XL CILAMCE
IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
NOVEMBER 11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

COMPARATIVE STUDY ON MULTIPLE WIDTH-DEFINING METHODS FOR RADIAL BASIS FUNCTIONS

Leonardo G. Ribeiro
leogoncalvesrib@gmail.com
Marina A. Maia
marinaalvesmaia@gmail.com
Evandro Parente Junior
evandro@ufc.br
Antônio M. C. de Melo
macario@ufc.br
Laboratório de Mecânica Computacional e Visualização (LMCV), Departamento de Engenharia Estrutural e Construção Civil, Universidade Federal do Ceará, Campus do Pici, Centro de Tecnologia, Bloco 728, 60455-760, Fortaleza, Ceará, Brasil

Abstract. In structural problems, numerical methods such as the Finite Element Method are often used due to the scarce and limited applicability of analytical methods. In these cases, the design optimization may become computationally costly and the time consumed starts to be a hindrance. To overcome this problem, a significant effort has been made by researchers to understand and improve the so-called surrogate models. Surrogate models provide computational efficiency by using a few samples from the true function to build an approximated response surface to predict points in the design space not yet evaluated during the optimization process. This approximated surface may also be improved at each generation with the addition of new samples in regions of interest on a methodology known as Sequential Approximate Optimization (SAO). In this context, the Radial Basis Functions (RBF) are a powerful and robust surrogate model while keeping implementation simple. The Gaussian function is often chosen as the basis function despite uncertainty on the definition of one of its main parameters: the kernel width (σ). This paper performed a comparative study on different methods to estimate the width parameter using two types of solutions: closed-form expressions proposed by different researchers in the last few years and direct search methods. The efficiency of each of these approaches is assessed using metrics such as the number of high fidelity model evaluations and the error at the end of each optimization.

Keywords: Surrogate modeling, Radial basis functions, Kernel width.

1 Introduction

With the ever-increasing processing capacity of computers, engineers started to shift their attention to not only finding a feasible design but to optimize it as well. That way, most structural problems focus on trying to find the best possible design for a structure while respecting a set of constraints. Although in some cases an analytical solution might aid the structural analysis, in practical applications it is necessary to use numerical methods, such as the Finite Element Method (FEM). However, this approach may result in time-consuming analyses. Therefore, the use of surrogate modeling has become in recent years an active research field. Surrogate models such as the Radial Basis Functions (RBF) [1, 2], the Support Vector Regression (SVR) [3, 4] and Kriging [5, 6] have shown good performance (i.e., low computational cost and good accuracy).

Wang and Shan [7] highlight that these models are mostly used to approximate computational-intensive processes, to explore the design space and to assist in design optimization. Comparative studies suggest that the RBF is a good choice to approximate higher-order non-linear functions, whereas Kriging works best with low-order ones [8, 9]. Forrester et al. [10] point out that, due to the necessity of parameter-optimization, Kriging might become costly on high-dimensional problems, suggesting the RBF is a more appropriate choice.

As for the RBF, several basis functions can be used to interpolate the sampling points. These functions are usually classified as fixed bases (i.e., no hyper-parameter needs to be defined) or parametric bases [10]. While the former presents a simpler formulation, the latter allows for better control of its characteristics, as well as the possibility of a more generalized method. Mongillo [11] carried out a comparison between three types of parametric basis, the multi-quadratic, the inverse multi-quadratic and the Gaussian kernels, and, while acknowledging that this choice had deep implications on the surrogate model performance, all three basis achieved low errors when σ was optimized. Mehmani et al. [12] presented the COSMOS framework to choose the set (model type, basis function and hyper-parameter values) that presents the smallest error. It is worth mentioning that although this alternative provides a good generalization of the surrogate, it might become extremely time-consuming. Acar [13] proposed a formulation where both the shape parameter and the model weights are optimized at the same time. However, this method may also hinder the optimization process and increase the cost of the algorithm.

The Gaussian kernel is one of the most popular basis functions used in the literature due to the good results shown in comparative studies. In addition to that, models based on the Gaussian process also allow the calculation of an estimate of the error of the model, which might be important for Sequential Approximate Optimization (SAO) [10]. This kernel type is also often used by the Kriging predictor.

However, despite the robustness of the RBF model, it is still unclear how to best define the shape parameter σ, also referred to as the width of the kernel. The most straightforward approach is the Leave-One-Out Cross-Validation (LOOCV), usually adopted when there are no concerns related to the cost of building and evaluating the model. This approach has been successfully used by numerous researchers [12, 14]. However, on larger sample sizes and high-dimensional problems, this method might become too expensive. An alternative is the k-Fold Cross-Validation (k-FCV), where the validation is performed separating the samples in k groups. Müller and Shoemaker [15] performed a hybrid method, in which the LOOCV is used while the sampling size is less than 50 and the k-FCV is applied after that point with k being adaptively adjusted according to the sampling plan size.

However, cross-validation techniques may become a problem when large design spaces are considered. To address these problems, researchers usually make use of simple mathematical formulations to define the basis functions widths. Most derivations come from the formulation presented on Haykin [16] with the consideration of a few improvements [17–19] that shall be discussed later on. Wu et al. [20] presented a novel method to define the width parameter based on the local density of sampling points, showing good performance. However, the *ad-hoc* nature of these formulations makes the choice for a method for defining the widths an unknown and doubtful field to the researches. In addition to that, a Sequential Approximate Optimization (SAO) technique is employed to improve the surrogate model accuracy with the addition of new sample points.

This paper focus on the comparison between several width-defining methods for the Gaussian kernel aiming propositions to facilitate the choice of this parameter for future research. The rest of the paper is organized as follows: In Section 2 the main features of the Surrogate Modeling are introduced, including the sampling plan definition and the infill criteria considered for the comparisons. In Section 3 the mathematical formulation of the RBF model are presented. In Section 4 the importance of the shape parameter and the different types of width-defining methods considered is discussed. The results obtained are presented in Section 5 and finally, in Section 6 the main conclusions are brought together.

2 Surrogate modeling

Surrogate modeling is a technique that tries to emulate the behavior of a given function at a lower computational cost based on the function values at a set of sampling points [10]. In this approach, the initial data are usually determined by a Design of Experiments (DoE) [21]. For computer experiments, the basic argument is that samples must be uniformly distributed [9, 22], although there is, in fact, no clear correlation between the prediction error of the surrogate model and the uniformity of the samples [23, 24].

Table 1. Definition of sample size

Type of function	Low sample size	Medium sample size	High sample size
Low dimension	$1.5K$	$3.5K$	$6K$
High dimension	$1.5K$	$2.5K$	$5K$

In this paper, the deterministic Hammersley Sequence Sampling (HSS) is used to determine the initial sampling plan, defined by the sample vector $\mathbf{x}$ and its respective objective functions $\mathbf{y}$. To better establish the model, the sample vector should be normalized before computed [10]. The number of points (n) used was defined as proposed by Amouzgar and Strömberg [25] and is summarized in Table 1, being a function of a parameter K, defined as:

$$K = \frac{(m+1)(m+2)}{2}. \tag{1}$$

A major flaw in using DoE to define the sample vector is that there is no consideration over the behavior of the objective function. In most cases, however, that is the best the user can do at that point [14]. One of the advantages of using surrogate models in optimization problems is the possibility of, throughout the optimization, being able to add new sampling points on regions of interest [10, 26]. This is why this paper will employ the Sequential Approximate Optimization (SAO) in order to further improve the model's prediction. This technique was first proposed by Schmit and Farshi [27] in order to continuously update the trust regions considered throughout the optimization. In the present work, the SAO will be used with the sole purpose of defining regions of interest and performing the insertion of new and more efficient sampling points. These points are chosen using an *infill criterion*.

There are two main strategies when choosing an infill criterion: the *exploitation* approach, when one tries to improve the prediction in regions close to the current optimal point, and the *exploration*, when one aims to sample unseen areas where prediction is highly uncertain. There are also methods that try to combine both features in order to increase efficiency [26].

