Time Complexity Reduction in Efficient Global Optimization using Cluster Kriging

Hao Wang
LIACS, Leiden University
Niels Bohrweg 1
Leiden, Netherlands
h.wang@liacs.leidenuniv.nl

Bas van Stein
LIACS, Leiden University
Niels Bohrweg 1
Leiden, Netherlands
b.van.stein@liacs.leidenuniv.nl

Michael Emmerich
LIACS, Leiden University
Niels Bohrweg 1
Leiden, Netherlands
m.t.m.emmerich@liacs.leidenuniv.nl

Thomas Bäck
LIACS, Leiden University
Niels Bohrweg 1
Leiden, Netherlands
t.h.w.baeck@liacs.leidenuniv.nl

ABSTRACT

Efficient Global Optimization (EGO) is an effective method to optimize expensive black-box functions and utilizes Kriging models (or Gaussian process regression) trained on a relatively small design data set. In real-world applications, such as experimental optimization, where a large data set is available, the EGO algorithm becomes computationally infeasible due to the time and space complexity of Kriging. Recently, the so-called Cluster Kriging methods have been proposed to reduce such complexities for the big data, where data sets are clustered and Kriging models are built on each cluster. Furthermore, Kriging models are combined in an optimal way for the prediction. In addition, we analyze the Cluster Kriging landscape to adopt the existing infill-criteria, e.g., the expected improvement. The approach is tested on selected global optimization problems. It is shown by the empirical studies that this approach significantly reduces the CPU time of the EGO algorithm while maintaining the convergence rate of the algorithm.

CCS CONCEPTS

•**Theory of computation → Evolutionary algorithms; Gaussian processes;** *Divide and conquer;*

KEYWORDS

Efficient Global Optimization, Big-data, Kriging, Surrogate-assisted optimization

ACM Reference format:
Hao Wang, Bas van Stein, Michael Emmerich, and Thomas Bäck. 2017. Time Complexity Reduction in Efficient Global Optimization using Cluster Kriging. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071321

DOI: http://dx.doi.org/10.1145/3071178.3071321

1 INTRODUCTION

In many real-world optimization problems, such as optimizing the manufacturing of car body parts, function evaluations are costly, either in time or money. *Efficient Global Optimization* (EGO) [11] is a procedure designed to use a very low number of function evaluations while optimizing a specific function. The procedure uses a surrogate model to approximate the response surface of the real function. The surrogate model is fitted using an initial space filling *Design of Experiments* (DOE) [17]. Once the surrogate model is fitted on this data, optimization on the surrogate model's response surface can be performed to find good candidate solutions for the black-box function to be optimized. This step does not require any additional expensive function evaluations since it uses the surrogate model. For the selection of these candidate points, EGO uses an infill-criterion, which is meant to provide a nice balance between exploration and exploitation. The newly found candidate solution is then evaluated against the black-box function and added to the data set and used to re-fit the surrogate model. This procedure is repeated untill the convergence criteria are met.

EGO normally uses *Kriging* [20] as the surrogate model. Kriging, or *Gaussian Process Regression* is a popular regression model, capable of modeling very complex functions. Unfortunately, Kriging is not designed to be used on relatively large or high dimensional datasets due to its cubic time complexity and squared memory complexity. Real world problems often consist of many parameters and therefore require several hundreds or thousands of data points to provide a good model fitting. One of the main assumptions of EGO is that the black-box function evaluations are extremely costly. This assumption justifies the expensive time and space complexity of Kriging when data sets are relatively small. However, when the data available is more than a few thousand points, the time complexity of Kriging becomes a real bottleneck. In many complex real-world optimization problems, a lot of initial data is already available, using such a big data set for the standard EGO algorithm would be computationally infeasible while using only a subset of the initial data would result in poorly fitted Kriging models and therefor a poor optimization performance.

Contributions. Proposed is the use of a Kriging approximation algorithm, *Cluster Kriging* [28, 29], in the EGO procedure, to make EGO feasible on big data sets without losing its good convergence rate. Additionally, proposed are several modifications of the Model Tree Cluster Kriging (MTCK) algorithm to give an additional speedup, by making the model update only a part of its internal Kriging models when applying new points. In Section 2, related modifications and improvements to the EGO procedure are discussed and how they differ from the proposed solution. A more in-depth discussion of Efficient Global Optimization is given in Section 3 and the proposed algorithm and modifications are explained into detail in Section 4. Several experiments are conducted to compare different Kriging approximation techniques with the original EGO implementation using Ordinary Kriging. The experimental setup is explained in Section 5, with the empirical results shown. Both the convergence of the EGO procedures as well as the execution time are taken into account for the comparison. Finally, in Section 6, conclusions are drawn and future research steps are proposed.

2 RELATED RESEARCH

EGO has received quite some attention in the last several years, different variants of the original EGO algorithm are proposed, using different surrogate models or adapting the EGO algorithm to solve a specific set of problems more efficiently. In Viana *et al.* [30], multiple surrogate efficient global optimization is proposed, using multiple surrogate techniques in parallel to allow multiple candidate solutions per optimization run. Apart from the traditional Kriging model, additional surrogate models are used which use the uncertainty estimate of the Kriging model in combination with their predictions. Due to this inheritance of the uncertainty estimate, all kinds of different surrogate models can be used, which might be beneficial to both the time complexity as well as the convergence rate. However, a Kriging model or similar meta-model that provides the uncertainty estimate is still required. In Basudhar *et al.* [3], Support Vector Machines are used to apply EGO for constrained optimization problems. The main surrogate model used in the EGO procedure however, remains a Kriging model. Radial Basis Functions are also used in combination with EGO in Sóbester *et al.* [25], to perform parallel multi-point optimization.

