Journal Pre-proof

An Extended Two-Stage Sequential Optimization Approach: Properties and Performance

Giulia Pedrielli, Songhao Wang, Szu Hui Ng

PII: S0377-2217(20)30398-2
DOI: https://doi.org/10.1016/j.ejor.2020.04.045
Reference: EOR 16496

To appear in: *European Journal of Operational Research*

Received date: 18 October 2018
Accepted date: 23 April 2020

Please cite this article as: Giulia Pedrielli, Songhao Wang, Szu Hui Ng, An Extended Two-Stage Sequential Optimization Approach: Properties and Performance, *European Journal of Operational Research* (2020), doi: https://doi.org/10.1016/j.ejor.2020.04.045

Manuscript: EJOR-S-18-03478

Title: “An Extended Two Stage Sequential Optimization Approach: Properties and Performance”

Authors: Giulia Pedrielli, Songhao Wang, Szu Hui Ng

File Type: HIGHLIGHTS

- eTSSO solves optimization problems extending the previous TSSO algorithm eliminating the need to select the budget per iteration.
- A theoretical adaptive budget allocation is provided.
- Four algorithmic variants were implemented.
- eTSSO asymptotic properties are theoretically analyzed.
- eTSSO in its implementations improves on TSSO under different classes of functions, noise levels and total budget.

An Extended Two-Stage Sequential Optimization Approach: Properties and Performance

Giulia Pedrielli

School of Computing, Informatics, and Decision Systems Engineering

Arizona State University.

Songhao Wang, Szu Hui Ng

Department of Industrial Systems Engineering and Management,
National University of Singapore.

Abstract

This paper looks into nonlinear non convex stochastic unconstrained optimization with finite simulation budget. Our work builds upon the Two-Stage Sequential Optimization (TSSO) algorithm that addresses the class of problems of interest by using the modified nugget effect kriging (MNEK) metamodel and proposing a budget allocation followed by a two-stage sequential procedure. Despite its efficiency and performance, we have observed that, given a finite budget, the choice of the number of replications per iteration, currently left to the user, is particularly critical for the algorithm performance. A fixed a-priori assignment can affect the ability to control the algorithm making it particularly sensitive to the initial settings. In this paper, we propose the extended TSSO (eTSSO). Specifically, a general simulation budget allocation scheme is proposed with the objective to balance the need of accurate function estimations to improve the selection in the search stage, with the need to explore the solution space. The new scheme adaptively, and recursively, increases the simulation budget based upon information iteratively returned by the optimizer itself. We analyze the asymptotic properties of eTSSO. Subsequently, we propose four alternative variants of the general allocation that we empirically analyze by comparing the quality of the estimated optimum input combination and the corresponding estimated optimum output against TSSO and other state of the art algorithms.

Keywords: Global Optimization; Simulation-Optimization; Two-Stage Sequential Optimization; eTSSO; Convergence

1 Introduction and Motivation

In several real world applications, the behavior of large complex systems is highly nonlinear. As a result, approaches grounded in the exploitation of structural properties of functions emulating this behavior have been replaced by black box approaches (Fu, 2015). This large family of methods considers the function as unknown and it only assumes that point estimates can be obtained by repeatedly calling a simulation oracle (i.e., the black box) (Wright and Nocedal, 1999). A search algorithm is then responsible, using a variable level of memory and complexity, to process the estimations and produce sampling decisions until a stopping criterion is met (Tekin and Sabuncuoglu, 2004).

In this work, we assume that the point evaluations returned by the oracle are affected by noise and we design an algorithm that produces an estimate of the location $\boldsymbol{x}$ minimizing the possibly nonlinear non-convex function of interest, i.e., $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{X}} E\left(f\left(\boldsymbol{x}\right)\right)$, where $\mathbb{X}$ represents the solution set, assumed continuous in this research, and $f\left(\boldsymbol{x}\right)$ is the univariate function whose noisy measurements can be obtained through the oracle (also referred to as simulator in the remainder of the manuscript).

Two families of approaches, in both deterministic and stochastic settings, can be identified in a way that is relevant to this work: (1) direct methods, calling the simulator at each iteration to obtain an estimate of the response, and typically calculating a direction for the next move, and (2) surrogate methods, which use simulation to estimate a metamodel of the response surface and use this model to guide the selection of the next sampled point (Tekin and Sabuncuoglu, 2004; Kleijnen, 2008; Zhu et al., 2013; Figueira and Almada-Lobo, 2014; Fu, 2015; Xu et al., 2015; Jalali et al., 2017).

Popular direct algorithms include COMPASS (Xu et al., 2010), R-SPLINE (Wang et al., 2013), Ranking and Selection method (Kim and Nelson, 2007), stochastic approximation methods (Yin and Kushner, 2003), the Nested Partitions Method (Shi and Ólafsson, 2000), and the recent ASTRO-DF (Shashaani et al., 2018).

A major drawback of this family of approaches is their cost when simulation runs require high computational effort. In such cases, meta-modeling based (surrogate) search offers the possibility to use the information coming from the simulation runs to infer about regions where simulation has not been performed. Specifically, surrogate methods use a few simulation runs to estimate a model of the response (i.e., the meta-model), which can be used by the search procedure to quickly evaluate the performance at any given location in the design space without the need to run the simulator (Wan et al., 2005). Response Surface Methodology (RSM) (Myers et al., 2009) is among the most popular techniques in this class due to its ease of implementation. RSM uses first-order linear regression models and switches to second order models when approaching a local minimum. These models are fitted with respect to a sequence of local regions, and they are used to guide the search towards the optimum. A more complex and commonly adopted model form is Kriging, also known as Gaussian process (GP) modeling, which has been particularly successful in deterministic computer experiments (Santner et al., 2003). Recently, a noticeable effort has been dedicated to extend the kriging model structure to the case of stochastic simulations, including homoscedastic (homogeneous random noise in the design space) and heteroscedastic (heterogeneous random noise in the design space) cases. Specifically, for heteroscedastic simulations, Ankenman et al. (2010); Yin et al. (2011) proposed the Stochastic Kriging (SK) model and the Modified Nugget Effect Kriging (MNEK), respectively.

In both direct and surrogate search, when the total number of available simulations is finite, the way the simulation runs are allocated to the sampled points is a critical decision. Optimal Computing Budget Allocation (OCBA) has received particular attention in this regard, especially within the Ranking and Selection literature and, more recently, in the area of kriging-based simulation-optimization. As an example, Quan et al. (2013) proposed the Two-Stage Sequential Optimization (TSSO) algorithm to solve unconstrained stochastic simulation-optimization problems for the heteroscedastic case, where, in the first stage, called the "search", TSSO uses the MNEK model to explore the region and to determine the next point to sample and in the second stage, called the "evaluation", it runs simulation experiments at each sampled point according to OCBA (Chen et al., 2000). The kriging model is updated using the simulation

results, and the algorithm proceeds. TSSO has been empirically shown to perform well, but it requires several user defined parameters. In particular, the number of simulation replications B to be performed at each iteration has to be provided as input and it has an important impact on the effectiveness of the procedure.

We propose and analyze the extended TSSO (eTSSO), which generates an adaptive sequence of values of B ($\{B_k\}$, where k is the iteration index) as the algorithm progresses. This mitigates the possible effect of an inappropriately selected value of B in TSSO.

A first version of the algorithm eTSSO was used in the comparison paper (Jalali et al., 2017). In this manuscript, with the objective to study the theoretical properties of eTSSO, we provide a new general allocation scheme and four alternative variants that manage differently the balance between the need of accurate function estimations to improve the selection in the search stage, with the need to explore the solution space. All the variants we propose are novel. As the original first version of eTSSO, implemented in (Jalali et al., 2017) does not satisfy our general allocation rule conditions, it is not considered in this study. In Mehdad and Kleijnen (2018), the authors study an optimal computing budget allocation variant for the derivation of the budget at each iteration. The paper is empirical in nature and did not identify a winning algorithm for the case of random simulations, which is the focus of this paper. A key advantage of TSSO and, consequently, eTSSO is that no structure of the noise function is required for the algorithms to work. While surely the knowledge of the structure of the noise leads to better results (when such knowledge is correct), as recognized in (Jalali et al., 2017), in real cases we will rarely have such information. Nonetheless, if the noise structure is known, we would recommend to make use of an algorithm that can exploit that knowledge. Concerning the theoretical contribution of the paper, we highlight that a preliminary conference version is in (Pedrielli and Ng, 2015), where a first proof of concept of the convergence analysis was presented. This paper develops and presents eTSSO, for the first time, along with the new analysis of the convergence and the convergence rates and its relationship with the stochastic budget allocation scheme.

The remainder of the paper is structured as follows: section 2 presents the background to the presented work. In section 3, eTSSO is presented with the four budget allocation variants. Section 4 characterizes the behavior of eTSSO under the theoretical budget scheme in terms of both convergence as well as convergence rates. In section 5, eTSSO in its four variants is tested over multi-dimensional functions to assess its performance against TSSO. Finally, section 6 draws the conclusions of the paper.

2 Background: Meta-modeling and optimization with stochastic kriging

In this work, we will assume that the nonlinear optimization problem is defined over a compact solution set $\mathbb{X}$. While the original objective function $Y : \boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^d \rightarrow Y(\mathbf{x}) \in \mathbb{R}$ is deterministic in nature, the oracle is affected by noise. Therefore, when we run simulation at a specific location $\boldsymbol{x} \in \mathbb{X} \subseteq \mathbb{R}^d$, only an estimate of the function value is returned. The objective is to develop an efficient search algorithm that

finds the global minimum of $Y : \mathbb{X} \to \mathbb{R}$, namely:

$$P : \min Y(\boldsymbol{x}) = \mathbb{E}_Y [f(\boldsymbol{x})] \quad \text{s.to} \quad \boldsymbol{x} \in \mathbb{X} \tag{1}$$

where $\mathbb{E}_Y$ refers to the, unknown, expectation of $f(\boldsymbol{x})$ that can only be estimated pointwise, by running expensive simulations. Section 2.1 provides some preliminaries on the deterministic version of problem P, which will be particularly helpful in the convergence analysis in section 4, whereas section 2.2 provides more details on the adopted Kriging model in the stochastic setting and the related optimization procedure.

2.1 Deterministic Simulation-Optimization with Kriging

In deterministic settings, the Efficient Global Optimization (EGO) method, derived from the Bayesian framework, has been the basis for most of the Kriging based search algorithms (Jones et al., 1998). In this framework, $f(\boldsymbol{x})$ is interpreted as a realization from an infinite family of random functions, namely, stationary Gaussian process $Y(\boldsymbol{x})$. According to this interpretation, at points $\boldsymbol{x}$ that have not been simulated, we assume that the function $y(\mathbf{x})$ is jointly Gaussian and can be fully characterized by the mean, and covariance functions. We will refer to the statistical model of the unknown function Y as π, which we characterize in terms of optimal predictor and predictive error as (Santner et al., 2003; Bull, 2011)

$$\hat{Y}_\pi(\boldsymbol{x}) := \hat{\mu}_\pi + \mathbf{c}^T \mathbf{R}^{-1} (\mathbf{y} - \hat{\mu}_\pi 1), \tag{2}$$

$$s^2_{\pi_k}(\boldsymbol{x}) := \tau^2 \left(1 - \mathbf{c}^T \mathbf{R}^{-1} \mathbf{c} + \frac{\left(1 - 1^T \mathbf{R}^{-1} \mathbf{c}\right)^2}{1^T \mathbf{R}^{-1} 1} \right). \tag{3}$$

Where,

$$\hat{\mu}_\pi := \frac{1^T \mathbf{R}^{-1} \mathbf{y}}{1^T \mathbf{R}^{-1} 1}. \tag{4}$$

Here, 1 is a vector having all elements equal to 1, $\mathbf{R} = (\mathbb{K}_\phi(\boldsymbol{x}_i - \boldsymbol{x}_j))_{i=1}^k$ is the spatial variance-covariance matrix of kernel $\mathbb{K}$ parametrized by ϕ, where k is the number of sampled points. Assuming an exponential form for the variance covariance matrix, we have

$$\mathbf{R}(Y(\boldsymbol{x}_i), Y(\mathbf{x}_j)) = \prod_{l=1}^{d} \exp\left(-\phi_{z,l} |x_{i,l} - x_{j,l}|^t\right). \tag{5}$$

Where the scale correlation coefficient $\phi_{z,l}$ is the sensitivity parameter that controls how fast the correlation decays with the distance between points $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in the l-th dimension, and the parameter t controls the smoothness of the response. When $t = 1$, equation (5) is known as the exponential correlation function, and when $t = 2$ it is known as the Gaussian correlation function (Santner et al., 2003; Picheny et al.,

2013), used in this paper. Assuming this type of correlation, the vector $\mathbf{c} = (\mathbf{R}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_i))_{i=1}^{k}$ results in

$$\mathbf{c}(\boldsymbol{x}, \cdot; \phi_z)^T = \left(e^{-\phi_z d^2_{\boldsymbol{x},\boldsymbol{x}_1}} \cdots e^{-\phi_z d^2_{\boldsymbol{x},\boldsymbol{x}_k}}\right). \tag{6}$$

Where, $d_{\boldsymbol{x},\boldsymbol{x}_i}$ represents the Euclidean distance between the prediction point $\boldsymbol{x}$ and the location $\boldsymbol{x}_i$, $i = 1, \ldots, k$.

EGO is an iterative algorithm that, at each step, estimates the *expected improvement* and samples its maximizer. As the algorithm progresses, a random sequence of sampled points $\{\boldsymbol{x}_k\}$ is generated as well as the sequence of the estimates of the optimum location, i.e., $\{\boldsymbol{x}_k^*\}$ over the compact space $\mathbb{X}$. The available data at the k-th algorithm iteration, constitute the set $\mathcal{F}_{\pi_k}$, the *filtration*, made by the σ-algebra containing the collection of $(\boldsymbol{x}_i, y(\boldsymbol{x}_i) : i \leq k)$. Given $\mathcal{F}_{\pi_k}$, the best guess of the optimum at iteration k, referred to as $\boldsymbol{x}_k^*$, is the location in the sampled set $\mathbb{S}_k \subset \mathbb{X}$ that achieved the best function value so far. Formally, at iteration k, the algorithm samples the location that maximizes the *Expected Improvement* function defined as (Jones et al., 1998)

$$T_{\pi_k}(\boldsymbol{x}, \mathcal{F}_{\pi_k}) := \max\left(\mathbb{E}_\pi\left[y(\boldsymbol{x}_k^*) - \hat{Y}(x) | \mathcal{F}_{\pi_k}\right], 0\right). \tag{7}$$

Where, $\hat{Y}(\boldsymbol{x})$ is the prediction produced by the meta-model at location $\boldsymbol{x}$.

2.2 Stochastic Simulation-Optimization with Kriging

For the case of function measurements affected by noise, Huang et al. (2006) proposed to use the nugget effect kriging model (which assumes constant variance throughout the sample space) replacing the EI criterion with the Augmented Expected Improvement (AEI) in order to deal with noisy function measurements.

To consider the heterogeneous variance and the finite simulation budget, Quan et al. (2013) proposed the Two-Stage Sequential Optimization (TSSO) algorithm which relies on the Modified Nugget Effect Kriging (MNEK) model (Yin et al., 2011). According to Yin et al. (2011), $y(\boldsymbol{x}_i)$ is the output from the stochastic simulation at $\boldsymbol{x}_i \in \mathbb{X}$, and it assumes that $y(\boldsymbol{x}_i)$ are realizations of a random process that can be described by the model $\tilde{\pi}$ defined as

$$\tilde{\pi} := Y(\boldsymbol{x}_i) = Z(\boldsymbol{x}_i) + \xi(\boldsymbol{x}_i) \quad i = 1, \ldots, k. \tag{8}$$

The general form of equation (8) is similar to that proposed in Ankenman et al. (2010). Also, similarly to the deterministic case, $Z(\boldsymbol{x}_i)$ is modeled as a Gaussian process with covariance function $\tau^2\mathbf{R}_z$, where τ^2 is the process variance and $\mathbf{R}_z$ the matrix of process correlation; formally, $Z(\boldsymbol{x}_i)$ is a $GP\left(\mu(\boldsymbol{x}), \tau^2\mathbf{R}_z\right)$. A commonly adopted correlation function $\mathbf{R}_z$ was presented in equation (5) for the deterministic optimization case. The noise term $\xi(\boldsymbol{x})$ in equation (8) represents the major difference from the deterministic setting. The noise process is typically assumed centered around zero and having as variance covariance function $\sigma_\xi^2\mathbf{R}_\xi$. Intuitively, $\mathbf{R}_\xi$ models the correlation that arises from dependencies of the pseudorandom numbers employed by the oracle (simulator). The random component is referred to "intrinsic variance" to be distinguished from the "extrinsic variance", which represents the model variance. Error variances are generally not constant and they may depend on $\boldsymbol{x}$. With independent sampling (i.e., no Common

Random Numbers, CRN), $\mathbf{R}_\xi$ is diagonal, and equation (8) reduces to the independent sampling noise model (Yin et al., 2011; Ng and Yin, 2012).

