



**HAL**  
open science

# A surrogate model based on Non-Uniform Rational B-Splines hypersurfaces

Yohann Audoux, Marco Montemurro, Jérôme Pailhès

► **To cite this version:**

Yohann Audoux, Marco Montemurro, Jérôme Pailhès. A surrogate model based on Non-Uniform Rational B-Splines hypersurfaces. *Procedia CIRP*, 2018, 70, pp.463-468. hal-02354416

**HAL Id: hal-02354416**

**<https://hal.science/hal-02354416v1>**

Submitted on 7 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

28th CIRP Design Conference, May 2018, Nantes, France

# A surrogate model based on Non-Uniform Rational B-Splines hypersurfaces

Y. Audoux<sup>a</sup>, M. Montemurro<sup>a,\*</sup>, J. Pailhes<sup>a</sup>

<sup>a</sup> Arts et Métiers ParisTech, 12M, CNRS UMR 5295, Esplanade des Arts et Métiers, F-33400 Talence, France

\* Corresponding author. Tel.: +33-556-845-333. E-mail address: marco.montemurro@ensam.eu

## Abstract

This study aims at providing an original metamodeling technique based on the Non-Uniform Rational B-Splines (NURBS) formalism. The proposed approach is able to fit general non-convex sets of target points (TPs) by extending the NURBS formalism to the N-dimensional (N-D) case, getting in this way a general NURBS hypersurface. The shape of such a hypersurface is tuned by several parameters: the number of control points (CPs), their coordinates and related weights, the degrees of the blending functions and the knot-vector components defined along each direction. The goal of the proposed strategy is to find the best NURBS hypersurface approximating a given set of TPs. To this purpose the problem is formulated as an unconstrained least-square distance problem wherein the optimisation variables are all the parameters tuning the shape of the NURBS hypersurface. Nevertheless, when the number of CPs and the degrees of the basis functions are included among the design variables the resulting problem is defined over a space having a variable dimension. To deal with this aspect, a special genetic algorithm, able to solve problems characterised by a variable number of design variables, is considered to determine automatically (i.e. without the user's intervention) the optimum value of both the design space size (related to the integer variables of the NURBS hypersurface) and the NURBS hypersurface continuous parameters. The effectiveness of the proposed approach is proven by means of a meaningful benchmark.

© 2018 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the 28th CIRP Design Conference 2018.

*Keywords:* Optimization ; Genetic Algorithm ; NURBS ;

## 1. Introduction

Computers increasing performances are widely utilized to improve simulation complexity over computation time. However, some applications need quasi-real-time models especially for design/optimisation purposes. Metamodeling techniques can minimise the computational effort to realize complex multi-field and/or multi-scale simulations which must be integrated within an optimisation process. Currently, Artificial Neural Networks (ANN) [1,2], Proper Generalized Decomposition (PGD) [3], Fuzzy Logic (FL) [4] and Radial Basis Functions (RBFs) [5] are the most common methods available in literature for realising surrogate models. Of course, each technique is characterised by both advantages and shortcomings. For example, ANN are not suitable for multiple outputs request [2] (when considering multiple responses ANN

accuracy often decreases), while PGD provides less accurate results for nonlinear problems [3]. As a general remark, setting the parameters number for classical metamodeling techniques could be a quite difficult task which often needs a trial-and-error approach. For instance, improving arbitrarily the number of layers and/or neurons for ANN or improving the modes number for PGD can lead to overfitting and to a subsequent increase of the overall complexity for the problem at hand. In any case, all these methods need a great amount of available data, thus implying extensive experimental and/or numerical campaigns.

The surrogate model proposed in this study relies on the utilisation of  $N$ -dimension ( $N$ -D) Non-Uniform Rational Basis Spline (NURBS) hypersurfaces to fit a given set of data points, also called target points (TPs) and aims at overcoming the previous limitations by automatically determining the optimum

number of the NURBS hypersurface parameters without requiring a trial-and-error approach.

Up to now, only few research works focus on the formulation/implementation of surrogate models based on the NURBS formalism [6–8]. NURBS curves and surfaces are standard geometrical entities widely used in Computer Aided Design (CAD) software. NURBS hypersurfaces represent a generalisation of these entities. A NURBS hypersurface is defined through the number of dimensions (related to the size of the problem at hand), the degree of blending-functions along each dimension, the overall number of control points (CPs), the coordinates of each CPs and the related weights, the knot vector components along each dimension. The large number of parameters tuning the shape of a NURBS hypersurface renders it a very versatile tool for many mathematical and engineering applications, not only for formulating surrogate models [9–12]. However, the significant amount of parameters defining the NURBS hypersurface also constitutes the main drawback: it is very hard to properly tune all these parameters without making some simplifying assumptions or preliminary choices as done in [6,7].

