Elsevier Editorial System(tm) for
Mathematics and Computers in Simulation
Manuscript Draft

Manuscript Number: MATCOM-D-18-00099R1

Title: Optimal knots allocation in the cubic and bicubic spline interpolation problems

Article Type: SI: MAMERN VII-2017

Keywords: Interpolation
Knots allocation
Cubic/bicubic B-splines

Corresponding Author: Dr. Pedro Gonzalez-Rodelas, Ph. D.

Corresponding Author's Institution: University of Granada

First Author: Pedro Gonzalez-Rodelas, Ph. D.

Order of Authors: Pedro Gonzalez-Rodelas, Ph. D.; Hasan Idais, Ph. D. student; Mohammed Yasin, Ph. D. student; Miguel Pasadas, Ph. D.

Abstract: Interpolation, together with approximation, are two major and ubiquitous problems in Mathematics, but also in almost every scientific field. Another interesting question is the optimal knots placement when interpolating, or approximating, certain functions using splines. In this work, a powerful methodology is presented for optimal knots placement when interpolating a curve, or a surface, using cubic or bicubic splines, respectively. For this, a Multi-Objective-Genetic Algorithm (MOGA) has been developed, in a way that ensures avoiding the large number of local minima existing in the problem of random knots placement. A new technique is presented to optimize both the number of knots and its optimal placement for cubic or bi-cubic interpolating splines. The performance of the proposed methodology has been evaluated using functions of one and two variables, respectively.

Optimal knots allocation in the cubic and bicubic spline interpolation problems

H. Idais[a], M. Yasin[a], M. Pasadas[a], P. González[a,*]

[a] *Dpt. of Applied Mathematics. University of Granada, 18071, Granada, Spain*

Abstract

Interpolation, together with approximation, are two major and ubiquitous problems in Mathematics, but also in almost every scientific field. Another interesting question is the optimal knots placement when interpolating or approximating certain functions using splines. In this work, a powerful methodology is presented for optimal knots placement when interpolating a curve, or a surface, using cubic or bicubic splines, respectively. For this, a Multi-Objective-Genetic Algorithm (MOGA) has been developed, in a way that ensures avoiding the large number of local minima existing in the problem of random knots placement. A new technique is presented to optimize both the number of knots and its optimal placement for cubic or bicubic interpolating splines. The performance of the proposed methodology has been evaluated using functions of one and two variables, respectively.

Keywords: Interpolation, Knots allocation, Cubic/bicubic B-splines
2010 MSC: 41A15, 65D05, 65D15, 65D17

1. Introduction

Interpolation and approximation are very important problems, in Mathematics and also in many applied fields. For example, they are the key technology in every reverse engineering procedure, systematic function or data analysis, sig-

*Corresponding author
Email address: {hasan.correo, yaseen.correo, mpasadas, prodelas}@ugr.es (P. González)

nal processing, data representation, stockage and/or compression, etc. In our case, the main aim is to obtain a structured and suitable digital representation of curves and surfaces, related to some environmental and/or natural resources practical problem, by using cubic or bicubic splines, respectively.

There exist several methodologies for approximating or interpolating a surface or some given data. In any case, multiple authors, like those in [1, 2, 3] describe the important and significant effect of the placement of the knots in spline approximation or interpolation on the performance of the final results.

Many methodologies have been presented for the selection and optimization of parameters within B-splines, using techniques based on selecting some special knots, called dominant points [4, 5], or by a data selection process as in [6]. Also the methodology in [7] gave a technique for automatic knots modification using an elitist clonal selection algorithm. Other procedures using some least squares methods use uniform knots distributions, in connection with a possible sensitive parametrization [1]. In this sense it is worth to take into account the works [2] and [3].

In [8] the authors present a method for selecting the unknown knots by minimizing a certain cost function, and this is one of the key ingredients in the implementation of the so-called Genetic Algorithms (GA). Other methodologies, as for example that in [9], describe techniques for using a computational Artificial Immune System (AIS), based on a hierarchical paradigm structure to determine the number and knots location automatically. AIS are adaptive computational algorithms inspired by theories and models of real biological inmune systems when applied to mathematical or enginering problem solving. Some key aspects on the designing of AIS are:

- The chosen internal representation
- Mechanisms to evaluate the immune interactions
- The adaptation processes and mechanisms

Following the ideas of Man et al. in [10] also the computational representation of the chromosomes in a GA can also be arranged in more complex hierarchical structure, to emulate the formation of the biological DNA in living beings. In this way, the computational chromosomes of these Hierarchical Genetic Algorithms (HGAs) consists of two types of genes, known as *control* and *parametric* genes, that typically are encoded as binary digits and real numbers (or any other particular adequate representation), respectively. Both types of genes can be simultaneously optimized using this HGAs but in general are much more complicated to be implemented.

The authors in [11] pointed out a method for curve and surface fitting by implicit polynomials. In [12] an algorithm is presented to approximate scattered data with a B-spline surface. But there are many other related works, for instance the method in [13] optimizes the location and the number of knots in curve fitting with splines by a sparse optimization model; and in [14], a fast and computationally efficient methodology for optimal placement of the knots by using simulated annealing (SA) is also presented. In this stochastic procedure, a combination of hill-climbing and random walks is followed in order to simulate a know physical process in tempered metallurgy. So, SA is a descent algorithm modified by random ascent moves in order to escape local minima which are not global ones. Other more or less “intelligent” Teaching and Learning Based Optimization (TLBO) [15] and self-adaptive multi-population based Jaya algorithms [16] for Engineering problems have also been developed more recently, but we do not find them specially well-suited for our particular problem, taking also into account the more specialized implementation and applications that inspired them.

In this paper, a new multi-objective NSGA-II methodology for optimal placement of random knots, when interpolating a function of one or two variables, using cubic or bicubic splines, is developed. But, we have to emphasize that, although some of the previous more basic evolutionary algorithms presented have been applied in the existing literature to the knots placement in the fitting or approximating 1D problem, we have not found references where the interpolating case is also treated, and neither the 2D approximating or interpolating cases.

We also have to mention that in most of the cases, the GAs encountered in the literature were applied to a certain combinatorial binary problem of selecting the knots among a particular uniform partition of the interval, whereas we are aplying a more sophisticated NSGA-II MOGA with real representation of the chromosomes, in such a way that these interpolation knots/points can freely move, or even be deleted when necessary, until a certain threshold of errors and number of knots is accomplished. So, this is the main originality of the present work, together with the ability to treat with functions of two variables as well.

2. Background material

2.1. Cubic interpolation splines in one variable

In this subsection, just to fix the notation used in the sequel of the article, the usual cubic splines will be presented for this interpolation problem. In principle, we define the space of cubic splines of class $\mathcal{C}^2$ on the partition of $[a, b]$ in n subintervals, from an increasing sequence of (uniform or non uniform) points or knots $\triangle_n = \{a = x_0 < ... < x_n = b\}$, and we define the cubic spline of class $\mathcal{C}^2$ on the partition $\triangle_n$ as every function $s : [a, b] \longrightarrow \mathbb{R}$ such that

i) $s \in \mathcal{C}^2[a, b]$

ii) $s\mid_{[x_i,x_{i+1}]} \in \mathbb{P}_3([x_i, x_{i+1}]), i = 0, \ldots, n-1$

where $\mathbb{P}_3([x_i, x_{i+1}]),\ i = 0, \ldots, n-1$, is the space of all restrictions of polynomial functions of degree less than or equal to three in the interval $[x_i, x_{i+1}]$.

