A General Hybrid Optimization Strategy for Curve Fitting in the Non-Uniform Rational Basis Spline Framework

Giulio Costa
Marco Montemurro*
Jérôme Pailhès
Arts et Métiers ParisTech, I2M CNRS UMR 5295, F-33400 Talence, France.

This is a pre-print of an article published in *Journal of Optimization Theory and Applications*.
The final authenticated version is available online at: https://doi.org/10.1007/s10957-017-1192-2

Corresponding author:
Marco Montemurro, PhD,
Arts et Métiers ParisTech,
I2M CNRS UMR 5295,
F-33400 Talence, France,
tel: +33 55 68 45 422,
fax: +33 54 00 06 964,
e.mail: marco.montemurro@ensam.eu,
marco.montemurro@u-bordeaux1.fr

*Corresponding author.

Abstract

In this paper, a general methodology to approximate sets of data points through Non-Uniform Rational Basis Spline curves is provided. The proposed approach aims at integrating and optimizing the full set of design variables (both integer and continuous) defining the shape of the Non-Uniform Rational Basis Spline curve. To this purpose, a new formulation of the curve fitting problem is required: it is stated in the form of a Constrained Non-Linear Programming Problem by introducing a suitable constraint on the curvature of the curve. In addition, the resulting optimization problem is defined over a domain having variable dimension, wherein both the number and the value of the design variables are optimized. To deal with this class of Constrained Non-Linear Programming Problems, a global optimization hybrid tool has been employed. The optimization procedure is split in two steps: firstly, an improved genetic algorithm optimizes both the value and the number of design variables by means of a two-level Darwinian strategy allowing the simultaneous evolution of individuals and species; secondly, the optimum solution provided by the genetic algorithm constitutes the initial guess for the subsequent gradient-based optimization, which aims at improving the accuracy of the fitting curve. The effectiveness of the proposed methodology is proven through some mathematical benchmarks as well as a real-world engineering problem.

Keywords:

NURBS curves; Curve Fitting; Genetic Algorithms; Reverse Engineering; Modular Systems; Optimization

1 Introduction

Curve fitting is a widely studied topic in informatics, geometric modelling and reverse engineering. The goal is to find all the parameters which uniquely identify a parametric curve approximating a set of data points, i.e. the *target points* (TPs). The curve fitting problem can be stated as a classical least squares problem wherein the Euclidean distance between TPs and a set of suitable points belonging to the curve is minimized. Standard gradient optimization methods have been broadly employed in order to solve the curve fitting problem [1, 2, 3]. In particular, in [1] and [3], the formulation of the objective function was modified by introducing the *tangent distance minimization method* and the *square distance minimization method*. The most relevant contribution of these techniques is on the improvement of the convergence rate and the stability of the solution. Ueng *et al.* [2] enhance the objective function by inserting information about tangent and curvature of the approximating curve as weighted quantities. However, weight parameters must be carefully tuned a-priori by the designer in [2]: accordingly, their definition is problem-dependent.

Several methodologies deal with the curve fitting problem in the framework of Non-Uniform Rational Basis Spline (NURBS). A NURBS curve is defined by the degree of the blending functions, the number and the coordinates of control points, the knot vector components and the weight values [4]. This large amount of parameters makes NURBS curves and surfaces a very versatile and interesting tool for many mathematical and engineering applications, not only for the curve

fitting problem. Performing a curve fitting by means of a NURBS curve is particularly advantageous because this geometric entity is completely CAD-compatible, i.e. its parameters can be transferred through standard format files to CAD software: in fact, NURBS constitute one of the milestones of CAD design and they are widely utilized for reverse engineering problems. However, the considerable quantity of parameters defining a NURBS curve also constitutes the main drawback: it is very hard to properly tune all parameters defining the shape of a NURBS curve. In the last three decades, the massive development of metaheuristic procedures has brought engineers to apply such strategies in the framework of the curve and/or surface fitting problem. As well known, the most significant advantages of metaheuristics are the abilities of dealing with large set of data and of exploiting the related information to effectively explore the search space, in order to find the global minimum. The main drawback is the high computational time. Conversely, in the case of gradient-based strategies, the major drawbacks are related, on the one hand, to the need of an initial guess for the set of parameters describing the curve shape and, on the other hand, to the possibility of falling on a local minimum. To overcome the latter drawback, Li *et al.* [5] present a preprocessing method, based on the discrete evaluation of the curvature, to provide a starting Basis Spline (referred as Bspline in the following) knot vector which reflects the shape of the curve to be approximated. Concerning the utilization of metaheuristics for solving the curve/surface fitting problem, Limaiem *et al.* [6] make use of a genetic algorithm (GA) to find the optimum value of the parameters defining the approximating curve. In [7], a particle swarm optimization (PSO) algorithm has been employed to approximate the TPs by means of NURBS surfaces. Kang *et al.* [8] use a sparse optimization to iteratively update the knot vector length and components of the approximating BSpline. Furthermore, even if conceived for the problem of surface fitting through NURBS surfaces, interesting suggestions are provided in [9], where some stability requirements are imposed on the final position of control points. Recently, Garcia-Capulin *et al.* [10] employed a Hierarchical GA to optimize both the number and the value of the knots of a Bspline curve. However, the approach presented in [10] is based on the resolution of a bi-objective unconstrained optimization problem that needs the definition of a "fictitious" objective function to economize the number of knots, which is not related to any geometrical requirement. Moreover, the degree of the basis functions is kept constant in [10] and the problem is not stated in the more general framework of NURBS curves.

As it can be easily deduced from this (non-exhaustive) state of the art on curve fitting in the mathematical framework of NURBS representation, the main limitations and drawbacks characterising the vast majority of the studies on this topic are essentially two:

- the lack of a proper problem formulation (without considering arbitrary penalization coefficients, which must be defined by the user and that are problem-dependent);
- the lack of a very general numerical strategy, able to simultaneously optimize the *number* as well as the *value* of the constitutive parameters (i.e. the *design variables*) defining the shape

of the NURBS curve.

To overcome the previous restrictions, in this work, an innovative approach to the curve fitting problem is proposed. A new formulation of the mathematical problem has been developed: the curve fitting problem is stated as a Constrained Non-Linear Programming Problem (CNLPP) by introducing a constraint on the maximum value of the curvature.

In this study, the curve fitting problem is solved in the framework of NURBS curves. The main idea is to keep all the parameters defining the NURBS curve as design variables in order to state the curve fitting problem in the most general sense. Nevertheless, this fact implies some consequences of paramount importance, constituting just as many difficulties in solving the related CNLPP.

- When the curve fitting problem is formulated by including the number of control points and the degree of the basis functions among the unknowns, the *overall number* of design variables (i.e. the overall number of parameters defining the shape of the curve) for the problem at hand is not fixed *a-priori*: hence, the resulting CNLPP is defined over a search space of variable dimension.
- The optimization variables of the CNLPP are of different nature (continuous and discrete).
- The numerical strategy chosen to face such a problem must be able to handle design variables of different nature and to optimize, at the same time, the dimension of the design domain as well as the value of each constitutive parameter of the NURBS curve.