Next to optimizing complex black-box functions, EGO is also recently used to optimize machine learning parameters. The first time that such an idea was proposed is in Bartz-Beielstein *et al.* [2], their sequential parameter optimization (SPO) toolbox uses EGO to analyze and optimize various algorithms' parameters. In continuation of this work, Hutter *et al.* proposes a time-bound SPO using an approximation of Kriging named projected process approximation [19] and later on Hutter uses the *Random Forest* model as the surrogate model for EGO. The Random Forest model is used primarily such that categorical parameters of the machine learning algorithms can be better optimized, since Random Forests support categorical parameters natively. However, to use Random Forests for the application of optimizing complex black-box functions would likely result in less fitted models and therefore slower convergence.

In addition to modifications of the EGO algorithm with different surrogate techniques and using EGO for different purposes, other work has been done in attempts to improve the traditional EGO algorithm. For example, an adaption to the EGO algorithm is proposed, in Kleijnen *et al.* [12], to improve the uncertainty estimate of the Kriging model using a Kriging bootstrapping approach. In another work, EGO is adapted to optimize stochastic black-box functions using an augmented expected improvement function [9].

This paper is focused on the time complexity reduction of EGO with Ordinary Kriging, and does not take the EGO modifications mentioned before into account since they can be easily combined with the proposed algorithm and its modifications. For later research, it would be interesting to compare EGO procedures designed to optimize machine learning parameters with those that are designed to optimize expensive black-box functions.

3 EFFICIENT GLOBAL OPTIMIZATION

The Efficient Global Optimization [11] or Bayesian optimization [14, 16] is a sequential model-based global optimization algorithm that is built on stochastic models over the unknown objective function. The Kriging modeling technique [13] is originally proposed as the underlying model in EGO. We shall briefly introduce the Kriging model and discuss its computational bottleneck.

3.1 Kriging

Kriging originates from spatial analysis/geostatistics and is widely used in Bayesian optimization and design and analysis of computer experiments (DACE) [22, 23]. As a nonparametric regression method, Kriging (or Gaussian process regression) models the distribution of an unknown function $f : \mathbb{R}^d \to \mathbb{R}$ by placing a **prior** stochastic process on it[1]. After evaluating the objective function at input points $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\} \subset \mathbb{R}^d$, the corresponding (noisy) observations: $\mathbf{y} = [y^{(1)}, y^{(2)}, \ldots, y^{(n)}]^\top$ are collected and used as a data set to update the prior distribution into a so-called **posterior** process, via Bayesian inference. Specifically, the mostly used variant of Kriging, *Ordinary Kriging* (OK), treats the unknown function f as the combination of a centered Gaussian Process ε (of zero mean) with an unknown constant trend term μ:

$$y(\mathbf{x}) = \overbrace{\mu + \varepsilon(\mathbf{x})}^{f} + \gamma(\mathbf{x}),$$
$$\varepsilon(\mathbf{x}) \sim \mathcal{N}(0, \sigma_\varepsilon^2(\mathbf{x})), \quad \gamma(\mathbf{x}) \sim \mathcal{N}(0, \sigma_\gamma^2)$$

Note that γ is the error variable in the regression which is known as the "nugget" effect in Kriging. In noiseless computer experiments, its variance is often set to a small number in order to relax the conditioning of the covariance matrix [1]. In this paper, it is assumed that noise term γ is homoscedastic and independent from each other and the Gaussian Process ε. The centered Gaussian Process ε is a stochastic process of zero mean and any finite collection of its random variables has a joint Gaussian distribution [20]. It can be completely specified by providing a covariance function $k(\cdot, \cdot)$: $\mathrm{Cov}[\varepsilon(\mathbf{x}), \varepsilon(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}')$. Throughout this paper, we choose the well-known Matérn 3/2 kernel function for k:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_\varepsilon^2 \left(1 + \sqrt{3}l\right) e^{-\sqrt{3}l}, \quad l = \sqrt{\sum_{i=1}^{d} \theta_i \left(x_i - x_i'\right)^2} \qquad (1)$$

[1]When such a prior stochastic process is assumed to be Gaussian, Kriging is equivalent to Gaussian Process Regression (GPR).

where θ_i's are the *hyper-parameters* of the model that are commonly chosen through the maximum likelihood principle. Using the Bayesian inference principle to estimate the unknown trend μ, the *posterior* distribution of y is obtained, that is again a Gaussian process[2]:

$$y \mid \mathcal{X}, \mathbf{y} \sim \boldsymbol{\mathcal{N}}\left(m(\mathbf{x}), s^2(\mathbf{x})\right) \tag{2}$$

The posterior mean function $m(\cdot)$ is the maximum a posteriori probability (MAP) estimate of the unknown f at $\mathbf{x}$ and the posterior $s^2(\cdot)$ measures the mean squared error (MSE) of the estimation. The prediction variance is of high importance for the development of the acquisition functions in EGO.

The Kriging model mainly suffers from the high time and space complexity when applied to large data sets. The major bottleneck is in the model fitting procedure: The covariance matrix Σ ($\Sigma_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$) of the input data $\mathcal{X}$ needs to be inverted to calculate the model likelihood value. Such operations take roughly $O(n^3)$ time complexity. Note that, such a high overhead is embedded in each iteration of the optimization procedure on the hyper-parameters, which renders the model fitting inapplicable for a large input data set $\mathcal{X}, \mathbf{y}$.