If now we have:

$x_{-3} \leq x_{-2} \leq x_{-1} \leq x_0 = a < \cdots < x_n = b \leq x_{n+1} \leq x_{n+2} \leq x_{n+3}$,

we can define for each $x \in [a, b]$

$$B_i^0(x) = \begin{cases} 1, & if \quad x_{i-3} \leq x < x_{i-2}, \\ 0, & otherwise \end{cases} , \ for \quad i = 0, \ldots, n+5$$

and $B_i^k(x),\ \ k = 1, 2, 3$, from the recursive relation:

$$B_i^k(x) = \frac{x - x_{i-3}}{x_{i+k-3} - x_{i-3}} B_i^{k-1}(x) + \frac{x_{i+k-2} - x}{x_{i+k-2} - x_{i-2}} B_{i+1}^{k-1}(x),\ i = 0, \ldots, n+5-k.$$

These functions verify the following properties:

i) They are positive in the interior of their support,

$$B_i^k(x) \geq 0, \ \forall x \in [a,b].$$

ii) They form a partition of unity,

$$\sum_{i=0}^{n+5-k} B_i^k(x) = 1, \ \forall x \in [a,b].$$

iii) $\left\{B_0^k, \cdots, B_{n+5-k}^k\right\}$ are linearly independent for all $k = 0,1,2,3$.

Besides, if $\mathcal{S}_3(\triangle_n)$ represents the set of cubic spline functions of degree less than or equal to three and class $\mathcal{C}^2$, then $dim\,\mathcal{S}_3(\triangle_n) = n+3$ and a basis of $\mathcal{S}_3(\triangle_n)$ would be $\left\{B_0^3, \cdots, B_{n+2}^3\right\}$, called B-spline basis functions of fourth order, and degree three.

Now, given $u_i \in \mathbb{R}$ (for $i = 0, \ldots, n$) and $\{a = x_0 < ... < x_n = b\}$, we want to obtain the usual natural cubic spline (with vanishing second derivatives at the endpoints of the interval $[a,b]$) interpolating these values at these specified knots, but using now these B-spline basis functions of $\mathcal{S}_3(\triangle_n)$, $\left\{{B_i}^3\right\}_{i=0,\ldots,n+2}$ instead of the usual piecewise representation on each one of the subintervals. Thus $s : [a,b] \to \mathbb{R}$ such that $s(x_i) = u_i, i = 0, \ldots, n$ can be expressed as

$$s(x) = \sum_{i=0}^{n+2} \alpha_i B_i^3(x), \ x \in [a,b], \tag{1}$$

where $\alpha \equiv (\alpha_0, \ldots, \alpha_{n+2})$ is the solution of the linear system $A\alpha = B$ obtained under the following conditions

1. $s(x_i) = u_i, \quad i = 0, ..., n,$
2. $s''(x_0) = 0 = s''(x_n);$

where the matrix $A = \begin{pmatrix} A_1 \\ \hline A_2 \end{pmatrix} \in \mathbb{R}^{(n+3)\times(n+3)}$ and the vector $B = \begin{pmatrix} b_1 \\ \hline b_2 \end{pmatrix} \in \mathbb{R}^{n+3}$ are composed of two submatrices and subvectors, respectively:

$$A_1 = ({B_j}^3(x_i))_{\substack{0\leq i\leq n \\ 0\leq j\leq (n+2)}} \in \mathbb{R}^{(n+1)\times(n+3)}$$

$$A_2 = \begin{pmatrix} (B_j^3)''(x_0) \\ (B_j^3)''(x_n) \end{pmatrix}_{0\le j\le n+2} \in \mathbb{R}^{2\times(n+3)}$$

$$b_1 = (u_i)_{0\le i\le n} \in \mathbb{R}^{n+1}, \quad b_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \mathbb{R}^2.$$

2.2. Bicubic interpolating splines of class $\mathcal{C}^2$.

We will also extend the classical cubic univariate B-splines to bicubic B-splines in two variables. Let $R = (a,b)\times(c,d) \subset \mathbb{R}^2$ be a rectangular open set. We start with two partitions with knot sequences $\triangle_n$ of $[a,b]$ in n subintervals, and $\triangle_m$ of $[c,d]$ in m subintervals: $\triangle_n = \{a = x_0 < x_1 < \ldots < x_n = b\}$ and $\triangle_m = \{c = y_0 < y_1 < \ldots < y_m = d\}$; then, $\triangle_n \times \triangle_m$ is a grid partition of R, and we define the bicubic spline of class $\mathcal{C}^2$ on the partition $\triangle_n \times \triangle_m$ as every function $S : R \longrightarrow \mathbb{R}$ such that

i) $S \in \mathcal{C}^2(R)$

ii) $S\,|_{[x_i,x_{i+1}]\times[y_j,y_{j+1}]} \in \mathbb{P}_3([x_i,x_{i+1}]\times[y_j,y_{j+1}])$, for every $i = 0,\ldots,n-1, j = 0,\ldots,m-1$

where $\mathbb{P}_3([x_i,x_{i+1}]\times[y_j,y_{j+1}])$, $i = 0,\ldots,n-1, j = 0,\ldots,m-1$, is the space of all restrictions of polynomial functions of two variables and partial degree less than or equal to three to the sub-rectangle $[x_i,x_{i+1}]\times[y_j,y_{j+1}]$. Given $x_{-3}, x_{-2}, x_{-1}, x_{n+1}, x_{n+2}, x_{n+3} \in \mathbb{R}$ and $y_{-3}, y_{-2}, y_{-1}, y_{m+1}, y_{m+2}, y_{m+3} \in \mathbb{R}$, such that, $x_{-3} \le x_{-2} \le x_{-1} \le x_0 < \cdots < x_n \le x_{n+1} \le x_{n+2} \le x_{n+3}$, and $y_{-3} \le y_{-2} \le y_{-1} \le y_0 < \cdots < y_m \le y_{m+1} \le y_{m+2} \le y_{m+3}$, we can construct, as presented in the previous subsection, the corresponding cubic splines of class $\mathcal{C}^2$ in each one of the variables, x and y, on the partitions $\triangle_n$ and $\triangle_m$ respectively.