This kind of problems is referred as optimization of "modular systems" in bibliography, see [11]. Here, the numerical strategy considered for the solution search of CNLPP of modular systems is based on an improved GA [11, 12, 13], able of dealing with optimization problems with "variable number of design variables".

The paper is organized as follows: the general theoretical framework of NURBS curves is briefly discussed in section 2. In section 3, the new formulation for the curve fitting problem is introduced: the problem variables are highlighted and the objective function is carefully explained, together with the optimization constraint. Section 4 focuses on the main features of the considered numerical strategy, whilst the numerical results are presented and discussed in section 5. Finally, section 6 ends the paper with some conclusive remarks and perspectives.

2 Theoretical Framework

In this section, the fundamentals of the NURBS curves theory are briefly recalled. According to the notation introduced in [4], the parametric implicit form of a NURBS curve is:

$$\mathbf{C}(u) = \sum_{i=0}^{n} R_{i,p}(u)\mathbf{P}_{\mathrm{i}}, \tag{1}$$

where $\mathbf{C}(u) = \{x(u), y(u), z(u)\}$ are the Cartesian coordinates of the curve, whilst $R_{i,p}(u)$ is the generic rational basis function having the form

$$R_{i,p}(u) = \frac{N_{i,p}(u)w_i}{\sum_{j=0}^{n} N_{j,p}(u)w_j}. \tag{2}$$

In Eqs. (1) and (2), u is a dimensionless parameter defined in the range $[0, 1]$, $N_{i,p}(u)$ are the basis functions, recursively defined according to Bernstein polynomials, p is the maximum degree, w_i are the weights and $\mathbf{P}_\mathrm{i} = \{x_i, y_i, z_i\}$ the Cartesian coordinates of the control points. The set of the $(n+1)$ control points form the so-called *control polygon*. The blending functions $N_{i,p}(u)$ are defined as

$$N_{i,0}(u) = \begin{cases} 1, \text{ if } U_i \leq u < U_{i+1}, \\ 0, \text{ otherwise}, \end{cases} \tag{3}$$

$$N_{i,q}(u) = \frac{u - U_i}{U_{i+q} - U_i} N_{i,q-1}(u) + \frac{U_{i+q+1} - u}{U_{i+q+1} - U_{i+1}} N_{i+1,q-1}(u), \ q = 1, ..., p, \tag{4}$$

where U_i is the i-th component of the following non-periodic non-uniform *knot vector*:

$$\mathbf{U} = \{\underbrace{0, \ldots, 0}_{p+1}, U_{p+1}, \ldots, U_{m-p-1}, \underbrace{1, \ldots, 1}_{p+1}\}. \tag{5}$$

It is noteworthy that the size of the knot vector is $m + 1$,

$$m = n + p + 1. \tag{6}$$

The knot vector is a non-decreasing sequence of real numbers that can be interpreted as a discrete collection of values of the dimensionless parameter u splitting the curve in arcs. The components of $\mathbf{U}$ are called *knots* and each knot can have a multiplicity λ. One basic property of a NURBS curve is related to the continuity and differentiability of the basis function $N_{i,p}(u)$ at a knot: it is $p - \lambda$ times continuously differentiable. Thus, increasing the degree increases the continuity, whilst increasing the knot multiplicity decreases the continuity. It is evident that the knot vector strongly affects the basis functions and, accordingly, the shape of a NURBS curve. For a deeper insight in the matter, the reader is addressed to [4].

3 Mathematical Formulation of the Curve Fitting Problem

In this section, the curve fitting problem is stated as a CNLPP and it is formulated in the most general case, i.e. by considering the full-set of design variables describing the shape of the parametric curve.

Let us consider the classical form of the curve fitting problem, namely

$$\min_{\mathbf{x}} f(\mathbf{x}), \ f = \sum_{k=0}^{\mu} \|\mathbf{C}(u_k) - \mathbf{Q}_k\|^2. \tag{7}$$

In Eq. (7), $(\mu+1)$ is the number of TPs, $\mathbf{Q}_k$ the generic k-th point, $\mathbf{Q}_k = \{\overline{x}_k, \overline{y}_k, \overline{z}_k\}$ are the Cartesian coordinates of the TPs, while $\mathbf{C}(u_k) = \{C_x(u_k),$
$C_y(u_k), C_z(u_k)\}$ are their counterpart belonging to the parametric curve when the dimensionless parameter u gets the value u_k. In the same equation, vector $\mathbf{x}$ collects all the optimization variables, i.e. the full set of parameters (of different nature) defining the shape of the curve. In the most general case, when the parametric curve of Eq. (7) is represented in the mathematical framework of NURBS basis functions, its shape depends upon the following parameters:

- *integer parameters*, i.e. the number of control points $n+1$, the number of knots $m+1$ and the degree of the blending functions p;
- *continuous parameters*, namely the non-decreasing sequence of components of the knot vector $U_j,\ j \in [p+1, m-p-1]$, the coordinates of the control points $\mathbf{P}_\mathrm{i} = \{x_i, y_i, z_i\},\ i \in [0, n]$, the weights values $w_i,\ i \in [0, n]$ and the set of suitable values of the dimensionless parameter of the curve $u_k,\ k \in [0, \mu]$.

Firstly, let us consider the integer parameters: Eq. (6) gives the relationship among m, p and n. In standard approaches [1, 2, 3, 5], the maximum control point index n is fixed *a-priori*, while the value of p is chosen as compromise between accuracy and noise introduction. Then, the maximum index of the knot vector components is deduced accordingly. Unlike standard approaches, no assumptions are made on the integer parameters of a NURBS curve in this work. In particular, m and p are included into the vector of design variables, whilst n will be calculated according to Eq. (6).

Secondly, let us consider the set of continuous parameters. The u_k values of the curve dimensionless parameters are calculated through the *chord length method* [4], so they are no longer design variables. In particular, the *chord length* L_{TP} of the curve can be defined in terms of Euclidean distance among consecutive TPs,

$$L_{TP} = \sum_{k=0}^{\mu-1} \|\mathbf{Q}_{k+1} - \mathbf{Q}_k\|. \tag{8}$$

Assumed that $u_0 = 0$ and $u_\mu = 1$, the general parameter u_k can be computed through

$$u_{k+1} = u_k + \frac{\|\mathbf{Q}_{k+1} - \mathbf{Q}_k\|}{L_{TP}},\ k = 0, ..., \mu-2. \tag{9}$$

For more details on the chord length method, the interested reader is addressed to [4].