3.2 The Efficient Global Optimization Algorithm

EGO [11] is proposed to optimize expensive objective functions by sequentially choosing new candidate solutions from an underlying Kriging model. The candidate solutions are obtained by maximizing the so-called *acquisition function* or infill-criterion. Acquisition functions usually take the mean and variance of the posterior process (Eq. 2) into account, in order to balance the exploration and exploitation of the global search. Adding the newly obtained data points into the underlying Kriging model, its posterior process is modified and the acquisition function is updated accordingly. In this manner, a sequence of new solutions are generated iteratively. This algorithm is summarized in Algorithm 1. Many acquisition

Algorithm 1 Efficient Global Optimization

1 Generate the initial data set $\mathcal{X}, \mathbf{y}$
2 Fit the Kriging model hyper-parameters on the initial data set $\mathcal{X}, \mathbf{y}$.
3 **while** the stop criteria are not fulfilled **do**
4 Find global optimum of the infill criterion:

$$\mathbf{x}^* = argmax_{\mathbf{x}}\, \mathrm{EI}(\mathbf{x})$$

5 Evaluate $\mathbf{x}^*$: $y^* = y(\mathbf{x}^*)$ and append $\mathbf{x}^*, y^*$ to $\mathcal{X}, \mathbf{y}$.
6 Re-estimate the Kriging model hyper parameters
7 **end while**

functions have been proposed and investigated [10]. The most popular ones are: Lower Bound (LB) [5], the Probability of Improvement (PI) [15, 31] and Expected Improvement (EI) [11]. In this paper, we focus only on the expected improvement, that is defined as follows, in terms of minimization:

$$\begin{aligned}\mathrm{EI}(\mathbf{x}) &= \mathrm{E}[\max\{0, \min(\mathbf{y}) - y(\mathbf{x})\} \mid \mathbf{y}] \\ &= (\min(\mathbf{y}) - m(\mathbf{x}))\Phi\left(\frac{\min(\mathbf{y}) - m(\mathbf{x})}{s(\mathbf{x})}\right) \\ &+ s(\mathbf{x})\phi\left(\frac{\min(\mathbf{y}) - m(\mathbf{x})}{s(\mathbf{x})}\right)\end{aligned} \tag{3}$$

where $\Phi(\cdot), \phi(\cdot)$ denote the cumulative distribution function and the probability density function of the standard normal distribution, respectively. It takes into account the quantity of the expected improvement and also rewards a higher variance. In addition, the gradient of the expected improvement is given in the Equation 4, as it is required by the quasi-Newton optimization procedure, that is used in the next sections.

$$\begin{aligned}\nabla\, \mathrm{EI}(\mathbf{x}) &= \phi(u)\nabla s(\mathbf{x}) - \Phi(u)\nabla m(\mathbf{x}) \\ u &= \frac{\min(\mathbf{y}) - m(\mathbf{x})}{s(\mathbf{x})}\end{aligned} \tag{4}$$

4 CLUSTER KRIGING-BASED EGO

When applying the EGO algorithm to a large initial data set (e.g. in the experiment design), the CPU time spent on the hyper-parameter re-estimation becomes computationally infeasible. To relax this issue, it is proposed to use time complexity reduction techniques that have been developed for the Kriging model. In this paper, we adopt the so-called **Cluster Kriging** models that are proposed by van Stein and Wang et al [27–29].

4.1 Time complexity reduction for Kriging

Cluster Kriging employs a divide-and-conquer strategy that splits a huge data set into several small clusters. For each cluster, a Kriging model is built using only the data set $\mathcal{X}_i, \mathbf{y}_i$ of this clusters:

$$y \mid \mathcal{X}_i, \mathbf{y}_i \sim \boldsymbol{\mathcal{N}}\left(m_i(\mathbf{x}), s_i^2(\mathbf{x})\right), \quad i = 1, \ldots, q \tag{5}$$

Using q clusters on the data set, the time complexity for building the Kriging model above is $O(n^3/q^2)$, if all the clusters has roughly the same size. Compared to the original complexity $O(n^3)$ for Kriging, the reduction of CPU time will be significantly large in practice, if the number of clusters is large or proportional to n. Furthermore, the model fitting procedures for each Kriging model are independent such that they can also be **parallelized**, which leads to a time complexity $O(n^3/q^3)$.

The predictions are made by combining the predictions from all clusters in a smart way. When constructing such a model, mainly two modeling aspects should be considered: 1) which clustering algorithm to use? and 2) should the Kriging model built on each cluster be considered as local models, that is, not used for any prediction beyond its cluster boundary? Depending on these options, the following three variants are proposed as alternative models to be used in EGO.

Superposition of Kriging models. In this approach, the input data set is partitioned by hard clustering methods (K-means for instance). Due to the fact that there is no overlap between clusters, the Kriging models built on different clusters are considered independent stochastic processes. Consequently, a global posterior Gaussian process can be constructed by the superposition or linear combination

[2] It is possible to give the posterior covariance function. See [6] for the detail.

of the Kriging models built on each cluster:

$$y \mid \mathcal{X}, \mathbf{y} \sim \mathcal{N}\left(\sum_{i=1}^{q} w_i m_i(\mathbf{x}), \sum_{i=1}^{q} w_i^2 s_i^2(\mathbf{x})\right) \tag{6}$$

Note that $m_i(\cdot), s_i^2(\cdot)$ stand for the posterior mean and variance function of the Kriging model built on the i-th cluster. The mean function above is again used for the prediction. The combination weights are chosen by minimizing the variance of the global process [27]: $w_i = s_i^{-2}(\mathbf{x}) / \sum_{i=1}^{k} s_i^{-2}(\mathbf{x})$.

Mixture of Kriging models. As an alternative to the superposition of Kriging models, it is possible to construct the global process as the *mixture* of Kriging models from each of the clusters. In this manner, the posterior density function g of y is treated as a weighted combination of those on each cluster:

$$g(y \mid \mathcal{X}, \mathbf{y}) = \sum_{i=1}^{q} w_i \phi(y \mid \mathcal{X}_i, \mathbf{y}_i)$$

The combination weights can be specified as a user preference or obtained from a membership probability to each clusters when using fuzzy clustering methods, e.g. the Gaussian mixture models (GMM) [28]. In this approach, the global process after combination is no longer Gaussian and its mean and variance function are given as follows:

$$m(\mathbf{x}) = \sum_{i=1}^{q} w_i m_i(\mathbf{x}) \tag{7}$$

$$s^2(\mathbf{x}) = \sum_{i=1}^{q} w_i \left(s_i^2(\mathbf{x}) + m_i^2(\mathbf{x})\right) - \left(\sum_{i=1}^{q} w_i m_i(\mathbf{x})\right)^2 \tag{8}$$

The mean function is used for prediction as it is an unbiased estimator.