In this paper, an approach very similar to the one proposed by Kitayama et al. [28] will be used. At each generation, two points are added to the initial sampling plan. The first one refers to the exploitation and the best non-repeated individual found by the optimization up to that iteration is added and the second point is related to the exploration. The exploration sample point is determined by using a Density

Function, which tries to represent the distance between the sampling points and any other point over the design space. Kitayama et al. [28] propose a simple formulation in order to take advantage of the RBF: basically, the model is built the same way as the surrogate model itself, but, in this case, the $\mathbf{y}$ vector is a vector of ones. In this paper, the σ vector of the density function is taken as the one obtained by the true function surrogate model.

Figure 1 depicts the Density Function for a 1D problem. The blue circles represent the samples considered while the red dot is the sample obtained by the minimization of the Density Function (i.e., leading to points in sparse areas). An interesting aspect of this approach is that it usually performs the addition of points at the borders of the design space on earlier generations.

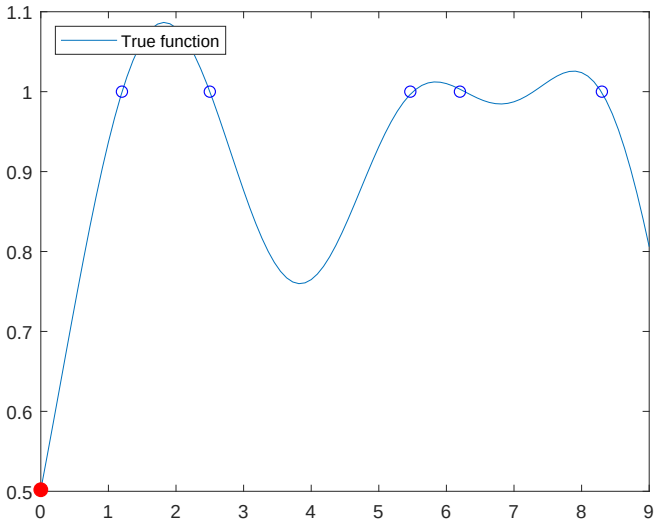
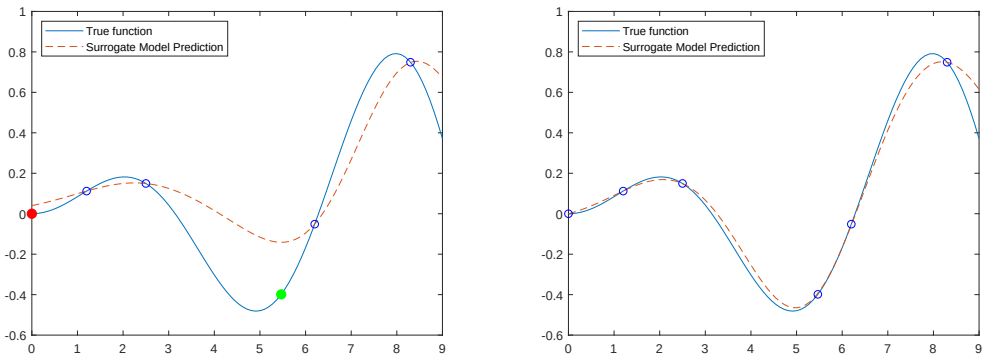


Figure 1. Example of a Density Function for a 1D problem

Figure 2(a) illustrates a surrogate model and the effect in the model's accuracy due to the addition of two points (see Fig. 2(b)). The blue circles represent the initial sample points, while the green and red dots are the points to be added, related to exploitation and to exploration respectively. This addition not only improves the surrogate accuracy through the whole design domain, but it also helps the algorithm to find solutions closer to the optimal global of the true function.



(a) Initial surrogate model and new sample points

(b) Updated surrogate model

Figure 2. Addition of new sample points to the surrogate model

3 Radial Basis Functions

The Radial Basis Function (RBF) model was initially proposed by Hardy [29] to assist data interpolation in topography and geography problems. It is based on the idea of conceiving basis functions ψ capable of interpolating a set of data given *a priori* in order to build an estimate for the entire design space given by:

$$\hat{y} = \sum_{i=1}^{b} w_i \psi(\|\mathbf{x} - \mathbf{c}_i\|) \tag{2}$$

where b refers to the number of bases, w_i refers to the weight corresponding to basis i, while $\mathbf{c}_i$ refers to its center. In this paper, the number of bases was considered equal to the sample size.

One of the advantages of the RBF is the possibility of choosing which basis function ψ to be used. As briefly discussed in Section 1, the *fixed basis* provides a simpler formulation, simplifying the modeling process and allowing a faster model building. Examples of these types are the linear ($\psi(r) = r$), the cubic ($\psi(r) = r^3$) and the thin plate spline basis functions ($\psi(r) = r^2 ln(r)$). On the other hand, the *parametric basis* requires the definition of a shape parameter (σ) and allows better control of the surrogate model performance. However, this does not come without a price: the user must define the shape parameter of this function, which might become a hard and time-consuming task in some cases. Examples of these are the multi-quadratic ($\psi(r) = \sqrt{(r^2 + \sigma^2)}$), the inverse multi-quadratic ($\psi(r) = \sqrt{(r^2 + \sigma^2)}^{-1}$) and the widely known Gaussian kernel ($\psi(r) = e^{-(r^2/(\sigma^2))}$) [2]. Fig. 3 illustrates the behavior of different basis functions considering $\sigma = 1.0$. In this paper, the Gaussian function is adopted.

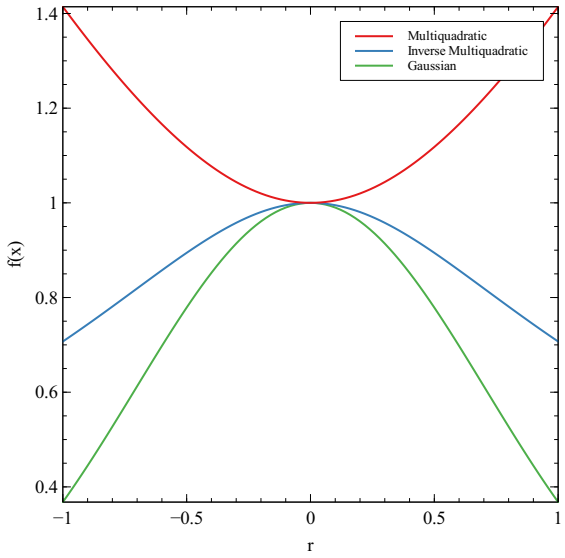


Figure 3. Behavior of different kernel types, depending on the r parameter

The Gaussian kernel depends on the Euclidean distance r between two points (the center $\mathbf{c}_i$ and the point $\mathbf{x}$ that will be evaluated by the RBF) and the corresponding width for the i-th basis. The RBF formulation can be written in matrix form as:

$$\hat{y} = \mathbf{w}^T \boldsymbol{\psi}. \tag{3}$$

This way, there are two variable sets that have yet to be defined: the weights $\mathbf{w}$ and the widths $\boldsymbol{\sigma}$. One of the advantages of using RBF over other surrogate models is the ease achieved in calculating its weight

vector. Its basic idea is that, at sampling points, the response given by the surrogate model must be equal to $\mathbf{y}$ (i.e., $\hat{\mathbf{y}} = \mathbf{y}$). Thus, the data interpolation can be written as:

$$\boldsymbol{\Psi}\mathbf{w} = \mathbf{y}. \tag{4}$$

Here, Ψ is the Gram matrix, based on the sample vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$, e.g. if a uniform $\boldsymbol{\sigma}$ vector is used, the Gram matrix is defined as:

$$\boldsymbol{\Psi} = \begin{bmatrix} \psi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & ... & \psi(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \psi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & ... & \psi(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ \vdots & \ddots & \vdots \\ \psi(\|\mathbf{x}_n - \mathbf{x}_1\|) & ... & \psi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix}. \tag{5}$$

This formulation, although successful in many cases, might result in bad conditioning of the matrix if the sampling points are too close to each other [16]. To overcome this issue, Kitayama and Yamazaki [19] proposed an alternative which uses a regularization parameter λ to calculate the quadratic error as:

$$E = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \sum_{j=1}^{b}\lambda {w_j}^2. \tag{6}$$

It is worth noting that λ should be small enough to not interfere too much on the model's prediction, but high enough for it to be effective. The weight vector that minimizes Eq. (6) can be calculated as:

$$\mathbf{w} = (\boldsymbol{\Psi}^T\boldsymbol{\Psi} + \mathbf{I}\lambda)^{-1}\boldsymbol{\Psi}^T\mathbf{y} \tag{7}$$

where $\mathbf{I}$ is an identity matrix.

