



HAL
open science

An improved approach to estimate the hyper-parameters of the kriging model for high-dimensional problems through the Partial Least Squares method

Mohamed Amine Bouhlef, Nathalie Bartoli, Abdelkader Otsmane, Joseph Morlier

► To cite this version:

Mohamed Amine Bouhlef, Nathalie Bartoli, Abdelkader Otsmane, Joseph Morlier. An improved approach to estimate the hyper-parameters of the kriging model for high-dimensional problems through the Partial Least Squares method. *Mathematical Problems in Engineering*, 2016, 2016, pp.1-11. 10.1155/2016/6723410 . hal-01342357

HAL Id: hal-01342357

<https://hal.science/hal-01342357v1>

Submitted on 5 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An improved approach to estimate the hyper-parameters of the kriging model for high-dimensional problems through the Partial Least Squares method

Mohamed Amine Bouhlel^a, Nathalie Bartoli^b, Abdelkader Otsmane^c,
Joseph Morlier^d

^a*University of Michigan, Department of Aerospace Engineering, Ann Arbor, MI, United States; bouhlel.mohamed.amine@gmail.com*

^b*ONERA, 2 avenue Édouard Belin, 31055 Toulouse, France; nathalie.bartoli@onera.fr*

^c*SNECMA, Rond-point René Ravaud-Réau, 77550 Moissy-Cramayel, France; abdelkader.otsmane@sneema.fr*

^d*Université de Toulouse, Institut Clément Ader, CNRS, ISAE-SUPAERO, 10 Avenue Edouard Belin, 31055 Toulouse Cedex 4, France; joseph.morlier@isae.fr*

Abstract

During the last years, the kriging model has become one of the most popular methods in computer simulation and machine learning. Many engineering applications use the kriging model to approximate the physical phenomena which is modeled by expensive simulation models, especially aerodynamic and structural models. When many input variables are used, the conventional kriging model is inefficient mainly due to an exorbitant computational time required during its construction. To handle high-dimensional problems (up to 100), one method is recently proposed that combines the kriging model with the Partial Least Squares technique, the so-called KPLS model. For a large number of design variables (100+), this method has shown interesting results in terms of saving CPU time required to build model while maintaining sufficient accuracy, on both academic and industrial problems. However, the KPLS model has provided a poor accuracy compared to the conventional kriging model on multimodal functions. To handle this issue, this paper proposes adding a new step during the construction of the KPLS model to improve its accuracy for multimodal functions. When the exponential covariance functions are used, this step is based on a simple identification between the covariance function of KPLS and kriging. The developed method

is validated especially by using a multimodal academic function, known as Griewank function in the literature, and we show the gain in terms of accuracy and computer time by comparing with KPLS and ordinary kriging.

Keywords: Kriging, KPLS, Partial Least Squares, Principal components, Exponential covariance kernels

Symbols and notation

Matrices and vectors are in bold type.

<u>Symbol</u>	<u>Meaning</u>
$ \cdot $	absolute value
\mathbb{R}	set of real number
\mathbb{R}^+	set of positive real number
n	number of sampling points
d	dimension
h	number of principal components retained
\mathbf{x}	a $1 \times d$ vector
x_i	the i^{th} element of a vector \mathbf{x}
\mathbf{X}	a $n \times d$ matrix containing sampling points
$y(\mathbf{x})$	the true function y performed on the vector \mathbf{x}
\mathbf{y}	a $n \times 1$ vector containing simulation of \mathbf{X}
$\hat{y}(\mathbf{x})$	the prediction of the true function $y(\mathbf{x})$
$Y(\mathbf{x})$	a stochastic process
$\mathbf{x}^{(i)}$	the i^{th} training point for $i = 1, \dots, n$ (a $1 \times d$ vector)
$\mathbf{w}^{(l)}$	a $d \times 1$ vector containing X-weights given by the l^{th} PLS-iteration for $l = 1, \dots, h$
$\mathbf{X}^{(0)}$	\mathbf{X}

<u>Symbol</u>	<u>Meaning</u>
$\mathbf{X}^{(l-1)}$	Matrix containing residual of the inner regression of the $(l - 1)^{th}$ PLS-iteration for $l = 1, \dots, h$
$k(\cdot, \cdot)$	a covariance function
\mathbf{x}^t	Superscript t denotes the transpose operation of the vector \mathbf{x}
\approx	approximately sign

1. Introduction

During the last years, the kriging model [1, 2, 3, 4], which is referred to as the Gaussian process model [5], has become one of the most popular methods in computer simulation and machine learning. It is used as a substitute of high-fidelity codes representing a physical phenomena and aims to reduce the computational time of a particular process. For instance, the kriging model is used successfully in several optimization problems [6, 7, 8, 9, 10, 11]. Kriging is not well adapted to high-dimensional problem, principally due to large matrix inversion problems. In fact, the kriging model becomes much time consuming when a large number of input variables are used since a large number of sampling points are required. Indeed, it is recommended in [12] to use $10d$ sampling points, with d the number of dimension, for obtaining a good accuracy of the kriging model. As a result, we need to increase the size of the kriging covariance matrix which becomes computationally very expensive to invert. Moreover, this inversion's problem induces difficulty in the classical hyper-parameters estimation through the maximization of the likelihood function.

A recent method, called KPLS [13], is developed to reduce computational time which uses, during a construction of the kriging model, the dimensional reduction method "Partial Least Squares" (PLS). This method is able to reduce the number of hyper-parameters of a kriging model, such that their number becomes equal to the number of principal components retained by the PLS method. The KPLS method is thus able to rapidly build a kriging model for high-dimensional problems (100+) while maintaining a good accuracy. However, it has been shown in [13] that the KPLS model is less accurate than the kriging model in many cases, in particular for multimodal functions.

In this paper, we propose an extra step that supplements [13] in order to improve its accuracy. Under hypothesis that kernels used for building the KPLS model are of exponential type with the same form (all Gaussian kernels for instance), we choose the hyper-parameters found by the KPLS model as an initial point to optimize the likelihood function of a conventional kriging model. In fact, this approach is performed by identifying the covariance function of the KPLS model as a covariance function of a kriging model. The fact of considering the identified kriging model, instead of the KPLS model, leads to extend the search space where the hyper-parameters are defined, and thus to make the resulting model more flexible than the KPLS model.