Tree-based local Kriging models. Another flavor of Cluster Kriging, called *Model Tree Cluster Kriging* (MTCK) [29], partitions the search space by building the **regression tree** [4]. After the clustering procedure, a local Kriging model is built for each *leaf node* of the tree. Unlike two previous approaches, the Kriging model for each cluster is considered as local. To predict the function value at a point $\mathbf{x}$, the cluster that $\mathbf{x}$ belongs to is determined first and only the Kriging model on this cluster is used for the prediction (Eq. 5)

In addition to the time complexity reduction, this approach also brings advantages in the model fitting. In the regression tree model, the search space is recursively divided into smaller hypercube, in an optimal way that reduces the variances on each node. The small variance in the data set makes the Kriging model fitting more numerically stable, as the covariance matrix tends to become singular when the data points varies abruptly. An example of the MTCK is illustrated on the 2-D Ackley function in Fig. 1. On the top row, the function landscape is shown on the left and the contours of MTCK mean function is in the middle, where each leaf node of the regression tree is depicted by the dashed boundaries. On the right subplot, the expected improvement function is drawn and it is clear that each leaf node contains a maximum of EI (marked by the red star symbol). Compared to the Ordinary Kriging model built on the same function (on the bottom row), although the mean function of MTCK resembles OK, the landscape of EI is multi-modal.

4.2 The algorithm

It is proposed to exploit the three Cluster Kriging variants in the EGO algorithm, for time complexity reduction. Although various complexity reduction (or approximation) methods exist for Kriging (for instance, FITC [18, 24] and Bayesian Committee Machines [26]), we state that Cluster Kriging is more suitable for the EGO algorithm for the following reasons.

Firstly, the Kriging models (posterior processes in Eq. 5) on each cluster can be executed in parallel, which yields an additional linear speedup in practice. Secondly, after a new candidate solution is found through the acquisition function, the hyper-parameters of Kriging needs to be re-estimated. Taking the cluster information into account, it is proposed to only re-estimate the Kriging models on the clusters that this new solution belongs to. This operation results in another linear speedup in the hyper-parameter re-estimation procedure, as in the best scenario, only one Kriging model is subject to re-fitting. Thirdly, the acquisition function, e.g. the expected improvement is still well-defined on Cluster Kriging because either the posterior process (Eq. 6) or at least the mean and variance function (Eq. 7) can be derived. The algorithm is presented in Alg. 2.

Algorithm 2 Cluster Kriging based Efficient Global Optimization (CK-EGO)

Input: Data set $\mathcal{X}, \mathbf{y}$ obtained on a black-box function f. The number of clusters q. The clustering method is chosen from K-means, GMM or regression trees by the user.

1: Initial Clustering: $\{\mathcal{X}_i, \mathbf{y}_i\}_{i=1}^{q} \leftarrow \mathcal{X}, \mathbf{y}$
2: Create the Kriging model for each cluster:
$$y \mid \mathcal{X}_i, \mathbf{y}_i \sim \mathcal{N}\left(m_i(\mathbf{x}), s_i^2(\mathbf{x})\right), \quad i = 1, \ldots, q$$
3: $c \leftarrow 0$
4: **while** the stop criteria are not fulfilled **do**
5:   $\mathbf{x}^* = argmax_{\mathbf{x}} \, \mathrm{EI}(\mathbf{x})$
6:   Evaluation: $y^* = f(\mathbf{x}^*)$
7:   $c \leftarrow c + 1$
8:   **if** $c >$ 10% the number of data points in $\mathcal{X}$ **then**
9:     Merge the data set: $\mathcal{X}, \mathbf{y} \leftarrow \{\mathcal{X}_i, \mathbf{y}_i\}_{i=1}^{q}$
10:     Clustering the data set $\mathcal{X}, \mathbf{y}$ and re-create the Kriging models for each cluster.
11:     $c \leftarrow 0$
12:   **else**
13:     **for** every cluster i that $\mathbf{x}^*$ belongs to **do**
14:       Append $\mathbf{x}^*, y^*$ to $\mathcal{X}_i, \mathbf{y}_i$.
15:       Re-estimate the hyper-parameter for the Kriging model on cluster i.
16:     **end for**
17:   **end if**
18: **end while**
19: **return** $\mathbf{x}^*$

In the algorithm, the initial fitting procedure can be parallelized (line 2). Usually, the cluster (and the Kriging model on it) that the new solution belongs to is updated (line 13-16). A counter c is incremented every time when a new candidate solution is generated