Meanwhile, if $\mathcal{S}_3(\triangle_n \times \triangle_m)$ represents the set of bicubic spline functions of degree less than or equal to three and class $\mathcal{C}^2$ in both variables, then $dim\,\mathcal{S}_3(\triangle_n\times\triangle_m) = (n+3)\times(m+3)$ and if the basis of $\mathcal{S}_3(\triangle_n)$ and $\mathcal{S}_3(\triangle_m)$ are denoted by $\{B_0^3(x), ..., B_{n+2}^3(x)\}$ and $\{B_0^3(y), ..., B_{m+2}^3(y)\}$, respectively, then a basis of $\mathcal{S}_3(\triangle_n \times \triangle_m)$ will be

$$\left\{ B_q^3(x,y) \equiv B_i^3(x) B_j^3(y), \substack{i=0,\dots,n+2,\\ j=0,\dots,m+2,\\ q=(n+3)j+i+1} \right\}.$$

Now, given a function f, well defined and smooth enough (at least of class $\mathcal{C}^1$); in this section we deal with the following problem: Find another smooth function (mainly a bivariate polynomial or spline) $S : R \longrightarrow \mathbb{R}$ to interpolate the following $N \equiv (n+1) \times (m+1)$ points in $3D$:

$$\{(x_i, y_j, u_l) : i = 0, ..., n, j = 0, ..., m, l = (n+1)j + i + 1\}. \tag{2}$$

where we will also denote $U \equiv \{u_l = f(x_i, y_j)\}_{l=1,\dots,N} \subset \mathbb{R}$

In fact here, we want to obtain $S \in \mathcal{S}_3(\triangle_n \times \triangle_m)$ such that, we can obtain for every $i = 0, \dots, n-1$, $j = 0, \dots, m-1$

$$S(x_i, y_j) = u_l, \; l = (n+1)j + i + 1.$$

Thus, we would write

$$S(x,y) = \sum_{k=1}^{(n+3)(m+3)} \alpha_k B_k(x,y), \; \forall (x,y) \in R, \tag{3}$$

where $\alpha \equiv (\alpha_1, \dots, \alpha_{(n+3)(m+3)})$ is the solution of the linear system $A\alpha = B$ obtained under the following conditions

1) $S(x_i, y_j) = (u_l), \; i = 0, ..., n, \; j = 0, ...m, \; l = (n+1)j + i + 1,$

2) $\dfrac{\partial^2 S}{\partial y^2}(x_i, c) = 0 = \dfrac{\partial^2 S}{\partial y^2}(x_i, d), \;\; i = 0, ..., n,$

3) $\dfrac{\partial^2 S}{\partial x^2}(a, y_j) = 0 = \dfrac{\partial^2 S}{\partial x^2}(b, y_j), \;\; j = 0, ..., m,$

4) $\dfrac{\partial^4 S}{\partial x^2 \partial y^2}(a,c) = \dfrac{\partial^4 S}{\partial x^2 \partial y^2}(a,d) = \dfrac{\partial^4 S}{\partial x^2 \partial y^2}(b,c) = \dfrac{\partial^4 S}{\partial x^2 \partial y^2}(b,d) = 0,$

Where $A = (\dfrac{A_1}{A_2})$,
$A_1 = (B_k(d_l)), \;\; l = 1, ..., (n+1)(m+1), \;\; k = 1, ..., (n+3)(m+3)$, and
$d_l = (x_i, y_j), \;\; i = 0, ..., n, \;\; j = 0, ..., m, \;\; l = (n+1)j + i + 1,$

$$(A_2)_{\hat{l}k} = \begin{cases} \dfrac{\partial^2}{\partial y^2} B_k(x_i, c) \\ \dfrac{\partial^2}{\partial y^2} B_k(x_i, d) \\ \dfrac{\partial^2}{\partial x^2} B_k(a, y_j) \\ \dfrac{\partial^2}{\partial x^2} B_k(b, y_j) \\ \dfrac{\partial^4}{\partial x^2 \partial y^2} B_k(a, c) \\ \dfrac{\partial^4}{\partial x^2 \partial y^2} B_k(a, d) \\ \dfrac{\partial^4}{\partial x^2 \partial y^2} B_k(b, c) \\ \dfrac{\partial^4}{\partial x^2 \partial y^2} B_k(b, d) \end{cases} \quad i = 0, \cdots, n, \; j = 0, \cdots, m,$$

where $\hat{l} = 1, ..., 2(n+1)+2(m+1)+4$, $k = 1, ..., (n+3)(m+3)$, and $B = (\dfrac{b_1}{b_2})$, $b_1 = (u_l)$, $l = 1, ..., (n+1)(m+1)$, and $b_2 = (0) \in \mathbb{R}^K$, with $K = (n+3)(m+3) - (n+1)(m+1)$.

2.3. Optimization paradigm for placement of knots in cubic splines interpolation.

At this stage, it is important to define some ideas about multi-objective optimization problems and describe fundamental issues for GAs in general and NSGA-II algorithm in particular (see [17] and the references below for more details). So, in this subsection we also clarify this methodology of using MOGA as an optimization strategy for the determination of the knots placement for bicubic interpolation splines.

It is well-known that many mathematical and realistic models can be formulated as multi-objective optimization problems and that customized genetic algorithms (GAs) have been demonstrated to be particularly effective obtaining very satisfactory solutions to these type of problems, where usually there is not a unique *best* solution. The concept of GA was introduced in the last sixties and fully developed by Holland [18] and many others during several decades and is fully inspired in the usual selection process of most adapted individuals in natural evolution of species. GA operates then with a collection (called the *population*) of randomly chosen *individuals*, with their particular *chromosomes*, that can be appropriately combined (using the so called *crossover* operator) and/or mutated (using the corresponding *mutation* operator). So the parents must to be adequately chosen in order that their corresponding chromosomes would be combined to produce *better* offsprings (taking into account the *objective/cost* functions considered). Also the mutation operator may introduce some particular random little changes into these chromosomes, although the *mutation rate* (probability of changing the properties of a gene) is usually quite small. In this way, a slow process of change/evolution it is also implemented so that unsuccessful changes are automatically eliminated by this "natural" selection process, while good or adapted changes will prevail.

The first GAs were used mainly to optimize single cost (or mono-objective) function problems; but soon, the necessity of solving also several or many cost (or multi-objective) optimization ones was completely compulsory, due to the great diversity of real-life problems in such situation. In fact, the first multi-objective GA was proposed by Schaffer [19] called Vector Evaluated Genetic Algorithm (VEGA). Afterward, several important Multi-Objective Evolutionary or Genetic Algorithms (MOGA) were developed (see for example [20], Niched Evolutionary Algorithms [21], Random Weighted Genetic Algorithms (RWGA) [22] and many others, that can be consulted in the website [23]. But one of the most reliable and succesful estrategies is the application of the Goldberg's notion of *nondominated sorting* in the GA along with a niche and speciation method to find multiple Pareto-optimal points simultaneously [24]. Also appropriate extensions and improvements are accomplished in the NSGA-II and NSGA-III versions (see [25] for a performance comparison between these two versions). The key points concerning these two new versions of the original one are:

- A fast *nondominated sorting* procedure where he population is sorted into a hierarchy of sub-populations based on the ordering of Pareto dominance. Similarity between members of each sub-group is evaluated on the Pareto front, and the resulting groups and similarity measures are used to promote a diverse front of non-dominated solutions.

- The appropriate consideration of the *elitism approach* in order to enhance the convergence properties of the algorithm.

- A *parameterless niching operator, in order to maintain certain level of multiple solutions.*

More recently, a unified approach for single, multiple and many objective optimization algorithm have been proposed and analyzed in [26]. In this unified version of these algorithms the authors also show how it works with usual Evolutionary Algorithms (EAs) parameters and no additional tunable ones are needed. They also emphasize that certain MOGA procedures, as the NSGA-II, runs very well for bi-objective problems but does not scale up to solve many-objective optimization problems efficiently, although they can work quite well solving mono-objective ones. On the other hand, for three and many-objective problems, the more recent NSGA-III algorithm is preferred, but they propose a single unified efficient procedure capable of handling one to many-objective problems without having to reimplement or change the optimization algorithm if we have to deal with different objective dimensions of the original problem.

