Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

ASTRO-DF: ADAPTIVE SAMPLING TRUST-REGION OPTIMIZATION ALGORITHMS, HEURISTICS, AND NUMERICAL EXPERIENCE

Sara Shashaani
Susan R. Hunter

Department of Industrial Engineering
Purdue University
West Lafayette, IN 47907, USA

Raghu Pasupathy

Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

ABSTRACT

ASTRO-DF is a class of adaptive sampling algorithms for solving simulation optimization problems in which only estimates of the objective function are available by executing a Monte Carlo simulation. ASTRO-DF algorithms are iterative trust-region algorithms, where a local model is repeatedly constructed and optimized as iterates evolve through the search space. The ASTRO-DF class of algorithms is derivative-free in the sense that it does not rely on direct observations of the function derivatives. A salient feature of ASTRO-DF is the incorporation of adaptive sampling and replication to keep the model error and the trust-region radius in lock-step, to ensure efficiency. ASTRO-DF has been demonstrated to generate iterates that globally converge to a first-order critical point with probability one. In this paper, we describe and list ASTRO-DF, and discuss key heuristics that ensure good finite-time performance. We report our numerical experience with ASTRO-DF on test problems in low to moderate dimensions.

1 INTRODUCTION

We consider simulation optimization (SO) problems, that is, optimization problems where the objective function can only be expressed implicitly via a Monte Carlo simulation oracle. In the recent years, SO has been adopted in a wide range of problem contexts including quality control (Rani and Moreira 2010), telecommunication networks (Hou et al. 2014), traffic control (Osorio and Bierlaire 2009), epidemic forecasting (Nsoesie et al. 2013), and health care (Alagoz et al. 2009). For a collection of SO problems, see (Pasupathy and Henderson 2006) and the testbed available at `www.simopt.org`. The SO problem is formally stated as

$$\text{Problem } P: \quad \text{minimize} \quad f(\boldsymbol{x}) := \mathbb{E}[F(\boldsymbol{x})] \quad \text{s.t.} \quad \boldsymbol{x} \in \mathbb{R}^d,$$

where $f(\cdot)$ is known only through a Monte Carlo simulation capable of generating copies of the random variable $F(\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathbb{R}^d$. The estimator of $f(\boldsymbol{x})$, denoted $\bar{F}(\boldsymbol{x},n)$, is constructed as the sample mean of n i.i.d copies of $F(\boldsymbol{x})$.

Monte Carlo oracle calls are often much more expensive than routine numerical operations, and hence explicit derivative estimation is computationally expensive. Moreover, choosing the step-size for finite-differencing in the presence of stochastic error can be a delicate operation, resulting in estimators having poor quality (Asmussen and Glynn 2007). As a result, in such contexts, fast (asymptotic) convergence rates to a critical point are arguably less relevant than stability and the need for "good" solutions. This motivates our interest in developing derivative-free algorithms for the SO context.

While trust-region optimization (TRO) algorithms have existed for several decades and represent a mature class of algorithms, deterministic model-based derivative-free trust-region (TRO-DF) algorithms (Conn et al.

2009b) have become popular only over the last decade. The family of algorithms we propose, called ASTRO-DF, follows a logic that is analogous to TRO-DF. ASTRO-DF is an iterative algorithm in which during each iteration k a local (and analytically convenient) model of the objective function is constructed by strategically placing design points within a trust-region around the current iterate $\boldsymbol{X}_k$, and then obtaining Monte Carlo estimates of the objective function at each design point. As we shall see, the number of Monte Carlo calls at each design point is *adaptive* — just enough to ensure that the sampling variability of function observations at the point is commensurate with the estimated model error. After such model construction and (stochastic) certification, the constructed model is used within an optimization step to identify the next candidate solution $\tilde{\boldsymbol{X}}_{k+1}$. The candidate solution $\tilde{\boldsymbol{X}}_{k+1}$ is not immediately accepted. Instead, if the model-predicted and Monte Carlo-estimated function decrease values from the current iterate $\boldsymbol{X}_k$ to the candidate point $\tilde{\boldsymbol{X}}_{k+1}$ are comparable in a certain sense, then $\tilde{\boldsymbol{X}}_{k+1}$ is accepted as the next iterate $\boldsymbol{X}_{k+1}$ and the trust-region expanded; otherwise, $\tilde{\boldsymbol{X}}_{k+1}$ is rejected and the trust-region is shrunk in an attempt to improve the quality of the local model around $\boldsymbol{X}_k$. ASTRO-DF has been shown to globally converge to a first-order critical point almost surely. For the details on theoretical results see the recent paper on ASTRO-DF (Shashaani et al. 2015).

A number of features within ASTRO-DF are noteworthy. First, we introduce certificates of stochastic model sufficiency for use in ASTRO-DF's model construction step. Such certificates of model sufficiency correspond to notions of fully linear and fully quadratic models (Conn et al. 2009b) in the deterministic context, and are important to guarantee ASTRO-DF's convergence. Second, sampling within ASTRO-DF, most of which happens within the model construction step, is adaptive. Such adaptivity complicates ASTRO-DF's analysis but ensures efficiency through explicit dependence on algorithmic trajectory. Third, three different sources of error are present within ASTRO-DF: (i) *stochastic sampling error* arising due to the fact that function evaluations are through Monte Carlo simulation; (ii) *model bias* arising due to the choice of the local model; and (iii) *stochastic interpolation error* arising due to the fact that model prediction at unobserved points is a combination of the model bias and the error in (i). (Of these, only (ii) is present in the deterministic context.) As we shall see, the sampling and searching logic within ASTRO-DF, towards fostering efficiency, keeps the three sources of error (i), (ii), and (iii) in lock-step.

2 ASTRO-DF ALGORITHM

To facilitate our discussion of ASTRO-DF, we first provide the listing for a simplified deterministic trust-region optimization (DTRO) algorithm in Algorithm 1. While this algorithm does employ gradients, we use it to build basic intuition for trust-region algorithms and for the ASTRO-DF algorithm that will follow. Our presentation and notation closely follow that of Nocedal and Wright (2006).

On the kth iteration of DTRO, we first construct a quadratic model at the current point $\boldsymbol{x}_k$ (Step 2). Next, we optimize (Step 3) the model in a *trust-region*, yielding a candidate point $\tilde{\boldsymbol{x}}_{k+1}$. This is followed by the calculation of the success ratio in Step 4 in preparing to evaluate the quality of the candidate point $\tilde{\boldsymbol{x}}_{k+1}$. In Steps 5–9, the candidate point $\tilde{\boldsymbol{x}}_{k+1}$ is accepted and the trust-region radius expanded if the success ratio exceeds a certain threshold; otherwise, the incumbent point $\boldsymbol{x}_k$ remains unchanged and the trust-region radius is shrunk in an attempt to improve the quality of the local quadratic model.

Given the above basic framework for a trust-region algorithm, we now describe ASTRO-DF in greater detail. A detailed algorithm listing is provided in Algorithm 2, where the numbering of all steps corresponds to the analogous steps in DTRO. Since we assume that gradient estimates are not available, model construction is a key step in ASTRO-DF. Detailed steps for model construction are listed in Algorithm 3.