This paper is organized in 3 main sections. In section 2, we present a review of the KPLS model. In section 3, we discuss our new approach under the hypothesis needed for its applicability. Finally, numerical results are shown to confirm the efficiency of our method followed by a summary of what we have achieved.

2. Construction of KPLS

In this section, we introduce the notation and describe the theory behind the construction of the KPLS model. Assume that we have evaluated a cost deterministic function of n points $\mathbf{x}^{(i)}$ ($i = 1, \dots, n$) with $\mathbf{x}^{(i)} = [x_1^{(i)}, \dots, x_d^{(i)}] \in B \subset \mathbb{R}^d$, and we denote \mathbf{X} by the matrix $[\mathbf{x}^{(1)t}, \dots, \mathbf{x}^{(n)t}]^t$. For simplicity, B is considered to be a hypercube expressed by the product between intervals of each direction space, i.e., $B = \prod_{j=1}^d [a_j, b_j]$, where $a_j, b_j \in \mathbb{R}$ with $a_j \leq b_j$ for $j = 1, \dots, d$. Simulating these n inputs gives the outputs $\mathbf{y} = [y^{(1)}, \dots, y^{(n)}]^t$ with $y^{(i)} = y(\mathbf{x}^{(i)})$, for $i = 1, \dots, n$.

2.1. Construction of the kriging model

For building the kriging model, we assume that the deterministic response $y(\mathbf{x})$ is a realization of a stochastic process [14, 15, 16, 17]:

$$Y(\mathbf{x}) = \beta_0 + Z(\mathbf{x}). \quad (1)$$

The presented formula, with β_0 an unknown constant, corresponds to ordinary kriging [8] which is a particular case of universal kriging [15]. The

stochastic term $Z(\mathbf{x})$ is considered as a realization of a stationary Gaussian process with $\mathbb{E}[Z(\mathbf{x})] = 0$ and a covariance function, also called kernel function, given by:

$$\text{Cov}(Z(\mathbf{x}), Z(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') = \sigma^2 r(\mathbf{x}, \mathbf{x}') = \sigma^2 r_{\mathbf{x}\mathbf{x}'}, \quad \forall \mathbf{x}, \mathbf{x}' \in B, \quad (2)$$

where σ^2 is the process variance and $r_{\mathbf{x}\mathbf{x}'}$ is the correlation function between \mathbf{x} and \mathbf{x}' . However, the correlation function r depends on hyper-parameters θ which are considered to be known. We also denote the $n \times 1$ vector as $\mathbf{r}_{\mathbf{x}\mathbf{X}} = [r_{\mathbf{x}\mathbf{x}^{(1)}}, \dots, r_{\mathbf{x}\mathbf{x}^{(n)}}]^t$ and the $n \times n$ correlation matrix as $\mathbf{R} = [\mathbf{r}_{\mathbf{x}^{(1)}\mathbf{X}}, \dots, \mathbf{r}_{\mathbf{x}^{(n)}\mathbf{X}}]$. We use $\hat{y}(\mathbf{x})$ to denote the prediction of the true function $y(\mathbf{x})$. Under the hypothesis above, the best linear unbiased predictor for $y(\mathbf{x})$, given the observations \mathbf{y} , is:

$$\hat{y}(\mathbf{x}) = \hat{\beta}_0 + \mathbf{r}_{\mathbf{x}\mathbf{X}}^t \mathbf{R}^{-1} (\mathbf{y} - \hat{\beta}_0 \mathbf{1}), \quad (3)$$

where $\mathbf{1}$ denotes an n -vector of ones and:

$$\hat{\beta}_0 = (\mathbf{1}^t \mathbf{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^t \mathbf{R}^{-1} \mathbf{y}. \quad (4)$$

In addition, the estimation of σ^2 is given by:

$$\hat{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{1} \hat{\beta}_0)^t \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1} \hat{\beta}_0). \quad (5)$$

Moreover, ordinary kriging provides an estimate of the variance of the prediction, which is given by:

$$s^2(\mathbf{x}) = \hat{\sigma}^2 (1 - \mathbf{r}_{\mathbf{x}\mathbf{X}}^t \mathbf{R}^{-1} \mathbf{r}_{\mathbf{x}\mathbf{X}}). \quad (6)$$

Note that the assumption of a known covariance function with known parameters θ is unrealistic in reality and they are often unknown. For this reason, the covariance function is typically chosen from among a parametric family of kernels. In this work, only the covariance functions of exponential type are considered, in particular the Gaussian kernel. Indeed, the Gaussian kernel is the most popular kernel in kriging metamodels of simulation models, which is given by:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{i=1}^d \exp\left(-\theta_i (x_i - x'_i)^2\right), \quad \forall \theta_i \in \mathbb{R}^+. \quad (7)$$

We note that the parameters θ_i , for $i = 1, \dots, d$, can be interpreted as measuring how strongly the input variables x_1, \dots, x_d , respectively, affect the output y . If θ_i is very large, the kernel $k(\mathbf{x}, \mathbf{x}')$ given by equation (7) tends to zero and thus leads to a low correlation. In fact, we see in figure 1 how the correlation curve rapidly varies from a point to another when $\theta = 10$.

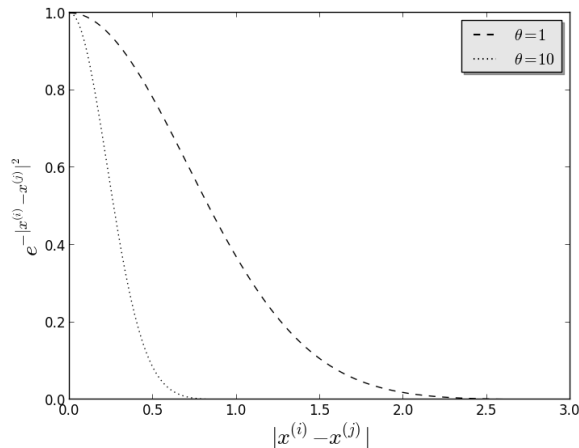


Figure 1: Theta smoothness can be tuned to adapt spatial influence to our problem. The magnitude of θ dictates how quickly the squared exponential function variates.