In order to verify the ability of generalization of the new multi-objective strategy for the determination of the knots placement for the proposed interpolating bicubic splines, a test data set (TDS) is used and denoted by $X^{Test} = [(x_1, y_1)^{Test}, ..., (x_{ntest}, y_{ntest})^{Test}]$, composed of the nodes and its corresponding output data set: $TDS = [X^{Test}; Z^{Test}]$. With the data of TDS, it is possible to obtain the output data approximation using the presented interpolating bicubic splines methodology. This output set is termed as $\hat{Z}^{Test}$, being $\hat{Z}^{Test} = [\hat{z}_1^{Test}, ..., \hat{z}_{ntest}^{Test}]$, with $ntest$ being the number of nodes in the set X^{Test}.

There are some important parameters that should be defined before running any MOGA algorithm. The most important parameters for the evolutionary strategy are presented in Table 1 , but usually are problem-dependent and have to be chosen empirically.

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

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
Fitnes functions	E_c and E_l
Knots' deletion tolerance	$0.3 * 10^{-2}$

Different forms of fitness, or objective functions, can be used in a NSGA-II procedure, but the main goal is to minimize some of the usual errors between the original function and the interpolating bicubic spline constructed from each population of random knots. We consider two approximation error estimations that are appropriate normalizations of the discrete version of the usual norms in $\mathcal{C}(R)$ and $\mathcal{L}^2(R)$, and are given by the expressions:

$$E_c = \frac{\max\limits_{i=1,\ldots,M} |f(a_i) - S(a_i)|}{\max\limits_{i=1,\ldots,M} |f(a_i)|} \tag{4}$$

$$E_l = \sqrt{\frac{\sum_{i=1}^{M}(f(a_i) - S(a_i))^2}{\sum_{i=1}^{M}(f(a_i))^2}} \tag{5}$$

where $f \in \mathcal{C}^2(R)$ is a given function, $S \in \mathcal{S}_3(\triangle_n \times \triangle_m)$ is the interpolating bicubic spline associated with the given data set (2), and $\{a_1, \ldots, a_M\} \subset R$ is a given scattered random point set where the errors are computed. In the case of functions of only one variable, the expressions are totally equivalent, but the points will be chosen inside the corresponding interval.

Also, in this case of interpolation, we need to consider some deleting or colliding procedure, in order to can remove some of the interpolation knots when they come too close to each other, making instable the associated interpolation procedure (because the matrix of the corresponding linear system become almost singular). So, a certain tolerance parameter is also introduced, in this interpolation case, in order to avoid the possible instabilities and/or bad conditioning of the matrices involved in this problem. This issue could also be avoided if we choose the interpolation points independently of the knots, but this is not the case here, and could be the subject of a much more general procedure, where some Shoenberg-Whitney conditions would also must be taken into consideration [27].

3. Simulation results

To study the behavior of the approximation for the presented methodology, performed by optimization of the knots placement of bicubic spline functions by MOGA, different experiments have been carried out. In order to perform the interpolation, using the proposed methodology, we present the most important parameters for the evolutionary strategy in Table 1, and the following functions are used:

Example 1: $F_1 : [0, \pi] \longrightarrow \mathbb{R}$

$$F_1(x) = 0.12 + .25 \exp^{-4(x-\frac{\pi}{4})^2} \cos(2x) \sin(2\pi x) \quad .$$

We can see clearly in Figure 1 that the evolution of the distribution of knots is located where the function $F_1(x)$ change the most within its domain, and shows the results of the approximating function compared with the original one using B-spline function interpolation. The corresponding Pareto front, taking also into account the error in the interpolation vs. the number of knots used is shown in Figure 2. Similar results (see Figures 4 and 3) are obtained for the second example $F_2(x)$ below.

Example 2: $F_2 : [0,1] \longrightarrow \mathbb{R}$

$$F_2(x) = (2 + e^{-50(2x-0.25)})^{-1} \quad .$$

For experiments with two independent variables, we will use some very well known functions, as the Franke's one in Example 3, and just a paraboloid in Example 4. We can see their graphics in the Figures 5 and 6.

Example 3: $F_3 : [0,1] \times [0,1] \longrightarrow \mathbb{R}$

$$F_3(x,y) := \frac{3}{4}e^{-((9x-2)^2+(9y-2)^2)/4} + \frac{3}{4}e^{-((9x+1)^2/49-(9y+1)/10)} + \frac{1}{2}e^{-((9x-7)^2+(9y-3)^2)/4} - \frac{1}{5}e^{-((9x-4)^2+(9y-7)^2)} \quad .$$

Example 4: $F_4 : [-1,1] \times [-1,1] \longrightarrow \mathbb{R}$

$$F_4(x,y) := x^2 + y^2 \quad .$$

In order to analyze the behavior of the bicubic spline interpolation procedure, a TDS with a large number of knots is not necessary. In Figures 7 and 10 we can see the evolution of interpolating knots in both cases, whereas Figures 8 and 9 show the Pareto fronts for the functions $F_3(x,y)$ and $F_4(x,y)$, respectively.

4. Conclusions

In this paper, a novel methodology is presented for knots placement for cubic and bicubic splines interpolation of functions of one or two variables, respectively, showing the effectiveness of the strategy for different types of functions.