4 The width parameter

The only missing parameter for the building and evaluation of the model is the width σ. As mentioned earlier, there is no clear way of defining this value, which is why some authors use *ad-hoc* techniques. However, the parameter is of utmost importance for model prediction [14] since the width σ_i is related to the influence area concerning the basis i, as illustrated in Fig. 4.

Figure 5 depicts an example function for different widths. Note how the prediction goes from looking like a "needles in the haystack" function to a smoother and flatter one. It is needless to say that even though for this example the optimal width seems to be around 0.1972, it might not be the case for other functions. This value was picked by performing one of the analytical formulations, proposed by Nakayama et al. [17]. Two factors can affect the optimal value of the parameter: the true function and the data set used to build the interpolation [11]. The sample size and the number of variables are usually representative of the latter.

Two main approaches can be taken here: the first one generates a uniform vector, where all bases i share the same width. However, this might not be ideal, especially when working with non-uniform sample sets [19]. The second approach tries to create a non-uniform vector in which a different σ is assigned for each basis. That being said, to estimate the $\boldsymbol{\sigma}$ vectors, researchers can either use analytical formulations or Cross-Validation methods.

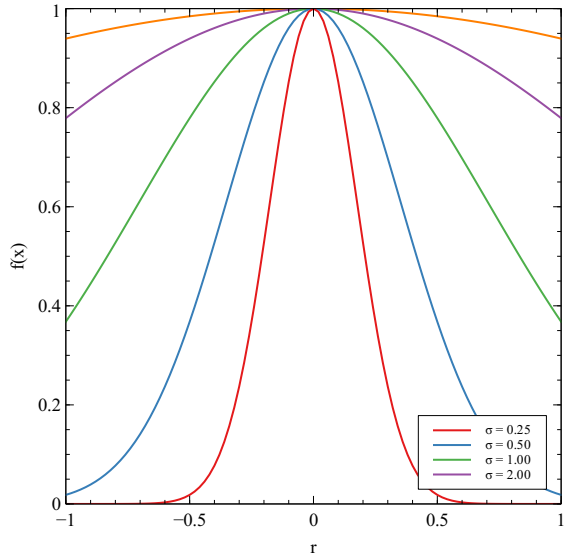


Figure 4. Behavior of the Gaussian function, depending on the σ parameter

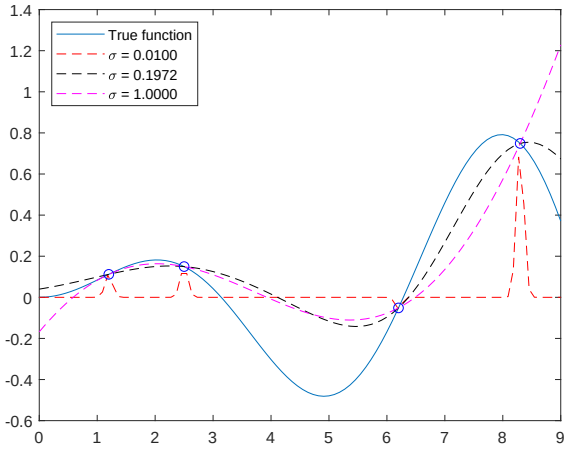


Figure 5. Prediction of the model, depending on the σ parameter

4.1 Analytical formulations

The analytical formulations are among the most simple and easy-to-use methods to define the width of Gaussian kernels. However, they are often treated with disbelief due to their *ad-hoc* nature. Most of them try to improve the following formulation presented on Haykin [16]:

$$\sigma = \frac{d_{max}}{\sqrt{2n}}, \tag{8}$$

where d_{max} refers to the maximum distance between two sampling points in the design space and n to the number of samples, the same as the number of bases in our case. Nakayama et al. [17] proposed an alternative in order to also take into consideration the number of design variables m as:

$$\sigma = \frac{d_{max}}{\sqrt[m]{mn}}. \tag{9}$$

Kitayama and Yamazaki [19] also proposed a new formulation, since according to the authors, the last proposal overestimates the parameter by underestimating the importance of the number of design variables m:

$$\sigma = \frac{d_{max}}{\sqrt{m}\sqrt[m]{n}}. \tag{10}$$

Note that all three formulations presented so far provide the same values for a two-dimensional design space. In addition to those equations, Kitayama and Yamazaki [19] also proposed a way to better study non-uniform samples by generating a σ vector with different values of width for each basis. In this paper, to better distinguish between the two proposals made by Kitayama and Yamazaki [19], the former one will be named as Uniform Kitayama, while the Non-Uniform Kitayama is presented as:

$$\sigma_i = \frac{d_{i,max}}{\sqrt{m}\sqrt[m]{n-1}} \tag{11}$$

where $d_{i,max}$ is the maximum distance from basis i and any other sample. Finally, the authors proposed a technique named *Adaptive Scaling*, where the whole $\boldsymbol{\sigma}$ vector is evaluated as shown in Eq. (11) and should be continuously multiplied by a scalar δ until the lowest value in $\boldsymbol{\sigma}$ is higher than 1.0. In this paper, δ was set to 1.1, as originally proposed by the authors.

4.2 Cross-validation methods

Although the analytical propositions do usually make satisfactory predictions about the optimal basis widths, these approaches do not perform any consideration about the behavior of the function to be estimated. Thus, despite the lower computational cost of analytical formulations, it is usual to perform the optimization of this parameter. Acar [13] goes even further and proposes a formulation where the widths and the weights are optimized at the same time. However, the usual way to perform this is through Leave-One-Out Cross-Validation (LOOCV). In this proposal, a number n_w of σ are tested. For each one of them, n surrogate models are built, each excluding one different sample from model construction. This sample is then used as validation. To compare its value with the one evaluated from the model built, the Mean Squared Error (MSE) is calculated as:

$$MSE = \frac{(y_i - \hat{y}_i)^2}{n_v} \tag{12}$$

where n_v refers to the number of samples used as validation at each iteration. The vector $\boldsymbol{\sigma}$ with the lower sum of MSEs is then chosen for model building and evaluation. Figure 6 summarizes these instructions.

The stopping criteria shown in Fig. 6 may be either related to a minimum accuracy value achieved or simply to the number of widths (n_w) tried. Sobester et al. [14] propose an approach where uniform $\boldsymbol{\sigma}$ vectors are tested through LOOCV, where direct search is run over the domain $[10^{-2}, 10^{1}]$ to find the best σ, while Mehmani et al. [12] search σ values over the domain $[10^{-1}, 3]$ using an optimization algorithm.

Remark 1: Direct Search Domain. On their paper, Sobester et al. [14] use a slightly different formulation for the Gaussian function, $\psi(r) = e^{-(r^2/(2\sigma^2))}$. To obtain the same results as the authors, in this paper one works on the domain $[10^{-2}, 10^{1}] \cdot \sqrt{2}$, considering $n_w = 20$, uniformly distributed in a logarithm scale.

This approach offers a generalization with a certain level of reliability but at the price of a higher computational cost, since it requires $n \cdot n_w$ model buildings and evaluations to determine σ, which may

render the process unfeasible, particularly for large sample sizes. It is important to notice that the former is the most expensive phase since it involves many matrix operations, while the latter involves more basic operations.