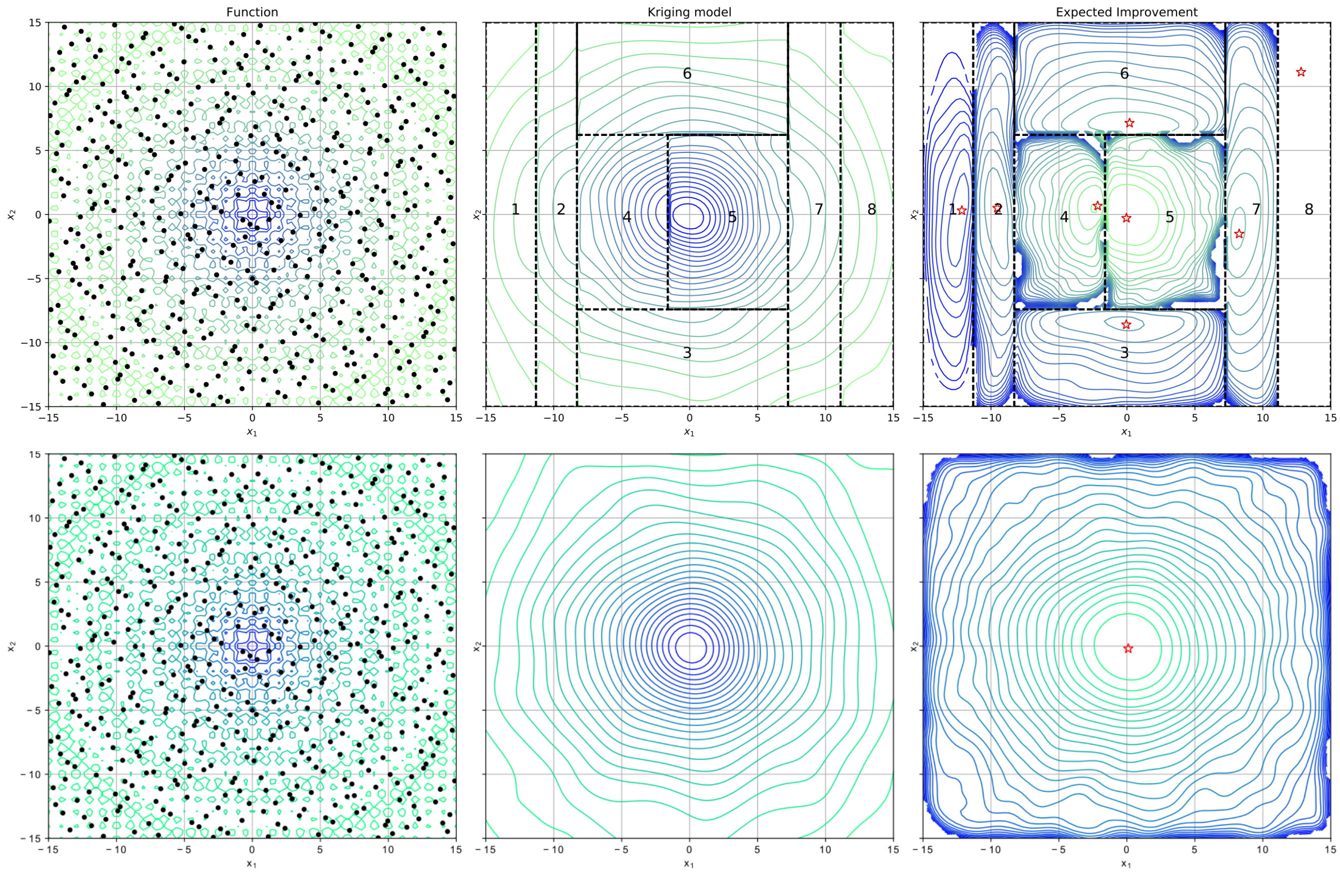


Figure 1: Comparison between Tree-based local Kriging models and Ordinary Kriging. *Top row*: Model tree Cluster Kriging and *bottom row*: Ordinary Kriging. *First column*: The landscape of the two dimensional Ackley function, *second column* is the contours of the model mean function, with the tree partitioning visualized by dashed lines for the MTCK model. The index of the clusters are shown in middle of each rectangle. *Third column*: The expected improvement function and the maximum point (red star) that is found by the quasi-Newton method. For the MTCK, multiple maximum points are obtained by conducting the quasi-Newton runs on each partition.

(line 5). If the c value, that is the recently appended data points, are more than 10% of the initial data set, the clustering is performed again to keep the size of each cluster balanced and capture the information contained in the newly added points.

4.3 Maximization of the Expected Improvement

For the maximization of the expected improvement (line 5 in Alg. 2), it is possible to exploit fast black-box optimization algorithms, for instance the well-known *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES) [7, 8], because the evaluation of the expected improvement is not expensive compared to the Kriging fitting procedure. However, as the formula of EI is well-expressed, its gradient is frequently utilized for the optimization. In this paper, it is proposed to optimize the expected improvement by first conducting a quasi-Newton method (using the gradient of EI) with random restarts and then exploiting the CMA-ES to optimize it again. The maximum of EI is chosen from the best runs from these two optimization algorithms.

To align with existing work [21] on using gradient-based optimization techniques for EI, we give the gradient of the mean and variance function in Cluster Kriging, as they are required by the computation of the EI gradient (Eq. 4). For the superposition of Kriging models (Eq. 6), the gradient of its mean and variance are:

$$\nabla m(\mathbf{x}) = \sum_{i=1}^{q} \left(w_i \nabla m_i + m_i \nabla w_i \right)$$

$$\nabla s^2(\mathbf{x}) = \sum_{i=1}^{q} \left(w_i^2 \nabla s_i^2 + 2 w_i s_i^2 \nabla w_i \right)$$

$$\nabla w_i = \sum_{i=1}^{q} \left(\frac{\nabla s_i^2}{s_i^4 M} + \frac{\sum_{i=1}^{q} \nabla s_i^2 / s_i^4}{s_i^2 M^2} \right), \; M = \sum_{i=1}^{q} \left(s_i^2 \right)^{-1}$$

The gradient of the Kriging model on each cluster $\nabla m_i, \nabla s_i^2$, is the usual gradient for Ordinary Kriging [21]. The gradient of the mixture of Kriging models can be obtained in a similar way. We omit this here for simplicity.

In addition, for the Tree-based local Kriging models (MTCK), it is shown (Fig. 1) that each cluster (leaf node) can be treated as a sub-problem in the EI maximization. Therefore, it is proposed to conduct independent optimization in each leaf node of the regression tree and choose the best point from all these sub-problems. In this manner, it is also possible to balance the search budget in each region of the search space such that a bigger leaf node will receive a high function evaluation budget.