The operations during each iteration of ASTRO-DF are encapsulated within four repeating stages that are modified versions of their DTRO counterparts: (i) local (stochastic) model construction and certification through adaptive sampling; (ii) constrained optimization of the constructed model (within the trust-region) for identifying the next candidate solution; (iii) re-estimation of the objective function at the candidate solution through adaptive sampling and evaluation of the candidate solution; and (iv) (stochastic) sufficient

Algorithm 1 Deterministic Trust-Region Optimization (DTRO) Algorithm

Require: Initial guess $\boldsymbol{x}_0 \in \mathbf{R}^d$, initial trust-region radius $\Delta_0 > 0$ and maximum radius $\Delta_{\max} > 0$, model "fitness" threshold $\eta_1 > 0$, trust-region expansion constant $\gamma_1 > 1$ and contraction constant $\gamma_2 \in (0,1)$.

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

Model Construction: Construct the quadratic model $m_k(\boldsymbol{x}_k + \boldsymbol{p}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T B_k \boldsymbol{p}$ at the current point $\boldsymbol{x}_k$.

TR Subproblem: Obtain the kth step by minimizing the model in trust-region, $\boldsymbol{p}_k = \operatorname{argmin}_{\|\boldsymbol{p}\| \le \Delta_k} m_k(\boldsymbol{x}_k + \boldsymbol{p})$, and set the new candidate point $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_k + \boldsymbol{p}_k$.

Evaluate: Evaluate success ratio

$$\rho_k = \frac{f(\boldsymbol{x}_k) - f(\tilde{\boldsymbol{x}}_{k+1})}{m_k(\boldsymbol{x}_k) - m_k(\tilde{\boldsymbol{x}}_{k+1})}. \quad \begin{matrix}\{\text{numerator: actual objective value "reduction;" could be positive or negative}\} \\ \{\text{denominator: objective value reduction predicted by model; always positive}\}\end{matrix}$$

Update:

if $\rho_k > \eta_1$ **then** {If the actual objective value reduces "enough" relative to the model,}

$\boldsymbol{x}_{k+1} = \tilde{\boldsymbol{x}}_{k+1}$, $\Delta_{k+1} = \min\{\gamma_1 \Delta_k, \Delta_{\max}\}$. {accept the candidate point and expand the trust-region radius.}

else

$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$, $\Delta_{k+1} = \gamma_2 \Delta_k$. {Else, remain at $\boldsymbol{x}_k$ and reduce the trust-region radius.}

end if

end for

Algorithm 2 ASTRO-DF Main Algorithm

Require: *Parameters from DTRO:* Initial guess $\boldsymbol{x}_0 \in \mathbf{R}^d$, initial trust-region radius $\tilde{\Delta}_0 > 0$ and maximum radius $\Delta_{\max} > 0$, model "fitness" threshold $\eta_1 > 0$, trust-region expansion constant $\gamma_1 > 1$ and contraction constant $\gamma_2 \in (0,1)$.
Parameters specific to ASTRO-DF: initial sample size n_0, sample size lower bound sequence $\{\lambda_k\}$ such that $k^{(1+\varepsilon)} = O(\lambda_k)$, initial sample set $\tilde{\mathcal{Y}}_0 = \{\boldsymbol{x}_0\}$, and outer adaptive sampling constant κ_{oas}.

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

Model Construction: Construct the model at $\boldsymbol{X}_k$ by calling Algorithm 3 with the candidate trust-region radius $\tilde{\Delta}_k$ and candidate set of sample points $\tilde{\mathcal{Y}}_k$, $[M_k(\boldsymbol{X}_k + \boldsymbol{s}), \Delta_k, \mathcal{Y}_k]$=`AdaptiveModelConstruction`$(\tilde{\Delta}_k, \tilde{\mathcal{Y}}_k)$.

TR Subproblem: Approximate the kth step by minimizing the model in the trust-region, $\boldsymbol{S}_k = \operatorname{argmin}_{\|\boldsymbol{s}\| \le \Delta_k} M_k(\boldsymbol{X}_k + \boldsymbol{s})$, and set the new candidate point $\tilde{\boldsymbol{X}}_{k+1} = \boldsymbol{X}_k + \boldsymbol{S}_k$.

Evaluate: Estimate the function at the candidate point using adaptive sampling to obtain $\bar{F}(\tilde{\boldsymbol{X}}_{k+1}, \tilde{N}_{k+1})$, where

$$\tilde{N}_{k+1} = \max\left\{\lambda_k, \min\left\{n : \frac{\hat{\sigma}\left(\tilde{\boldsymbol{X}}_{k+1}, n\right)}{\sqrt{n}} \le \frac{\kappa_{oas}\Delta_k^2}{\sqrt{\lambda_k}}\right\}\right\}. \tag{1}$$

Let $\tilde{N}_k = N(\boldsymbol{X}_k)$ obtained from Step 2 and evaluate the estimated success ratio

$$\hat{\rho}_k = \frac{\bar{F}\left(\boldsymbol{X}_k, \tilde{N}_k\right) - \bar{F}\left(\tilde{\boldsymbol{X}}_{k+1}, \tilde{N}_{k+1}\right)}{M_k(\boldsymbol{X}_k) - M_k(\tilde{\boldsymbol{X}}_{k+1})}.$$

Update:

if $\hat{\rho}_k > \eta_1$ **then**

$\boldsymbol{X}_{k+1} = \tilde{\boldsymbol{X}}_{k+1}$, $\tilde{\Delta}_{k+1} = \min\{\gamma_1 \Delta_k, \Delta_{\max}\}$, $N_{k+1} = \tilde{N}_{k+1}$.
Set $\tilde{\boldsymbol{Y}}_{\max} := \arg\max_{\boldsymbol{Y}_i \in \mathcal{Y}_k} \left\{\left\|\tilde{\boldsymbol{X}}_{k+1} - \boldsymbol{Y}_i\right\|\right\}$. Update the sample set $\tilde{\mathcal{Y}}_{k+1} = \mathcal{Y}_k \setminus \left\{\tilde{\boldsymbol{Y}}_{\max}\right\} \cup \{\boldsymbol{X}_{k+1}\}$.

else

$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k$, $\tilde{\Delta}_{k+1} = \gamma_2 \Delta_k$, $N_{k+1} = \tilde{N}_k$.
Set $\boldsymbol{Y}_{\max} := \arg\max_{\boldsymbol{Y}_i \in \mathcal{Y}_k} \{\|\boldsymbol{X}_k - \boldsymbol{Y}_i\|\}$. If $\tilde{\boldsymbol{X}}_{k+1} \neq \boldsymbol{Y}_{\max}$, then update $\tilde{\mathcal{Y}}_{k+1} = \mathcal{Y}_k \setminus \{\boldsymbol{Y}_{\max}\} \cup \left\{\tilde{\boldsymbol{X}}_{k+1}\right\}$.

end if

end for

decrease check by comparing predicted and estimated function decrease, and iterate and trust-region update. These stages are labeled in italics in Algorithm 2.

