On the search of the shape parameter in radial basis functions using univariate global optimization methods

Noname manuscript No.
(will be inserted by the editor)

On the Search of the Shape Parameter in Radial Basis Functions Using Univariate Global Optimization Methods

R. Cavoretto · A. De Rossi · M.S. Mukhametzhanov · Ya.D. Sergeyev

Received: date / Accepted: date

Abstract In this paper we consider the problem of finding an optimal value of the shape parameter in radial basis function interpolation. In particular, we propose the use of a leave-one-out cross validation (LOOCV) technique combined with univariate global optimization methods, which involve strategies of Global Optimization with Pessimistic Improvement (GOPI) and Global Optimization with Optimistic Improvement (GOOI). This choice is carried out to overcome serious issues of commonly used optimization routines that sometimes result in shape parameter values are not globally optimal. New locally-biased versions of geometric and information Lipschitz global optimization algorithms are presented. Numerical experiments and applications to real-world problems show a promising performance and efficacy of the new algorithms, called LOOCV-GOPI and LOOCV-GOOI, in comparison with their direct competitors.

Keywords RBF interpolation · shape parameter · global optimization

Mathematics Subject Classification (2010) 65D05 · 65D15 · 90C26 · 65K05

R. Cavoretto, A. De Rossi
Department of Mathematics "Giuseppe Peano", University of Torino, via Carlo Alberto 10, 10123 Torino, Italy
E-mail: {roberto.cavoretto, alessandra.derossi}@unito.it

M. S. Mukhametzhanov, Ya. D. Sergeyev
DIMES, University of Calabria, via P. Bucci, Cubo 42-C, 87036 Rende (CS), Italy
ITMM, Lobachevsky Nizhni Novgorod State University, Gagarin Avenue 23, Nizhni Novgorod 603950, Russia
E-mail: {m.mukhametzhanov, yaro}@dimes.unical.it

Corresponding author: Ya. D. Sergeyev

1 Introduction

In the Radial Basis Function (RBF) community the problem of selecting a good (in some sense) or an optimal value of the shape parameter has been a great issue for a very long time. It is in fact well-known that in RBF interpolation and collocation methods the choice of the radial basis associated with its shape parameter significantly influences the obtained results both in terms of accuracy and stability (see e.g. [14, 51]). For many years, people who actively worked in the field of approximation theory or, more in general, in numerical analysis carried out ad-hoc choices of the shape parameter or sometimes even ignored its effect by treating it as a constant (see [15, 23] and references therein). Much more systematic approaches have been suggested first in statistics literature [50], and later extended to the community of RBF approximation [13, 16, 34, 36]. Among several techniques proposed to predict an optimal value, or even an interval [4], of the RBF shape parameter, we focus on a popular version of cross validation, called Leave-One-Out Cross Validation (LOOCV). This method was introduced in the context of RBF interpolation by Rippa [34], but now it is commonly used in several fields of science and engineering such as numerical analysis, mathematical modeling, optimization and machine learning (see e.g. [9, 10, 35, 48, 49]). However, the use of a standard LOOCV approach is quite expensive from a computational standpoint. It is therefore evident the importance of designing methods and algorithms that allow us to reduce the computational effort. This task can be done by solving an optimization problem, thus detecting the shape parameter value for which the interpolation error is the smallest possible. Nevertheless, as we will also show in our numerical sections, some commonly used minimization algorithms (see e.g. [49, 52]) can sometimes fail and return a locally optimal value that in practice is rather distant from the true global optimum. This fact motivates us to enhance the current state of the art.

In this work, in order to find an optimal shape parameter for the RBF interpolation, we propose to use a LOOCV technique combined with univariate global optimization methods, which in this context requires the computation of large Lipschitz constants. For this reason, we address our attention on the construction of new algorithms based on deterministic global optimization methods by applying efficient local tuning and local improvement techniques to estimate the local Lipschitz constants and accelerate significantly the search process (see [28, 40, 44, 45]). In particular, we focus on the Global Optimization with Pessimistic Improvement (GOPI) and the Global Optimization with Optimistic Improvement (GOOI) (see [44, 45]). The combination of LOOCV with these two global optimization methods allows us to get two new algorithms, named LOOCV-GOPI and LOOCV-GOOI. In our extensive numerical experiments we show the performance of these algorithms, also highlighting their ability in determining the global minimum of the objective (or cost) function. Several comparisons with a standard LOOCV method (whose results can be viewed as the exact solution of our minimization problem) and the LOOCV method associated with the MATLAB minimization routine *fminbnd*, called

LOOCV-min, are also given. Finally, the efficacy of our new LOOCV-GOOI approach is also pointed out for solving RBF interpolation problems in real-world applications.

The rest of the paper is organized as follows. In Section 2, the RBF interpolation problem is stated as well as several traditional methods for selecting a good value of the shape parameter of the RBFs are discussed. In Section 3, a univariate global optimization framework is briefly described. In Section 4, the proposed methods for selecting the value of the shape parameter using the global optimization techniques are presented. Section 5 contains results of numerical experiments on several two-dimensional test problems. In Section 6, results on two real-life applications using different RBFs are also presented. Finally, Section 7 concludes the paper.

2 Problem Statement and Traditional Methods

2.1 Radial Basis Function Interpolation

Given a domain $\Omega \subseteq \mathbb{R}^s$, a set $X_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \Omega$ of n distinct *data points* or *nodes* and the corresponding set $F_n = \{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)\}$ of *data* or *function values* obtained by possibly sampling any (unknown) function $f : \Omega \to \mathbb{R}$, a standard RBF interpolant $\mathcal{I}_f : \Omega \to \mathbb{R}$ is a linear combination of RBFs of the form

$$\mathcal{I}_f(\boldsymbol{x}) = \sum_{i=1}^{n} c_i \phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{x}_i||_2), \quad \boldsymbol{x} \in \Omega, \tag{1}$$

where $c_i,\ i = 1, ..., n$, are unknown real coefficients, $||\cdot||_2$ denotes the Euclidean norm, and $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a strictly positive definite RBF depending on a *shape parameter* $\varepsilon > 0$ such that

$$\phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{y}||_2) = \phi(\varepsilon||\boldsymbol{x} - \boldsymbol{y}||_2), \quad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \Omega.$$

For simplicity, from now on we refer to ϕ_ε as ϕ. In Table 1 we report a list of some strictly positive definite RBFs with their smoothness degrees. In particular, we remark that Gaussian, Inverse MultiQuadric, Inverse Quadratic and Matérn functions are globally supported and strictly positive definite in $\mathbb{R}^s$ for any s, whereas Wendland functions are compactly supported – with support $[0, 1/\varepsilon]$ – and strictly positive definite in $\mathbb{R}^s$ for $s \leq 3$ (see [13,51]).

The coefficients $c_1, \ldots, c_n$ in (1) are determined by enforcing the interpolation conditions

$$\mathcal{I}_f(\boldsymbol{x}_i) = f(\boldsymbol{x}_i), \quad i = 1, \ldots, n. \tag{2}$$

As a result, solving the interpolation problem (2) results in a symmetric linear system of equations

$$\begin{pmatrix} \phi(||\boldsymbol{x}_1 - \boldsymbol{x}_1||_2) & \cdots & \phi(||\boldsymbol{x}_1 - \boldsymbol{x}_n||_2) \\ \vdots & \vdots & \vdots \\ \phi(||\boldsymbol{x}_n - \boldsymbol{x}_1||_2) & \cdots & \phi(||\boldsymbol{x}_n - \boldsymbol{x}_n||_2) \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}, \tag{3}$$

RBF	$\phi_\varepsilon(r)$
Gaussian C^∞ (GA)	$\mathrm{e}^{-\varepsilon^2 r^2}$
Inverse MultiQuadric C^∞ (IMQ)	$(1+\varepsilon^2 r^2)^{-1/2}$
Inverse Quadric C^∞ (IQ)	$(1+\varepsilon^2 r^2)^{-1}$
Matérn C^6 (M6)	$\mathrm{e}^{-\varepsilon r}(\varepsilon^3 r^3 + 6\varepsilon^2 r^2 + 15\varepsilon r + 15)$
Matérn C^4 (M4)	$\mathrm{e}^{-\varepsilon r}(\varepsilon^2 r^2 + 3\varepsilon r + 3)$
Matérn C^2 (M2)	$\mathrm{e}^{-\varepsilon r}(\varepsilon r + 1)$
Wendland C^6 (W6)	$(1-\varepsilon r)_+^8\,(32\varepsilon^3 r^3 + 25\varepsilon^2 r^2 + 8\varepsilon r + 1)$
Wendland C^4 (W4)	$(1-\varepsilon r)_+^6\,(35\varepsilon^2 r^2 + 18\varepsilon r + 3)$
Wendland C^2 (W2)	$(1-\varepsilon r)_+^4\,(4\varepsilon r + 1)$

Table 1: Examples of strictly positive definite RBFs with their orders of smoothness and shape parameter $\varepsilon > 0$; $r = ||\cdot||_2$ is the Euclidean norm and $(\cdot)_+$ denotes the truncated power function.

or simply

$$A\boldsymbol{c} = \boldsymbol{f}, \tag{4}$$

where $\boldsymbol{c} = (c_1, \ldots, c_n)^T$, $\boldsymbol{f} = (f_1, \ldots, f_n)^T$, and the interpolation matrix $A \in \mathbb{R}^{n\times n}$ is given by $A_{ki} = \phi(||\boldsymbol{x}_k - \boldsymbol{x}_i||_2)$, $k, i = 1, \ldots, n$. Since ϕ is a strictly positive function, the associated matrix A is nonsingular and the RBF interpolation problem is well-posed, hence a solution to the problem exists and is unique [5,14].

Further, to each strictly positive definite RBF ϕ we can associate a strictly positive definite and symmetric kernel $\Phi : \Omega \times \Omega \to \mathbb{R}$, i.e.

$$\Phi(\boldsymbol{x}, \boldsymbol{y}) = \phi(||\boldsymbol{x} - \boldsymbol{y}||_2) \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega,$$

whose real Hilbert space $\mathcal{N}_\Phi(\Omega)$ is called the *native space*. So we consider $H_\Phi(\Omega) = \mathrm{span}\{\Phi(\cdot, \boldsymbol{x}),\ \boldsymbol{x} \in \Omega\}$, equipped with the bilinear form $(\cdot,\cdot)_{H_\Phi(\Omega)}$. Since $H_\Phi(\Omega)$ is a pre-Hilbert space with reproducing kernel Φ, the native space $\mathcal{N}_\Phi(\Omega)$ of Φ is then its completion with respect to the norm $||\cdot||_{H_\Phi(\Omega)}$, that is $||f||_{H_\Phi(\Omega)} = ||f||_{\mathcal{N}_\Phi(\Omega)}$, for all $f \in H_\Phi(\Omega)$ (cf. [14,51]).