Moreover, the optimum value of the control points coordinates can be obtained through the analytical approach of Ueng *et al.* [2]. Let $\mathbf{X}_P, \mathbf{Y}_P, \mathbf{Z}_P \in \mathbb{R}^{n+1}$ be column vectors collecting the x, y and z coordinates of the control points and $\overline{\mathbf{X}}_Q, \overline{\mathbf{Y}}_Q, \overline{\mathbf{Z}}_Q \in \mathbb{R}^{\mu+1}$ the counterparts for TPs. Furthermore, matrix $[\mathbf{A}] \in \mathbb{R}^{(\mu+1)\times(n+1)}$ can be defined as

$$A_{k,i} = R_{i,p}(u_k),\ k = 0, ..., \mu+1,\ i = 0, ..., n+1, \tag{10}$$

and matrix $[\mathbf{B}] \in \mathbb{R}^{(n+1)\times(n+1)}$ as

$$[\mathbf{B}] = \left([\mathbf{A}]^T[\mathbf{A}]\right)^{-1}. \tag{11}$$

Therefore, the following proposition applies.

Theorem 1 *For a NURBS curve of assigned degree p, number of control points $(n+1)$, knot vector $\mathbf{U}$ and weights w_i $(i = 0, ..., n)$, the control point coordinates minimising the cost function f of problem (7) are*

$$\mathbf{X}_P = [\mathbf{B}][\mathbf{A}]^T \overline{\mathbf{X}}_Q, \ \mathbf{Y}_P = [\mathbf{B}][\mathbf{A}]^T \overline{\mathbf{Y}}_Q, \ \mathbf{Z}_P = [\mathbf{B}][\mathbf{A}]^T \overline{\mathbf{Z}}_Q. \tag{12}$$

Proof. The proof is provided here for the coordinate x and can be easily extended to other coordinates. Since the objective function f is convex (in terms of control points coordinates), a necessary and sufficient condition for getting the minimum is

$$\frac{\partial f}{\partial x_l} = 0, \forall l = 0, ..., n. \tag{13}$$

After few simple passages, the previous relationship can be written as

$$\begin{aligned} &\textstyle\sum_{k=0}^{\mu} \left[2 \left(C_x(u_k) - \overline{x}_k \right) \frac{\partial C_x(u_k)}{\partial x_l} \right] = 0, \\ &\textstyle\sum_{k=0}^{\mu} \left[\left(\sum_{i=0}^{n} R_{i,p}(u_k) x_i - \overline{x}_k \right) R_{l,p}(u_k) \right] = 0, \\ &\textstyle\sum_{k=0}^{\mu} \sum_{i=0}^{n} R_{l,p}(u_k) R_{i,p}(u_k) x_i = \sum_{k=0}^{\mu} R_{l,p}(u_k) \overline{x}_k, \forall l = 0, ..., n \end{aligned} \tag{14}$$

The last relation of Eq. (14) must be satisfied for each control point and can be stated in a more compact form:

$$[\mathbf{A}]^T [\mathbf{A}] \mathbf{X}_P = [\mathbf{A}]^T \overline{\mathbf{X}}_Q. \tag{15}$$

Finally, the inversion of matrix $\left([\mathbf{A}]^T [\mathbf{A}] \right)$ allows for obtaining the vector $\mathbf{X}_P$. ∎

It is noteworthy that matrix $\left([\mathbf{A}]^T [\mathbf{A}] \right)$ could have some almost null eigenvalue, so its inversion could be ill-conditioned. In this paper, the inversion has been performed by means of Moore-PenroseâĂŹs pseudo-inverse matrix [2], in order to overcome this issue.

A quick glance to Eqs. (10)-(12) suffices to deduce that the Cartesian coordinates of the control points are affected by the other parameters of the NURBS curve, so they are no longer design variables but they can be interpreted as derived quantities. More precisely, matrix $[\mathbf{A}]$ depends upon the NURBS blending functions, hence its terms depend on the value of both integer and continuous variables, i.e. m, p, U_j and w_i, as well as on the u_k values. As a consequence of the previous considerations, design variables can be ranged in two vectors $\boldsymbol{\xi_1}$ and $\boldsymbol{\xi_2}$:

- $\boldsymbol{\xi_1}$ collects the *integer variables*, i.e. the knot vector maximum index m and the curve degree p;
- $\boldsymbol{\xi_2}$ collects *continuous variables*, i.e. the knot vector non-trivial components U_j and the weights w_i.

Mathematically speaking, vectors $\boldsymbol{\xi_1}$ and $\boldsymbol{\xi_2}$ are represented as

$$\boldsymbol{\xi_1} = \{m, p\} \in \mathbb{N}^2, \tag{16}$$

$$\boldsymbol{\xi_2} = \{U_{p+1}, \ldots, U_{m-p-1}, w_0, \ldots, w_{m-p-1}\} \in \mathbb{R}^{N_v}, \tag{17}$$

where

$$N_v = 2m - 3p - 1. \tag{18}$$

$(N_v + 2)$ is the overall number of design variables.

As previously stated, in this work, the curve approximation problem is still framed as an optimization problem, but a more general formulation is introduced. On the one hand, the objective function has been modified with respect to Eq. (7), namely:

$$\min_{\boldsymbol{\xi_1}, \boldsymbol{\xi_2}} \Phi(\boldsymbol{\xi_1}, \boldsymbol{\xi_2}) = \min_{\boldsymbol{\xi_1}, \boldsymbol{\xi_2}} \left[\frac{\sum_{k=0}^{\mu} \|\mathbf{C}(u_k) - \mathbf{Q}_k\|^2}{L_{TP}^2} \right]^{1/m}. \tag{19}$$

In Eq. (19), the parameter $1/m$ appears as power of the sum of squares of Euclidean distances divided by the square of *chord length* of the curve L_{TP}, refer to Eq. (8). On the other hand, an optimization constraint on the maximum radius of curvature of the NURBS curve is introduced: in real-world engineering problems, such a requirement is often imposed to improve the smoothness of the approximating curve. This constraint can be stated as:

$$g(\boldsymbol{\xi_1}, \boldsymbol{\xi_2}) = \frac{\chi_{max} - \chi_{adm}}{\chi_{adm}}, \tag{20}$$

with

$$\chi_{max} = \max_u \chi(u), \tag{21}$$

$$\chi(u) = \frac{\|\mathbf{C}'(u) \wedge \mathbf{C}''(u)\|}{\|\mathbf{C}'(u)\|^3}. \tag{22}$$

In Eq. (20), χ_{adm} is the admissible value for the curvature which must be established according to the problem at hand. It should be noticed that the purpose of the constraint on the maximum curvature of the NURBS curve is twofold: on the one hand, it constitutes a precise technological requirement that affects the final shape of the curve; on the other hand, it allows for defining a well-posed mathematical problem, because it limits the growth of the degree p of the blending functions during optimization.

