

Article

Optimal Centers' Allocation in Smoothing or Interpolating with Radial Basis Functions

Pedro González-Rodelas ^{1,*}, Hasan M. H. Idais ^{2,†}, Mohammed Yasin ^{3,†} and Miguel Pasadas ^{1,†}¹ Departamento Matemática Aplicada, Universidad de Granada, 18071 Granada, Spain; mpasadas@ugr.es² Department of Mathematics and Statistics, Arab American University, Jenin P240, Palestine; Hasan.Idais@aaup.edu³ Department of Mathematics, An-Najah National University, Nablus P400, Palestine; m.yasin@najah.edu

* Correspondence: prodelas@ugr.es

† These authors contributed equally to this work.

Abstract: Function interpolation and approximation are classical problems of vital importance in many science/engineering areas and communities. In this paper, we propose a powerful methodology for the optimal placement of centers, when approximating or interpolating a curve or surface to a data set, using a base of functions of radial type. In fact, we chose a radial basis function under tension (RBFT), depending on a positive parameter, that also provides a convenient way to control the behavior of the corresponding interpolation or approximation method. We, therefore, propose a new technique, based on multi-objective genetic algorithms, to optimize both the number of centers of the base of radial functions and their optimal placement. To achieve this goal, we use a methodology based on an appropriate modification of a non-dominated genetic classification algorithm (of type NSGA-II). In our approach, the additional goal of maintaining the number of centers as small as possible was also taken into consideration. The good behavior and efficiency of the algorithm presented were tested using different experimental results, at least for functions of one independent variable.

Keywords: approximation; interpolation; RBFs; centers' allocation; MOGA; NSGA-II algorithms



Citation: González-Rodelas, P.; Idais, H.M.H.; Yasin, M.; Pasadas, M. Optimal Centers' Allocation in Smoothing or Interpolating with Radial Basis Functions. *Mathematics* **2022**, *10*, 59. <https://doi.org/10.3390/math10010059>

Academic Editors: Sara Remogna, Domingo Barrera and María José Ibáñez

Received: 20 November 2021

Accepted: 15 December 2021

Published: 24 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the last decades, radial basis function (RBF) methods have been developed in a way that contributed to their great spread and emergence, mainly for their simplicity and for having been lauded for the ability to solve multivariate scattered data approximation and interpolation problems (see for example [1–3], and many other references included therein). The main advantages of these methods are that they do not need a mesh or triangulation, that they are simple to implement and dimension independent, and that no staircasing or polygonization for boundaries is required [4,5].

Many authors have presented several methodologies for approximating or fitting a surface to data (see for example [6,7]), such as the methodology that is based on the polynomial interpolation theorem of Lagrange [8,9] and another important approach, that is based on orthogonal polynomials [10].

Also, some seminal works on dynamic node adaptive strategy, or knot selection placement for sparse reduced data, were published in [11,12].

In [13], the methodologies for single objective problems (SOP) have been presented for selection and optimization of centers, mainly based on techniques for the selection of center points through non-dominated sorting, where promising points are selected as centers. Also, some successful hierarchical genetic algorithm approaches for curve fitting with B-splines were applied in [14].

On the other hand, the authors in [15] developed techniques to perform centers' selection and estimation simultaneously by a componentwise boosting algorithm. Moreover,

the centers of RBFs are selected among the vertices of the Voronoi diagram of the sample data point in [16]. The methodology in [17,18] was to optimize the shape parameter of radial basis functions for the solution of ordinary differential equations. The authors in [19] used an approach based on genetic algorithms to optimize centers and radial basis function neural networks (RBFNNs). However, they used one type of neural network where the basis function is of Gaussian type. Research in neural networks has also attracted many computer and mathematical scientists over the last years and covers a broad spectrum of different subjects and real-world applications, such as image processing, pattern recognition and optimization in general (the reader can also consult the new upcoming book of the authors Grienggrai Rajchakit, Praveen Agarwal and Sriraman Ramalingam [20] for more information about the multiple applications and stability properties of these type of neural networks and many other mathematical aspects of them).

In this paper, a new methodology for optimal placement of random centers, for approximating or fitting a curve to data, using radial basis functions is developed. In fact, we chose a radial basis function under tension (RBFT) that depends on a positive parameter, that also provides a convenient way to control the behavior of the corresponding interpolation or approximation method. A new technique is presented to optimize both the number of radial basis centers and their optimal placement, using multi-objective genetic algorithms. More precisely, we used a new approach based on an appropriate modification of a goal-based non-dominated genetic classification algorithm (of type NSGA-II) and tested the good behavior and efficiency of the algorithm using different experimental results, at least for functions of one independent variable. However, the procedure and theory can be developed in the same way for functions of any number of variables, so we present it in this more general setting.

The key points concerning this goal-based NSGA-II version, compared with the original NSGA, are the following (see for example [21,22]):

- A fast, non-dominated sorting mechanism, grouping the entire population into a hierarchy of sub-populations, based on the Pareto dominance ordering.
- Inside each sub-group, the similarity between members is evaluated on the Pareto front; to promote a diverse front of non-dominated solutions, the resulting groups and similarity measures are used.
- For that, an adequate crowding distance is an appropriate mechanism of ranking members of a front which are dominating, or dominated by others.
- The appropriate consideration of a certain elitist mechanism, in order to improve the convergence properties of the algorithm.
- A niching operator without parameters is adopted, to maintain a certain level of diversity among the possible solutions.
- One or several decision maker's (DM) preferences within the evolutionary process are provided, in order to focus into a particular region of the Pareto front, where the more interesting features are present.
- In this way, a so-called region of interest (ROI) appears; it is the set of non-dominated solutions that are preferred by the DM.
- The DM can provide his/her preferences before (a priori), after (a posteriori), or during (interactively) the MOEA run.
- Therefore, some trade-off between objectives specifies that the gain of a unit on one target is worth the degradation on some others and vice versa.
- In our case, keeping the number of centers as small as possible while controlling some specific standard errors is our main goal.

Then, these ranking mechanisms are used with the corresponding genetic selection operators (usually of Tournament type) to create the population of possible solutions of the next generations.