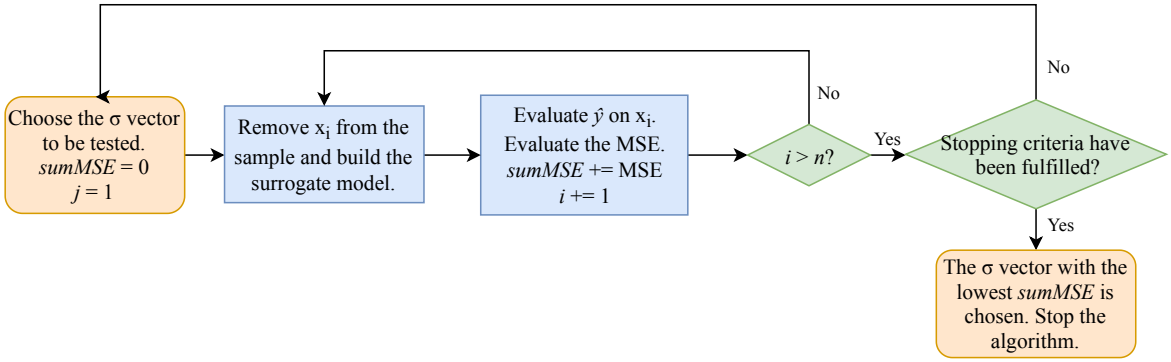


Figure 6. Leave-One-Out Cross-Validation flowchart for width definition

An efficient alternative approach is the use of the k-Fold Cross-Validation (k-FCV). It is very similar to the LOOCV but instead of building a different model for each sampling point, this technique randomly organizes the sampling points in k groups. Each surrogate model is then built upon all groups but one, which should be used to validate the model. The process is ended when all groups have been put to test. Figure 7 presents the scheme of operation of this approach. Note how it is similar to the one presented in Fig. 6. Actually, if the number of groups k is set as equal as n, there is no difference between the two methods. It should be pointed out that, if $k \geq n$, the algorithm should set $k = n$.

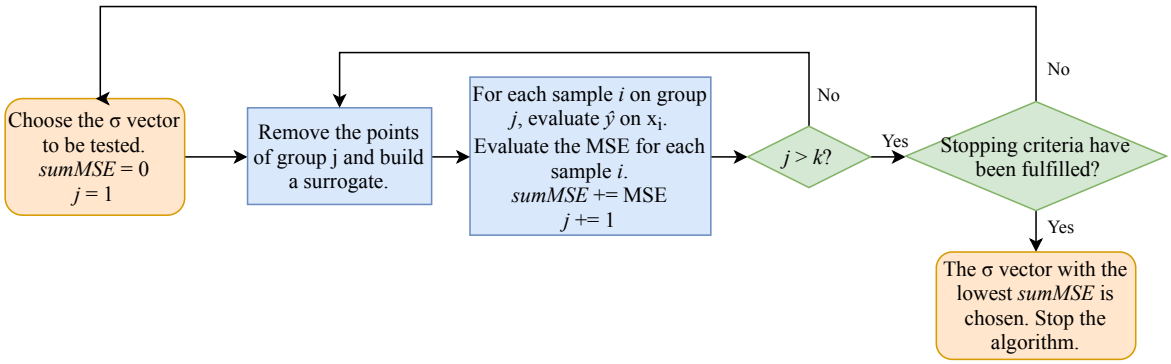


Figure 7. k-Fold Cross-Validation flowchart for width definition

That way, the method greatly reduces the required number of model buildings. While LOOCV needs to build $n \cdot n_w$ surrogates, k-FCV performs this procedure only $k \cdot n_w$ times. It is important to note that, although the number of evaluations is the same as in LOOCV, this step has a low computational cost when compared with model building. In the applications, two variations are considered: $k = 10$ and $k = 5$.

It is noticeable how the analytical formulations try to simplify the task, given that even though the optimal value of σ varies for each function and depends on $f(\mathbf{x})$, the closed-form equations do not consider this aspect. Despite the shortcomings, numerical studies suggest that these equations may offer predictions with reasonable accuracy [1, 17, 19, 30] at a low cost, since they did not require any model buildings and evaluations to compute the width (σ).

On the other hand, both LOOCV and k-FCV get more expensive as the number of sampling points grows. On DoE, the risen in dimensionality is heavily related to an increase in the number of sampling

points: as the design space gets bigger, the sample size increases exponentially. This problem is commonly known as the *curse of dimensionality* and is a major concern in designing an efficient surrogate model [10]. Furthermore, as new points are added to the sample, the cost of the cross-validation methods increases as well.

In this work, several width-defining methods are compared considering their optimization performance. Hopefully, this will help guide future researchers on which method best suits their needs.

5 Examples

The algorithm was implemented on the Biologically Inspired Optimization System (BIOS), a software written in C++ language using the Object-Oriented Programming paradigm. BIOS has been developed at the Laboratório de Mecânica Computacional e Visualização (LMCV), from the Universidade Federal do Ceará (UFC). In the examples, the proposition made by Nakayama et al. [17] is referred to as NAK, while UKIT, KIT and ASKIT refer to the methods proposed by Kitayama and Yamazaki [19], the Uniform Kitayama, Non-Uniform Kitayama and the Non-Uniform Kitayama associated with the Adaptive Scaling technique, respectively.

A comparison between the true response at the optimum point found at the end of each optimization and the actual optimum of the true function will be evaluated using:

$$error = \frac{f_{min\ HFM}}{f_{opt}}. \tag{13}$$

Thus, the closer the error is to 1, the better is the answer. Note that, in optimization problems, the accuracy of the whole surface is not a concern as long as the algorithm is capable of reaching the optimum response.

5.1 Minimization of the Peaks function

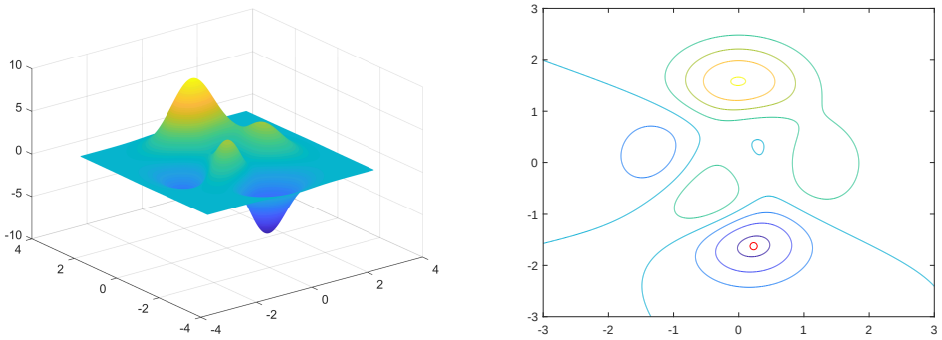
Peaks is a two-dimensional function with a couple local minimums and maximums. The function is evaluated over the domain $[-3, 3]^2$ and may be described as:

$$f(x) = 3(1-x_1)^2 e^{-\left({x_1}^2+(x_2+1)^2\right)} - 10\left(\frac{x_1}{5} - {x_1}^3 - {x_2}^5\right) e^{-\left({x_1}^2+{x_2}^2\right)} - \frac{1}{3} e^{-\left({x_2}^2+(x_1+1)^2\right)}. \tag{14}$$

Figure 8 illustrates its behavior over the design space as well as the location of the global minimum, on $f(0.228, -1.626) = 6.551$.