In [8] a NURBS-based surrogate model for inverse characterization of composite materials is presented. In this background Turner and Crawford [6] developed an iterative procedure for determining a suitable number of CPs for the NURBS hypersurface. The technique proposed by Turner and Crawford is based upon an empirical rule which computes the position of new control points on the basis of the cost function of the problem at hand (typically the maximum error of approximation). Nevertheless, their approach is characterised by some restrictions:

- the empirical rule for updating the number of CPs as well as their coordinates is strongly problem-dependent;
- the degrees of blending functions were set *a priori*;
- the knot vectors components were uniformly distributed or calculated by means of simple empirical rules.

To overcome the previous restrictions, in this work, an innovative surrogate model based on  $N$ -D NURBS hypersurfaces is proposed. The problem of approximating a given set of TPs is formulated as an unconventional Unconstrained Non-Linear Programming Problem (UNLPP) defined over a space of *variable dimension*. Furthermore, this UNLPP is formulated without considering simplifying hypotheses, thus by considering as design variables all the parameters tuning the NURBS hypersurface shape (both *integer* and *continuous* parameters).

Of course, when dealing with an optimisation problem defined over a domain having a variable dimension, a particular care must be put in the choice of a proper numerical tool to perform the solution search. To this purpose in this study a special genetic algorithm (GA) [11,13] able to deal with problems characterised by a “variable number of design variables” is utilised as optimisation tool.

The effectiveness of the proposed approach will be proven through a meaningful benchmark.

## 2. A NURBS-based surrogate model

### 2.1. Classic NURBS surfaces theoretical framework

Non-Uniform Rational B-Splines curves and surfaces are explicit parametric geometric entities mostly used in CAD software. According to the notation introduced in [14] the Cartesian coordinates of a generic point of a NURBS surface can be written as:

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(u) N_{j,q}(v) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_{i,p}(u) N_{j,q}(v) \omega_{i,j}}, \quad (1)$$

where  $u, v \in [0, 1]$  are the dimensionless parameters of the surface,  $\mathbf{S} \in \mathfrak{R}^3$  is the vector collecting the coordinates of the generic point belonging to the surface, whilst  $\mathbf{P}_{i,j}$  is the array

of the CPs coordinates, i.e. one array of size  $\mathfrak{R}^{(n_u+1) \times (n_v+1)}$  is required for each coordinate.  $(n_u + 1)$  and  $(n_v + 1)$  are the number of CPs along  $u$  and  $v$  directions, respectively.

$\omega_{i,j} \in \mathfrak{R}^{(n_u+1) \times (n_v+1)}$  is the array of weights related to each CP.  $N_{i,p}(u)$  and  $N_{j,q}(v)$  are the  $p$ -th and  $q$ -th -degree B-spline blending functions, which are recursively defined. For example, the expression of  $N_{i,p}(u)$  is written as:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } U_i \leq u < U_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

$$N_{i,\tau}(u) = \frac{u - U_i}{U_{i+\tau} - U_i} N_{i,\tau-1}(u) + \frac{U_{i+\tau+1} - u}{U_{i+\tau+1} - U_{i+1}} N_{i+1,\tau-1}(u),$$

$$i = 0, \dots, n_u, \tau = 1, \dots, p.$$

Similar expressions apply for  $N_{i,q}(v)$ . In Eq. (2)  $U_i$  is the  $i$ -th component of the non-periodic non-uniform knot vector along  $u$  direction. Of course, a knot vector is required also in  $v$ -direction in order to define the associated blending functions. The generic expression for both knot vectors is:

$$\mathbf{U} = \left\{ \underbrace{0, \dots, 0}_{p+1}, U_{p+1}, \dots, U_{m_u-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\}, \quad (3)$$

$$\mathbf{V} = \left\{ \underbrace{0, \dots, 0}_{q+1}, U_{q+1}, \dots, U_{m_v-q-1}, \underbrace{1, \dots, 1}_{q+1} \right\}.$$

The size of knot vectors  $\mathbf{U}$  and  $\mathbf{V}$  are  $(m_u + 1)$  and  $(m_v + 1)$ , respectively. Their size strictly depends upon the number of CPs as well as on the degrees of blending functions along each direction, namely

$$m_u = n_u + p + 1, \quad (4)$$

$$m_v = n_v + q + 1.$$

Each knot vector is a non-decreasing sequence of real numbers that can be interpreted as a discrete collection of values of the dimensionless parameter of the surface. All the components of knot vectors can have a multiplicity  $\mu$ . If a knot has a

multiplicity  $\mu$  along  $u$ -direction then the basis function  $N_{i,p}(u)$  is  $p-\mu$  times continuously differentiable (at the knot).