We now describe each step of Algorithm 2. In Step 2, a stochastic model of the function $f(\cdot)$, constrained to the trust-region $\mathcal{B}(\boldsymbol{X}_k; \Delta_k)$, is constructed using Algorithm 3. Equipped with a certified model $M_k(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{B}(\boldsymbol{X}_k, \Delta_k)$, Step 3 in Algorithm 2 involves approximately solving the constrained optimization problem $\boldsymbol{S}_k = \arg\min_{\|\boldsymbol{s}\| \le \Delta_k} M_k(\boldsymbol{X}_k + \boldsymbol{s})$ to obtain a candidate solution $\tilde{\boldsymbol{X}}_{k+1} = \boldsymbol{X}_k + \boldsymbol{S}_k$ that attains a fraction

Algorithm 3 $[M_k(\boldsymbol{X}_k+\boldsymbol{s}),\Delta_k,\mathcal{Y}_k]$=`AdaptiveModelConstruction`$(\tilde{\Delta}_k,\tilde{\mathcal{Y}}_k)$

Require: *Parameters from ASTRO-DF:* candidate trust-region radius $\tilde{\Delta}_k$ and candidate sample set $\tilde{\mathcal{Y}}_k$.
Parameters specific to `AdaptiveModelConstruction`: trust-region contraction factor $w\in(0,1)$, trust-region and gradient balance constant μ, gradient inflation constant β with $0<\beta<\mu$, and inner adaptive sampling constant κ_{ias}.

1: Initialize $j_k=1$, and set $\mathcal{Y}_k=\tilde{\mathcal{Y}}_k$, and set $\boldsymbol{Y}_1^{(j_k)}=\boldsymbol{X}_k$ where $\boldsymbol{X}_k$ is the first element of $\tilde{\mathcal{Y}}_k$.
2: **repeat**
3: Improve $\mathcal{Y}_k^{(j_k)}=\left\{\boldsymbol{Y}_1^{(j_k)},\boldsymbol{Y}_2^{(j_k)},\ldots,\boldsymbol{Y}_p^{(j_k)}\right\}$ by choosing $\boldsymbol{Y}_i^{(j_k)}$, $i=2,3,\cdots,p$ to make a poised set in $\mathcal{B}(\boldsymbol{X}_k;\tilde{\Delta}_k w^{j_k-1})$.
4: **for** $i=1$ to p **do**
5: Estimate $\bar{F}\left(\boldsymbol{Y}_i^{(j_k)},N\left(\boldsymbol{Y}_i^{(j_k)}\right)\right)$, where

$$N(\boldsymbol{Y}_i^{(j_k)})=\max\left\{\lambda_k,\min\left\{n:\frac{\hat{\sigma}\left(\boldsymbol{Y}_i^{(j_k)},n\right)}{\sqrt{n}}\le\frac{\kappa_{ias}(\tilde{\Delta}_k w^{j_k-1})^2}{\sqrt{\lambda_k}}\right\}\right\}. \tag{2}$$

6: **end for**
7: Construct a quadratic model $M_k^{(j_k)}(\boldsymbol{X}_k+\boldsymbol{s})$ via interpolation.
8: Set $j_k=j_k+1$.
9: **until** $\tilde{\Delta}_k w^{j_k-1}\le\mu\|\nabla M_k^{(j_k)}(\boldsymbol{X}_k)\|$.
10: Set $\nabla M_k(\boldsymbol{X}_k)=\nabla M_k^{(j_k)}(\boldsymbol{X}_k)$, $\nabla^2 M_k(\boldsymbol{X}_k)=\nabla^2 M_k^{(j_k)}(\boldsymbol{X}_k)$.
11: **return** $M_k(\boldsymbol{X}_k+\boldsymbol{s})=M_k^{(j_k)}(\boldsymbol{X}_k+\boldsymbol{s})$, $\Delta_k=\min\{\tilde{\Delta}_k,\max\{\beta\|\nabla M_k(\boldsymbol{X}_k)\|,\tilde{\Delta}_k w^{j_k-1}\}\}$, and $\mathcal{Y}_k=\mathcal{Y}_k^{(j_k)}$.

of the Cauchy decrease, that is, identify $\boldsymbol{S}_k$ so that for a positive constant κ_{fcd}

$$M_k(\boldsymbol{X}_k)-M_k(\boldsymbol{X}_k+\boldsymbol{S}_k)\ge\frac{\kappa_{fcd}}{2}\|\nabla M_k(\boldsymbol{X}_k)\|\min\left\{\frac{\|\nabla M_k(\boldsymbol{X}_k)\|}{\|\nabla^2 M_k(\boldsymbol{X}_k)\|},\Delta_k\right\}. \tag{3}$$

In preparation for checking if the candidate solution $\tilde{\boldsymbol{X}}_{k+1}$ provides sufficient decrease, Step 4 of Algorithm 2 obtains a Monte Carlo estimate of the function value at $\tilde{\boldsymbol{X}}_{k+1}$; enough sampling is performed so that the estimated standard error of the function estimate is smaller than a deflated multiple of square of the trust-region radius ($\Delta_k^2/\sqrt{\lambda_k}$), subject to the sample size being at least λ_k. The obtained function estimate is then used to check if the ratio $\hat{\rho}_k$ of the observed function decrease to the predicted function decrease at the point $\tilde{\boldsymbol{X}}_k$ exceeds a fixed threshold. If $\hat{\rho}_k$ exceeds the specified threshold, the point $\tilde{\boldsymbol{X}}_{k+1}$ is accepted, the iteration is deemed successful, and the trust-region expands; if $\hat{\rho}_k$ falls below the specified threshold, the point $\tilde{\boldsymbol{X}}_{k+1}$ is rejected, the iteration is deemed unsuccessful, and the trust-region shrinks. In either case, the algorithm proceeds to the next iteration.

Recall that Step 2 is a model construction step where a stochastic model of the function $f(\cdot)$ is constructed using a loop inside Algorithm 3. Specifically, Algorithm 3 obtains Monte Carlo function estimates at each point of a poised interpolation set $\mathcal{Y}_k^{(j_k)}=\left\{\boldsymbol{Y}_1^{(j_k)},\boldsymbol{Y}_2^{(j_k)},\ldots,\boldsymbol{Y}_p^{(j_k)}\right\}$. In Step 5 of Algorithm 3, note that Monte Carlo sampling at each point $\boldsymbol{Y}_i^{(j_k)}\in\mathcal{Y}_k^{(j_k)}$ continues until the estimated standard error $\hat{\sigma}\left(\boldsymbol{Y}_i^{(j_k)},N\left(\boldsymbol{Y}_i^{(j_k)}\right)\right)/\sqrt{N\left(\boldsymbol{Y}_i^{(j_k)}\right)}$ of the function estimate $\bar{F}\left(\boldsymbol{Y}_i^{(j_k)},N\left(\boldsymbol{Y}_i^{(j_k)}\right)\right)$ drops below a deflated multiple of square of the trust-region radius, subject to the sample size being at least λ_k. Then an interpolation model is fit to the observed function values in the design set. If the gradient of the constructed model is not large enough relative to the trust-region radius, the trust-region is contracted and the whole process is repeated.

Algorithm 3 performs the two crucial roles of model construction and certification. That Algorithm 3 terminates almost surely, is a nontrivial fact that is proved rigorously in Shashaani et al. (2015). The increased sampling resulting from the use of the inflation sequence λ_k ensures that the spurious effects of stochastic sampling decay at a fast enough rate to secure the almost sure convergence of the iterates $\{\boldsymbol{X}_k\}$.

3 KEY IMPLEMENTATION HEURISTICS

Notwithstanding the global convergence proofs, certain implementation heuristics appear to be important to ensure ASTRO-DF's good finite-time performance. For example, the choice of interpolation points in the model construction step, trust-region management details, the manner in which historical iterates are re-used in the model construction step, and the specific methods used for updating iterates, all affect ASTRO-DF's functioning. In this section we detail five such aspects listed here in order of importance.