For RBF interpolation, we give a first error estimate in terms of the well-known *power function* P_{Φ,X_n} (see e.g. [14, Theorem 14.2]).

Theorem 1 *Let $\Omega \subseteq \mathbb{R}^s$, $\Phi \in C(\Omega \times \Omega)$ be strictly positive definite on $\mathbb{R}^s$, and suppose that $X_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ consists of distinct points. Then for all $f \in \mathcal{N}_\Phi(\Omega)$ we have*

$$|f(\boldsymbol{x}) - \mathcal{I}_f(\boldsymbol{x})| \le P_{\Phi,X_n}(\boldsymbol{x})||f||_{\mathcal{N}_{\Phi(\Omega)}}, \quad \boldsymbol{x} \in \Omega. \tag{5}$$

To refine this error bound, we define the so-called *fill distance*

$$h_{X_n,\Omega} = \sup_{\boldsymbol{x}\in\Omega} \min_{\boldsymbol{x}_i \in X_n} ||\boldsymbol{x} - \boldsymbol{x}_i||_2,$$

which is a common indicator of data distribution and indicates how well the data fill out the domain Ω. The generic error bound of Theorem 1 can then be formulated in terms of the fill distance (see [14, Theorem 14.5] for details).

Theorem 2 *Let $\Omega \subseteq \mathbb{R}^s$ be bounded and satisfies an interior cone condition. Suppose that $\Phi \in C^{2k}(\Omega \times \Omega)$ is symmetric and strictly positive definite. Then for all $f \in \mathcal{N}_\Phi(\Omega)$ there exist constants h_0, $C > 0$ (independent of $\boldsymbol{x}$, f and Φ) such that*

$$|f(\boldsymbol{x}) - \mathcal{I}_f(\boldsymbol{x})| \leq C h_{X_n,\Omega}^k \sqrt{C_\Phi(\boldsymbol{x})}\, \|f\|_{\mathcal{N}_\Phi(\Omega)}\,,$$

provided $h_{X_n,\Omega} \leq h_0$. Here

$$C_\Phi(\boldsymbol{x}) = \max_{|\boldsymbol{\beta}|=2k,\, \boldsymbol{w},\boldsymbol{z} \in \Omega \cap B(\boldsymbol{x}, c_2 h_{X_n,\Omega})} \left| D_2^{\boldsymbol{\beta}} \Phi(\boldsymbol{w}, \boldsymbol{z}) \right|$$

with $B(\boldsymbol{x}, c_2 h_{X_n,\Omega})$ denoting the ball of radius $c_2 h_{X_n,\Omega}$ centred at $\boldsymbol{x}$ and $D_2^{\boldsymbol{\beta}}$ is the differential operator applied to Φ viewed as a function of the second variable.

Theorem 2 says that interpolation with a C^{2k} smooth kernel Φ has approximation order k. Thus, we deduce that: (a) for infinitely smooth strictly positive definite functions, the approximation order k is arbitrarily high; (b) for strictly positive definite functions with limited smoothness, the approximation order is limited by the smoothness of the basis function. For more refined error estimates, see the book [51].

In the recent paper [54], a problem of approximating the parameters for a Gaussian random field is considered. Since the problem of approximating the scale parameters β in the Gaussian random field in case of fixed (or independently approximated) parameters μ and σ and with the value $\gamma = 2$ is similar to finding the shape parameter ε for a Gaussian type RBF, some results obtained in [54] can be also verified for the present research. In particular, it has been shown in [54] that approximation of the parameters for such the model can be very difficult due to computational efforts related to ill-conditioning of the covariation matrix (similar to the matrix A from (4)). It has been shown that the condition number of the matrix can be extremely high and traditional methods for approximating β (e.g., the Maximum Likelihood Estimates) cannot be used for this reason. However, the paper [54] is concentrated at Gaussian functions, so the obtained results cannot be generalized for different RBFs: for instance, the RBFs with a small smoothness degree (e.g., W2 or W4) allow one to generate the matrices with relatively small condition number (with respect to the Gaussian-type RBF) and can be useful in some cases. Moreover, the main attention in [54] is paid to computational efforts in approximation of the parameters instead of methods of approximation of the scale parameters β, whereas the main aim of the present paper is to propose a new way of finding a good value of the shape parameter ε. In addition, since the dimension s of the interpolated function $f(x)$ is considered larger than 1 and the sample size n is not very large, then the condition number of the

matrix A becomes much smaller than the condition number of the correlation matrix with $d = 1$ in [54], which allows us to find an "optimal" value of ε.

2.2 Traditional Methods for Selecting a Good Value of the Shape Parameter

Over the years, several approaches have been proposed to select good, or even optimal, shape parameters for RBF interpolation and collocation methods. In the literature, there are many techniques that can guide us in making a decision regarding this selection. It can be done either by finding a single shape parameter uniformly across the whole domain or by determining a shape parameter that varies spatially. Though some work has been carried out in the latter case (see e.g. [11,13,23,29] and references therein), most strategies focus on the selection of only one shape parameter associated with the RBF. Therefore, we move our investigation in this direction, discussing the main strategies to choose a good value of the shape parameter ε and mentioning some recent contributions.

The simplest way is to perform a number of experiments by varying the value of the shape parameter and then to take the best one. This strategy, known as *trial and error*, can be used for academic purposes and if one knows in advance the function f which generates the data so that some interpolation error can be computed. However, it is evident that in practical situations or in real-life applications the data function f is not available at all (see e.g. [1, 8,14]).

A further strategy to find ε is to use the power function P_{Φ,X_n} from (5) as an indicator. In fact, the error estimate (5) provided in Theorem 1 separates the interpolation error into a component independent of the function f and another one depending on it. This means that chosen the scaled basic function Φ and given the data set X_n we can use the power function to minimize the error component that turns out to be independent of f. On the one hand, this technique takes advantage of not depending on the function f, but on the other one, it cannot be optimal because the second component of the error bound also depends on the basic function through the native space norm. Possible modifications and other details on this approach can be found, for instance, in [15,36].

Another approach for selecting an optimal shape parameter is based on a cross validation approach. This technique is quite popular in statistics and uses the given data to predict optimal values of model parameters for data fitting. Here the basic idea consists in splitting the data points into a training set and a validation set, computing then the error obtained by gauging the accuracy of the fit built from the information on the training set at points in the validation set [13]. A specific case of cross validation is given by LOOCV [34]. It considers all points in the training set, excepting one that is left out and in turn is the sole member of the validation set. An optimal value of the shape parameter is thus selected by minimizing the error for a fit to the data based on an interpolant for which one of the data points is removed.

Since the dependence of the data function f is here taken into account, the LOOCV gives us a better prediction of ε than the previous power function methods can do. Some changes and applications of LOOCV in the field of RBF approximation and in the context of PDEs are proposed in [9,10,16,35, 48,49,52]. Moreover, various extensions and generalizations of cross validation, as well as the connection with the approach of *maximum likelihood estimation*, are given in [15,36]. For further details and a more complete overview on this topic, see the book [13].

2.3 Choosing the RBF Shape Parameter via LOOCV

The accuracy of RBF interpolants highly depends upon the shape parameter ε of the basis functions, which is responsible for the flatness of the functions. In particular, for smooth problems the best accuracy is typically achieved when ε is small, but in such cases the condition number of the linear system becomes very large. Therefore, in order to get reliable approximation results, we need to find a technique that allows detecting a suitable value of ε. In fact, since an RBF interpolation method is based on the solution of a linear system of the form (3) (or (4)), the selection of ε greatly affects the accuracy of the interpolant (1).

A good way to select a shape parameter ε is to use the LOOCV by applying Rippa's method [34]. This approach is widely used in literature and can be applied in different fields of machine learning including regression, classification, clustering, etc. The idea behind LOOCV is to split the data into two different sets:

- a *training set* $\{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_{k-1}), f(\boldsymbol{x}_{k+1}), \ldots, f(\boldsymbol{x}_n)\}$,
- a *validation set* consisting of only the single value $f(\boldsymbol{x}_k)$ which was left out when creating the training set.

Now, for a fixed $k \in \{1, \ldots, n\}$ and fixed ε, we define the partial RBF interpolant

$$\mathcal{I}_f^{[k]}(\boldsymbol{x}) = \sum_{i=1,\ i\neq k}^{n} c_i^{[k]} \phi(||\boldsymbol{x} - \boldsymbol{x}_i||_2),$$

whose coefficients $c_i^{[k]}$ are determined by interpolating the training data, i.e.

$$\mathcal{I}_f^{[k]}(\boldsymbol{x}_i) = f(\boldsymbol{x}_i), \qquad i = 1, \ldots, k-1, k+1, \ldots, n.$$

In order to measure the quality of this attempt, we define the error

$$e_k(\varepsilon) = f(\boldsymbol{x}_k) - \mathcal{I}_f^{[k]}(\boldsymbol{x}_k) \tag{6}$$

at the one validation point $\boldsymbol{x}_k$ not used to determine the interpolant. The "optimal" value of ε is found as

$$\varepsilon_{opt} = \text{argmin}_{\varepsilon} ||\boldsymbol{e}(\varepsilon)||, \qquad \boldsymbol{e} = (e_1, \ldots, e_n)^T, \tag{7}$$

where $||\cdot||$ is any norm used in the minimization problem, for instance, the ∞-norm.

The important aspect is that we can determine the error vector $\boldsymbol{e}$ without solving n problems, each of size $(n-1)\times(n-1)$. In fact, instead of (6), the computation of the error components can be expressed in terms of the interpolation matrix A in (4), i.e.

$$e_k(\varepsilon) = \frac{c_k}{A^{-1}_{kk}}, \tag{8}$$

where c_k is the k-th coefficient of the full RBF interpolant $\mathcal{I}_f$ in (1) and A^{-1}_{kk} is the kth diagonal element of the inverse of the corresponding $n\times n$ interpolation matrix A [15]. So from (7) and (8) it follows that the LOOCV cost function to be minimized is

$$\mathrm{LOOCV}(\varepsilon) = ||\boldsymbol{e}(\varepsilon)||_\infty = \max_{k=1,\ldots,n}\left|\frac{c_k}{A^{-1}_{kk}}\right|. \tag{9}$$

3 Univariate Global Optimization

In this work, in order to find a good shape parameter for the RBF interpolation using the LOOCV technique presented in Subsection 2.3, univariate global optimization methods are used. In order to proceed, let us give a brief introduction in this field. Let us consider the following univariate global optimization problem:

$$g^* = g(x^*) = \min g(x), \quad x \in D = [a,b] \subset \mathbb{R}. \tag{10}$$

The objective function[1] $g(x)$ in (10) is supposed to be multiextremal, non-differentiable, and hard to evaluate even at one point (see, e.g., [2,3,22,24,30, 53]). Moreover, it is supposed to be Lipschitz-continuous over the interval D, i. e.,

$$|g(x_1) - g(x_2)| \le L|x_1 - x_2|, \quad x_1, x_2 \in D, \tag{11}$$

where $L,\ 0 < L < \infty$, is the Lipschitz constant.