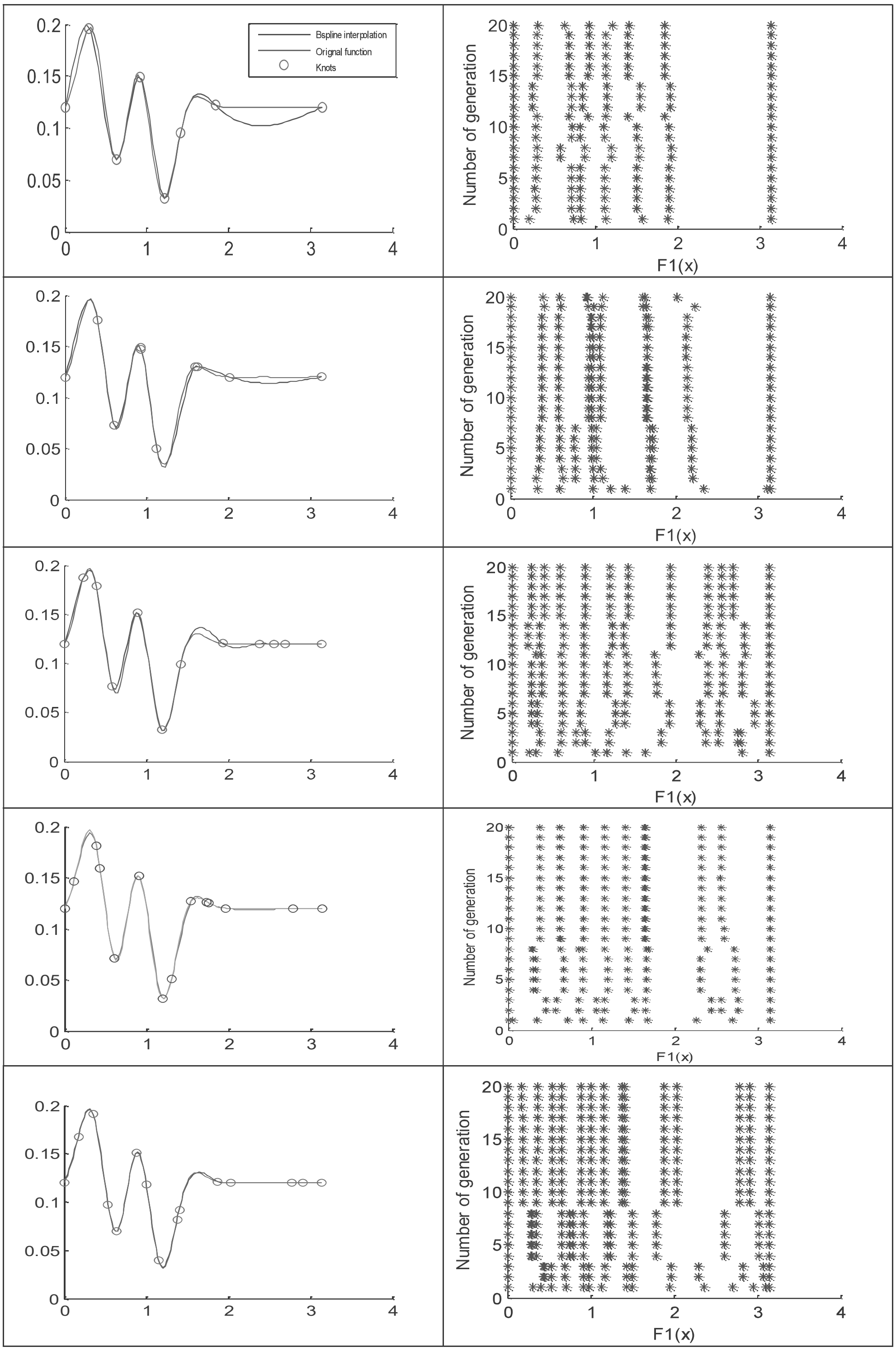

Figure 1: We can see in the left colum the cubic interpolating splines corresponding to the last knots' distribution, whose evolution is also showed in the right column, with increasing number of interior knots for the function $F_1(x)$.

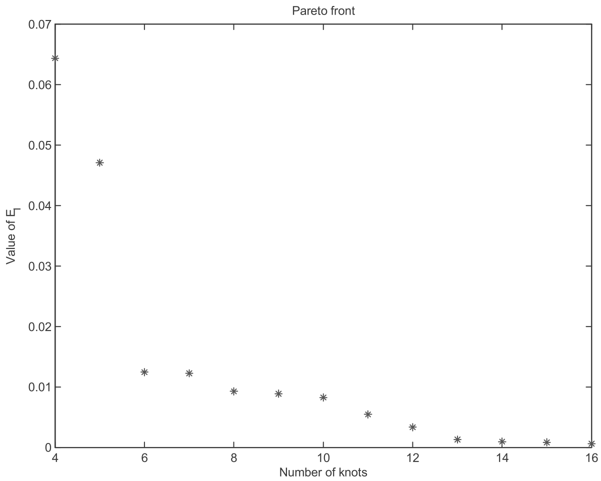

Figure 2: Pareto front of E_l error vs. number of knots for $F_1(x)$.

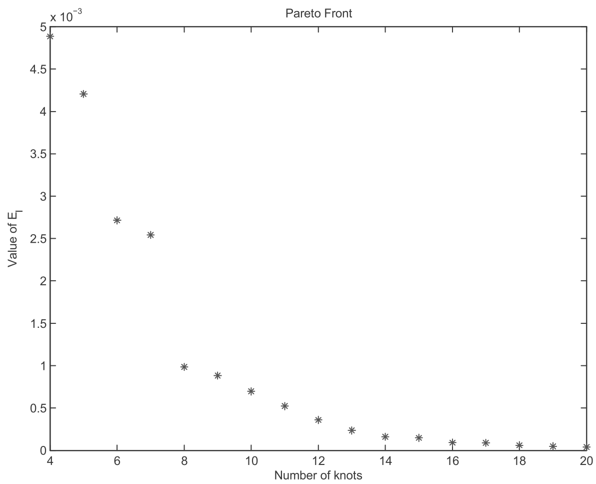

Figure 3: Pareto front of E_l error vs. number of knots for $F_2(x)$.

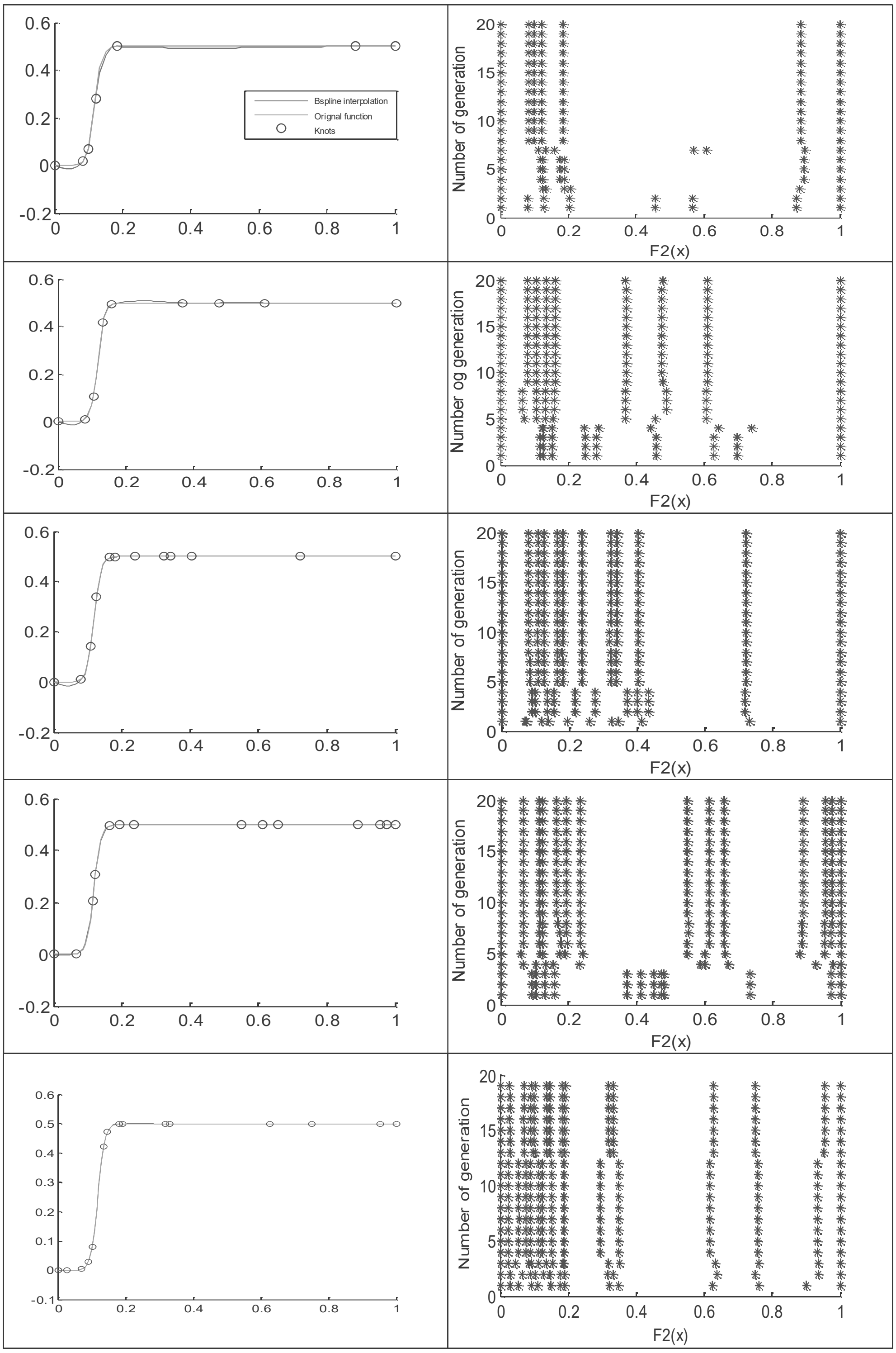

Figure 4: We can see in the left colum the cubic interpolating splines corresponding to the last knots' distribution, whose evolution is also showed in the right column, with increasing number of interior knots for the function $F_2(x)$.

