



HAL
open science

Automatic selection for general surrogate models

Malek Ben Salem, Lionel Tomaso

► **To cite this version:**

Malek Ben Salem, Lionel Tomaso. Automatic selection for general surrogate models. Structural and Multidisciplinary Optimization, 2018. hal-01685848

HAL Id: hal-01685848

<https://hal.science/hal-01685848v1>

Submitted on 16 Jan 2018

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.

Automatic selection for general surrogate models

Malek Ben Salem · Lionel Tomaso

Received: date / Accepted: date

Abstract In design engineering problems, the use of surrogate models (also called metamodels) instead of expensive simulations have become very popular. Surrogate models include individual models (regression, kriging, neural network...) or a combination of individual models often called aggregation or ensemble. Since different surrogate types with various tunings are available, users often struggle to choose the most suitable one for a given problem. Thus, there is a great interest in automatic selection algorithms. In this paper, we introduce a universal criterion that can be applied to any type of surrogate models. It is composed of three complementary components measuring the quality of general surrogate models: internal accuracy (on design points), predictive performance (cross-validation) and a roughness penalty. Based on this criterion, we propose two automatic selection algorithms. The first selection scheme finds the optimal ensemble of a set of given surrogate models. The second selection scheme further explores the space of surrogate models by using an evolutionary algorithm where each individual is a surrogate model. Finally, the performances of the algorithms are illustrated on 15 classical test functions and compared to different individual surrogate models. The results show the efficiency of our approach. In particular, we

Malek BEN SALEM is funded by a CIFRE grant from the ANSYS company, subsidized by the French National Association for Research and Technology (ANRT, CIFRE grant number 2014/1349).

M. Ben Salem
Ecole des mines de St-Etienne - ANSYS, Inc. Villeurbanne
Tel.: +33 - ...
Fax.: +33 - ...
E-mail: firstname.ben-salem[@]emse.fr
L. Tomaso
ANSYS, Inc. Villeurbanne
Tel.: +33 - ...
Fax.: +33 - ...
E-mail: firstname.name[@]ansys.com

observe that the three components of the proposed criterion act all together to improve accuracy and limit over-fitting.

Keywords Surrogate modeling · Multiple Surrogate models · Surrogate Model selection · Cross-Validation errors

PACS PACS fx122 · PACS fx122 · MSC 4563

1 Introduction

Computer simulations are an efficient tool to study complex physical behaviors. However, high-fidelity simulations are generally computationally expensive. Therefore, surrogate models, also known as metamodels or response surfaces, are usually instead used. They provide an approximation of a response of interest based on a limited number of expensive simulations. There are several methods of construction of such approximations. Among the popular surrogate model types, we can cite for example Kriging (Matheron, 1963), support vector machines (SVM) (Smola and Schlkopf, 2004), Moving least squares (Lancaster and Salkuskas, 1981) and Multivariate Adaptive Regressive Splines (MARS) (Friedman, 1991). Generally, a metamodel family comes with several possible tunings. In the same time, there is no universal optimal surrogate for all the problems. Some users face some difficulties in selecting the most suitable surrogate for their problem. Thus, there is a great interest in automatic model selection algorithms. The main purpose is to choose the surrogate that provides the best prediction performances on the whole parametric space.

In the literature, this problem is generally studied along three different approaches.

- 1) The first approach consists in using algorithms to optimize the settings of a particular surrogate model type. For instance, (Chen et al, 2004; Lessmann et al, 2006)

work on SVM, (Zhang et al, 2000) on neural networks, and (Tomioka et al, 2007) deal with least squares regression.

- 2) A second approach consists in considering multiple surrogates or ensembles. The automatic surrogate selection is so a model selection method. Often, the selected model is a weighted sum of different surrogate models. For example, (Viana et al, 2009; Zhou et al, 2011; Acar and Rais-Rohani, 2009; Goel et al, 2007) discuss different ways to build such aggregations.
- 3) The last approach consists in selecting a *good* member among different types of surrogate models with different settings. We refer for instance to the works of (Gorissen et al, 2009; Shi et al, 2012; Zhou and Jiang, 2016).

The main objective of our paper is to propose a new relevant surrogate model selection algorithm that can handle different type of surrogates. To achieve such a goal, we define a universal criterion. This criterion may evaluate the accuracy of any surrogate model.

The paper is organized as follows. We introduce and discuss in Section 2 our criterion called the Penalized Predictive Score (*PPS*). We show in Section 3 that *PPS* is suitable to optimize weights of surrogate models ensembles. In Section 4, we present an evolutionary selection algorithm that explores the space of surrogate models. The algorithm is called *PPS* Genetic Aggregations (*PPS-GA*). Finally, the performances of the algorithm on 15 test cases are displayed in Section 5. The results show the efficiency of the *PPS*, the complementary role of its three components and the relevance of the proposed selection algorithms.

2 Penalized Predictive Score (PPS)

2.1 Definition

Assessing the quality of a surrogate is very challenging. It is desirable to use an independent set to assess the predictive capabilities of a given method. But, this is computationally expensive in practice. One can also estimate the errors by computing the errors on design points. Unfortunately, a small MSE does not imply good predictive capabilities. Therefore, resampling techniques such as Cross-Validation (CV) (Stone, 1974) or bootstrap (Efron and Tibshirani, 1993) are generally used. Such techniques reduce the bias of the estimation. Nevertheless, they does not prevent overparameterized models. We will introduce a criterion that will do this job. This criterion is called the Penalized Predictive Score (*PPS* Equation (1)). It combines three components:

- a) The internal accuracy (or fit): we use the mean squared errors (MSE) on design points.
- b) The predictive capability: we propose to use the 10F-CV PRESS errors.

- c) A roughness penalty: We propose to use the Bending Energy Functional (BEF) ((Duchon, 1977)).

$$PPS(m, \mathbf{Z}_n) = \underbrace{\alpha \widehat{\mathcal{R}}_{l_2, \mathbf{Z}_n}(m)}_a + \underbrace{\beta \mathcal{R}_{10-CV}(m)}_b + \underbrace{\gamma E_n(m)}_c \quad (1)$$

Here, as it will be described below, $\widehat{\mathcal{R}}_{l_2, \mathbf{Z}_n}(m)$ denotes the MSE criterion, $\mathcal{R}_{10-CV}(m)$ the 10-Fold cross-validation estimate of the errors and $E_n(m)$ a roughness penalty. Further, α, β, γ are weights in \mathbb{R}_+ . In all our implementations, we use $\alpha = 2\beta$ and $\beta = 2\gamma$.

2.2 Internal accuracy

Let $\Omega = [0, 1]^d$ be the parametric space of dimension d . $\mathbf{X}_n = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \in \Omega^n$ and $\mathbf{Y}_n = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ form the set of design points $\mathbf{Z}_n = (\mathbf{X}_n, \mathbf{Y}_n)$ where $y_i = f(\mathbf{x}_i)$ for $i = 1, \dots, n$ and $f \in \mathbb{R}^\Omega$ is an expensive-to-evaluate function. A surrogate model $\widehat{m}_{|\mathbf{Z}_n} \in \Omega^{\mathbb{R}}$ is used to replace f based on the design \mathbf{Z}_n . We call the construction method a “surrogate model builder”. For instance, if m is a surrogate model builder, then we build the surrogate model $\widehat{m}_{|\mathbf{Z}_n} \in \Omega^{\mathbb{R}}$ based on the design \mathbf{Z}_n .

The assessment of the performance of a surrogate model is extremely important in practice (Hastie et al, 2009). It relies on the evaluation on the set of design points of the prediction capabilities of the surrogate model. It is generally based on a *contrast function* (or *loss function*) that measures the errors between the predicted and the true models. A typical choice is the square error $l_2(x, y) = (x - y)^2$. The integral form of the MSE is the l_2 -risk overall the parametric space.

$$\mathcal{R}_{l_2, \mathbf{Z}_n}(m) = \int_{\Omega} l_2(\widehat{m}_{|\mathbf{Z}_n}(\mathbf{x}), f(\mathbf{x})) d\mathbf{x} \quad (2)$$

Since f is unknown, we can only use an approximation to estimate this risk. Ideally, the performance of the surrogate model would be evaluated on an extra set of points. However, generating such set is sometimes computationally expensive. Therefore, one use an empirical distribution associated to the set of design points. Computing the mean square errors (MSE) (Equation (3)) on the set of design points for the surrogate model $\widehat{m}_{|\mathbf{Z}_n}$ is an empirical approximation of $\mathcal{R}_{l_2, \mathbf{Z}_n}(m)$ defined in Equation (2).

$$\begin{aligned} \widehat{\mathcal{R}}_{l_2, \mathbf{Z}_n}(m) &= \frac{1}{n} \sum_{i=1}^n l_2(\widehat{m}_{|\mathbf{Z}_n}(x_i), y_i) \\ &= \frac{1}{n} \sum_{i=1}^n (\widehat{m}_{|\mathbf{Z}_n}(x_i) - y_i)^2 \end{aligned} \quad (3)$$

Note that computing the MSE on the set of design points is a biased estimate of the error in the whole space. In fact,

for any interpolating surrogate model m , $\widehat{\mathcal{R}}_{l_2, \mathbf{Z}_n}(m) = 0$. This does not necessarily mean that the surrogate model fits the real function in the whole space.