5 EXPERIMENTS

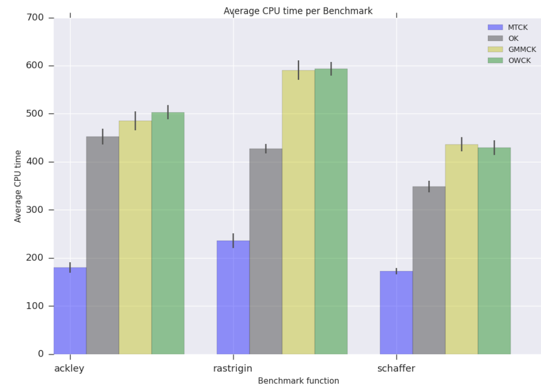
Several experiments are conducted to show both the empirical time complexity and convergence rate of the proposed Cluster Kriging based EGO, including all the variants of Cluster Kriging discussed in section 4. The performance of the proposed algorithm is compared to the original EGO that uses *Ordinary Kriging* (OK). For our experiments, the benchmark functions chosen are *Ackley*, *Rastrigin* and *Schaffer*. These functions are chosen because they are used often in optimization experiments, are highly multi modal, and are of a relatively high complexity.

Experiment 1 The algorithms compared are: EGO with Ordinary Kriging (OK), Tree-based local Kriging models (MTCK), Superposition of Kriging models (OWCK) and the mixture of Kriging models (GMMCK). Each of the Cluster Kriging variants uses 5 clusters. Both execution time and convergence rate are being measured with a fixed set of EGO iterations and optimization budget. The convergence is measured by taking the absolute error between the real optimum of the benchmark functions and the found optimum for each iteration of EGO. Each EGO run performs 10 iterations for the three benchmark functions in two dimensions. Three different initial sample sizes are used to train the surrogate models, 500, 1000 and 5000 points in order to illustrate the growth of CPU time required per algorithm, when the size of the data available increases. For each different experimental setup, the average time and distance to the optimum is recorded over 20 runs with different random seed.

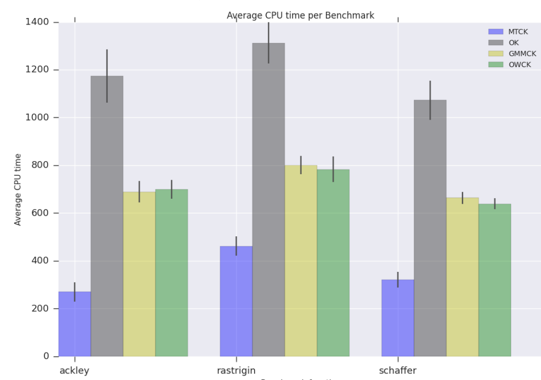
Experiment 2 The algorithms, OK, MTCK and OWCK are compared in five dimensions on the benchmark functions Ackley and Rastrigin also varying the algorithm that maximizes the expected improvement. CMA-ES and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm are compared.

Results From Figure 3 it can be observed that the Cluster Kriging based EGO variants perform very similar to OK, depending on the target function, a specific variant even outperforms Ordinary Kriging. Due to the relatively large variance in the results it is difficult to judge which algorithm performs better. However, from the CPU time in Figure 2 it can be observed that Cluster Kriging and in particular MTCK takes only a fragment of the time that Ordinary Kriging requires. Using a sample size of 500 points this difference is mainly due to the re-fitting of only one local model at a time. This can be seen by comparing MTCK with GMMCK and OWCK, since all three cluster Kriging variants use the same number of local models and only MTCK uses an adaptive local model strategy. When the number of points increases to 1.000 and even 5.000, the difference between the three cluster Kriging variants decreases but the difference with Ordinary Kriging becomes enormous. This shows that using EGO with Ordinary Kriging quickly becomes infeasible when the number of data points grow.

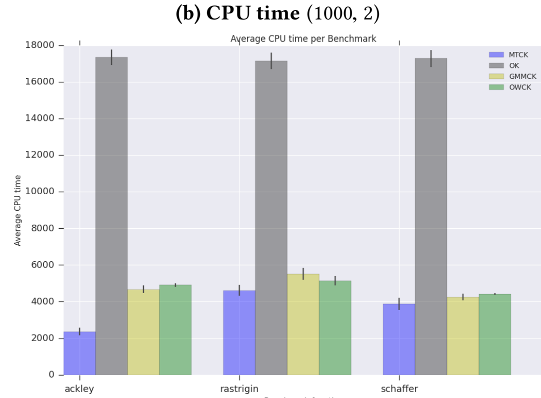
From Figure 4 it can be observed that also in higher dimensions Cluster Kriging does not under-perform Ordinary Kriging. In addition, it can be observed that using different optimization strategies



(a) CPU time (500, 2)