Yin et al. (2011) shows how the MSE optimal predictor corresponding to (8) at location $\boldsymbol{x}_0$ when k points have been previously sampled, is

$$\hat{Y}\left(\boldsymbol{x}_0\right) = \sum_{i=1}^{k} \left(\mathbf{c}^T \left(\mathbf{R}_z + \mathbf{R}_\xi\right)^{-1} e_i + \mathbf{1}^T \left(\mathbf{R}_z + \mathbf{R}_\xi\right)^{-1} \frac{\left[1 - \mathbf{1}^T \left(\mathbf{R}_z + \mathbf{R}_\xi\right)^{-1} \mathbf{c}\right]^T}{\mathbf{1}^T \left(\mathbf{R}_z + \mathbf{R}_\xi\right)^{-1} \mathbf{1}} e_i \right) \bar{y}\left(\boldsymbol{x}_i\right). \tag{9}$$

Where $\bar{y}_{i,k}$ is the function evaluations average for the sampled locations $\boldsymbol{x}_i$, with $i = 1, \ldots, k$; $\mathbf{c}\left(\boldsymbol{x}_0, \cdot; \phi_z\right)$ is the correlation vector modeled as in equation (6). e_i is a vector of size k (where k is the number of sampled points) having all elements equal to 0 except the i-th element which is equal to 1. The optimal MSE results (Yin et al., 2011)

$$MSE_{\tilde{\pi}_k}(\boldsymbol{x}_0) = c_0\left(\boldsymbol{x}_0\right) + \tau^2 \left(1 - \left[\mathbf{c} + \mathbf{1} \frac{\left(\mathbf{1} - \mathbf{1}^T \mathbf{R}'^{-1} \mathbf{c}\right)}{\mathbf{1}^T \mathbf{R}'^{-1} \mathbf{1}} \right]^T \mathbf{R}'^{-1} \mathbf{c} + \frac{\left(\mathbf{1} - \mathbf{1}^T \mathbf{R}'^{-1} \mathbf{c}\right)}{\mathbf{1}^T \mathbf{R}'^{-1} \mathbf{1}} \right). \tag{10}$$

Where $\mathbf{R}' = \mathbf{R}_z + \mathbf{R}_\xi$, and c_0 is the nugget effect value. Note that equations (9)-(10) are implemented in several packages (Erickson et al., 2018).

While in equation (10) the parameters (τ, ϕ, c_0) are assumed known, they require estimation (as in the deterministic case). In the stochastic case, the "new" parameter c_0 can be estimated from the sample variance as $\hat{c}_0\left(\boldsymbol{x}_0\right) = \hat{\sigma}_\xi^2\left(\boldsymbol{x}_0\right)/n$. As discussed in (Yin et al., 2011), a closed form estimator for the predicted variance in points that have not been sampled yet is not available. Therefore, as suggested by Yin et al., we obtain $MSE_{\tilde{\pi}_k}(\boldsymbol{x}_0)$, $\boldsymbol{x}_0 \notin \mathbb{S}$, where $\mathbb{S}$ is the set of sampled points, as the piecewise linear interpolation of the available estimates for the sampled points $\boldsymbol{x} \in \mathbb{S}$. In particular, piecewise linear interpolation is used to extrapolate the variance at any un-sampled location using the two closest neighboring points within the sample set (Kleijnen and Van Beers, 2005).
Furthermore, in the rest of the paper, we will refer to the *deterministic counterpart* of the estimate in equation (10) at the k-th iteration as the *extrinsic variance* $s^2_{\tilde{\pi}_k}$, corresponding to the form

$$s^2_{\tilde{\pi}_k}(x_0) = \tau^2 \left(1 - \left[\mathbf{c} + \mathbf{1} \frac{\left(\mathbf{1} - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{c}\right)}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right]^T \mathbf{R}^{-1} \mathbf{c} + \frac{\left(\mathbf{1} - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{c}\right)}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right). \tag{11}$$

Differently from EGO and the Sequential Kriging Optimization (SKO) (Huang et al., 2006), TSSO is a two-stage algorithm which uses the first stage to balance the effort between exploration and exploitation when the total number of available replications is limited. Specifically, it allocates the budget between exploration and exploitation according to the following rule:

$$r_{S,k} = r_{S,k-1} - \Delta_r \text{ and } r_{A,k} = r_{A,k-1} + \Delta_r. \tag{12}$$

Here, $r_{S,k}$ is the number of replications to assign to the search stage at iteration k, $r_{A,k}$ is the number of replications for the evaluation at iteration $k = 1, 2, \ldots, K$, where K is the maximum number of

iterations. $\Delta_r = \lfloor (B - r_{\min})/K \rfloor$ represents the fixed rate of decay (increase) of $r_{S,k}$ ($r_{A,k}$), being B the budget assigned to each iteration i, and $r_{\min}$ the minimum number of replications required to sample a new point. We will denote our total simulation budget as T. Finally, we will let N_0 be the size of the initial design (Latin Hypercube Sampling (LHS) in this manuscript (Kleijnen, 2008; Brochoff et al., 2015)) used to estimate the initial MNEK model $\tilde{\pi}_0$. Given N_0 and T, the maximum number of iterations that can be performed by TSSO is, then, $K = \lfloor (T - N_0 B)/B \rfloor$. Once the budget allocation has been performed, in analogy to the deterministic case, the algorithm uses the MNEK model (equation (9)) to estimate the function values at potential infill points $\boldsymbol{x} \in \mathbb{X} \notin \mathbb{S}$. In this setting, the filtration $\mathcal{F}_{\tilde{\pi}_k}$ is the sigma algebra $\sigma\left(\boldsymbol{x}_i, \bar{y}(\mathbf{x}_i), \hat{\sigma}^2_{\xi}(\mathbf{x}_i) : i \leq k\right)$. In particular, $\bar{\mathbf{y}} \equiv [\bar{y}(\mathbf{x}_1), \ldots, \bar{y}(\mathbf{x}_k)]'$ represents the vector of the sample averages at the selected points $i = 1, \ldots, k$, and $\hat{\boldsymbol{\sigma}}^2_{\xi}(\mathbf{x}_i)$ is the related sample variance.

Similar to the deterministic case, the next location $\boldsymbol{x}_{k+1}$ to be evaluated, and added to the set $\mathbb{S}_k$, maximizes the so-called *modified expected improvement* function $T_{\tilde{\pi}_k}(\mathbf{x})$ introduced in (Quan et al., 2013):

$$T_{\tilde{\pi}_k}(\boldsymbol{x}) = \max\left(\mathbb{E}_{\tilde{\pi}_k}\left[\bar{y}(\boldsymbol{x}^*_k) - \hat{Y}_k(\boldsymbol{x}) \,|\mathcal{F}_{\tilde{\pi}_k}\right], 0\right). \tag{13}$$

Where $\bar{y}(\boldsymbol{x}^*_k)$ represents the lowest *sample mean* up to iteration k, and $\hat{Y}_k(\boldsymbol{x})$ is random with mean corresponding to the Gaussian process mean function at location $\boldsymbol{x}$ and variance given by the model spatial prediction uncertainty $s^2_{\tilde{\pi}_k}(\boldsymbol{x})$. In order to obtain an evaluation from $\boldsymbol{x}_k$ the algorithm allocates $r_{S,k}$ replications to the location that is subsequently added to the set $\mathbb{S}_k$. Besides the underlying model $\tilde{\pi}$, the criterion in (13) differs from (7), since the sample average $\bar{y}(\boldsymbol{x}^*_k)$ needs to replace the true function value $y(\boldsymbol{x}^*_k)$, which is clearly not available.

In the second stage, TSSO uses the OCBA technique to assign the available replications, $r_{A,k} = B - r_{S,k}$, to each of the sampled points $\boldsymbol{x} \in \mathbb{S}$. Specifically, the authors use the results in (Chen et al., 2000) to compute the relative budget allocation between non best locations (equation (14)), and the relative budget allocation between the location associated with the best function value and the rest of the sampled points (equation (15)).

$$n_i/n_j = \left(\frac{\hat{\sigma}_{\boldsymbol{x}_i}(\boldsymbol{x}_i)/\delta_{b,i}}{\hat{\sigma}_{xj}(\boldsymbol{x}_j)/\delta_{b,j}}\right)^2, \tag{14}$$

$$n_b = \hat{\sigma}_{\boldsymbol{x}_b}(\boldsymbol{x}_b)\sqrt{\sum_{x \in \mathbb{X}: x \neq \boldsymbol{x}_b} \frac{n_i^2}{\hat{\sigma}^2_{\boldsymbol{x}_i}}}, \tag{15}$$

Here $\boldsymbol{x}_b, \hat{\sigma}_{\boldsymbol{x}_b}(\boldsymbol{x}_b), n_b$ are the sampled location with the lowest function value estimate, the associated sample standard deviation, and the allocated number of simulations, respectively. Similarly, $\hat{\sigma}_{\boldsymbol{x}_i}, n_i$ are the estimated standard deviation at location $\boldsymbol{x}_i$, and the number of associated replications, respectively. Finally, $\delta_{b,i}$ is the difference between the function estimation at location $\boldsymbol{x}_i$ and $\boldsymbol{x}_b$.

In the TSSO algorithm, the budget B available at each iteration is chosen at the algorithm start and it stays constant as the search progresses. Then, the TSSO allocation rule is such that the constant budget B is dynamically divided between the search ($r_{S,k}$) and the evaluation ($r_{A,k}$) (equation (12)).

From equation (12), we observe that, since $r_{A,k} \geq 0$, B must satisfy the following condition:

$$B \geq B_{\min} = \left\lceil \frac{r_{\min} - N_0 + \sqrt{(N_0 - r_{\min})^2 + 4T}}{2} \right\rceil \tag{16}$$

When the assigned budget B is below this limit, TSSO will only perform search, i.e., $r_{A,k} = 0 \ \forall k$. Also, the value in (16) bounds from above the number of iterations that TSSO will perform:

$$K_{\max} = \left\lfloor \frac{T - B_{\min} N_0}{B_{\min}} \right\rfloor \tag{17}$$

3 eTSSO Algorithm

Although TSSO has been effectively applied to real world problems (Quan et al., 2013), its performance is influenced by the number of simulation replications (i.e., the budget) allocated to each iteration given the total available budget, i.e., the pair (T, B). In fact, B affects the accuracy of the measurements since it determines the maximum number of simulation experiments we can allocate for evaluating the sampled points, and it influences the search by determining the number of iterations (equation (17)). eTSSO tackles the problem of computational effort management, by intelligently and adaptively determining the budget sequence $\{B_k\}$. In order to do so, $\{B_k\}$ is interpreted as a random sequence, instead of a constant as in TSSO, of non-decreasing values, with a random growth rate that can be computed using the information coming from simulation and model estimation. Specifically, at the generic iteration k, we use the estimate of the intrinsic variance at each sampled point and the extrinsic variance (equation (11)), to update the number of replications B_k using the following:

$$B_k = \max\left(\left\lfloor B_{k-1} \left(1 + \frac{\hat{\sigma}^2_{\xi,k}}{\hat{\sigma}^2_{\xi,k} + s^2_{\tilde{\pi}_k}} \right) \right\rfloor, N_0 + k \right), k > 1. \tag{18}$$

In applying (18), the problem of how to estimate the variance components $\hat{\sigma}^2_{\xi,k}$ and $s^2_{\tilde{\pi}_k}$ arises. In general, the intent of equation (18) is twofold: on one hand, we should have enough budget to guarantee evaluation accuracy to support the generation of the next promising point, on the other hand, we do not wish to use up too much simulation budget since this would hinder exploration of the solution space. In this manuscript, we propose four alternative solutions that heuristically attempt to respond to these needs, and we present these variants in the following.

OCBA-driven Budget Resource Allocation (eTSSO$_{\mathbf{O}}$). This policy sets the budget sequence to grow according to the following:

$$\begin{aligned} \hat{\sigma}^2_{\xi,k} &:= \quad \hat{\sigma}^2_{\xi,k}\left(\boldsymbol{x}_k^{\mathrm{OCBA}}\right), \\ s^2_{\tilde{\pi}_k} &:= \quad s^2_{\tilde{\pi}_k}\left(\boldsymbol{x}_k^{\mathrm{OCBA}}\right), \\ \text{where} \quad & \boldsymbol{x}_k^{\mathrm{OCBA}} \in \arg\max_{\boldsymbol{x} \in \mathbb{S}_k} n_{\boldsymbol{x}}. \end{aligned}$$

The basic idea is that if the intrinsic variance is predominant, then the budget per replication should increase at a faster rate to reduce the effect of noise and obtain more reliable estimates of the objective

function values. If the point with largest OCBA-allocated budget at the k-th iteration, $\boldsymbol{x}_k^{\text{OCBA}}$, has large intrinsic variance, $\hat{\sigma}^2_{\xi,k}$, we would increase the budget assigned to the iteration in order to drive down the noise. On the other hand, when extrinsic variance, which measures the prediction uncertainty of the MNEK metamodel, is large, it means the solution space has not been sufficiently explored and thus the increase in the number of replications per iteration should slow down to allow more iterations. The approach still focuses on the noisiest point (as selected by OCBA), but it balances it off with the extrinsic variance at that point, moderating the exploitative aspect of the policy.

Average Budget Allocation Rule (eTSSO$_{\text{A}}$). While the previous policy only looks into sampled points and uses the extrinsic variance as a proxy for the out-of-sample uncertainty, several allocations can be proposed that use the sampled points to estimate the intrinsic variance and only unsampled points to produce an estimate of the "relevant" extrinsic variance. The policy eTSSO$_{\text{A}}$ derives such information through averaging, formally:

$$\hat{\sigma}^2_{\xi,k} := \frac{1}{|\mathbb{S}_k|} \sum_{\boldsymbol{x}\in\mathbb{S}_k} \hat{\sigma}^2_{\xi,k}(\boldsymbol{x})$$
$$s^2_{\tilde{\pi}_k} = \frac{1}{|\mathbb{X}\setminus\mathbb{S}_k|} \sum_{\boldsymbol{x}\in\mathbb{X}\setminus\mathbb{S}_k} s^2_{\tilde{\pi}_k}(\boldsymbol{x})$$

We use the average of the intrinsic variance over the sampled points against the average of the extrinsic variance over the un-sampled points. In order to compute the second element, we use a grid over the solution space (a continuous version would anyway require numerical approximation). For the case of problems in high (e.g., $d \geq 10$) dimensions, we recommend to proceed with a Latin Hypercube Design instead of a grid. This choice, given a number N of points, would guarantee some form of coverage while controlling the number of sampled points.

Goal-driven Budget Resource Allocation (eTSSO$_{\text{G}}$). While eTSSO$_{\text{A}}$ does not require to choose any location within the un-sampled set, it could be argued that "interesting points" could be selected to bias sampling allocation in favor of the ultimate goal of eTSSO: finding the global minimum. The policy eTSSO$_{\text{G}}$ selects as reference points the locations with the best function value so far, $\boldsymbol{x}_k^* \in \arg\min_{\boldsymbol{x}\in\mathbb{S}_k} \bar{y}(\boldsymbol{x})$, and the location with associated maximum expected improvement $\boldsymbol{x}_k^{\text{EI}} \in \arg\max_{\boldsymbol{x}\notin\mathbb{S}_k} T_{\tilde{\pi}_k}(\boldsymbol{x})$. Formally:

$$\hat{\sigma}^2_{\xi,k} = \hat{\sigma}^2_{\xi,k}(\boldsymbol{x}_k^*),$$
$$s^2_{\tilde{\pi}_k} = s^2_{\tilde{\pi}_k}(\boldsymbol{x}_k^{\text{EI}}).$$

The idea here is to compare the noise currently associated to our best guess, against the strongest candidate (un-sampled) point. This criterion is more goal driven in the sense that it balances the need to produce an accurate estimation for the current best point, with the goal to explore potentially better solutions. When large/promising regions are unexplored (which is the case at the start of the procedure) we expect this criterion to boost search.