Remark In order to understand the latter assertion, let us consider a very simple parametric curve γ in the $x - y$ plane, namely,

$$x(t) = t, \; y(t) = t^p, \tag{23}$$

For this case, the curvature $\chi_\gamma(t)$ writes

$$\chi_\gamma(t) = \frac{|p(p-1)t^{p-2}|}{\left(1 + p^2 t^{2(p-1)}\right)^{3/2}}. \tag{24}$$

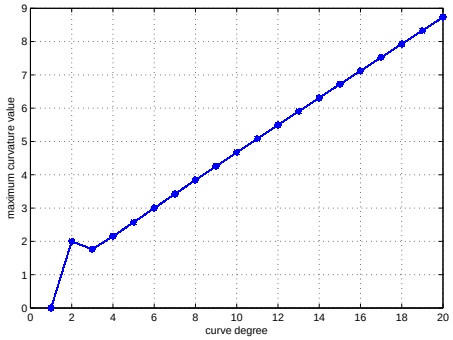


Figure 1: Trend $\chi_{\gamma max}$ vs p for the curve γ

Of course, $\chi_\gamma(t)$ depends upon the local abscissa t as well as on the curve degree p. The maximum value of $\chi_\gamma(t)$ can be calculated for different values of p. The result of such a computation is synthetically illustrated in Fig. 1. As it can be deduced from Fig. 1, increasing the degree implies a higher value of the maximum curvature value for a simple polynomial curve as γ. Being the NURBS curves defined through special polynomial-based blending functions, intuitively it can be stated that imposing a constraint on the maximum curvature value means also limiting the maximum curve degree.

Finally, the curve fitting problem can be stated in the standard form of a CNLPP of modular systems [11] as follows:

$$\begin{aligned} &\min_{\boldsymbol{\xi_1},\boldsymbol{\xi_2}} \quad \Phi\left(\boldsymbol{\xi_1},\boldsymbol{\xi_2}\right), \\ &\text{subject to:} \\ &\begin{cases} g(\boldsymbol{\xi_1},\boldsymbol{\xi_2}) \leq 0, \\ \boldsymbol{\xi}_{1-lb} \leq \boldsymbol{\xi_1} \leq \boldsymbol{\xi}_{1-ub}, \boldsymbol{\xi_1} \in \mathbb{N}^2, \\ \boldsymbol{\xi}_{2-lb} \leq \boldsymbol{\xi_2} \leq \boldsymbol{\xi}_{2-ub}, \boldsymbol{\xi_2} \in \mathbb{R}^{N_v}. \end{cases} \end{aligned} \tag{25}$$

In Eq. (25), $\boldsymbol{\xi}_{i-lb}$ and $\boldsymbol{\xi}_{i-ub}$ $(i = 1, 2)$ represent the lower and upper bounds, respectively, of the vector $\boldsymbol{\xi}_i$.

Remark To the best of the authors' knowledge, no analytical solutions are available in literature for problem (25). This is essentially due to the following difficulties.

- The problem aims at optimizing both discrete and continuous variables: pure gradient-based methods are automatically discarded and hybrid strategy must be considered.
- Since the dimension of the continuous design variables vector $\boldsymbol{\xi_2}$ depends on the discrete design variables collected in $\boldsymbol{\xi_1}$, problem (25) is stated on a domain having variable dimension, see Eqs. (16), (17) and (18). To the best of the authors' knowledge, pure gradient-based methods are not able to provide the solution in such cases.
- When considering the full set of design variables, both the objective and the curvature constraint functions become non-linear and non-convex.

Since the solution cannot be provided in a closed form, an approximate, i.e. pseudo-optimal, solution of problem (25) can be found by making use of a suitable meta-heuristic (a genetic algorithm) combined with a classic gradient-based method. The problem formulation (25) together with the special features of the proposed algorithm (see section 4) allows for determining a pseudo-optimal feasible solution.

Furthermore, the unusual form of objective function (19) allows the algorithm to automatically determine the best compromise between the number of knot vector components (and implicitly the number of design variables) and the precision of the solution. Let consider Eq. (19): assume $\varphi = \frac{\sum_{k=0}^{\mu} \|\mathbf{C}(u_k) - \mathbf{Q}_k\|^2}{L_{TP}^2}$. During the first iterations, it could happen either $\varphi > 1$ or $\varphi < 1$ if the least square distance is greater or smaller than L_{TP}, respectively. If $\varphi > 1$, the number of knot vector components is encouraged to quickly grow in order to minimize the overall objective function. Consequently, in the next iterations, the algorithm will tend towards a solution with $\varphi < 1$. So, after a certain number of iterations, the case $\varphi < 1$ will become predominant and, from that moment, increasing the number of knot vector components will not necessarily imply better performances: in fact, increasing the parameter m means getting a lower value of $\varphi < 1$ but, meanwhile, a decreasing exponent $1/m$. Therefore, the best value of m will be determined as a result of the compromise between these two contrasting effects.

4 Numerical Strategy

Considering the mathematical features of problem (25), a hybrid optimization tool composed of the new version of the GA BIANCA [13], interfaced with the MATLAB *fmincon* algorithm [14], has been developed, see Fig. 2.

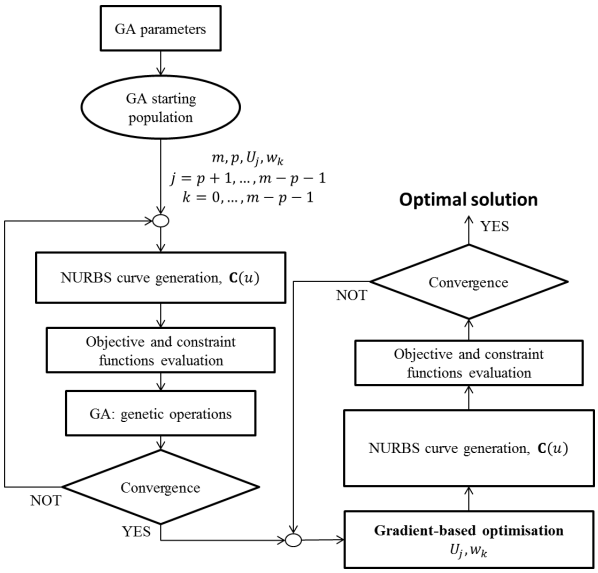


Figure 2: Overview of the global numerical strategy for the curve fitting problem

The GA BIANCA was already successfully applied to solve different kinds of real-world engineering problems, e.g. [15, 16, 17, 18, 19, 20]. As shown in Fig. 2, the optimization procedure for problem (25) 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 CNLPPs 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. Indeed, unlike the vast majority of GAs reported in literature (which are often characterized by a mono-chromosome algebraic structure), in BIANCA the information is organized in a genome (or 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, each constitutive module is represented by a chromosome, while each gene composing a chromosome codes a design variable related to the module.

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. 3. Moreover, in BIANCA the information restrained in the population is exploited in order to allow for a deep mixing of the individual genotype: in fact, all the genetic operators act on every single gene of the individual, so allowing for a true independent evolution of each design variable. For more details on the GA BIANCA the reader is addressed to [13].