2.3 Predictive capabilities

On one hand, the use of design points to estimate the errors yields an optimistic result (Arlot and Celisse, 2010). On the other hand, using a validation set can be expensive. Therefore, it is convenient to use re-sampling techniques such as Cross-Validation (CV) (Stone, 1974) and bootstrap (Efron and Tibshirani, 1993) to estimate the predicted errors. Resampling techniques estimate the errors by using subsets of the design points to build several *sub-surrogate models*. For instance, computing the Leave-One-Out Cross-Validation (LOO-CV) errors of a surrogate model $\widehat{m}_{\mathbf{Z}_n}$ consists in computing the errors of an observation (\mathbf{x}_i, y_i) based on the surrogate model $\widehat{m}_{\mathbf{Z}_n, -i}$ built on the subset of all the design points except the i^{th} design point ($\mathbf{Z}_{n, -i} = (\mathbf{x}_j, y_j)_{j \neq i}$). In the same way, k -fold cross-validation (k F-CV) consists in dividing the data into k subsets. Each subset plays the role of validation set while the remaining $k - 1$ subsets are used together as the training set. If k is the number of folds, for $i \in 1, \dots, k$ let $\mathbf{Z}^{(i)} \in \mathcal{P}(\mathbf{Z}_n)$ be a subset of \mathbf{Z}_n such that $\cup_{i=1}^k \mathbf{Z}^{(i)} = \mathbf{Z}_n$. The k F-CV estimates of the l_2 errors (Equation (4)) by computing the loss of a point in the i^{th} fold $\mathbf{Z}^{(i)}$ compared to the prediction of the surrogate model built on the remaining folds ($\mathbf{Z}_n \setminus \mathbf{Z}^{(i)}$).

$$\mathcal{R}_{k\text{-CV}}(m) = \frac{1}{n} \sum_{i=1}^k \sum_{(x', y') \in \mathbf{Z}^{(i)}} l_2(\widehat{m}_{\mathbf{Z}_n \setminus \mathbf{Z}^{(i)}}(x'), y') \quad (4)$$

where $\mathbf{z} \in \mathbf{Z}_n \setminus \mathbf{Z}^{(i)}$ if and only if $\mathbf{x} \in \mathbf{Z}_n$ and $\mathbf{x} \notin \mathbf{Z}^{(i)}$.

(Queipo et al, 2005) pointed out that the main advantage of CV is that it provides a nearly unbiased estimate. Further, (Kohavi, 1995) studied Cross-Validation and Bootstrap performances on a large dataset and recommended using stratified 10-fold-cross-validation. (James et al, 2013) stated that k F-CV with $k = 5$ or $k = 10$ yield test error estimates that suffer neither from excessively high bias nor from very high variance.

2.4 Penalization

Penalties are used in several model selection frameworks in order to prevent over-fitting. Selection criteria such as the Bayesian Information Criterion (BIC) (Schwarz et al, 1978) or Akaike Information Criterion (AIC) (Akaike, 1974) penalize the models by their degrees of freedom. Most penalties are designed for a particular family of surrogates. Here, we are interested in universal methods. So that, we prefer to

deal with the smoothness of the surrogate model rather than with its structural complexity. For instance, (Nguyen et al, 2011) introduce a criterion called Linear Reference Model (LRM). It scores a surrogate model by computing the deviation between its predictions and a local linear model \widehat{l}_{rm} . The LRM is computed over a set of N points $x^{(k)}$ for $k = 1, \dots, N$ (see Equation (5)).

$$\mathcal{R}_{LRM}(m) = \frac{1}{N} \sum_{k=1}^N l_2(\widehat{m}_{\mathbf{Z}_n}(x^{(k)}), \widehat{l}_{rm}(x^{(k)})) \quad (5)$$

Computationally, this last criteria needs the construction of a Delaunay tessellation (Watson, 1981) to compute \widehat{l}_{rm} . The computational cost of such construction in high dimension is too expensive. We suggest to use a criterion that penalize the roughness of surrogate models: the thin plate spline (TPS) (Duchon, 1977) Bending Energy Functional (BEF). It is a second order partial derivatives-based penalty. For a dimension d , the roughness penalty E_n is the integral of the squared term of the Hessian (Equation (6)).

$$E_n(\hat{f}) = \int_{\Omega} \sum_{i=1}^d \sum_{j=1}^d \left(\frac{\partial^2 \hat{f}}{\partial x_i \partial x_j} \right)^2 dx \quad (6)$$

LRM can be used in place of the BEF in the selection criterion *PPS*. It penalizes the deviation from a linear model regardless of its roughness. It still gives good predictive capabilities also. Nevertheless, some rough surrogates may be selected.

3 Surrogate model ensemble: PPS-OS

3.1 Overview

Surrogate model selection consists in selecting a surrogate model among a collection of them. This means that we evaluate the performances of several surrogate models and then choose one of them. (Acar and Rais-Rohani, 2009) stated that *this practice has some shortcomings as it does not take full advantage of the resources devoted to constructing different metamodels*. In fact, it is possible to consider a weighted combination of surrogates without any significant extra computational cost. These combinations are called: ensembles, aggregations and multiple surrogates.

(Forrester and Keane, 2009) show that these aggregation methods drastically improve the performances of the surrogate models. In general, ensembles require small computational resources compared to the cost of the simulations (Queipo et al, 2005). The general form of an aggregation of

p surrogate models $\widehat{m}^{(i)}|_{\mathbf{z}_n}$, for $i = 1, \dots, p$ is given in Equation (7):

$$\widehat{A}|_{\mathbf{z}_n}(x) = \sum_{i=0}^p w_i(x) \widehat{m}^{(i)}|_{\mathbf{z}_n}(x) \quad (7)$$

For instance, (Zerpa et al, 2005) considered a local combination called weighted average model where the weights are based on the local expected variances of the surrogate models. (Goel et al, 2007) extended the use of ensembles to the identification of region with high error. They presented also several heuristics to weight ensembles.

However, (Gorissen et al, 2009) used a simple average ensemble (all the weights are equal). (Müller and Piché, 2011) proposed to weight the aggregation using the Dempster-Shafer theory where the error estimates are used as basic probability assignments. (Viana et al, 2009) proposed to use an ensemble of surrogate models that minimize the CV errors. In fact, if for $k = 1, \dots, n$, \mathbf{v}_k is the vector of CV errors of the surrogate model $\widehat{m}^{(i)}|_{\mathbf{z}_n}$, the CV errors of the aggregation is then $\mathbf{W}^T \mathbf{C} \mathbf{W}$. The weights are selected to minimize the CV errors of the aggregation under the constraint $\sum_{i=1}^p w_i = 1$. The optimal weighted surrogate *OWS* is obtained using the weights of Equation (8).

$$\mathbf{W} = \frac{\mathbf{C}^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}} \quad (8)$$

where the elements of the matrix \mathbf{C} , $c_{ij} = \langle \mathbf{v}_i, \mathbf{v}_j \rangle$. (Viana et al, 2009) noticed that the solution may include negative values. They stated that this additional freedom to the weights estimation amplify errors. In fact, the matrix \mathbf{C} is an approximation of the covariance of the errors of the surrogate models. To overcome the problem, the authors suggested to use only the diagonal elements of \mathbf{C} . Then, the weights are $w_i = \frac{c_{ii}^{-1}}{\sum_{k=1}^n c_{kk}^{-1}}$. This formulation is close to the weights of the PRESS weighted surrogate (*PWS*) given in (Goel et al, 2007) (equation (9)), with $\alpha = 0, \beta = -2$.

$$w_i = \frac{(\sqrt{c_{ii}} + \frac{\alpha}{n} \sum_{j=1}^n \sqrt{c_{jj}})^\beta}{\sum_{k=1}^n (\sqrt{c_{kk}} + \frac{\alpha}{n} \sum_{j=1}^n \sqrt{c_{jj}})^\beta} \quad (9)$$

3.2 PPS-optimal ensemble

Let us consider $(\widehat{m}^{(1)}|_{\mathbf{z}_n}, \dots, \widehat{m}^{(n)}|_{\mathbf{z}_n})$ a set of p surrogate models. Let A be an aggregation of these surrogate models

weighted by the vector $\mathbf{W} = (w_1, \dots, w_n)$ (Equation (10)).

$$\widehat{A}(x) = \sum_{k=1}^p w_k \widehat{m}^{(k)}|_{\mathbf{z}_n}(x) \quad (10)$$

In our formulation, we compute the weights of the aggregations by optimizing the *PPS* of the aggregation under the constraint $\sum_{i=1}^p w_i = 1$. The *PPS*-Optimal aggregation is then the aggregation in which the weights are the solution of the optimization Problem (11).

$$\begin{aligned} \min_{\mathbf{W}} \quad & PPS(A, \mathbf{Z}_n) \\ \text{u.c.} \quad & \sum_{i=1}^p w_i = 1 \end{aligned} \quad (11)$$

For each k in $1, \dots, p$, let:

- \mathbf{e}_k be the vector of errors on design points.
- \mathbf{v}_k the vector of cross-validation error of the surrogate model $\widehat{m}^{(k)}|_{\mathbf{z}_n}$.