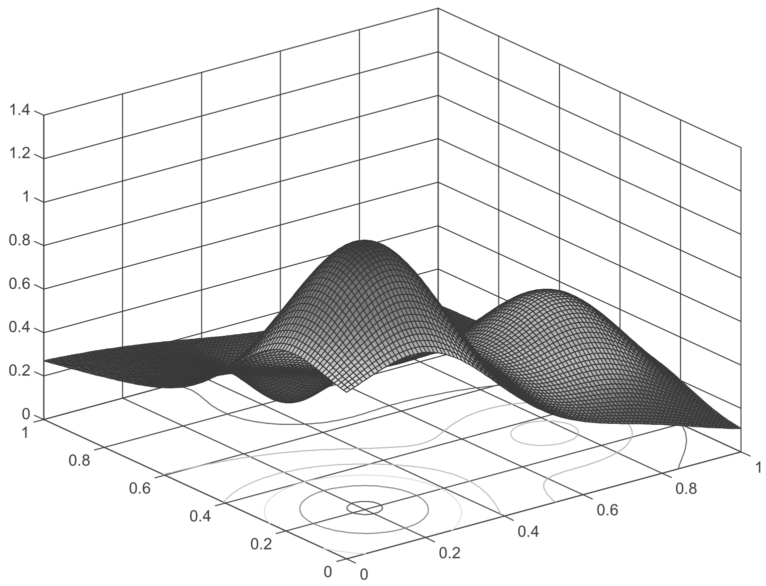

Figure 5: 3D graphic and iso-lines of the Franke's function $F_3(x, y)$.

So, the goal of using a MOGA for placement of the knots in such case of interpolating functions can be summarized as follows:

(1) It has been sufficiently proven that the placement of the knots in spline interpolation has an important and considerable effect on the behavior of the final results, but the optimal placement or location of knots is not known a priori.

(2) The number of knots to be used in classical MOGA approaches, should be selected a priori by the designer; but using our procedure, a Pareto front for different or variable number of knots used can also be directly optimized when less knots are necessary to obtain the same level of approximation.

As can be seen, in all the examples the mean square error (MSE) with

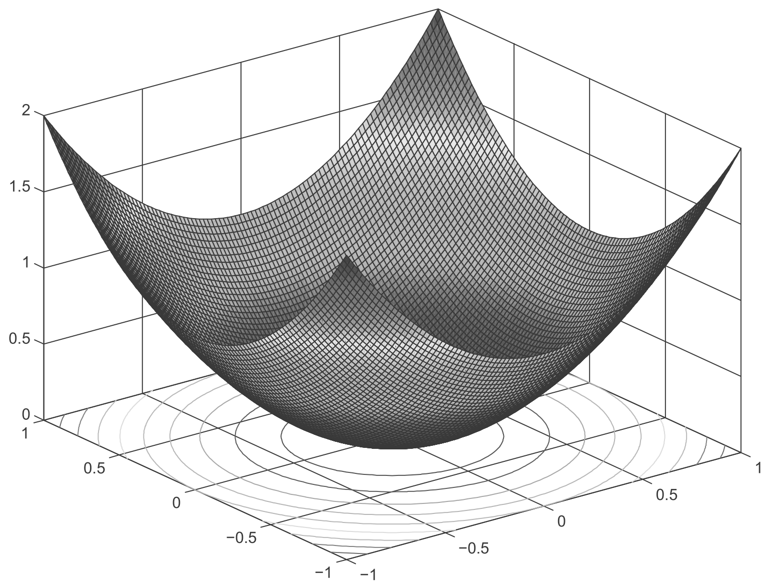

Figure 6: 3D graphic and iso-lines of the $F_4(x, y)$ function.

the MOGA tends to be reduced when the number of knots to construct the interpolating B-spline increase, but after the appropriate evolution of the interpolating knots, not too many of these points are needed to obtain acceptable, or even good, results. In our particular case, we are seeking to minimize at the same time only two normalized discrete versions of some approximate errors, obtained from the usual norms in $\mathcal{C}(R)$ (4) and $\mathcal{L}^2(R)$ (5), that are not opposite convex functions of the involved variables of the MOGA; so the framework of the NSGA-II setting is the more convenient. In fact our main goal minimizing at the same time these discretized errors with our interpolation procedure is not only solve this interpolation issue, but also obtain a good fitting between the