However, the estimator of the kriging parameters ($\hat{\beta}_0$, $\hat{\sigma}^2$ and $\theta_1, \dots, \theta_d$) makes the kriging predictor, given by equation (3), nonlinear and makes the estimated predictor variance, given by equation (6), biased. We note that the vector \mathbf{r} and the matrix \mathbf{R} should get hats above but it is ignored in practice [18].

2.2. Partial Least Squares

The PLS method is a statistical method which searches out the best multidimensional direction \mathbf{X} that explains the characteristics of the output \mathbf{y} . It finds a linear relationship between input variables and output variable by projecting input variables onto principal components, also called latent variables. The PLS technique reduces dimension and reveals how inputs depend on output. In the following, we use h to denote the number of principal components retained which are a lot lower than d ($h \ll d$); h does not generally exceed 4, in practice. In addition, the principal components can be computed sequentially. In fact, the principal component $\mathbf{t}^{(l)}$, for

$l = 1, \dots, h$, is computed by seeking the best direction $\mathbf{w}^{(l)}$ which maximizes the squared covariance between $\mathbf{t}^{(l)} = \mathbf{X}^{(l-1)}\mathbf{w}^{(l)}$ and $\mathbf{y}^{(l-1)}$:

$$\mathbf{w}^{(l)} = \begin{cases} \operatorname{argmax}_{\mathbf{w}^{(l)}} \mathbf{w}^{(l)t} \mathbf{X}^{(l-1)t} \mathbf{y}^{(l-1)} \mathbf{y}^{(l-1)t} \mathbf{X}^{(l-1)} \mathbf{w}^{(l)} \\ \text{such that } \mathbf{w}^{(l)t} \mathbf{w}^{(l)} = 1. \end{cases} \quad (8)$$

where $\mathbf{X} = \mathbf{X}^{(0)}$, $\mathbf{y} = \mathbf{y}^{(0)}$, and, for $l = 1, \dots, h$, $\mathbf{X}^{(l)}$ and $\mathbf{y}^{(l)}$ are the residual matrix from the local regression of $\mathbf{X}^{(l-1)}$ onto the principal component $\mathbf{t}^{(l)}$ and from the local regression of $\mathbf{y}^{(l-1)}$ onto the principal component $\mathbf{t}^{(l)}$, respectively, such that:

$$\begin{aligned} \mathbf{X}^{(l)} &= \mathbf{X}^{(l-1)} - \mathbf{t}^{(l)} \mathbf{p}^{(l)}, \\ \mathbf{y}^{(l)} &= \mathbf{y}^{(l-1)} - c_l \mathbf{t}^{(l)}, \end{aligned} \quad (9)$$

where $\mathbf{p}^{(l)}$ (a $1 \times d$ vector) and c_l (a coefficient) contain the regression coefficients. For more details of how PLS method works, please see [19, 20, 21].

The principal components represent the new coordinate system obtained upon rotating the original system with axes, x_1, \dots, x_d [21]. For $l = 1, \dots, h$, $\mathbf{t}^{(l)}$ can be written as:

$$\mathbf{t}^{(l)} = \mathbf{X}^{(l-1)} \mathbf{w}^{(l)} = \mathbf{X} \mathbf{w}_*^{(l)}. \quad (10)$$

This important relationship is mainly used for developing the KPLS model which is detailed in section 2.3. The vectors $\mathbf{w}_*^{(l)}$, for $l = 1, \dots, h$, are given by the following matrix $\mathbf{W}_* = [\mathbf{w}_*^{(1)}, \dots, \mathbf{w}_*^{(h)}]$ which is obtained by (for more details, see [22]):

$$\mathbf{W}_* = \mathbf{W} (\mathbf{P}^t \mathbf{W})^{-1},$$

where $\mathbf{W} = [\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(h)}]$ and $\mathbf{P} = [\mathbf{p}^{(1)t}, \dots, \mathbf{p}^{(h)t}]$.

2.3. Construction of the KPLS model

The hyper-parameters $\theta = \{\theta_i\}$, for $i = 1, \dots, d$, given by equation (7) are found by maximum likelihood estimation (MLE) method. Their estimation becomes more and more expensive when d increases. The vector θ can be interpreted as measuring how strongly the variables x_1, \dots, x_d affect the output y , respectively. For building KPLS, coefficients given by vectors $\mathbf{w}_*^{(l)}$ will be considered as a measuring of the influence of the input variables

x_1, \dots, x_d on the output y . By some elementary operations on the kernel functions, we define the KPLS kernel by:

$$k_{kpls1:h}(\mathbf{x}, \mathbf{x}') = \prod_{l=1}^h k_l(F_l(\mathbf{x}), F_l(\mathbf{x}')), \quad (11)$$

where $k_l : B \times B \rightarrow \mathbb{R}$ is an isotropic stationary kernel and

$$\begin{aligned} F_l : B &\longrightarrow B \\ \mathbf{x} &\longmapsto \left[w_{*1}^{(l)} x_1, \dots, w_{*d}^{(l)} x_d \right]. \end{aligned} \quad (12)$$

More details of such construction are given in [13]. Considering the example of the Gaussian kernel given by the equation (7), we obtain:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{l=1}^h \prod_{i=1}^d \exp \left[-\theta_l \left(w_{*i}^{(l)} x_i - w_{*i}^{(l)} x'_i \right)^2 \right], \quad \forall \theta_l \in \mathbb{R}^+. \quad (13)$$

Since a small number of principal components are retained, the estimation of the hyper-parameters $\theta_1, \dots, \theta_h$ is faster than the hyper-parameters $\theta_1, \dots, \theta_d$ given by the equation (7), where d is very high (100+).

3. Transition from the KPLS model to the kriging model using the exponential covariance functions

In this section, we show that if all kernels k_l , for $l = 1, \dots, h$, used in equation (11) are of the exponential type with the same form (all Gaussian kernels for instance), then the kernel $k_{kpls1:h}$ given by equation (11) will be of the exponential type with the same form as k_l (Gaussian if all k_l are Gaussian).