1. Choosing the set of design points $\mathcal{Y}_k^{(j_k)}$ for model construction in Algorithm 3 (Section 3.1).
2. Choosing the algorithm parameters to enhance practical efficiency (Section 3.2).
3. Pre-processing to identify the initial point $\boldsymbol{X}_0$ and the initial trust-region radius Δ_0 (Section 3.3).
4. Solving the trust-region sub-problem (Section 3.4).
5. Choosing an iterate subsequent to a successful iteration (Section 3.5).

The almost sure convergence results of ASTRO-DF are not affected by the choices suggested in subsections 3.2, 3.3, and 3.4; they are, however, affected by our proposals in subsections 3.1 and 3.5. Specifically, the convergence proofs for ASTRO-DF in (Shashaani et al. 2015) require the interpolation set $\mathcal{Y}_k$ to remain certifiably fully poised in every iteration. The implementation of ASTRO-DF that we discuss here relaxes this stipulation, thereby threatening convergence. We speculate that the convergence proofs in (Shashaani et al. 2015) could be generalized to subsume the relaxation we propose, by stipulating full linearity only on a subsequence of iterations.

3.1 Choosing Design Points for the Model Construction Step

The quality of models constructed within ASTRO-DF crucially affects ASTRO-DF's performance. There is, however, a natural tension between constructing accurate models and the fast convergence of ASTRO-DF. Constructing accurate models entails identifying a "well dispersed" set of design points and then sampling adequately at each of these identified points. And, the need to identify a well-dispersed set of points means that past iterates, which are usually highly correlated, can only be used sparingly, if at all. In what follows, we detail a proposal that balances the competing need for well dispersed points and the inclusion of past algorithm iterates into the design set. (What we detail here applies toward executing Step 3 in Algorithm 3.)

Our proposal to identify the $p = (d+1)(d+2)/2$ design points needed to construct a full set $\mathcal{Y}_k$ involves the following two steps.

(i) Identify a well dispersed subset, defined in a certain rigorous sense, from amongst the already observed points for inclusion into $\mathcal{Y}_k^{(j_k)}$; and
(ii) if the cardinality of the set identified in (i) is less than p, identify additional well dispersed points to complete the full set $\mathcal{Y}_k^{(j_k)}$.

The steps for choosing design points towards constructing a model are listed in Algorithm 4, requiring the TR radius and model gradient norm in the latest iteration of the model construction loop of Algorithm 3, as well as the history of all visited points.

For (i) (Steps 1–11), a convenient method for the identification of "poised" points, denoted as $\mathcal{Y}_{init}$ is through the maximization of Lagrange functions, as detailed in Algorithm 6.2 in (Conn et al. 2009b, p. 95) , where the Lagrange functions are first reset to the normal basis of a quadratic interpolation model, that is $\Phi(\mathbf{z}) := (\phi_1, \phi_2, \ldots, \phi_p) = \left(1, z_1, z_2, \ldots, z_d, \frac{1}{2}z_1^2, z_1z_2, \ldots, \frac{1}{2}z_2^2, \ldots, \frac{1}{2}z_d^2\right)$, and then updated according to the new design points added to the set. We use the COBYLA (Constrained Optimization BY Linear Approximation) procedure (Powell 1994) for this purpose. Moreover we identify, from amongst all points visited by ASTRO-DF and lying within the current trust-region, a subset of points such that the distance between any two points included within the subset is at least $\theta\tilde{\Delta}_k w^{j_k-1}$, $\theta \in (0,1)$. We call this subset $\mathcal{Y}_{pool}$.

Then, for re-using purposes, the *equivalent* points in $\mathcal{Y}_{pool}$ to those in $\mathcal{Y}_{init}$ are considered for inclusion in the sample set. An *equivalent* of a point is defined as the closest one of $\mathcal{Y}_{pool}$ with the distance of at most $\theta'\tilde{\Delta}_k w^{j_k-1}$, $\theta' < \theta$, to the point.

The current iterate (and centre of the trust-region) is always included within the sample set. In the unlikely event that the cardinality of the subset identified in (i) is equal to p, we have successfully identified the complete sample set $\mathcal{Y}_k^{(j_k)}$. Otherwise, as part of (ii) (Steps 12–22), we search for additional points that would complete the set $\mathcal{Y}_k^{(j_k)}$ while satisfying the minimum separation $\theta\tilde{\Delta}_k w^{j_k-1}$ between all pairs of points. The criticality alert, triggered if the most recent model gradient norm is small, enforces high quality models by choosing the remainder of the sample set from new points in $\mathcal{Y}_{init}$. Albeit with no evidence of criticality, the additional points are selected from old points in $\mathcal{Y}_{pool}$ in the order of most distant from all the current points in the $\mathcal{Y}_k^{(j_k)}$ to keep the set well-spread.

Algorithm 4 $[\mathcal{Y}_k^{(j_k)}]$=`SampleSelection`$(\tilde{\Delta}_k w^{j_k-1}, \boldsymbol{X}_k, \left\|\nabla M_k^{(j_k)}(\boldsymbol{X}_k)\right\|, \bigcup_{\ell=0}^{k-1}\bigcup_{t=1}^{j_\ell}\mathcal{Y}_\ell^{(t)})$

Require: *Parameters from* `AdaptiveModelConstruction`: TR radius $\tilde{\Delta}_k w^{j_k-1}$, current iterate $\boldsymbol{X}_k$, current model gradient norm $\left\|\nabla M_k^{(j_k)}(\boldsymbol{X}_k)\right\|$, and previous sample sets $\bigcup_{\ell=0}^{k-1}\bigcup_{t=1}^{j_\ell}\mathcal{Y}_\ell^{(t)}$.
Parameters specific to `SampleSelection`: minimum separation constant $0<\theta<1$, equivalence constant $0<\theta'<\theta$ and criticality constant ε_g.