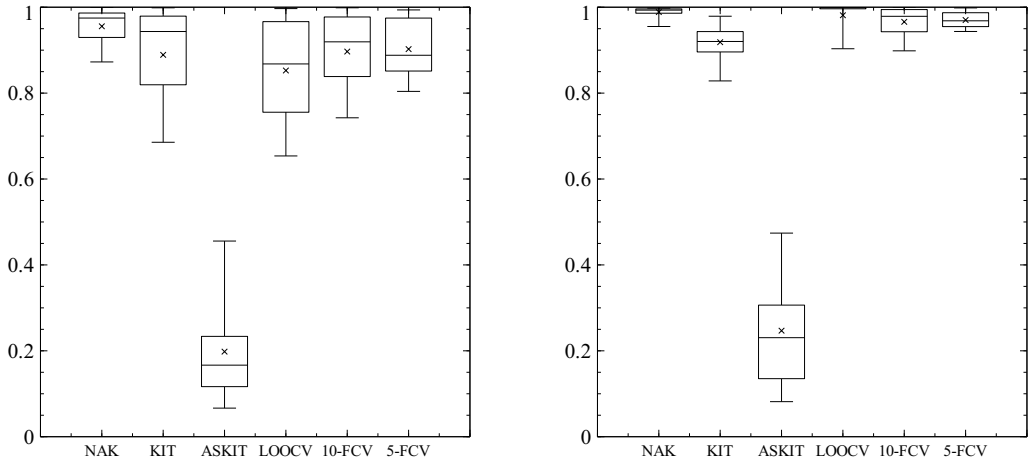
The optimization was performed using LOOCV, 10-FCV, 5-FCV and the methods proposed by Nakayama et al. [17] and Kitayama and Yamazaki [19], presented in Eq. (9) and Eq. (11) respectively. This last formulation was used both directly and using the Adaptive Scaling technique. Figure 9 depicts the results found for each method normalized by the optimum point considering 10 runs and three value for the maximum number of generations: 25, 50 and 75. The x mark inside each *Boxplot* represents the average value found.

Due to the low number of generations used in Fig. 9(a), the optimal value found presented a high variation, even when adopting the same method. It is interesting to note that both 10-FCV and 5-FCV seem to have found better results than LOOCV in general and, more importantly, this method presents a lower computational cost. The method proposed by Nakayama et al. [17] seems to have shown the best results, even though it may seem simplistic at first glance. The same trend can be observed in Fig. 9(b) and Fig. 9(c). Even though LOOCV results improve substantially, 10-FCV and 5-FCV also show good performances on these, at a much lower cost.

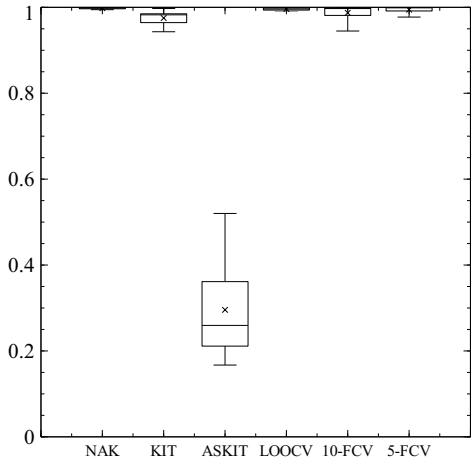


(a) Peaks HFM (b) Contour lines

Figure 8. Peaks function



(a) 25 generations (b) 50 generations



(c) 75 generations

Figure 9. Normalized results for Peaks function optimization

Another important observation is the poor performance of the Adaptive Scaling technique. This behavior may be explained by studying how the $\boldsymbol{\sigma}$ vector changed over the optimization for LOOCV, 5-FCV and the method proposed on Nakayama et al. [17] (see Fig. 10). It is clear that, while the optimum

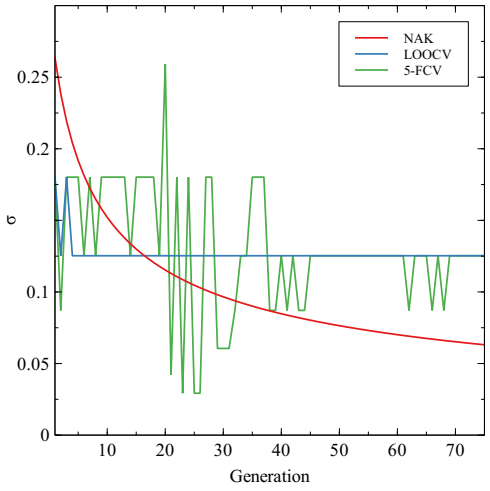


Figure 10. Width values used for the Peaks function

σ value might be around 0.13, the Adaptive Scaling technique keeps pushing all values to be greater than 1.0, which causes the poor results of the method. This trend will continue to happen on low-dimensional examples due to the smaller design space.

Also, both cross-validation methods did not show a perceptible continuous decrease of the σ values used, expected due to the increase in the number of samples over the optimization. This is probably due to the way the σ values were discretized to perform the direct search of these methods, as proposed by Sobester et al. [14].

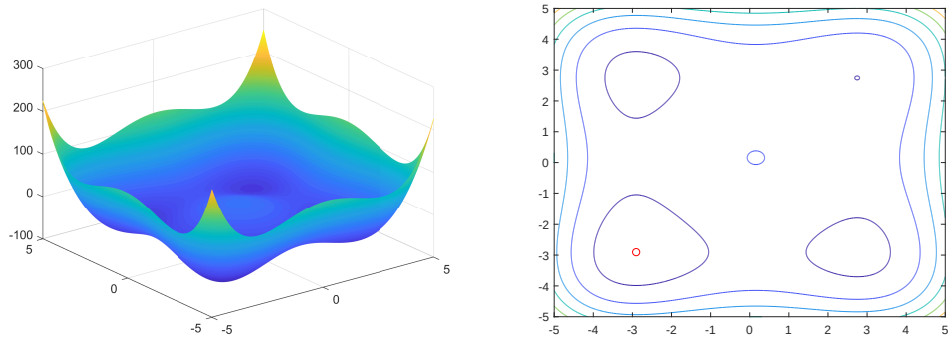
5.2 Minimization of the Styblinski-Tang function

The Styblinski-Tang is an m-dimensional function with some local minima and maxima, albeit with a flatter shape than the previous one. As m increases, so does the number of local optima, which may hinder the optimization process. Its general formulation is presented in Eq. (15), while Fig. 11 presents the HFM of the two-dimensional version over its domain as well as the location of the global minimum, on $f(-2.904, -2.904) = -39.661 \cdot m$. In this paper, this function will be tested with m equal to 4, 8, 12 and 16 and evaluated over the domain $[-5, 5]^m$.

$$f(x) = \frac{1}{2}\sum_{i=1}^{m}(x_i^4 - 16x_i^2 + 5x_i). \tag{15}$$

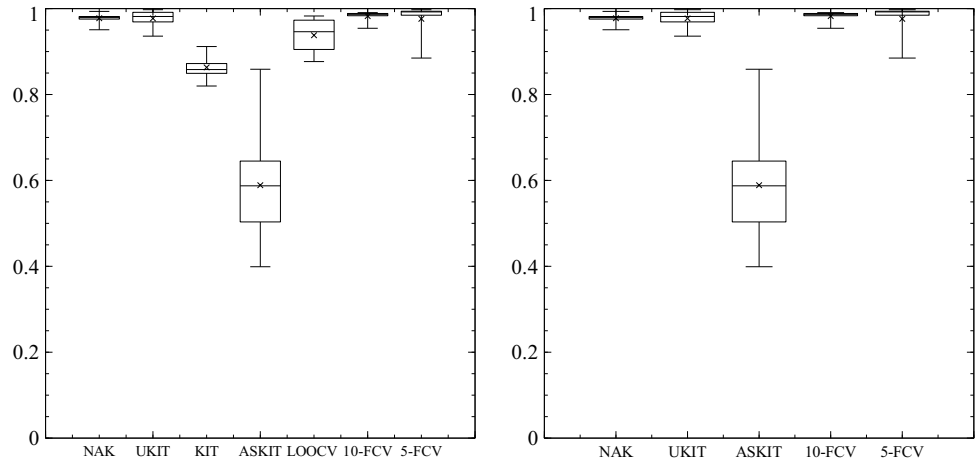
Figure 12 shows the results for the 4D, 8D, 12D and 16D versions considering 10 runs and 200 individuals at each of them. On the 4D problem, the maximum number of generations was set to 75, while on the higher dimensional versions, this parameter was set to 200. In Fig. 12(a), all methods were used, even Uniform Kitayama (see Eq. (10)). It was not used in the previous example because, on two-dimensional problems, this method provides the same σ values obtained using the formulation from Nakayama et al. [17]. In Fig. 12(b) both Non-Uniform Kitayama and LOOCV were not assessed, the former due to bad results and the latter due to time consumption. On the other hand, the Adaptive Scaling technique was kept (despite its poor performance on Peaks minimization) to better study its implications over high dimensional problems. Finally, in Fig. 12(c) and Fig. 12(d), 10-FCV was also cut from the testing due to time consumption.