Regarding implementation issues, we developed a MATLAB code through a suitable adaptation of some previous subroutines for the NSGA-II algorithm by Mostapha Kalami

Heris in [23] (see also [24]); we have also implemented the corresponding approximation or interpolation procedures, using the specified RBFs according with [25].

2. Proposed Methodology

Let Ω be an open bounded connected nonempty subset of \mathbb{R}^d (usually $d = 1, 2, 3$), with Lipschitz-continuous boundary. We also use the classical notation $H^k(\Omega)$ to denote the usual Sobolev space of all distributions u , whose partial derivatives, up to order k , are in the classical Lebesgue space $\mathcal{L}^2(\Omega)$.

Let $m > 1$ be a positive integer (usually small) and let $\Pi_{m-1}(\mathbb{R}^d)$ denote the space of polynomials on \mathbb{R}^d of total degree at most $m - 1$, whose dimension is denoted by $d(m)$ and let $\{q_1, \dots, q_{d(m)}\}$ be the standard basis of $\Pi_{m-1}(\mathbb{R}^d)$. Let us give an arbitrary finite set $\{\mathbf{b}_1, \dots, \mathbf{b}_M\} \subset \Omega \subset \mathbb{R}^d$ of distinct approximation points and a set of real values $U = \mathbf{f} \equiv \{f_1, \dots, f_M\}$, that usually comes from the evaluation of a certain function $f : \Omega \rightarrow \mathbb{R}$. We also use a set of centers $\{\mathbf{c}_1, \dots, \mathbf{c}_N\} \subset \mathbb{R}^d$ and, for each of them, the translation of an appropriate radial function $\Phi(\cdot - \mathbf{c}_i)$, $i = 1, \dots, N$, as in (2).

The main goal is to approximate, in the best possible way, the points

$$\{(\mathbf{b}_1, f_1), \dots, (\mathbf{b}_M, f_M)\} \subset \mathbb{R}^{d+1}$$

by means of an adequate linear combination of radial basis functions (RBFs).

2.1. Spaces of Radial Functions

We define the radial basis functions that is used in the sequel. We consider the following function (for $\varepsilon \in \mathbb{R}^+$, $t \geq 0$):

$$\phi_\varepsilon(t) = -\frac{1}{2\varepsilon^3}(e^{-\varepsilon\sqrt{t}} + \varepsilon\sqrt{t}) \tag{1}$$

that is the fundamental solution, in the distribution sense, of the linear fourth-order differential equation

$$\frac{d^4\phi_\varepsilon}{dt^4} - \varepsilon^2 \frac{d^2\phi_\varepsilon}{dt^2} = \delta,$$

where δ is the Dirac's measure at the origin. The real parameter ε is called a tension parameter.

In our case, we use radial basis functions that are the translations of the radial function (now, for $\mathbf{x} \in \mathbb{R}^d$)

$$\Phi_\varepsilon(\mathbf{x}) = \phi_\varepsilon(\langle \mathbf{x} \rangle_d^2) = -\frac{1}{2\varepsilon^3}(e^{-\varepsilon\langle \mathbf{x} \rangle_d} + \varepsilon\langle \mathbf{x} \rangle_d) \tag{2}$$

where $\langle \cdot \rangle_d$ denote the Euclidean norm on \mathbb{R}^d .

This type of radial basis functions under tension (RBFTs) were introduced by A. Bouhamidi and A. Le Mehauté [26] in 2004. They depend on a positive parameter that provides a convenient way of controlling the behavior of the corresponding interpolation or approximation method. Moreover, error estimates of the corresponding interpolation method [26] are given in terms of Sobolev semi-norms, that also help in the convergence of the associated procedure and provide attractive geometrical tension effects.

2.2. Smoothing Radial Basis Approximations

Therefore, the idea is to use a finite-dimensional space H^N , whose basis is of the type

$$H^N \equiv span\left\{\Phi_\varepsilon(\cdot - \mathbf{c}_1), \dots, \Phi_\varepsilon(\cdot - \mathbf{c}_N), q_1(\cdot), \dots, q_{d(m)}(\cdot)\right\}.$$

In this case, the problem to be resolved can be formulated as follows: Given a set of real values and an approximation data set $\{(\mathbf{b}_i, f_i) : i = 1, \dots, M\}$, we still want to obtain $\sigma_N \in H^N$ such that

$$\sigma_N(\mathbf{b}_i) \approx f_i, \quad i = 1, \dots, M$$

by means of the following minimization problem.

Find $\sigma_N \in H^N$ such that

$$\mathcal{J}(\sigma_N) \leq \mathcal{J}(v), \quad \forall v \in H^N \tag{3}$$

where $\tau \in (0, \infty)$ and

$$\mathcal{J}(v) = \sum_{i=1}^M (f_i - v(\mathbf{b}_i))^2 + \tau |v|_2^2$$

where

$$|v|_2^2 = \sum_{|\alpha|=2} \int_{\Omega} (\partial^\alpha v(\mathbf{x}))^2 dx$$

with

$$|\alpha| = \alpha_1 + \dots + \alpha_d, \forall \alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d, \quad \partial^\alpha v(x_1, \dots, x_d) = \frac{\partial^{|\alpha|} v}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}.$$

Thus,

$$\sigma_N(\mathbf{x}) = \sum_{l=1}^{N+d(m)} \beta_l w_l(\mathbf{x}),$$

where

$$w_l(\cdot) = \left\{ \begin{array}{ll} \Phi_\epsilon(\cdot - \mathbf{c}_l), & l = 1, \dots, N \\ q_{l-N}(\cdot), & l = N + 1, \dots, N + d(m) \end{array} \right\};$$

and $\beta_1, \dots, \beta_{N+d(m)} \in \mathbb{R}$ are the control coefficients (i.e., the problem’s unknowns), obtained as the solution of the variational problem formulated in the following Theorem (see for example [27,28]).