In this study, the improved version of the GA BIANCA has been recoded into the MATLAB

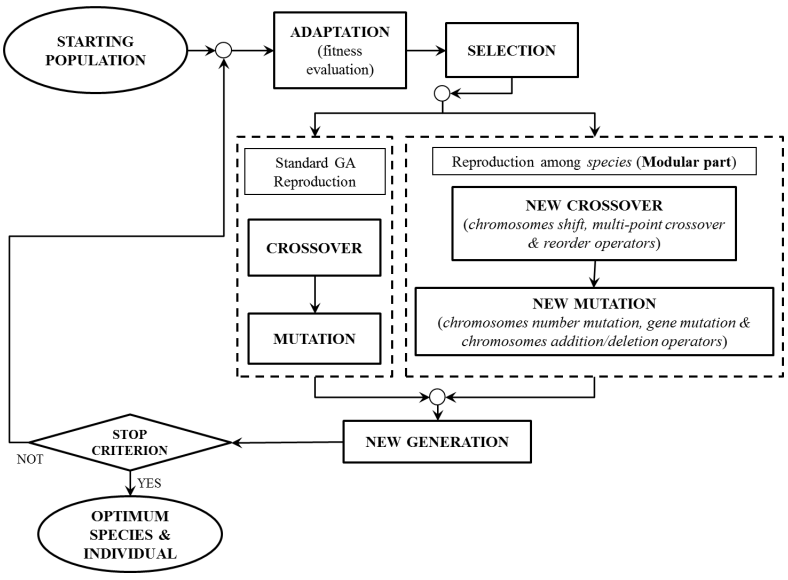


Figure 3: The genetic algorithm BIANCA: interactions of main operators

environment. Even though this choice penalizes the computational time, the utilization of the MATLAB version of the GA is easier when compared to the ancient FORTRAN version. In addition, thanks to the MATLAB structured variables, the architecture of the individual's genotype has been enriched and generalized as illustrated in Fig. 4. Without loss of generality, let N_m

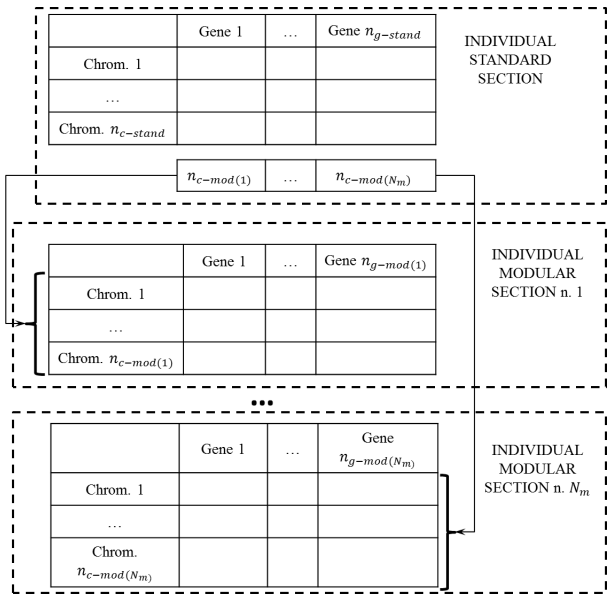


Figure 4: The general individual's structure for the MATLAB version of BIANCA

be the number of different types of modules for the problem at hand. Each individual (i.e. a point in the design space) is characterized by a genome composed of $N_m + 1$ sections having a precise hierarchy. The first section (i.e. the *standard section*) is linked to the non-modular part of the problem and its genotype is split in two parts: the first one is composed of a fixed number ($n_{c-stand}$) of chromosomes and each chromosome is made of $n_{g-stand}$ genes. The second part is composed of only one chromosome having N_m genes which can be related (or not) to the values of some genes of the first part. This first section undergoes the action of the standard GA operators, see Fig. 3. As shown in Fig. 4, each gene belonging to the mono-chromosome structure of the *standard section* is related to the number of modules $n_{c-mod(k)}$ of the generic k-th *modular section*, ($k = 1, , N_m$). Accordingly, each one of the remaining N_m modular sections is characterized by a genotype composed of $n_{c-mod(k)}$ chromosomes and $n_{g-mod(k)}$ genes. Of course, the reproduction between species by means of the new genetic operators [13] is allowed only on the modular sections. The structure of the individual's genotype for problem (25) is illustrated in Fig. 5. The first part of the *standard section* is characterized by one chromosome composed of two genes coding the design variables m and p, respectively. The second part of the *standard section* is constituted of a single chromosome with two genes coding the number of non-trivial components of the knot vector (the number of modules of the first type, i.e. $n_{c-mod(1)} = m - 2p - 1$) and the number of weights (the number of modules of the second type, i.e. $n_{c-mod(2)} = m - p$). Accordingly, the individual's genome possesses two modular sections: the first one is composed of $m - 2p - 1$ chromosomes with

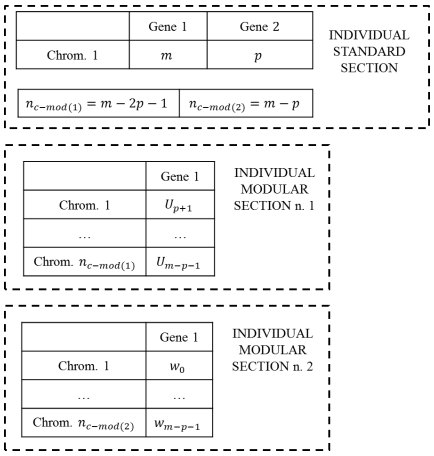


Figure 5: The individual's structure for the curve fitting problem

only one gene coding the value of the knot vector component U_j, while the second one is made of $m - p$ chromosomes with a single gene coding the value of the weight w_k in each control point. Due to the strong non-linearity of problem (25), 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, where the MATLAB *fmincon* gradient-based algorithm is employed to finalize the solution search. During this second phase only the components of the knot vector and the weights are considered as design variables, see Fig. 2.

5 Studied Cases and Results

In this section, some meaningful numerical examples are considered in order to prove the effectiveness of the proposed approach when dealing with the problem of the curve fitting. The set of genetic parameters tuning the behavior of the GA (for each case) is listed in Table 1.

Parameter	Value
Number of populations (N_{pop})	1
Number of individuals (N_{ind})	250
Number of generations (N_{gen})	320
Cross-over probability (p_{cross})	0.85
Gene mutation probability (p_{mut})	$1/N_{ind}$
Chromosome shift probability (p_{shift})	0.5
Chromosome number mutation probability ($p_{mut-chrom}$)	$(n_{ch_{ub}} - n_{ch_{lb}})/N_{ind}$
Selection Operator	Roulette wheel
Elitism Operator	Active

Table 1: Setting of genetic parameters

In addition, the handling of optimization constraints is carried out through the automatic dynamic penalization (ADP) technique, see [21]. It is noteworthy that the number of both individuals and generations are chosen to get $N_{ind} \times N_{gen} = 80000$ function evaluations (as it is usual in lit-

erature [13]) for each considered problem. Furthermore, Table 2 summarizes the design variables together with their bounds for problem (25).