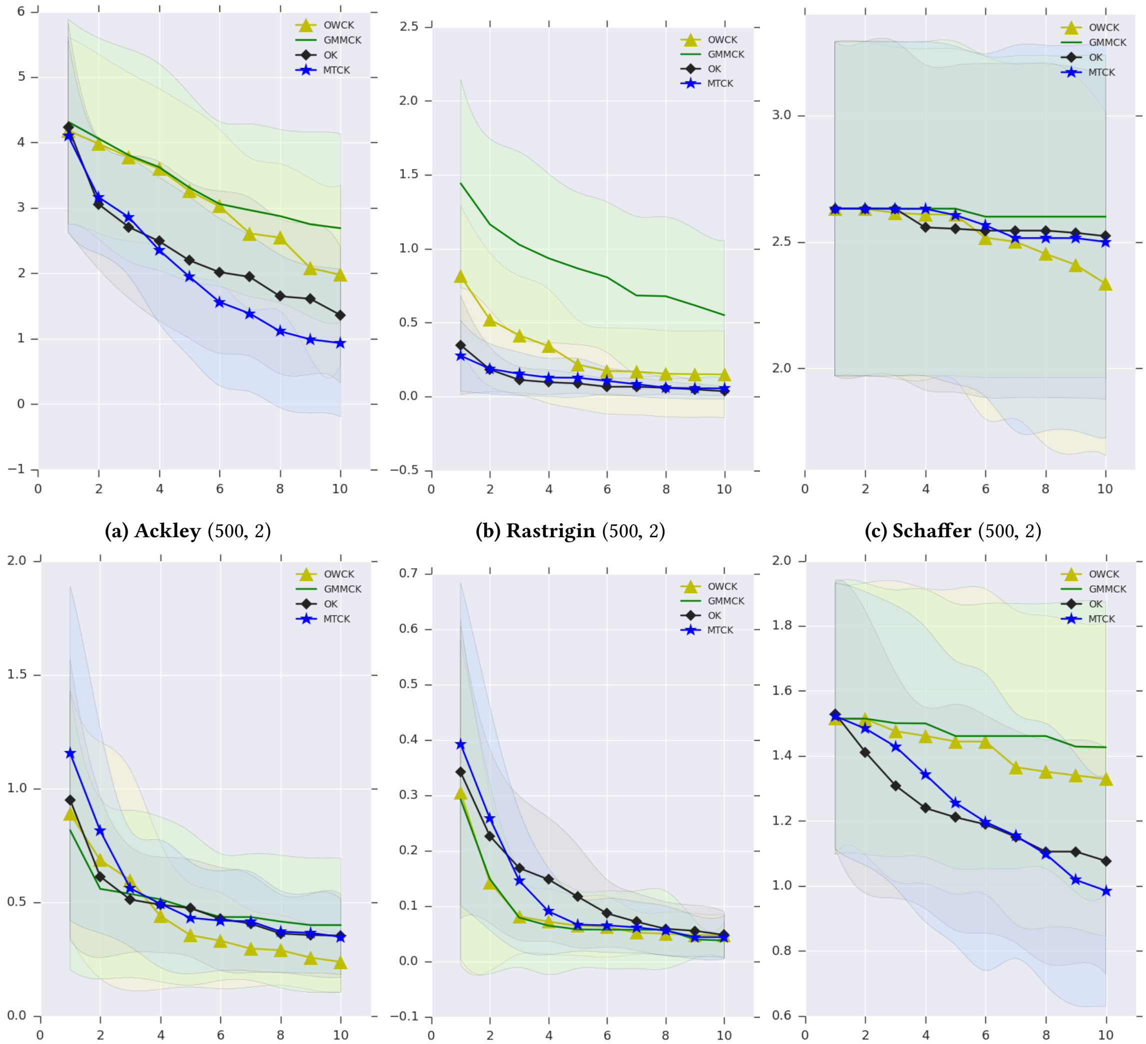


(b) CPU time (1000, 2)



(c) CPU time (5000, 2)

Figure 2: Average CPU time (in sec.) per benchmark function for varying sample sizes ($n_{samples}$, $d_{dimensions}$)**.**



(a) Ackley (500, 2) **(b) Rastrigin** (500, 2) **(c) Schaffer** (500, 2)

(d) Ackley (5000, 2) **(e) Rastrigin** (5000, 2) **(f) Schaffer** (5000, 2)

Figure 3: Average convergence of the absolute error of three benchmark functions in two dimensions, with varying training sample sizes n and 10 iterations of EGO. Shown is the average over 20 runs (lines) and one standard deviation (shaded areas).

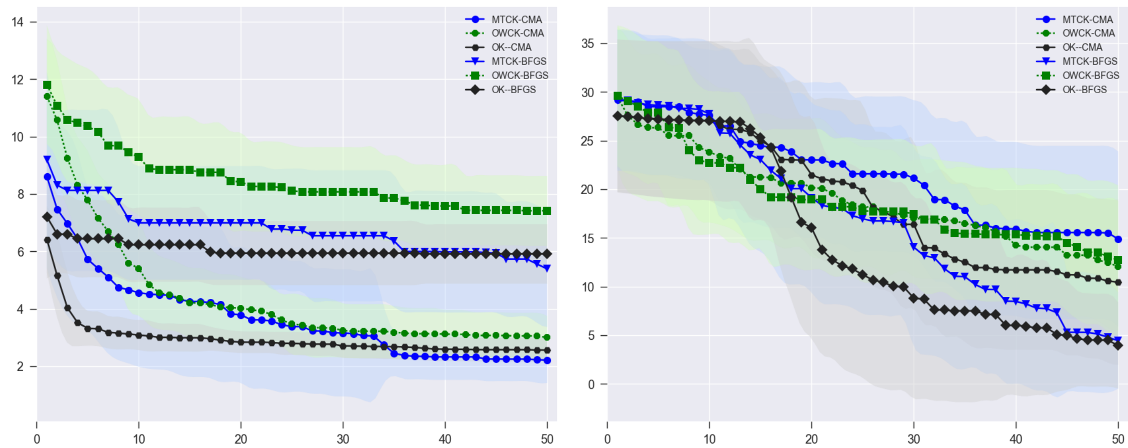
for the expected improvement affects the convergence rate. However, the best optimization strategy clearly depends on the target function.

6 CONCLUSION AND FUTURE RESEARCH

In this paper, we propose to relax the time complexity issue of the EGO algorithm by adopting a complexity reduction technique, the so-called Cluster Kriging as the surrogate model. In this approach, a collection of small Kriging models are created on the data clusters, which are obtained in the clustering method. In EGO, a global Kriging model is constructed by combining all the small Kriging models in a reasonable way. Three variants of the Cluster Kriging are proposed for the EGO and their performance is validated on some test functions.

Based on the empirical results that are shown in Section 5, it can be concluded that EGO using the Cluster Kriging is much faster in terms of time complexity compared to the traditional EGO that employs a Ordinary Kriging model. Moreover, each of the Cluster Kriging variants perform very well compared EGO using Ordinary Kriging in terms of convergence speed. From the results shown in Section 4, it can be inferred that the MTCK model fits the objective function well due to the reason that it captures local information much better than Ordinary Kriging.

