of optimality error, i.e., $N_k := \min\{n \ge \lambda_k : \hat{\sigma}_F(\boldsymbol{X}_k, n)/\sqrt{n} \le \Delta_k^\beta/\sqrt{\lambda_k}\}$ at the center point, with a logarithmically growing deterministic lower bound λ_k in k. The power of the trust-region on the right hand side, $\beta \in [1, 2]$ can decrease with the assumption of continuous sample paths and use of

common random numbers in evaluation. Once constructed, the model recommends a candidate for the next incumbent $\widetilde{\boldsymbol{X}}_{k+1}$ that minimizes the model within the trust-region, albeit in an inexact manner. The function value estimate at $\widetilde{\boldsymbol{X}}_{k+1}$ is also obtained using the same adaptive sampling scheme. Next, the best among $2d+2$ points becomes another candidate $\hat{\boldsymbol{X}}_{k+1}$ for iteration $k+1$. If $\hat{\boldsymbol{X}}_{k+1}$ gives a sufficient reduction, i.e., $\bar{F}(\boldsymbol{X}_k, N_k) - \bar{F}(\hat{\boldsymbol{X}}_{k+1}, \hat{N}_{k+1}) \geq \alpha\Delta_k^2$ for a $\alpha > 0$, then $\hat{\boldsymbol{X}}_{k+1}$ is accepted and the trust-region expanded. Otherwise, the algorithm accepts $\widetilde{\boldsymbol{X}}_{k+1}$, the candidate obtained from the local model, as the next iterate if the success ratio, which measures the accuracy of the model's predictions of the objective function, exceeds a certain threshold η. If both criteria fail, the algorithm is deemed unsuccessful. The listing of the ASTRO-DF in Algorithm 1, i.e., history-informed ASTRO-DF, includes these refinements along with the reuse strategy that we will describe in Section 4.

Algorithm 1 History-Informed ASTRO-DF

Require: Initial guess $\boldsymbol{x}_0 \in \mathbb{R}^d$, initial and maximum trust-region radius $\Delta_0, \Delta_{\max} > 0$, model "fitness" threshold $0 < \eta_1 < \eta_2 \leq 1$, sufficient reduction constant $\alpha > 0$, expansion constant $\gamma_1 > 1$ and shrinkage constant $\gamma_2 \in (0,1)$, sample size lower bound λ_k, adaptive sampling constants $\kappa > 0$, and criticality threshold $0 < \mu$.

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

2: *Design Set Selection:* Select $\mathcal{X}_k = \{\boldsymbol{X}_k^{(i)}\}_{i=0}^{2d} \subset \mathcal{B}(\boldsymbol{X}_k; \Delta_k)$ by calling Algorithm 2:

$$\mathcal{X}_k\texttt{=PickDesignSet(}\Delta_k, \boldsymbol{X}_k, \mathcal{F}_k\texttt{)}.$$

3: *Model Construction:* Estimate $\bar{F}(\boldsymbol{X}_k^{(i)}, N\boldsymbol{X}_k^{(i)}))$, where

$$N(\boldsymbol{X}_k^{(i)}) = \min\left\{n \geq \lambda_k : \frac{\hat{\sigma}(\boldsymbol{X}_k^{(i)}, n)}{\sqrt{n}} \leq \frac{\kappa\Delta_k^2}{\sqrt{\lambda_k}}\right\}, \tag{4}$$

for $i = 0, 1, \ldots, 2d$ and construct the model $M_k(\boldsymbol{X}_k + \boldsymbol{s})$ via interpolation.

4: *Subproblem Minimization:* Approximate the k-th step by minimizing the model in the trust-region, $\boldsymbol{S}_k = \operatorname{argmin}_{\|\boldsymbol{s}\| \leq \Delta_k} M_k(\boldsymbol{X}_k + \boldsymbol{s})$, and set $\widetilde{\boldsymbol{X}}_{k+1} = \boldsymbol{X}_k + \boldsymbol{S}_k$.

5: *Candidate Evaluation:* Estimate the function at the candidate point using adaptive sampling to obtain $\bar{F}(\widetilde{\boldsymbol{X}}_{k+1}, \widetilde{N}_{k+1})$ following (4). Define the best design point $\hat{\boldsymbol{X}}_{k+1} = \operatorname{argmin}_{\boldsymbol{x} \in \mathcal{X}_k \cup \{\widetilde{\boldsymbol{X}}_{k+1}\}} \bar{F}(\boldsymbol{x}, N(\boldsymbol{x}))$, its sample size $\hat{N}_{k+1} = N(\hat{\boldsymbol{X}}_{k+1})$, sample size of incumbent $\hat{N}_k = N(\boldsymbol{X}_k)$, direct-search reduction $\hat{R}_k = \bar{F}(\boldsymbol{X}_k, \hat{N}_k) - \bar{F}(\hat{\boldsymbol{X}}_{k+1}, \hat{N}_{k+1})$, subproblem reduction $\widetilde{R}_k = \bar{F}(\boldsymbol{X}_k, \hat{N}_k) - \bar{F}(\widetilde{\boldsymbol{X}}_{k+1}, \widetilde{N}_{k+1})$, and model reduction $R_k = M_k(\boldsymbol{X}_k) - M_k(\widetilde{\boldsymbol{X}}_{k+1})$.