Eager Budget Resource Allocation (eTSSO$_{\mathbf{E}}$). One potential issue of all previous allocation policies is that, even if they are likely to guarantee exploration, at least at the beginning, they may favor exploitation quite early in the search. We propose, in this direction, to use as reference points the location with the lowest intrinsic variance, and the point with maximum extrinsic variance. Formally:

$$\hat{\sigma}^2_{\xi,k} = \min_{\boldsymbol{x}\in\mathbb{S}_k} \hat{\sigma}^2_{\xi,k},$$
$$s^2_{\tilde{\pi}_k} = \max_{\boldsymbol{x}\notin\mathbb{S}_k} s^2_{\tilde{\pi}_k}.$$

At the early stages of the search procedure, the extrinsic variance will likely be larger, hence, especially in case of low intrinsic noise, the allocation rule will not provide large values of budget increase. This avoids allocating a significant evaluation effort to the first algorithm iterations, thus favoring the search by increasing the number of potential points to sample. When the extrinsic variance reduces, the budget for the evaluation increases. This is desirable since, in such a situation, we are likely to have identified the optimum region. Hence, we will be willing to spend more budget in order to correctly identify the best point among the sampled ones.

It is important to highlight that, in case of small budget T, and even more, with particularly large noise levels, increasing the budget might lead to poor performance due to the early termination of the algorithm. In such circumstances, we may expect TSSO to perform better than the eTSSO algorithm. However, given a small computer budget and high intrinsic or extrinsic noise, any algorithm performs poorly. eTSSO is summarized in Algorithm 1. Concerning the cross-validation step in Algorithm 1, we perform Leave One Out Cross Validation (LOOCV) (Vehtari et al., 2017; Efron and Tibshirani, 1997) with a threshold α. Specifically, for each point $\mathbf{x}_i, i = 1, \ldots, N_0$, we estimate a Gaussian Process model leaving out the i-th observation, i.e., $\mathbf{x}_i$. This produces two quantities for the cross-validation test: $\hat{y}(\mathbf{x}_i)$, and $\mathrm{MSE}_{\tilde{\pi}_0}(\mathbf{x}_i)$ (using equations (9) and (10), respectively). If $|\hat{y}(\mathbf{x}_i) - y(\mathbf{x}_i)|/\mathrm{MSE}_{\tilde{\pi}_0}(\mathbf{x}_i) > \alpha$, the cross-validation fails, where α is an input parameter provided by the user. In fact, if the threshold is violated for at least one location within the set of N_0 initial sampling points, we re-sample the initial set and increase the $r_{\min}$ of a $\delta_{r_{\min}}$ amount until the cross-validation is passed. In practice, for the tests in section 5.2, we ran the initial sample and minimum budget selection procedure offline, creating the common design for all the experiments related to each function.

Algorithm 1: eTSSO Algorithm

Initialization:
Define $T, N_0, \alpha, r_{\min}, \delta_{r_{\min}}, \delta_{N_0}, k \leftarrow 0$;
Intial Model Fit (input: $r_{\min}, N_0, \alpha, \delta_{r_{\min}}, \delta_{N_0}$**):**
Passed $\leftarrow$ False;
while *Passed==False* **do**
  Generate the initial sample set $\{\boldsymbol{x}_i\}_{i=1}^{N_0}$ with an LHS design of N_0 points;
  Simulate each location in $\{\boldsymbol{x}_i\}_{i=1}^{N_0}$ with $r_{\min}$ replications;
  Fit the MNEK model to the set of sample means;
  Apply LOOCV cross-validation with threshold α to evaluate the quality of the model;
  if *LOOCV Fail* **then**
    Set $r_{\min} \leftarrow r_{\min} + \delta_{r_{\min}}$, $N_0 \leftarrow N_0 + \delta_{N_0}$;
    Go to Step 6;
  end
  else
    Passed $\leftarrow$ True;
    $\mathbb{S}_0 \leftarrow \{\boldsymbol{x}_i\}_i^{N_0} = 1$
  end
end
Set the initial available budget $B_0 = r_{\min}$ and collect the initial data;
$T \leftarrow T - B_0 N_0$, $A = T$, $k \leftarrow 1$.
while $A > 0$ **do**
  Search:
  if $A > r_{min}$ **then**
    Find the point $\boldsymbol{x}_k \in \arg\max_{\boldsymbol{x} \in \mathbb{X} \notin \mathbb{S}_k} T_{\tilde{\pi}_k}(\boldsymbol{x})$, $\mathbb{S}_k \leftarrow \mathbb{S}_k \cup \boldsymbol{x}_k$;
    Run $r_{\min}$ replications to evaluate the function in $\boldsymbol{x}_k$;
    $A \leftarrow A - r_{\min}$;
  end
  if $k > 1$ **then**
    Budget Computation:
    Calculate B_k using equation (18) and the allocation rule of choice;
    Evaluation Stage:
    if $A > B_k$ **then**
      Assign one observation to each point in $\mathbb{S}_k$;
      Use equations (14)-(15) to allocate $B_k - |\mathbb{S}_k|$ simulations to the sampled points;
      Update $(\boldsymbol{y}_k(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{S}_k)$, and fit the kriging model π_k according to the updated information;
      $A \leftarrow A - B_k$;
    end
    else
      Use equations (14)-(15) to allocate A_k simulation to the sampled points;
      Update $(\boldsymbol{y}_k(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{S}_k)$ and fit the kriging model π_k according to the updated information;
      $A \leftarrow 0$;
    end
  end
  $k \leftarrow k + 1$;
end
Return the location with the maximum $\boldsymbol{x}_K^* \in \arg\max_{\boldsymbol{x} \in \mathbb{S}_K} \bar{y}(\boldsymbol{x})$, with K being the final iteration.

4 eTSSO Asymptotic behavior

In this section, we present the asymptotic analysis of Algorithm 1, both in terms of asymptotic convergence as well as convergence rates. The asymptotic convergence and the convergence rate of eTSSO are investigated by interpreting the kriging-based search as a stochastic recursion. We show the parallelism between the two paradigms and exploit the deterministic counterpart of eTSSO, the widely known Efficient Global Optimization (EGO) procedure (section 2.1) to perform our study. In fact, results on convergence and convergence rates for EGO have been proposed in Locatelli (1997) and Bull (2011), respectively. The basic idea is to analyze eTSSO as the stochastic counterpart of EGO. This idea allows to use some of the results in Pasupathy et al. (2018), while dealing with the difficulty of having a simulation budget that is a stochastic sequence. The proof scheme we propose is articulated into three parts:

- Part 1: the deterministic analogue of eTSSO (EGO) is analyzed in terms of convergence properties to ensure the reference algorithm of eTSSO has a good behavior. Lemmas 1-2 serve this purpose by guaranteeing Lipschitz continuity of the EGO recursion and its convergence to the global optimum,

respectively;

- Part 2: the stochastic algorithm eTSSO is characterized in terms of (a) boundedness of the budget for any finite number of iterations, and (b) convergence of the stochastic MNEK model to the deterministic counterpart in the case of dense infinite sampling. Lemmas 3-4 report these results, respectively.

- Part 3: the asymptotic convergence of the stochastic recursion to the deterministic recursion is characterized in Theorem 1 and Theorem 2. Finally, the convergence rate is characterized in Theorem 3.

Section 4.1 provides the main definitions we will adopt for the proofs in section 4.2.

4.1 Notation and Terminology

Let $\{\mathbf{X}_n\} \xrightarrow{wp1} x$ represent a stochastic sequence of random variables, $\{\mathbf{X}_n\}$, that converges to $\boldsymbol{x}$ with probability 1. Furthermore, we will refer to the expectation of a random variable V computed at iteration k as $E_k V$. Given $\{a_n\}$ sequence of real numbers, then $a_n = o(1)$ if $lim_{n\to\infty} a_n = 0$, and $a_n = O(1)$ if $\exists c \in (0, \infty)$ with $|a_n| < c$ for large enough n; also $a_n = \Theta(1)$ if $0 < \liminf a_n \leq \limsup a_n < \infty$. In the analysis that follows, two convergence definitions will be adopted:

Definition 1 (Linear convergence)**.** $\{\boldsymbol{x}_k\}$ *exhibits a* linear(ℓ) *convergence to* $\boldsymbol{x}^*$ *if* $\limsup_{k\to\infty} \frac{||\boldsymbol{x}_{k+1} - \boldsymbol{x}^*||}{||\boldsymbol{x}_k - \boldsymbol{x}^*||} = \ell \in (0, 1)$

The following definition characterizes the control of the sample size sequence we created for the stochastic algorithm eTSSO.

Definition 2 (Geometric growth of a sequence)**.** *A sequence* $\{m_k\}$ *exhibits geometric growth if* $m_{k+1} = c \cdot m_k,\ k = 1, 2, \ldots$ *for some* $c \in (1, \infty)$.

4.2 Main Results

In this proof, we will adopt the recursive algorithm setting to discuss eTSSO behavior referring to the theoretical budget allocation in equation (18). A first justification for the proposed approach can be found in the literature in advanced random search (Mete et al., 2011) where the link between recursion and sampling from target distributions is established. In fact, recursive iterations are studied in terms of resulting probability of sampling in the feasible region. This creates a link between the meta-modeling environment where we start assuming a distribution of the response, against recursive methods which iteratively and implicitly construct this distribution by means, for example, of gradient information.
In traditional recursion algorithms, at the k-th iteration, the next point in the search procedure satisfies (Pasupathy et al., 2018):

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h(\boldsymbol{x}_k) \tag{19}$$

In different recursions, $h(\boldsymbol{x}_k)$ can be interpreted as the product of a step or the inverse Hessian and the function gradient estimated at the current location. Under a geometric interpretation, the right hand side of the update step represents the (vector) of the linear distance(s) between the current point and the

next point which will be sampled. The basic idea of our approach is to interpret the algorithm Efficient Global Optimization (EGO) (Jones et al., 1998) as a stochastic recursion. According to this view, at the k-th iteration, we can modify the definition provided in equation (7) obtaining

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k^* + \text{dist}\left(\boldsymbol{x}_k^*, \arg\max_{\boldsymbol{x}\in\mathbb{X}\setminus\mathbb{S}_k} T_{\pi_k}(\boldsymbol{x})\right). \tag{20}$$

Where $\text{dist}(\cdot,\cdot)$ is the distance vector having as components $\left\{\boldsymbol{x}_{h,k}^* - \boldsymbol{x}_{h,k+1}\right\}_{h=1}^{d}$, and d is the dimensionality of the solution space.We have:

- $\boldsymbol{x}_{k+1}$ represents the next point to sample;
- $\boldsymbol{x}_k \leftarrow g\left(\{\boldsymbol{x}_i\}_{i=1}^k\right) \triangleq \boldsymbol{x}_k^* \in \arg\min_{x\in\mathbb{S}_k} y(x)$, i.e., the current solution is, at each step, the point with the lowest sampled value. In fact, if the search stops at iteration k, a traditional recursion would return $\boldsymbol{x}_k^*$ as the best solution so far. To the same extent, EGO (and eTSSO) would return $\arg\min_{\boldsymbol{x}\in\mathbb{S}_k} y(\boldsymbol{x})$. This means that, differently from the original recursion, $\boldsymbol{x}_k$ does not only depend on the previous sampling decision, but on the sequence of sampled solutions. This justifies the notation $\{\boldsymbol{x}_i\}_{i=1}^k$. This difference between the original approach and the meta-model based search will require us an additional consistency result with respect to the framework in (Pasupathy et al., 2018), which we provide in Property 1.
- $h(\boldsymbol{x}) \leftarrow h(\{\boldsymbol{x}_i\}_{i=1}^k) \triangleq \text{dist}\left(\boldsymbol{x}_k^*, \arg\max_{\boldsymbol{x}\in\mathbb{X}\setminus\mathbb{S}_k} T_\pi(\boldsymbol{x})\right)$. In this case, we observe the parallelism between the two algorithms under the aforementioned geometrical interpretation: the right hand side of the update step represents the (vector) of the linear distance(s) between the current point and the next point which will be sampled.

In the stochastic counterpart, the traditional recursion can be modeled as (Pasupathy et al., 2018)

$$\mathbf{X}_{k+1} = \mathbf{X}_k + H(W_k, \mathbf{X}_k).$$

Where W_k is the number of simulations that have been ran up to iteration k. Similarly to the approach in (20), we can formulate the "meta-model" version of the iteration as

$$\mathbf{X}_{k+1} = \mathbf{X}_k^* + \text{dist}\left(\mathbf{X}_k^*, \arg\max_{\boldsymbol{x}\in\mathbb{X}\setminus\mathbb{S}_k} T_{\tilde{\pi}_k}(\boldsymbol{x})\right). \tag{21}$$

Where, $\tilde{\pi}_k$ refers to the model in (8) replacing π in equation (2).
$\mathbf{X}_k \leftarrow \mathbf{X}_k^* \triangleq G\left(W_k, \{\mathbf{X}_i\}_{i=1}^k\right)$ equivalently $X_k^* \in \arg\min_{\boldsymbol{x}\in\mathbb{S}_k} \bar{Y}(\boldsymbol{x})$, whereas

$$H(W_k, \mathbf{X}_k) \leftarrow H\left(W_k, \{\mathbf{X}_i\}_{i=1}^k\right) \triangleq \text{dist}\left(\mathbf{X}_k^*, \arg\max_{\boldsymbol{x}\in\mathbb{X}\setminus\mathbb{S}_k} T_{\tilde{\pi}_k}(x)\right).$$

W_k represents the total simulation budget used up to iteration k, i.e., $\sum_k B_k$ and according to (18), is a random variable. Due to the budget stochasticity we need to guarantee further results with respect to (Pasupathy et al., 2018) that deals with a deterministic number of simulation runs.
Reminding that $\mathbb{S}$ represents the set of sampled points, we can define $\mathbb{S}_k$ as the set of point sampled up

to iteration k by the EGO algorithm and $\tilde{\mathbb{S}}_k$ the corresponding set generated by the stochastic analogue (eTSSO). As a result, we can formulate $G\ (g)$ and $H\ (h)$ as $G\left(W_k, \tilde{\mathbb{S}}_k\right)\ (g\left(\mathbb{S}_k\right))$ and $H\left(W_k, \tilde{\mathbb{S}}_k\right)\ (h\left(\mathbb{S}_k\right))$, respectively.

This new interpretation of surrogate based optimization is relevant since it allows us to understand the behavior of a complex stochastic algorithm based on its counterpart. We will use (Pasupathy et al., 2018) that established foundational results in the analysis of recursions. Specifically, we will treat our algorithm as the stochastic counterpart of the well known EGO (Jones et al., 1998). In the following, we list the assumptions at the basis of the asymptotic analysis.

Assumption 1. *Denote $N_k(\boldsymbol{x})$ as the total number of replications at design point $\boldsymbol{x}$ by iteration k and $\sigma_0^2 \triangleq \max_{\boldsymbol{x}\in\mathbb{X}} \sigma_\xi^2(\boldsymbol{x})$. There exist a sequence $\{r_1, ..., r_k ...\}$ such that $r_{k+1} \geq r_k$, $r_k \to \infty$ as $k \to \infty$ and that $\sum_{k=1}^{\infty} k \exp(-\kappa r_k) < \infty$, $\forall \kappa > 0$. The allocation rule ensures that $N_k(\boldsymbol{x}) \geq r_k$, for all design point $\boldsymbol{x}$.*

Assumption 1, is required to guarantee that OCBA does not impact negatively the convergence of the algorithm. In fact, the OCBA technique was originally developed for optimization problems with finite number of alternatives. To satisfy this assumption, in this work, at each evaluation stage, we first spare some budget to ensure that all design points receive at least r_k replications, with $\{r_k = k\}$. This can be easily obtained from our budget B_k by assigning a single observation first to the sampled points, subsequently using OCBA to allocate the remaining budget (refer to step 34 in Algorithm 1).

Assumption 2. *The number of replications B_k assigned at each iteration satisfies $B_k \geq B_{k-1}$, $\forall k = 1, 2, \ldots$ and $B_k \to \infty$ as $k \to \infty$. Moreover, for any $\epsilon > 0$ there exists a $\delta_\epsilon \in (0, 1)$ and a $\bar{k}_\epsilon > 0$ such that $\psi^{2k}\mathcal{L}\left(B_{k-1}, \epsilon\right) \leq \left(\delta_\epsilon\right)^k$, $\forall k \geq \bar{k}_\epsilon$, where $\mathcal{L}\left(\cdot, \cdot\right)$ is strictly decreasing in B_{k-1} and non–increasing in ϵ.*

Assumptions 1-2 are concerned with bounding the behavior of the stochastic sequence of budgets. While Assumption 1 looks at each single point budget allocation and serves the purpose to characterize the convergence of the search, Assumption 2 looks at the overall budget per iteration and serves the purpose to study the convergence of the surrogate model.