The same considerations can be repeated for  $N_{i,q}(v)$  if a component of  $\mathbf{V}$  has multiplicity  $\mu$ . Therefore, for a fixed degree along a given direction, increasing the knot multiplicity decreases the continuity of the surface at that knot. This means that the value of the knot vectors components strongly affects the local shape of a NURBS surface. For a deeper insight in the matter, the reader is addressed to [14].

### 2.2. The case of NURBS hypersurfaces

Let  $N$  be the dimension of the problem and  $M$  that of the hypersurface. Let  $\mathbf{S} : \mathfrak{R}^N \rightarrow \mathfrak{R}^M$  be the parametric vector equation of the hypersurface. Analogously to the 2D case, the explicit parametric expression of the  $k$ -th coordinate of the NURBS hypersurface writes:

$$S^{(k)}(u_1, \dots, u_N) = \frac{\sum_{i_1=0}^{n_1} \dots \sum_{i_N=0}^{n_N} N_{i_1, p_1}(u_1) \dots N_{i_N, p_N}(u_N) \omega_{i_1, \dots, i_N} P_{i_1, \dots, i_N}^{(k)}}{\sum_{i_1=0}^{n_1} \dots \sum_{i_N=0}^{n_N} N_{i_1, p_1}(u_1) \dots N_{i_N, p_N}(u_N) \omega_{i_1, \dots, i_N}}, \quad k=1, \dots, M \quad (5)$$

In Eq. (5),  $u_1, \dots, u_N, \in [0, 1]$  are the dimensionless parameters of the hypersurface, while  $S^{(k)}$  is the  $k$ -th coordinate of the generic point belonging to the hypersurface.  $P_{i_1, \dots, i_N}^{(k)} \in \mathfrak{R}^{(n_1+1) \times \dots \times (n_N+1)}$  is the generic component of the array collecting the  $k$ -th coordinate for each CP.  $(n_i + 1)$  is the number of CPs along the  $i$ -th dimension, while  $n_{CP} = \prod_{i=1}^N (n_i + 1)$  is the overall number of CPs which form

the so-called *hyper-net*.  $\omega_{i_1, \dots, i_N} \in \mathfrak{R}^{(n_1+1) \times \dots \times (n_N+1)}$  is the generic component of the array of weights related to each CP. The expression of the generic  $p_i$ -th degree blending function along dimension  $i$  can be stated as:

$$N_{j,0}(u_i) = \begin{cases} 1 & \text{if } U_j^{(i)} \leq u_i \leq U_{j+1}^{(i)} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{j,\tau}(u_i) = \frac{u_i - U_j^{(i)}}{U_{j+\tau}^{(i)} - U_j^{(i)}} N_{j,\tau-1}(u_i) + \frac{U_{j+\tau+1}^{(i)} - u_i}{U_{j+\tau+1}^{(i)} - U_{j+1}^{(i)}} N_{j+1,\tau-1}(u_i), \quad (6)$$

$i = 1, \dots, N, j = 0, \dots, n_i, \tau = 1, \dots, p.$

Of course, blending functions are defined by considering knot vectors component along each dimension. The expression of the generic knot vector along dimension  $i$  is

$$\mathbf{U}^{(i)} = \left\{ \underbrace{0, \dots, 0}_{p_i+1}, U_{p_i+1}^{(i)}, \dots, U_{m_i-p_i-1}^{(i)}, \underbrace{1, \dots, 1}_{p_i+1} \right\}, \quad (7)$$

where the size of the knot vector is given by

$$m_i = n_i + p_i + 1, \quad i = 1, \dots, N. \quad (8)$$

## 3. Optimization problem formulation

### 3.1. Problem description

The problem under consideration here focuses on the approximation of the temperature field of a thin plate of size  $L_x = 0.1$  m and  $L_y = 0.1$  m along  $x$  and  $y$  axes, respectively. The thickness of the plate is  $t$ . A constant flux  $\Phi_0 = 100$  W/m<sup>2</sup> is applied at  $x = 0$  and  $x = L_x$  while the rest of the plate is under convection flux with coefficient  $h$  and infinite temperature  $T_\infty = 298,15$  K. A pictorial representation is given in Fig. 1.