6: *Update:* Set $(\boldsymbol{X}_{k+1}, N_{k+1}, \Delta_{k+1}) =$

$$\begin{cases} (\hat{\boldsymbol{X}}_{k+1}, \hat{N}_{k+1}, \min\{\gamma_1\Delta_k, \Delta_{\max}\}) & \text{if } \hat{R}_k > \max\{\widetilde{R}_k, \alpha\Delta_k^2\}, \\ (\widetilde{\boldsymbol{X}}_{k+1}, \widetilde{N}_{k+1}, \min\{\gamma_1\Delta_k, \Delta_{\max}\}) & \text{else if } \widetilde{R}_k \geq \eta_2 R_k \text{ and } \mu\|\nabla M_k(\boldsymbol{X}_k)\| \geq \Delta_k, \\ (\widetilde{\boldsymbol{X}}_{k+1}, \widetilde{N}_{k+1}, \Delta_k) & \text{else if } \widetilde{R}_k \geq \eta_1 R_k \text{ and } \mu\|\nabla M_k(\boldsymbol{X}_k)\| \geq \Delta_k, \\ (\boldsymbol{X}_k, \hat{N}_k, \gamma_2\Delta_k) & \text{otherwise,} \end{cases}$$

and $k = k + 1$.

7: **end for**

3 THE REUSE STRATEGY

Most stochastic trust-region methods, such as STORM (Chen et al. 2018), STRONG (Chang et al. 2013), and ASTRO-DF, have the potential to significantly reduce the computational burden of each iteration k by leveraging previously visited design points when constructing the local model. For instance, the iteration k subsequent to an unsuccessful iteration $k-1$ would require a new model M_k that is more focused within a shrunk trust region for a better approximation of the objective function (Figure 1). Evidently, two design points, namely, X_{k-1} and $\widetilde{X}_k$ (all one-dimensional design points and hence in regular font), are still present within $\mathcal{B}(X_k, \Delta_k)$ as it partially overlaps with $\mathcal{B}(X_{k-1}, \Delta_{k-1})$. These points and their replications can be reused for constructing M_k, thereby reducing W_k from $\sum_{i=0}^{p} N_k^{(i)}$ to $\sum_{i=0}^{p} N_k^{(i)} - \sum_{j\in\mathcal{R}} N_{k-1}^{(j)}$, where $\mathcal{R}$ is a reusable points set.

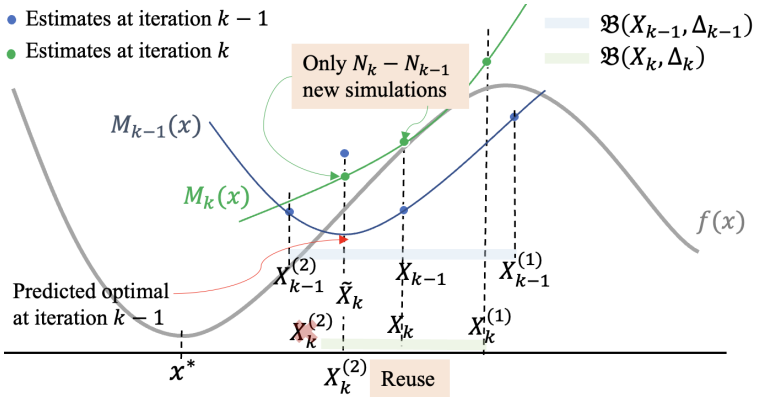


Figure 1: ASTRO-DF with the reuse strategy is exemplified in a one-dimensional problem. $f(x)$ is the true unknown function. After an unsuccessful iteration $k-1$ and shrinkage of trust region, its candidate point is reused as a design point for iteration k. This will lead to fewer simulations but still a good new model M_k that leads towards the true optimal solution x^*.

3.1 Could Reusing Points be Harmful?

As illustrated, implementing a reuse strategy can effectively reduce computational expenses during each iteration, consequently fostering improved work complexity. However, complexity improvement necessitates an additional assumption: maintaining the iteration complexity consistent with the version with no reuse. In this section, we provide evidence that the adoption of a reuse strategy may actually lead to an increase in the required number of iterations, indicating a potential worsened work complexity.

Reuse strategy in model construction has significant implications for the quality of the resulting models, as the current model's performance becomes dependent on the previous model. In cases where the previous model does not accurately represent the objective function leading to an unsuccessful iteration, the current model will likely struggle to approximate the objective function, leading to another unsuccessful iteration. To understand why the current iteration can also become unsuccessful, we delve deeper into the underlying conditions for a successful iteration. Success in iteration k can be achieved by satisfying the requirements stated in Lemma 5.2 by Shashaani et al. (2018):

$$|\bar{F}(\widetilde{\boldsymbol{X}}_{k+1}, \widetilde{N}_{k+1}) - M_k(\widetilde{\boldsymbol{X}}_{k+1})| < c\Delta_k\|\boldsymbol{G}_k\| \text{ and } \Delta_k \leq \mu\|\boldsymbol{G}_k\|,$$

where c and μ are positive constants, following the assumption on the subproblem minimizer being at least at good a Cauchy point $\boldsymbol{X}_k + \boldsymbol{S}^c$, i.e.,

$$M_k(\boldsymbol{X}_k) - M_k(\boldsymbol{X}_k + \boldsymbol{S}^c) \geq \frac{1}{2}\|\boldsymbol{G}_k\| \min\left\{\frac{\|\boldsymbol{G}_k\|}{\|\mathsf{H}_k\|}, \Delta_k\right\}. \tag{5}$$