There exists a huge number of methods that solve the problem (10)–(11) (see, e.g., [17,21,26,27,47,55,56]). Among them there are metaheuristic nature-inspired algorithms and deterministic Lipschitz global optimization algorithms. In particular, in the works [32] and [46], two deterministic algorithms for solving the problem (10)–(11) have been proposed. The first one uses geometric properties of the Lipschitz condition. It adaptively builds piecewise linear minorants of the objective function during the search using a priori given estimate of the Lipschitz constant. The other method is based on stochastic models using calculation of probabilities of locating global minimizers within

[1] In this section, "the objective function" means the function to be optimized, i.e., the error function in the approximation problem.

the subintervals of the search space. This algorithm uses adaptive global estimates of the Lipschitz constant. These two methods became basic methods for two classes of univariate Lipschitz global optimization algorithms called *geometric* and *information* global optimization methods, respectively.

It has been shown in [25,40,41] that both geometric and information global optimization algorithms are more efficient in terms of the number of the objective function's evaluations (called *trials* hereinafter) than popular metaheuristic methods (e.g., Firefly algorithm, Particle Swarm Optimization, Artificial Bee Colony, Differential Evolution, Simulated Annealing method, etc.) if they are applied to univariate global optimization problems with large Lipschitz constants (see also [43] for the comparison on multidimensional test problems). For this reason, in this paper, only deterministic global optimization algorithms are considered.

It should be mentioned that there exist efficient univariate global optimization methods for smooth objective functions with the Lipschitz first derivative (see, e.g., [6,19,28,39]). However, since in our framework the function to be minimized can be non-smooth, then the geometric and information approaches that do not use derivatives for constructing global optimization methods are considered in this paper. These frameworks allow one to construct global optimization algorithms for continuous functions only under the assumption of their Lipschitz continuity. Moreover, the Lipschitz condition is used only for guarantying the global convergence of the methods.

In the recent papers [20,45], several ideas introduced to speed up the global search have been proposed. First, as it has been shown in [28,42,45], *local tuning techniques* for estimating local Lipschitz constants significantly accelerate the search. In Figure 1, an example of the auxiliary function for a Lipschitz function $g(x)$ over the interval $[a,b]$ constructed by different estimation techniques is presented. Here, the objective function is shown by a solid black line; the minorant function $F^G(x)$ using a global estimate of the Lipschitz constant is shown by a dashed black line and the auxiliary functions $F^M(x)$ and $F^{MA}(x)$ using two different local tuning techniques are also shown by blue and red solid thin lines. The first local tuning technique called "Maximum" local tuning (its auxiliary function $F^M(x)$ is shown by the blue solid thin line) has been proposed in [37,38]. It realizes a balancing between the global and the local information about the behavior of the objective function obtained during the search using maximum convolution product between them. Another local tuning technique called "Maximum-Additive" local tuning (its auxiliary function $F^{MA}(x)$ is shown by the red solid thin line) has been proposed in [45]. It also realizes a smart mixture of the information about the local and global behavior of the objective function but it uses both the maximum and additive convolution formulae in order to estimate better the local Lipschitz constants and to keep theoretical convergence properties of the original "Maximum" local tuning technique.

It has been also shown in [44,45] that *local improvement techniques* can accelerate the global search significantly, as well. These techniques realize an improvement of the current best obtained value. The main point here is that

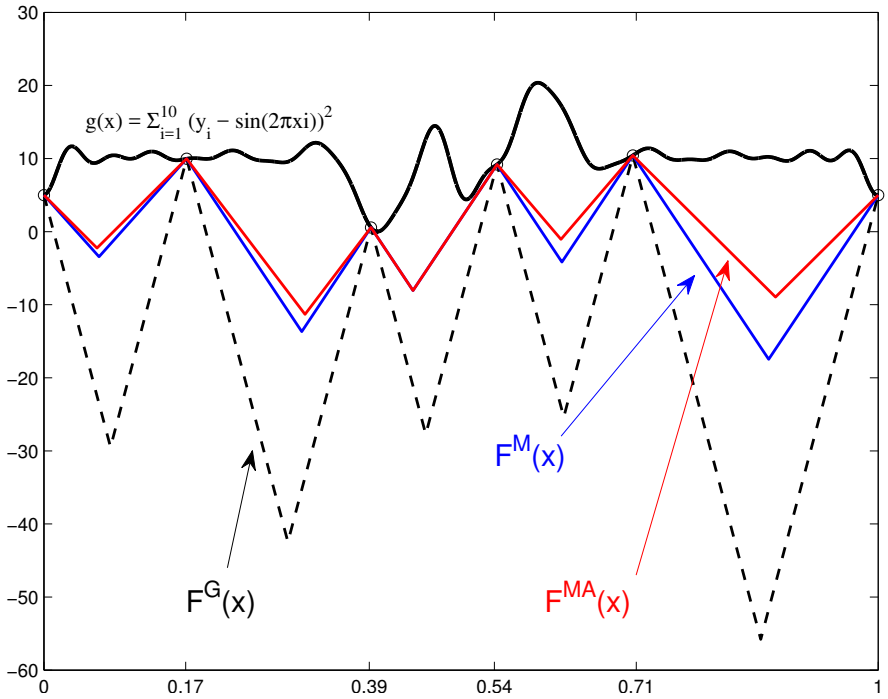


Fig. 1: Auxiliary functions for a Lipschitz function $g(x)$ over $[a, b]$, constructed by using a global estimate of the Lipschitz constant (black dashed line), "Maximum" local tuning (blue solid thin line) and "Maximum-Additive" local tuning (red solid thin line, coincides with the auxiliary function of the "Maximum" local tuning on the third interval). Trial values are circled.

the local improvement techniques from [45] automatically realize a local behavior in the promising subregions of the search domain without the necessity to stop the global optimization procedure. Moreover, it should be noticed that all evaluations of the objective function executed during the local phases are used also in the course of the global search. Two different local improvement techniques have been introduced in [45]: Pessimistic Local Improvement and Optimistic Local Improvement. The first one implements the local search procedures only until a predefined accuracy (which cannot be smaller than the global search accuracy) is reached. After this the global optimization phase restarts its work and local and global phases can be repeated many times. In this way it is guaranteed that the search can stop only after the satisfaction of the global search stopping condition in order to keep theoretical convergence to the global minimizers. In the Optimistic Local Improvement strategy, the search can stop after satisfaction of the local search stopping criterion before convergence to the global minimizers. This can be reasonable if the computa-

tional budget is limited and the best objective function's value is required as soon as possible.

4 New Methods and Their Direct Competitors

Let us consider the following univariate optimization problem: it is required to find the point ε^* and the corresponding value LOOCV^*, where $\mathrm{LOOCV}(\varepsilon)$ is from (9), such that

$$\mathrm{LOOCV}^* = \mathrm{LOOCV}(\varepsilon^*) = \min \mathrm{LOOCV}(\varepsilon), \quad \varepsilon \in D = [0, \varepsilon_{max}] \subset \mathbb{R}, \quad (12)$$

where ε_{max} is large enough.

In this section, we describe algorithms for solving the problem (12) that use as a basic step the algorithms from [45] that were modified as follows to find an optimal shape parameter ε.

General Scheme of the locally-biased versions of the univariate global optimization algorithms for the minimization of the error function using LOOCV:

Step 1. *Preliminary Search.* A global optimization algorithm belonging to the General Scheme from [45] is chosen and launched over the interval $[0, \varepsilon_{max}]$, using N_1^{max} maximum number of trials and generating N_1^{max} values $\varepsilon_{i,1},\ i = 1, ..., N_1^{max}$:

$$0 = \varepsilon_{1,1} < \varepsilon_{2,1} < ... < \varepsilon_{N_1^{max},1} = \varepsilon_{max}.$$

Step 2. *Ill-conditioned region refinement.* Since the error function can be ill-conditioned if ε is close to 0, the algorithm is applied for the same problem at the search interval $[0, \varepsilon_{2,1}]$, using N_2^{max} maximum number of trials and generating N_2^{max} values $\varepsilon_{i,2},\ i = 1, ..., N_2^{max}$:

$$0 < \varepsilon_{1,2} < \varepsilon_{2,2} < ... < \varepsilon_{N_2^{max},2} < \varepsilon_{2,1}.$$

Step 3. *Aggregation.* Both the sets of trials are unified to one set

$$0 = \varepsilon_1 < \varepsilon_2 < ... < \varepsilon_{N_2^{max}} < ... < \varepsilon_{N_1^{max}+N_2^{max}} = \varepsilon_{max},$$

where

$$\varepsilon_i = \begin{cases} \varepsilon_{i-1,2}, & if\ 1 < i \leq N_2^{max} + 1, \\ \varepsilon_{i-N_2^{max},1}, & if\ i > N_2^{max} + 1. \end{cases}$$

Step 4. *Restriction of the search interval.* The best obtained value ε_j with the smallest error, i.e.,

$$j = \underset{i=1,...,N_1^{max}+N_2^{max}}{argmin} (\mathrm{LOOCV}(\varepsilon_i)),$$

after $N_1^{max} + N_2^{max}$ trials is determined. The subset

$$\varepsilon_{imin}, \varepsilon_{imin+1}, ..., \varepsilon_j, ..., \varepsilon_{imax-1}, \varepsilon_{imax}$$

is kept, where

$$imin = \max(1, j - d_x), \quad imax = \min(N_1^{max} + N_2^{max}, j + d_x),$$

where $d_x,\ d_x \geq 0$, is the number of trials closest to the current minimum from the left or from the right taken for the main search step. If $d_x = 0$, then only the preliminary search and the ill-conditioned region refinement are performed without the main search step. If $d_x \geq N_1^{max} + N_2^{max}$, then all trials obtained during the first two steps are used also in the main search.

Step 5. *The Main Search.* The global search is performed over the locally biased interval $[\varepsilon_{imin}, \varepsilon_{imax}]$, using all the values $\varepsilon_i,\ i = imin, imin + 1, ..., imax$, obtained previously, until a stopping criterion is satisfied.