As stated above, the proposed NURBS-based surrogate model aims at approximating the temperature field of the plate for different combinations of variables  $t$  and  $h$  which can be seen as design parameters for the problem at hand. Therefore, the temperature field needs to be determined for each value of the previous parameters, namely:

$$T = f(x, y, t, h) \quad (9)$$

The goal is thus to determine a suitable NURBS hypersurface fitting (with a certain level of accuracy) a set of given TPs. These TPs must be calculated for different values of  $t$  and  $h$  in each point of the plate. For the considered example the overall dimensions of the NURBS hypersurface are  $N=4$  and  $M=1$ , i.e., the NURBS hypersurface reduces to a scalar function depending on four independent parameters as stated in Eq. (9).

The TPs coordinate (i.e. the temperature) can be collected into a  $N$ -D array:

$$Q_{i_1, i_2, i_3, i_4} = T(x_{i_1}, y_{i_2}, t_{i_3}, h_{i_4}), \quad \forall i_j \in [0; r_j] \quad (10)$$

$r_j+1$  being the number of TPs along dimension  $j$ . The set of TPs is composed of an overall number  $n_Q$  of points given by:

$$n_{TP} = \prod_{j=1}^N (r_j + 1). \quad (11)$$

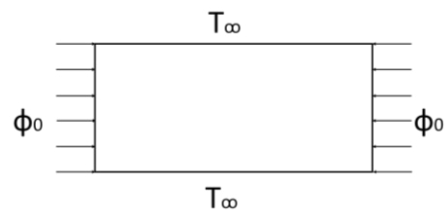


Fig. 1. Geometry and boundary conditions for the plate.

The set of TPs (i.e. the temperature field in each point of the plate for different combinations of  $t$  and  $h$ ) has been obtained by means of a finite element (FE) analysis. The FE model has been coded into ANSYS environment and it is composed of 101x101 *Shell131* elements with a single DOF per node. Fig. 2 illustrates the trend of the temperature field (dimensionless) over the plate for a given pair of thickness and convection coefficient values.

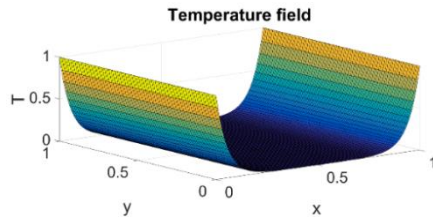


Fig. 2. Dimensionless temperature field over the plate for a given combination of thickness and convection coefficient.

### 3.2. Mathematical statement of the problem

The main goal of the proposed approach is to obtain a surrogate model formulated in the theoretical framework of NURBS hypersurfaces, characterized by an overall number of CPs ( $n_{CP}$ ) lower than that of TPs ( $n_{TP}$ ) describing the response of the system with a sufficient accuracy (according to the problem requirements).

The idea is to search for the best value of the parameters tuning the shape of the NURBS hypersurface minimizing the following error function:

$$\min_{\mathbf{X}} \Phi(\mathbf{X}), \quad \Phi = \sum_{k_1=0}^{r_1} \dots \sum_{k_N=0}^{r_N} \left\| \mathbf{S}(\mathbf{u}_{k_1, \dots, k_N}) - \mathbf{Q}_{k_1, \dots, k_N} \right\|^2 \quad (12)$$

without introducing any simplifying hypotheses or empirical rule on the overall number of parameters involved in the NURBS hypersurface definition.

As clearly appears from Eq. (12), the NURBS hypersurface fitting problem can be stated as an unconstrained least-square problem.  $\mathbf{X}$  is the vector collecting the optimization variables,  $\mathbf{Q}_{k_1, \dots, k_N}$  is the array of TPs coordinates while  $\mathbf{S}(\mathbf{u}_{k_1, \dots, k_N})$  is its counterpart belonging to the NURBS hypersurface when the dimensionless parameters  $u_i$  take the values  $u_i^{(k)}$ , ( $k = 0, \dots, r_i$ ,  $i = 0, \dots, 4$ ).  $\mathbf{u}_{k_1, \dots, k_N}$  is the array containing all the  $u_i^{(k)}$ .