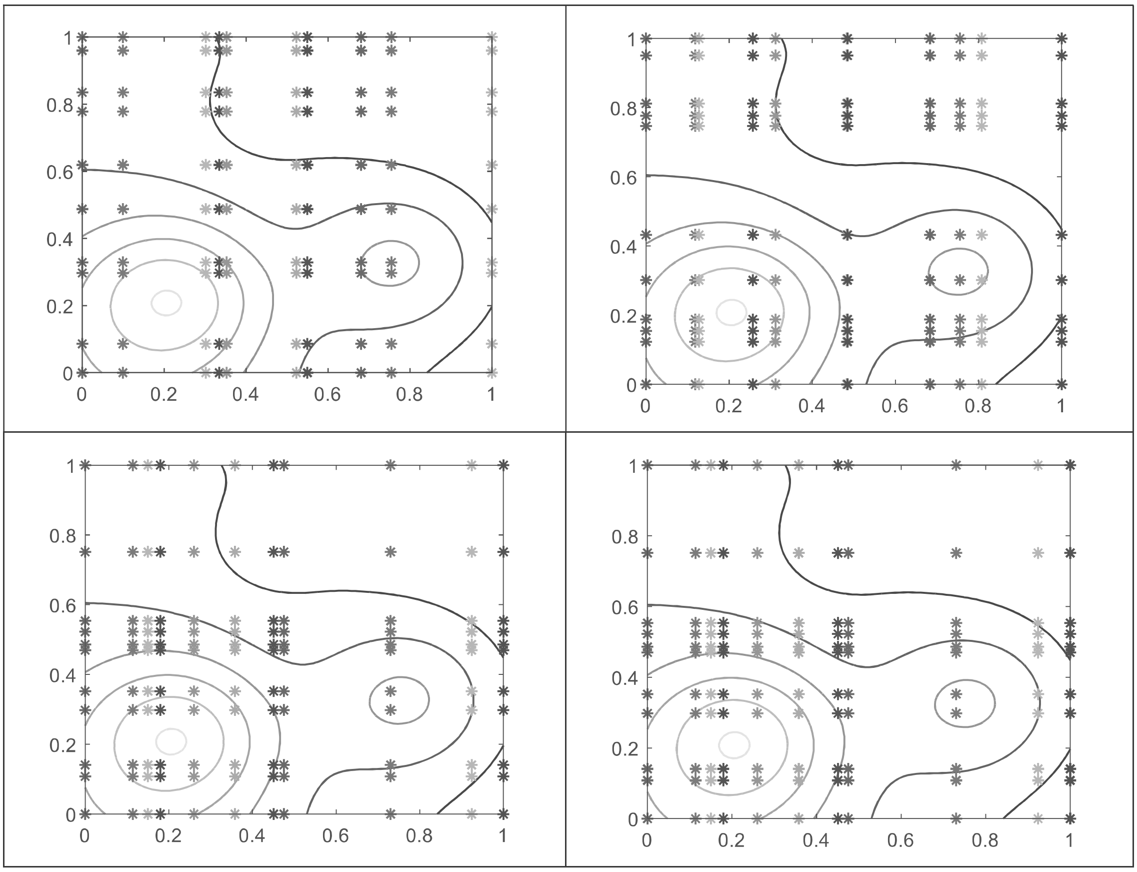

Figure 7: From left to right and top to down we show the evolution of the distribution of 11 points on each axis for $F_3(x, y)$.

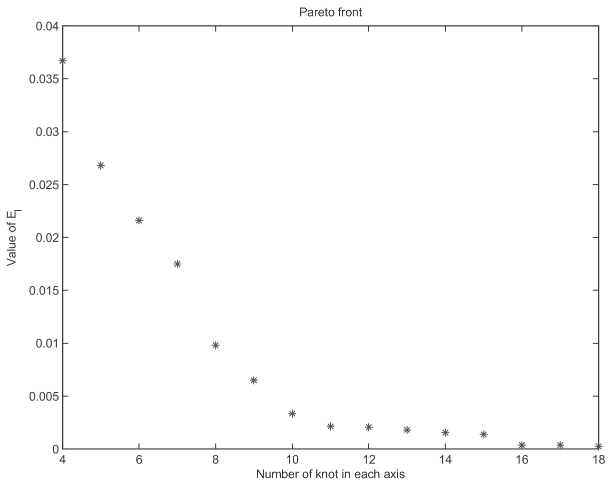

Figure 8: Pareto Front of E_l error vs. number of knots for $F_3(x)$.

original and the interpolated curve or surface ir order to capture the maximum information of it with the representation of the obtained interpolating curve or surface.

References

References

[1] Y.-K. Lai, S.-M. Hu, H. Pottmann, Surface fitting based on a feature sensitive parametrization, Computer-Aided Design 38 (7) (2006) 800 – 807. doi:https://doi.org/10.1016/j.cad.2006.04.007.
URL http://www.sciencedirect.com/science/article/pii/S0010448506000686

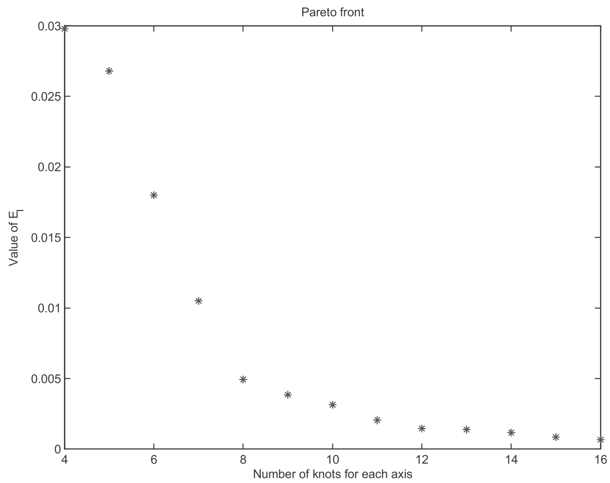

Figure 9: Pareto front of E_l error vs. number of knots for $F_4(x, y)$.

[2] T. Lyche, K. Mørken, The sensitivity of a spline function to perturbations of the knots, BIT Numerical Mathematics 39 (2) (1999) 305–322. doi:10.1023/A:1022346030560.
URL https://doi.org/10.1023/A:1022346030560

[3] R.-H. Wang, J.-X. Wang, The poisedness of interpolation problem for splines, Applied Numerical Mathematics 54 (1) (2005) 95 – 103. doi:https://doi.org/10.1016/j.apnum.2004.08.001.
URL http://www.sciencedirect.com/science/article/pii/S0168927404001205

[4] H. Park, An error-bounded approximate method for representing planar curves in b-splines, Computer Aided Geometric Design 21 (5) (2004) 479 – 497. doi:https://doi.org/10.1016/j.cagd.2004.03.003.

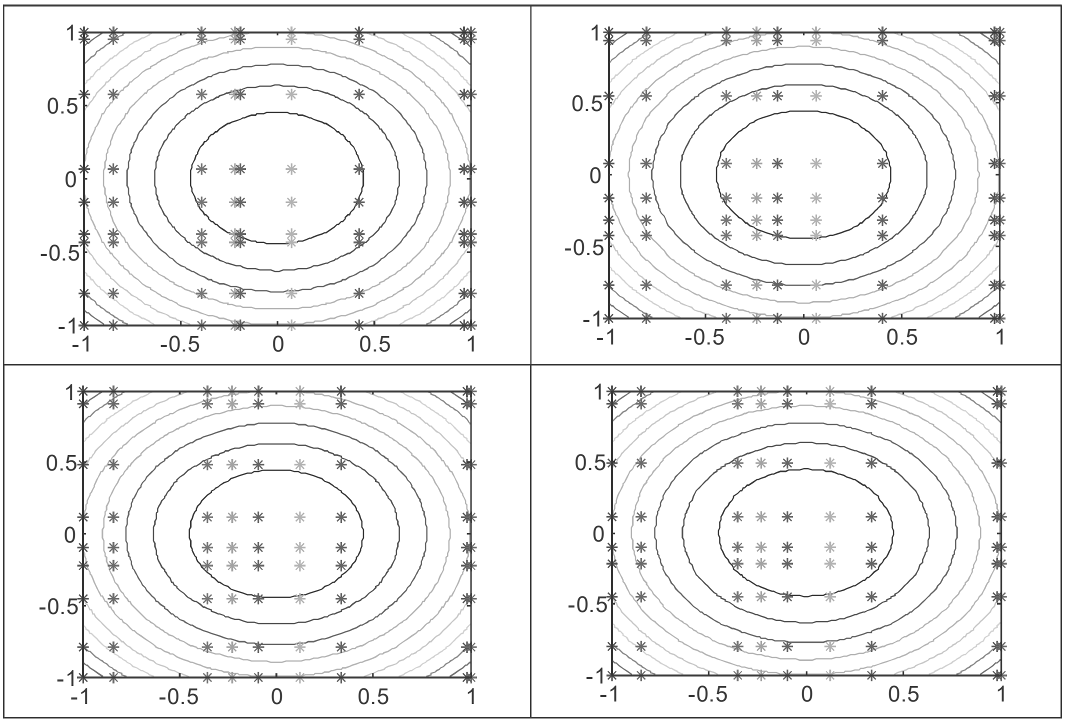

Figure 10: From left to right and top to down we show the evolution of the distribution of interpolating knots for $F_4(x, y)$.