Once again, in Fig. 12(a), both 10-FCV and 5-FCV presented better results than LOOCV. While both uniform propositions (NAK and UKIT) showed good results, the other analytical formulations did not perform so well, particularly the Adaptive Scaling technique (although it showed improvement if compared to the previous example). In Fig. 12(b), the same methods that stood out on a lower dimension

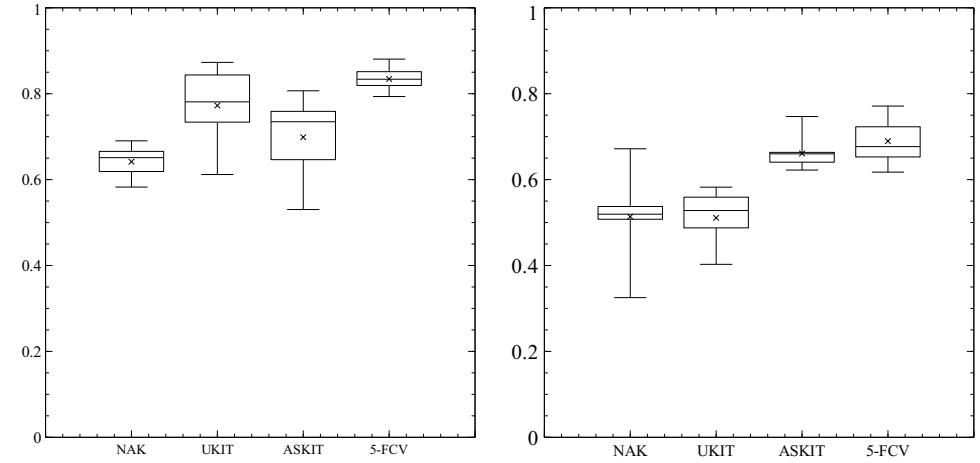


(a) Styblinski-Tang HFM (b) Contour lines

Figure 11. Two-dimensional Styblinski-Tang function



(a) 4D version (b) 8D version



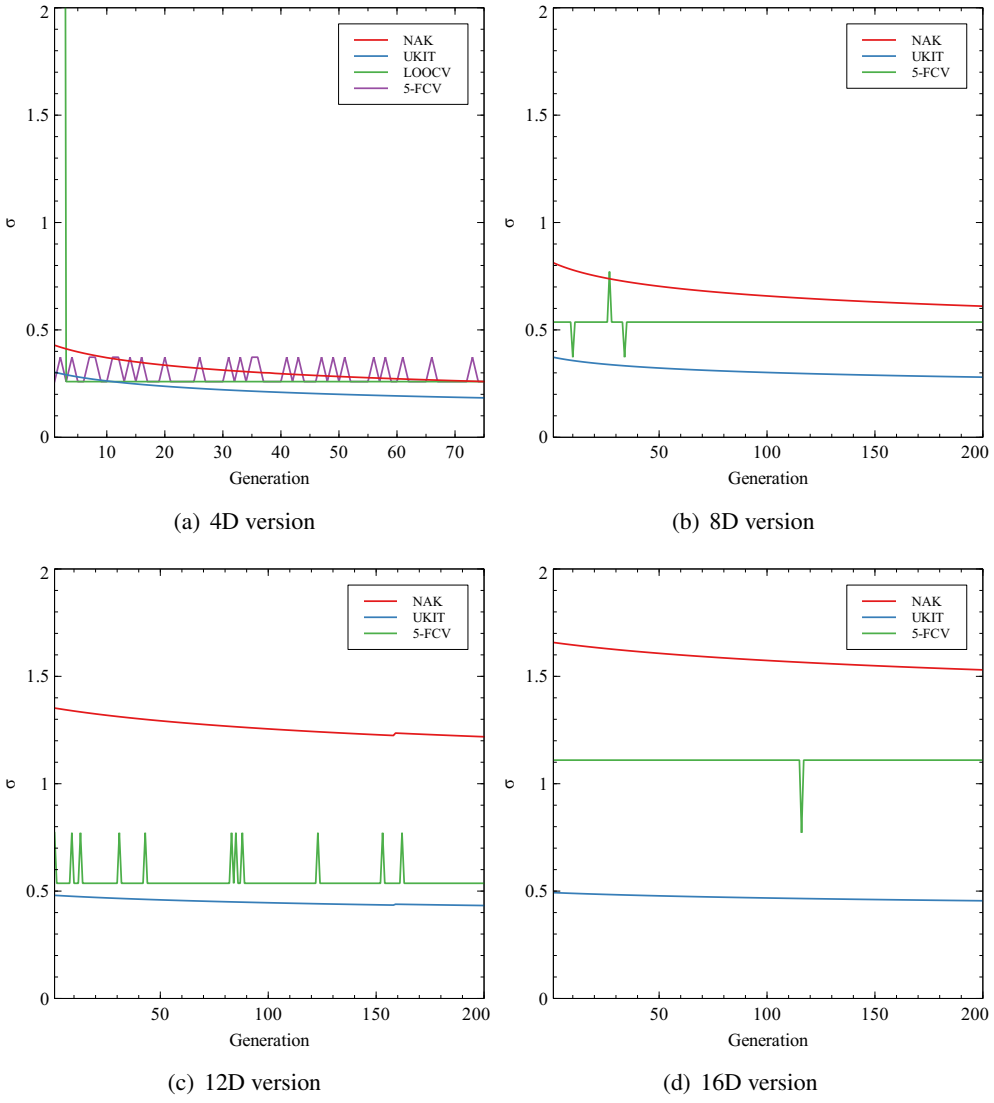
(c) 12D version (d) 16D version

Figure 12. Normalized results from the optimization of the Styblinski-Tang function

presented good performance again. Also, even though the design space has become wider, the Adaptive Scaling showed no visible improvement.

Finally, in both Fig. 12(c) and Fig. 12(d), none of the methods used have produced particularly

good results, probably due to the many local minima this function presents at these dimensions. The optimization's cost greatly increased as the required number of samples were also higher, especially when using cross-validation methods. This is also a limitation to the number of maximum generations considered. Among the methods considered, the 5-FCV presented the best performance. In Fig. 12(c), UKIT seems to have performed the best among the analytical formulations. In Fig. 12(d), the best results were obtained using the Adaptive Scaling technique. Figure 13 presents the σ values chosen over the generations by some of the methods for the three versions studied.



(a) 4D version (b) 8D version

(c) 12D version (d) 16D version

Figure 13. Width values found for the Styblinski-Tang function

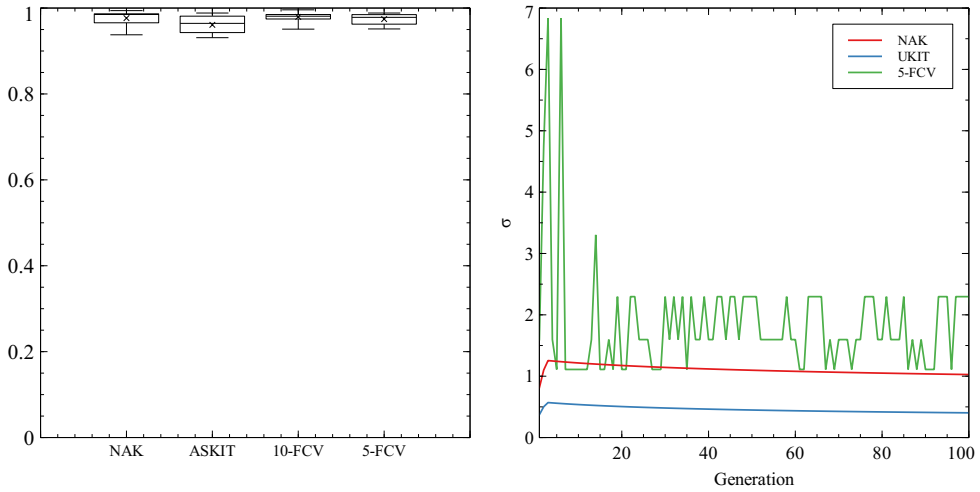
Note how the σ values increase with the problem dimensionality. However, both Fig. 13(a) and Fig. 13(b) still did not achieve σ values greater than 1.0, which explains the bad results produced by the Adaptive Scaling technique in these problems. In Fig. 13(c), NAK did show σ values greater than 1.0, however, these values seem overestimated if compared to those obtained by UKIT and 5-FCV. Luckily, this is not the case in Fig. 13(d), when the widths determined by the 5-FCV was also higher than 1.0. An exception here was the Uniform Kitayama, which still presented widths lower than 1.0, which actually seems underestimated in this case.