For the problem at hand,  $\mathbf{X}$  collects only the *independent design variables* of the NURBS hypersurface.

From Eq. (5) the definition of a NURBS hypersurface involves parameters of different nature:

- integers ones as the numbers of both knot vector components and CPs, ( $m_{i+1}$ ,  $n_{i+1}$ , respectively) as well as the degrees of the blending functions  $p_i$  along each dimension;
- continuous variables as internal knot vector values  $U_{p_i+1}^{(i)}, \dots, U_{m_i-p_i-1}^{(i)}$ , CPs coordinates along each direction  $\mathbf{P}_{i_1, \dots, i_N}^{(k)}$ , weights  $\omega_{i_1, \dots, i_N}$  and the dimensionless parameters  $u_i^{(k)}$ .

Some of these parameters are interdependent, whereas other can be smartly chosen.

In particular, as far as concerns the set of continuous dimensionless parameters, i.e.  $u_i^{(k)}$ , they are obtained according to the *chord length method* [14]. Furthermore, due to the smoothness of the set of TPs (which form a convex set) a B-Spline hypersurface is sufficient to build the surrogate model (in this case all weights have been set equal to one).

The number of CPs along each dimension can be determined once the size of the knot vector and the degree of the blending functions along the same dimension are known, see Eq. (8).

Moreover, when the size of the knot vector as well as its internal components along each dimension are known, the degree of the blending functions (along each direction) is given and the values of  $u_i^{(k)}$  have been computed by means of the chord length method, finding the optimum value of the CPs coordinates is a quite trivial task. Indeed, problem (12) is convex in terms of CPs coordinates. It can be proven that the optimum value of CPs coordinates can be determined by generalizing the well-known surface fitting algorithm A9.7 from [14] to the  $N$ -D case. A schematic flow chart of this general algorithm is shown in Fig 3. For sake of brevity, it can be stated that the optimum value of CPs coordinates can be computed as a succession of curve fitting problems along each dimension. In particular, this algorithm provides the CPs coordinates by solving successively linear systems, as follows:

$$(\mathbf{N}_i^T \mathbf{N}_i) \mathbf{P}_i^{(k)} = \mathbf{N}_i^T \mathbf{R}_i^{(k)}, \quad (13)$$

with

$$\mathbf{N}_i = \begin{bmatrix} N_{0,p_i}(u_i^{(0)}) & \Lambda & N_{n_i,p_i}(u_i^{(0)}) \\ \vdots & & \vdots \\ M & O & M \\ \vdots & & \vdots \\ N_{0,p_i}(u_i^{(r_i)}) & \Lambda & N_{n_i,p_i}(u_i^{(r_i)}) \end{bmatrix}, \quad \mathbf{P}_i^{(k)} = \begin{bmatrix} P_{0_i, \dots, i_N}^{(k)} \\ \vdots \\ P_{r_i, \dots, i_N}^{(k)} \end{bmatrix}, \quad \mathbf{R}_i^{(k)} = \begin{bmatrix} Q_{0_i, \dots, i_N}^{(k)} \\ \vdots \\ Q_{r_i, \dots, i_N}^{(k)} \end{bmatrix}, \quad (14)$$

$i = 1, \dots, N$ ;  $k = 1, \dots, M$ .

In Eq. (14)  $\mathbf{P}_i^{(k)} \in \mathfrak{R}^{(n_i+1) \times (r_2+1) \times \dots \times (r_N+1)}$  is a temporary array collecting the unknowns for the curve fitting problem along dimension  $i$ . It is calculated “column by column” along the remaining dimensions. When passing to the curve fitting problem along dimension 2 (i.e., the second iteration of the calculation) its size changes to  $\mathbf{P}_i^{(k)} \in \mathfrak{R}^{(n_1+1) \times (n_2+1) \times \dots \times (r_N+1)}$  until iteration  $N$  where the size is finally  $\mathbf{P}_i^{(k)} \in \mathfrak{R}^{(n_1+1) \times (n_2+1) \times \dots \times (n_N+1)}$ .

Generally speaking, for surface fitting problems, matrix  $\mathbf{N}_i^T \mathbf{N}_i$  can be ill-conditioned (depending on the values of the different parameters tuning the shape of the NURBS hypersurface). To this purpose a particular attention must be put in calculating its inverse. Therefore, the Moore-Penrose pseudo-inverse method has been considered to solve the linear system of Eq. (14).