Assumption 3. *$\mathbb{X}$ is a compact space.*

Assumption 4. *Each dimension in the space is defined between $[0, 1]$.*

This scaling operation is frequently operated in the surrogate model literature (Picheny et al., 2013; Kleijnen, 2008). While it does not lead to any loss in generality, such an assumption allows to easily derive our bounding argument for the EGO expected improvement function used in Lemma 2.

Assumption 5. *The Gaussian correlation function is adopted to model the spatial variance-covariance matrix.*

Assumption 5 is a sufficient condition for the existence of the derivative processes and it ensures that the various variance-covariance matrices are positive definite, i.e., non-singular (Ankenman et al., 2010). These will be used in Lemma 1, which characterizes the expected improvement function in (7).

Assumption 6. *The parameters τ, ϕ and σ_ξ^2 of the MNEK model are assumed known.*

Assumption 7. *The initial sample* $\{\boldsymbol{x}_i\}_{i=1}^{N_0}$*, and minimum number of replications* r_{min} *are such to produce an initial fit of the MNEK model* π_k*,* $k = 0$*, satisfying cross-validation criteria.*

Assumption 7 is important in achieving uniform convergence. Our cross-validation procedure (algorithm 1, Steps 6-19) allows to generate such initial conditions.

Assumption 8. *The true function to be optimized over the compact space* $\mathbb{X}$ *is bounded and has a unique global minimum* $\boldsymbol{x}^*$*.*

We start characterizing the "distance" function $h(\boldsymbol{x})$ defined in equations (19)-(20).

Lemma 1. *There exists* $\kappa \in \mathbb{R}$ *such that, for any* $(\mathbb{S}, \mathbb{S}') : \mathbb{S} \subset \mathbb{X}, \mathbb{S}' \subset \mathbb{X}$*,* $||h(\mathbb{S}) - h(\mathbb{S}')|| \leq \kappa D(\mathbb{S}, \mathbb{S}')$*, where* $D(\cdot)$ *represents the distance between two sets of points.*

Proof. Proof in the Appendix. □

Lemma 1 characterizes the behavior of the component $h(\cdot)$ of the recursion in equation (20). In particular, it guarantees that similar sets of sampled points return similar values for the recursion, where similarity is characterized by the distance between two sets.
The following Lemma states that EGO produces iterates which are converging to the global optimum and the result relies on the study in (Locatelli, 1997).

Lemma 2. *Let us consider a Gaussian correlation function, then* $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}^*$ *will hold for the EGO algorithm. Moreover, if Assumption 7 is satisfied, the result will hold for any initial sampling* $\{\mathbf{x}_{k_0}\}_{k_0=1}^{N_0}$*, i.e., we have uniform convergence.*

Proof. Proof in the Appendix. □

Firstly, we need to characterize stochastic sequence $W_k = \sum_{i=1}^{k} B_i$, where the simulation budget B_i is generated by equation (18). We can observe what follows:

Lemma 3. *The cumulated budget* W_k *satisfies* $W_k \xrightarrow{k\to\infty} \infty$ *w.p.1, and the expected budget at iteration* k *is finite for any finite value of* k*.*

Proof. Proof in the Appendix. □

Lemma 4. *As the number of iterations* $k \to \infty$*, under assumptions 2-8, the MNEK model* $\tilde{\pi}_k$ *approaches its deterministic counterpart* π_k*.*

Proof. Proof in the Appendix. □

Lemma 5 characterizes the distribution of the predictor produced by the MNEK for $k \to \infty$.

Lemma 5. *As the number of iterations* $k \to \infty$*, the stochastic predictor* $\hat{Y}$ *resulting from the MNEK model* $\tilde{\pi}_k$ *becomes* $\hat{Y}(\boldsymbol{x}|\mathcal{F}_{\tilde{\pi}_k}) \sim N\left(\mu_{\tilde{\pi}_k}(x), s^2_{\tilde{\pi}_k}(\boldsymbol{x})\right)$*, where* $\mu_{\tilde{\pi}_k}(\boldsymbol{x}), s^2_{\tilde{\pi}_k}(\boldsymbol{x})$ *correspond to the moments for the deterministic-response Gaussian model* π_k*.*

Proof. Asymptotically ($k \to \infty$) Lemma 4 holds. Then, according to the result in Stein (1999) (Appendix A), we have that $\hat{Y}\left(\boldsymbol{x}|\mathcal{F}_{\tilde{\pi}_k}\right)$ is normally distributed and parameterized by:

$$\mu_{\tilde{\pi}_k}\left(W_k, \boldsymbol{x}, \phi\right) \xrightarrow{k\to\infty} \left(\mathbf{c}^T\mathbf{R}^{-1} + \mathbf{1}^T\mathbf{R}^{-1}\frac{\left[1 - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{c}\right]^T}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right)\bar{\mathbf{y}} \tag{22}$$

$$s^2_{\tilde{\pi}_k}\left(W_k, \boldsymbol{x}, \phi\right) \xrightarrow{k\to\infty} \tau^2\left(1 - \left[\mathbf{c} + \mathbf{1}\frac{\left(\mathbf{1} - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{c}\right)}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right]^T\mathbf{R}^{-1}\mathbf{c} + \frac{\left(\mathbf{1} - \mathbf{1}^T\mathbf{R}^{-1}\mathbf{c}\right)}{\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}}\right). \tag{23}$$

These correspond to the predictors obtained in Yin et al. (2011) for deterministic responses. □

Now we analyze the effect of the results in the previous Lemmas onto $G\left(\cdot,\cdot\right)$ as well as $H\left(\cdot,\cdot\right)$.

Theorem 1 (Convergence of $G\left(W_k, \tilde{\mathbb{S}}_k\right)$)**.** *For any $\delta > 0$, and with $E_kW \to \infty$, when $k \to \infty$,* $\sup_{\mathbb{S}_k, \tilde{\mathbb{S}}_k \subseteq \mathbb{X}} Pr\left\{||G(E_kW, \tilde{\mathbb{S}_k}) - g(\mathbb{S}_k)|| > \delta\right\} = O\left(\left[E_kW\left(\mathbf{X}^*_k\right)\right]^{-1/2}\right)$.

Proof. Proof in the Appendix. □

The study of $H(W_k, \tilde{\mathbb{S}}_k)$ is the key to analyze the efficiency and consistency of eTSSO. The first step in this direction is to establish the relationship between $H(\cdot)$ and $h(\cdot)$.

Theorem 2 (Convergence of $H\left(W_k, \tilde{\mathbb{S}}_k\right)$)**.** *(i) Let $k \to \infty$, and let $\mathbb{S}_k$ represent any possible subset of feasible points of size k. Then the estimator $H\left(W_k, \tilde{\mathbb{S}}_k\right)$ satisfies, for any $\Delta > 0$,*

$$\sup_{\mathbb{S}_k \subset \mathbb{X}} Pr\left\{||H\left(W_k, \mathbb{S}_k\right) - h\left(\mathbb{S}_k\right)|| > \Delta\right\} = O\left(\left(E_kW\right)^{-2\alpha}\right).$$

(ii) If the sequence of simulation budgets $\{W_{i,k}\}$ satisfy $W_{i,k} \to \infty$ a.s., then $||H\left(W_k, \mathbb{S}_k\right) - h\left(\mathbb{S}_k\right)|| \xrightarrow{wp1} 0$.

Proof. The result in (i) can be obtained from Lemma 4 by setting $\alpha = \frac{1}{2}$ and $W = \max_{\mathbf{x}_j \in \mathbb{S}} W_{j,k}$. In fact, Lemma 3 proved that the budget goes to infinity a.s., and it has finite expectation for finite k, and Lemma 4 proves that the stochastic model converges to the deterministic model.

Concerning part (ii), Lemma 3 guarantees the budget to reach infinity when the iterations satisfy $k \to \infty$. Again, we consider Lemma 4 that proves the stochastic model converges to the deterministic model. This means that the sequence of points generated by the expected improvement function will converge. Considering Theorem 1 and Lemma 1, part (ii) is also guaranteed. □

The results in Theorems 1-2 are at the basis for the efficiency analysis of the proposed algorithm. In this phase, we make use of the results presented in (Pasupathy et al., 2018), as it will be specified in the following.

Property 1 (Characterization of $H\left(W_k, \mathbb{S}_k\right)$)**.** *Let $k \to \infty$, then the estimator $H\left(W_k, \mathbb{S}_k\right)$ satisfies* $\sup_{\mathbb{S}_k \subset \mathbb{X}} E\left(H\left(W_k, \mathbb{S}_k\right) - h\left(\mathbb{S}_k\right)\right) = \Theta\left(\left(\tau^2 \cdot E_k\left(\min_{\mathbf{x}\in\mathbb{S}_k} W_k(\mathbf{x})\right)\right)^{-1}\right)$.

Proof. The main ingredient to prove the theorem and, main challenge, is the analysis of the behavior at convergence of the sequence of expected improvements $T_{\tilde{\pi}_k}$ generated by our algorithm. Under Assumption 3 and Assumption 7, $T_{\tilde{\pi}_k}$ is finite, and, from Lemma 1, we know that the function is differentiable and

Lipschitz Continuous. Under these premises, the expected improvement function satisfies the assumptions of epi-convergence (Attouch, 1984), i.e., $T_{\tilde{\pi}_k} \xrightarrow[k\to\infty]{\text{epi}} T_{\pi_k}, w.p.1$, if $W_k(\boldsymbol{x}) \to \infty$. While our budget allocation is stochastic in nature, we showed in Theorem 1 that, under Assumptions 1 and Assumption 2, $W_k(\boldsymbol{x}) \xrightarrow[k\to\infty]{} \infty, w.p.1$

Given the epi-convergence is valid, Theorem 3.4 in (Robinson, 1996) also applies, so that the sequence $T_{\tilde{\pi}_k} \xrightarrow{\text{epi}} T_{\pi_k}$ and the sequence of selected points $\mathbf{X}_k \in \arg\max T_{\tilde{\pi}_k} \xrightarrow{\text{epi}} \boldsymbol{x}_k$. □

Theorem 3 (Convergence rate of Algorithm 1)**.** *Let us define* $C_k := 1 + \frac{\hat{\sigma}^2_{\xi,k}}{\hat{\sigma}^2_{\xi,k} + s^2_{\tilde{\pi}_k}(\boldsymbol{x}_{k+1})}$ *and* $\ell = \left(1 - \frac{1}{k}\right)^{1/d}$. *Given that EGO exhibits linear convergence (Bull, 2011), for any* $\varepsilon \geq 0$ *satisfying* $\ell + \varepsilon < 1$ *and as* $k \to \infty$, *the following holds for* $\mathbb{E}_k = E\left[||\boldsymbol{X}^*_{k+1} - \boldsymbol{x}^*||\right]$*:*

$$E[C_k] \geq \ell^{-2}, \quad \mathbb{E}_k = O\left(\left(E[C_k]^{-1/2}\left(\ell + \varepsilon\right)^{-1}\right)^{-k} E_k W_{k+1}^{-1/2}\right) \tag{24}$$

Proof. Proof in the Appendix. □

The proofs have been developed assuming known parameters. For the deterministic case, convergence rates are discussed in (Bull, 2011), where boundedness of the Maximum Likelihood estimation is required. For the stochastic case, the role played by the bias was empirically discussed in (Kleijnen et al., 2012) where the authors recognize that the consistency of the bias plays a major role. In fact, as long as the bias is consistent, the optimal location is identified according to the empirical evidence.

5 Empirical Results

While results for the empirical convergence rate were provided in the conference paper (Pedrielli and Ng, 2015), herein we focus on the impact of the budget and the finite time performance of the algorithm in its four variants.

Section 5.1 shows the impact of the adaptive budget allocation over relatively simple test functions with the aim to show the negative effect that a wrong choice of the budget can have over the TSSO algorithm and how eTSSO tackles this challenge. In this part of the analysis, we show the results for the variant eTSSO_O (similar results were obtained running the other variants of the allocation). Subsequently, section 5.2 focuses on the performance of the proposed algorithm over increasingly complex functions when the choice of the budget for TSSO is performed according to the the recommendations from the analysis in section 5.1. In this part of the analysis, all the variants of eTSSO are studied in order to try to provide insights on the most promising family of allocation rules.

5.1 Impact of the Budget Allocation Rule for TSSO and eTSSO

To quantify the impact of the choice of B, we propose to study the following 1-d function represented in Figure 1:

$$Y(x) = (2x + 9.96)\cos(13x - 0.26) \tag{25}$$

This function has a global minimum in $\boldsymbol{x}^* = 0.746$ with function value $y^* = y(\boldsymbol{x}^*) = -11.45$ and a local minimum in $x = 0.2628$, with $\mathbb{X} = [0, 1]$. As noise, we applied to the function an additive Gaussian Process $\xi(x)$ with mean $\mathbf{0}$ and diagonal variance covariance matrix with elements:

$$\sigma_\xi^2(x) = \delta \cdot x \tag{26}$$

Where δ represents the magnitude of the noise.

We set the total budget $T = 300$ ($T = 3000$ for the high noise case), the minimum number of replications to sample a new point to $r_{\min} = 10$ and the number of initial sampling points to $N_0 = 6$.
As a result of the previous settings, the minimum budget per iteration $B_{\min}$, obtained applying equation (16), is $B_{\min} = 19$ ($B_{\min} = 57$ for the high noise case). $B_{\max}$, i.e., the budget such that all the available replications are used for the evaluation of the initial design, is $B_{\max} = 50$ ($B_{\max} = 500$ for the high noise case). Finally the minimum "feasible" budget is $r_{\min} = 10$.

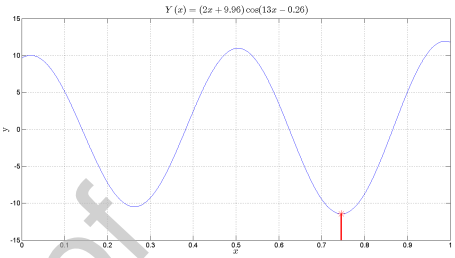
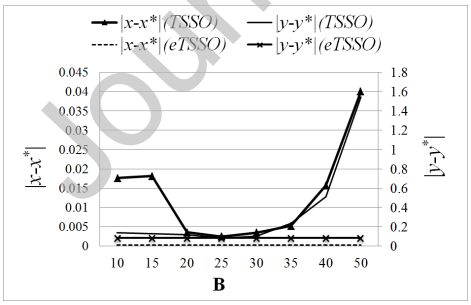
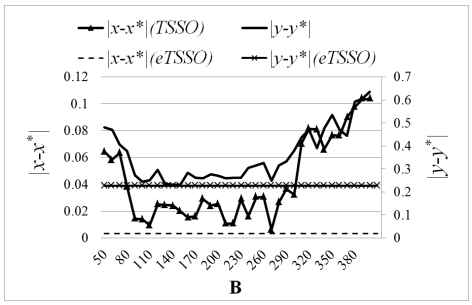


Figure 1: 1–d Function representation

Figure 2(a) and Figure 2(b), show the performance of the original TSSO algorithm in terms of optimum location and optimal function value estimation, for values of the budget $r_{\min} \leq B \leq B_{\max}$ under low noise, $\delta = 0.1$ and large noise, $\delta = 10.0$, respectively. In the figures, each point represents the average performance obtained from 100 macro-replications of the algorithm. The *location performance* is meant to be the Euclidean distance between the point associated with the minimum predicted value x and the true global optimum of the function x^* and it is referred to as $|x - x^*|$, the *estimation performance* refers to the absolute difference between the best performance according to the final prediction y and the true optimal value y^* and it is referred to as $|y - y^*|$.



(a) $\delta = 1.0$ 1-d case



(b) $\delta = 10.0$ 1-d case

Figure 2: Effect of the budget per iteration, 1-d case for TSSO and the effect of the adaptive budget allocation (O-BAR allocation policy was used).

The performance of TSSO in terms of location $|x - x^*|$ as well as $|y - y^*|$ are non-monotone in the assigned B. Specifically, as B increases, the noise effect is mitigated as more replications are performed at each sampled point. On the other hand, the number of iterations that can be performed decreases since the total budget is fixed (equation (17)). The first effect leads to a potential decrease in $|y - y^*|$,

especially when large noise is considered (Figure 2(b)). However, the second effect can be critical, resulting in an increase of $|x - x^*|$ and a consequent increase of $|y - y^*|$ as observed in the right extremes of both Figure 2(a) and 2(b). As a result, we observe that TSSO can be less effective in cases where either *large* or *low* values of B are chosen by the user.

Figure 2(a) shows that the error in the location (see left-hand vertical axis) resulting from the new algorithm, is far below the average error obtained when TSSO is applied over all B's.