For each phase of the General Scheme described above, the search stopped either if the length of the subinterval for the next subdivision became smaller than δ or if the maximum number of trials (indicated as $N_1^{max},\ N_2^{max}$, and N^{max} for the preliminary search, ill-conditioned region refinement, and the main search, respectively) was achieved.

The motivation of this procedure is the following. First, as it can be seen, e.g., from Figures 2–3, the error function has a lot of local minimizers within very small regions of attraction due to ill-conditioning of the function $g(x)$. In this case, the global search is not able to investigate well these minimizers at the initial trials, but it produces several promising values investigating the trend during the first run over the interval $[0, \varepsilon_{max}]$. Then, as it has been shown in many papers and books (see e.g. [8,14,15,18] and references therein), the error function LOOCV(ε) become significantly ill-conditioned with small values of ε. In this case, it can be reasonable to study better an interval with small values of ε, than intervals with large values of ε, as it is done during the second run over the interval $[0, \varepsilon_{2,1}]$ that still is reasonably wide. Then, after $N_1^{max} + N_2^{max}$ preliminary iterations the search in the neighborhood $[\varepsilon_{imin}, \varepsilon_{imax}]$ of the best found value LOOCV(ε_j) is performed. In this case, the search interval becomes smaller and the oscillations of the error function become more evident to the method, allowing so to find a good value of the shape parameter ε.

Since a concrete implementation of a univariate Lipschitz global optimization algorithm is used at each iteration, then the convergence conditions of the proposed algorithms to the global minimizers coincide with the ones of the standard geometric and information global optimization algorithms from [45], if the value d_x is large enough, i.e., if the main search interval $[\varepsilon_{imin}, \varepsilon_{imax}]$ contains the global solution ε^*. In particular, if $d_x \geq N_1^{max} + N_2^{max}$, then the main search step consists of the application of the global optimization algorithms on the whole search interval $[0, \varepsilon_{max}]$ using additional information obtained from the first two steps and, as a consequence, the convergence of the biased algorithm to the global minimizer is satisfied. The values of d_x smaller than $N_1^{max} + N_2^{max}$ are used only to accelerate the global search.

The following two locally-biased global optimization methods (GOPI and GOOI) were used in our experiments since the performance of their original not biased versions was one of the best for the problems with large Lipschitz constants (see [45]). They were combined with the LOOCV technique and were named:

- LOOCV-GOPI – Information Global Optimization Algorithm with Pessimistic Local Improvement.
- LOOCV-GOOI – Information Global Optimization Algorithm with Optimistic Local Improvement.

A standard LOOCV method from [14] using the uniform grid of the values ε with the stepsize $h = \varepsilon_{max}/499$ and the LOOCV method with the MATLAB's minimization procedure *fminbnd* (called LOOCV-min) were used for the comparison with the described above global optimization methods. The procedure *fminbnd* implements the local search method based on golden-section search and parabolic interpolation of the objective function. It uses the parameter $tolX$ (which has been set to 10^{-15} in our experiments in order to achieve the machine precision) for the stopping criterion. All other parameters of the method were set by their default values. Note that the standard LOOCV method mentioned above can be viewed as the reference method for finding the minimum of an error function (12). This observation turns out to be very important to fully/further esteem numerical and graphical results we will show in detail in Sections 5–6.

5 Numerical Experiments and Discussion

In this work, information univariate global optimization algorithms from [45] are used for the global search in all three phases of the General Scheme described in the previous section. The value ε_{max} was set equal to 20 in our experiments. The "Maximum-additive" local tuning with the reliability parameter $r = 2$ is used as one of the best techniques for estimating the local Lipschitz constants. On the first two preliminary phases (over the intervals $[0, \varepsilon_{max}]$ and $[0, \varepsilon_{2,1}]$, respectively) the optimistic local improvement is used. The value δ for the stopping criterion has been set to 0.1 and the values N_1^{max} and N_2^{max} have been set to 12 and 10, respectively. The value d_x has been set to 5 in our experiments. For the main phase, both the pessimistic and optimistic local improvement techniques have been used. The stopping criterion δ on this phase has been set to 10^{-2} and 10^{-3} in order to study if increasing the accuracy of the global search improves the result significantly. The local improvement accuracy δ_{LI} for the pessimistic local improvement has been set equal to δ as it is recommended in [45]. The maximum number of trials N^{max} on this phase was set to 5000.

The algorithms have been coded and compiled in MATLAB (R2011b) on a HP-15-ba090ur machine under the MS Windows 10 operating system.

In the experiments we analyzed the performance of the methods LOOCV, LOOCV-min, LOOCV-GOPI, and LOOCV-GOOI by using the Gaussian RBF

(see Table 1) and considering the following eight test problems/functions taken from [12,14,31]:

$$
\begin{aligned}
\#1:\ f_1(x_1,x_2) &= \frac{3}{4}\exp\left(-\frac{(9x_1-2)^2+(9x_2-2)^2}{4}\right)+\frac{3}{4}\exp\left(-\frac{(9x_1+1)^2}{49}-\frac{9x_2+1}{10}\right)\\
&\quad+\frac{1}{2}\exp\left(-\frac{(9x_1-7)^2+(9x_2-3)^2}{4}\right)-\frac{1}{5}\exp\left(-(9x_1-4)^2-(9x_2-7)^2\right),\\
\#2:\ f_2(x_1,x_2) &= \cos\left(10(x_1+x_2)\right),\\
\#3:\ f_3(x_1,x_2) &= (x_1+x_2-1)^9,\\
\#4:\ f_4(x_1,x_2) &= \exp\left(-\frac{1}{4}\left(x_1^2+(x_2+0.9)^2\right)\right)+\exp\left(-\frac{1}{4}\left(x_1^2+(x_2-1.1)^2\right)\right)\\
&\quad+\exp\left(-\frac{1}{4}\left((x_1+0.4)^2+x_2^2\right)\right)+\exp\left(-\frac{9}{25}\left((x_1-0.2)^2+x_2^2\right)\right),\\
\#5:\ f_5(x_1,x_2) &= \exp\left(-\left(x_1^2+(x_2+1.2)^2\right)\right)+2\exp\left(-\left((x_1+0.4)^2+(x_2-0.5)^2\right)\right)\\
&\quad-2\exp\left(-\left((x_1+0.4)^2+(x_2-1.1)^2\right)\right)\\
&\quad+3\exp\left(-\left((x_1-1.2)^2+(x_2-1.3)^2\right)\right),\\
\#6:\ f_6(x_1,x_2) &= \exp\left(|x_1-x_2|\right)-1,\\
\#7:\ f_7(x_1,x_2) &= \sin(x_1)+\cos(x_2),\\
\#8:\ f_8(x_1,x_2) &= 4^2\prod_{i=1}^{2}x_i(1-x_i).
\end{aligned}
$$

Note that these bivariate functions are commonly used (see e.g. [10,12,31]) in approximation processes to test and validate new methods and algorithms, then making them usable in the field of applications.

For each test problem, we took $n = 289$ Halton-type data points that were generated in the unit square $\Omega = [0,1]\times[0,1]$ by the MATLAB program *haltonseq.m* (see [14]). This node distribution is a typical example of uniformly random or scattered data point set. The results are given in Tables 2–3.

In Table 2, for each test problem, the number of performed trials N, the best found value of ε^* and the respective value of the error-function LOOCV(ε^*) are presented for all methods (for the basic method LOOCV, only the values ε^* and LOOCV(ε^*) are presented since the number of trials for each test problem was set equal to 500). Table 3 presents for the same algorithms and test problems the best obtained errors. As it can be seen from Tables 2–3, the local search method LOOCV-min was not able to find the global minimum for the problem #5. Moreover, its obtained errors are always the worst ones for all test problems (except the problems #1 and #6, where they coincide with the errors obtained by the other methods). It can be seen also that the best error values have been obtained by the LOOCV-GOPI method with $\delta = 10^{-3}$ for test problems #1, #4–#8, while the method LOOCV-GOOI was the best one on the problems #1, #4, #6 and #7. It should be noted that the method with the pessimistic local improvement (LOOCV-GOPI) executed much more trials,

than other methods, due to large Lipschitz constants. However, the method with the optimistic local improvement (LOOCV-GOOI) performs much less trials obtaining small enough errors (in fact, the errors obtained by the method LOOCV-GOOI were the best ones for four test problems). It can be seen also that the number of trials performed by the LOOCV-GOOI method is not increased significantly if the higher accuracy δ is used.

Figures 2–3 present the graphs of the error functions being the objective functions for the optimization problems and the best obtained values by the method LOOCV, LOOCV-min, and LOOCV-GOOI (with $\delta = 10^{-3}$). It can be seen that the obtained value ε^* by the method LOOCV-GOOI is always located at the neighborhood of the global solution of each problem and is much closer to it than the value obtained by the LOOCV-min method.

Finally, Table 4 presents the execution times for three algorithms LOOCV, LOOCV-min, and LOOCV-GOOI (with $\delta = 10^{-3}$). Here, for each test problem the number of performed trials, the execution time for each algorithm and the execution time per trial, i.e., the execution time divided by the number of trials, are presented (for the method LOOCV the number of trials is not indicated since it is equal to 500 for each test problem). As it can be seen from Table 4, the working time per trial is almost always the smallest for the global search method. This means that the main impact to the working time is devoted to computations of the error function and that the internal procedures of the method are not heavy in terms of temporal resources. It can be seen also that the total working time of the method LOOCV-GOOI for each test problem is comparable with the execution time of the method LOOCV-min, while the execution time of the standard LOOCV method is more than ten times higher.

#	LOOCV		LOOCV-min			LOOCV-GOPI $\delta = 10^{-2}$			LOOCV-GOOI $\delta = 10^{-2}$			LOOCV-GOPI $\delta = 10^{-3}$			LOOCV-GOOI $\delta = 10^{-3}$		
	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)
1	6.212	2.23e-03	37	6.213	2.23e-03	57	6.212	2.23e-03	47	6.212	2.23e-03	60	6.213	2.23e-03	55	6.213	2.23e-03
2	2.966	1.98e-04	32	4.085	4.49e-04	47	3.754	2.14e-04	38	3.564	2.52e-04	54	3.753	2.14e-04	54	3.753	2.14e-04
3	2.485	4.38e-04	38	3.051	6.88e-04	51	2.456	5.31e-04	28	3.0906	6.00e-04	141	2.429	5.22e-04	54	2.429	5.23e-04
4	0.521	6.36e-06	40	3.255	5.07e-05	260	0.505	3.11e-06	37	0.505	3.11e-06	2137	0.511	3.05e-06	49	0.511	3.05e-06
5	1.082	8.64e-06	34	4.772	1.05e-03	242	1.029	5.75e-06	35	0.854	7.55e-06	2431	1.027	5.67e-06	43	0.856	7.25e-06
6	7.976	1.07e-01	40	7.943	1.07e-01	67	7.945	1.07e-01	47	7.945	1.07e-01	74	7.943	1.07e-01	59	7.943	1.07e-01
7	2.084	1.36e-05	29	3.284	6.69e-05	417	2.075	1.30e-05	43	1.455	1.75e-05	3743	2.074	1.29e-05	77	2.074	1.29e-05
8	2.124	9.97e-05	34	3.189	2.36e-04	35	2.621	1.49e-04	35	2.621	1.49e-04	202	2.12	9.80e-05	41	2.619	1.48e-04

Table 2: Results on eight test problems for the methods LOOCV, LOOCV-min, LOOCV-GOPI with $\delta = 10^{-2}$, LOOCV-GOOI with $\delta = 10^{-2}$ and the algorithms LOOCV-GOPI and LOOCV-GOOI with $\delta = 10^{-3}$.