3.1. *Proof of the equivalence between the kernels of the KPLS model and the kriging model*

Let us define, for $i = 1, \dots, d$, $\eta_i = \sum_{l=1}^h \theta_l w_{*i}^{(l)2}$, we have:

$$\begin{aligned}
 k_{1:h}(\mathbf{x}, \mathbf{x}') &= \prod_{l=1}^h \prod_{i=1}^d \exp\left(-\theta_l w_{*i}^{(l)2} (x_i - x'_i)^2\right) \\
 &= \exp\left(\sum_{i=1}^d \sum_{l=1}^h -\theta_l w_{*i}^{(l)2} (x_i - x'_i)^2\right) \\
 &= \exp\left(\sum_{i=1}^d -\eta_i (x_i - x'_i)^2\right) \\
 &= \prod_{i=1}^d \exp(-\eta_i (x_i - x'_i)^2).
 \end{aligned} \tag{14}$$

In the same way, we can show this equivalence for the other exponential kernels where $p_1 = \dots = p_h$:

$$k_{1:h}(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{l=1}^h \prod_{i=1}^d \exp\left(-\theta_l \left|w_{*i}^{(l)} (x_i - x'_i)\right|^{p_l}\right). \tag{15}$$

However, we must caution that the above proof shows an equivalence between the covariance functions of KPLS and kriging only on a subspace domain. More precisely, the KPLS covariance function is defined in a subspace from \mathbb{R}^{+d} whereas the kriging covariance function is defined in the complete \mathbb{R}^{+d} domain. Thus, our original idea is to extend the space where the KPLS covariance function is defined for the complete space \mathbb{R}^{+d} .

3.2. *A new step during the construction of the KPLS model: KPLS+K*

By considering the equivalence shown in the last section, we propose to add a new step during the construction of the KPLS model. This step occurs just after the θ_l -estimation, for $l = 1, \dots, h$. It involves making a local optimization of the likelihood function of the kriging model equivalent to the KPLS model. Moreover, we use $\eta_i = \sum_{l=1}^h \theta_l w_{*i}^{(l)2}$, for $i = 1, \dots, d$, as a starting point of the local optimization by considering the solution θ_l , for $l = 1, \dots, h$, found by the KPLS method. Thus, this optimization is done in the complete space where the vector $\eta = \{\eta_i\} \in \mathbb{R}^{+d}$.

This approach, called KPLS+K, aims to improve the MLE of the kriging model equivalent to the associated KPLS model. In fact, the local optimization of the equivalent kriging offers more possibilities for improving the MLE, by considering a wider search space, and thus it will be able to correct the estimation of many directions. These directions are represented by η_i for the i^{th} direction which is badly estimated by the KPLS method. Because estimating the equivalent kriging hyper-parameters can be time consuming, especially when d is large, we improve the MLE by a local optimization at the cost of a slight increase of computational time.

Figure 2 recalls the principal stages of building a KPLS+K model.

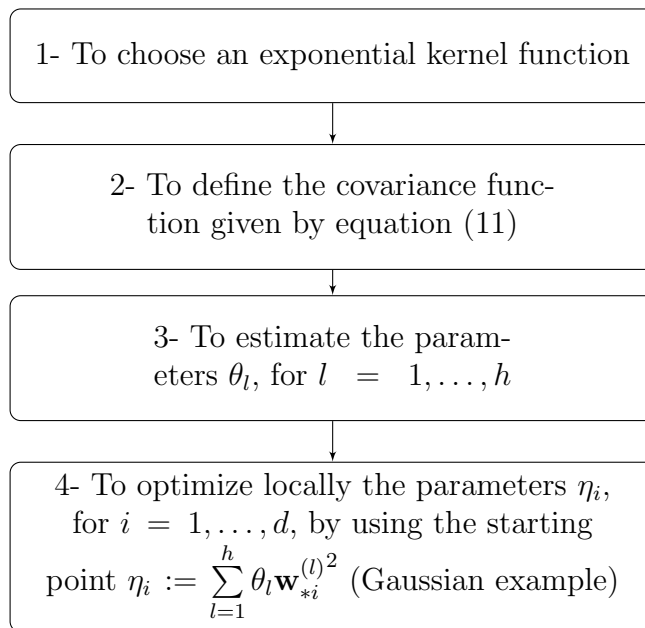


Figure 2: Principal stages for building a KPLS+K model.

4. Numerical simulations

We now focus on the performance of KPLS+K by comparing it with the KPLS model and the ordinary kriging model. For this purpose, we use the academic function, named Griewank, over the interval $[-5, 5]$ which is studied in [13]. 20 and 60 dimensions are considered for this function. In addition, an engineering example, done at Snecma for a multidisciplinary

optimization, is used. This engineering case is chosen since it was shown in [13] that KPLS is less accurate than ordinary kriging. The Gaussian kernel is used for all surrogate models used herein, i.e., ordinary kriging, KPLS and KPLS+K. For KPLS and KPLS+K using h principal components, for $h \leq d$, will be denoted by $\text{KPLS}h$ and $\text{KPLS}h+\text{K}$, respectively, and this h is varied from 1 to 3. The Python toolbox Scikit-learn v.0.14 [23] is used to achieve the proposed numerical tests, except for ordinary kriging used on the industrial case, where the Optimus version is used. The training and the validation points used in [13] are reused in the following.