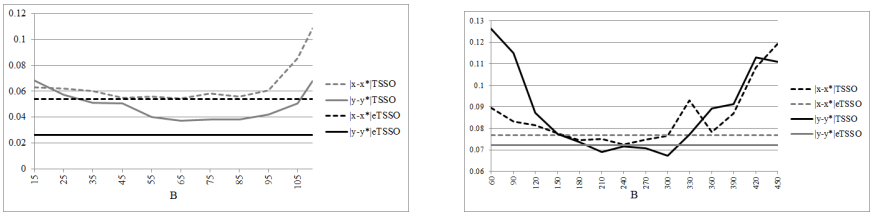
It is important to consider that TSSO might lead to better performance under specific values of B, given the total budget T. However, in practice, since no structural properties are defined, running the algorithm is the only way to determine a suitable value for B. Focusing on the $|y - y^*|$ performance (see right-hand vertical axis), we observe an expected good result from eTSSO_{O}. In this specific case, the extended algorithm is always better than the original TSSO. Since eTSSO explicitly considers the response noise, by increasing the budget when this is particularly large, we can expect a better performance in terms of function value estimation, especially with large noise levels. This aspect is important as it reflects in the location performance $|\boldsymbol{x} - \boldsymbol{x}^*|$. Indeed, as the algorithm progresses, convergence to the optimal location is guaranteed only if the function value is correctly estimated (Vogel and Lachout, 2003).

Figure 2(b) further investigates the effect of the noise. In particular, it shows the results from the same experimental settings used in Figure 2(a), with noise level $\delta = 10.0$ and total budget $T = 3000$ because of the increased noise. Despite a decrease in the algorithm performance overall (both TSSO and eTSSO), due to the increased noise level, it is possible to observe a similar behavior as in the lower noise case.

We also studied the 2-d tetra-modal function:

$$Y(x_1, x_2) = -5(1 - (2x_1 - 1)^2)(1 - (2x_2 - 1)^2)(4 + 2x_1 - 1)\left(0.05^{(2x_1-1)^2} - 0.05^{(2x_2-1)^2}\right)^2. \quad (27)$$

Where the dimensions of the test function, x_1 and x_2, are scaled to $[0, 1]$. The global minimum is located at $[0.85, 0.5]$ and has the response value -7.098. As noise, we applied to the function an additive Gaussian Process $\xi(x)$ with mean $\mathbf{0}$ and diagonal variance covariance matrix with diagonal elements $\sigma_\xi^2(x_1, x_2) = \delta \cdot (x_1 + x_2)$. We set $N_0 = 20$. The first experiment set was performed with $T = 2400, r_{\min} = 15, \delta = 1.0$ resulting in $B_{\min} = 45$ ($B_{\max} = 120$), whereas the second with $T = 9600, r_{\min} = 60, \delta = 10$ resulting in $B_{\min} = 120$ ($B_{\max} = 480$). For both conditions we performed 100 macro-replications. Figures 3(a)-3(b) report the results for the lower and larger noise level, respectively.



(a) $\delta = 1.0$ 2–d case (b) $\delta = 10.0$ 2–d case

Figure 3: TSSO performance under different values of budget per iteration B_k for the 2-d case compared to eTSSO

The same observations as in the 1–d case can be drawn for the 2–d case.

Based on the observed results, an appropriate sequence of B_k, where $k = 1, \ldots, K$ are the performed iterations, should satisfy two main desirable characteristics:

1. In the case of low noise, B_k should not be increased to a great extent especially at the first iterations, to favor the search, since only few replications are needed to provide a good point estimate of the function value.

2. In the case of high noise, larger B_k should be allocated to improve the accuracy in presence of noisy function estimations.

If we look into the eTSSO results in Figures 3(a)-3(b), it is important to notice that, in case of high noise for the 2-d function we observe that TSSO can be slightly better than eTSSO. In particular, we observe that, with $B = 240$, the average location error $|\boldsymbol{x} - \boldsymbol{x}^*| = 0.0723$ while eTSSO reaches $|\boldsymbol{x} - \boldsymbol{x}^*| = 0.0768$. This shows that, potentially, TSSO can be better than the new algorithm: the 100 macro-replications enable a statistical test of the significance of the difference between the two observed sample averages (namely, 0.0723 and 0.0768), and, for this value of the budget, TSSO was statistically better according to a 95% confidence paired t-test. Nevertheless, finding the value $B = 240$ is far from an easy task.

Summary observations In the following, we summarize the main differences between TSSO and eTSSO:

- eTSSO has good performance with respect to TSSO and it is always performing better than the average performance of TSSO, where the average is taken over the different values of budget per iteration B. A relevant advantage of the algorithm is that it does not require the user to define any arbitrary value for B;

- The adaptiveness of the allocation scheme results in eTSSO performing better than TSSO in the estimation of the function ($|y(X^*) - y^*|$) when the noise is larger since the budget increase enhances the evaluation. This contributes to improve the identification of the optimal location which is always satisfactory;

- The adaptiveness of the new allocation leads to a more effective use of the available budget: while TSSO would require to size the appropriate B based on the specific total available budget, eTSSO adapts to T and its performance is consistently improving as the total number of replications that can be performed increases.

The consistency of the performance of eTSSO with the associated new budget management is verified in higher dimensions in Section 5.2.

5.2 Performance Comparison

In this section, we compare the performance of eTSSO and TSSO using several test functions. In addition to TSSO, we implemented the Minimum Quantile (MQ) algorithm (Picheny et al., 2013), and the SKO algorithm (Huang et al., 2006), which we extended to the heterogeneous variance case following the same approach proposed in Jalali et al. (2017).

Minimum Quantile algorithm The MQ algorithm chooses the point with minimum Kriging quantile, $q(\mathbf{x}) = \hat{y}(\mathbf{x}) + \Phi^{-1}(\beta_{\text{MQ}})\, s(\mathbf{x})$, with $\beta_{\text{MQ}} \in (0, 0.5]$, as infill point, i.e., $\mathbf{x} \in \arg\min_{\mathbf{x}\in\mathbb{X}} q(\mathbf{x})$. MQ does not necessitate information about the variance form and it allocates a fixed number B_{MQ} of replications per iteration which is decided prior to running the algorithm. MQ allows for revisits: at any iteration k, it is possible to sample a point that has been already evaluated, and add B_{MQ} additional replications to it. As a result, the sampled points can receive a different total number of replications depending on how often they are re-sampled.

Adapted Sequential Kriging Optimization (SKO) The SKO in Huang et al. (2006) chooses the location that maximizes the AEI score defined as

$$AEI(\mathbf{x}) = E\left[\max\left(\hat{y}(\mathbf{x}_k^*) - \hat{y}(\mathbf{x}), 0\right)\right]\left(1 - \frac{\tau(\mathbf{x})}{\sqrt{s^2(\mathbf{x}) + \tau^2(\mathbf{x})}}\right).$$

Where $\hat{y}(\mathbf{x}_k^*)$ is the kriging prediction at the point with minimum $q(\mathbf{x})$ among the simulated points with $\beta_{\text{SKO}} \in (0.5, 1]$. The algorithm was originally designed for homogeneous noise: $\tau^2(\mathbf{x})$ was introduced in Jalali et al. (2017) to reflect the presence of heterogeneous noise. As proposed in Jalali et al. (2017), the $\tau^2(\mathbf{x})$ prediction is obtained by estimating a deterministic Kriging model for the sample variance. SKO, as MQ, uses a fixed number of replications per iteration B_{SKO} and allows for revisits. Like MQ only one point at the time is sampled and or re-evaluated.

Objective of the Empirical Study We analyze eTSSO under different levels of the total budget, T, and noise magnitudes, δ, to evaluate the robustness and performance of the algorithm in its variants. In particular, we collect and discuss the following output metrics: (1) the location error $|\boldsymbol{x} - \boldsymbol{x}^*|$ computed as the Euclidean distance between the location identified by eTSSO and the, known, global optimum, and (2) the error in the function estimation $|y - y^*|$, where $y = y(\boldsymbol{x})$. We perform this study on several functions with increasing dimensionality. We consider a two-dimensional (tetra-modal), a three-dimensional (Hartmann 3), and a six-dimensional (Hartmann 6) test function. Preliminary tests were conducted to set an appropriate value of the total budget T as well as the minimum number of replications. For TSSO, we set the value B to be in the middle of the range $r_{\min}$ and $B_{\max}$, i.e., in the region where we obtained the best results according to the numerical evidence for both the 1-d and 2-d case studied in section 5.1. For eTSSO, no matter the variant, such a setting is not needed, and we only need to set $B_0 = r_{\min}$.

Tables 1, 4 and 7, report the noise level δ, the size of the initial set of points N_0, the total budget T, the algorithm to which each row of the table refers (Algorithm), the minimum number of replications $r_{\min}$, and B, which is only required for MQ, SKO, and TSSO. In all the Tables, the average for the location error ($|\boldsymbol{x} - \boldsymbol{x}^*|$) or the function evaluation error ($|y - y^*|$) is reported in bold, normal, or italics when the algorithm statistically outperforms, is identical, or under performs TSSO, respectively. The test is conducted with 95% confidence level. In particular, we considered 6 simultaneous comparisons (each algorithm against the original TSSO), therefore a confidence $1 - (0.05/6)$ was adopted to determine the statistical significance of each single test (Miller, 1981; Montgomery, 2017).

For all the tests, following the parametrization recommended in Jalali et al. (2017), we set $B_{\text{MQ}} = B_{\text{SKO}} = 55$, and $\beta_{\text{MQ}} = 0.1, \beta_{\text{SKO}} = 0.84$. Note that the budget B_{MQ}, as well as B_{SKO}, is substantially different from the B used in TSSO. Indeed, while B is distributed among several sampled points, MQ

and SKO only sample one location at a time, whether it is a new or revisited solution.

Two dimensional case The tetra-modal function below was analyzed

$$Y(x_1, x_2) = -5(1-(2x_1-1)^2)(1-(2x_2-1)^2)(4+2x_1-1)\left(0.05^{(2x_1-1)^2} - 0.05^{(2x_2-1)^2}\right)^2.$$

As noise, we applied to the function an additive Gaussian Process $\xi(x)$ with mean $\mathbf{0}$ and diagonal variance covariance matrix with diagonal elements $\sigma_\xi^2 = \delta \cdot (x_1 + x_2)$.

Table 1 shows the results and the parameter settings adopted to compare the performance the different algorithms. All this information characterize the experiment settings, whereas the location and estimation errors are reported in terms of both mean and standard error over 100 macro-replications.

Table 1: Summary results for the tetra-modal function (low noise, $\delta = 1.0$)

N_0	T	Algorithm	$r_{\min}$	B	$\lvert\boldsymbol{x} - \boldsymbol{x}^*\rvert$		$\lvert y - y^*\rvert$	
					average	std err	average	std err
10	2400	MQ	-	55	0.0445	0.0546	0.3886	0.3078
10	2400	SKO	-	55	0.0445	0.0546	0.3886	0.3078
10	2400	TSSO	10	130	0.0083	0.0007	0.0694	0.0053
10	2400	eTSSO_O	10	–	0.0064	0.0005	**0.0422**	0.0039
10	2400	eTSSO_A	10	–	**0.0034**	0.0006	**0.0357**	0.0025
10	2400	eTSSO_G	10	–	**0.0033**	0.0007	**0.0330**	0.0027
10	2400	eTSSO_E	10	–	**0.0020**	0.0005	**0.0385**	0.0029
20	2400	MQ	-	55	*0.0772*	0.0142	*0.3524*	0.0192
20	2400	SKO	-	55	*0.0772*	0.0142	*0.3524*	0.0192
20	2400	TSSO	10	70	0.0119	0.0009	0.0764	0.0056
20	2400	eTSSO_O	10	–	**0.0072**	0.0005	**0.0462**	0.0041
20	2400	eTSSO_A	10	–	**0.0027**	0.0006	**0.0339**	0.0028
20	2400	eTSSO_G	10	–	**0.0026**	0.0007	**0.0332**	0.0034
20	2400	eTSSO_E	10	–	**0.0022**	0.0006	**0.0400**	0.004

Consistent with the results already obtained, we observe that eTSSO is statistically better or equivalent to TSSO in an least one variant (when considering an overall confidence of 95% and the normal approximation), while both TSSO and eTSSO appear to be superior to MQ and SKO for most of the instances. The result is less "statistically" striking for the high noise case, mainly due to the fact that attaining 95% simultaneous confidence leads to large intervals. Nonetheless, results from experiment with noise are consistent with those in low noise.

Focusing on the low noise case, we can see that, independently from the initial conditions, eTSSO beats TSSO in all its variants. In particular, eTSSO_O appears to be the worst performer and this may be due to the fact that this allocation was observed as the most conservative in low noise settings leading to larger budget increase early on in the search (this was also observed in (Jalali et al., 2017)). On the other hand, as we predicted, the eager variant eTSSO_E shows the best location performance in this family of experiments. In fact, eTSSO_E is more biased towards exploration leading to lower budget allocations per iterate. While such exploration is beneficial in the low noise case in identifying a good solution, we observe that the error in the function evaluation is higher. Finally, eTSSO_A and eTSSO_G appear to have similar

behavior. This makes sense: while eTSSO_A uses averaging as a means to mix sample and un-sampled points information, eTSSO_G does the same choosing two "representative points".

Table 2: Summary results for the tetra-modal function (high noise, $\delta = 5.0$)

N_0	T	Algorithm	$r_{\min}$	B	$\lvert\boldsymbol{x} - \boldsymbol{x}^*\rvert$		$\lvert y - y^*\rvert$	
					average	std err	average	std err
10	6000	MQ	-	55	*0.4803*	0.0301	*0.8886*	0.0421
10	6000	SKO	-	55	*0.4804*	0.0301	*0.8787*	0.0413
10	6000	TSSO	20	315	0.0125	0.0009	0.1135	0.0094
10	6000	eTSSO_O	20	–	0.0085	0.0009	0.0852	0.0074
10	6000	eTSSO_A	20	-	0.0105	0.0010	0.1299	0.0101
10	6000	eTSSO_G	20	-	0.0125	0.0012	0.1460	0.0136
10	6000	eTSSO_E	20	–	*0.0335*	0.0068	*0.3020*	0.0166
20	6000	MQ	-	55	*0.4803*	0.0301	*0.8886*	0.0421
20	6000	SKO	-	55	*0.5128*	0.0287	*0.9372*	0.0409
20	6000	TSSO	20	165	0.0145	0.001	0.1346	0.0124
20	6000	eTSSO_O	20	–	**0.0094**	0.0007	0.0870	0.0079
20	6000	eTSSO_A	20	–	0.0118	0.011	0.1180	0.0089
20	6000	eTSSO_G	20	–	0.0113	0.0012	0.1173	0.0113
20	6000	eTSSO_E	20	–	0.0321	0.0094	*0.2553*	0.0232

Concerning the higher noise case (Table 2), we notice that eTSSO_O shows better performance relatively to the low-noise case. In this case the approach empirically produces better selections. Still, eTSSO_A and eTSSO_G are competitive with eTSSO_O and behave similarly as in the lower noise case. Not surprisingly, the eager algorithm eTSSO_E does not perform well due to the low budget allocated to evaluation. Also, we observed that, in case of larger noise levels, the number of points sampled by eTSSO decreases despite the fact that a larger initial budget T is available. This is reasonable: as the noise increases, the budget increases at faster rates (refer to equation (18)) as the algorithm progresses. As a result, the budget is quickly exhausted. This is detrimental for the performance in terms of search as the algorithm samples less points. However, the fewer sampled points are characterized by lower sample variance because of the large allocated number of replications, thus improving the model estimation. This can be observed from the improved performance of eTSSO in the response estimation. Intuitively, it seems clear that the simulation of more points - but with large intrinsic noise - does not improve the insight into the behavior of the I/O function, so further effort in simulating new points would not be effective.

The results also suggest that the number of initial points N_0 is not always significant across the different variants of eTSSO (confirming the outcomes in (Picheny et al., 2013)). However, we noticed a strong interaction between N_0 and $r_{\min}$. In the case of low noise and finite budget, we should focus on the search and maximize the number of sampled points, since only a small number of replications is required to evaluate the function at each location. As a result, in case a large initial value of $r_{\min}$ is assigned, better results can be achieved with lower N_0 as the algorithm has more budget to perform the search. Hence, the better results observed for the case $N_0 = 10$ in Table 1. On the other hand, under large noise set ups, algorithms that have larger initial samples appear to be more competitive.