#	LOOCV	LOOCV-min	LOOCV-GOPI $\delta = 10^{-2}$	LOOCV-GOOI $\delta = 10^{-2}$	LOOCV-GOPI $\delta = 10^{-3}$	LOOCV-GOOI $\delta = 10^{-3}$
1	**2.23e-03**	**2.23e-03**	**2.23e-03**	**2.23e-03**	**2.23e-03**	**2.23e-03**
2	**1.98e-04**	4.49e-04	2.14e-04	2.52e-04	2.14e-04	2.14e-04
3	**4.38e-04**	6.88e-04	5.31e-04	6.00e-04	5.22e-04	5.23e-04
4	6.36e-06	5.07e-05	3.11e-06	3.11e-06	**3.05e-06**	**3.05e-06**
5	8.64e-06	1.05e-03	5.75e-06	7.55e-06	**5.67e-06**	7.25e-06
6	**1.07e-01**	**1.07e-01**	**1.07e-01**	**1.07e-01**	**1.07e-01**	**1.07e-01**
7	1.36e-05	6.69e-05	1.30e-05	1.75e-05	**1.29e-05**	**1.29e-05**
8	9.97e-05	2.36e-04	1.49e-04	1.49e-04	**9.80e-05**	1.48e-04

Table 3: Results on eight test problems for the methods LOOCV, LOOCV-min, LOOCV-GOPI with $\delta = 10^{-2}$, LOOCV-GOOI with $\delta = 10^{-2}$ and the algorithms LOOCV-GOPI and LOOCV-GOOI with $\delta = 10^{-3}$. Only the best obtained errors (LOOCV(ε^*)) are presented in this table. The best obtained value for each test problem is bolded.

#	LOOCV		LOOCV-min			LOOCV-GOOI		
	Total	Per trial	N	Total	Per trial	N	Total	Per trial
1	40.53552	0.08107	37	3.22887	0.08727	55	3.91932	0.07126
2	39.43944	0.07888	32	2.09048	0.06533	54	3.2435	0.06006
3	36.0497	0.0721	38	2.06889	0.05444	54	2.97745	0.05514
4	34.81647	0.06963	40	2.31993	0.058	49	2.34862	0.04793
5	34.71593	0.06943	34	2.31109	0.06797	43	2.08862	0.04857
6	39.17958	0.07836	40	3.5633	0.08908	59	3.98124	0.06748
7	39.96465	0.07993	29	1.82254	0.06285	77	4.16466	0.05409
8	40.42387	0.08085	34	2.13739	0.06286	41	2.62396	0.064

Table 4: Working times (in seconds) of the methods LOOCV, LOOCV-min, and LOOCV-GOOI (with $\delta = 10^{-3}$) on eight test problems. For each test problem, the number of trials, the execution time and the execution time divided by the number of trials are indicated (for the method LOOCV the number of trials is not indicated since it is equal to 500 for each test problem).

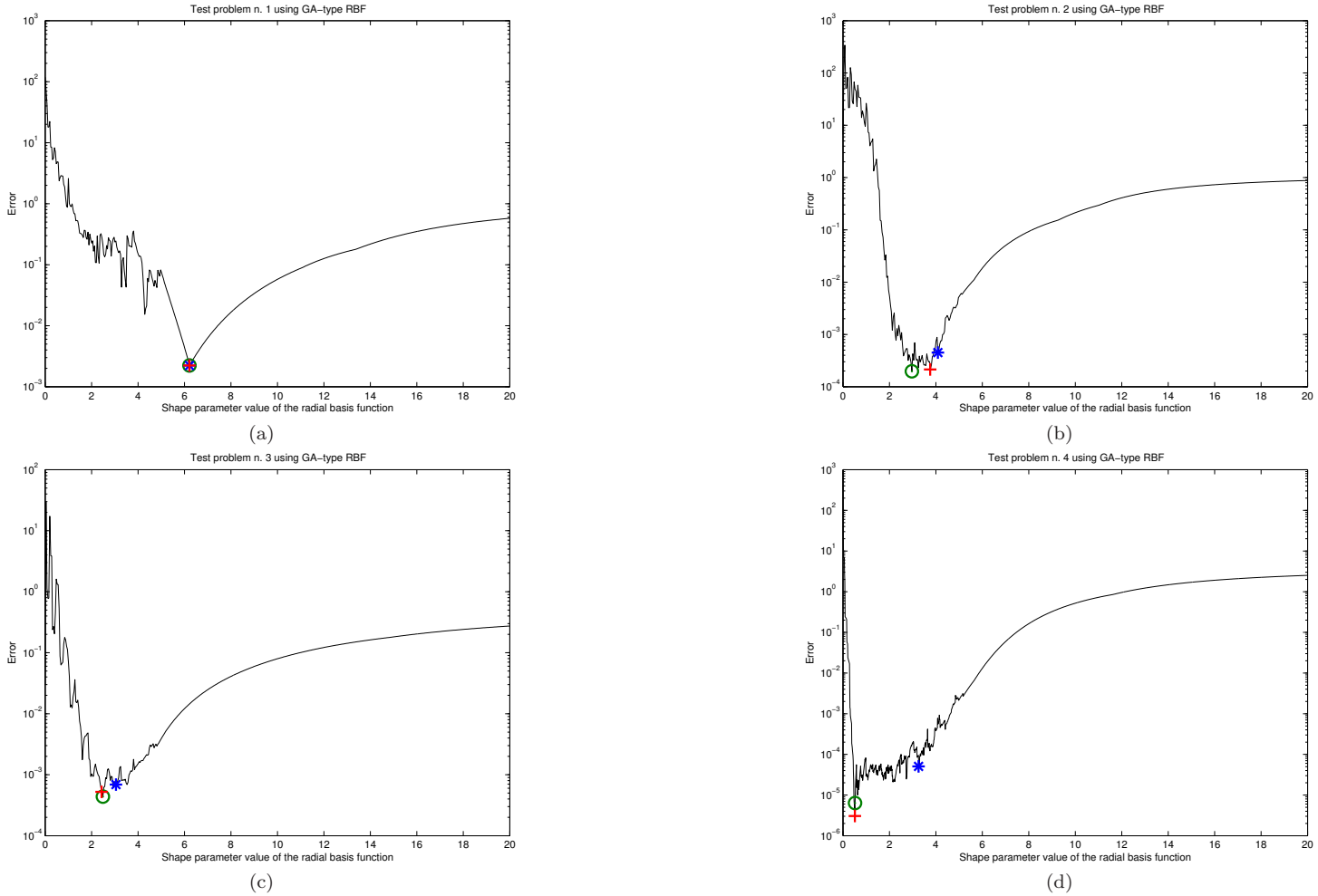


Fig. 2: The first four error functions LOOCV(ε) used in the experiments. The best found values by LOOCV, LOOCV-min, and LOOCV-GOOI (with $\delta = 10^{-3}$) are indicated as "o", "*", and "+", respectively.

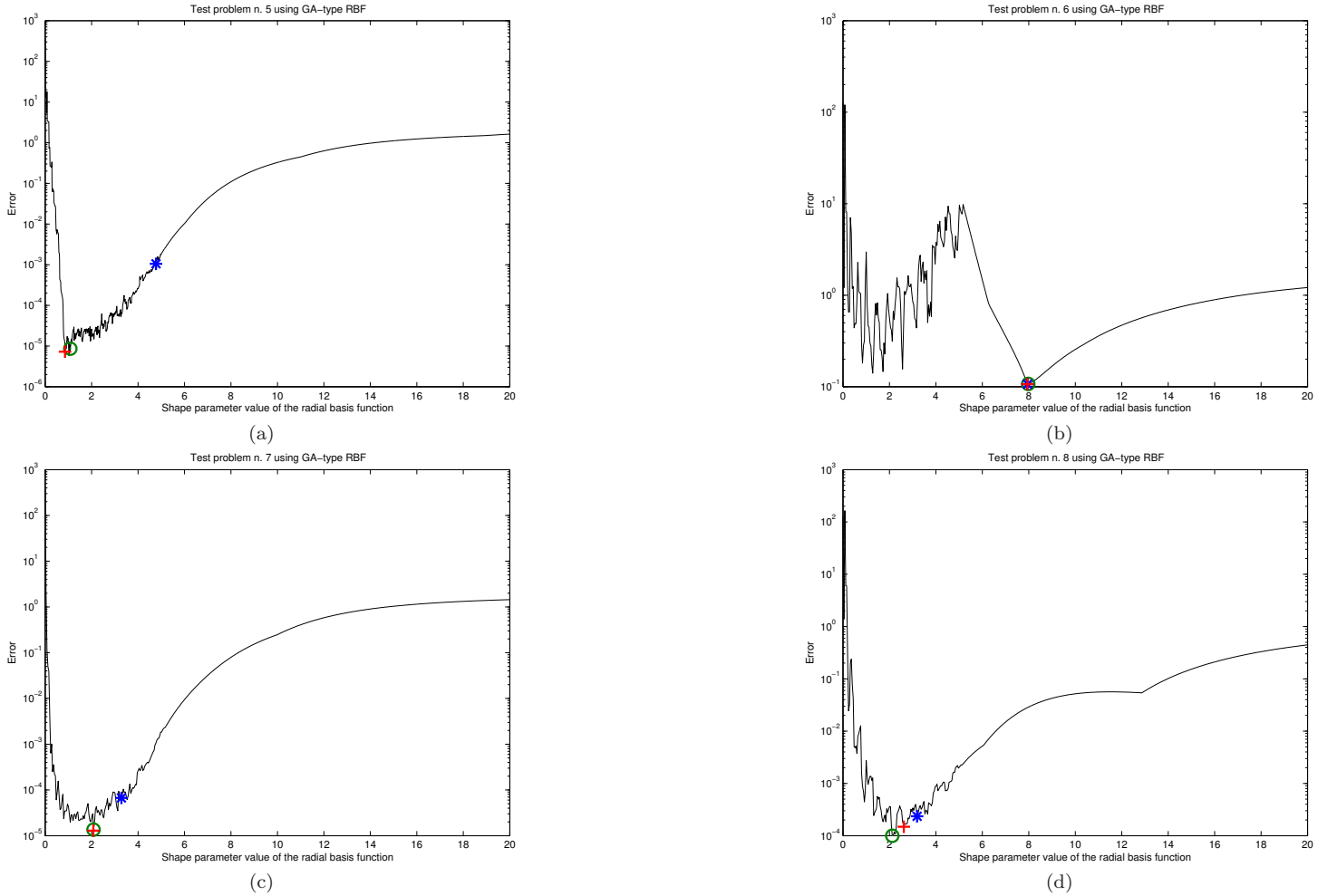


Fig. 3: The last four error functions LOOCV(ε) used in the experiments. The best found values by LOOCV, LOOCV-min, and LOOCV-GOOI (with $\delta = 10^{-3}$) are indicated as "o", "*", and "+", respectively.