5.3 Minimization of the Dette & Pepelyshev function

This is an 8D function presented on Dette and Pepelyshev [24], evaluated over the domain $[0, 1]^8$, and it is highly curved in some variables but less so on others. Its formulation is shown below:

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

The results obtained are shown in Fig. 14(a) and the sigmas used some of the methods are shown in Fig. 14(b). On this example, it was considered 10 runs, 100 generations each with 200 individuals.



(a) Normalized results (b) Width values used

Figure 14. Results from the optimization of the Dette & Pepelyshev function

It is noticeable how all methods considered performed well in this example, especially the one using the Adaptive Scaling. This time, both Nakayama's proposition and the 5-FCV did achieve values greater than 1.0, contrasting with the values found in Fig. 13(b), which also deals with an 8D function. It is interesting to note that, in this case, the d_{max}, from Eq. (9), is a good enough parameter to assist Nakayama's formulation in representing the behavior of the function.

5.4 Strength maximization of a rectangular plate

This example was optimized for the first time by Kogiso et al. [31] and refers to a load factor maximization of a simply supported laminated rectangular plate (0.508 m x 0.127 m) subjected to biaxial compression. The thickness of each ply is 0.127 mm and the problem is said to contain many local minima and maxima. In this version, the ratio between N_x and N_y was considered as 0.125. The failure load must be maximized, considering the Maximum Strain Criterion as:

$$\lambda_s = min_k \left(min \left(\frac{\varepsilon_1^u}{S_f|\varepsilon_1^k|}, \frac{\varepsilon_2^u}{S_f|\varepsilon_2^k|}, \frac{\gamma_{12}^u}{S_f|\gamma_{12}^k|} \right) \right). \quad (17)$$

where S_f is the Safety factor (considered as 1.5). The buckling load factor (λ_b) is evaluated as proposed by Reddy [32]:

$$\lambda_b = \frac{D_{11}\left(\frac{p}{a}\right)^4 + 2(D_{12}+2D_{66})\left(\frac{p}{a}\right)^2\left(\frac{q}{b}\right)^2 + D_{22}\left(\frac{q}{b}\right)^4}{\left(\frac{p}{a}\right)^2 N_x + \left(\frac{q}{b}\right)^2 N_y}. \tag{18}$$

where p and q are the number of half-waves in each of the axis. To stipulate these values, 20 values (ranging from 1 to 20) of each one are tested, and the minimum value of λ_b found is used.

This problem has 48 plies and the variables are the ply orientations, ranging from 0 to 90 degrees in increments of 45. As the laminate is balanced and symmetric, there are 12 design variables. One constraint is also applied to limit the maximum number of contiguous plies to $cp_{max} = 4$. This way, the optimization process of this problem may be represented by:

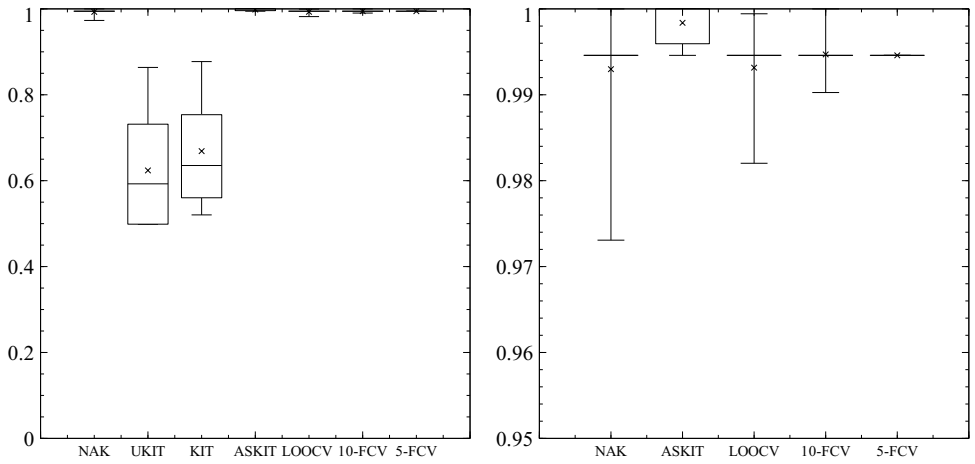
$$\begin{aligned} &\text{Find} && \mathbf{x} = [\theta_1, \theta_2, ..., \theta_{12}]. \\ &\text{that maximize} && min(\lambda_b, \lambda_s). \\ &\text{such that} && \text{Max contiguous plies} \leq cp_{max}. \end{aligned} \tag{19}$$

The properties of the material are show in Table 2. Even though Kogiso et al. [31] states that the minimum has a $\lambda = 13518.7\ kN$, local optimization using the High Fidelity Model found the value of $\lambda = 13535.1kN$ [33, 34], which will be considered for this example.

Table 2. Properties of the material used (Graphite-epoxy)

E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	ν_{12}	ε_1^u	ε_2^u	γ_{12}^u
130.71	6.36	4.18	0.32	0.008	0.029	0.015

Figure 15(a) depicts the results found using all methods considering 10 runs, 20 generations each and 100 individuals. Due to the bad results found in both Uniform and Non-Uniform Kitayama, 15(b) excludes these two methods and depicts the other results, using a better scaling factor.



(a) All methods (b) Excluding both UKIT and KIT

Figure 15. Normalized results for strength maximization of a rectangular plate problem

The use of Adaptive Scaling did greatly improve the proposal by Kitayama and Yamazaki [19] in this example. As a matter of fact, it showed the best results over all methods, in general. However, all methods shown in Fig. 15(b) have achieved excellent results. Comparing the σ values of each method

in Fig. 16, it seems that Uniform Kitayama underestimates the value, which is much lower than the one found by the other methods. This trend has happened in most other examples, although only now this has resulted so negatively on the results.

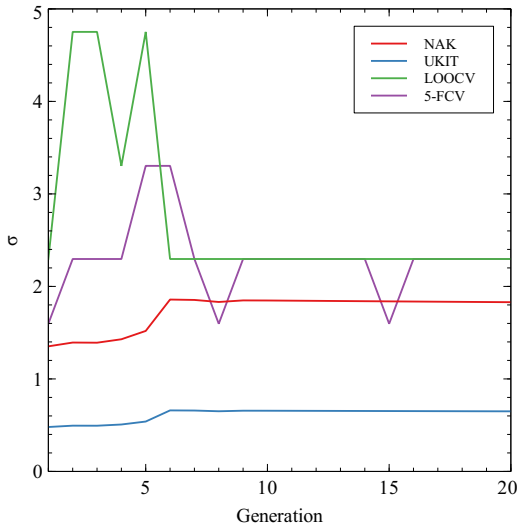


Figure 16. Width values used for the strength maximization of a rectangular plate problem

6 Conclusion

A comparison between some of the most used width-defining methods for Gaussian kernels on Radial Basis Function was performed in this paper. The main goal of this work is to aid future researchers on choosing the best method to suit their own needs by assessing the RBF performance using different methods, including simple analytical propositions [17, 19] and cross-validation methods, such as LOOCV and k-FCV.

The optimal $\boldsymbol{\sigma}$ vector is directly related to the function to be optimized. This way, by suggesting estimates without considering this aspect, it may seem that the analytical propositions would present a poor performance. However, on most functions tested, this was not the case. The closed-form expression resulted in performances close to those obtained by the cross-validation methods with a much lower cost.