1: Find a new poised set $\mathcal{Y}_{init} = \{\boldsymbol{X}_k, \boldsymbol{Y}_2, \boldsymbol{Y}_3, \cdots, \boldsymbol{Y}_p\}$ using Lagrange polynomials. Let $\mathcal{Y}_k^{(j_k)} = \{\boldsymbol{X}_k\}$, $\mathcal{J} = \varnothing$ and $\mathcal{Y}_{pool} = \varnothing$.
2: **for all** $\boldsymbol{y} \in \bigcup_{t=1}^{j_{k-i}} \mathcal{Y}_{k-i}^{(t)} \cap \mathcal{B}\left(\boldsymbol{X}_k; \tilde{\Delta}_k w^{j_k-1}\right)$, $i = 1, 2, \cdots, k$ **do** {Check the visited points, starting from the most recent.}
3: **if** $\boldsymbol{y} \notin \bigcup_{\mathbf{z}\in\mathcal{Y}_{pool}} \mathcal{B}\left(\mathbf{z}; \theta\tilde{\Delta}_k w^{j_k-1}\right)$ **then** {If not within minimum separation with other points, add to the pool.}
4: Set $\mathcal{Y}_{pool} = \mathcal{Y}_{pool} \cup \{\boldsymbol{y}\}$.
5: **end if**
6: **end for**
7: **for all** $\boldsymbol{Y}_i \in \mathcal{Y}_{init}$, $i = 2, 3, \cdots, p$ **do**
8: **if** $\boldsymbol{Y}_i' := \underset{\mathbf{z}\in\mathcal{Y}_{pool}\cap\mathcal{B}\left(\boldsymbol{Y}_i;\theta'\tilde{\Delta}_k w^{j_k-1}\right)}{\arg\min} \|\mathbf{z}-\boldsymbol{Y}_i\|_2$ exists, **then** {Select those points of $\mathcal{Y}_{init}$ that have equivalents in $\mathcal{Y}_{pool}$.}
9: Set $\mathcal{Y}_k^{(j_k)} = \mathcal{Y}_k^{(j_k)} \cup \{\boldsymbol{Y}_i'\}$ and $\mathcal{J} = \mathcal{J} \cup \{i\}$. {Place their closest equivalent in $\mathcal{Y}_k^{(j_k)}$.}
10: **end if**
11: **end for**
12: **if** $\left|\mathcal{Y}_k^{(j_k)}\right| < p$, **then** {If the sample set does not have p points in it choose the rest based on criticality.}
13: **if** $\left\|\nabla M_k^{(j_k)}(\boldsymbol{X}_k)\right\| < \varepsilon_g$, **then** {Alert if the current TR is in critical region, implying poised-ness must be maintained.}
14: **for all** $i = 2$ to p and $i \notin \mathcal{J}$ **do** {Choose the remainder of the points from the new points in the poised set $\mathcal{Y}_{init}$.}
15: $\mathcal{Y}_k^{(j_k)} = \mathcal{Y}_k^{(j_k)} \cap \{\boldsymbol{Y}_i\}$.
16: **end for**
17: **else**
18: **while** $\left|\mathcal{Y}_k^{(j_k)}\right| \neq p$ **do** {Choose the remainder of the points from the points in $\mathcal{Y}_{pool}$.}
19: Set $\boldsymbol{Y}_{best} := \underset{\mathbf{z}\in\mathcal{Y}_{pool}}{\arg\max} \sum_{\boldsymbol{y}\in\mathcal{Y}_k^{(j_k)}} \|\mathbf{z}-\boldsymbol{y}\|_2$. {Choose the point with largest cumulative distance to all members of $\mathcal{Y}_k^{(j_k)}$.}
Set $\mathcal{Y}_{pool} = \mathcal{Y}_{pool} \setminus \{\boldsymbol{Y}_{best}\}$ and $\mathcal{Y}_k^{(j_k)} = \mathcal{Y}_k^{(j_k)} \cup \{\boldsymbol{Y}_{best}\}$
20: **end while**
21: **end if**
22: **end if**

3.2 Choosing Algorithm Parameters

The parameters in ASTRO-DF fall into two categories: general trust-region parameters and adaptive sampling parameters. We now discuss the choice and effect of these parameters in broad terms. It must be understood that, just as in much of algorithm design, there is a certain subjectivity in the choice of algorithm parameters. Convergence theory frequently leaves open a wide range of possibilities for algorithm

parameter choice, which must then be narrowed through empirical experience. In accordance with the philosophy that a well designed algorithm implementation should not expect a user to choose algorithm parameters, we suggest default values for all parameters we discuss here. All results reported in the section on numerical results were obtained using default parameter settings.

3.2.1 General Trust-Region Parameters

The general parameters in the trust-region framework include $\eta_1, \gamma_1, \gamma_2, \beta, \mu$ and w. For all experiments that we report in section 4 we have used the following default parameter settings: $\gamma_1 = 1.2, \gamma_2 = 0.9, \beta = 0.5, \mu = 2.0$, and $w = 0.9$. In what follows, we provide some intuition on each of these parameters.

The parameter η_1 is a threshold for sufficient reduction in the function estimated value when moving from the current iterate $\boldsymbol{X}_k$ to the candidate solution $\tilde{\boldsymbol{X}}_{k+1}$. Large values of η_1 make the sufficient reduction condition more stringent, stipulating higher model accuracies; small values of η_1 make the sufficient reduction condition more lax, allowing for explorative moves. It is worth noting that the ASTRO-DF algorithm as listed in this paper includes only a sufficient decrease condition. By contrast, the deterministic TRO-DF algorithm proposed by Conn et al. (2009a) includes an additional constant η_0 that is meant to allow a *simple decrease* condition in addition to the *sufficient decrease* condition.

ASTRO-DF accepts the candidate point as the next iterate when the reduction predicted by the model exceeds the estimated reduction by a factor η_1; such acceptance then amounts to a tacit acknowledgement that the newly constructed model can perhaps adequately represent the objective function in a region with a radius that is larger than the incumbent trust-region radius. The parameter γ_1 controls the extent of such increase in the trust-region radius post candidate acceptance. Conversely, when a candidate point is not accepted due to the predicted decrease being too small a fraction of the estimated decrease, ASTRO-DF reposes less faith in the model, leading to contraction of the trust-region radius. The extent to which such reduction happens is controlled by the parameter γ_2. The other contraction factor is w in the inner loop of Algorithm 3. Small values for both of these contraction factors can result in changes in the model as a result of changes in the sample set, and a corresponding faster consumption of the simulation budget.

The parameter β, along with the parameter μ, enforces the model gradient to be in lock-step with the trust-region radius. Algorithm 3 continues to be executed until a model of specified quality is constructed in a trust-region whose radius does not exceed the product of μ and the model gradient. A large value of μ thus allows for greater lenience, resulting in a poorer model. On the other hand, the parameter β is used to prevent the trust-region radius resulting from the execution of Algorithm 3 from becoming too small. Towards satisfying the stipulated lock-step, Algorithm 3 repeatedly shrinks the trust-region radius using the constant factor w, thereby introducing the possibility of a final trust-region with a radius that is very small. The parameter β prevents this possibility. As an example, if the parameter μ is set equal to β, the size of the trust-region that exits Algorithm 3 is strictly in lock-step with the product of β and the model gradient norm.

Furthermore the default parameter settings in the sample selection heuristic in our experiments are $\theta = 0.2, \theta' = 0.05$, and $\varepsilon_g = 10$, chosen in an ad-hoc manner.

3.2.2 Adaptive Sampling Parameters

Whenever the objective function needs to be estimated at a specified design point, ASTRO-DF has to make a decision on how much sampling effort needs to be exerted for estimation. One of the salient features of ASTRO-DF is that decisions on the extent of sampling are, at least to a certain degree, adaptive. Specifically, the sampling rules in expressions (1) and (2) control ASTRO-DF's sampling rate with the two parameters κ_{oas} and κ_{ias} corresponding to the adaptive sampling constants for the outer-loop and the inner loops respectively. Small values of κ_{oas} and κ_{ias} make ASTRO-DF trajectories appear deterministic due to increased sampling leading to reduced sampling error. On the other hand, large values of κ_{oas} and

κ_{ias} imply less sampling and increased variability in sample paths. The parameters κ_{oas} and κ_{ias} far more affect the convergence rate of ASTRO-DF than whether or not ASTRO-DF converges.

The other important adaptive sampling parameter is the inflation factor λ_k. This parameter implicitly sets a lower bound for the sample size during each iteration. As specified in the inputs of Algorithm 2, the sequence $\{\lambda_k\}$ should satisfy $k^{(1+\varepsilon)} = O(\lambda_k)$, that is, λ_k is roughly of the same order as the iteration number. (we use $\varepsilon = 10E-4$). Our extensive numerical experience indicates that the lower bound sample size imposed through the sequence $\{\lambda_k\}$ is rarely binding, especially as ASTRO-DF's iterates approach a stationary point. This is consistent with what has been predicted by theory in other contexts.