URL http://www.sciencedirect.com/science/article/pii/S0167839604000445

[5] H. Park, J.-H. Lee, B-spline curve fitting based on adaptive curve refinement using dominant points, Comput. Aided Des. 39 (6) (2007) 439–451. doi:10.1016/j.cad.2006.12.006.
URL http://dx.doi.org/10.1016/j.cad.2006.12.006

[6] M. Pasadas, M. Rodríguez, An approximation method with data selection process, Mathematics and Computers in Simulation 79 (12) (2009) 3567 – 3576, the International Conference on Approximation Methods and numerical Modeling in Environment and Natural Resources. doi:https://doi.org/10.1016/j.matcom.2009.04.019.

URL http://www.sciencedirect.com/science/article/pii/S0378475409001207

[7] A. Gálvez, A. Iglesias, A. Avila, C. Otero, R. Arias, C. Manchado, Elitist clonal selection algorithm for optimal choice of free knots in b-spline data fitting, Appl. Soft Comput. 26 (C) (2015) 90–106. doi:10.1016/j.asoc.2014.09.030.
URL http://dx.doi.org/10.1016/j.asoc.2014.09.030

[8] R. Kozera, L. Noakes, Optimal knots selection for sparse reduced data, in: F. Huang, A. Sugimoto (Eds.), Image and Video Technology – PSIVT 2015 Workshops, Springer International Publishing, Cham, 2016, pp. 3–14.

[9] C. H. Garcia-Capulin, F. J. Cuevas, G. Trejo-Caballero, H. Rostro-Gonzalez, A hierarchical genetic algorithm approach for curve fitting with b-splines, Genetic Programming and Evolvable Machines 16 (2) (2015) 151–166. doi:10.1007/s10710-014-9231-3.
URL http://dx.doi.org/10.1007/s10710-014-9231-3

[10] K. F. Man, K. S. Tang, S. Kwong, Genetic Algorithms: Concepts and Designs with Disk, 2nd Edition, Springer-Verlag, Berlin, Heidelberg, 1999.

[11] R. Interian, J. M. Otero, C. C. Ribeiro, A. A. Montenegro, Curve and surface fitting by implicit polynomials: Optimum degree finding and heuristic refinement, Computers & Graphics 67 (2017) 14 – 23. doi:https://doi.org/10.1016/j.cag.2017.05.002.
URL http://www.sciencedirect.com/science/article/pii/S0097849317300444

[12] Y. Zhang, J. Cao, Z. Chen, X. Li, X.-M. Zeng, B-spline surface fitting with knot position optimization, Comput. Graph. 58 (C) (2016) 73–83. doi:10.1016/j.cag.2016.05.010.
URL http://dx.doi.org/10.1016/j.cag.2016.05.010

[13] H. Kang, F. Chen, Y. Li, J. Deng, Z. Yang, Knot calculation for spline fitting via sparse optimization, Comput. Aided Des. 58 (C) (2015) 179–188. doi:10.1016/j.cad.2014.08.022.
URL http://dx.doi.org/10.1016/j.cad.2014.08.022

[14] O. Valenzuela, M. Pasadas, Using simulated annealing for knot placement for cubic spline approximation, in: Proceedings of the 2010 International Conference on Mathematical Models for Engineering Science, MMES'10, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 2010, pp. 148–154.
URL http://dl.acm.org/citation.cfm?id=1965658.1965698

[15] R. V. Rao, Teaching Learning Based Optimization Algorithm: And Its Engineering Applications, 1st Edition, Springer Publishing Company, Incorporated, 2015.

[16] R. V. Rao, A. Saroj, A self-adaptive multi-population based jaya algorithm for engineering optimization, Swarm and Evolutionary Computation 37 (2017) 1 – 26. doi:https://doi.org/10.1016/j.swevo.2017.04.008.
URL http://www.sciencedirect.com/science/article/pii/S2210650216303510

[17] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, Trans. Evol. Comp 6 (2) (2002) 182–197. doi:10.1109/4235.996017.
URL http://dx.doi.org/10.1109/4235.996017

[18] J. H. Holland, Adaptation in Natural and Artificial Systems, 1975.
URL https://mitpress.mit.edu/books/adaptation-natural-and-artificial-systems

[19] J. D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1985,

pp. 93–100.
URL http://dl.acm.org/citation.cfm?id=645511.657079

[20] C. M. Fonseca, P. J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evol. Comput. 3 (1) (1995) 1–16. doi:10.1162/evco.1995.3.1.1.
URL http://dx.doi.org/10.1162/evco.1995.3.1.1

[21] O. Kramer, H. Danielsiek, Dbscan-based multi-objective niching to approximate equivalent pareto-subsets, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10, ACM, New York, NY, USA, 2010, pp. 503–510. doi:10.1145/1830483.1830575.
URL http://doi.acm.org/10.1145/1830483.1830575

[22] S. Wang, S. Ali, A. Gotlieb, Random-weighted search-based multi-objective optimization revisited, in: C. Le Goues, S. Yoo (Eds.), Search-Based Software Engineering, Springer International Publishing, Cham, 2014, pp. 199–214.

[23] C. Coello, List of references on evolutionary multiobjective optimization.
URL http://delta.cs.cinvestav.mx/~ccoello/EMOO/emoopage.html

[24] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. 2 (3) (1994) 221–248. doi:10.1162/evco.1994.2.3.221.
URL http://dx.doi.org/10.1162/evco.1994.2.3.221

[25] H. Ishibuchi, R. Imada, Y. Setoguchi, Y. Nojima, Performance comparison of nsga-ii and nsga-iii on various many-objective test problems, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 3045–3052. doi:10.1109/CEC.2016.7744174.

[26] H. Seada, K. Deb, U-nsga-iii: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results,

in: A. Gaspar-Cunha, C. Henggeler Antunes, C. C. Coello (Eds.), Evolutionary Multi-Criterion Optimization, Springer International Publishing, Cham, 2015, pp. 34–49.

[27] M. Sommer, H. Strauss, A condition of Schoenberg–Whitney type for multivariate spline interpolation 5 (1) (1996) 381–397. doi:https://doi.org/10.1007/BF02124752.
URL http://link.springer.com/article/10.1007/BF02124752

***Response to Reviewers**

Following the suggestions of the editors and referees, we have highlight in red color all the revised and changed parts of the article, correcting all the little errata and explaining in more detail the concepts and procedures that were not sufficiently clear before; making also an effort to place the work with regards to the related literature.

Concerning the comments of the Reviewer 1:
- We have explained with much more detail how MOGA works in general and what it does for this particular problem.
- We do not have explicit comparison time and errors with other methods in the literature, because they do not solve the same problem of interpolation, just the approximation case, and only in one variable. We focus our procedure in two variables as well, and only present the one variable case for sake of completeness.

Concerning the comments of the Reviewer 2:
- We have corrected all the little erratas encountered
- Higlight the originality and differences between our procedure and the rest of methods in the literature
- Explained in more detail and more clearly the AIS and the NSGA procedures, - Answered in the text, other little questions and comments made for this reviewer.