Notice then that the MSE of the aggregation is a quadratic form of the weights

$$\mathcal{R}_{l_2, \widehat{P}_n}(A) = \left\| \sum_{i=1}^p w_i \mathbf{e}_i \right\|^2 = \mathbf{W}^T \mathbf{E} \mathbf{W} \quad (12)$$

Where the elements of \mathbf{E} , $E_{ij} = \langle \mathbf{e}_i, \mathbf{e}_j \rangle$. Similarly, the cross validation errors of the aggregation is also a quadratic form of the weights (Equation (13)) where \mathbf{C} is the same defined in the previous section.

$$\mathcal{R}_{CV}(A) = \mathbf{W}^T \mathbf{C} \mathbf{W} \quad (13)$$

Last, the energy functional is also a quadratic form of the weights (Equation 14).

$$\begin{aligned} E_n(\widehat{A}) &= \int_{\Omega} \sum_{i=1}^d \sum_{j=1}^d \left(\frac{\sum_{k=1}^p w_k \partial^2 \widehat{m}^{(k)}|_{\mathbf{z}_n}(x)}{\partial x_i \partial x_j} \right)^2 dx \\ &= \mathbf{W}^T \mathbf{K} \mathbf{W} \end{aligned} \quad (14)$$

$$\text{where: } \mathbf{K} = \left[k_{kl} = \sum_{i=1}^d \sum_{j=1}^d \int_{\Omega} \left(\frac{\partial^2 \widehat{m}^{(k)}|_{\mathbf{z}_n}(x)}{\partial x_i \partial x_j} \right) \left(\frac{\partial^2 \widehat{m}^{(l)}|_{\mathbf{z}_n}(x)}{\partial x_i \partial x_j} \right) dx \right].$$

Let $\mathbf{R} = \alpha \mathbf{E} + \beta \mathbf{C} + \gamma \mathbf{K}$. The *PPS* of the aggregation is then a quadratic form of the weights \mathbf{W} : $PPS(\widehat{A}) = \mathbf{W}^T \mathbf{R} \mathbf{W}$. The *PPS*-Optimal aggregation is then the aggregation that minimizes the *PPS* under the constraint $\sum_{i=1}^n w_i = 1$. The solution is defined in Equation (15):

$$\mathbf{W}^* = \frac{\mathbf{R}^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \quad (15)$$

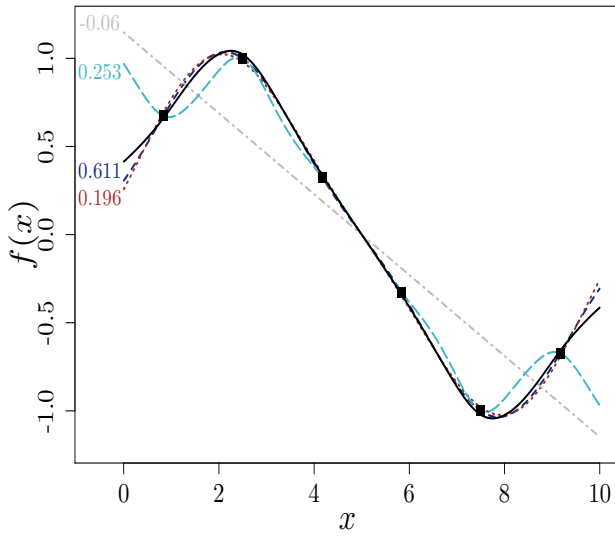


Fig. 1: Example of *PPS*-Optimal ensemble, Dashed lines: 4 meta-models predictions. Solid line: *PPS*-optimal ensemble predictions. Black squares: design points

Similarly to Equation (8), the solution of Equation (15) may include negative weights as well as weights greater than one. Unlike, in (Viana et al, 2009) in which the writers suggested to use only the diagonal terms in the matrix to ensure the positivity, here we tolerate such weights since this freedom is controlled by the BEF penalization. As a matter of fact, the BEF penalization prevents to artificial oscillations on the aggregated surrogate.

3.3 Illustrative example

We consider the example in Figure 1. The ensemble is the optimal trade-off defined by the *PPS* parameters. The ensemble is relatively smoother than the interpolating ones of the initial collection. Further, its CV error is lesser than the best prediction of this collection.

3.4 One shot metamodel selection: PPS-OS

We suppose that we have at hands p possible surrogate model builders where p is relatively small (typically $p \leq 35$). One select the model that has the best *PPS*. In order to improve the result, we select the *PPS*-Optimal ensemble. We consider this procedure (described in Algorithm 3.1) as a model selection algorithm. Notice that the aggregation does not increase significantly the computational cost of the procedure as the errors have been generally previously evaluated.

Algorithm 3.1 *PPS* One Shot (*PPS*-OS) model selection algorithm

Inputs: Design Points Z_n

Generate the list of first population of surrogate models builder $L = (m_1, m_2, \dots, m_p)$.

Compute the *PPS*-Optimal aggregation \hat{A}

Outputs: \hat{A}

In our implementation, *PPS*-OS selects the *PPS*-optimal aggregation of 32 surrogate models from 4 different surrogate types (Kriging, SVM, Polynomial regression and MLS).

4 *PPS*-based Genetic Aggregation for model selection : (*PPS*-GA)

As discussed in the previous section, the use of *PPS* to perform model selection is straightforward if the number of the available surrogate model is moderate. In that case, one can consider a weighted *PPS*-Optimal aggregation of all the possible surrogate models. However, there are many types of surrogate models and each type has several possible settings. For instance, to tune a universal kriging surrogate model, there are various possible choices for covariance function and trend function. Consequently, one cannot evaluate the *PPS* for all the possible combinations. Even with a good selection criterion, one need to explore the space of available surrogate models to select the best one.

(Gorissen et al, 2009) proposed an evolutionary algorithm to perform surrogate model selection and to explore the space of surrogate models. The surrogate models are considered as the individuals of the population. The settings of the surrogate models are considered as the genetic information of the individuals. The mutation and cross-over operators between two surrogate models of the same type are performed by modifying or exchanging the surrogate models settings. Further, they generate an equally weighted surrogate model ensemble when the cross-over is between two surrogate models of different types. Their algorithm uses the island model of evolutionary algorithms.

We now introduce our selection algorithm based on the genetic aggregation called *PPS*-GA. Similarly to (Gorissen et al, 2009)'s heuristic, the mutation and cross-over operators are performed over surrogate model builders settings. In our algorithm, all the aggregation weights are now optimized according to the *PPS*. Moreover, we add new aggregations at each iteration. The members of these aggregations are generated randomly. Further, we do not adopt the island model. We consider that the heterogeneous set of surrogate model builders "lives" together in the same space. The selection method is designed to conserve the diversity.

Algorithm 4.1 *PPS* Genetic Aggregation (*PPS-GA*) model selection algorithm

Inputs: Design Points \mathbf{Z}_n , $l = 10$.

 Generate the list of first surrogate models builders $L = (m_1, m_2, \dots, m_k)$.

for $Generation = 1$ to $MaxGeneration$ **do**
 m_{agg} = Compute the optimal aggregation of the l best surrogate models according to *PPS*
 L_{new} = Perform mutation and cross-over operations

 $L = L \cup L_{new} \cup m_{agg}$
 L = Select the best k surrogate models according to *PPS*.

end
 $m_{|\mathbf{Z}_n}^*$ = Select the best surrogate model of L .

Outputs: $m_{|\mathbf{Z}_n}^*$

In our implementation, we consider several surrogate types with various settings: Kriging, moving least squares, polynomial regression and support vector machines regression. *PPS-GA* has another interesting property. It is easy to enrich the set of surrogate model builders. In fact, the algorithm does not require any particular assumption. It is in part due to the universality of *PPS*.

5 Numerical examples

5.1 Benchmark problems

In order to check the efficiency of *PPS-OS* and *PPS-GA*, we tested their performances on a benchmark of 15 func-

tions (see Table 1 and formula given in Appendix. B). For each function, we generated 10 different optimized maximin Latin hypercube sampling (LHS) (McKay et al, 1979) of size N . We generated an extra test set of size $n_t = 1000 \times N$ by a fast optimized LHS algorithm (Viana et al, 2010). We use the *RMSE* criterion (Equation (16)) to evaluate the performances on the set of verification points.

$$RMSE = \sqrt{\frac{1}{n_t} \sum_{i=1}^{n_t} (y_i - \hat{y}_i)^2} \quad (16)$$

For each function, we compare the performance of the selection algorithms (*PPS-OS* and *PPS-GA*) to the performances of 4 witness surrogate models:

- a) A kriging surrogate model using an an-isotropic Matérn 5/2 kernel and a linear trend function.
- b) A support vector regression using a Gaussian kernel and ε -regression paradigm.
- c) A moving least squares surrogate model using a Gaussian weighting function and second order polynomial regression.
- d) Full second order polynomial regression, we use least-norm when the equation system is undetermined.