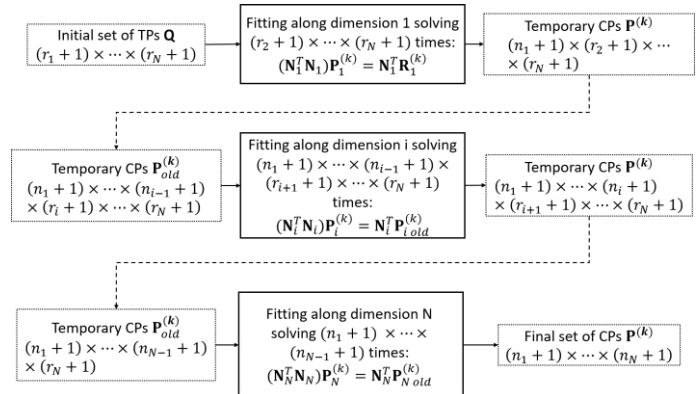


Fig. 3. Logical flow of the surface fitting algorithm [14] extended to the  $N$ -D case.

Beside the values  $u_i^{(k)}$ , the algorithm of Fig. 3 needs some further input quantities in order to be executed:

- integer parameters, i.e.,  $p_i$  and  $m_i$ ,  $i = 1, \dots, N$ ,
- continuous parameters, i.e., internal knot vector components  $U_{p_i+1}^{(i)}, \dots, U_{m_i-p_i-1}^{(i)}$ .

These quantities represent the independent optimization variables collected into the vector  $\mathbf{X}$ .

### 3.3. Numerical strategy

Problem (12) is a non-standard UNLPP because it is defined over a space of *variable dimension*. Indeed, the size of the vector of optimization variables depends upon the current value of the integer design parameters, namely  $p_i$  and  $m_i$ . In particular the size of vector  $\mathbf{X}$  is:

$$2N + \sum_{i=1}^N m_i - 2p_i - 1. \quad (15)$$

Considering the unique mathematical features of problem (12) a hybrid optimization tool composed of the new version of the GA BIANCA [13], interfaced with the MATLAB *fmincon* algorithm [15], has been developed, see Fig. 4. As shown in Fig. 4, the optimization procedure for problem (12) is split in two phases.

During the first phase, solely the GA BIANCA is utilized to perform the solution search and the full set of design variables is taken into account. BIANCA is a special GA able to deal with optimization problems characterized by a variable number of design variables, i.e., optimization problems of modular systems. This goal can be achieved thanks to the original features of such a GA. In BIANCA the information is organized in a *genotype* composed of *chromosomes* which are in turn made of *genes* (each gene codes a specific design variable). When the object of the optimization problem is a *modular system* (i.e., a system composed of a variable number of modules characterized by the same vector of variables) each constitutive module is represented by a chromosome, while each gene codes a design variable related to the module.

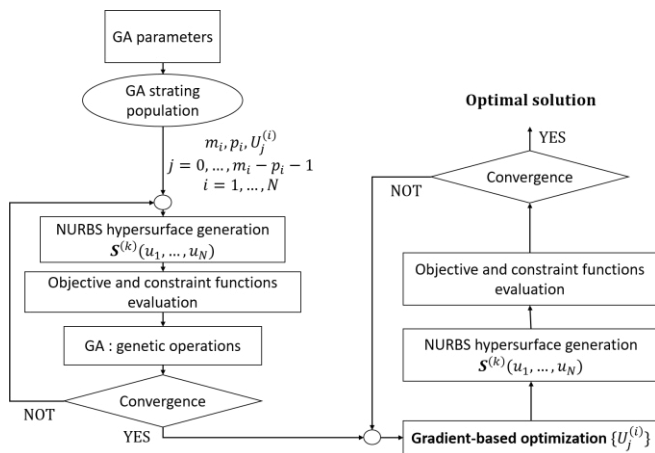


Fig. 4. Hybrid optimization algorithm.

In agreement with the paradigms of natural sciences, individuals characterized by a different number of chromosomes (i.e., modular structures composed of a different number of modules) belong to different species. BIANCA has

been conceived for crossing also different species, thus making possible (and without distinction) the *simultaneous optimization of species and of individuals*. This task can be attained thanks to some special genetic operators that have been implemented to perform the reproduction phase between individuals belonging to different species, see Fig. 5. For more details on the original features of such a GA the interested reader is addressed to [13].