This condition can be transformed into the following equation:

$$|\bar{F}(\widetilde{\boldsymbol{X}}_{k+1}, \widetilde{N}_{k+1}) - f(\widetilde{\boldsymbol{X}}_{k+1})| + |f(\widetilde{\boldsymbol{X}}_{k+1}) - m_k(\widetilde{\boldsymbol{X}}_{k+1})| + |m_k(\widetilde{\boldsymbol{X}}_{k+1}) - M_k(\widetilde{\boldsymbol{X}}_{k+1})| < c\mu^{-1}\Delta_k^2. \quad (6)$$

The first term of this equation is not affected by the reuse strategy since $\widetilde{\boldsymbol{X}}_{k+1}$ is always a new design point for any $k \in \mathbb{N}$. However, the second and third terms are impacted by the reuse strategy. The second term, $|f(\widetilde{\boldsymbol{X}}_{k+1}) - m_k(\widetilde{\boldsymbol{X}}_{k+1})|$, is bounded by $\kappa_{ef}\Delta_k^2$ for some positive constant κ_{ef} according to Definition 3. The value of κ_{ef} depends on the location of the design set, and since the design set following Definition 2 is the optimal design set (Ragonneau and Zhang 2023), the reuse strategy results in a larger κ_{ef}. Consequently, the probability of satisfying (6) decreases. For the third term, if iteration $k-1$ is unsuccessful then there exists a $c' > 0$ such that $|m_{k-1}(\widetilde{\boldsymbol{X}}_k) - M_{k-1}(\widetilde{\boldsymbol{X}}_k)| > c'\Delta_{k-1}^2$. As a result, $\mathbb{P}\{|m_k(\widetilde{\boldsymbol{X}}_{k+1}) - M_k(\widetilde{\boldsymbol{X}}_{k+1})| < \tilde{c}\Delta_k^2\}$ can be bounded by

$$\mathbb{P}\left\{|m_k(\widetilde{\boldsymbol{X}}_{k+1}) - M_k(\widetilde{\boldsymbol{X}}_{k+1})| < c(3\mu)^{-1}\Delta_k^2 \,\middle|\, |m_k(\widetilde{\boldsymbol{X}}_k) - M_k(\widetilde{\boldsymbol{X}}_k)| > c'\Delta_{k-1}^2\right\}. \quad (7)$$

Reusing points introduces a positive correlation between the absolute differences of $m_k(\cdot)$ and $M_k(\cdot)$ and those of $m_{k-1}(\cdot)$ and $M_{k-1}(\cdot)$. This means that the probability in (7) with the reuse strategy has a lower value compared to the one without it. These observations imply that it may become less likely to have successful iterations with the reuse strategy. This means that the decreased number of oracle calls per iteration may not necessarily result in improved work complexity.

Two key factors cause a significant impact on the algorithm's efficiency when the number of unsuccessful iterations increases. The first factor is slow progress in improving the solution, as even in the subsequent successful iterations, only a marginal reduction in the objective function is attainable. The second factor is the rapid shrinkage of trust region size and an increased number of oracle calls at least quadratically faster. The algorithm is unable to make a progress while wasting a significant portion of the budget. Besides significantly hindering the overall performance and efficiency of the algorithm, reusing points can also threaten the algorithm's consistency if the algorithm is sensitive to model dependence. For example, the almost sure convergence result of STORM (Chen et al. 2018) relies on the independence of the models and function estimates. But reuse strategy forces the models to depend on each other, directly invalidating the almost sure convergence result. In the refined ASTRO-DF, the inefficiency of reusing points is not limited to the model step, as it also hinders the effectiveness of the direct search step. Unless the function estimate at the reused points changes significantly, perhaps due to either a notable increase in the number of oracle calls at those points or an underlying high inherent variance at those points, reusing points lowers the likelihood of more exploration and escaping inferior solutions. In conclusion, care is required when implementing the reuse strategy to enhance computational efficiency in practice.

3.2 Relationship between Reusing Points and ASTRO-DF

The refined ASTRO-DF uses a local model constructed via a predetermined design set on a stencil around the incumbent solution (Definition 2). This approach entails selecting $2d$ new design points per iteration. Since $|\mathcal{X}_k^{cb}| = 2d + 1$, the only reusable point is the center point. In other words, a refined ASTRO-DF will not reuse $\widetilde{\boldsymbol{X}}_{k+1}$ (suggested by the subproblem) following an unsuccessful iteration, provided that it is contained in the contracted trust region, or the $\hat{\boldsymbol{X}}_{k+1}$ point (suggested by the direct search) following a successful iteration, provided that it is contained in the expanded trust region. Reusing more design points without care, in this case, diminishes the coordinate basis leverage as detailed in Definition 2, and renders the design set suboptimal (Ragonneau and Zhang 2023). In addition, blindly reusing already visited points could be wasteful if they failed the direct search criteria during previous iterations, signaling their inferiority and limiting the discovery of better solutions. In summary, when contemplating the utilization of reuse strategy, it is essential to evaluate whether or not the following two advantages of the refined ASTRO-DF are compromised: (a) evenly distributed Λ-poised design set with sufficiently small Λ, i.e., a well-poised set, and (b) sufficiently high probability of success through direct search.