These surrogates are selected among the 32 surrogates of *PPS-OS* as follows: We consider the 150 functions (15×10 repetitions). For each surrogate \hat{m} , we compute $N_{best}(\hat{m})$: the number of times where \hat{m} is the best individual surrogate. Each witness surrogate models is the one with highest N_{best} among its type. The surrogate with the highest N_{best} is the kriging using an an-isotropic Matérn 5/2 kernel and a linear trend function. It is the best individual surrogate in 25 test (16%).

Table 1: Test functions

Name	Dimension d	Number of design points N	Number of test points n_t
1. Wing weight function	10	45	45000
2. Borehole function	8	40	40000
3. Dette & Pepelyshev (8-Dim) function	8	75	75000
4. Piston simulation function	7	60	60000
5. OTL circuit function	6	35	35000
6. Gramacy & Lee (2009) function	6	85	85000
7. Friedman function	5	35	35000
8. Dette & Pepelyshev exponential function	3	16	16000
9. Dette & Pepelyshev curved function	3	18	18000
10. Lim non-polynomial function	2	12	12000
11. Currin exponential function	2	20	20000
12. Franke's function	2	10	10000
13. Gramacy & Lee (2008) function	2	45	45000
14. Sasena function	2	10	10000
15. Gramacy & Lee (2012) function	1	15	15000

5.2 Results

We display the results of the benchmark in Table 2 and in Figures 2-16:

- In Table 2, the median and the standard deviation of the RMSE of each surrogate model are given. The best median value is in bold.
- In Figures 2-16, the box-plots illustrate the variability with respect to the design set.

The results show the efficiency of the selection algorithms: the models selected by *PPS-OS* and *PPS-GA* outperform each individual surrogate models in the predictive capabilities for at least one function. Generally, the RMSE of the selected surrogates is generally either the best or close to the best one.

	MLS		SVM		Poly		Kriging		<i>PPS-OS</i>		<i>PPS-GA</i>	
	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd	mean	sd
Wing Weight	6.646	0.500	12.889	0.225	15.890	4.332	5.800	1.076	3.873	0.708	3.701	0.560
Borehole	12.077	1.933	13.267	0.442	1341.448	2050.563	9.014	2.128	3.197	0.418	3.627	0.467
Dette & Pepelyshev 8-Dim	14.574	11.524	5.236	0.134	10.819	9.006	1.771	0.780	1.995	0.902	3.609	0.162
Piston Simulation	0.037	0.002	0.040	0.001	0.087	0.083	0.016	0.006	0.011	0.001	0.014	0.003
OTL Circuit	0.287	0.141	0.312	0.004	0.303	0.172	0.112	0.037	0.036	0.011	0.055	0.013
Gramacy & Lee 2009	1.421	0.498	0.669	0.012	1.223	0.667	0.410	0.092	0.243	0.139	0.380	0.179
Friedman	4.215	1.607	1.522	0.107	4.218	1.714	1.251	0.244	0.634	0.284	0.854	0.195
Dette & Pepelyshev Exp	0.955	0.038	2.860	0.147	0.998	0.032	3.280	0.175	1.139	0.362	1.293	0.665
Dette & Pepelyshev Curved	1.765	0.129	3.330	0.146	2.034	0.048	2.466	0.796	1.414	0.409	1.821	0.592
Lim Non Polynomial	0.395	0.044	0.374	0.048	0.433	0.037	0.251	0.033	0.441	0.187	0.460	0.095
Currin Exp	0.970	0.142	1.049	0.098	1.331	0.050	0.692	0.324	0.554	0.268	0.438	0.199
Franke	0.093	0.007	0.062	0.004	0.132	0.002	0.060	0.016	0.052	0.010	0.062	0.013
Gramacy & Lee 2008	0.058	0.002	0.069	0.001	0.074	0.001	0.040	0.006	0.035	0.008	0.035	0.006
Sasena	2.942	0.056	3.512	0.119	4.423	0.358	2.434	0.399	2.341	0.608	2.138	0.504
Gramacy & Lee 2012	0.426	0.067	0.527	0.097	0.508	0.034	0.456	0.071	0.458	0.073	0.471	0.127