6 Applications

6.1 Application I: Big Sur Bathymetric Data

In this subsection we consider an application to bathymetric data, known as *Big Sur* [7]. The data consist of 64 water temperature measurements taken from a boat traveling on tracks approximately perpendicular to the shore off the coast of Big Sur, California. More precisely, this data set lies along five distinct tracks scaled to the unit square.

The method LOOCV-GOOI with $\delta = 10^{-3}$ has been used as global optimization method and compared with the methods LOOCV and LOOCV-min. As different types of the RBFs lead to different types of the error-function to be optimized (which can be much more ill-conditioned for the RBFs with large orders of smoothness), we used different values of the reliability parameter r of the method LOOCV-GOOI. In particular, for the GA, IMQ, IQ, M6 and W6 RBFs we used $r = 12$ for the preliminary search, $r = 8$ for the ill-conditioned region refinement, and $r = 4$ for the main search, for the M4 and W4 RBFs we used $r = 6$ for both the preliminary search and the ill-conditioned region refinement and $r = 4$ for the main search. Finally, for the M2 and W2 RBFs we used the value $r = 2$ for all three stages of the search. The obtained results are presented in Table 5 and Figures 4–5.

RBF #	LOOCV			LOOCV-min			LOOCV-GOOI $\delta = 10^{-3}$	
	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)
GA	0.56112	4.75e+00	24	7.59435	9.71e+00	65	0.59199	4.59e+00
IMQ	8.13627	2.65e+00	39	8.13957	2.65e+00	62	8.1397	2.65e+00
IQ	0.16032	4.26e+00	40	5.99119	4.52e+00	47	0.30108	3.01e+00
M6	0.52104	3.17e+00	31	16.00471	6.01e+00	49	0.55084	3.12e+00
M4	0.24048	2.13e+00	33	5.23072	2.97e+00	49	0.24023	2.12e+00
M2	0.04008	8.32e-01	40	0.03869	8.32e-01	61	0.05351	8.32e-01
W6	0.08016	3.49e+00	22	1.92589	8.50e+00	38	0.07003	3.37e+00
W4	0.04008	2.13e+00	40	0.5117	3.44e+00	58	0.04097	2.13e+00
W2	0.52104	8.29e-01	44	0.52415	8.29e-01	55	0.52387	8.29e-01

Table 5: Results on the Big Sur problem using nine different RBFs from Table 1 for the methods LOOCV, LOOCV-min, and LOOCV-GOOI with $\delta = 10^{-3}$.

It can be seen from Table 5 and Figures 4–5 that the method LOOCV-GOOI performs better than the method LOOCV and the local search method LOOCV-min on this problem. In particular, it executes a smaller number of trials with respect to the LOOCV method but the obtained error is the smallest for the GA, IQ, M6, M4, and W6 RBFs, while for the W4 the error obtained by the LOOCV-GOOI is smaller than the error obtained by the LOOCV-min and for the remaining RBFs the error is the same for all three methods. Even though the number of trials executed by the LOOCV-min method is smaller for all RBFs than the number of trials executed by the LOOCV-GOOI (but always of the same order) the errors obtained by the LOOCV-GOOI method are better than the ones obtained by the LOOCV-min.

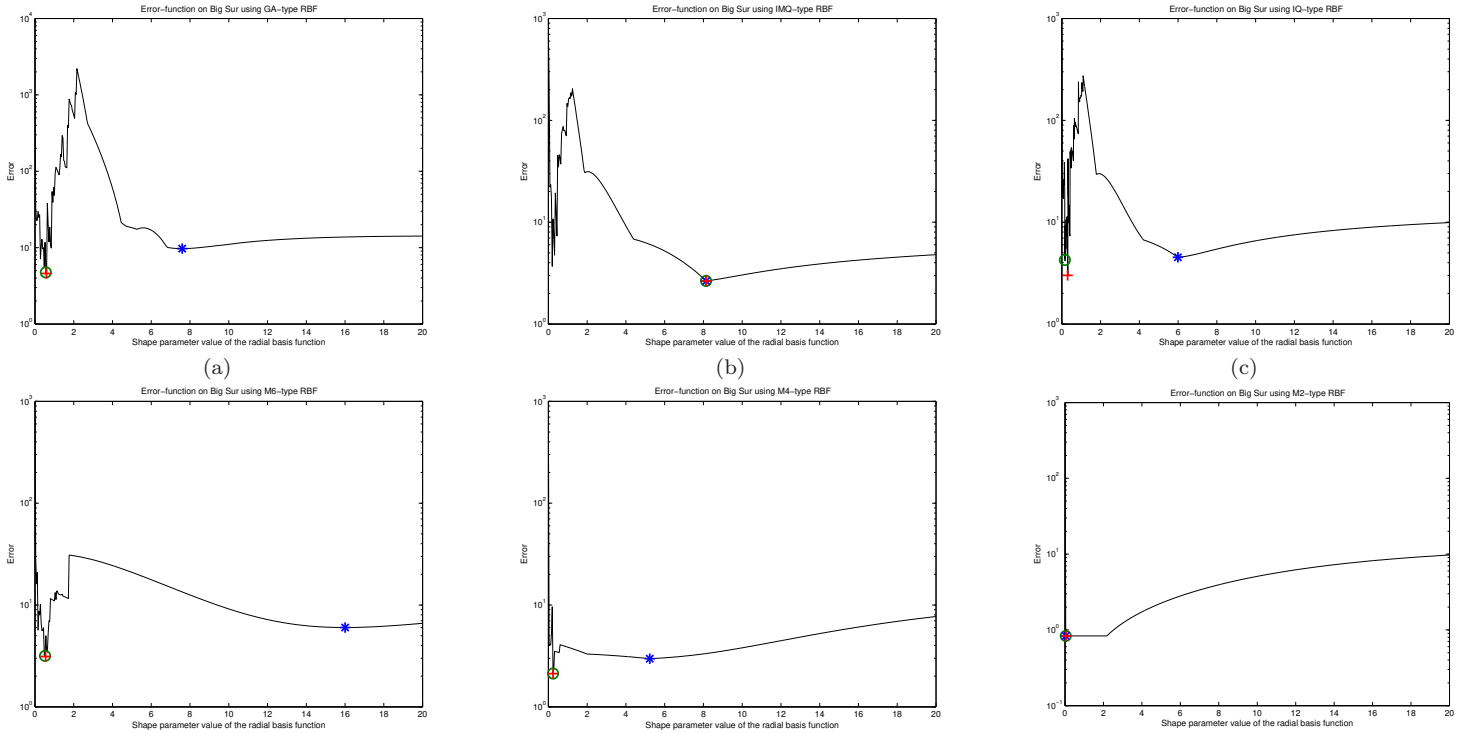
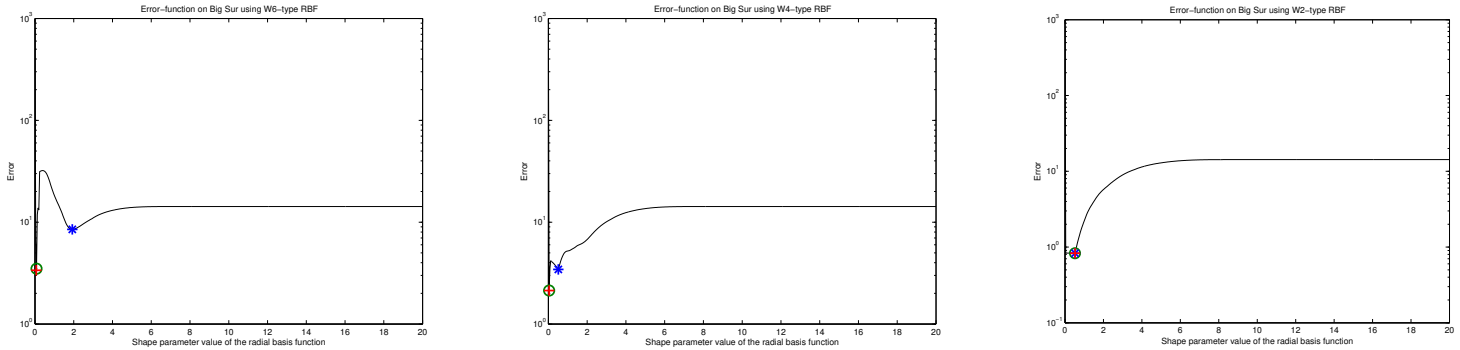


Fig. 4: The error functions LOOCV(ε) obtained on the Big Sur problem using the first six RBFs from Table 1. The best found values by LOOCV, LOOCV-min, and LOOCV-GOOI (with $\delta = 10^{-3}$) are indicated as "o", "*", and "+", respectively.



(a) (b) (c)

Fig. 5: The error functions LOOCV(ε) obtained on the Big Sur problem using the last three RBFs from Table 1. The best found values by LOOCV, LOOCV-min, and LOOCV-GOOI (with $\delta = 10^{-3}$) are indicated as "o", "*", and "+", respectively.

Fig. 6: A view of the Maunga Whau (Mt. Eden) volcano in Auckland, NZ.

6.2 Application II: Maunga Whau Volcano Data

In this subsection, we consider an application to Earth's topography to test our new algorithms on a set of real world data, i.e., the Maunga Whau volcano data points, available in [33]. These data represent 5307 elevation measurements obtained from Maunga Whau (Mt. Eden) in Auckland, New Zealand, sampled on a 10 m $\times$ 10 m grid. A picture of this area is shown in Figure 6.