4 HISTORY-INFORMED ASTRO-DF

In this section, we introduce a new heuristic for the refined version of ASTRO-DF to leverage the history, i.e., the visited design points and their replications. As clarified earlier, the refined ASTRO-DF cannot reuse any design points except the center point due to the predetermined design set and the dynamics of the trust region. We suggest using the design set with a rotated coordinate basis instead of $\mathcal{X}_k^{cb}$ in the refined ASTRO-DF scheme. Unlike $\mathcal{X}_k^{cb}$, the design set with the rotated coordinate basis contains unit vector $\boldsymbol{U}_{k,1}$ and its related orthonormal basis. As a result, the first new design point except $\boldsymbol{X}_k$ for the design set can be placed in the trust region flexibly with arbitrary $\boldsymbol{U}_{k,1}$, leading to one reusable design point as $\boldsymbol{X}_k + P_k\boldsymbol{U}_{k,1}$ for some $P_k \leq \Delta_k$. In this case, the design set becomes

$$\mathcal{X}_k^{rcb} := \{\boldsymbol{X}_k, \boldsymbol{X}_k + P_k\boldsymbol{U}_{k,1}, \boldsymbol{X}_k + \Delta_k\boldsymbol{U}_{k,2}, \ldots, \boldsymbol{X}_k + \Delta_k\boldsymbol{U}_{k,d}, \boldsymbol{X}_k - \Delta_k\boldsymbol{U}_{k,1}, \ldots, \boldsymbol{X}_k - \Delta_k\boldsymbol{U}_{k,d}\},$$

where $\mathcal{U}_k := \{\boldsymbol{U}_{k,1}, \boldsymbol{U}_{k,2}, \ldots, \boldsymbol{U}_{k,d}\}$ is the orthonormal basis for $\mathbb{R}^d$. Therefore, the history-informed ASTRO-DF selects its design set $\mathcal{X}_k$ following Algorithm 2. Figure 2 illustrates using $\mathcal{X}_k^{rcb}$ in the refined ASTRO-DF scheme.

Algorithm 2 $\mathcal{X}_k$=`PickDesignSet`$(\Delta_k, \boldsymbol{X}_k, \mathcal{F}_k)$

Require: Parameters from ASTRO-DF: trust-region radius Δ_k, iterate $\boldsymbol{X}_k$, and history $\mathcal{F}_k$.

1: Find the reusable points set $\mathcal{R}_k$ using $\mathcal{F}_k$ within the trust region.
2: **if** $\boldsymbol{X}_k$ is the only design point within the trust-region, i.e., $\mathcal{R}_k$ is empty **then**
3: Select the design set $\mathcal{X}_k = \mathcal{X}_k^{cb}$ following Definition 2.
4: **else**
5: Pick the farthest point from $\boldsymbol{X}_k = \boldsymbol{X}_k^{(0)}$ as $\boldsymbol{X}_k^{(1)}$, which implies obtaining P_k and $\boldsymbol{U}_{k,1}$.

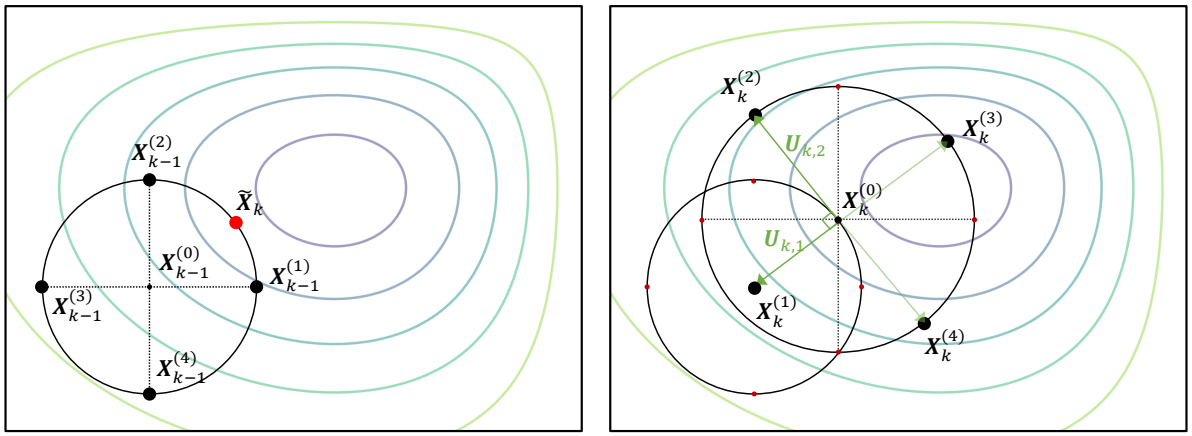
$$\boldsymbol{X}_k^{(1)} = \operatorname*{argmax}_{\boldsymbol{x}\in\mathcal{R}_k} \|\boldsymbol{X}_k - \boldsymbol{x}\|_2 = \boldsymbol{X}_k + P_k\boldsymbol{U}_{k,1}.$$

6: Obtain $\mathcal{U}_k$ based on $\boldsymbol{U}_{k,1}$ and select $\mathcal{X}_k = \mathcal{X}_k^{rcb}$.
7: **end if**
8: **Return** $\mathcal{X}_k$.