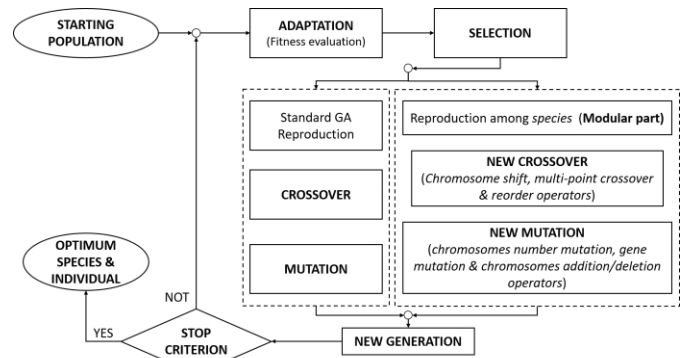


Fig. 5. BIANCA algorithm [13].

Due to the strong nonlinearity of problem (12), the aim of the genetic calculation is to provide a potential sub-optimal point in the design space, which constitutes the initial guess for the subsequent phase, i.e., the local optimization performed via the *fmincon* gradient-based algorithm. During this second phase, only the components of the knot vector along each dimension are considered as design variables, see Fig. 4.

## 4. Numerical results

### 4.1. Results with iterative method

In this first case, the hypersurface fitting problem (12) has been solved by using the iterative procedure proposed in [14]. The algorithm presented in [14] tries to solve problem (12) by using, during the first iterations, few CPs along each dimension. Then, if the stop criterion is not met (i.e. the cost function is not lower than or equal to a certain threshold value defined by the user) the algorithm repeats the fitting process by adding CPs in the direction where the ratio  $\rho_i$  between the number of CPs  $n_i$  and the number of TPs  $r_i$  is the lowest. In this background, knot vectors components are computed in a manner that ensures every knot of the  $\mathbf{U}^{(i)}$  contains at least one  $u_i^{(k)}$ , dimensionless parameters are computed according to *chord length* method [14] and the degrees are set equal to two, see [6,7] for more details. Table 1 lists the number of both TPs  $r_i$  and CPs  $n_i$  and the internal knot values  $\{U_j^{(i)}\}$  obtained via the iterative method. As it can be seen from this table, the average error is  $8.881917e^{-05}$ . The overall number of TPs utilized in this example is  $n_Q = 4498641$  while the overall number of CPs at the end of the procedure is  $n_{CP} = 4096$ .

Table 1. Results of iterative procedure (precision  $1.e^{-04}$ ).

Variable	x	y	Th	h
$r_i$	101	101	21	21

$n_i$	16	16	4	4
$U_1 = U_2$	{0.000000; 0.000000; 0.000000; 0.057333; 0.124667; 0.192000; 0.259333; 0.326667; 0.394000; 0.461333; 0.528667; 0.596000; 0.663333; 0.730667; 0.798000; 0.865333; 0.932667; 1.000000; 1.000000}			
$U_3 = U_4$	{0.000000; 0.000000; 0.000000; 0.300000; 0.650000; 1.000000; 1.000000; 1.000000}			
Average error:	8.881917e <sup>-05</sup>			
Max error:	7.080100e <sup>-03</sup>			

#### 4.2. Results with the hybrid optimization tool and discussion

Here the results of the hybrid optimization strategy are presented. At the end of the optimisation process several “equivalent” optimum solutions have been found. This means that the problem of finding a suitable B-Spline hypersurface fitting the (convex) set of TPs becomes a non-convex problem when integrating the degree and the number of knots (along each direction) among the design variables. Indeed, at the end of the optimisation process, within the final population one can find solutions minimising the number of B-Spline hypersurface parameters and characterised by an excellent accuracy level.

However, in order to highlight the (significant) influence of the value of the knot vectors components on the value of the objective function, it has been chosen to present a result characterised by the same value of both the degrees and the number of knot vectors components as those provided by the iterative procedure explained in the previous section. Table 2 lists the optimum value of the knot vector components (along each dimension) resulting from the optimization process. In this case, the values of the knot vectors components are quite different. Accordingly, the value of the objective function (i.e., the average error) is about an order of magnitude lower than that resulting from the iterative procedure. To attain the same level of accuracy, when using the iterative procedure, the number of CPs along each dimension must be increased as follows:  $n_1 = n_2 = 36$  and  $n_3 = n_4 = 8$ , leading in this way to an overall number of 82944 CPs (i.e. 20 times those provided by the optimization strategy).

Table 2. Results of *fmincon* function optimization.