Problem	p_{lb}	p_{ub}	m_{lb}	m_{ub}	U_{jlb}	U_{jub}	w_{ilb}	w_{iub}
The Descartes' folium	1	6	9	38	0	1	1	3
The four-leaf clover	1	8	8	67	0	1	1	3
The flame	1	8	100	130	0	1	1	3
The tennis ball stitching	1	8	8	67	0	1	1	3
The paddle curves	1	8	9	37	0	1	1	3

Table 2: Setting of variables boundaries.

As far as concerns the *fmincon* optimization tool employed for the local solution search at the end of the first step, the numerical algorithm chosen to carry out the calculations is the *active-set* method with non-linear constraints. For more details on the gradient-based approaches implemented into MATLAB, the reader is addressed to [14]. The numerical results, for each case, are collected in Table 3 and Table 4.

Curve	p	n	m	L_{TP}	$\Phi\left(\boldsymbol{\xi_1}, \boldsymbol{\xi_2}\right)$	$g(\boldsymbol{\xi_1}, \boldsymbol{\xi_2})$
Descartes'folium	4	15	20	3.01	0.4684	-7.00×10^{-2}
Four-leaf clover	5	33	39	7.75	0.7572	-6.00×10^{-4}
Flame	4	109	114	284.66	0.9232	-1.42×10^{-1}
Tennis ball stitching	6	39	46	33.78	0.6235	-1.76×10^{-2}
Paddle - c1	2	10	13	44.68	0.4522	-8.20×10^{-3}
Paddle - c2	3	7	11	58.85	0.3979	-1.00×10^{-3}
Paddle - c3	2	6	9	83.92	0.2981	-7.00×10^{-4}
Paddle - c4	4	9	14	99.63	0.4455	-8.91×10^{-2}
Paddle - c5	2	10	13	119.26	0.4059	-2.28×10^{-2}
Paddle - c6	5	8	14	130.33	0.4775	-4.70×10^{-3}
Paddle - c7	3	10	14	141.28	0.4229	-4.70×10^{-2}
Paddle - c8	3	10	14	129.94	0.4697	-5.95×10^{-2}
Paddle - c9	3	10	14	105.72	0.4285	-9.23×10^{-2}
Paddle - c10	2	11	14	40.37	0.6360	-4.68×10^{-2}
Paddle - t1	2	16	19	475.36	0.5552	-1.00×10^{-4}
Paddle - t2	2	14	17	548.43	0.5051	-1.00×10^{-4}

Table 3: Genetic Algorithm: Numerical Results.

Here, it is remarked that the objective function of the gradient based algorithm is provided by

$$\Phi_{grad}\left(\boldsymbol{\xi_2}\right) = L_{TP}^2 \Phi\left(\boldsymbol{\xi_1}, \boldsymbol{\xi_2}\right)^m, \tag{26}$$

that is the classic objective function for the curve fitting problem. It should be highlighted that the current objective function does not depend any more upon the discrete NURBS parameters: they have been optimized through the genetic step and they are kept constant in the gradient step.

Curve	$\mu+1$	$\Phi_{grad}\left(\boldsymbol{\xi_2}\right)$	$d_{average}$
Descartes'folium	50	1.60×10^{-6}	2.53×10^{-5}
Four-leaf clover	211	6.67×10^{-4}	1.23×10^{-4}
Flame	315	7.16×10^{-1}	2.69×10^{-3}
Tennis ball stitching	201	3.98×10^{-7}	3.14×10^{-6}
Paddle - c1	86	5.94×10^{-2}	2.83×10^{-3}
Paddle - c2	97	1.19×10^{-1}	3.56×10^{-3}
Paddle - c3	93	1.26×10^{-1}	3.82×10^{-3}
Paddle - c4	89	1.10×10^{-1}	3.72×10^{-3}
Paddle - c5	86	1.12×10^{-1}	3.89×10^{-3}
Paddle - c6	93	5.21×10^{-1}	7.76×10^{-3}
Paddle - c7	90	1.13×10^{-1}	3.73×10^{-3}
Paddle - c8	89	3.94×10^{-1}	7.05×10^{-3}
Paddle - c9	83	7.43×10^{-2}	3.28×10^{-3}
Paddle - c10	78	4.21×10^{0}	2.63×10^{-2}
Paddle - t1	89	5.40×10^{0}	2.61×10^{-2}
Paddle - t2	88	5.00×10^{0}	2.54×10^{-2}

Table 4: Gradient Algorithm: Numerical Results.

Finally, in Table 4, the quantity $d_{average}$ is defined as:

$$d_{average} = \frac{\Phi\left(\boldsymbol{\xi_2}\right)_{grad}^{1/2}}{\mu+1}, \tag{27}$$

which is an average distance between the TPs and the fitting curve, so $d_{average}$ gives an idea of the fairness of the method.

5.1 The Descartes' Folium

The Descartes' Folium is an open plane curve, whose parametric representation is

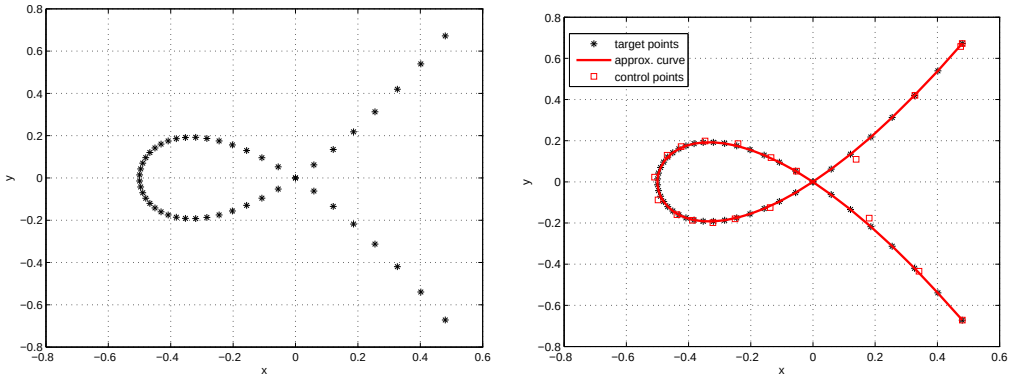
$$x(t) = at(t-1),\ y(t) = at(t-1)(2t-1). \tag{28}$$

The set of $\mu+1 = 50$ TPs is extracted from Eq. (28) by setting $a = 2$ and it is shown in Fig. 6a. As it can be seen from the graphic results (Fig. 6b), the presence of the loop does not affect the final quality of the approximating curve. From Table 3, it can be noticed that, due to the new form of the objective function and to the presence of the constraint on the maximum curvature, the optimum values of p and m are automatically determined by the GA because Eqs. (19) and (20) constitute implicit restrictions on both the degree of the basis functions and on the number of components of the knot vector.