4.1. Griewank function over the interval $[-5, 5]$

The Griewank function [24, 13] is defined by:

$$y_{\text{Griewank}}(\mathbf{x}) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \\ -5 \leq x_i \leq 5, \text{ for } i = 1, \dots, d.$$

Figure 3 shows the degree of complexity of such function which is very multimodal. As in [13], we consider $d = 20$ and $d = 60$ input variables. For each

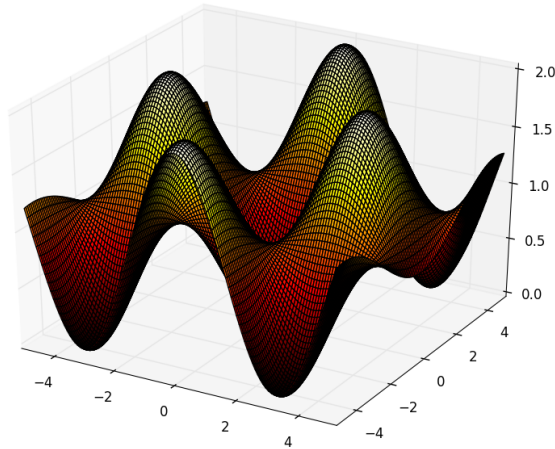


Figure 3: A 2D Griewank function over the interval $[-5, 5]$.

problem, ten experiments based on the random Latin hypercube design are built with n (number of sampling points) equals to 50, 100, 200, and 300.

To better visualize the results, boxplots are used to show the CPU time and the relative error RE given by

$$\text{Error} = \frac{\|\hat{\mathbf{Y}} - \mathbf{Y}\|_2}{\|\mathbf{Y}\|_2} 100, \quad (16)$$

where $\|\cdot\|_2$ represents the usual L^2 norm, $\hat{\mathbf{Y}}$ and \mathbf{Y} are the vectors containing the prediction and the real values of 5000 randomly selected validation points for each case. The mean and the standard error are given in tables A.2 and A.3, respectively, in appendix A. However, the results of the ordinary kriging model and the KPLS model are reported from [13].

For 20 input variables and 50 sampling points, the KPLS models always give a more accurate solution than ordinary kriging and KPLS+K, as shown in figure 4a. Indeed, the best result is given by KPLS3 with a mean of RE equals to 0.51%. However, the KPLS+K models give more accurate models than ordinary kriging in this case (0.58% for KPLS2+K and KPLS3+K vs 0.62% for ordinary kriging). For the KPLS model, the rate of improvement with respect to the number of sampling points is less than for ordinary kriging and KPLS+K (see figures 4b–4d). As a result, KPLS h +K, for $h = 1, \dots, 3$, and ordinary kriging give almost the same accuracy ($\approx 0.16\%$) when 300 sampling points are used (as shown in figure 4d), whereas the KPLS models give a RE of 0.35% as a best result, when $h = 3$.

Nevertheless, the results shown in figure 5 indicate that the KPLS+K models lead to an important reduction in CPU time for the various number of sampling points compared to ordinary kriging. For instance, 20.49s are required for building KPLS3+K when 300 training points are used, whereas ordinary kriging is built in 94.31s; in this case, KPLS3+K is thus approximately 4 times cheaper than the ordinary kriging model. Moreover, the computational time required for building KPLS+K is more stable than the computational time for building ordinary kriging; a standard deviation of approximately 3s for KPLS+K and 22s for the ordinary kriging model are observed.

For 60 input variables and 50 sampling points, a slight difference of the results occurs compared to the 20 input variables case (figure 6a). Indeed, the KPLS models remain always better, with a mean of RE approximately equals to 0.92%, than KPLS+K and ordinary kriging. However, the KPLS+K models give more accurate results than ordinary kriging with an accuracy close to that of KPLS ($\approx 0.99\%$ vs 1.39%). Increasing the number of sampling points,

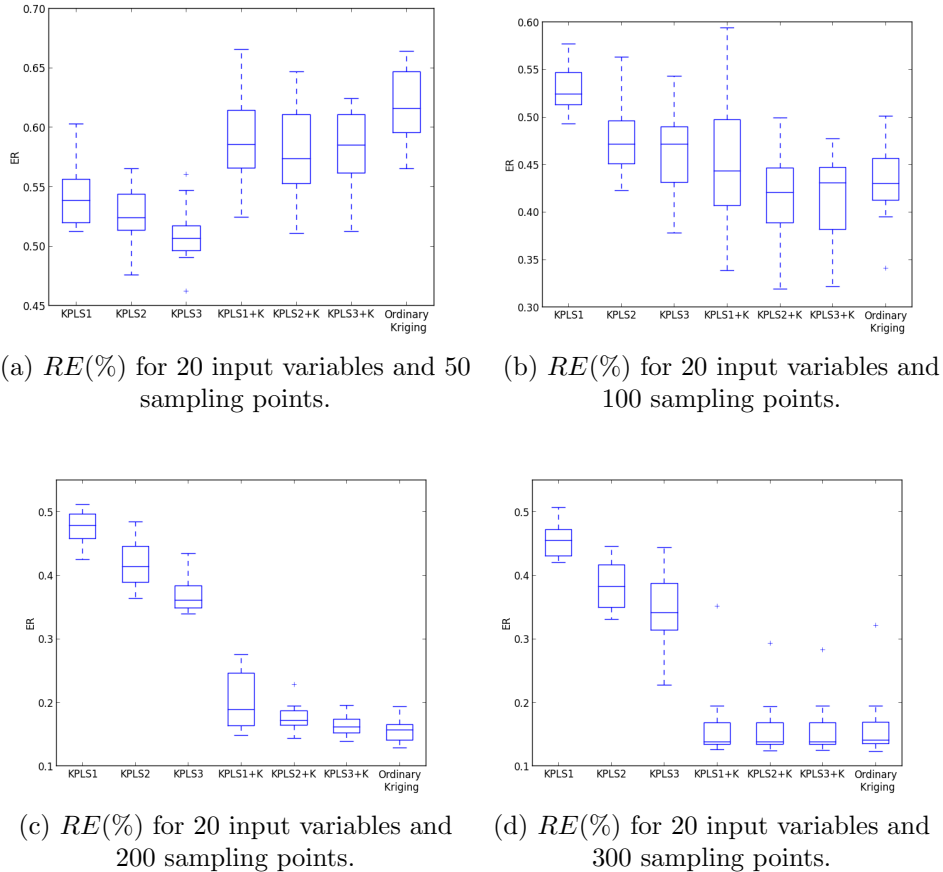
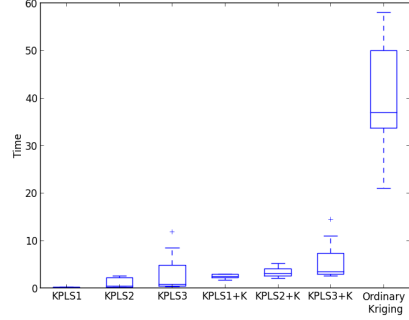
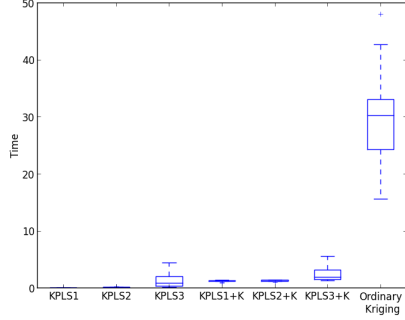