Table 2: Mean and Standard deviation of RMSE

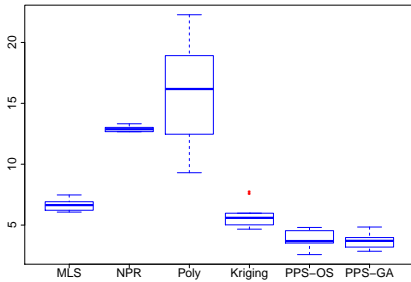


Fig. 2: Wing weight function

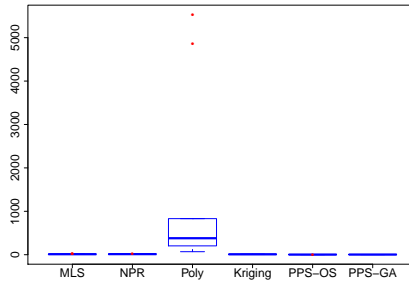


Fig. 3: Borehole function

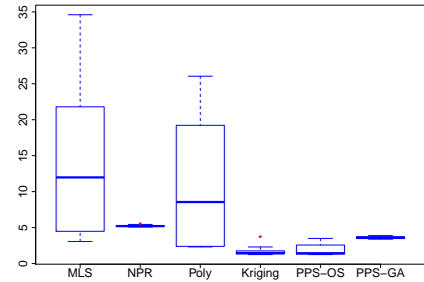


Fig. 4: D & P (8-Dim) function

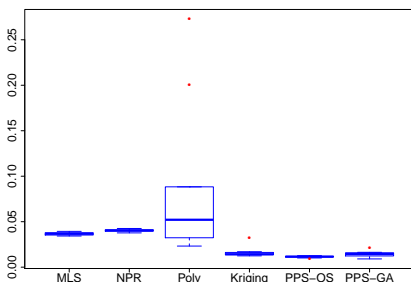


Fig. 5: Piston simulation function

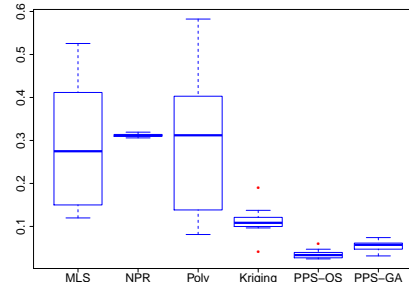


Fig. 6: OTL circuit function

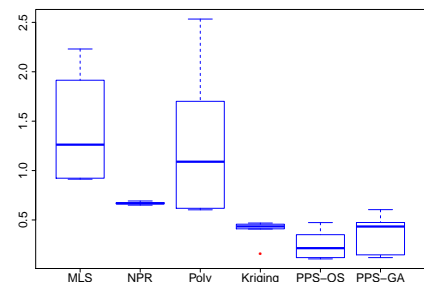


Fig. 7: G & L 2009 function

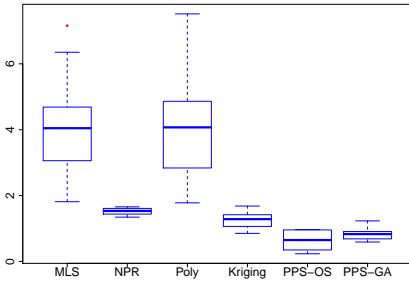


Fig. 8: Friedman function

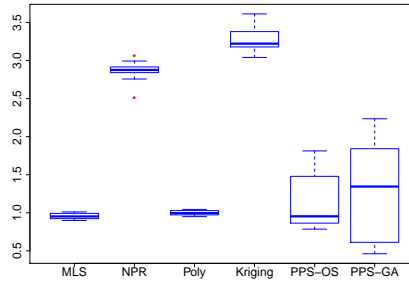


Fig. 9: D & P exponential function

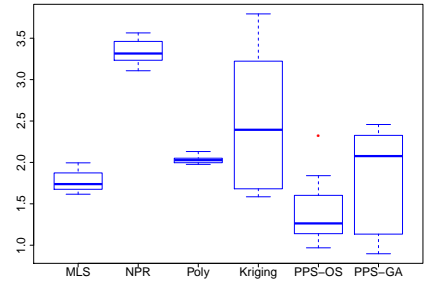


Fig. 10: D & P curved function

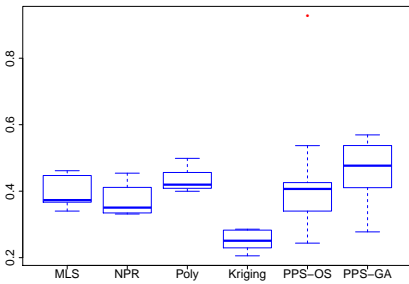


Fig. 11: Lim non-polynomial function

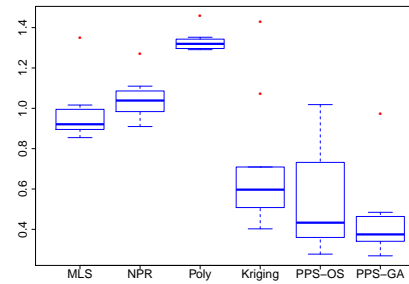


Fig. 12: Currin exponential function

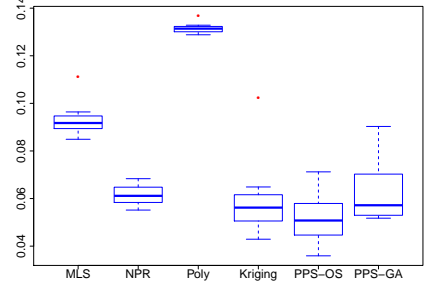


Fig. 13: Franke function

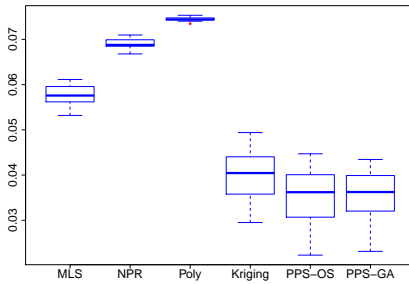


Fig. 14: G & L (2008) function

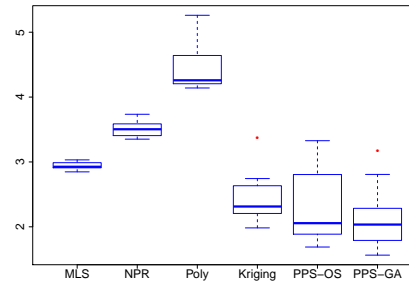


Fig. 15: Sasena function

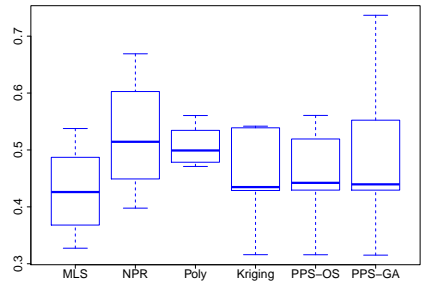


Fig. 16: G & L 2012 function

5.3 PPS-based ensembles

We also use the same test bench to compare the PPS-optimal ensemble, the OWS ensemble and the PWS ensemble with $\alpha = 0.05$ and $\beta = -1$. Here, we have at hands 10 surrogate models and we compute the weights by these three different techniques. The results are given in Figure 18. In order to display all the results in the same figure, we have rescaled the values of all the bench functions in $[0,1]$.

Generally, PPS-optimal ensemble give the best result except for the Dette & Pepelyshev Exp function where the PWS is better and for the Dette & Pepelyshev 8-Dim function where OWS is better. Moreover, for all the functions, PPS-optimal ensemble never has the worst RMSE. This shows the suitability of PPS to construct ensembles.

5.4 On the choice of α , β and γ

Further, recall that the PPS criterion is composed of 3 components: MSE, CV and Pen. It is then useful to understand the role played by each component in the performances. To do so, we compared the performance of the algorithms based on the PPS (PPS-GA and PPS-OS) to same algorithms in which we replaced the selection criterion by only one or a sum of two components of the PPS. Therefore, the entire set of tested algorithms are:

- PPS-based algorithms (PPS-GA, PPS-OS)
 - MSE ●
 - CV ●
 - P ●
- MSE only
 - MSE ●
 - CV ○
 - P ○
- CV only
 - MSE ○
 - CV ●
 - P ○

- Penalty only
 - MSE ○
 - CV ○
 - P ●
- CV + Penalty
 - MSE ○
 - CV ●
 - P ●
- MSE + Penalty
 - MSE ●
 - CV ○
 - P ○
- MSE + CV
 - MSE ●
 - CV ●
 - P ○

Here, we used LRM as penalty because it gives a reasonable prediction quality when used as a selection criterion. BEF cannot be used in this context. It is only a penalty that favors monomials and constant functions regardless of the output value. We display in Figures 19, 20, 21 and 22 the results of the comparison between the different algorithms for some functions of Table 1: Currin function, Sasena function, Dette & Pepelyshev curved function and Dette & Pepelyshev 8-dim function. For each function, 10 different maximin LHS design are generated of size $N = 10d$ where d is the space dimension.

For these functions, we can notice how the different components of the PPS act together to select a convenient surrogate model in different scenarios. In fact, the results highlight the effect of each component. Obviously, neither a single criterion nor any combination of two criteria is better than PPS in all the cases. This is due to:

- Any interpolating surrogate model is MSE-optimal. It is a misleading criterion to the overall errors.
- CV is a convenient estimate of the predictive capabilities. But, it is a pessimistic one.

We also study the choice of the values of the parameters of the PPS on the benchmark. We used ten surrogate models and we computed the sum of RMSE for each value of β and γ , α being fixed to 1. Let (β^*, γ^*) denotes the global minimum. We display the contour plot of the sum of mean square errors (MSE) in Figure 17. Notice that the proposed values of Section 2 (1, 0.5, 0.25), are close to the optimum. Further, they give better sum of MSE error 0.3760 if compared to (β^*, γ^*) that leads to 0.3771. We display also the errors of the ensembles using these two parameters in Appendix A. Notice that the prediction errors are close.

5.5 On the relevance of ensembles

Finally, we display in Figure 23 the number of surrogate models in the selected surrogate of PPS-GA. We notice that the algorithm selects generally a PPS-optimal ensemble. This is due in part to the PPS suitability to ensemble construction and it shows that aggregations are relevant in metamodel se-

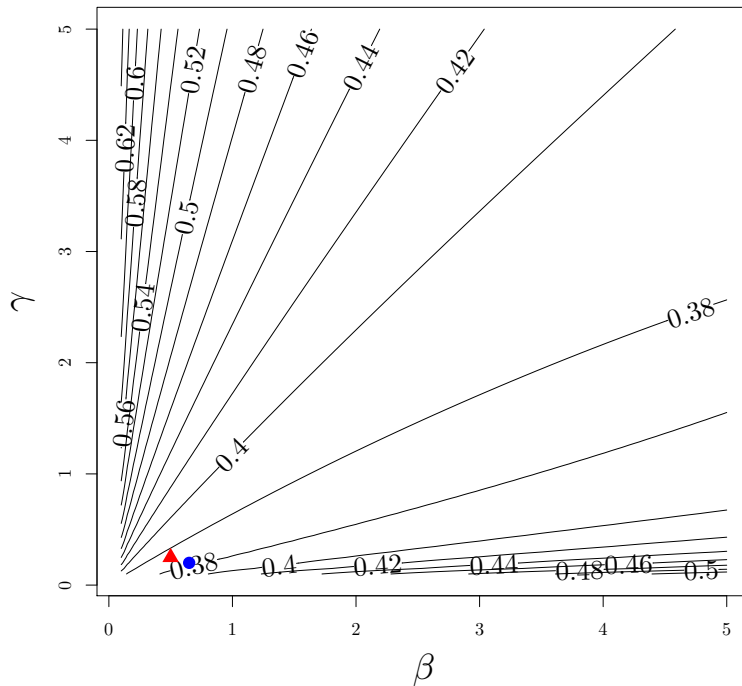


Fig. 17: Contour plot of the sum of scaled MSE of 150 test functions (15 × 10 repetitions), Blue circle: optimum of sum of RMSE, Red triangle: our proposed value.