5.2 The Four-Leaf Clover

The Four-Leaf Clover is a plane closed curve described by the parametric equation

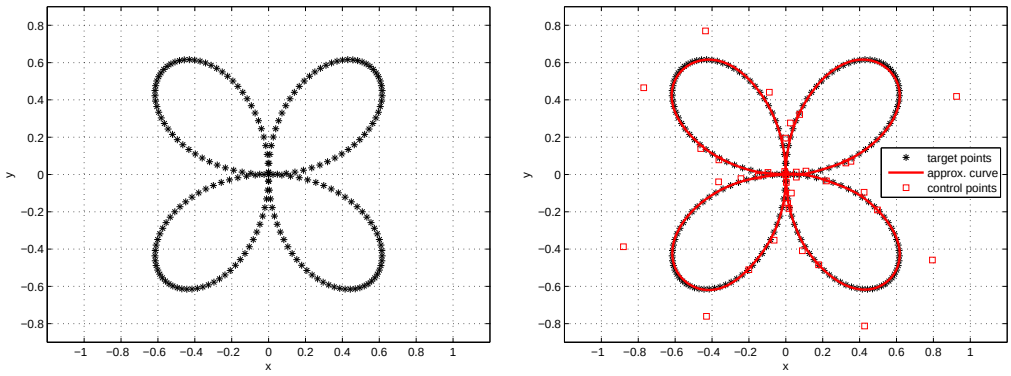
$$x(\theta) = cos(\theta)sin(2\theta),\ y(\theta) = sin(\theta)sin(2\theta). \tag{29}$$



(a) *Target points.* (b) *Approximating curve.*

Figure 6: The Descartes' Folium

In this case, $\mu+1 = 211$ TPs have been extracted from the previous equation. The optimum fitting curve is illustrated in Fig. 7b, while the related numerical results are listed in Table 3 and Table 4. Regarding the optimum value of p and m, the same considerations as those of example 5.1 can be repeated here.



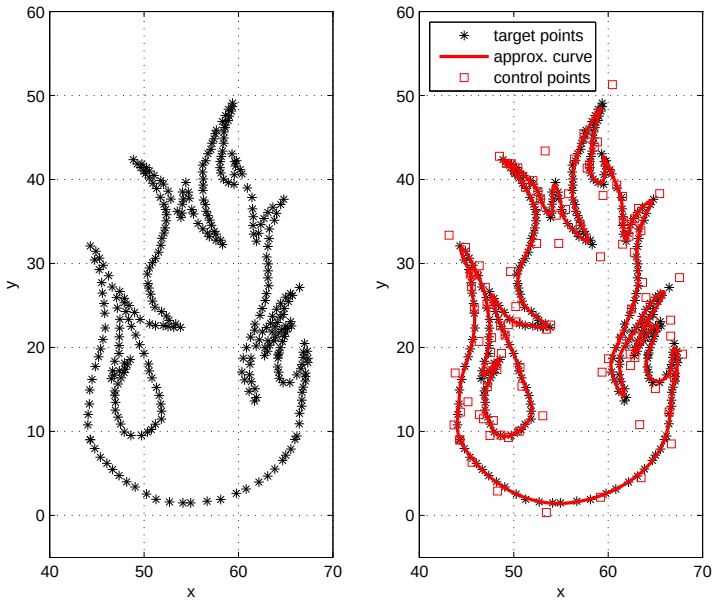
(a) *Target points.* (b) *Approximating curve.*

Figure 7: The four-leaf clover

5.3 The Flame

The third test case is a non-parametrized plane closed curve. 315 TPs have been sampled by the image of a flame, see Fig. 8a. This is a very challenging test case because of the complicated shape and the derivatives discontinuity. Indeed, the boundaries of the two first design variables have been broadened, in order to allow the curve to correctly evolve (see Table 2).

It must be pointed out that the constraint on the curvature is weaker than the previous cases, see Table 3: this is due, of course, to the presence of the cuspids. Only for this example, the resulting knot vector and weights are provided in Appendix to highlight the efficiency of the adopted strategy: some components are marked in bold font because they are very close, even the same. This fact reflects a well known NURBS property: if a knot has a multiplicity equal to λ, then the curve is $p - \lambda$ times continuously differentiable at the knot. As listed in Appendix,



(a) *Target points.* (b) *Approximating curve.*

Figure 8: The flame

the NURBS fitting curve is characterized by weights of different value: in particular, such weights get higher values for the control points located in the neighborhood of the cusps of the flame, see Fig. 9. However, all the weights values are close to the unity, which means that the cusps can be

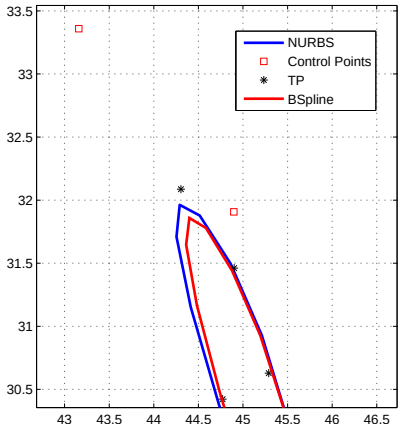


Figure 9: Detail on the NURBS approximating the flame

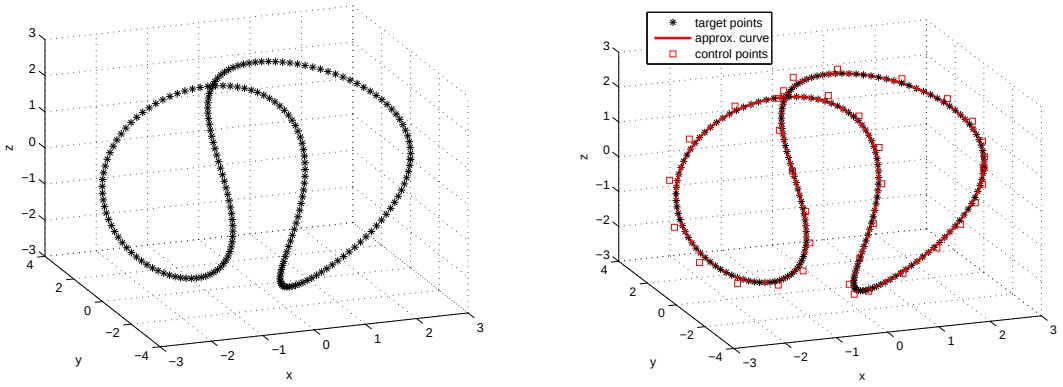
properly described through a smart choice of the knot vector components.

5.4 The Tennis Ball Stitching

The tennis ball stitching is a three-dimensional parametric curve. It has been chosen in order to provide a 3D test case for the curve fitting problem. The parametric form is:

$$x(t) = a\cos(t) + b\cos(3t),\ y(t) = a\sin(t) - b\sin(3t),\ z(t) = c\sin(2t). \tag{30}$$

The $\mu + 1 = 201$ TPs are extracted from Eq. (30) by setting $a = 2$, $b = 1$ and $c = 2\sqrt{2}$. The TPs as well as the optimum fitting curve are illustrated respectively in Fig. 10a and Fig. 10b.