Theorem 1. *The minimization problem (3) has a unique solution, that is also the only solution of the following variational problem.*

Find $\sigma_N \in H^N$ such that, for all $v \in H^N$,

$$\sum_{i=1}^M \sigma_N(\mathbf{b}_i) v(\mathbf{b}_i) + \tau(\sigma_N, v)_2 = \sum_{i=1}^M v(\mathbf{b}_i) f_i \tag{4}$$

where

$$(u, v)_2 = \sum_{|\alpha|=2} \int_{\Omega} (\partial^\alpha u(\mathbf{x}) \partial^\alpha v(\mathbf{x})) dx$$

Proof. The expression

$$[[v]] := \left(\sum_{i=1}^M v(\mathbf{b}_i)^2 + \tau (v, v)_2 \right)^{\frac{1}{2}}$$

constitutes a norm on H^N equivalent to its usual norm. Then, as a consequence, the continuous symmetric bilinear form

$$a : H^N \times H^N \longrightarrow \mathbb{R}$$

defined by

$$a(u, v) = \sum_{i=1}^M u(\mathbf{b}_i)v(\mathbf{b}_i) + \tau(u, v)_2$$

is H^N -elliptic. Besides, $\psi : H^N \rightarrow \mathbb{R}$, so

$$\psi(v) := \sum_{i=1}^M f_i v(\mathbf{b}_i)$$

is clearly a continuous linear form on this space.

Then, applying the Lax–Milgram lemma (see [29] for example), there exists a unique function $\sigma_N \in H^N$ such that

$$a(\sigma_N, v) = \psi(v), \quad \forall v \in H^N;$$

that is, (4) holds. Moreover, σ_N is the unique function of H^N that minimizes the functional

$$\tilde{\mathcal{J}}(v) = 2\left(\frac{1}{2}a(v, v) - \psi(v)\right),$$

concluding that σ_N also minimizes the functional \mathcal{J} . \square

By linearity, we can reduce the problem (4) to the following linear system:

$$(\mathbf{A}\mathbf{A}^\top + \tau\mathbf{R})\boldsymbol{\beta} = \mathbf{A}\mathbf{f}$$

where $\mathbf{A} = (w_i(\mathbf{b}_j))$, $i = 1, \dots, N + d(m)$, $j = 1, \dots, M$.

$$\mathbf{R} = \left(\sum_{|\alpha|=2} \int_{\Omega} \partial^\alpha w_i \partial^\alpha w_j dx \right)_{i,j=1,\dots,N+d(m)}$$

$$\boldsymbol{\beta} = (\beta_i)_{i=1,\dots,N+d(m)} \in \mathbb{R}^{N+d(m)}, \quad \mathbf{f} \equiv (f_i)_{i=1,\dots,M}$$

Remark 1. When $M = N$ and $\mathbf{b}_i \equiv \mathbf{c}_i$ for $i = 1, \dots, N$, we could also consider an interpolating problem at the centers, where the interpolating function $\hat{\sigma}_N$ has explicitly the following form:

$$\hat{\sigma}_N(\mathbf{x}) = \sum_{i=1}^N \lambda_i \Phi_\epsilon(\mathbf{x} - \mathbf{c}_i) + \sum_{j=1}^{d(m)} \beta_j q_j(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \tag{5}$$

where the coefficients $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_N)^\top$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_{d(m)})^\top$ are now the solution of the following linear system:

$$\begin{pmatrix} \mathbf{A}_\epsilon & \mathbf{M} \\ \mathbf{M}^\top & \mathbf{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \tag{6}$$

where $\mathbf{A}_\epsilon = (\Phi_\epsilon(\mathbf{c}_i - \mathbf{c}_j))_{1 \leq i, j \leq N}$ is an $N \times N$ matrix, $\mathbf{M} = (q_j(\mathbf{c}_i))_{\substack{1 \leq i \leq N \\ 1 \leq j \leq d(m)}}$ is an $N \times d(m)$ matrix, \mathbf{M}^\top denotes the transpose of \mathbf{M} , \mathbf{O} is the $d(m) \times d(m)$ zero matrix, $\mathbf{f} \equiv (f_1, \dots, f_N)^\top \in \mathbb{R}^N$ and $\mathbf{0}$ is the zero vector of $\mathbb{R}^{d(m)}$.

Take into account that we are considering the interpolating conditions

$$\hat{\sigma}_N(\mathbf{c}_i) = f_i, \quad i = 1, \dots, N;$$

together with the constraints (the usual orthogonality conditions to the space of polynomials $\Pi_{m-1}(\mathbb{R}^d)$ when we do not have the strict positiveness of the radial kernel function, to assure the $\Pi_{m-1}(\mathbb{R}^d)$ -unisolvency of the centers; see for example [1])

$$\sum_{i=1}^N \lambda_i q_j(\mathbf{c}_i) = 0, j = 1, \dots, d(m).$$

3. Methodology Overview

At this stage, it is important to define some ideas about multi-objective optimization problems and describe fundamental issues for NSGA-II-type algorithms (see [22]). The non-dominated sorting genetic algorithm is a multiple objective optimization algorithm, whose objective is to improve the adaptive fit of a population of candidate solutions to a Pareto front, constrained by a set of objective functions $G \equiv (g_1, \dots, g_l, \dots, g_L)$, each one defined in an appropriate domain.

Amongst the first algorithms to explicitly exert certain tendency towards the discovery of nondominated solutions, we could mention, for example, Fonseca and Fleming’s MOGA [30]; it uses fitness sharing amongst solutions of the same rank, coupled along a proportional adaptive selection to help promoting diversity.

The authors Srinivas and Deb’s [31] original nondominated sorting genetic algorithm (NSGA) works in a similar way, but assigns fitness based on dividing the population into a certain number of fronts of equal domination. To achieve this, the algorithm iteratively seeks all the nondominated points in the population that still have not been labeled as belonging to the current front and increments the front count, repeating until all solutions have been finally labeled. Therefore, each point in a given front obtains, as its raw score, the count of all solutions in inferior fronts. After this, Deb [22] and his coworkers, proposed a revised NSGA-II algorithm, which still uses the idea of non-dominated fronts, but also incorporates the following important changes:

- A crowding distance metric is defined for each point as the average side length of the cuboid defined by its nearest neighbors in the same front. The larger the value, the fewer solutions reside in the vicinity of the point.
- An appropriate survivor selection strategy, where the new population is obtained by accepting the individuals from progressively inferior fronts until it is full, so that, if not all the individuals in the last front considered can be accepted, they are chosen on the basis of their crowding distance.
- Parent selection uses a modified tournament operator that considers, first, the dominance rank, then crowding distance as the second criterium.