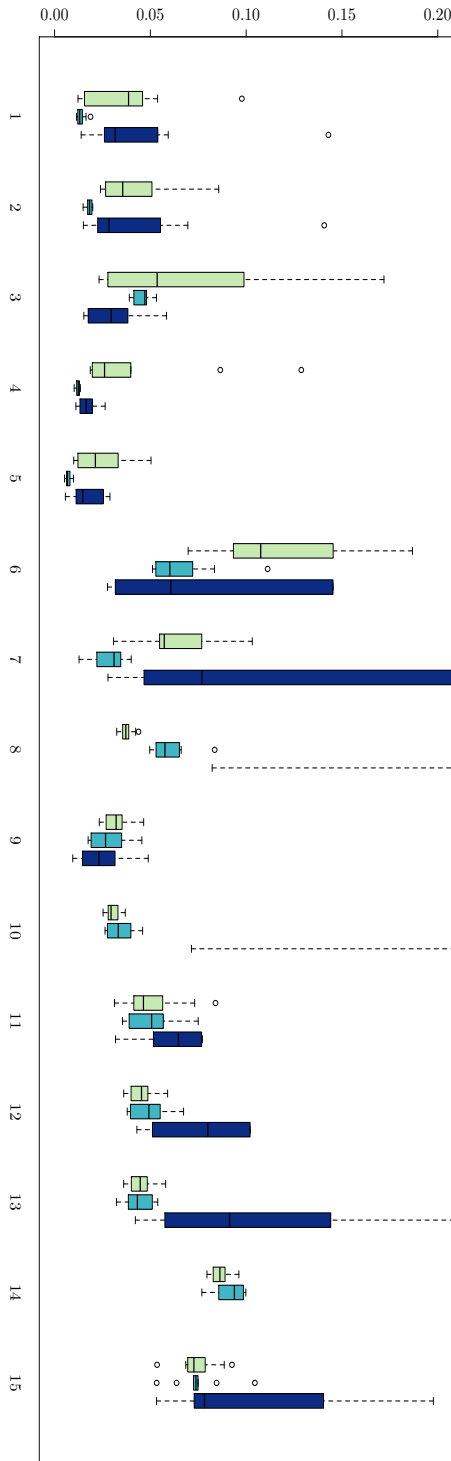


Fig. 18: For each function: Left: *PWS* method in light green. Middle: *PPS*-optimal ensemble in light blue. Right: *OWS* ensemble in dark blue. The function number is as in Table (1)

lection. This shows the usefulness of the ensemble approach.

5.6 Computational cost

We give in Table 3, the quantiles and the sum of the computing time of all the 150 benchmark functions. It is expected that the selection methods needs more time than individual surrogates. We can notice also that *PPS-GA* is computationally more expensive than *PPS-OS*. This is due to the cost of exploration. Finally, notice that these values are negligible compared to the computing time of one complex simulation.

	MLS	NPR	Poly	Kriging	<i>PPS-OS</i>	<i>PPS-GA</i>
0%	0.000	0.000	0.000	0.000	0.012	0.386
25%	0.000	0.001	0.000	0.001	0.065	0.933
50%	0.000	0.001	0.000	0.014	0.336	1.769
75%	0.000	0.003	0.003	0.068	0.886	2.865
100%	0.001	0.027	0.022	0.136	1.884	5.055
Sum	0.007	0.394	0.406	5.398	79.758	297.509

Table 3: Elapsed time in seconds to construct each surrogate model

6 Conclusion

In this paper, we propose a new selection criterion called the penalized predictive score. *PPS* can be computed for all the types of surrogate models. By construction, *PPS* is especially suitable for functions that have specific characteristics such as regularity and smoothness. Generally these characteristics are implicitly expected with the meta-modeling framework. We showed also that it enables the construction of relevant ensembles. The *PPS*-optimal ensemble are easily computed and avoid over-fitting.

We study also two surrogate model selection schemes based on the *PPS*. The first one compute the *PPS*-optimal ensemble rather than selecting one surrogate model. The second one is based on a evolutionary framework that enables the exploration of the space of surrogate models. Tests shows that the proposed algorithms give very good results. It remains important to notice that this algorithm does not necessarily give an accurate approximation in all the cases. For instance, the algorithm will fail if we use a small amount of observations for a highly nonlinear behavior. It aims at selecting the best surrogate among the possible choices. Assessing the level of confidence of a prediction is left for future research.

Acknowledgements We gratefully thank Olivier Roustant and Fabrice Gamboa for the help in writing this paper and for their valuable