Three dimensional case In this part, we analyze the Hartmann–3 function:

$$Y(x_1, x_2, x_3) = -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{3} A_{ij}\left(x_j - p_{ij}\right)^2\right]$$

Table 3: Parameters A_{ij} and P_{ij} of the Hartmann-3 function

A_{ij}			p_{ij}		
3	10	30	0.3689	0.117	0.2673
0.1	10	35	0.4699	0.4387	0.747
3	10	30	0.1091	0.8732	0.5547
0.1	10	35	0.03815	0.5743	0.8828

With $0 \leq x_i \leq 1$ for $i = 1, 2, 3$; parameters $\alpha = (1.0, 1.2, 3.0, 3.2)$, and A_{ij} and Pij given in Table 3. The function has a global minimum at $\boldsymbol{x}^* = (0.114614, 0.555649, 0.852547)$ with $y(\boldsymbol{x}^*) = -3.86278$; the function has three additional local minima. As noise, we applied to the function an additive Gaussian Process $\xi(\boldsymbol{x})$ with mean $\mathbf{0}$ and diagonal variance covariance matrix with diagonal elements $\sigma_\xi^2 = \delta \cdot \left(\sum_{i=1}^{3} |x_i|\right)$. Table 4 shows the obtained results.

Table 4: Summary results for the Hartmann 3 function (low noise, $\delta = 1.0$)

N_0	T	Algorithm	$r_{\min}$	B	$\lvert\boldsymbol{x} - \boldsymbol{x}^*\rvert$		$\lvert y - y^*\rvert$	
					average	std err	average	std err
20	3200	MQ	-	55	*0.3551*	0.0261	*0.3071*	0.0235
20	3200	SKO	-	55	*0.4202*	0.0298	*0.3461*	0.0260
20	3200	TSSO	15	87	0.1788	0.0156	0.1034	0.0075
20	3200	eTSSO_O	15	–	0.1294	0.0126	0.0763	0.0074
20	3200	eTSSO_A	15	–	0.1021	0.0223	**0.0550**	0.0054
20	3200	eTSSO_G	15	–	0.1009	0.0208	**0.0516**	0.0047
20	3200	eTSSO_E	15	–	**0.0492**	0.0151	**0.0294**	0.0039
30	3200	MQ	-	55	*0.3613*	0.0255	*0.2852*	0.0209
30	3200	SKO	-	55	*0.5315*	0.0322	*0.3637*	0.0263
30	3200	TSSO	15	60	0.1891	0.0166	0.1372	0.0085
30	3200	eTSSO_O	15	–	0.1254	0.0115	**0.0824**	0.006
30	3200	eTSSO_A	15	–	**0.0835**	0.0166	**0.0602**	0.0054
30	3200	eTSSO_G	15	–	0.1165	0.0199	**0.0749**	0.0061
0	3200	eTSSO_E	15	–	**0.0789**	0.0190	**0.0422**	0.0050

First, we notice that the increased level of complexity of the function causes a decrease in the performance on both TSSO and eTSSO in all its variants. Nonetheless, eTSSO is statistically better or equivalent to TSSO in almost all variants, and both TSSO and eTSSO appear to be superior to MQ and SKO.

Table 5: Summary results for the Hartmann 3 function (high noise, $\delta = 5.0$)

N_0	T	Algorithm	$r_{\min}$	B	$\lvert\boldsymbol{x} - \boldsymbol{x}^*\rvert$		$\lvert y - y^*\rvert$	
					average	std err	average	std err
20	8000	MQ	-	55	*0.5406*	0.0302	*1.1372*	0.1001
20	8000	SKO	-	55	*0.5636*	0.0297	*1.1315*	0.0989
20	8000	TSSO	25	212	0.2277	0.0205	0.2756	0.0178
20	8000	eTSSO_O	25	–	0.2716	0.0224	0.2072	0.0156
20	8000	eTSSO_A	25	–	0.2747	0.0299	0.2271	0.0110
20	8000	eTSSO_G	25	–	0.2053	0.0286	**0.2051**	0.0089
20	8000	eTSSO_E	25	–	0.3438	0.0299	0.2270	0.0120
30	8000	MQ	-	55	*0.5819*	0.0313	*1.1168*	0.0894
30	8000	SKO	-	55	*0.6681*	0.0350	*1.3564*	0.1052
30	8000	TSSO	25	145	0.3154	0.0235	0.3488	0.0239
30	8000	eTSSO_O	25	–	0.2216	0.0231	0.2600	0.0207
30	8000	eTSSO_A	25	–	0.2633	0.0298	0.2612	0.0141
30	8000	eTSSO_G	25	–	0.2374	0.0286	**0.2240**	0.0117
30	8000	eTSSO_E	25	–	0.3181	0.0282	**0.2427**	0.0125

Nevertheless, eTSSO_E is again performing worse than all the alternative algorithms for the case with high noise. We highlight the case with $\delta = 5.0$ and $N_0 = 20$, where a better average performance is observed only for eTSSO_G. As already stated in section 3, TSSO can show better performance with respect to eTSSO, especially in case of large noise. Indeed, eTSSO might be affected by an early termination due to the budget exhaustion.

Six dimensional case Finally, we examined a six–dimensional case. In particular, we study the Hartmann-6 test function defined as

$$Y(x_1, x_2, x_3, x_4, x_5, x_6) = -\sum_{i=1}^{4} \alpha_i \exp\left[-\sum_{j=1}^{6} \alpha_{ij} (x_j - p_{ij})^2\right].$$

The parameters of the function are in Table 6.

Table 6: Parameters α_{ij} and p_{ij} of the Hartmann–6 function

α_{ij}	10.0	3.0	17.0	3.5	1.7	8.0
	0.05	10.0	17.0	0.1	8.0	14.0
	3.0	3.5	1.7	10.0	17.0	8.0
	17.0	8.0	0.05	10.0	0.1	14.0
p_{ij}	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

With $0 \leq x_i \leq 1$ for $i = 1, \ldots, 6$; parameters $\alpha = (1.0, 1.2, 3.0, 3.2)$, and α_{ij} and p_{ij} given in Table 6. This function has a global minimum at $\boldsymbol{x}^* = (0.20169, 0.150011, 0.476874, 0.275332, 0.311652, 0.6573)$

with $y(\boldsymbol{x}^*) = -3.32237$; the function also has five additional local minima. As noise, we applied to the function an additive Gaussian Process $\xi(x)$ with mean $\mathbf{0}$ and diagonal variance covariance matrix with diagonal elements $\sigma_\xi^2 = \delta \cdot \left(\sum_{i=1}^{6} |x_i|\right)$. The obtained results are in Table 7.

Table 7: Summary results for the Hartmann 6 function (low noise, $\delta = 1.0$)

N_0	T	Algorithm	$r_{\min}$	B	$\lvert\boldsymbol{x} - \boldsymbol{x}^*\rvert$		$\lvert y - y^*\rvert$	
					average	std err	average	std err
40	6400	MQ	-	55	*1.3919*	0.0562	*1.6070*	0.0314
40	6400	SKO	-	55	*1.4539*	0.0597	*1.6027*	0.0312
40	6400	TSSO	25	92	0.2746	0.0384	0.18629	0.0135
40	6400	eTSSO_O	25	-	0.16211	0.0384	0.16211	0.0155
40	6400	eTSSO_A	25	-	0.14807	0.0334	0.18928	0.0143
40	6400	eTSSO_G	25	-	0.14794	0.0473	0.21957	0.0909
40	6400	eTSSO_E	25	-	0.13832	0.0425	0.13676	0.0609
60	6400	MQ	-	55	0.3042	0.0165	*0.5052*	0.0214
60	6400	SKO	-	55	0.3025	0.0152	*0.5060*	0.0217
60	6400	TSSO	25	65	0.19409	0.0174	0.19942	0.0182
60	6400	eTSSO_O	25	-	0.16705	0.0356	**0.14417**	0.0137
60	6400	eTSSO_A	25	-	0.15483	0.0377	**0.14612**	0.0124
60	6400	eTSSO_G	25	-	0.15639	0.0309	0.15457	0.0189
60	6400	eTSSO_E	25	-	0.14378	0.0355	*0.25935*	0.0194

Despite the performance of the algorithms are generally worse than in the lower dimensional cases, we observe that eTSSO is better or equivalent to TSSO in at least one implementation. Also in this case, both TSSO and eTSSO appear to be superior to MQ and SKO.

Table 8: Summary results for the Hartmann 6 function (high noise, $\delta = 5.0$)

N_0	T	Algorithm	$r_{\min}$	B	$\lvert\boldsymbol{x} - \boldsymbol{x}^*\rvert$		$\lvert y - y^*\rvert$	
					average	std err	average	std err
40	16000	MQ	-	55	*1.5729*	0.0419	*1.8362*	0.0163
40	16000	SKO	-	55	*1.6764*	0.0519	*1.8304*	0.0216
40	16000	TSSO	35	217	0.33579	0.0472	0.35308	0.0348
40	16000	eTSSO_O	35	-	0.27443	0.0323	0.32487	0.0116
40	16000	eTSSO_A	35	-	0.27339	0.0340	0.29926	0.0214
40	16000	eTSSO_G	35	-	0.29575	0.0327	0.32266	0.0295
40	16000	eTSSO_E	35	-	0.33904	0.0425	0.34437	0.0344
60	16000	MQ	-	55	0.3619	0.0270	*0.5976*	0.0362
60	16000	SKO	-	55	0.3575	0.0297	*0.5880*	0.0349
60	16000	TSSO	35	150	0.35282	0.0429	0.36517	0.0258
60	16000	eTSSO_O	35	-	0.25337	0.0744	0.32929	0.0246
60	16000	eTSSO_A	35	-	0.23088	0.0243	0.32565	0.0342
60	16000	eTSSO_G	35	-	0.28314	0.0945	0.33891	0.0296
60	16000	eTSSO_E	35	-	0.37284	0.0445	0.39936	0.0544

Summary We observe good results from the several variants of eTSSO. In particular, across different dimensions and noise levels, eTSSO has always variants that are superior, statistically, to TSSO. Overall, eTSSO_{O} and eTSSO_{G} appear to be the most robust with respect to the different test cases. This is expected: eTSSO_{E} performs well for low noise, but it leads to under sampling in the case of high noise with a detrimental effect on the performance; eTSSO_{A} protects against low budgets, but it may be ineffective due to the high heterogeneity of the noise across the design space. In our implementation and tests, the MQ and SKO algorithms appear to never perform better than TSSO or eTSSO. While this is expected for MQ, treated generally as a benchmark, the issues in SKO reveal the importance of the underlying assumption of knowledge of the variance structure.

6 Conclusions

In this paper, we propose a two-stage sequential optimization procedure, eTSSO, which generalizes the previously proposed TSSO algorithm by trying to reduce its sensitivity to the budget allocated at each iteration k, namely, B_k. Indeed, we observed that increasing the budget at each iteration has the positive effect to decrease the influence of the budget per iteration B, adopted in the original TSSO, and the noise magnitude. Hence, we generalize TSSO by generating the sequence of the budget per iteration B_k, stochastically and dynamically, according to the updated information coming from the simulation. In this regard, we propose a general budget allocation rule that satisfies the conditions required for convergence. We then generate four different variants of the rule that put different emphasis on search and evaluation.

We analyzed the asymptotic properties in terms of convergence and convergence rate of eTSSO. In particular, in order to perform the analysis, we interpret eTSSO as a stochastic recursion procedure. Consequently, we are able to exploit the results from (Bull, 2011) and (Pasupathy et al., 2018) to prove the desired properties.

The numerical studies reveal a good finite time behavior of the algorithm in its four instantiations when the parameters of the underlying stochastic model are sequentially estimated as the search progresses. The performance of eTSSO have been tested against functions of increasing dimensions and results have been compared with the original TSSO. eTSSO is shown to be better or statistically equivalent to TSSO in most of the variants given the total available budget T, in the proposed examples. The performance of eTSSO is sensitive to the function dimensions, nonetheless the algorithm behavior is consistent with respect to the lower dimension cases, proving the generality of the proposed approach and of the empirical results. Also, eTSSO_{O} and eTSSO_{G} appear to be the most robust variants.

Future research includes the extension of the approach to the case where multiple constraints need to be considered that can only be evaluated with noise; another important extension of the framework is in the area of multiple objectives.

Acknowledgments

This research was partially supported by the NSF CMMI Grant #1829238.

References

Ankenman, B., B. L. Nelson, and J. Staum. 2010. "Stochastic kriging for simulation metamodeling". *Operations research* 58 (2): 371–382.

Attouch, H. 1984. "Variational Convergence of Functions and Operators".

Brochoff, D., B. Bischl, and T. Wagner. 2015. "The impact of initial designs on the performance of MATSuMoTo on the noiseless BBOB-2015 testbed: a preliminary study". *GECCO'15 Companion, Madrid, Spain*.

Bull, A. D. 2011. "Convergence rates of efficient global optimization algorithms". *The Journal of Machine Learning Research* 12:2879–2904.

Chen, C.-H., J. Lin, E. Yücesan, and S. E. Chick. 2000. "Simulation budget allocation for further enhancing the efficiency of ordinal optimization". *Discrete Event Dynamic Systems* 10 (3): 251–270.

Efron, B., and R. Tibshirani. 1997. "Improvements on cross-validation: the 632+ bootstrap method". *Journal of the American Statistical Association* 92 (438): 548–560.

Erickson, C. B., B. E. Ankenman, and S. M. Sanchez. 2018. "Comparison of Gaussian process modeling software". *European Journal of Operational Research* 266 (1): 179–192.

Figueira, G., and B. Almada-Lobo. 2014. "Hybrid simulation–optimization methods: A taxonomy and discussion". *Simulation Modelling Practice and Theory* 46:118–134.

Fu, M. C. 2015. *Handbook of simulation optimization*, Volume 216. Springer.

Huang, D., T. T. Allen, W. I. Notz, and N. Zeng. 2006. "Global optimization of stochastic black-box systems via sequential kriging meta-models". *Journal of global optimization* 34 (3): 441–466.

Jalali, H., I. Van Nieuwenhuyse, and V. Picheny. 2017. "Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise". *European Journal of Operational Research* 261 (1): 279–301.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient global optimization of expensive black-box functions". *Journal of Global optimization* 13 (4): 455–492.

Kim, S.-H., and B. L. Nelson. 2007. "Recent advances in ranking and selection". In *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come*, 162–172. IEEE Press.

Kleijnen, J. P. 2008. "Response surface methodology for constrained simulation optimization: An overview". *Simulation Modelling Practice and Theory* 16 (1): 50–64.

Kleijnen, J. P., W. van Beers, and I. Van Nieuwenhuyse. 2012. "Expected improvement in efficient global optimization through bootstrapped kriging". *Journal of global optimization* 54 (1): 59–73.

Kleijnen, J. P., and W. C. Van Beers. 2005. "Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments". *European Journal of Operational Research* 165 (3): 826–834.

Locatelli, M. 1997. "Bayesian algorithms for one-dimensional global optimization". *Journal of Global Optimization* 10 (1): 57–76.

Mehdad, E., and J. P. Kleijnen. 2018. "Efficient global optimisation for black-box simulation via sequential intrinsic Kriging". *Journal of the Operational Research Society* 69 (11): 1725–1737.

Mete, H. O., Y. Shen, Z. B. Zabinsky, S. Kiatsupaibul, and R. L. Smith. 2011. "Pattern discrete and mixed Hit-and-Run for global optimization". *Journal of Global Optimization* 50 (4): 597–627.

Miller, R. G. 1981. *Simultaneous statistical inference*. Springer.

Montgomery, D. C. 2017. *Design and analysis of experiments*. John wiley & sons.

Myers, R. H., D. C. Montgomery, and C. M. Anderson-Cook. 2009. *Response surface methodology: process and product optimization using designed experiments*, Volume 705. John Wiley & Sons.

Ng, S. H., and J. Yin. 2012. "Bayesian kriging analysis and design for stochastic simulations". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 22 (3): 17.

Pasupathy, R., P. Glynn, S. Ghosh, and F. S. Hashemi. 2018. "On sampling rates in simulation-based recursions". *SIAM Journal on Optimization* 28 (1): 45–73.

Pedrielli, G., and S. H. Ng. 2015. "Kriging-based simulation-optimization: a stochastic recursion perspective". In *Winter Simulation Conference (WSC), 2015*, 3834–3845. IEEE.

Picheny, V., T. Wagner, and D. Ginsbourger. 2013. "A benchmark of kriging-based infill criteria for noisy optimization". *Structural and Multidisciplinary Optimization* 48 (3): 607–626.

Quan, N., J. Yin, S. H. Ng, and L. H. Lee. 2013. "Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints". *Iie Transactions* 45 (7): 763–780.

Robinson, S. 1996. "Analysis of Sample–Path Optimization". *Mathematics of Operations Research* 21:513–528.

Santner, T. J., B. J. Williams, W. Notz, and B. J. Williams. 2003. *The design and analysis of computer experiments*, Volume 1. Springer.

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.