For future research additional modifications can be proposed to Cluster Kriging based EGO algorithm to further optimize the time complexity and convergence speed. For instance, it is possible to utilize the tree partitioning information to calculate the bound of the acquisition function on each leaf node. In this way, some leaf nodes can be pruned from the search space. In addition to these



(a) Ackley (500, 5) **(b) Rastrigin** (500, 5)

Figure 4: Average convergence of the absolute error of two benchmark functions in five dimensions using different optimization algorithms. 500 **training samples and** 50 **iterations of EGO are used. Shown is the average over** 20 **runs (lines) and one standard deviation (shaded areas).**

modifications, it would be interesting to make a comparison to other alternatives currently used for optimizing machine learning algorithms to see how Cluster Kriging based EGO performs on these problems.

REFERENCES

[1] Ioannis Andrianakis and Peter G Challenor. 2012. The effect of the nugget on Gaussian process emulators of computer models. *Computational Statistics & Data Analysis* 56, 12 (2012), 4215–4228.
[2] Thomas Bartz-Beielstein and Sandor Markon. 2004. Tuning search algorithms for real-world applications: A regression tree based approach. In *Evolutionary Computation, 2004. CEC2004. Congress on*, Vol. 1. IEEE, 1111–1118.
[3] Anirban Basudhar, Christoph Dribusch, Sylvain Lacaze, and Samy Missoum. 2012. Constrained efficient global optimization with support vector machines. *Structural and Multidisciplinary Optimization* 46, 2 (2012), 201–221. DOI: http://dx.doi.org/10.1007/s00158-011-0745-5
[4] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees.* CRC press.
[5] JE Dennis and Virginia Torczon. 1997. Managing approximation models in optimization. *Multidisciplinary design optimization: State-of-the-art* (1997), 330–347.
[6] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. 2010. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems.* Springer, 131–162.
[7] Nikolaus Hansen. 2006. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation.* Springer, 75–102.
[8] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary computation* 9, 2 (2001), 159–195.
[9] D. Huang, 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 (2006), 441–466. DOI: http://dx.doi.org/10.1007/s10898-005-2454-3
[10] Donald R Jones. 2001. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization* 21, 4 (2001), 345–383.
[11] Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13, 4 (1998), 455–492.
[12] Jack P. C. Kleijnen, Wim van Beers, and Inneke van Nieuwenhuyse. 2012. Expected improvement in efficient global optimization through bootstrapped kriging. *Journal of Global Optimization* 54, 1 (2012), 59–73. DOI: http://dx.doi.org/10.1007/s10898-011-9741-y
[13] Daniel G. Krige. 1951. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa* 52, 6 (Dec. 1951), 119–139.
[14] J Močkus. 1975. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference.* Springer, 400–404.
[15] Jonas Mockus. 1994. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization* 4, 4 (1994), 347–365.
[16] Jonas Mockus. 2012. *Bayesian approach to global optimization: theory and applications.* Vol. 37. Springer Science & Business Media.
[17] Douglas C Montgomery. 1991. Design and analysis of experiments. (1991).
[18] Andrew Naish-Guzman and Sean Holden. 2007. The generalized FITC approximation. In *Advances in Neural Information Processing Systems.* 1057–1064.
[19] Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Christopher KI Williams. 2007. Approximation methods for gaussian process regression. *Large-scale kernel machines* (2007), 203–224.
[20] C.E. Rasmussen and C.K.I. Williams. 2006. *Gaussian Processes for Machine Learning.* University Press Group Limited.
[21] Olivier Roustant, David Ginsbourger, and Yves Deville. 2012. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. (2012).
[22] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. 1989. Design and Analysis of Computer Experiments. *Statist. Sci.* 4, 4 (1989), 409–423.
[23] T.J. Santner, B.J. Williams, and W. Notz. 2003. *The Design and Analysis of Computer Experiments.* Springer.
[24] Edward Snelson and Zoubin Ghahramani. 2005. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems.* 1257–1264.
[25] A. Sóbester, S.J. Leary, and A.J. Keane. 2004. A parallel updating scheme for approximating and optimizing high fidelity computer simulations. *Structural and Multidisciplinary Optimization* 27, 5 (2004), 371–383. DOI: http://dx.doi.org/10.1007/s00158-004-0397-9
[26] V Tresp. 2000. A Bayesian committee machine. *Neural computation* 12, 11 (2000), 2719–2741. DOI: http://dx.doi.org/10.1162/089976600300014908
[27] Bas van Stein, Hao Wang, Wojtek Kowalczyk, Thomas Bäck, and Michael Emmerich. 2015. *Optimally Weighted Cluster Kriging for Big Data Regression.* Springer International Publishing, Cham, 310–321. DOI: http://dx.doi.org/10.1007/978-3-319-24465-5_27
[28] Bas van Stein, Hao Wang, Wojtek Kowalczyk, Michael Emmerich, and Thomas Bäck. 2016. Fuzzy clustering for Optimally Weighted Cluster Kriging. In *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on.* IEEE, 939–945.
[29] Bas van Stein, Hao Wang, Wojtek Kowalczyk, Michael Emmerich, and Thomas Bäck. 2017. Cluster-based Kriging Approximation Algorithms for Complexity Reduction. *arXiv preprint arXiv:1702.01313* (2017).
[30] Felipe A. C. Viana, Raphael T. Haftka, and Layne T. Watson. 2013. Efficient global optimization algorithm assisted by multiple surrogate techniques. *Journal of Global Optimization* 56, 2 (2013), 669–689. DOI: http://dx.doi.org/10.1007/s10898-012-9892-5
[31] Antanas Žilinskas. 1992. A review of statistical models for global optimization. *Journal of Global Optimization* 2, 2 (1992), 145–153.