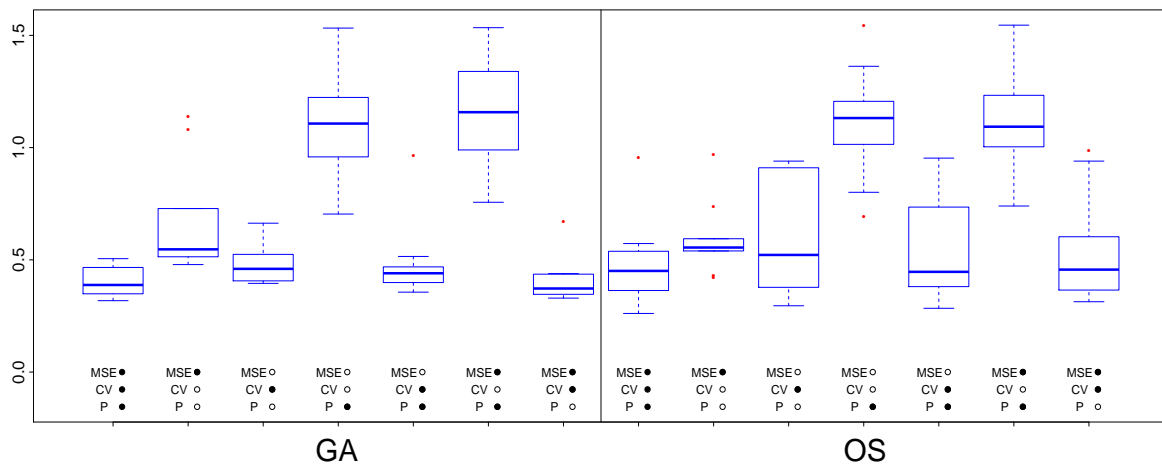


Fig. 19: Several selection criteria: Currin function

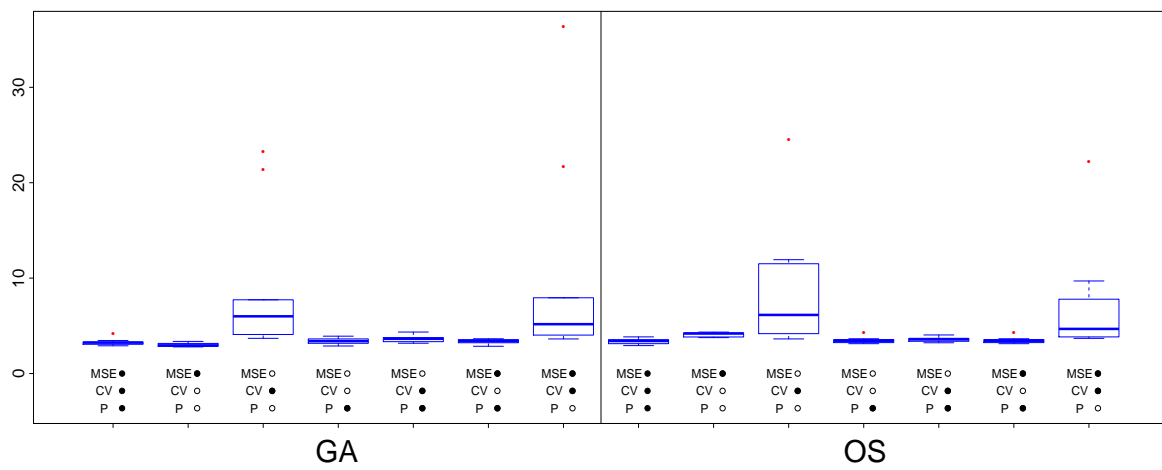


Fig. 20: Several selection criteria: Sasena function

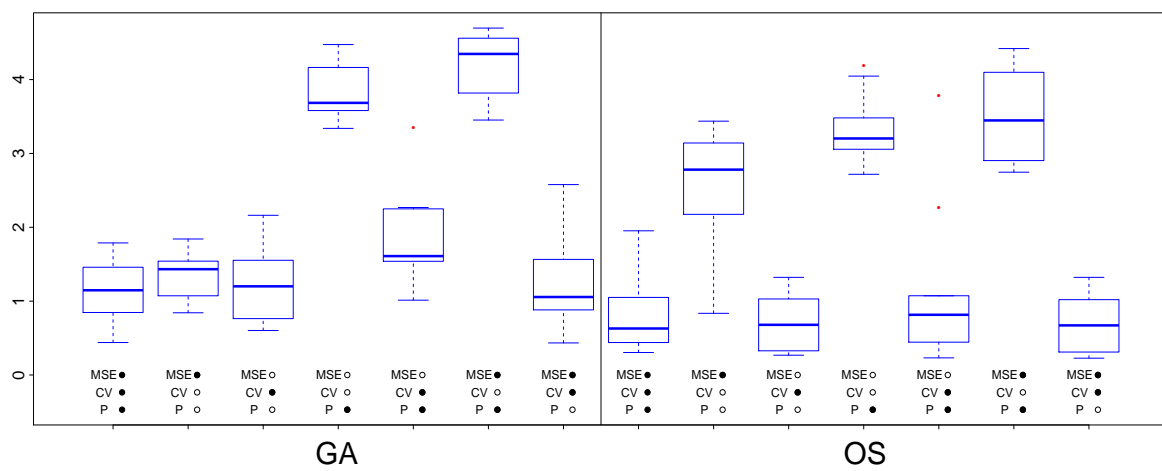


Fig. 21: Several selection criteria: D&P curved function

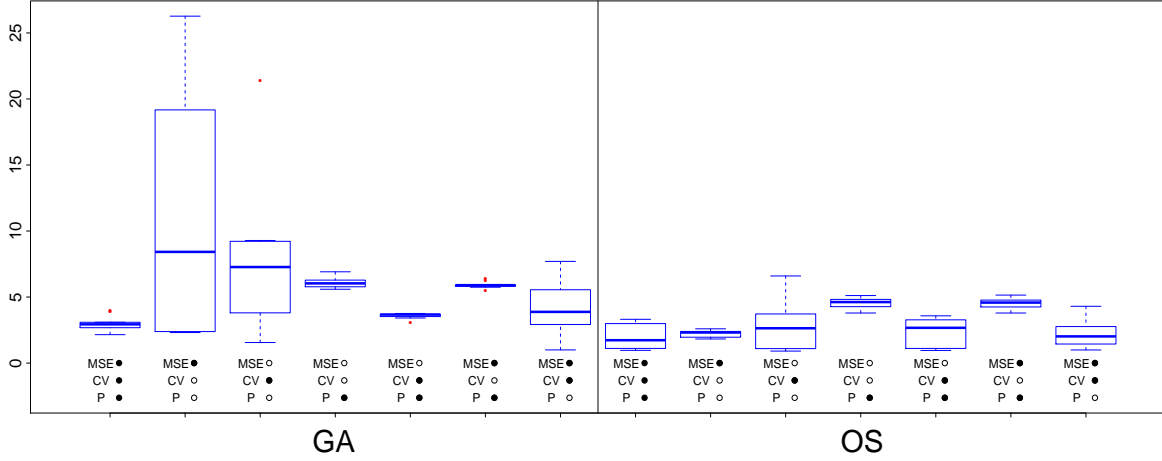


Fig. 22: Several selection criteria: D&P 8-dim function

remarks. We also warmly thank two anonymous reviewers for their valuable comments.

A Appendix: Comparison between the proposed PPS parameters and the optimal according to the sum of RMSE

B Appendix: Test functions

The equations and the input parameter space of the functions of Table 1 are defined below:

1/ Wing weight function:

Parameters: $S_w \in [150, 200]$, $W_{fw} \in [220, 300]$, $A \in [6, 10]$, $\gamma \in [-10, 10]$, $q \in [16, 45]$, $\lambda \in [0.5, 1]$, $t_c \in [0.08, 0.18]$, $N_z \in [2.5, 6]$, $W_{dg} \in [1700, 2500]$, $W_p \in [0.025, 0.08]$

For $\mathbf{x} = (S_w, W_{fw}A, \gamma, q, \lambda, t_c, N_z, W_{dg}, W_p)$

$$f_1(\mathbf{x}) = 0.036S_w^{0.758}W_{fw}^{0.758}\left(\frac{A}{\cos^2(\gamma)}\right)^{0.6}q^{0.006}\lambda^{0.04} \left(\frac{100t_c}{\cos(\gamma)}\right)^{-0.3}(N_zW_{dg})^{0.49} + S_wW_p \quad (17)$$

2/ Borehole function:

Parameters: $r_w \in [0.05, 0.15]$, $r \in [100, 50000]$, $T_u \in [63070, 115600]$, $H_u \in [990, 1110]$, $T_l \in [63.1, 116]$, $H_l \in [700, 820]$, $L \in [1120, 1680]$, $K_w \in [9855, 12045]$

For $\mathbf{x} = (r_w, r, T_u, H_u, T_l, H_l, L, K_w)$

$$f_2(\mathbf{x}) = \frac{2\pi T_u(H_u - H_l)}{\ln\left(\frac{r}{r_w}\right)\left(1 + \frac{2LT_u}{\ln\left(\frac{r}{r_w}\right)r_w^2K_w} + \frac{T_u}{T_l}\right)} \quad (18)$$

3/ Dette & Pepelyshev (2010a):

Parameters: for all $i = 1, \dots, 8$, $x_i \in [0, 1]$

$$f_3(\mathbf{x}) = 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2 + \sum_{i=4}^8 i \ln\left(1 + \sum_{j=3}^i x_j\right) \quad (19)$$

4/ Piston simulation function:

Parameters: $M \in [30, 60]$, $S \in [0.005, 0.020]$, $V_0 \in [0.002, 0.010]$, $k \in [1, 5] \times 10^3$, $P_0 \in [9, 11] \times 10^4$, $T_a \in [290, 296]$, $T_0 \in [340, 360]$

$$f_4(\mathbf{x}) = 2\pi \sqrt{\frac{M}{k + S^2 \frac{P_0 V_0}{T_0} \frac{T_a}{V^2}}} \quad (20)$$

where $V = \frac{S}{2k} \left(\sqrt{A^2 + 4k \frac{P_0 V_0}{T_0} T_a} - A \right)$

and $A = P_0 S + 19.62M - \frac{kV_0}{S}$

5/ OTL circuit function:

Parameters: $R_{b1} \in [50, 150]$, $R_{b2} \in [25, 70]$, $R_f \in [0.5, 3]$, $R_{c1} \in [1.2, 2.5]$, $R_{c2} \in [0.25, 1.2]$, $\beta \in [50, 300]$

$$f_5(\mathbf{R}, \beta) = \frac{\left(\frac{12R_{b2}}{R_{b1} + R_{b2}} + 0.74\right)\beta(R_{c2} + 9)}{\beta(R_{c2} + 9) + R_f} + \frac{11.35R_f}{\beta(R_{c2} + 9) + R_f} + \frac{0.75R_f\beta(R_{c2} + 9)}{(\beta(R_{c2} + 9) + R_f)R_{c1}} \quad (21)$$

6/ Gramacy & Lee (2009) function:

Parameters: for all $i = 1, \dots, 6$, $x_i \in [0, 1]$

$$f_6(\mathbf{x}) = \exp[\sin((0.9(x_1 + 0.48))^{10})] + x_2x_3 + x_4 \quad (22)$$

7/ Friedman function:

Parameters: for all $i = 1, \dots, 5$, $x_i \in [0, 1]$

$$f_7(\mathbf{x}) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \quad (23)$$