We can also see that, for the appropriate use of any NSGA-II procedure, it is necessary to describe the following fundamental issues:

- (1) Population initialization: the size of the population depends on the problem range.
- (2) Non-dominated sort. Before talking about this stage we define the following:
 - p and P , the individual and the main population, respectively.
 - S_p is the set of all individuals dominated by p .
 - n_p is the number of individuals that p dominates.
 - p_{rank} is the rank that an individual p has, depending on the front assigned to it.
 - Q is the set for storing temporarily the individuals in the $(i + 1)$ -th front.

(We recall that, in this context of multi-objective optimization, we say that p dominates q , or, equivalently, that q is dominated by p , if and only if p is no worse than q in all the objective functions, but it is strictly better in at least one of them.)

We describe the non-dominated procedure as follows:

- ★ For all individuals, $\forall p \in P$:
- ★ Initialize $S_p = \emptyset$ and $n_p = 0$
- ★ $\forall q \in P - \{p\}$
 - if p dominates q , then $S_p = S_p \cup \{q\}$
 - else, if q dominates p , then $n_p = n_p + 1$
- ★ if $n_p = 0$, then $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$ and $p_{rank} = 1$.

- ★ Begin initializing the new front, from the first one (with $i = 1$).
- ★ Next, for every front \mathcal{F}_i , while $\mathcal{F}_i \neq \emptyset$,

$$\mathcal{Q} = \emptyset$$

$$\forall p \in \mathcal{F}_i \text{ and } \forall q \in S_p$$

$$n_q = n_q - 1$$

$$\text{if } n_q = 0, \text{ then } \mathcal{Q} = \mathcal{Q} \cup \{q\} \text{ and } q_{rank} = i + 1$$

- ★ $i = i + 1$ (augment the front counter by one).
- ★ At this stage, the new front is $\mathcal{F}_i = \mathcal{Q}$.

(3) Crowding distance: It is very important to compare between individuals belonging to the same front \mathcal{F}_i , so that each individual has a crowding distance that estimates the density of solutions surrounding each member in this front. The method of calculating the crowding distance is summarized below.

For all considered fronts \mathcal{F}_i and every objective function $g_l, l = 1, \dots, L$, we perform the following:

- ★ We previously sort all of the members of the front \mathcal{F}_i , according to the l th objective function $sort(\mathcal{F}_i, l) \equiv \{p_{i,1}^{[l]}, \dots, p_{i,N_i}^{[l]}\}$, with $N_i \equiv card(\mathcal{F}_i)$; (where $g_l(p_{i,1}^{[l]}) \equiv g_l^{min}$ and $g_l(p_{i,N_i}^{[l]}) \equiv g_l^{max}$). Now, we use this arrangement to calculate the crowding distance for each individual in \mathcal{F}_i with the following passages.
- ★ First, initialize the distances of each member of the front to zero.
- ★ Namely, $\forall p_{i,j}^{[l]} \in \mathcal{F}_i$, with $j = 1, \dots, N_i$. Let us define and compute the following:
- ★ $d_{i,1}^{[l]} = d_{i,N_i}^{[l]} = \infty$ (to assure that these points are taken for next generations).
- ★ $d_{i,j}^{[l]} \equiv d_{p_{i,j}^{[l]}}^{[l]}(\mathcal{F}_i) = 0, \quad j = 2, \dots, N_i - 1$.
- ★ For $k = 2 : (N_i - 1)$,

$$d_{i,k}^{[l]} = d_{i,k}^{[l]} + \frac{g_l(p_{i,k+1}^{[l]}) - g_l(p_{i,k-1}^{[l]})}{g_l^{max} - g_l^{min}}$$

(4) Selection: In order to select individuals, we use a crowded comparison operator that depends on a binary tournament selection. The operation depends on \mathcal{F}_i and p_{rank} ; we say that $p \prec_c q$ if

$$p_{rank} < q_{rank}$$

$$\text{or if } p \text{ and } q \in \mathcal{F}_i$$

$$\text{then } d_p^{[l]}(\mathcal{F}_i) > d_q^{[l]}(\mathcal{F}_i)$$

(5) Genetic operator: Now, we describe the two parts of the simulated binary crossover and that of polynomial mutation, both performed componentwise:

- Binary crossover: The following expressions represent the simulated binary crossover (for each component k) of two contestants in the tournament, p_1 and p_2 :

$$c_{1,k} = \frac{1}{2}((1 - \beta_k)p_{1,k} + (1 + \beta_k)p_{2,k})$$

$$c_{2,k} = \frac{1}{2}((1 - \beta_k)p_{1,k} + (1 + \beta_k)p_{2,k})$$

with $\beta_k \geq 0$ and $c_{i,k}$ represents the k th component to the i th child. However, the density of any random sample is generated by

$$p(\beta) = \begin{cases} \frac{1}{2}(\eta_c + 1)\beta^{\eta_c}, & \text{if } 0 \leq \beta \leq 1 \\ \frac{1}{2}(\eta_c + 1)\frac{1}{\beta^{\eta_c+2}}, & \text{if } \beta > 1 \end{cases}$$

where η_c is defined as the distribution index of crossover. Let u be a uniform sample random number included in $(0, 1)$; then, we have

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta_c+1}}, & \text{if } u < 0.5 \\ \frac{1}{2(1-u)^{\frac{1}{\eta_c+1}}}, & \text{otherwise} \end{cases}$$

- Polynomial Mutation: First, we define some important notions, such as the following:
 - $c_{i,k}$ is a child component.
 - $p_{i,k}$ is a parent component.
 - r_k is a uniform sample random in $(0, 1)$.
 - δ_k is a mutation distribution index.
 - $[p_{i,k}^{lower}, p_{i,k}^{upper}]$ is the box constraints imposed over the k th-component.