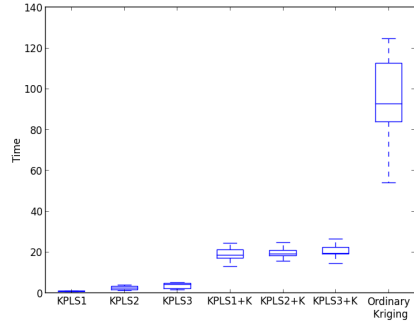
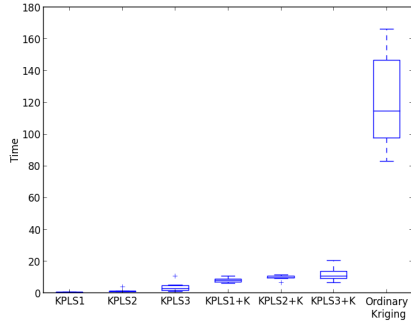
Figure 4: RE of the Griewank function in 20D over the interval $[-5, 5]$. The experiments are based on the 10-latin-hypercube design.

the accuracy of ordinary kriging becomes better than the accuracy given by the KPLS models, but it remains less accurate than for the $KPLSh+K$ models, for $h = 2$ or 3 . For instance, we obtain a mean of RE with 0.60% for $KPLS2+K$ against 0.65% for ordinary kriging (see figure 6d), when 300 sampling points are used.

As we can observe from figure 7d, a very important reduction in terms of computational time is obtained. Indeed, a mean time of 2894.56s is required for building ordinary kriging, whereas $KPLS2+K$ is built in 23.03s; $KPLS2+K$ is approximately 125 times cheaper than ordinary kriging in this case. In addition, the computational time for building $KPLS+K$ is more sta-



(a) CPU time for 20 input variables and (b) CPU time for 20 input variables and 50 sampling points.



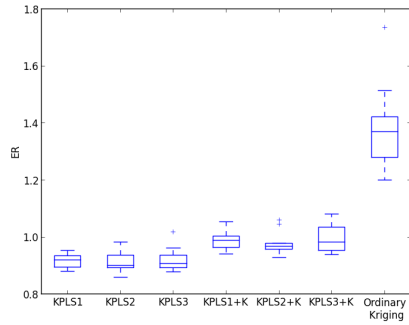
(c) CPU time for 20 input variables and (d) CPU time for 20 input variables and 200 sampling points.

Figure 5: CPU time of the Griewank function in 20D over the interval $[-5, 5]$. The experiments are based on the 10-latin-hypercube design.

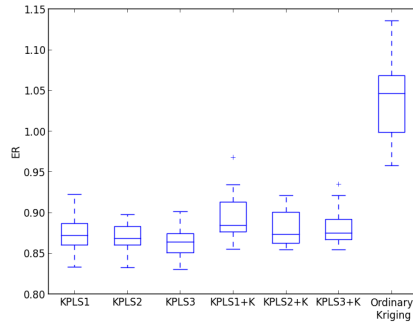
ble than ordinary kriging, except the KPLS3+K case; a standard deviation of approximately 0.30s for KPLS1+K and KPLS2+K is observed, against 728.48s for ordinary kriging. However, the relatively large standard of deviation of KPLS3+K (26.59s) is probably due from the dispersion caused by KPLS3 (26.59s). But, it remains too lower than the standard deviation of the ordinary kriging model.

For the Griewank function over the interval $[-5, 5]$, the KPLS+K models are slightly more time consuming than the KPLS models, but they are more accurate, in particular when the number of observations is greater than the dimension d , as is implied by the rule-of-thumb $n = 10d$ in [12]. They seem

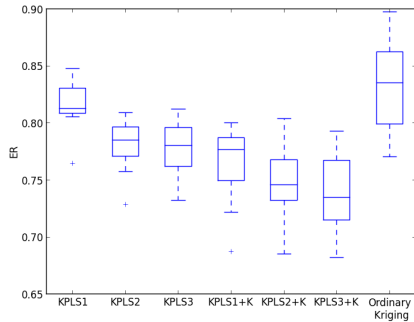
to perform well than the ordinary kriging model with an important gain in terms of saving CPU time.



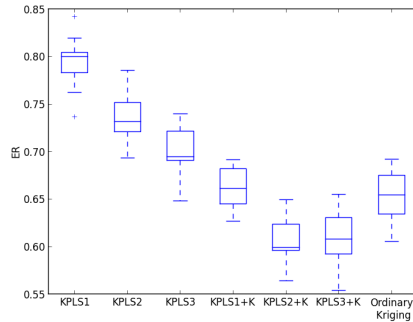
(a) $RE(\%)$ for 60 input variables and 50 sampling points.



(b) $RE(\%)$ for 60 input variables and 100 sampling points.



(c) $RE(\%)$ for 60 input variables and 200 sampling points.

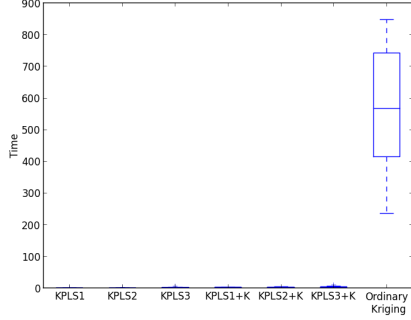


(d) $RE(\%)$ for 60 input variables and 300 sampling points.