To analyze the behavior of our global RBF interpolation method combined with the global optimization routines, we reduce the number of original volcano data points randomly selecting only $n = 118$ nodes for the interpolation process. The method LOOCV-GOOI with $\delta = 10^{-3}$ has been used as global optimization method and compared with the methods LOOCV and LOOCV-min. The parameters of the methods have been set as previously. The obtained results are presented in Table 6 and Figures 8–9.

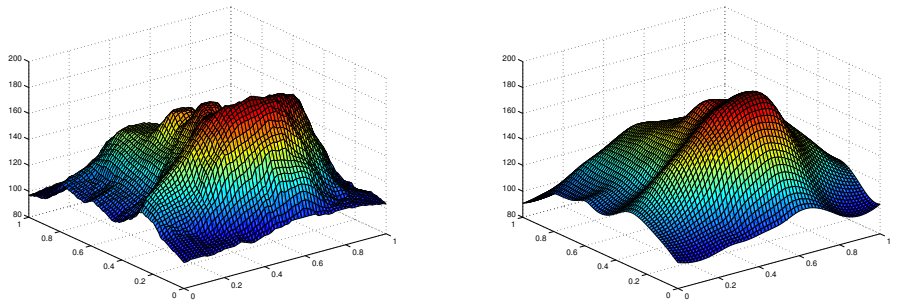
It can be seen from Table 6 and Figures 8–9 that the results are very similar to those obtained for the Big Sur problem. In particular, the method LOOCV-GOOI again performs better than the method LOOCV and the local search method LOOCV-min on this problem: it executes a smaller number of trials with respect to the LOOCV method but the obtained error is the smallest for the GA, IMQ, IQ, M4, W6, W4, and W2 RBFs, while for the remaining RBFs the error is the same for all three methods. Again, even though the number of trials executed by the LOOCV-min method is smaller for all RBFs than the number of trials executed by the LOOCV-GOOI (but always of the same order) the errors obtained by the LOOCV-GOOI method are better than the ones obtained by the LOOCV-min (for instance, the error obtained by the LOOCV-GOOI for the W4 RBF is almost ten times smaller than the one obtained by the LOOCV-min).

In Figure 7, the surfaces of the approximated function are presented using all 5307 points without interpolation (left) and using the M4-type RBF interpolant with the value $\varepsilon = 0.25074$ obtained by the LOOCV-GOOI method are presented. The surface of the interpolated function has been constructed as follows. First, the "optimal" value of ε has been taken from the Table 6 and the

respective type of the RBF has been chosen. Then, the interpolant function $\mathcal{I}_f(x)$ from (1) has been obtained. Finally, the surface of the interpolant using the remaining $n = 5307 - 118 = 5189$ points has been built. It can be seen that even using only 118 points, the surfaces are qualitatively similar, but, as it is expected, the higher accuracy can be reached using the higher number of data points.

RBF #	LOOCV			LOOCV-min			LOOCV-GOOI $\delta = 10^{-3}$	
	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)	N	ε^*	LOOCV(ε^*)
GA	2.20441	5.47e+01	39	9.10103	1.01e+02	65	2.22402	5.35e+01
IMQ	6.17234	1.42e+01	36	6.18574	1.41e+01	59	6.18576	1.41e+01
IQ	4.60922	2.93e+01	28	4.59072	2.90e+01	54	4.59076	2.90e+01
M6	5.09018	1.89e+01	39	5.09485	1.89e+01	57	5.09448	1.89e+01
M4	0.24048	1.13e+01	42	2.08086	1.25e+01	43	0.25074	9.11e+00
M2	1.12224	1.62e+01	25	1.11316	1.62e+01	51	1.11307	1.62e+01
W6	0.64128	2.92e+01	38	6.72209	1.03e+02	43	0.62448	2.91e+01
W4	0.08016	1.26e+01	39	5.36959	1.03e+02	50	0.08803	1.26e+01
W2	0.12024	1.75e+01	39	7.63932	1.03e+02	57	0.12567	1.75e+01

Table 6: Results on the Maunga Whau Volcano problem using nine different RBFs from Table 1 for the methods LOOCV, LOOCV-min, LOOCV-GOOI with $\delta = 10^{-3}$.



(a) (b)

Fig. 7: The surface of the Maunga Whau Volcano function using real values of 5307 data points (left) and using M4-type RBF with the value $\varepsilon = 0.25074$ obtained by the LOOCV-GOOI method with $\delta = 10^{-3}$ and using 118 data points (right).

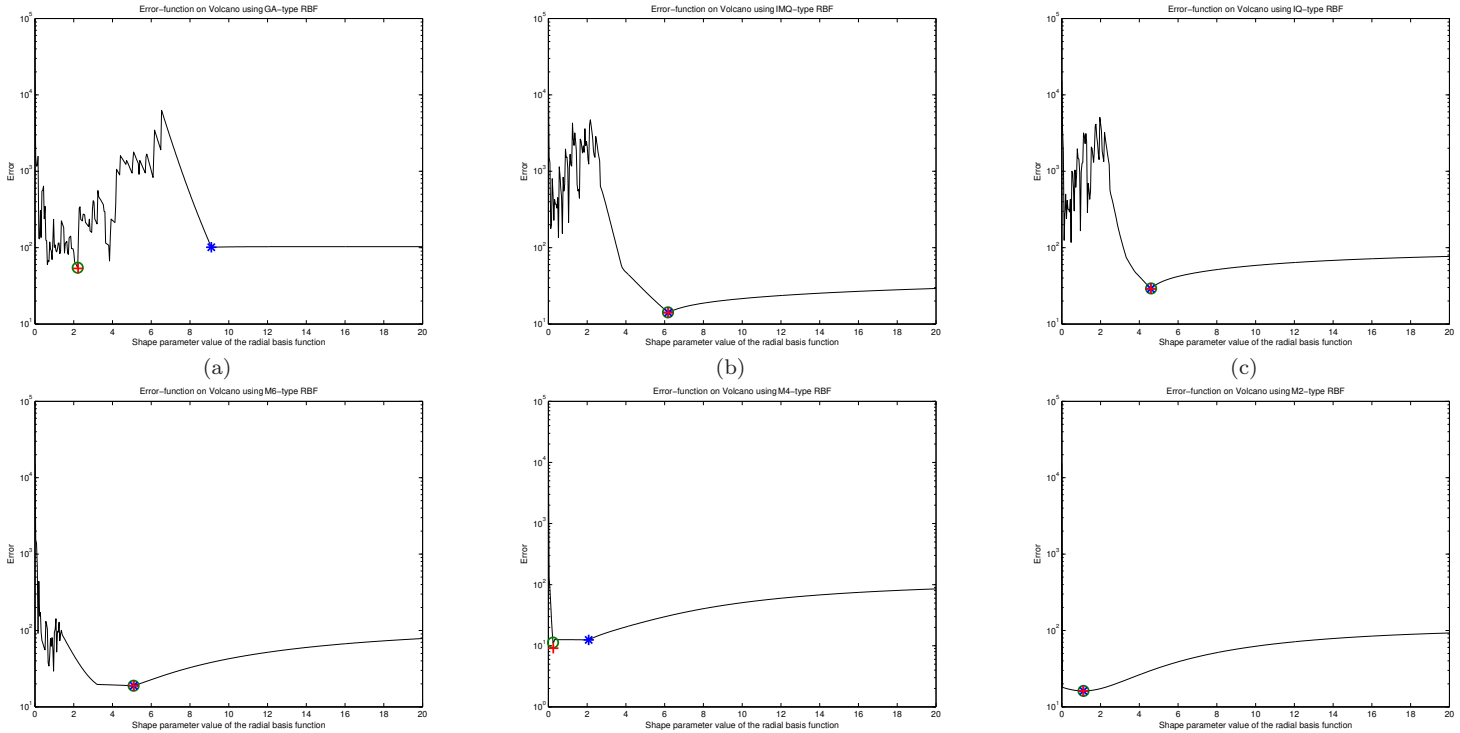
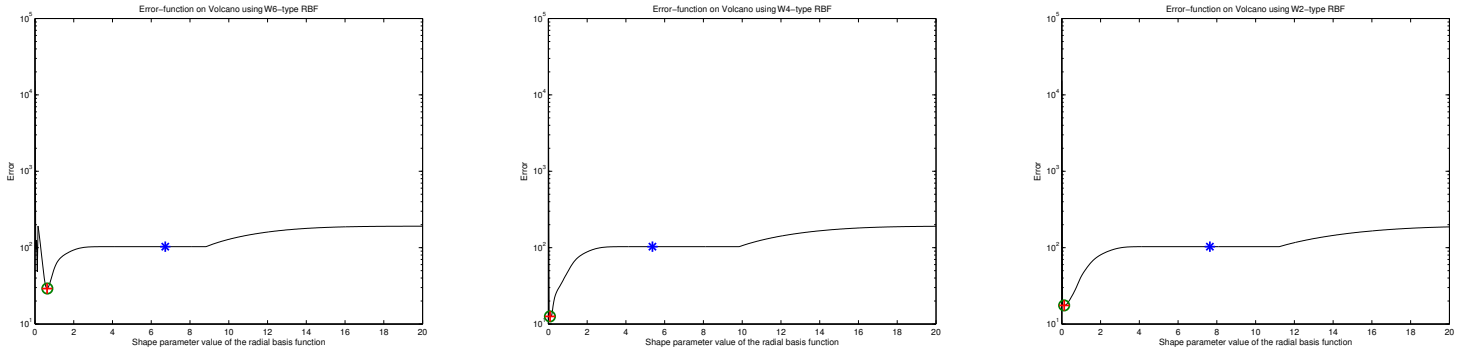


Fig. 8: The error functions LOOCV(ε) obtained on the Maunga Whau Volcano problem using the first six RBFs from Table 1. The best found values by LOOCV, LOOCV-min and LOOCV-GOOI (with $\delta = 10^{-3}$) are indicated as "o", "*" and "+", respectively.



(a) (b) (c)

Fig. 9: The error functions LOOCV(ε) obtained on the Maunga Whau Volcano problem using the last three RBFs from Table 1. The best found values by LOOCV, LOOCV-min and LOOCV-GOOI (with $\delta = 10^{-3}$) are indicated as "o", "*" and "+", respectively.

7 Conclusions and Future Work

In this paper we presented two new algorithms, called LOOCV-GOPI and LOOCV-GOOI, for fast and accurate detecting optimal shape parameters in the RBF interpolation. The basic approach consists of combining a LOOCV-based technique with some efficient univariate global optimization methods. Numerical results showed a promising performance of our algorithms on test problems and real-life applications, pointing out their efficacy in finding the global minimum of the error function. In general, this fact enabled a better localization of the optimal RBF shape parameter in comparison with existing optimization routines.