The suggested design set gives two advantages. Firstly, it can enjoy the more accurate directional derivative estimate $\nabla_{\boldsymbol{U}_{k,i}} M_k(\boldsymbol{X}_k)$ for $i \in \{2, \ldots, d\}$ like the optimal design set, i.e., $\mathcal{X}^{cb}$, where $\nabla_{\boldsymbol{U}_{k,i}} M_k(\boldsymbol{X}_k) = \boldsymbol{G}_k\boldsymbol{U}_{k,i}$. Since it reminisces the central finite difference with the new direction instead of the standard basis, $\nabla_{\boldsymbol{U}_{k,i}} M_k(\boldsymbol{X}_k)$ for $i \in \{2, \ldots, d\}$ can achieve $\mathcal{O}(\Delta_k^2)$ accuracy. Although, for $i = 1$, the directional derivative estimate only achieves $\mathcal{O}(\Delta_k)$ accuracy, it is still enough to obtain consistency, i.e., the almost sure convergence to the first-order stationary point (Shashaani et al. 2018). The accuracy of $\nabla_{\boldsymbol{U}_{k,1}} M_k(\boldsymbol{X}_k)$ depends on P_k, e.g., when $P_k = \Delta_k$, $\nabla_{\boldsymbol{U}_{k,1}} M_k(\boldsymbol{X}_k)$ achieves $\mathcal{O}(\Delta_k^2)$ accuracy. This is why the algorithm picks the farthest reusable design point from $\boldsymbol{X}_k$. Secondly, reusing two design points reduces the number of function evaluations for iteration k without threatening the advantage of the direct search method. Moreover, reusing only one more point in addition to the incumbent design does not significantly impact the probability of success with the direct search and the number of new design points, as elaborated in Section 3.2. Therefore this heuristic ensures negligible harm while saving budget. Lastly, to aid in the understanding of the different versions of ASTRO-DF, we have prepared a succinct summary highlighting the key distinctions among them. This overview is outlined in Table 1.

Vanilla ASTRO-DF: A random design set is selected within the trust-region as long as it forms a poised set. This allows for the possibility of reusing some of the design points from previous iterations if they lie within the new trust region. After the model construction, the candidate point is obtained implicitly by minimizing the model within the trust region.

Refined ASTRO-DF: The design set is selected in a deterministic manner by utilizing the coordinate basis (see Figure 2a), choosing $2d$ new points per iteration. The only reused point is the center point from the previous iteration. To update the incumbent, we minimize the model within the trust region to find a candidate point. If the candidate point is not better than the design points, we replace it with the best point found in that region.



(a) iteration $k-1$ (b) iteration k

Figure 2: Figure 2a shows the case of using the coordinate basis due to the absence of reusable design points within the trust region. Figure 2b shows the case of using the rotated coordinate basis. In this particular scenario (iteration k), the design point $\boldsymbol{X}_{k-1}$, being the farthest from $\boldsymbol{X}_k$ among the reusable design points, is *reused* as $\boldsymbol{X}_k^{(1)}$.

Table 1: Differences between ASTRO-DFs.

Algorithm	Vanilla ASTRO-DF	Refined ASTRO-DF	History-informed ASTRO-DF
Design set ($\mathcal{X}_k$) selection	Random	Coordinate basis	Rotated coordinate basis
$\lvert\mathcal{X}_k\rvert$	$(d+1)(d+2)/2$	$2d+1$	$2d+1$
Updating next incumbent	Model	Model + Direct Search	Model + Direct Search
# of possible reusing points	≥ 0	1	2
Increasing rate of λ_k in (4)	linearly	logarithmically	logarithmically

History-informed ASTRO-DF: The selection of the design set involves utilizing a rotated coordinate basis (see Figure 2b). By employing this method, the center point and another design point can be reused. Updating rule remains the same as the refined ASTRO-DF. The changes are notwithstanding the almost sure convergence to a first-order critical point, as formally stated below.

Theorem 1 Let $F(\boldsymbol{x},\xi)-f(\boldsymbol{x})$ exhibit a subexponential tail behavior with $\mathrm{Var}(F(\boldsymbol{x},\xi)-f(\boldsymbol{x}))\leq\sigma^2$ for all $\boldsymbol{x}\in\mathbb{R}^d$. Suppose further that function f is twice continuously differentiable in an open domain $\mathcal{X}\subset\mathbb{R}^d$ containing $\mathcal{B}(\boldsymbol{x}_0;\Delta_{\max})$ and ∇f is Lipschitz continuous in $\mathcal{X}$ with constant $\kappa_{Lg}>0$. Then, if model reduction R_k attains at least a fraction of the Cauchy decrease, as defined in (5), and the model Hessian terms $\|\mathsf{H}_k\|\leq\kappa_{\mathsf{H}}$ for all k and some $\kappa_{\mathsf{H}}>0$ with probability 1, then the sequence $\{\boldsymbol{X}_k\}$ of iterates generated by the history-informed ASTRO-DF satisfies $\lim_{k\to\infty}\|\nabla f(\boldsymbol{X}_k)\|=0$ almost surely.