Then, the polynomial mutation is defined componentwise by

$$c_{i,k} = p_{i,k} + (p_{i,k}^{upper} - p_{i,k}^{lower})\delta_k$$

where

$$\delta_k = \begin{cases} (2r_k)^{\frac{1}{\eta_l+1}} - 1, & \text{if } r_k < 0 \\ 1 - (2(1 - r_k))^{\frac{1}{\eta_l+1}}, & \text{if } r_k \geq 0 \end{cases}$$

- (6) Recombination and selection: The offspring population is combined with the current generation population and selection is performed to set the individuals of the next generation; the elitism is ensured, because the best individuals are added sequentially to the population. The selection of parents for the next generation is also based on their crowding distance, by selecting the individual at random, but taking into account their tournament score, then choosing the best individuals out of that set to be parents.
- (7) Selection of an appropriate region of interest (ROI), according with DM’s preferences: In this case, the preference of the user is always to maintain the number of centers as small as possible, if the degree of approximation remains within a preset range. In this manner, the proposed procedure attempts to find a set of solutions in the neighborhood of the corresponding Pareto optimal solution, so that the decision-maker can have a better idea of the desired region.

Different forms of fitness (or objective functions) can be used in a NSGA-II procedure, but the main goal, here, is to minimize some usual errors, such as E_C and/or E_L between the original function and the approximating or interpolating function, constructed from each population of random centers. At this step, it is also important to consider two error estimations that are appropriate normalizations of a discrete version of the usual norms in $\mathcal{C}(\Omega)$ and $\mathcal{L}^2(\Omega)$, respectively, and they are given by the following expressions:

$$E_C = \frac{\max_{i=1,\dots,K} |f(\mathbf{a}_i) - \sigma(\mathbf{a}_i)|}{\max_{i=1,\dots,K} |f(\mathbf{a}_i)|} \tag{7}$$

$$E_{\mathcal{L}} = \sqrt{\frac{\sum_{i=1}^K (f(\mathbf{a}_i) - \sigma(\mathbf{a}_i))^2}{\sum_{i=1}^K (f(\mathbf{a}_i))^2}} \tag{8}$$

where $f \in C^2(\Omega)$ is a given function, $\sigma \equiv \sigma_N$ is the approximating or $\sigma \equiv \hat{\sigma}_N$ interpolating RBF function associated with the given data set and $\{\mathbf{a}_1, \dots, \mathbf{a}_K\} \subset \Omega$ is another scattered random point test data set (TDS) where the errors are computed.

However, before running any MOGA algorithm, there are always some key parameters that should be defined. The most important ones are shown in Table 1 below.

Table 1. Parameters and functions used by MOGA in the simulations results.

Parameters of the MOGA	Value
Number of generations	20
Population size	40
Crossover function	Binary crossover
Selection function	Binary tournament
Crossover fraction	0.9
Pareto fraction	0.4
Mutation function	Polynomial mutation
Mutation rate	0.01
Centers' deletion tolerance	0.5×10^{-2}
Fitness' functions	Centers number and E_C or/and $E_{\mathcal{L}}$

Therefore, the most important goal of these methods will always be to try to improve the adaptive fit of a population of possible solutions to a Pareto front constrained by these often conflicting objectives. In this case, these objective functions are to minimize at least one or several normalized discrete versions of some approximate errors; however, the intention to maintain the number of centers as small as possible is also taken into account, so that we allow the procedure to delete some of the obtained centers, when they become sufficiently close to each other and the same level of approximation can be retained.

4. Simulation Examples

The objective of this study is to analyze the performance of this procedure for different types of functions, optimizing the centers placement using radial basis functions by the multi-objective genetic algorithms (MOGA) already used and explained in more detail in [32], but using cubic spline basis functions instead of RBFs. Different experiments were carried out and we show, in this section, the results for the approximation of several functions, also presenting the evolution of the optimal distribution of points, together with the related Pareto fronts. Similar results could be obtained for the case of interpolating RBFs, as explained in the Remark 1.

In order to analyze the behavior of the radial basis approximations, a TDS with a large (100) number of points was used, with a population number of 40 individuals, until 20 generations, in each one of the examples below. Concerning the shape parameter, it is also true that it can affect the approximation accuracy of the RBF and the overall computational stability of the procedure (see for example [33,34]), but it is not the goal of our study for the moment; therefore, a fixed value of $\varepsilon = 0.1$ was used without problems in all our computations. The same applies to the other smoothness parameter of the RBFs approximating procedure, τ , where a constant fixed value of 1 was used in all our numerical experiments.

One-Dimensional Examples

Example 1. $f_1 : [0, \pi] \rightarrow \mathbb{R}$

$$f_1(x) = 0.12 + 0.25 \exp^{-4(x-\frac{\pi}{4})^2} \cos(2x) \sin(2\pi x) .$$

To test the behavior of the approximation for the presented methodology, performed by optimization of the centers' placement of radial basis functions by a NSGA-II algorithm, Figure 1 shows the results of the approximation of function $f_1(x)$, using radial basis function approximations, while, in Figure 2, we can see how the $E_{\mathcal{L}}$ error decreased clearly when the number of centers increased. We can also see, in the left column of Figure 1, the radial basis approximation corresponding to the last centers' distribution, whose evolution is also showed in the right column, with the increase in the number of interior centers for the function $f_1(x)$ with 7, 9 and 11 centers, respectively.