As future work we propose to investigate the problem of finding optimal shape parameters in RBF partition of unity interpolation (see [9]). This method is based on the solution of a number of local RBF interpolation problems. So the selection of the local RBF shape parameters turns out to be a crucial task to achieve accurate approximation results. Another situation in which the choice of optimal RBF shape parameters is important concerns the solution of partial differential equations via collocation or pseudo-spectral RBF methods [14]. It is expected that for each of these applications the extension of the algorithms discussed in this paper might bring out analogous improvements as compared with standard optimization routines.

References

1. Allasia, G., Cavoretto, R., De Rossi, A.: Lobachevsky spline functions and interpolation to scattered data. Computational and Applied Mathematics **32**, 71–87 (2013)
2. Barkalov, K., Gergel, V., Lebedev, I.: Solving global optimization problems on GPU cluster. In: T.E. Simos (ed.) AIP Conference Proceedings, vol. 1738 (400006) (2016)
3. Barkalov, K., Strongin, R.: Solving a set of global optimization problems by the parallel technique with uniform convergence. Journal of Global Optimization **71(1)**, 21–36 (2018)
4. Biazar, B., Hosami, M.: An interval for the shape parameter in radial basis function approximation. Applied Mathematics and Computation **315**, 131–149 (2017)
5. Buhmann, M.D.: Radial Basis Functions: Theory and Implementation, *Cambridge Monographs on Applied and Computational Mathematics*, vol. 12. Cambridge University Press, Cambridge (2003)
6. Calvin, J., Žilinskas, A.: On the convergence of the P-Algorithm for one-dimensional global optimization of smooth functions. Journal of Optimization Theory and Applications **102(3)**, 479–495 (1999)
7. Carlson, R.E., Foley, T.A.: Interpolation of track data with radial basis methods. Computers & Mathematics with Applications **24**, 27–34 (1992)
8. Cavoretto, R., De Rossi, A.: A trivariate interpolation algorithm using a cube-partition searching procedure. SIAM Journal on Scientific Computing **37**, A1891–A1908 (2015)
9. Cavoretto, R., De Rossi, A., Perracchione, E.: Optimal selection of local approximants in RBF-PU interpolation. Journal of Scientific Computing **74**, 1–22 (2018)
10. Cavoretto, R., Schneider, T., Zulian, P.: OpenCL based parallel algorithm for RBF-PUM interpolation. Journal of Scientific Computing **74**, 267–289 (2018)
11. Cheng, A.D.: Multiquadric and its shape parameter–A numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. Engineering Analysis with Boundary Elements **36**, 220–239 (2012)

12. De Marchi, S., Santin, G.: Fast computation of orthonormal basis for rbf spaces through krylov space methods. BIT Numerical Mathematics **55**, 949–966 (2015)
13. Fasshauer, G., McCourt, M.: Kernel-based Approximation Methods using MATLAB, *Interdisciplinary Mathematical Sciences*, vol. 19. World Scientific, Singapore (2015)
14. Fasshauer, G.E.: Meshfree Approximation Methods with MATLAB, *Interdisciplinary Mathematical Sciences*, vol. 6. World Scientific, Singapore (2007)
15. Fasshauer, G.E.: Positive definite kernels: Past, present and future. Dolomites Research Notes on Approximation **4**, 21–63 (2011)
16. Fasshauer, G.E., Zhang, J.G.: On choosing "optimal" shape parameters for RBF approximation. Numerical Algorithms **45**, 345–368 (2007)
17. Floudas, C.A., Pardalos, P.M. (eds.): Encyclopedia of Optimization (6 Volumes), 2nd edn. Springer (2009)
18. Fornberg, B., Larsson, E., Flyer, N.: Stable computations with Gaussian radial basis functions. SIAM Journal on Scientific Computing **33**, 869–892 (2011)
19. Fowkes, J.M., Gould, N.I.M., Farmer, C.L.: A branch and bound algorithm for the global optimization of Hessian Lipschitz continuous functions. Journal of Global Optimization **56**, 1791–1815 (2013)
20. Gergel, V.P., Grishagin, V.A., Israfilov, R.A.: Local tuning in nested scheme of global optimization. Procedia Computer Science **51**, 865–874 (2015)
21. Gergel, V.P., Kuzmin, M.I., Solovyov, N.A., Grishagin, V.A.: Recognition of surface defects of cold-rolling sheets based on method of localities. International Review of Automatic Control **8(1)**, 51–55 (2015)
22. Gillard, J.W., Zhigljavsky, A.A.: Stochastic algorithms for solving structured low-rank matrix approximation problems. Communications in Nonlinear Science and Numerical Simulation **21**(1-3), 70–88 (2015)
23. Golbabai, A., Mohebianfar, E., Rabiei, H.: On the new variable shape parameter strategies for radial basis functions. Computational and Applied Mathematics **34**, 691–704 (2015)
24. Grishagin, V.A., Israfilov, R.A., Sergeyev, Y.D.: Convergence conditions and numerical comparison of global optimization methods based on dimensionality reduction schemes. Applied Mathematics and Computation **318**, 270–280 (2018)
25. Kvasov, D.E., Mukhametzhanov, M.S.: Metaheuristic vs. deterministic global optimization algorithms: The univariate case. Applied Mathematics and Computation **318**, 245–259 (2018)
26. Kvasov, D.E., Pizzuti, C., Sergeyev, Y.D.: Local tuning and partition strategies for diagonal GO methods. Numerische Mathematik **94**(1), 93–106 (2003)
27. Kvasov, D.E., Sergeyev, Y.D.: Deterministic approaches for solving practical black-box global optimization problems. Advances in Engineering Software **80**, 58–66 (2015)
28. Lera, D., Sergeyev, Y.D.: Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives. SIAM Journal on Optimization **23**(1), 508–529 (2013)
29. Luh, L.T.: The shape parameter in the Gaussian function. Computers & Mathematics with Applications **63**, 687–694 (2012)
30. Paulavičius, R., Žilinskas, J.: Simplicial Global Optimization. SpringerBriefs in Optimization. Springer, New York (2014)
31. Pazouki, M., Schaback, R.: Bases for kernel-based spaces. Journal of Computational and Applied Mathematics **236**, 575–588 (2011)
32. Piyavskij, S.A.: An algorithm for finding the absolute extremum of a function. USSR Computational Mathematics and Mathematical Physics **12**(4), 57–67 (1972). In Russian: Zh. Vychisl. Mat. Mat. Fiz., 12(4) (1972), pp. 888–896
33. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2014)
34. Rippa, S.: An algorithm for selecting a good value for the parameter c in radial basis function interpolation. Advances in Computational Mathematics **11**, 193–210 (1999)
35. Sanyasiraju, Y., Satyanarayana, C.: On optimization of the RBF shape parameter in a grid-free local scheme for convection dominated problems over non-uniform centers. Applied Mathematical Modelling **37**, 7245–7272 (2013)
36. Scheuerer, M.: An alternative procedure for selecting a good value for the parameter c in RBF-interpolation. Advances in Computational Mathematics **34**, 105–126 (2011)

37. Sergeyev, Y.D.: An information global optimization algorithm with local tuning. SIAM Journal on Optimization **5**(4), 858–870 (1995)
38. Sergeyev, Y.D.: A one-dimensional deterministic global minimization algorithm. Computational Mathematics and Mathematical Physics **35**(5), 705–717 (1995)
39. Sergeyev, Y.D.: Global one-dimensional optimization using smooth auxiliary functions. Mathematical Programming **81**(1), 127–146 (1998)
40. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M.S.: On the least-squares fitting of data by sinusoids. In: P.M. Pardalos, A. Zhigljavsky, J. Žilinskas (eds.) Advances in Stochastic and Deterministic Global Optimization, pp. 209–226. Springer (2016)
41. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M.S.: Operational zones for comparing metaheuristic and deterministic one-dimensional global optimization algorithms. Mathematics and Computers in Simulation **141**, 96–109 (2017)
42. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M.S.: On strong homogeneity of a class of global optimization algorithms working with infinite and infinitesimal scales. Communications in Nonlinear Science and Numerical Simulation **59**, 319–330 (2018)
43. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M.S.: On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. Scientific Reports **8**, 453 (2018)
44. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M.S., De Franco, A.: Acceleration techniques in the univariate Lipschitz global optimization. In: Y.D. Sergeyev, D.E. Kvasov, F. Dell'Accio, M.S. Mukhametzhanov (eds.) AIP Conference Proceedings, vol. 1776 (090051) (2016)
45. Sergeyev, Y.D., Mukhametzhanov, M.S., Kvasov, D.E., Lera, D.: Derivative-free local tuning and local improvement techniques embedded in the univariate global optimization. Journal of Optimization Theory and Applications **171**(1), 186–208 (2016)
46. Strongin, R.G.: Numerical Methods in Multiextremal Problems (Information-Statistical Algorithms). Nauka, Moscow (1978). In Russian
47. Strongin, R.G., Sergeyev, Y.D.: Global optimization: fractal approach and non-redundant parallelism. Journal of Global Optimization **27(1)**, 25–50 (2003)
48. Trahan, C.J., Wyatt, R.W.: Radial basis function interpolation in the quantum trajectory method: optimization of the multi-quadric shape parameter. Journal of Computational Physics **185**, 27–49 (2003)
49. Uddin, M.: On the selection of a good value of shape parameter in solving time-dependent partial differential equations using RBF approximation method. Applied Mathematical Modelling **38**, 135–144 (2014)
50. Wahba, G.: Spline Models for Observational Data, *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 59. SIAM, Philadelphia (1990)
51. Wendland, H.: Scattered Data Approximation, *Cambridge Monographs on Applied and Computational Mathematics*, vol. 17. Cambridge University Press, Cambridge (2005)
52. Yao, G., Duo, J., Chen, C., Shen, L.: Implicit local radial basis function interpolations based on function values. Applied Mathematics and Computation **265**, 91–102 (2015)
53. Zhigljavsky, A.A., Žilinskas, A.: Stochastic Global Optimization. Springer, New York (2008)
54. Zhigljavsky, A.A., Žilinskas, A.: Selection of a covariance function for a Gaussian random field aimed for modeling global optimization problems. Optimization Letters **13**, 249–259 (2019)
55. Žilinskas, A.: On similarities between two models of global optimization: Statistical models and radial basis functions. Journal of Global Optimization **48**(1), 173–182 (2010)
56. Žilinskas, A., Zhigljavsky, A.A.: Stochastic global optimization: A review on the occasion of 25 years of Informatica. Informatica **27(2)**, 229–256 (2016)