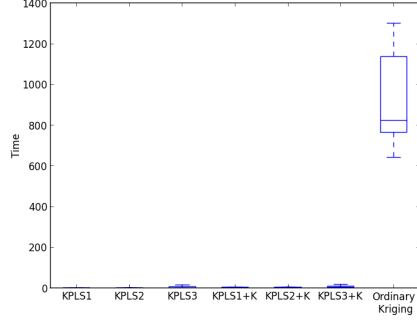
Figure 6: RE of the Griewank function in 60D over the interval $[-5, 5]$. The experiments are based on the 10-latin-hypercube design.

4.2. Engineering case

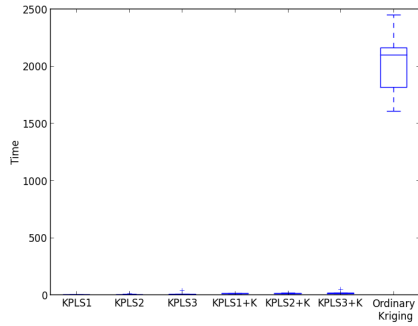
In this section, let us consider the third output, y_3 , from tab_1 problem studied in [13]. This test case is chosen because the KPLS models, from 1 to 3 principal components, do not perform well (see table 1). We recall that this problem contains 24 input variables. 99 training points and 52 validation points are used and the relative error (RE) given by equation (16) is considered.



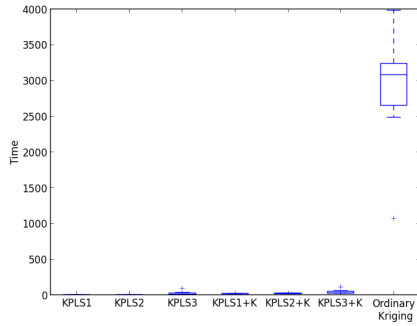
(a) CPU time for 60 input variables and 50 sampling points.



(b) CPU time for 60 input variables and 100 sampling points.



(c) CPU time for 60 input variables and 200 sampling points.



(d) CPU time for 60 input variables and 300 sampling points.

Figure 7: CPU time of the Griewank function in 60D over the interval $[-5, 5]$. The experiments are based on the 10-latin-hypercube design.

As we see in table 1, we improve the accuracy of KPLS by adding the step for building KPLS+K. This improvement is verified whatever the number of principal components used (1, 2 and 3 principal components). For these three models, a better accuracy is even found than the ordinary kriging model. The computational time required to build KPLS+k is approximately twice lower than the time required for ordinary kriging.

5. Conclusions

Motivated by the need to accurately approximate high-fidelity codes rapidly, we develop a new technique for building the kriging model faster than clas-

Table 1: Results for tab_1 experiment data (24 input variables, output variables y_3) obtained by using 99 training points, 52 validation points, and the error given by equation (16). “Kriging” refers to the ordinary kriging Optimus solution, “KPLS h ” and “KPLS h +K” refer to KPLS and KPLS+K with h principal components, respectively. Best results of the relative error are highlighted in bold type.

	Surrogate model	$RE(\%)$	CPU time
tab_1	Kriging	8.97	8.17s
	KPLS1	10.35	0.18s
	KPLS2	10.33	0.42s
	KPLS3	10.41	1.14s
	KPLS1+K	8.77	2.15s
	KPLS2+K	8.72	4.22s
	KPLS3+K	8.73	4.53s

sical techniques used in literature. The key idea for such construction relies on the choice of the start point for optimizing the likelihood function of the kriging model. For this purpose, we firstly prove equivalence between KPLS and kriging when an exponential covariance function is used. After optimizing hyper-parameters of KPLS, we then choose the solution obtained as an initial point to find the MLE of the equivalent kriging model. This approach will be applicable only if the kernels used for building KPLS are of the exponential type with the same form.

The performance of KPLS+K is verified in the Griewank function over the interval $[-5,5]$ with 20 and 60 dimensions, and an industrial case from Snecma, where the KPLS models do not perform well in terms of accuracy. The results of KPLS+K have shown a significant improvement in term of accuracy compared to the results of KPLS, at the cost of a slight increase in computational time. We have also seen, in some cases, that accuracy of KPLS+K is even better than accuracy given by the ordinary kriging model.

Acknowledgments

The authors thank the anonymous reviewers for their insightful and constructive comments. We also extend our grateful thanks to A. Chiplunkar from ISAE-SUPAERO, Toulouse for his careful correction of the manuscript.

Appendix A. Results of Griewank function in 20D and 60D over the interval $[-5, 5]$

In tables A.2 and A.3, the mean and the standard deviation (std) of the numerical experiments with the Griewank function are given for 20 and 60 dimensions, respectively.

Table A.2: Results of the Griewank function in 20D over the interval $[-5, 5]$. Ten trials are done for each test (50, 100, 200, and 300 training points). Best results of the relative error are highlighted in bold type for each case.

Surrogate	Statistic	50 points		100 points	
		error (%)	CPU time	error (%)	CPU time
kriging	mean	0.62	30.43s	0.43	40.09s
	std	0.03	9.03s	0.04	11.96s
KPLS1	mean	0.54	0.05s	0.53	0.12s
	std	0.03	0.007s	0.03	0.02s
KPLS2	mean	0.52	0.11s	0.48	1.04s
	std	0.03	0.05s	0.04	0.97s
KPLS3	mean	0.51	1.27s	0.46	3.09s
	std	0.03	1.29s	0.06	3.93s
KPLS1+K	mean	0.59	1.20s	0.45	2.42s
	std	0.04	0.16s	0.07	0.44s
KPLS2+K	mean	0.58	1.28s	0.42	3.38s
	std	0.04	0.15s	0.05	1.06s
KPLS3+K	mean	0.58	2.45s	0.41	5.61s
	std	0.03	1.32s	0.05	3.99s
	Statistic	200 points		300 points	
		error (%)	CPU time	error (%)	CPU time
kriging	mean	0.15	120.74s	0.16	94.31s
	std	0.02	27.49s	0.06	21.92s
KPLS1	mean	0.48	0.43s	0.45	0.89s
	std	0.03	0.08s	0.03	0.02s
KPLS2	mean	0.42	1.14s	0.38	2.45s
	std	0.04	0.92s	0.04	1s
KPLS3	mean	0.37	3.56s	0.35	3.52s
	std	0.03	2.75s	0.06	1.38s
KPLS1+K	mean	0.20	8.00s	0.17	19.07s
	std	0.04	1.51s	0.07	3.19s
KPLS2+K	mean	0.18	9.71s	0.16	19.89s
	std	0.02	1.29s	0.05	2.67s
KPLS3+K	mean	0.16	11.67s	0.16	20.49s
	std	0.02	3.88s	0.05	3.46s

Table A.3: Results of the Griewank function in 60D over the interval $[-5, 5]$. Ten trials are done for each test (50, 100, 200, and 300 training points). Best results of the relative error are highlighted in bold type for each case.