5 NUMERICAL RESULTS

We now provide an account of the finite-time performance of solvers for a range of problems, including Stochastic Activity Network (SAN), a 13-dimensional convex problem, and the (s, S) inventory problem, a 2-dimensional nonconvex problem, sourced from the SimOpt library (Eckman et al. 2023). SimOpt is a testbed and benchmarking platform with two distinct procedures. Initially, for each solver and problem, we carry out $m = 20$ macro-replications. During each macro-replication, each solver is given the same predetermined budget unique to that problem. Solvers estimate the objective function at each solution $\boldsymbol{x}$ with a certain number of replications to decide the next move. In their computations, adaptive solvers employ a variable replication size $N(\boldsymbol{x})$. The second procedure involves conducting $n = 200$ post-replications at each macro-replication's intermediate (recommended) solutions to produce objective function estimates free of any optimization bias. Common random numbers ensure a fair comparison and reduce variance for both macro-replications and post-replications. Our assessment of solver performance involves analyzing the confidence intervals associated with their generated solutions' post-replicated function estimates for insights into their behavior's consistency.

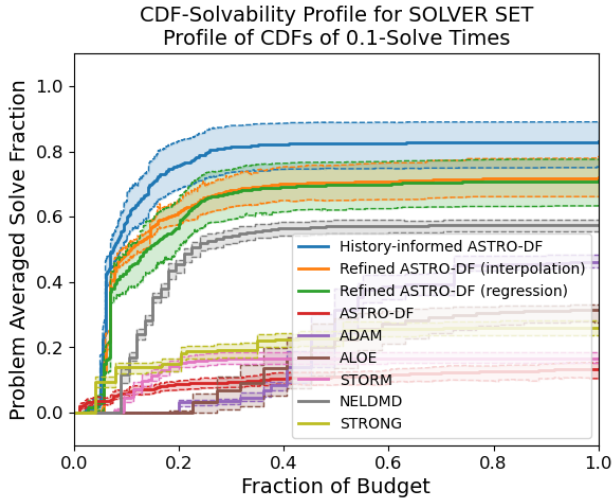


Figure 3: Solvability Profile using 60 problems from the SimOpt library. Y-axis depicts the percentage of test problems in which solvers reach the budget it takes to reach 90% optimality, .e.g., the refined ASTRO-DF is observed to solve roughly 60% of the problems while using only 20% of the available budget. The error bars represent 95% confidence region.

For both ASTRO-DF and STORM algorithms we use $\mu = 1000, \eta_1 = 0.1, \eta_2 = 0.5, \gamma_1 = 0.75$, and $\gamma_2 = 1.5$. We adjust the value of the scaling parameter in (4) at the initial iteration of each macro-replication as $\kappa = \bar{F}(\boldsymbol{X}_0, n_0)/\Delta_0^2$. To establish $\Delta_{\max}$ for each macro-replication, we generate random solutions using an inherent solution generator in the problem (property of SimOpt) and compute the largest distance between them. We then fine-tune the Δ_0 through a pilot run using three candidates: $0.05\Delta_{\max} \times \{0.1, 1, 10\}$ and allocate 1% of the total budget to each candidate. This dynamic tuning method enables us to adapt the scaling of $\Delta_0, \Delta_{\max}$, and κ according to the current macro-replication. We then compare the history-informed ASTRO-DF with other solvers in the SimOpt library, including Nelder-Mead (NELDMD), ALOE (Jin et al. 2021), ADAM (Kingma and Ba 2017), and STORM (Chen et al. 2018).

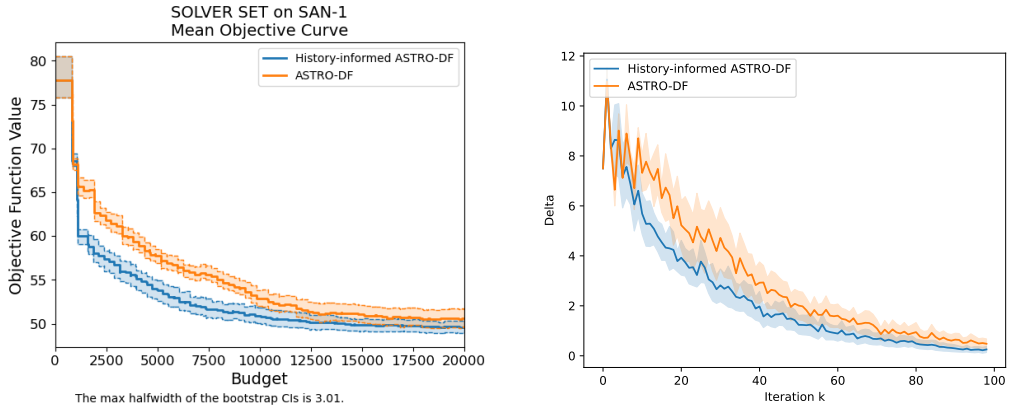
5.1 Effect of the Reuse Strategy

Next, we investigate the impact of reusing points in comparing history-informed ASTRO-DF and ASTRO-DF. We first test the two algorithms on SAN. Figure 4 displays the progress of the objective function value

and the trust region radius. The history-informed ASTRO-DF can identify better solutions using fewer function evaluations (Figure 4a). Notably, Δ_k in the history-informed ASTRO-DF diminishes more rapidly (Figure 4b), evidencing fewer successes yet with likely a more substantial decrease in the objective function value at each successful iteration. This behavior can be attributed to the acceptance of reused points as new iterates (via direct search) moving along the rotated coordinate bases, representing the direction that yielded the most promising reduction in the previous iteration. This ultimately contributes to faster convergence in finite time. Our second test is on a stochastic variant of Rosenbrock function suggested by Kim and Zhang (2010) as