In all experiments described in section 4 we impose a large number for the inner and outer loop sampling constants ($\kappa_{oas} = \kappa_{ias} = 10^3$) to enable more exploration throughout the search.

3.3 Pre-processing

Like any non-linear optimization algorithm, the choice of initial values, specifically, the initial guess $\boldsymbol{X}_0$ and the initial trust-region radius Δ_0, affect ASTRO-DF's performance. Accordingly, we have found it expedient to undertake a certain pre-processing step aimed at identifying good values for the initial guess x_0 and the starting trust-region radius Δ_0. With a fixed small budget we run ASTRO-DF with a vector of random initial points and a vector of random initial trust-region radii, giving each combination of the initial point and initial trust-region radius the same share of the pre-processing simulation budget. The best combination of the initial point and trust-region radius are then selected based on the resulting relative reduction in the model gradient norm.

3.4 Solving the TR Subproblem

The candidate point $\tilde{\boldsymbol{X}}_{k+1} = \boldsymbol{X}_k + \boldsymbol{S}_k$ that is the potential next incumbent solution in the search process comes from a constrained optimization problem in Step 3 of Algorithm 2. To find a good candidate solution $\boldsymbol{S}_k$, one can use the Cauchy step, which is the minimizer of the one-dimensional constrained optimization problem obtained by projecting the objective function along the negative gradient and constrained to the trust-region. The resulting step satisfies the Cauchy reduction in expression (3) that is required for the convergence of ASTRO-DF, with $\kappa_{fed} = 2$ for linear models and $\kappa_{fed} = 1$ for quadratic models. In such a case $\boldsymbol{S}_k$ is chosen as $\boldsymbol{S}_k = t_C \nabla M_k(\boldsymbol{X}_k)$, where

$$t_C = \underset{\alpha \in [0, \Delta_k]}{\arg\min}\, M_k(\boldsymbol{X}_k - \alpha \nabla M_k(\boldsymbol{X}_k)),$$

to satisfy a $\frac{1}{2}$-Cauchy decrease. (See Section 10.1 in Conn et al. (2009b) for additional details.)

Any routine to solve the TR subproblem that provides a candidate point with a higher reduction than that obtained through the Cauchy step is obviously preferred, although the resulting computational effort needs to be weighed against the reduction in objective function value. In the experiments reported in this paper we apply the constrained optimization method COBYLA (Powell 1994).

3.5 Updating the Next Iterate

Given that several design points (along with their function estimates) are observed during the model construction and the TR subproblem stages, an important question is which amongst these should be chosen as the subsequent iterate in the event that the the sufficient reduction step is satisfied leading to a successful iteration. An obvious choice is the candidate point $\tilde{\boldsymbol{X}}_{k+1}$ in Algorithm 2 that led to a successful sufficient reduction step. An alternative, and one that we propose, is to instead choose the best from amongst all points in the design set $\mathcal{Y}_k$ that were observed during model construction. No such step needs to be performed after unsuccessful iterations. The following steps formally list the heuristic we propose for updating an iterate after a successful step.

(a) When the iteration is successful,
- if the candidate point does not yield the best function estimate, that is,

$$\min_{\boldsymbol{Y}\in\mathcal{Y}_k} \bar{F}\left(\boldsymbol{Y}, N(\boldsymbol{Y})\right) \leq \bar{F}\left(\tilde{\boldsymbol{X}}_{k+1}, N\left(\tilde{\boldsymbol{X}}_{k+1}\right)\right)$$

accept $\boldsymbol{Y}_{\min} := \arg\min_{\boldsymbol{Y}\in\mathcal{Y}_k} \bar{F}\left(\boldsymbol{Y}, N(\boldsymbol{Y})\right)$ as the new iterate, and replace an existing point in $\mathcal{Y}_k$ (one located farthest from the new iterate) with the candidate point;
- else, that is, if the candidate point provides the best (lowest) estimated function value, update the next iterate to the candidate point.

Keep $\boldsymbol{X}_k$ in the set $\mathcal{Y}_{k+1}$ if it does not provide the worst (largest) estimated function value.

(b) When the iteration in unsuccessful: choose the current iterate as the iterate that starts the next iteration.

4 NUMERICAL EXPERIENCE

In this section, we report ASTRO-DF's performance on 20 nonlinear sum of squares problems included in CUTEst (Gould et al. 2015) library of problems. The dimensionality of the chosen problems varies from 2 to 8. The objective function for all problems in the set takes the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} f_i^2(\boldsymbol{x}), \tag{4}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is smooth, and most of the functions f_i are non-convex. The "noisy" observations are obtained by adding a normal random variable $\xi_i \sim \mathrm{N}\left(0, \sigma^2\right)$ to the sum, that is, $F_i(\boldsymbol{x}) = f(\boldsymbol{x}) + \xi_i$.

ASTRO-DF was executed until a specified simulation budget is exhausted. Suppose the specified simulation budget for ASTRO-DF is n_{total} and let $\boldsymbol{X}^i_{k_{max}}$ denotes the solution returned by the i-th execution of ASTRO-DF on a specific problem. If ASTRO-DF is executed m times, resulting in the m returned solutions $\boldsymbol{X}^i_{k_{max}}, i = 1,2,\ldots,m$, the estimated expectation and estimated square-root variance of the *true* optimality gap of ASTRO-DF's returned solution are given by

$$\begin{aligned}
\hat{\mathbb{E}}[f(\boldsymbol{X}^i_{k_{max}}) - f(\boldsymbol{x}^*)] &:= m^{-1}\sum_{j=1}^{m} f(\boldsymbol{X}^j_{k_{max}}) - f(\boldsymbol{x}^*);\\
\sqrt{\hat{\mathbb{V}}(f(\boldsymbol{X}^j_{k_{max}}) - f(\boldsymbol{x}^*))} &:= \sqrt{(m-1)^{-1}\sum_{j=1}^{m}(f(\boldsymbol{X}^j_{k_{max}}) - f(\boldsymbol{x}^*))^2},
\end{aligned} \tag{5}$$

where $f(\boldsymbol{x}^*)$ is the known minimum value attained by the function f. Each row in Table 1 corresponds to a specific problem in CUTEst and reports $\hat{\mathbb{E}}[f(\boldsymbol{X}^i_{k_{max}}) - f(\boldsymbol{x}^*)]$ and $\sqrt{\hat{\mathbb{V}}(f(\boldsymbol{X}^j_{k_{max}}) - f(\boldsymbol{x}^*))}$ (in parenthesis) for $m = 20$ independent executions of ASTRO-DF. A calculation similar to (5) for true gradient norms is

$$\begin{aligned}
\hat{\mathbb{E}}[\|\nabla f(\boldsymbol{X}^i_{k_{max}})\|] &:= m^{-1}\sum_{j=1}^{m} \|\nabla f(\boldsymbol{X}^i_{k_{max}})\|;\\
\sqrt{\hat{\mathbb{V}}(\|\nabla f(\boldsymbol{X}^i_{k_{max}})\|)} &:= \sqrt{(m-1)^{-1}\sum_{j=1}^{m} \|\nabla f(\boldsymbol{X}^i_{k_{max}})\|^2}.
\end{aligned} \tag{6}$$

It is important to note that since the convergence theory for ASTRO-DF only guarantees convergence to a stationary point, nothing can be said about the behavior of the true optimality gap even as the budget tends to infinity.