Surrogate	Statistic	50 points		100 points	
		error (%)	CPU time	error (%)	CPU time
kriging	mean	1.39	560.19s	1.04	920.41s
	std	0.15	200.27s	0.05	231.34s
KPLS1	mean	0.92	0.07s	0.87	0.10s
	std	0.02	0.02s	0.02	0.007s
KPLS2	mean	0.91	0.43s	0.87	0.66s
	std	0.03	0.54s	0.02	1.06s
KPLS3	mean	0.92	1.57s	0.86	3.87s
	std	0.04	1.98s	0.02	5.34s
KPLS1+K	mean	0.99	2.14s	0.90	2.90s
	std	0.03	0.72s	0.03	0.03s
KPLS2+K	mean	0.98	2.44s	0.88	3.44s
	std	0.04	0.63s	0.02	1.06s
KPLS3+K	mean	0.99	3.82s	0.88	6.68s
	std	0.05	2.33s	0.03	5.34s
	Statistic	200 points		300 points	
		error (%)	CPU time	error (%)	CPU time
kriging	mean	0.83	2015.39s	0.65	2894.56s
	std	0.04	239.11s	0.03	728.48s
KPLS1	mean	0.82	0.37s	0.79	0.86s
	std	0.02	0.02s	0.03	0.04s
KPLS2	mean	0.78	2.92s	0.74	1.85s
	std	0.02	2.57s	0.03	0.51s
KPLS3	mean	0.78	6.73s	0.70	20.01s
	std	0.02	10.94s	0.03	26.59s
KPLS1+K	mean	0.76	9.88s	0.66	22.00s
	std	0.03	0.06s	0.02	0.15s
KPLS2+K	mean	0.75	12.38s	0.60	23.03s
	std	0.03	2.56s	0.03	0.50s
KPLS3+K	mean	0.74	16.18s	0.61	41.13s
	std	0.03	10.95s	0.03	26.59s

- [1] D. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand, *Journal of the Chemical, Metallurgical and Mining Society* 52 (1951) 119–139.
- [2] G. Matheron, Principles of geostatistics, *Economic Geology* 58 (8) (1963) 1246–1266.
- [3] N. Cressie, Spatial prediction and ordinary kriging, *Mathematical Geology* 20 (4) (1988) 405–421.
- [4] J. Sacks, S. B. Schiller, W. J. Welch, Designs for computer experiments, *Technometrics* 31 (1) (1989) 41–47.
- [5] C. Rasmussen, C. Williams, *Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning*, MIT Press, Cambridge, MA, USA, 2006.
- [6] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, *Journal of Global Optimization* 13 (4) (1998) 455–492.
- [7] S. Sakata, F. Ashida, M. Zako, Structural Optimization Using Kriging Approximation., *Computer Methods in Applied Mechanics and Engineering* 192 (417) (2003) 923–939.
- [8] A. Forrester, A. Sobester, A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, 2008.
- [9] J. Laurenceau, Kriging based response surfaces for aerodynamic shape optimization, Theses, Institut National Polytechnique de Toulouse - INPT (Jun. 2008).
- [10] J. Kleijnen, W. van Beers, I. van Nieuwenhuyse, Constrained optimization in expensive simulation: Novel approach, *European Journal of Operational Research* 202 (1) (2010) 164 – 174.
- [11] J. Kleijnen, W. Beers, I. Nieuwenhuyse, Expected improvement in efficient global optimization through bootstrapped kriging, *Journal of Global Optimization* 54 (1) (2012) 59–73.

- [12] J. L. Loeppky, J. Sacks, W. J. Welch, Choosing the Sample Size of a Computer Experiment: A Practical Guide, *Technometrics* 51 (4) (2009) 366–376.
- [13] M. A. Bouhlel, N. Bartoli, A. Otsmane, J. Morlier, Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction”, *Structural and Multidisciplinary Optimization* 53 (5) (2016) 935–952.
- [14] J. Sacks, W. Welch, W. Mitchell, H. Wynn, Design and analysis of computer experiments, *Statistical Science* 4 (4) (1989) 409–435.
- [15] M. Sasena, Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations, Ph.D. thesis, University of Michigan (2002).
- [16] V. Picheny, D. Ginsbourger, Y. Richet, G. Caplin, Quantile-based optimization of noisy computer experiments with tunable precision, *Technometrics* 55 (1) (2013) 2–13.
- [17] O. Roustant, D. Ginsbourger, Y. Deville, Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization, *Journal of Statistical Software* 51 (1) (2012) 1–55.
- [18] J. P. Kleijnen, Design and analysis of simulation experiments, Vol. 230, Springer, 2015.
- [19] I. Helland, On structure of Partial Least Squares regression, *Communication in Statistics - Simulation and Computation* 17 (1988) 581–607.
- [20] I. E. Frank, J. H. Friedman, A statistical view of some chemometrics regression tools, *Technometrics* 35 (1993) 109–148.
- [21] P. Alberto, F. González, Partial least squares regression on symmetric positive-definite matrices, *Revista Colombiana de Estadística* 36 (1) (2012) 177–192.
- [22] R. Manne, Analysis of two partial-least-squares algorithms for multivariate calibration, *Chemometrics and Intelligent Laboratory Systems* 2 (1-3) (1987) 187–197.

- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. rettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn : Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [24] R. Regis, C. A. Shoemaker, Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization, *Engineering Optimization ahead-of-p* (2012) 1–27.