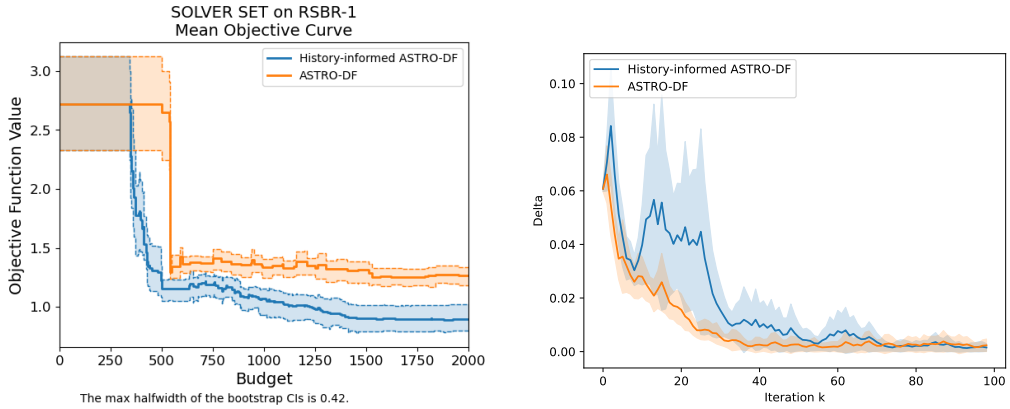
$$F(\boldsymbol{x}, \boldsymbol{\xi}) = 100\left(x_2 - \xi x_1^2\right)^2 + (\xi x_1 - 1)^2,$$

where $\xi \sim \mathcal{N}(1, 0.1)$. The result (Figure 5a) is commensurate with the previous case; the history-informed ASTRO-DF demonstrates faster convergence and a higher success rate in the first 40 iterations (Figure 5b) indicating that reuse strategy and the design set with the rotated coordinate basis lead to better solutions, even when dealing with the nonconvex problem.



(a) Objective Function Value (b) Δ_k

Figure 4: History-informed ASTRO-DF finite-time convergence improves in a Stochastic Activity Network.

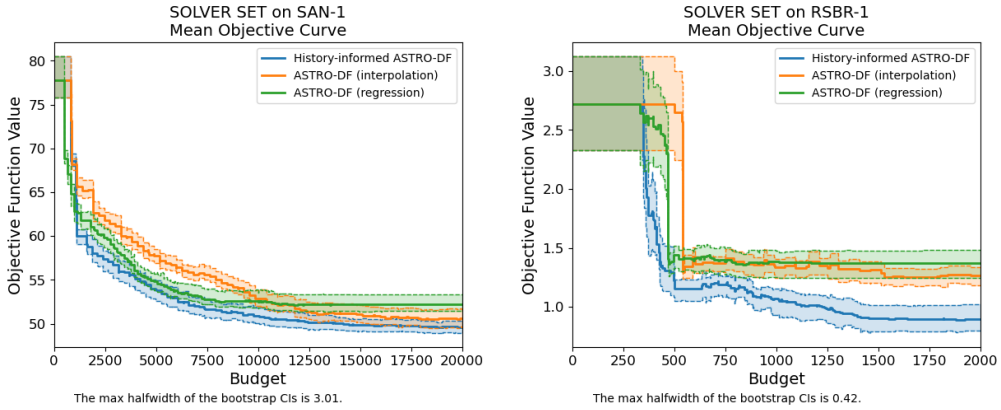


(a) Objective Function Value (b) Δ_k

Figure 5: History-informed ASTRO-DF converges faster in a stochastic variant of the Rosenbrock function.

5.2 Comparison Between Regression and Interpolation

The history-informed ASTRO-DF (Algorithm 1) utilizes interpolation for local models. However, regression models might present a natural approach for reusing all previous information. In Figure 2b, for instance, within the trust region, all design points, including $\boldsymbol{X}_{k-1}^{(i)}$ for $i \in \{0,1\}$ and $\boldsymbol{X}_{k}^{(j)}$ for $j \in \{0,1,2,3\}$, can be used through regression. While utilizing regression models may not directly affect the direct search method, it can still significantly impact the quality of the local model, which is crucial in guiding the optimization process toward the optimal solution. Using regression in the optimization process naturally raises the question of whether it can provide a better local approximation than interpolation. However, in derivative-free optimization, theoretically establishing whether or not regression can provide a superior gradient estimate compared to interpolation remains to be seen (Section 2.3 in (Conn et al. 2009)). In practice, regression models have been found to be useful (Wild et al. 2008), thereby revealing a gap between theory and practice. When dealing with a noisy environment, the situation tends to become considerably more intricate. One of the primary reasons is that the function estimates obtained from previous iterations may be inaccurate, resulting in even worse derivative estimations at the current iteration. As a result, the optimization process for ASTRO-DF with regression might deteriorate, as illustrated in Figure 6 for SAN and the stochastic Rosenbrock function.