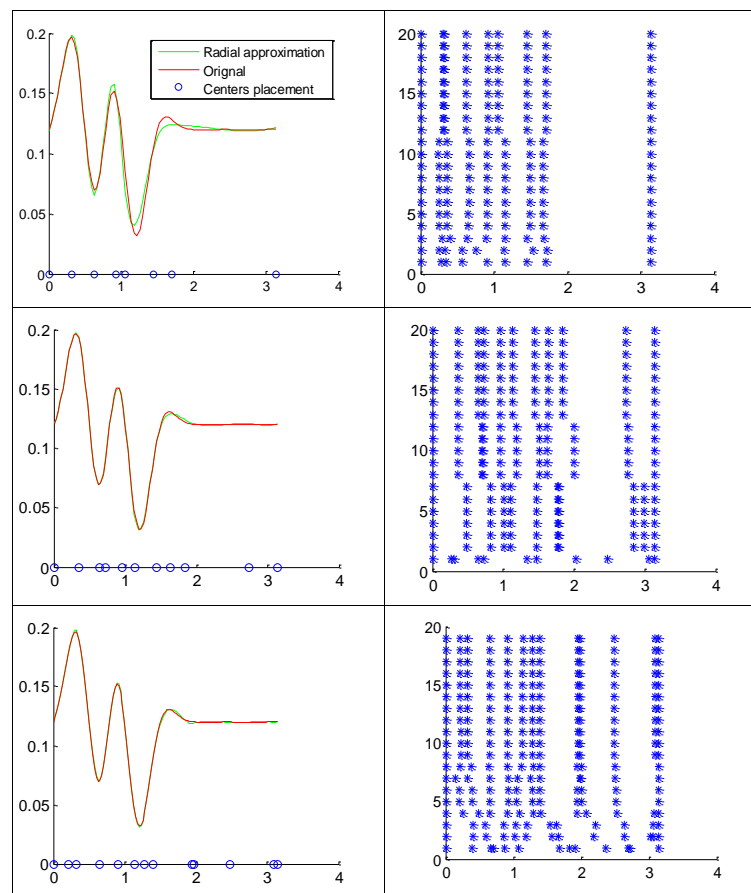


Figure 1. Radial basis approximation and evolution of final centers' placement for the function $f_1(x)$ with 7, 9 and 11 centers, respectively.

Example 2. $f_2 : [0, 1] \rightarrow \mathbb{R}$

$$f_2(x) = 0.5e^{-5(x-\pi/2)^2} \sin(4\pi x) \cos(4x) .$$

For Example 2, the numerical approximation results for $f_2(x)$ and the evaluation of centers are shown in Figure 3; on the other hand, the two normalized discretised errors for $f_2(x)$ are also described in Table 2.

We can see, in the left column of Figure 3, the radial basis approximation corresponding to the last centers' distribution, whose evolution is also showed at the right column, with the increase in the number of interior centers for the function $f_2(x)$, using 2, 4 and 6 interior centers, respectively.

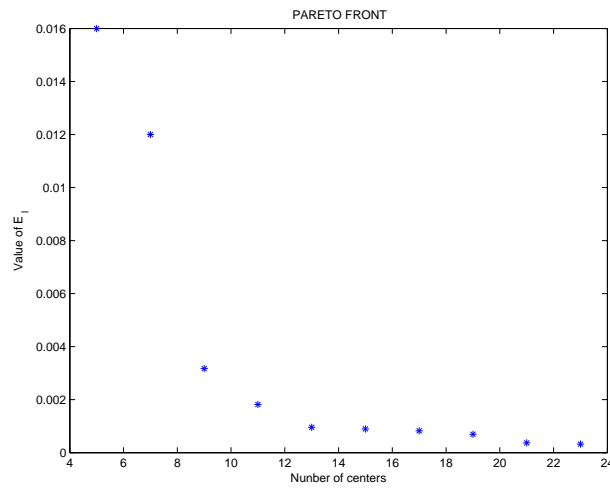


Figure 2. Pareto front of $E_{\mathcal{L}}$ error vs. number of centers for $f_1(x)$.

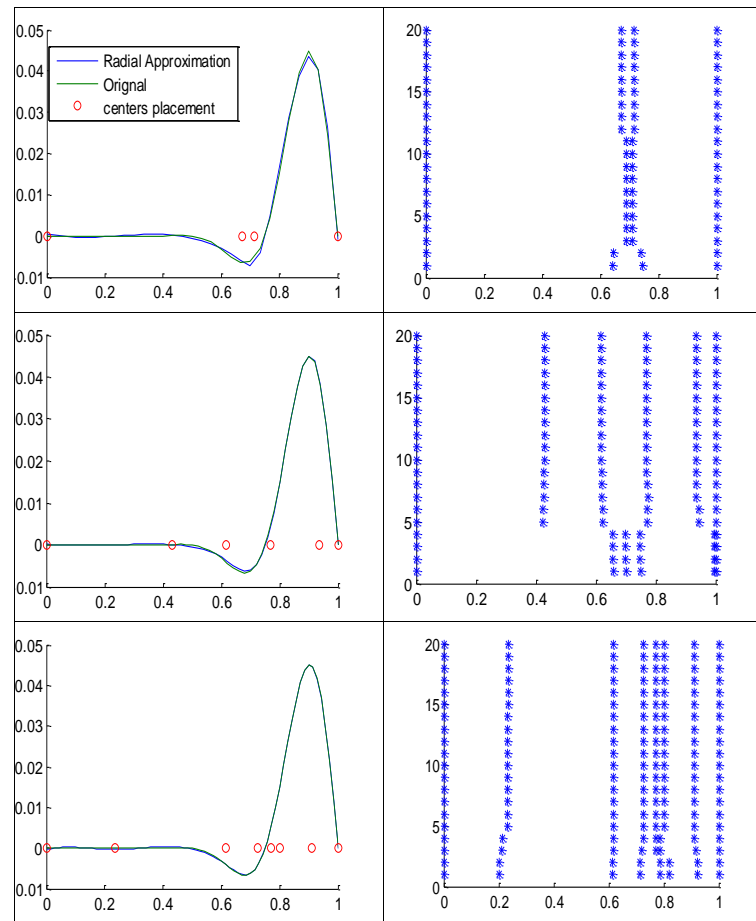


Figure 3. Radial basis approximation and evolution of final centers' placement for the function $f_2(x)$ with 2, 4 and 6 interior centers, respectively.

Table 2. Two approximation discretization errors for $f_2(x)$.