That being said, both the 5-FCV and the proposition presented on Nakayama et al. [17] should be highlighted due to their excellent results. It is interesting to note that the former, while performing at a much lower cost, did not show any significant difference from the 10-FCV or the LOOCV, even though these should provide better predictions. If the user is performing a low-budget optimization, the analytical methods will likely provide very solid results. However, if the cost of the problem relies solely on the High Fidelity Model analysis, it might be wiser to perform the 5-FCV.

The proposal by Kitayama and Yamazaki [19] using the Adaptive Scaling also presented good results on high dimensional problems. However, it should be used carefully, due to the poor results found in lower-dimensional functions where σ values should be lower than 1.0. Moreover, the criticism of the same author on the method proposed by Nakayama et al. [17] seems to have a basis in facts. However, in many examples, this turned out to be a problem since the parameter was actually being underestimated. The use of an intermediate approach might improve the results, even though it is impossible to propose an analytical approach that performs optimally for all types of problems.

Finally, it is easy to perceive how, in most examples, the σ values picked by cross-validation methods remain constant for a large number iterations, especially on later generations. That way, users could perform a better study over the search domain used in these methods. This paper followed the proposition made by Sobester et al. [14], searching σ values over the domain $[10^{-2}, 10]$. However, this might be too wide as it probably was for most functions used in this paper. By proposing a better search domain, the

user may achieve better predictions and lower the time consumption. Another interesting approach is that it is probably not necessary to measure these methods in every generation since it might not be efficient due to their cost. This way, users might be able to exploit previous assessments multiple times in order to achieve better computational efficiency. Implementations and testing over this matter are underway.

Acknowledgements

The financial support by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) is gratefully acknowledged.

References

[1] Kitayama, S., Huang, S., & Yamazaki, K., 2012. Optimization of variable blank holder force trajectory for springback reduction via sequential approximate optimization with radial basis function network. *Structural and Multidisciplinary Optimization*, vol. 47, n. 2, pp. 289–300.

[2] Zhang, Y., Gong, C., Fang, H., Su, H., Li, C., & Ronch, A. D., 2019. An efficient space division–based width optimization method for rbf network using fuzzy clustering algorithms. *Structural and Multidisciplinary Optimization*.

[3] Yan, C., Shen, X., & Guo, F., 2017. An improved support vector regression using least squares method. *Structural and Multidisciplinary Optimization*, vol. 57, n. 6, pp. 2431–2445.

[4] Yan, C., Shen, X., Guo, F., Zhao, S., & Zhang, L., 2019. A novel model modification method for support vector regression based on radial basis functions. *Structural and Multidisciplinary Optimization*.

[5] Watts, G., Pradyumna, S., & Singha, M., 2017. Nonlinear analysis of quadrilateral composite plates using moving kriging based element free galerkin method. *Composite Structures*, vol. 159, pp. 719–727.

[6] Jones, D. R., Schonlau, M., & Welch, W. J., 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, pp. 455–492.

[7] Wang, G. G. & Shan, S., 2007. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, vol. 129, n. 4, pp. 370.

[8] Gan, N. & Gu, J., 2018. Hybrid meta-model-based design space exploration method for expensive problems. *Structural and Multidisciplinary Optimization*, vol. 59, n. 3, pp. 907–917.

[9] Simpson, T. W., Lin, D. K. J., & Chen, W., 2002. Sampling strategies for computer experiments: Design and analysis. *International Journal of Reliability and Applications*.

[10] Forrester, A. I. J., Sobester, A., & Keane, A. J., 2008. *Engineering design via surrogate modelling: a practical guide*. Wiley.

[11] Mongillo, M., 2011. Choosing basis functions and shape parameters for radial basis function methods. *SIAM Undergraduate Research Online*, vol. 4, pp. 190–209.

[12] Mehmani, A., Chowdhury, S., Meinrenken, C., & Messac, A., 2017. Concurrent surrogate model selection (cosmos): optimizing model type, kernel function, and hyper-parameters. *Structural and Multidisciplinary Optimization*, vol. 57, n. 3, pp. 1093–1114.

[13] Acar, E., 2013. Simultaneous optimization of shape parameters and weight factors in ensemble of radial basis functions. *Structural and Multidisciplinary Optimization*, vol. 49, n. 6, pp. 969–978.

[14] Sobester, A., Leary, S. J., & Keane, A. J., 2005. On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization*, pp. 31–59.

[15] Müller, J. & Shoemaker, C. A., 2014. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, vol. 60, n. 2, pp. 123–144.

[16] Haykin, S., 1994. *Neural Networks: A Comprehensive Foundation*. Macmillan Publishing.

[17] Nakayama, H., Arakawa, M., & Sasaki, R., 2002. Simulation-based optimization using computational intelligence. *Optimization and Engineering*, vol. 3, n. 2, pp. 201–214.

[18] Benoudjit, N., Archambeau, C., Lendasse, A., Lee, J., & Verleysen, M., 2002. *Width optimization of the Gaussian kernels in Radial Basis Function Networks*.

[19] Kitayama, S. & Yamazaki, K., 2011. Simple estimate of the width in gaussian kernel with adaptive scaling technique. *Applied Soft Computing*, vol. 11, n. 8, pp. 4726–4737.

[20] Wu, Z., Wang, D., N, P. O., Jiang, Z., & Zhang, W., 2016. Unified estimate of gaussian kernel width for surrogate models. *Neurocomputing*, vol. 203, pp. 41–51.

[21] Tenne, Y., 2014. Initial sampling methods in metamodel-assisted optimization. *Engineering with Computers*, vol. 31, n. 4, pp. 661–680.

[22] Rennen, G., 2008. Subset selection from large datasets for kriging modeling. *Structural and Multidisciplinary Optimization*, vol. 38, n. 6, pp. 545–569.

[23] Liu, L. & Wakeland, W., 2005. Does more uniformly distributed sampling generally lead to more accurate prediction in computer experiments? *Proceedings of the 37th Winter Simulation Conference*.

[24] Dette, H. & Pepelyshev, A., 2010. Generalized latin hypercube design for computer experiments. *Technometrics*, vol. 52, n. 4, pp. 421–429.

[25] Amouzgar, K. & Strömberg, N., 2016. Radial basis functions as surrogate models with a priori bias in comparison with a posteriori bias. *Structural and Multidisciplinary Optimization*, vol. 55, n. 4, pp. 1453–1469.

[26] Yao, W., Chen, X., Huang, Y., & Tooren, M. V., 2013. A surrogate-based optimization method with rbf neural network enhanced by linear interpolation and hybrid infill strategy. *Optimization Methods and Software*, vol. 29, n. 2, pp. 406–429.

[27] Schmit, J. L. & Farshi, B., 1973. Some approximation concepts for structural synthesis. *14th Structures, Structural Dynamics, and Materials Conference*.

[28] Kitayama, S., Arakawa, M., & Yamazaki, K., 2010. Sequential approximate optimization using radial basis function network for engineering optimization. *Optimization and Engineering*, vol. 12, n. 4, pp. 535–557.

[29] Hardy, R. L., 1971. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, vol. 76, n. 8, pp. 1905–1915.

[30] Kitayama, S., Srirat, J., Arakawa, M., & Yamazaki, K., 2013. Sequential approximate multiobjective optimization using radial basis function network. *Structural and Multidisciplinary Optimization*, vol. 48, n. 3, pp. 501–515.

[31] Kogiso, N., Watson, L. T., Gürdal, Z., & Haftka, R. T., 1994. Genetic algorithms with local improvement for composite laminate design. *Structural Optimization*, vol. 7, n. 4, pp. 207–218.

[32] Reddy, J. N., 2004. *Mechanics of laminated composite plates and shells: theory and analysis*. CRC Press.

[33] Barroso, E. S., 2015. *Análise e otimização de estruturas laminadas utilizando a formulação isogeométrica*. PhD thesis, Universidade Federal do Ceará.

[34] Balreira, D. S., 2018. *Otimização sequencial aproximada de estruturas laminadas de material compósito*. PhD thesis, Universidade Federal do Ceará.