(a) SAN (b) Rosenbrock function

Figure 6: Comparison between the ASTRO-DF with interpolation and regression suggests that more care may be needed in utilizing regression for a derivative-free stochastic optimization routine such as ASTRO-DF. Although regression retains more information, it does not necessarily lead to better performance.

6 CONCLUSION

It is proven that ASTRO-DF achieves global convergence to a first-order critical point with probability one. The term "global convergence" indicates that the algorithm is guaranteed to converge to a critical point of the objective function regardless of the initial solution, thus providing a robust and reliable optimization technique. We suggest the enhanced version of ASTRO-DF with a method to reuse previous information aptly, named the history-informed ASTRO-DF. Since ASTRO-DF boasts several features, including a local model with diagonal Hessian and direct search method, which the reuse strategy may negatively impact, we have opted to utilize a design set with a rotated coordinate basis for retaining the benefits of the original ASTRO-DF. Implementing the reuse strategy offers an advantage in terms of computational budget. Our empirical results show that the rotated coordinate basis ensures a superior next candidate without compromising the algorithm's consistency or complexity. As for the complexity analysis of the history-informed ASTRO-DF, we leave it to future research to explore further.

The suggested reuse strategy can be applied to other SDFO algorithms. For example, in SGFM (Lin et al. 2022), the gradient is approximated using a central finite difference with a perturbation size δ and a randomly sampled direction $\boldsymbol{W}$. By employing the reuse strategy, $\boldsymbol{X}_k + \delta \boldsymbol{W}_k$ can be replaced with a design point from previous iterations, making δ a random sequence $\{\Delta_k\}$. However, the convergence and complexity analysis of this modified algorithm remain unknown, suggesting a potential future research direction.

ACKNOWLEDGMENTS

We thank Jeff Larson and Matt Menickelly for discussions that the ideas in this paper embarked on.

REFERENCES

Cao, L., A. S. Berahas, and K. Scheinberg. 2022. "First- and Second-Order High Probability Complexity Bounds for Trust-Region Methods with Noisy Oracles". *arXiv:2205.03667*.

Chang, K.-H., L. J. 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.

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. R., N. I. Gould, and P. L. Toint. 2000. *Trust Region Methods*. SIAM.

Conn, A. R., K. Scheinberg, and L. N. Vicente. 2009. *Introduction to derivative-free optimization*. 1st ed. Philadelphia: Society for Industrial and Applied Mathematics.

Eckman, D. J., S. G. Henderson, and S. Shashaani. 2023. "SimOpt: A Testbed for Simulation-Optimization Experiments". *INFORMS Journal on Computing* 35(2):495–508.

Ha, Y., and S. Shashaani. 2023. "Iteration Complexity and Finite-Time Efficiency of Adaptive Sampling Trust-Region Methods for Stochastic Derivative-Free Optimization". *arXiv:2305.10650*.

Ha, Y., S. Shashaani, and R. Pasupathy. 2023. "On Common-Random-Numbers and The Complexity of Adaptive Sampling Trust-Region Methods". *https://optimization-online.org/wp-content/uploads/2023/08/astrodf-complexity-online-version.pdf*.

Jin, B., K. Scheinberg, and M. Xie. 2021. "High Probability Complexity Bounds for Line Search Based on Stochastic Oracles". In *Advances in Neural Information Processing Systems*, Volume 34, 9193–9203: Curran Associates, Inc.

Kim, S., and D. Zhang. 2010. "Convergence Properties of Direct Search Methods for Stochastic Optimization". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 1003–1011. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kingma, D. P., and J. Ba. 2017. "Adam: A Method for Stochastic Optimization". *arXiv:1412.6980*.

Lin, T., Z. Zheng, and M. Jordan. 2022. "Gradient-Free Methods for Deterministic and Stochastic Nonsmooth Nonconvex Optimization". In *Advances in Neural Information Processing Systems*, Volume 35, 26160–26175. New York: Curran Associates, Inc.

Ragonneau, T. M., and Z. Zhang. 2023. "An Optimal Interpolation Set for Model-Based Derivative-Free Optimization Methods". *arXiv:2302.09992*.

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

Sun, S., and J. Nocedal. 2023. "A Trust Region Method for Noisy Unconstrained Optimization". *arXiv:2201.00973*.

Wild, S. M., R. G. Regis, and C. A. Shoemaker. 2008. "ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions". *SIAM Journal on Scientific Computing* 30(6):3197–3219.

AUTHOR BIOGRAPHIES

YUNSOO HA is a Ph.D. student in the Edward P. Fitts Department of Industrial and Systems Engineering at North Carolina State University. His research interests include simulation optimization, derivative-free optimization, and quantum computing. His e-mail address is yha3@ncsu.edu.

SARA SHASHAANI is an Assistant Professor in the Edward P. Fitts Department of Industrial and System Engineering at North Carolina State University. Her research interests are probabilistic data-driven models and simulation optimization. She is a co-creator of SimOpt. Her email address is sshasha2@ncsu.edu and her homepage is https://shashaani.wordpress.ncsu.edu/.