Vari able	x	y	th	h
$n_i$	16	16	4	4
$U_1$	{0.000000; 0.000000; 0.000000; 0,032466; 0,067869; 0,143467; 0,2892994; 0,6519176; 0,815020; 0,908234; 0,969446; 0,979813; 0,983679; 0,995262; 0,997999; 0,999455; 0,999455; 1.000000; 1.000000}			
$U_2$	{0.000000; 0.000000; 0.000000; 0,057281; 0,124735; 0,191993; 0,259237; 0,326633; 0,394234; 0,461419; 0,528678; 0,595977; 0,663156; 0,730557; 0,797903; 0,865612; 0,932640; 1.000000; 1.000000}			
$U_3$	{0.000000; 0.000000; 0.000000; 0,137774; 0,442168; 1.000000; 1.000000}			
$U_4$	{0.000000; 0.000000; 0.000000; 0,016338; 0,249257; 1.000000; 1.000000; 1.000000}			
Average error:	2.203175e <sup>-06</sup>			
Max error:	1.654188e <sup>-04</sup>			

## 5. Conclusions

A new metamodeling technique based on NURBS hypersurface formalism together with a hybrid optimization strategy has been presented in this paper. The originality of this approach is characterized by the fact that nor simplifying hypotheses neither empirical rules are utilized *a priori*. The proposed approach aims at minimizing the number of parameters defining the NURBS surface by ensuring, at the same time, a good level of accuracy (at least equal to those provided by classical iterative procedures). To achieve this result, the solution search for the hypersurface fitting problem is realized by means of a hybrid optimization tool of which the kernel is a special genetic algorithm able to deal with problems defined over a domain of variable dimension.

The effectiveness of this process has been proven through a meaningful benchmark and the obtained results are compared to those resulting from an iterative method taken from literature.

Further studies have to explore the influence of NURBS hypersurfaces parameters and the interest of using them instead of B-Spline ones. Moreover, the robustness and the effectiveness of the proposed approach must be validated through a real-world engineering problem, since first results are really promising.

## References

- [1] B. Yegnanarayana, Artificial Neural Networks, PHI Learning Pvt. Ltd., 2009.
- [2] V. Capecchi, M. Buscema, P. Contucci, B. D'Amore, eds., Applications of Mathematics in Models, Artificial Neural Networks and Arts, Springer Netherlands, Dordrecht, 2010.
- [3] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations, Springer International Publishing, Cham, 2014.
- [4] S.N. Sivanandam, S. Sumathi, S.N. Deepa, Introduction to fuzzy logic using MATLAB, Springer, Berlin, 2007.
- [5] M.D. Buhmann, Radial basis functions, Acta Numer. 2000.
- [6] C.J. Turner, HyPerModels: hyperdimensional performance models for engineering design, 2005.
- [7] C.J. Turner, R.H. Crawford, N-Dimensional Nonuniform Rational B-Splines for Metamodeling, J. Comput. Inf. Sci. Eng. 9 (2009) 031002.
- [8] J. Steuben, J. Michopoulos, A. Iliopoulos, C. Turner, Inverse characterization of composite materials via surrogate modeling, Compos. Struct. 132 (2015) 694–708. doi:10.1016/j.compstruct.2015.05.029.
- [9] G. Costa, M. Montemurro, J. Pailhès, A General Hybrid Optimization Strategy for Curve Fitting in the Non-uniform Rational Basis Spline Framework, J. Optim. Theory Appl. (2017) 1–27
- [10] G. Costa, M. Montemurro, J. Pailhès, A 2D topology optimisation algorithm in NURBS framework with geometric constraints, Int. J. Mech. Mater. Des. (2017) 1–28. doi:10.1007/s10999-017-9396-z.
- [11] M. Montemurro, A. Catapano, A new paradigm for the optimum design of variable angle tow laminates., Var. Anal. Aerosp. Eng. Math. Chall. Aerosp. Future Springer Optim. Its Appl. 116 (2016).
- [12] M. Montemurro, A. Catapano, On the effective integration of manufacturability constraints within the multi-scale methodology for designing variable angle-tow laminates., Compos. Struct. 161 (2017) 145–159.
- [13] M. Montemurro, Optimal design of advanced engineering modular systems through a new genetic approach., Université Pierre et Marie Curie Paris VI, 2012.
- [14] L. Piegl, W. Tiller, The NURBS Book, Springer-Verlag Berlin Heidelberg, 1995.
- [15] The Mathworks Inc., Optimization Toolbox User's Guide., (2011).