Shi, L., and S. Ólafsson. 2000. "Nested partitions method for global optimization". *Operations Research* 48 (3): 390–407.

Stein, M. L. 1999. *Interpolation of spatial data: some theory for kriging*. Springer.

Tekin, E., and I. Sabuncuoglu. 2004. "Simulation optimization: A comprehensive review on theory and applications". *IIE Transactions* 36 (11): 1067–1081.

Vehtari, A., A. Gelman, and J. Gabry. 2017. "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC". *Statistics and Computing* 27 (5): 1413–1432.

Vogel, S., and P. Lachout. 2003. "On continuous convergence and epi-convergence of random functions. Part I: Theory and relations". *Kybernetika* 39 (1): 75–98.

Wan, X., J. F. Pekny, and G. V. Reklaitis. 2005. "Simulation-based optimization with surrogate modelsApplication to supply chain management". *Computers & chemical engineering* 29 (6): 1317–1328.

Wang, H., R. Pasupathy, and B. W. Schmeiser. 2013. "Integer-Ordered Simulation Optimization using R-SPLINE: Retrospective Search with Piecewise-Linear Interpolation and Neighborhood Enumeration". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 23 (3): 17.

Wright, S., and J. Nocedal. 1999. "Numerical optimization". *Springer Science* 35 (67-68): 7.

Xu, J., E. Huang, C.-H. Chen, and L. H. Lee. 2015. "Simulation optimization: a review and exploration in the new era of cloud computing and big data". *Asia-Pacific Journal of Operational Research* 32 (03): 1550019.

Xu, J., B. L. Nelson, and J. Hong. 2010. "Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 20 (1): 3.

Yin, G. G., and H. J. Kushner. 2003. *Stochastic approximation and recursive algorithms and applications*. Springer.

Yin, J., S. H. Ng, and K. M. Ng. 2011. "Kriging metamodel with modified nugget-effect: The heteroscedastic variance case". *Computers & Industrial Engineering* 61 (3): 760–777.

Zhu, C., J. Xu, C.-H. Chen, L. H. Lee, and J. Hu. 2013. "Determining the optimal sampling set size for random search". In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, 1016–1024. IEEE Press.

APPENDIX

Proofs

In this section, we report the proofs of theorems in section 4.2.

Lemma. ***1*** *There exists $\kappa \in \mathbb{R}$ such that, for any $(\mathbb{S}, \mathbb{S}') : \mathbb{S} \subset \mathbb{X}, \mathbb{S}' \subset \mathbb{X}$, $||h(\mathbb{S}) - h(\mathbb{S}')|| \leq \kappa D(\mathbb{S}, \mathbb{S}')$, where $D(\cdot)$ represents the distance between two sets of points.*

Proof. We prove that, given two "close" (in Euclidean sense) sequences $\{\boldsymbol{x}_i\}_{i=1}^k, \{\boldsymbol{x}'_i\}_{i=1}^k$, and the associated filtrations $\mathcal{F}_{\pi_k}, \mathcal{F}'_{\pi_k}$, the corresponding "distances" $h\left(\{\boldsymbol{x}_i\}_{i\leq k}\right), h\left(\{\boldsymbol{x}'_i\}_{i\leq k}\right)$ must be close in Euclidean sense. According to (21), the sequences of interest are a result of the expected improvement T_{π_k}, and they are fully defined by:

$$\boldsymbol{x}^*_{lk} - \boldsymbol{x}_{l,k+1}, \quad l = 1, \ldots, d \tag{28}$$

where $\boldsymbol{x}_{k+1} := \arg\max_{\boldsymbol{x} \in \mathbb{X} \setminus \mathbb{S}} T_{\pi_k}(\boldsymbol{x})$ and the subscript refers to the l-th dimension. If Assumption 7 holds, the smoothness of the simulation response guarantees that $\{\boldsymbol{x}_i\}_{i=1}^k$ and $\{\boldsymbol{x}'_i\}_{i=1}^k$ will generate similar values of response functions, therefore, the generated $\boldsymbol{x}^*_k$ will be similar. However, to produce similar sequences, we need to guarantee that the generated sampling points will be similar. Hence, we need to focus the analysis on $\boldsymbol{x}_{k+1}$. Since the sampling points are generated by evaluating the expected improvement function T_{π_k}, we need to guarantee Lipschitz continuity for T_{π_k}.
Following (Locatelli, 1997), the expected improvement can be written as:

$$T_{\pi_k}(\boldsymbol{x}; \mathcal{F}_{\pi_k}) = s_{\pi_k}(\boldsymbol{x}_i)\phi\left(\frac{y(\boldsymbol{x}^*_k) - \hat{\mu}_k}{s_{\pi_k}(\boldsymbol{x})}\right) - (y(\boldsymbol{x}^*_k) - \hat{\mu}_k)\left(1 - \Phi\left(\frac{y(\boldsymbol{x}^*_k) - \hat{\mu}_k}{s_{\pi_k}(\boldsymbol{x})}\right)\right), \tag{29}$$

where ϕ, Φ are the normal distribution *pdf* and *cdf*, respectively. It is possible to show that $T_{\pi_k}(\boldsymbol{x}; \mathcal{F}_{\pi_k})$ satisfies Lipschitz continuity when $\hat{\mu}_k = (4)$ and $s_{\pi_k}(\boldsymbol{x}) = (3)$. Specifically, we prove that $\frac{dT_{\pi_k}(\boldsymbol{x}; \mathcal{F}_{\pi_k})}{dx} < \infty$, $\forall \boldsymbol{x} \in \mathbb{X}$. Under Gaussian Processes, $\phi\left(\frac{y(\boldsymbol{x}^*_k) - \hat{\mu}_k}{s_{\pi_k}(\boldsymbol{x})}\right)$ and $\Phi\left(\frac{y(\boldsymbol{x}^*_k)_k - \hat{\mu}_k}{s_{\pi_k}(\boldsymbol{x})}\right)$ are the *pdf* and *cdf* of a normal distribution, therefore $0 < \hat{\mu}_k, s^2_{\pi_k}(\mathbf{x}) < \infty$, but it is important to carefully consider the derivatives of $\hat{\mu}_k$ and $s_{\pi_k}(\boldsymbol{x})$. Let us rewrite equation (3) as:

$$s_{\pi_k}(\boldsymbol{x}) = \tau\left(1 - \left(\sum_{h=1}^{k}\sum_{g=1}^{k} e^{-\sum_{j=1}^{d}\phi_j(x_j - x_{hj})^2} e^{-\sum_{j=1}^{d}\phi_j(x_{gj} - x_j)^2} r^{-1}_{hg}\right)\right)^{1/2} \tag{30}$$

If assumption 5 holds, equation (30) is infinitely differentiable with respect to $\boldsymbol{x}_i$, but the derivative will be finite depending on $r^{-1}_{hg}, \quad \forall (h, g)$, i.e., the components of the matrix $\mathbf{R}^{-1}$ (h-th row, g-th column). Therefore, for the existence of the derivative, we need to require $\mathbf{R}$ to be non-singular, enforcing assumption 5 to hold. The same reasoning applies to the mean. Hence, function $T_{\pi_k}(\boldsymbol{x}; \mathcal{F}_{\pi_k})$ is Lipschitz continuous, proving the lemma. □

Lemma. ***2*** *Let us consider a Gaussian correlation function, then $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{x}^*$ will hold for the EGO algorithm. Moreover, if Assumption 7 is satisfied, the result will hold for any initial sampling $\{\mathbf{x}_{k_0}\}_{k_0=1}^{N_0}$,*

i.e., we have uniform convergence.

Proof. We observe that $s_{\pi_k}(\boldsymbol{x}) = \tau^2(1 - \mathbf{c}^T\mathbf{R}'^{-1}\mathbf{c} + \zeta(\boldsymbol{x})^T(\mathbf{1}^T\mathbf{R}'^{-1}\mathbf{1})^{-1}\zeta(\mathbf{x}))$, where $\zeta(\boldsymbol{x}) = 1 - \mathbf{c}^T\mathbf{R}'^{-1}\mathbf{1}$. Denote $\boldsymbol{x}_0$ as the closest design point to $\boldsymbol{x}$. It is easy to see that $s_{\pi_k}(\boldsymbol{x}) \leq \tau^2(1-e^{-\phi_z d^2_{\boldsymbol{x},\boldsymbol{x}_0}}+\zeta(\boldsymbol{x})^T(\mathbf{1}^T\mathbf{R}'^{-1}\mathbf{1})^{-1}\zeta(\boldsymbol{x}))$. Besides, in $\zeta(\boldsymbol{x})$, $\mathbf{c}^T\mathbf{R}'^{-1}\mathbf{1}$ can be treated as the GP prediction (where the mean function is 0) at $\boldsymbol{x}$, given the observations at the design points are all 1. It is then easy to check that $\zeta(\boldsymbol{x}) = \mathcal{O}(|\boldsymbol{x} - \boldsymbol{x}_0|)$ and that $(\mathbf{1}^T\mathbf{R}'^{-1}\mathbf{1})^{-1} < 1$. In this case, we can select a large value M_2 such that $s_{\pi_k}(\boldsymbol{x}) \leq \tau^2(1 - e^{-\phi_z d^2_{\boldsymbol{x},\boldsymbol{x}_0}} + M_2 d^2_{\boldsymbol{x},\boldsymbol{x}_0}) := s^0_{\pi_k}(\boldsymbol{x})$.

According to Assumption 8, the function is bounded. Here, we select a large value M such that the responses are bounded in $(-M, M)$, and thus we have $y(\boldsymbol{x}^*_k) - \hat{\mu}_k < 2M$. We can then consider the following:

$$T_{\pi_k}(\boldsymbol{x}_i; \mathcal{F}_{\pi_k}) \leq s^0_{\pi_k}(\boldsymbol{x}_i)\phi\left(\frac{2M}{s^0_{\pi_k}(\boldsymbol{x}_i)}\right) + 2M\Phi\left(\frac{2M}{s^0_{\pi_k}(\boldsymbol{x})}\right) := T^0_{\pi_k}(\boldsymbol{x}_i; \mathcal{F}_{\pi_k}). \tag{31}$$

Now, we find an upper bound for $T_{\pi_k}(\boldsymbol{x}_i; \mathcal{F}_{\pi_k})$. Note that this upper bound $T^0_{\pi_k}(\boldsymbol{x}_i; \mathcal{F}_{\pi_k})$ is a decreasing function of $s^0_{\pi_k}(\boldsymbol{x}_i)$ and $s^0_{\pi_k}(\boldsymbol{x}_i)$ is a decreasing function of $d^2_{\boldsymbol{x}_i,\boldsymbol{x}_0}$. Therefore, $T^0_{\pi_k}(\boldsymbol{x}_i; \mathcal{F}_{\pi_k})$ decreases as the unobserved point $\boldsymbol{x}_i$ becomes closer to existing design points and $T_{\pi_k}(\boldsymbol{x}_i; \mathcal{F}_{\pi_k})$ becomes even smaller. A similar result, in a single dimension, was obtained in (Locatelli, 1997). Hence, based on assumptions 4-2, equation (31) extends this result to the d-dimensional case. An important consequence of (31), is that it allows to apply the result in Lemma 1 in (Locatelli, 1997) (page 60), obtaining:

$$\lim_{k\to\infty} \max_{i,j\in\mathbb{S}} ||\boldsymbol{x}_i - \boldsymbol{x}_j|| = 0. \tag{32}$$

Equivalently, if the algorithm is never stopped. the sample points will be dense in $\mathbb{X}$, proving convergence of the algorithm.

We are left with the uniform convergence claim. For this, we can refer to the previous result in (Bull, 2011), that shows how under assumption 7 uniform convergence will be achieved and there exists a number of initial points N_0 such that the convergence is guaranteed independently from the specific initial set. □

Lemma. *3 The cumulated budget W_k satisfies $W_k \xrightarrow{k\to\infty} \infty$ w.p.1, and the expected budget at iteration k is finite for any finite value of k.*

Proof. Let us rewrite the sequence of generated budgets as:

$$B_{k+1} = B_0 \cdot \prod_{j=1}^{k}(1 + \Delta_j) \tag{33}$$

where the random variable Δ_k is defined as $\frac{\hat{\sigma}^2_{\xi,k}}{\hat{\sigma}^2_{\xi,k}+s^2_{\pi_k}}$ and it is a random variable. Since $0 < \Delta_k < 1$, we can approximate the previous by:

$$B_0 \leq B_{k+1} \leq B_0 \cdot 2^k \tag{34}$$

As a result, we can formulate $W_k = \sum_{i=1}^{k} B_i$ as:

$$kB_0 \leq W_{k+1} \leq B_0 \cdot \sum_{j=1}^{k} 2^j = B_0 \cdot (2^{k+1} - 2) \tag{35}$$

This sequence goes to infinity as the iterations go to infinity. Nevertheless, at each iteration of the algorithm, the expected budget is bounded below by $B_0(2^{k+1} - 2)$, which is finite for iteration k. □

Lemma. ***4*** *As the number of iterations $k \to \infty$, under assumptions 2-8, the MNEK model $\tilde{\pi}_k$ approaches its deterministic counterpart π_k.*

Proof. Under assumption 2, we consider $\mathcal{L} = Tr\left(\sigma_\xi^2 R_\xi\right)$, where $Tr(\cdot)$ is the trace of a matrix. We show that, as requested, $Tr\left(\sigma_\xi^2 R_\xi\right)$ has all the properties of $\mathcal{L}$. First, it is an error function, therefore strictly decreasing in W_k. We also need to show that there exist a finite number of iterations k satisfying $\mathcal{L} \leq \left(\delta_\epsilon/\psi^2\right)^k$, i.e., the algorithm returns estimates of $\mathcal{L}$ decreasing with δ_ϵ/ψ^2. To prove this aspect we write the covariance matrix in a more convenient form:

$$R^{'} = R_z + R_\xi = \begin{pmatrix} 1 & e^{\left(-\phi \cdot d_{12}^2\right)} & \dots & e^{\left(-\theta \cdot d_{1k}^2\right)} \\ e^{\left(-\phi \cdot d_{21}^2\right)} & 1 & \dots & e^{\left(-\phi \cdot d_{2k}^2\right)} \\ \vdots & \vdots & \vdots & \vdots \\ e^{\left(-\phi \cdot d_{k1}^2\right)} & \dots & \dots & 1 \end{pmatrix} + \begin{pmatrix} \frac{\sigma_\xi^2(\boldsymbol{x}_1)}{W_{1,k}\tau^2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \frac{\sigma_\xi^2(\boldsymbol{x}_k)}{W_{k,k}\tau^2} \end{pmatrix} \tag{36}$$

Here, d_{ij} represents the Euclidean distance between two points (i, j), $W_{i,k}$ represents the number of replications performed at location i up to iteration k according to the eTSSO budget allocation scheme. $R^{'}$ is a random matrix due to the fact that is contains the random budgets W_k. As a result, we analyze $E\left[R^{'}\right]$ and note that, as long as the Gaussian Process parameters are known, (assumption 6) the elements on the diagonal of R_ξ will be limited by decreasing values of δ_ϵ/ψ^2, where $\delta_\epsilon = \frac{1}{\left(\min_{\boldsymbol{x}_j \in \mathbb{S}} E_k W_{j,k}\right)}$ and $\psi^2 = \tau^2$. Hence, it holds that $\delta_\epsilon \in (0, 1)$; moreover, $\left(\left(\min_{\boldsymbol{x}_j \in \mathbb{S}} W_{j,k}\right)\tau^2\right)^{-1} \xrightarrow{\text{wp1}} 0$ as $k \to \infty$, since $\min_{\boldsymbol{x}_j \in \mathbb{S}} W_{j,k} > r_k \to \infty \ \forall k$ (Assumption 2). Hence, $R_\xi \xrightarrow{\text{wp1}} 0$. □

Theorem. ***1****[Convergence of $G\left(W_k, \tilde{\mathbb{S}}_k\right)$] For any $\delta > 0$, and with $E_k W \to \infty$, when $k \to \infty$,* $\sup_{\mathbb{S}_k, \tilde{\mathbb{S}}_k \subseteq \mathbb{X}} Pr\left\{||G(E_k W, \tilde{\mathbb{S}_k}) - g(\mathbb{S}_k)|| > \delta\right\} = O\left(\left[E_k W\left(\mathbf{X}_k^*\right)\right]^{-1/2}\right)$.

Proof. Here, $||\cdot||$ represents the Euclidean distance. $E_k W\left(X_k^*\right)$ is the total expected budget allocated to point X_k^* up to iteration k. Since X_k^* is a random variable, so is the related cumulated budget $W\left(X_k^*\right)$. We use the notation $G(W_k, \mathbb{S})$ to highlight the supremum is computed over any possible subset of size k of sampled points, where k represents the number of iterations of the algorithm *and* the total number of sampled points since a single point is sampled at each algorithm iteration.