Centers	$E_{\mathcal{L}}$	$E_{\mathcal{C}}$
2	4.3978×10^{-2}	1.8918×10^{-3}
4	2.6664×10^{-2}	6.7993×10^{-4}
6	4.0691×10^{-3}	1.9208×10^{-4}
8	3.9539×10^{-3}	1.7241×10^{-4}

Example 3. $f_3 : [-1, 1] \rightarrow \mathbb{R}$

$$f_3(x) = 1 - e^{-50|x|} .$$

The graphical result approximating $f_3(x)$ with 20 interior centers, using 20 generations in the NSGA-II algorithm, using radial basis functions, is showed in Figure 4.

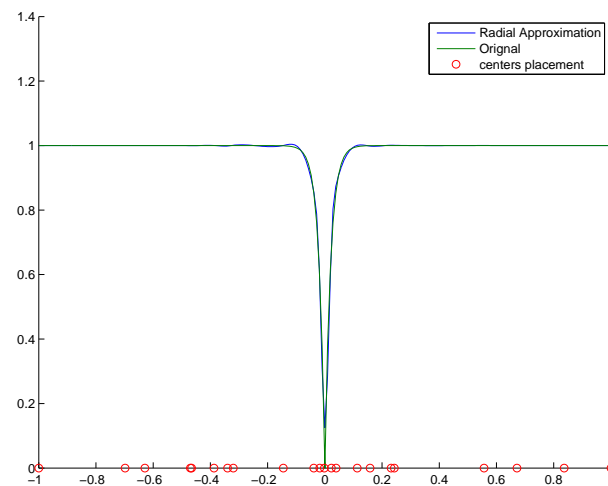


Figure 4. Graphical results for $f_3(x)$ with 20 interior centers (marked on the x -axis).

5. Conclusions and Future Work

Although there are some other approaches to select knots amongst a uniform partition of the interval (or domain) of definition of the function to be approximated, or interpolated, they are usually based on splines and the evolutionary procedure used to improve the final allocation of these knots usually relies on basic genetic algorithms, not very performing for general multi-objective problems, or even on self-adjusting or hybrid neural networks, with the requirement of the previous hard training process, etc. This paper presents another important and interesting possibility for optimizing both the number and the corresponding centers' placement in the problem of radial basis function approximation of functions of one or several independent variables, where we can clearly see the effectiveness of the strategy for different types of functions. We chose a radial basis function under tension (RBFT) depending on a positive parameter, that also provides a convenient way to control the behavior of the corresponding interpolation or approximation method. In addition, the preliminary numerical results of this MOGA strategy, for centers' placement in approximating functions of one variable, show that the placement of the centers in radial basis approximation function also has an important and significant impact on the behavior of the final results, because the ideal place or location of the centers is not known in advance. However, one important feature of this approach is that completely the same theoretical procedure can be developed for problems in more spatial dimensions, for the well-known meshless character of these type of functions. It is also well known that increasing the number of centers for the radial basis in the RBF approximation usually also increases the accuracy of the approximation, but only up to a certain level, where the computational effort is not worthy enough; therefore, in our procedure, we can admit the coalescence of very close centers, maintaining the same level of accuracy. In addition, in the interpolating case, this possible, excessive proximity of the centers has to be avoided, so as not to have excessive matrix conditioning problems with the linear systems to be solved. On the other hand, in classic allocation methods, the designer should choose a priori the number of centers that will be used and, many times, they are just equidistant uniform partitions of the domain. However, using such a MOGA procedure of the type considered, they can move freely inside the domain and the corresponding Pareto fronts can also be directly improved using a different or variable number of centers, at the same time as simulations are being performed.

An open problem would be the optimal election of both the shape parameter ε and the smoothness parameter τ , since they can depend not only on the number and distribution of data points and the data vector, but even on the precision of the computations, even though it is well known that very small values give, in general, highly accurate results. Then, the idea would be to introduce, in a future work, these parameters also as possible optimization variables in the general MOGA algorithm. In addition, the possibility of using the basis of RBFs of compact support, such as the well-known Wendland's RBFs, would be a good alternative to the RBFs used in this work.

Author Contributions: Investigation, P.G.-R., H.M.H.I., M.Y. and M.P. All authors contributed equally to this work and have read and agreed to the published version of the manuscript.

Funding: This work was supported by FEDER/Junta de Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades (Research Project A-FQM-76-UGR20, University of Granada) and by the Junta de Andalucía (Research Group FQM191).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We acknowledge the anonymous referees for their useful comments and suggestions, as well as the Department of Applied Mathematics of the University of Granada and the Mathematics Journal Editorial Board, for the financial aid offered for the final cost of the APC.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DM	decision maker
MATLAB	MAtrix LABoratory (by MathWorks)
MOEA	multi-objective evolutionary algorithm
MOGA	multi-objective genetic algorithm
NSGA-II	non-dominated sorting genetic algorithm, second version
RBF	radial basis function
RBFNN	radial basis function neural network
ROI	region of interest
SOP	single objective problem
TDS	test data set

References

1. Buhmann, M.D. Radial Basis Functions: Theory and Implementations. In *The Cambridge Monographs on Applied and Computational Mathematics*; Cambridge University Press: Cambridge, UK, 2003; Volume 12.
2. Powell, M.J.D. The Theory of Radial Basis Functions Approximation in 1990. In *Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions*; Light, W.A., Eds.; Clarendon Press: Oxford, UK, 1992; pp. 105–210.
3. Hoschek, J.; Lasser, D. *Fundamentals of Computer Aided Geometric Design*; A K Peters, Ltd.: Natick, MA, USA, 1993.
4. Buhmann, M.; Dyn, N. Spectral convergence of multiquadric interpolation. *Proc. Edinburgh Math. Soc.* **1993**, *36*, 319–333. [[CrossRef](#)]
5. Cheng, A.H.-D.; Golberg, M.A.; Kansa, E.J.; Zammito G. Exponential convergence and $H - c$ multiquadric collocation method for partial differential equations. *Numer. Methods Partial Diff. Equ.* **2003**, *19*, 571–594. [[CrossRef](#)]
6. Huang, M.C.; Tai, C.C. The pre-processing of data points for curve fitting in reverse engineering. *Int. J. Adv. Manuf. Technol.* **2000**, *16*, 635–642. [[CrossRef](#)]
7. Lai, Y.-K.; Hu, S.-M.; Pottmann, H. Surface fitting based on a feature sensitive parametrization. *Comput. Aided Des.* **2006**, *38*, 800–807. [[CrossRef](#)]
8. Jeffreys, H.; Jeffreys, B.S. Lagrange's Interpolation Formula. In *Methods of Mathematical Physics*, 3rd ed.; Cambridge University Press: Cambridge, UK, 1988; Volume 59.
9. Whittaker, E.T.; Robinson, G. Lagrange's formula of interpolation. In *The Calculus of Observations: A Treatise on Numerical Mathematics*, 4th ed.; Dover: New York, NY, USA, 1967; pp. 28–30.
10. Szego, G.; Sakai, M. *Orthogonal Polynomials*; American Mathematical Society: Providence, RI, USA, 1975.