Table 1 and Table 2 suggest that ASTRO-DF exhibits consistent and steady progress toward a stationary point across different problems. As is evident from the reported values for small budgets, ASTRO-DF's iterates rapidly approach a stationary point during the initial iterations, with the transient phase being longer for higher dimensional problems. The progress then seems to slow down in the later iterations, when the $O(1/\sqrt{n})$ Monte Carlo rate appears to become effective. Also, unlike optimality gaps expressed using function values, the optimality gaps measured in terms of the gradient norm (reported in Table 2) sometimes exhibit jumps. This could be due to the existence of "cliffs" in the objective function terrain that cause ASTRO-DF to suddenly encounter new stationary regions. Consistent with what is generally known to be characteristic of derivative-free trust-region algorithms in the deterministic context, the behavior of ASTRO-DF is generally stable but somewhat slow.

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

dim	Problem	Initial Gap	$n_{total} = 500$	$n_{total} = 1000$	$n_{total} = 5000$	$n_{total} = 10000$	$n_{total} = 20000$
2	CUBE	1,664,640,225.00	166.8 (12.01)	115.85 (71.15)	2.75 (0.06)	2.73 (0.07)	2.73 (0.07)
	DENSCHNB	83,493.00	223.26 (0.04)	83.19 (34.97)	0.22 (0.18)	0.08 (0.09)	0.06 (0.05)
	DENSCHNC	17,053,704.00	82.03 (159.96)	3.57 (3.59)	0.08 (0.07)	0.06 (0.06)	0.07 (0.07)
	DENSCHNF	6,825,024.00	63.91 (60.74)	5.53 (0.39)	0.05 (0.06)	0.03 (0.03)	0.03 (0.02)
	ROSENBR	7,398,689.00	3,228.74 (4,072.37)	14.47 (13.28)	1.52 (1.54)	0.80 (1.02)	0.54 (0.79)
	S308	589,825.00	1.20 (0.03)	0.97 (0.13)	0.87 (0.07)	0.85 (0.06)	0.85 (0.07)
	SINEVAL	265,359.79	62.54 (0.05)	62.54 (0.05)	37.78 (4.73)	33.86 (1.12)	25.61 (1.74)
3	BEALE	4,314,111,706.20	2,689.17 (8.31)	0.60 (0.01)	0.60 (0.01)	0.58 (0.04)	0.57 (0.03)
	DENSCHND	4,880,138,240.00	965,257.37 (215,883.46)	90,575.16 (28,683.84)	5,767.2 (8,664.57)	392.77 (502.34)	54.36 (107.3)
	DENSCHNE	57,857.00	127.92 (17.80)	74.80 (49.51)	5.60 (3.67)	1.41 (1.01)	1.05 (0.04)
	ENGVAL2	1,654,165.00	285,405.96 (98,770.26)	142,720.60 (93,007.83)	6,278.57 (7,539.4)	531.84 (850.6)	59.99 (93.88)
	YFITU	7,532.36	7,532.36 (224.99)	397.51 (0.01)	397.51 (0.01)	393.07 (5.36)	378.64 (6.02)
4	BROWNDEN	1,109,286,386.27	2,805,815.83 (384,187.4)	345,556.84 (196,785.90)	77,538.60 (63,343.09)	15,282.92 (11,667.43)	1,346.47 (2,564.37)
	HELIX	62,036.77	21.20 (1.99)	5.75 (1.04)	4.47 (2.23)	1.89 (1.68)	0.79 (0.08)
	HIMMELBF	18,223,594.79	25,133.51 (906.62)	24,919.70 (116.58)	24,919.70 (116.58)	23,718.34 (402.6)	22,082.45 (969.11)
	KOWOSB	407.52	1.09 (0.52)	0.22 (0.23)	0.21 (0.24)	0.21 (0.24)	0.21 (0.24)
6	PALMER5C	57,060.56	0.17 (0.42)	0.05 (0.06)	0.05 (0.06)	0.05 (0.06)	0.05 (0.06)
8	PALMER6C	234,351,624.60	9,244.66 (1,979.12)	8,337.96 (2,974.69)	7,950.99 (3,188.69)	6,137.43 (1,530.45)	5,660.21 (1,925.74)
	PALMER7C	938,543,568.98	19,198.73 (3,970.22)	17,806.34 (4,397.08)	6,511.37 (4,073.36)	5,848.43 (3,960.98)	5,007.86 (3,843.73)
	PALMER8C	284,731,957.52	45,662.36 (4,776.17)	6,815.97 (7,067.04)	5,410.86 (5,841.45)	4,677.09 (5,067.15)	3,723.63 (3,979.81)

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

dim	Problem	Initial Gradient	$n_{total} = 500$	$n_{total} = 1000$	$n_{total} = 5000$	$n_{total} = 10000$	$n_{total} = 20000$
2	CUBE	626,688,561.25	5,641.58 (215.93)	4,148.01 (2,133.23)	55.23 (38.39)	26.35 (18.69)	23.19 (24.82)
	DENSCHNB	13,918.26	115.42 (0.63)	25.31 (14.66)	1.0 (0.32)	0.59 (0.41)	0.51 (0.21)
	DENSCHNC	6,327,252.19	99.51 (180.17)	9.83 (20.25)	1.13 (1.08)	1.04 (0.75)	0.97 (0.75)
	DENSCHNF	1,228,253.22	1.91 (0.05)	42.89 (1.77)	4.43 (2.82)	3.04 (1.56)	2.77 (0.27)
	ROSENBR	1,741,683.78	2,215.21 (2,450.05)	95.27 (69.98)	11.19 (10.03)	3.67 (2.65)	2.9 (2.18)
	S308	104,267.14	1.07 (0.01)	0.74 (0.23)	0.57 (0.17)	0.56 (0.19)	0.52 (0.2)
	SINEVAL	45,109.93	418.43 (0.43)	418.43 (0.43)	23.04 (16.16)	15.12 (11.84)	14.22 (12.6)
3	BEALE	1,702,889,243.95	6154.35 (14.66)	223.04 (201.23)	26.53 (23.07)	13.63 (11.33)	9.11 (10.47)
	DENSCHND	2,284,598,497.73	671,367.94 (102,757.39)	87,701.12 (35,407.02)	3,257.27 (3,186.4)	665.65 (676.33)	90.61 (159.8)
	DENSCHNE	14,880.03	29.08 (4.72)	22.55 (12.75)	3.10 (3.48)	0.76 (1.22)	0.36 (0.23)
	ENGVAL2	1,328,958.48	157,877.43 (86,706.09)	91,714.81 (81,874.81)	7,955.82 (12,353.36)	1,018.2 (1,486.14)	138.61 (87.25)
	YFITU	6,698.93	6,698.93 (0.00)	894.61 (0.00)	894.61 (0.00)	533.38 (369.76)	164.9 (105.88)
4	BROWNDEN	14,484,286.01	705,654.61 (195,512.81)	74,951.12 (34,455.6)	23,518.29 (11,544.48)	12,489.10 (6,017.72)	4,301.18 (4,244.95)
	HELIX	4,989.97	185.34 (10.03)	45.38 (0.90)	38.23 (13.02)	27.41 (15.79	5.14 (2.42)
	HIMMELBF	4,787,770.96	178,763.72 (85,856.99)	11,085.61 (7,056.75)	1,819.48 (948.6)	1,373.75 (525.99)	1,295.14 (405.65)
	KOWOSB	76.84	1.78 (0.53)	0.66 (0.46)	0.62 (0.47)	0.62 (0.47)	0.62 (0.47)
6	PALMER5C	1,510.43	1.31 (1.44)	0.78 (0.49)	0.86 (0.51)	0.86 (0.51)	0.86 (0.51)
8	PALMER6C	19,027,217.80	3,090.00 (3,960.13)	3,186.54 (1,846.33)	2,188.84 (1,097.99)	2,216.65 (1,26.77)	2,003.91 (989.47)
	PALMER7C	80,645,963.34	146,744.86 (218,010.63)	177,031.03 (135,919.20)	8,473.06 (12,862.42)	7,679.50 (11,881.42)	4,241.71 (6,463.97)
	PALMER8C	22,681,728.35	13,119.43 (12,556.37)	7,589.97 (7,386.3)	2,847.42 (2,274.93)	2,101.04 (2,547.86)	1,670.81 (1,388.25)