Due to Lemma 4, we know that the model converges to the related deterministic counterpart. From Lemma 2, we know that, in such deterministic settings, density is achieved in the solution space, which implies the sampling of all points for $k \to \infty$. This implies that, if $\tilde{\mathbb{S}_k}, \mathbb{S}_k$ represent the set of sampled points by eTSSO and EGO at step k, respectively, then we have $\tilde{\mathbb{S}_k} \xrightarrow[k \to \infty]{} \mathbb{S}_k$.

Recall that $\boldsymbol{x}_k^* = \arg\min_{\boldsymbol{x}\in\mathbb{S}_k} y(\boldsymbol{x})$, $X_k^* = \arg\min_{\boldsymbol{x}\in\tilde{\mathbb{S}_k}} \bar{Y}(\boldsymbol{x})$. We next prove $X_k^* \xrightarrow[k\to\infty]{} \boldsymbol{x}_k^*$ w.p.1. As $\tilde{\mathbb{S}_k} \xrightarrow[k\to\infty]{} \mathbb{S}_k$, we only need to prove that $X_k^* \xrightarrow[k\to\infty]{} X_k^0$ w.p.1, where $X_k^0 = \arg\min_{\boldsymbol{x}\in\tilde{\mathbb{S}_k}} y(\boldsymbol{x})$.

We first prove that $\sum_{k=1}^{\infty} Pr[|\bar{Y}(X_k^*) - y(X_k^0)| > \delta] < \infty, \forall \delta > 0$:

$$\begin{aligned}
&Pr[|\bar{Y}(X_k^*) - y(X_k^0)| > \delta] \\
=&Pr[|\bar{Y}(X_k^*) - y(X_k^*) + y(X_k^*) - y(X_k^0)| > \delta] \\
<&Pr[|\bar{Y}(X_k^*) - y(X_k^*)| > \frac{\delta}{2}] + Pr[|y(X_k^*) - y(X_k^0)| > \frac{\delta}{2}],
\end{aligned}$$

Note that $\forall \boldsymbol{x} \in \tilde{\mathbb{S}_k}$, $|\bar{Y}(\boldsymbol{x}) - y(\boldsymbol{x})|$ is a normal random variable $\mathcal{N}(0, \sigma_\xi^2(\boldsymbol{x})/N_k(x))$. Recall that $\sigma_0^2 = \max_{\boldsymbol{x}\in\mathbb{X}} \sigma_\xi^2(x)$, we have that

$$Pr[|\bar{Y}(\boldsymbol{x}) - y(\boldsymbol{x})| > \frac{\delta}{2}] \le 2\exp(-\frac{\delta^2 N_k(\boldsymbol{x})}{8\sigma_\xi^2(\boldsymbol{x})}) \le 2\exp(-\frac{\delta^2 r_k}{8\sigma_0^2}).$$

The second inequality is based on Assumption 2 that the total number of replications at each input $\boldsymbol{x}$, $N_k(\boldsymbol{x}) > r_k$. Therefore, by union bound,

$$Pr[\max_{\boldsymbol{x}\in\tilde{\mathbb{S}_k}} |\bar{Y}(\boldsymbol{x}) - y(\boldsymbol{x})| > \frac{\delta}{2}] \le 2(k + N_0)\exp(-\frac{\delta^2 r_k}{8\sigma_0^2}).$$

It follows that,

$$Pr[|\bar{Y}(X_k^*) - y(X_k^*)| > \frac{\delta}{2}] \le Pr[\max_{x\in\tilde{\mathbb{S}_k}} |\bar{Y}(\boldsymbol{x}) - y(\boldsymbol{x})| > \frac{\delta}{2}] \le 2(k + N_0)\exp(-\frac{\delta^2 r_k}{8\sigma_0^2}).$$

$$Pr[|\bar{Y}(X_k^0) - y(X_k^0)| > \frac{\delta}{2}] \le Pr[\max_{x\in\tilde{\mathbb{S}_k}} |\bar{Y}(\boldsymbol{x}) - y(\boldsymbol{x})| > \frac{\delta}{2}] \le 2(k + N_0)\exp(-\frac{\delta^2 r_k}{8\sigma_0^2}).$$

To quantify the second term, we define set $A = \{|\bar{Y}(X_k^*) - y(X_k^*)| \le \frac{\delta}{5}\}$ and $B = \{|\bar{Y}(X_k^0) - y(X_k^0)| \le \frac{\delta}{5}\}$. we notice that

$$\begin{aligned}
&Pr[|y(X_k^*) - y(X_k^0)| > \frac{\delta}{2}] \\
=&Pr[\{|y(X_k^*) - y(X_k^0)| > \frac{\delta}{2}\} \cap \{A \cap B\}] + Pr[\{|y(X_k^*) - y(X_k^0)| > \frac{\delta}{2}\} \cap \{A \cap B\}^{\complement}],
\end{aligned}$$

where $\{A \cap B\}^{\complement}$ is the complement of $\{A \cap B\}$. The first probability is 0. This can be proved by contradiction. When $y(X_k^*) - y(X_k^0) \ge \frac{\delta}{2}$, as $|\bar{Y}(X_k^*) - y(X_k^*)| \le \frac{\delta}{5}$ (Set A) and $|\bar{Y}(X_k^0) - y(X_k^0)| \le \frac{\delta}{5}$ (Set B), it is easy to see that $\bar{Y}(X_k^*) > \bar{Y}(X_k^0)$. This contradict with the fact that X_k^* is the best observed point at iteration k, i.e., $X_k^* = \arg\min_{x\in\tilde{\mathbb{S}_k}} \bar{Y}(x)$. It follows that the first term is 0. Besides,

$$\begin{aligned}
&Pr[\{|y(X_k^*) - y(X_k^0)| > \frac{\delta}{2}\} \cap \{A \cap B\}^{\complement}] \\
<&Pr[\{A \cap B\}^{\complement}] = 1 - Pr[A \cap B] < 2 - Pr[A] - Pr[B] < 4(k + N_0)\exp(-\frac{\delta^2 r_k}{50\sigma_0^2}).
\end{aligned}$$

The last inequality follows that $1 - Pr[A] = Pr[|\bar{Y}(X_k^*) - y(X_k^*)| > \frac{\delta}{5}] < 2(k+N_0)\exp(-\frac{\delta^2 r_k}{50\sigma_0^2})$ (similar for $1 - Pr[B]$). Therefore, $Pr[|y(X_k^*) - y(X_k^0)| > \frac{\delta}{2}] < 4(k+N_0)\exp(-\frac{\delta^2 r_k}{50\sigma_0^2})$, and thus

$$Pr[|\bar{Y}(X_k^*) - y(X_k^0)| > \delta] < 2(k+N_0)\exp(-\frac{\delta^2 r_k}{8\sigma_0^2}) + 4(k+N_0)\exp(-\frac{\delta^2 r_k}{50\sigma_0^2}) < 6(k+N_0)\exp(-\frac{\delta^2 r_k}{50\sigma_0^2}).$$

Then, by Assumption 2,

$$\sum_{k=1}^{\infty} Pr[|\bar{Y}(X_k^*) - y(X_k^0)| > \delta] < 6\sum_{k=1}^{\infty}(k+N_0)\exp(-\frac{\delta^2 r_k}{50\sigma_0^2}) < \infty.$$

It follows that $\bar{Y}(X_k^*) \xrightarrow[k\to\infty]{} y(X_k^0)$, w.p.1 and that $y(X_k^*) \xrightarrow[k\to\infty]{} y(X_k^0)$, w.p.1. Since the function has only one global optimum, we have $X_k^* \xrightarrow[k\to\infty]{} X_k^0$ and therefore, $X_k^* \xrightarrow[k\to\infty]{} \mathbf{x}_k^*$. As a result, Theorem 1 holds. □

Theorem. ***3*** *[Convergence rate of Algorithm 1] Let us define* $C_k := 1 + \frac{\hat{\sigma}^2_{\xi,k}}{\hat{\sigma}^2_{\xi,k} + s^2_{\tilde{\pi}_k}(\mathbf{x}_{k+1})}$ *and* $\ell = \left(1 - \frac{1}{k}\right)^{1/d}$. *Given that EGO exhibits linear convergence, for any* $\varepsilon \geq 0$ *satisfying* $\ell + \varepsilon < 1$ *and as* $k \to \infty$, *the following holds for* $\mathbb{E}_k = E\left[||\mathbf{X}_{k+1}^* - \mathbf{x}^*||\right]$*:*

$$E[C_k] \geq \ell^{-2}, \quad \mathbb{E}_k = O\left(\left(E[C_k]^{-1/2}(\ell+\varepsilon)^{-1}\right)^{-k} E_k W_{k+1}^{-1/2}\right)$$

Proof. Convergence rates are shown for the EGO algorithm in (Bull, 2011). In his contribution, the author uses the Reproducing Kernel Hilbert Space $\mathcal{H}(X)$ of functions over the space $\mathbb{X}$ constructed from the kernel K and establishes the convergence rates of the loss function $L_k(T_{\pi_k}, \mathcal{H}_\theta(X), \rho) := \sup_{||y||_{\mathcal{H}_\theta(X)} \leq \rho} E_{\pi_k}\left[y(\mathbf{x}_k^*) - \hat{Y}\right]$ over the ball of radius ρ, β_ρ, in $\mathcal{H}(\mathbb{X})$ after k steps as (Theorem 2, page 2887, (Bull, 2011)):

$$L_k(T_{\pi_k}, \mathcal{H}_\theta(X), \rho) := \sup_{||y||_{\mathcal{H}_\theta(X)} \leq \rho} E_{\pi_k}\left[y(\mathbf{x}_k^*) - \hat{Y}(x) | \mathcal{F}_{\pi_k}\right] = O\left(k^{-1/d}\right). \tag{37}$$

As a result of (37), EGO exhibits *linear* convergence rates: $\lim_{k\to\infty} \frac{||\boldsymbol{x}_{k+1} - \boldsymbol{x}^*||}{||\boldsymbol{x}_k - \boldsymbol{x}^*||} = O\left(\left(1 - \frac{1}{k}\right)^{1/d}\right)$. From Theorem 2, we have that $sup_{\boldsymbol{x}\in\mathbb{X}} Pr\{||H(W_k, \boldsymbol{x}) - h(\boldsymbol{x})|| > \Delta\} = O\left(E_k W^{-2\alpha}\right)$ with $\alpha = 1/2$. From (18), we observe that the coefficient C_k for the geometric increase of the budget at each algorithm iteration satisfies $C_k \leq 2$ a.s. Since the budget increase is stochastic, we need to consider the expected coefficient $E[C_k]$ to verify that $E[C_k] \geq \ell^{-2}$, differently from Pasupathy et al. (2018). In the following, in order to simplify the notation, we will interpret W_k as $E_k W_k$.

We start analyzing the $E[c_k]$. First, let us re-write C_k as it follows:

$$C_k = 2 - s^2_{\tilde{\pi}}(\boldsymbol{x}_{k+1}) \frac{1}{\hat{\sigma}^2_{\xi,k}(\boldsymbol{x}) + s^2_{\tilde{\pi}_k}(\boldsymbol{x}_{k+1})} \tag{38}$$

Now, we will refer to the random variable Δ_k as:

$$\Delta_k = \frac{s^2_{\tilde{\pi}}\left(\boldsymbol{x}_{k+1}\right)}{\hat{\sigma}^2_{\xi,k}\left(\boldsymbol{x}\right) + s^2_{\tilde{\pi}_k}(\boldsymbol{x}_{k+1})} \tag{39}$$

The sample estimator of the variance satisfies $\hat{\sigma}^2_{\xi,k}\frac{W_k(\boldsymbol{x})-1}{\sigma^2_\epsilon} \sim \chi^2\left(W_{k+1}(\boldsymbol{x})-1\right)$, we can re-write Δ_k as:

$$\Delta_k = \frac{\frac{s^2_{\tilde{\pi}}(\boldsymbol{x}_{k+1})\cdot(W_k(\boldsymbol{x})-1)}{\sigma^2_\epsilon}}{\hat{\sigma}^2_{\xi,k}\frac{W_k(\mathbf{x})-1}{\sigma^2_\epsilon} + s^2_{\tilde{\pi}_k}(\boldsymbol{x}_{k+1})\frac{W_k(\boldsymbol{x})-1}{\sigma^2_\epsilon}} \tag{40}$$

Let us assume that σ^2_ϵ, characterizing the simulator noise, is known (which holds under assumption 6). Then the distribution associated to the random variable in (40) results:

$$\frac{1}{2^{\frac{W_k(\mathbf{x})-1}{2}}\Gamma\left(\frac{W_k(\boldsymbol{x})-1}{2}\right)} \cdot \frac{\sigma^2_\epsilon\left((\boldsymbol{x})\right)}{s_{\tilde{\pi}}\left(\boldsymbol{x}_{k+1}\right)W_k(\boldsymbol{x})} \cdot \left(\frac{1}{s_{\tilde{\pi}}\frac{W_k(\boldsymbol{x})}{\sigma^2_\epsilon(\boldsymbol{x})}\cdot t}\right)^{\frac{W_k(\boldsymbol{x})-1}{2}+1} \cdot \exp\left(-\frac{1}{s_{\tilde{\pi}}\frac{W_k(\boldsymbol{x})}{\sigma^2_\epsilon(\boldsymbol{x})}\cdot 2t}\right) \tag{41}$$

Operating a change of variable $b = t\cdot\left(W_k(\boldsymbol{x})-1\right)$, we obtain the following density:

$$\frac{1}{2^{\frac{W_k(\boldsymbol{x})-1}{2}}\Gamma\left(\frac{W_k(\boldsymbol{x})-1}{2}\right)} \cdot \frac{\sigma^2_\epsilon\left(\boldsymbol{x}\right)}{s_{\tilde{\pi}}\left(\boldsymbol{x}_{k+1}\right)W_k(\boldsymbol{x})} \cdot \left(\frac{W_k(\boldsymbol{x})-1}{s_{\tilde{\pi}}\left(\boldsymbol{x}_{k+1}\right)\frac{W_k(\boldsymbol{x})}{\sigma^2_\epsilon(\boldsymbol{x})}\cdot b}\right)^{\frac{W_k(\boldsymbol{x})-1}{2}+1} \cdot$$
$$\cdot \exp\left(-\frac{W_k(\mathbf{x})-1}{s_{\tilde{\pi}}\left(\boldsymbol{x}_{k+1}\right)\frac{W_k(\boldsymbol{x})}{\sigma^2_\epsilon(\boldsymbol{x})}\cdot 2b}\right) \cdot \frac{1}{W_k\left(\boldsymbol{x}\right)-1} \tag{42}$$

Expression (42) is a scaled inverse χ^2 density. In particular, let us define $\upsilon = W_k - 1$ and $\tau = \frac{1}{s_{\tilde{\pi}}(\boldsymbol{x}_{k+1})\frac{W_k(\boldsymbol{x})}{\sigma^2_\epsilon(\boldsymbol{x})}}$, then the expectation of Δ_k, results:

$$(40) = \frac{\upsilon^{\upsilon/2+1}\tau^{2\upsilon}}{(\upsilon+2)\left(\tau^2\upsilon\right)^{\upsilon/2}} = \frac{\upsilon^{\upsilon/2+1}\tau^{\upsilon}}{\left(\upsilon^{\upsilon/2+1}+2\upsilon^{\upsilon/2}\right)} = \frac{\sigma^2_\epsilon\left(\boldsymbol{x}\right)}{s_\pi\left(\boldsymbol{x}_{k+1}\right)} \cdot \frac{\upsilon^{\upsilon/2+1}}{\left(\upsilon^{\upsilon/2+1}+2\upsilon^{\upsilon/2}\right)(\upsilon+1)^{\upsilon}} \tag{43}$$

It can be observed that (43)$\to 0$ a.s. as $k\to\infty$, as a result $E\left[C_k\right]\to 2$. Since ℓ converges to 1 the asymptotic rate satisfies the condition $E[C_k]\geq \ell^{-2}$.

Using Theorems 1-2, we can use Theorem 6.6 page 58 in Pasupathy et al. (2018) to prove that the resulting convergence rate is:

$$E[C_k]\geq \ell^{-2}, \quad \mathbb{E}_k = O\left(\left(E[C_k]^{-1/2}\left(\ell+\varepsilon\right)^{-1}\right)^{-k} E_k W_{k+1}^{-1/2}\right)$$

Note that this result is better than the one obtained with basic geometric increase (i.e., $c_k = 1$), which would lead to a rate $O\left(E_k W_k^{-1/2}\right)$. □