11. Esmailbeigi M.; Hosseini M.M. Dynamic node adaptive strategy for nearly singular problems on large domains. *Eng. Anal. Bound. Elem.* **2012**, *36*, 1311–1321. [[CrossRef](#)]
12. Kozera, R.; Noakes, L. Optimal Knots Selection for Sparse Reduced Data. *Appl. Math. Inform.* **2016**, *LNCS 9555*, 3–14.
13. Krityakierne, T.; Akhtar, T.; Shoemaker, C.A. SOP: parallel surrogate global optimization with Pareto-center selection for computationally expensive single-objective problems. *J. Glob. Optim.* **2016**, *66*, 417–437. [[CrossRef](#)]
14. García-Capulin, C.H.; Cuevas, F.J.; Trejo-Caballero, G.; Rostro-González, H. A hierarchical genetic algorithm approach for curve fitting with B-splines. *Genet. Program. Evolvable Mach.* **2015**, *16*, 151–166. [[CrossRef](#)]
15. Leitenstorfer, F.; Tutz, G. Knot Selection by Boosting Techniques. *Comput. Stat. Data Anal.* **2007**, *51*, 4605–4621. [[CrossRef](#)]
16. Samozino, M.; Alexa, M.; Alliez, P.; Yvinec, M. Reconstruction with Voronoi Centered Radial Basis Functions. In Proceedings of the SGP'06: Fourth Eurographics Symposium on Geometry Processing, Cagliari-Sardinia, Italy, 26–28 June 2006; Polthier, K., Sheffer, A., Eds.; Eurographics Association: Goslar, Germany, 2006; pp. 51–60.
17. Afiahdoust, F.; Esmailbeigi, M. Optimal variable shape parameters using genetic algorithm for radial basis function approximation. *Ain Shams Eng. J.* **2015**, *6*, 639–647. [[CrossRef](#)]
18. Zheng, S.; Feng, R.; Huang, A. The Optimal Shape Parameter for the Least Squares Approximation Based on the Radial Basis Function. *Mathematics* **2020**, *8*, 1923. [[CrossRef](#)]
19. Awad, M.; Pomares, H.; Herrera, L.J.; González, J.; Guillén, A.; Rojas F. Approximating I/O Data Using Radial Basis Functions: A New Clustering-Based Approach. In *Computational Intelligence and BioInspired Systems, IWANN 2005, LNCS, 3512*; Cabestany, J., Prieto, A., Sandoval, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2005.
20. Rajchakit, G.; Agarwal, P.; Ramalingam, S. *Stability Analysis of Neural Networks*, 1st ed.; Springer: Singapore, 2021.
21. Bechikh, S.; Kessentini, M.; Said, L.B.; Ghédira, K. Chapter Four—Preference Incorporation in Evolutionary Multiobjective Optimization: A Survey of the State-of-the-Art. *Adv. Comput.* **2015**, *98*, 141–207.
22. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
23. Heris, M.K. NSGA-II in MATLAB, 2015. Available online: <https://yarpiz.com/56/ypea120-nsga2> (accessed on 1 May 2017).
24. Brownlee, J. *Clever Algorithms: Nature-Inspired Programming Recipes*. Available online: <http://www.cleveralgorithms.com> (accessed on 15 September 2020).
25. Scattered Data Interpolation and Approximation Using Radial Base Functions. Available online: <https://es.mathworks.com/matlabcentral/fileexchange/10056-scattered-data-interpolation-and-approximation-using-radial-base-functions> (accessed on 1 June 2020).
26. Bouhamidi, A.; Le Mehauté, A. Radial Basis Functions Under Tension. *J. Approx. Theory* **2004**, *127*, 135–154. [[CrossRef](#)]
27. Kouibia, A.; Pasadas, M. Approximation of surfaces by fairness bicubic splines. *Adv. Comput. Math.* **2004**, *20*, 87–103. [[CrossRef](#)]
28. Kouibia, A.; Pasadas, M. Approximation of curves by fairness cubic splines. *Appl. Math. Sci.* **2007**, *1*, 227–240.
29. Atkinson, K.; Han, W. *Theoretical Numerical Analysis*, 2nd ed.; Springer: New York, NY, USA, 2005.
30. Fonseca, C.M.; Fleming, P.J. An overview of evolutionary algorithms in multiobjective optimization. *Evol. Comput.* **1995**, *3*, 1–16.
31. Srinivas, N.; Deb, K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [[CrossRef](#)]
32. Idais, H.; Yasin, M.; Pasadas, M.; González, P. Optimal knots allocation in the cubic and bicubic spline interpolation problems. *Math. Comput. Simul.* **2019**, *164*, 131–145. [[CrossRef](#)]
33. Sarra, S.A.; Sturgill, D. A random variable shape parameter strategy for radial basis function approximation methods. *Eng. Anal. Bound. Elem.* **2009**, *33*, 1239–1245. [[CrossRef](#)]
34. Skala, V.; Karim, S.; Zabran, M. Radial Basis Function Approximation Optimal Shape Parameters Estimation: Preliminary Experimental Results. In *Computational Science-ICCS 2020, Part VI, LNCS 12142*; Springer: Cham, Switzerland, 2020; pp. 309–317.