5 CONCLUDING REMARKS

Over the last decade or so, derivative-free trust-region algorithms have enjoyed great attention and success in the deterministic context. We believe that the question of developing analogous algorithms for the Monte Carlo context is poorly studied; it is worthy of further inquiry because the settings for which derivative-free trust-region methods are devised seem predominant within Monte Carlo contexts. In this paper, we make some important initial steps towards answering this question. Specifically, we experiment our newly developed adaptive sampling trust-region optimization algorithms (called ASTRO-DF) for solving low to moderate dimensional simulation optimization problems. The algorithms we propose are convergent, but more importantly, gain practical efficiency through certain key steps related to adaptive sampling, model certification, and the careful balancing of interpolation, sampling, and model errors. The numerical results reported in this study confirm that ASTRO-DF is a reasonable framework for simulation oracles with low to high variability. Ongoing research focuses on a number of theoretical and practical issues within ASTRO-DF, some of which we summarize below.

(i) Convergence theory for ASTRO-DF dictates the asymptotic sampling rate to be $O\left(\Delta_k^{-4}\right)$, where Δ_k is the incumbent trust-region radius. A similar requirement has been prescribed by two other recent prominent investigations (Chen et al. 2015, Larson 2012). Is this sampling rate fundamental in any sense? Does the $O\left(\Delta_k^{-4}\right)$ rate translate to the $O(1/\sqrt{n})$ Monte Carlo canonical rate?

(ii) Unlike the deterministic TRO-DF algorithm (Conn et al. 2009a), ASTRO-DF includes a model construction step in every iteration. It seems that such a stringent requirement can be relaxed without sacrificing convergence guarantees. In fact, the implementation of ASTRO-DF that we have used in Section 4 does just that by adding a criticality step which stipulates that the model construction step be invoked only when the model gradient is sufficiently small. Such a simple rule improves practical efficiency; whether it preserves convergence is an open question.

(iii) A crucial unresolved issue that is somewhat related to the remark in (ii) relates to the manner of model construction. Specifically, how should the model construction step balance re-using already visited points with carefully placed new points within the trust-region? While using already visited points enhance efficiency by preserving simulation budget, they invariably result in poorer models because iterates visited by ASTRO-DF tend to be highly spatially correlated.

(iv) The slowly increasing sequence $\{\lambda_k\}$ ensures that the sample sizes within ASTRO-DF are forced to infinity asymptotically, and that the effects of infrequent spurious observations are limited. Our numerical experience strongly suggests that $\{\lambda_k\}$ is only rarely binding, and almost never so asymptotically. Can it be established that the probability of the lower bound sequence $\{\lambda_k\}$ being binding infinitely often is zero with probability one?

REFERENCES

Alagoz, O., A. J. Schaefer, and M. S. Roberts. 2009. "Optimizing Organ Allocation and Acceptance". In *Handbook of Optimization in Medicine*, 1–24. Springer.

Asmussen, S., and P. W. Glynn. 2007. *Stochastic Simulation: Algorithms and Analysis*, Volume 57 of *Stochastic modeling and applied probability*. Springer-Verlag New York.

Chen, R., M. Menickelly, and K. Scheinberg. 2015. "Stochastic Optimization Using a Trust-Region Method and Random Models". *Mathematical Programming*. Submitted.

Conn, A. R., K. Scheinberg, and L. N. Vicente. 2009a. "Global Convergence of General Derivative-Free Trust-Region Algorithms to First-and Second-Order Critical Points". *SIAM Journal on Optimization* 20 (1): 387–415.

Conn, A. R., K. Scheinberg, and L. N. Vicente. 2009b. *Introduction to Derivative-Free Optimization*, Volume 8. Society for Industrial and Applied Mathematics.

Gould, N. I., D. Orban, and P. L. Toint. 2015. "CUTEst: A Constrained and Unconstrained Testing Environment with Safe Threads for Mathematical Optimization". *Computational Optimization and Applications* 60 (3): 545–557.

Hou, Y. T., Y. Shi, and H. D. Sherali. 2014. *Applied Optimization Methods for Wireless Networks*. Cambridge University Press.

Larson, J. M. 2012. *Derivative-Free Optimization of Noisy Functions*. Ph. D. thesis, Department of Applied Mathematics, University of Colorado, Denver, CO.

Nocedal, J., and S. J. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research and Financial Engineering. New York: Springer.

Nsoesie, E. O., R. J. Beckman, S. Shashaani, K. S. Nagaraj, and M. V. Marathe. 2013. "A Simulation Optimization Approach to Epidemic Forecasting". *PloS one* 8 (6): e67164.

Osorio, C., and M. Bierlaire. 2009. "A Surrogate Model for Traffic Optimization of Congested Networks: An Analytic Queueing Network Approach". *Report TRANSP-OR* 90825:1–23.

Pasupathy, R., and S. G. Henderson. 2006. "A Testbed of Simulation-Optimization Problems". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 255–263. Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc.

Powell, M. J. D. 1994. "A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation". In *Advances in Optimization and Numerical Analysis*, Mathematics and Its Applications. Kluwer Academic, Dordrecht.

Rani, D., and M. M. Moreira. 2010. "Simulation-Optimization Modeling: A Survey and Potential Application in Reservoir Systems Operation". *Water Resources Management* 24 (6): 1107–1138.

Shashaani, S., F. S. Hashemi, and R. Pasupathy. 2015. "ASTRO-DF: A Class of Adaptive Sampling Trust-Region Algorithms for Derivative-Free Simulation Optimization". *Under review at SIAM Journal of Optimization*.

AUTHOR BIOGRAPHIES

SARA SHASHAANI is PhD student in the Department of Industrial Engineering at Purdue University. Her research interests include Monte Carlo methodology, simulation optimization theory and algorithms, and derivative-free optimization. She is the recipient of INFORMS best poster presentation award in 2012, and the Ross Fellowship at Purdue in 2014-15. Her email address is sshashaa@purdue.com.

SUSAN R. HUNTER is an assistant professor in the School of Industrial Engineering at Purdue University. Her research interests include Monte Carlo methods and simulation optimization, especially in the presence of multiple performance measures. Her e-mail address is susanhunter@purdue.edu, and her website is http://web.ics.purdue.edu/~hunter63/.

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 simulation optimization. He is a member of INFORMS, IIE, and ASA, and serves as an associate editor for Operations Research and INFORMS Journal on Computing. He is the Area Editor for the Simulation Desk at IIE Transactions. His email address is pasupath@purdue.edu and his web page is http://web.ics.purdue.edu/ pasupath.