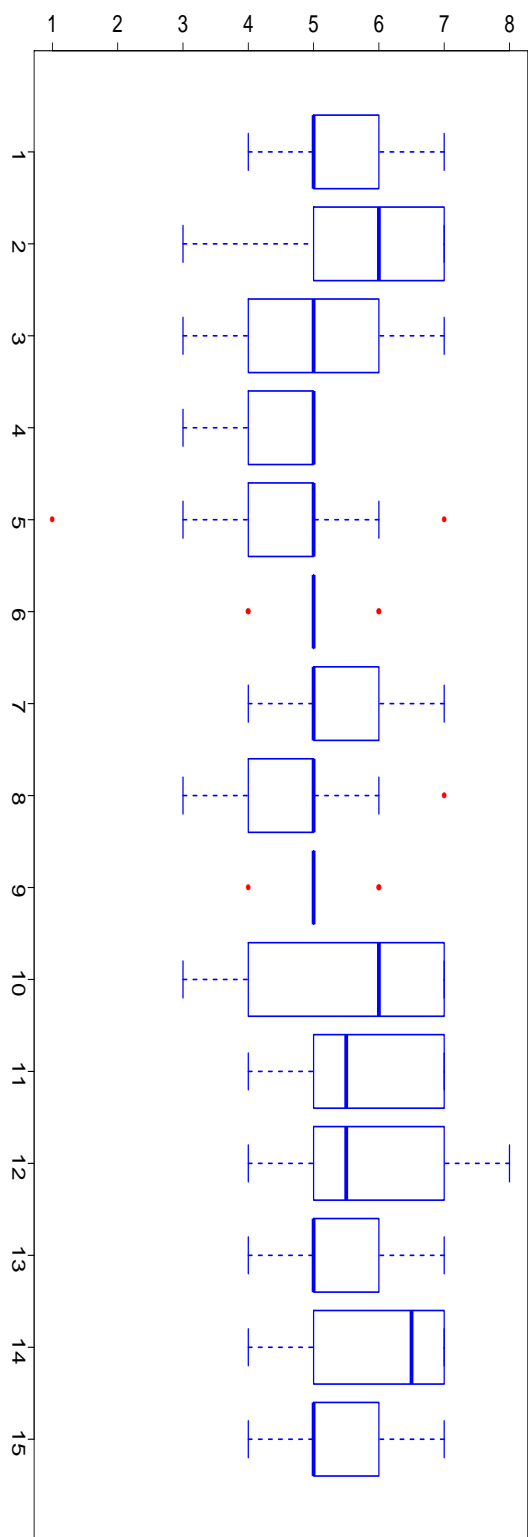


Fig. 23: Number of members in the best ensemble

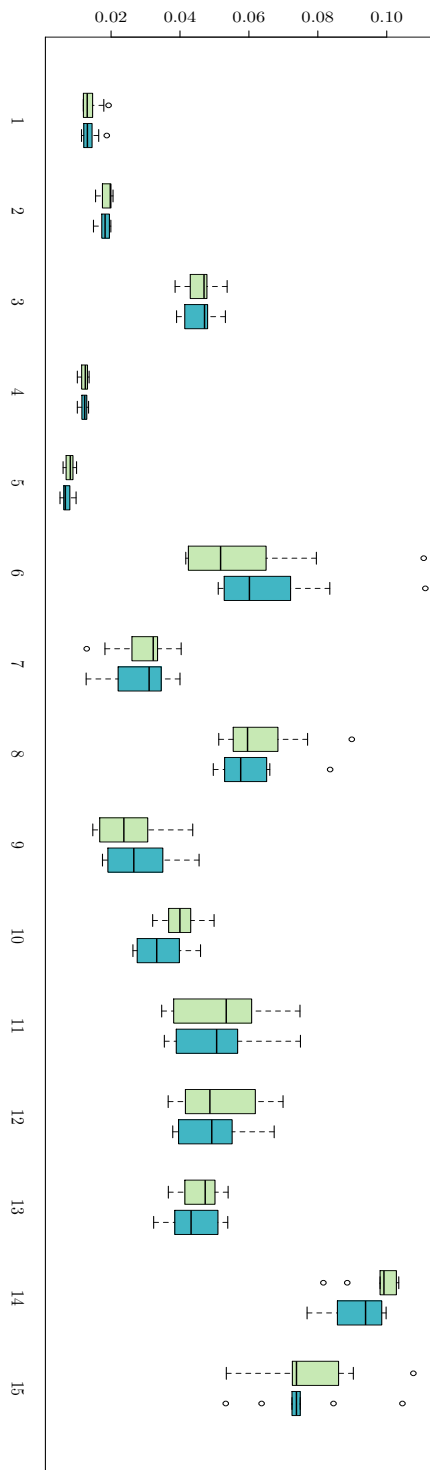


Fig. 24: The scaled RMSE for each PPS-optimal ensemble, For each function: Left: using $(\alpha, \beta, \gamma) = (1, \beta^*, \gamma^*)$ in light green. Right: using $(\alpha, \beta, \gamma) = (1, 0.5, 0.25)$ in light blue. The function number is as in Table (1)

8/ Dette & Pepelyshev exponential function:

Parameters: for all $i = 1, \dots, 3, x_i \in [0, 1]$

$$f_8(\mathbf{x}) = 100(e^{-2/x_1^{1.75}} + e^{-2/x_2^{1.5}} + e^{-2/x_3^{1.25}}) \quad (24)$$

9/ Dette & Pepelyshev curved function:

Parameters: for all $i = 1, \dots, 3$, $x_i \in [0, 1]$

$$f_9(\mathbf{x}) = 4(x_1 - 2 + 8x_2 - 8x_2^2)^2 + (3 - 4x_2)^2 + 16\sqrt{x_3 + 1}(2x_3 - 1)^2 \quad (25)$$

10/ Lim non-polynomial function:

Parameters: $x_1, x_2 \in [0, 1]$

$$f_{10}(\mathbf{x}) = \frac{1}{6}[(30 + 5x_1 \sin(5x_1))(4 + \exp(-5x_2)) - 100] \quad (26)$$

11/ Currin exponential function:

Parameters: $x_1, x_2 \in [0, 1]$

$$f_{11}(\mathbf{x}) = [1 - \exp(-\frac{1}{2x_2})] \times \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20} \quad (27)$$

12/ Franke function:

Parameters: $x_1, x_2 \in [0, 1]$

$$f_{12}(\mathbf{x}) = 0.75 \exp(-\frac{(9x_1 - 2)^2 + (9x_2 - 2)^2}{4}) + 0.75 \exp(-\frac{(9x_1 + 2)^2}{49} - \frac{9x_2 + 1}{10}) + 0.5 \exp(-\frac{(9x_1 - 7)^2}{4} - \frac{(9x_2 - 3)^2}{4}) + 0.2 \exp(-9(x_1 - 4)^2 - (9x_2 - 7)^2) \quad (28)$$

13/ Gramacy & Lee (2008) function:

Parameters: $x_1, x_2 \in [-2, 6]$

$$f_{13}(\mathbf{x}) = x_1 \exp(-x_1^2 - x_2^2) \quad (29)$$

14/ Sasena function:

Parameters: $x_1, x_2 \in [0.0, 5]$

$$f_{14}(\mathbf{x}) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(2 - x_2)^2 + 7 \sin(0.5x_1) \sin(0.7x_1x_2) \quad (30)$$

15/ Gramacy & Lee (2012) function:

Parameters: $x \in [0.5, 2.5]$

$$f_{15}(x) = \frac{\sin(10\pi x)}{2x} + (x - 1)^4 \quad (31)$$

References

- Acar E, Rais-Rohani M (2009) Ensemble of metamodels with optimized weight factors. *Structural and Multidisciplinary Optimization* 37(3):279–294
- Akaike H (1974) A new look at the statistical model identification. *Automatic Control, IEEE Transactions on* 19(6):716–723
- Arlot S, Celisse A (2010) A survey of cross-validation procedures for model selection. *Statist Surv* 4:40–79
- Chen PW, Wang JY, Lee HM (2004) Model selection of svms using ga approach. In: *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, IEEE*, vol 3, pp 2035–2040
- Duchon J (1977) Splines minimizing rotation-invariant semi-norms in sobolev spaces. In: *Constructive theory of functions of several variables*, Springer, pp 85–100
- Efron B, Tibshirani RJ (1993) *An introduction to the bootstrap*
- Forrester AI, Keane AJ (2009) Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences* 45(1):50–79
- Friedman JH (1991) Multivariate adaptive regression splines. *The annals of statistics* pp 1–67
- Goel T, Haftka RT, Shyy W, Queipo NV (2007) Ensemble of surrogates. *Structural and Multidisciplinary Optimization* 33(3):199–216
- Gorissen D, Dhaene T, Turck FD (2009) Evolutionary model type selection for global surrogate modeling. *The Journal of Machine Learning Research* 10:2039–2078
- Hastie T, Tibshirani R, Friedman J (2009) *The elements of statistical learning*, vol 2. Springer
- James G, Witten D, Hastie T, Tibshirani R (2013) *An introduction to statistical learning*, vol 6. Springer
- Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, Morgan Kaufmann Publishers Inc., IJCAI'95, pp 1137–1143
- Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. *Mathematics of computation* 37(155):141–158
- Lessmann S, Stahlbock R, Crone SF (2006) Genetic algorithms for support vector machine model selection. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE*, pp 3063–3069
- Matheron G (1963) Principles of geostatistics. *Economic geology* 58(8):1246–1266
- McKay MD, Beckman RJ, Conover WJ (1979) Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245
- Müller J, Piché R (2011) Mixture surrogate models based on dempster-shafer theory for global optimization problems. *Journal of Global Optimization* 51(1):79–104
- Nguyen H, Couckuyt I, Knockaert L, Dhaene T, Gorissen D, Saeys Y (2011) An alternative approach to avoid overfitting for surrogate models. In: *Simulation Conference (WSC), Proceedings of the 2011 Winter*, pp 2760–2771
- Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2005) Surrogate-based analysis and optimization

- tion. *Progress in aerospace sciences* 41(1):1–28
- Schwarz G, et al (1978) Estimating the dimension of a model. *The annals of statistics* 6(2):461–464
- Shi L, Yang R, Zhu P (2012) A method for selecting surrogate models in crashworthiness optimization. *Structural and Multidisciplinary Optimization* 46(2):159–170
- Smola A, Schlkopf B (2004) A tutorial on support vector regression. *Statistics and Computing* 14(3):199–222
- Stone M (1974) Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society Series B (Methodological)* pp 111–147
- Tomioka S, Nisiyama S, Enoto T (2007) Nonlinear least square regression by adaptive domain method with multiple genetic algorithms. *IEEE Transactions on Evolutionary Computation* 11(1):1–16
- Viana FA, Haftka RT, Steffen V (2009) Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. *Structural and Multidisciplinary Optimization* 39(4):439–457
- Viana FA, Venter G, Balabanov V (2010) An algorithm for fast optimal latin hypercube design of experiments. *International journal for numerical methods in engineering* 82(2):135–156
- Watson DF (1981) Computing the n-dimensional delaunay tessellation with application to voronoi polytopes. *The computer journal* 24(2):167–172
- Zerpa LE, Queipo NV, Pintos S, Salager JL (2005) An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. *Journal of Petroleum Science and Engineering* 47(3):197–208
- Zhang C, Shao H, Li Y (2000) Particle swarm optimisation for evolving artificial neural network. In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on, IEEE*, vol 4, pp 2487–2490
- Zhou X, Jiang T (2016) Metamodel selection based on stepwise regression. *Structural and Multidisciplinary Optimization* 54(3):641–657
- Zhou XJ, Ma YZ, Li XF (2011) Ensemble of surrogates with recursive arithmetic average. *Structural and Multidisciplinary Optimization* 44(5):651–671