(a) *Target points.* (b) *Approximating curve.*

Figure 10: The tennis ball stitching

5.5 The Paddle

In this subsection, a real-world engineering problem is faced. A paddle has been scanned and all points representing its external surface are shown in Fig. 11. Hence, twelve subsets of TPs have been extracted (see Fig. 12a): each set is supposed to constitute a primitive three-dimensional curve that will be employed during the CAD reconstruction of the paddle. For each curve, a

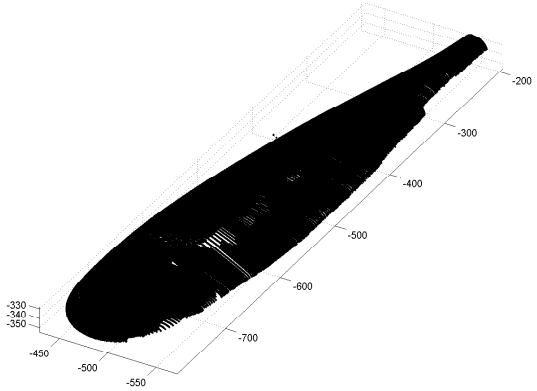


Figure 11: Starting data set for the paddle problem

technological constraint on the curvature has been considered, as shown in Table 5.

Here, the effectiveness of the presented method is remarked through this real-world engineering application, since a complicated set of scanned points can be easily treated and the resulting NURBS curves restrain all the necessary information in order to rebuild the paddle, by adding a technological constraint. Fig. 12b gives a global overview of the shape of the paddle primitive curves provided by the proposed optimization procedure.

Curve	χ_{adm}
Descartes'folium	7.0000
Four-leaf clover	6.0000
Flame	70.0000
Tennis ball stitching	0.5500
Paddle - c1	0.2000
Paddle - c2	0.2000
Paddle - c3	0.1700
Paddle - c4	0.9000
Paddle - c5	0.9000
Paddle - c6	1.0000
Paddle - c7	0.2500
Paddle - c8	0.9000
Paddle - c9	0.8000
Paddle - c10	0.5500
Paddle - t1	0.0150
Paddle - t2	0.0120

Table 5: Maximum allowed curvature values

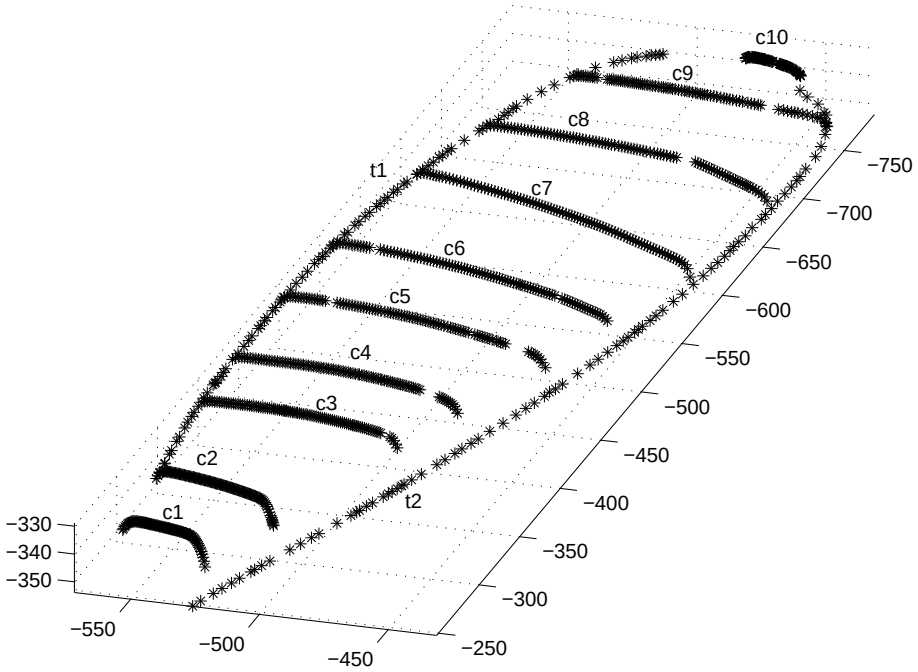
5.6 Discussion on the Presented Methodology

In this section, some remarks inherent to the parameters tuning the behavior of the GA (to be set by the user) are discussed. A particular attention is dedicated to the definition of the bounds for the design variables, which have been established according to the following considerations. *Continuous parameters* bounds are simple to set.

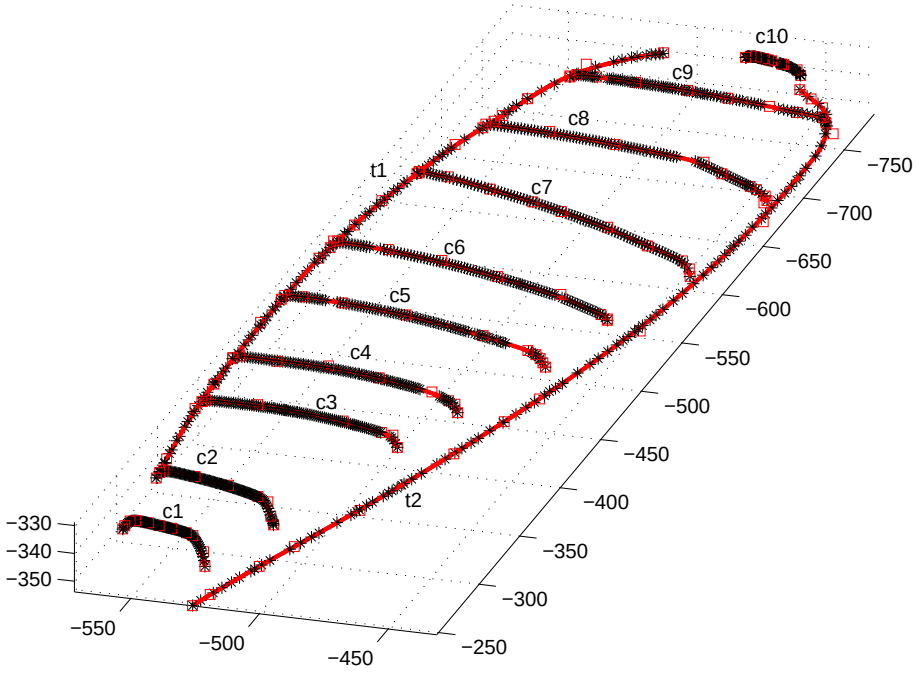
- The knot vector components are defined between 0 and 1, so $U_{jlb} = 0$ and $U_{jub} = 1$.
- The weights of the NURBS curve can get, *a priori*, any real value in the range $]0, \infty[$. After a preliminary check on the first three proposed benchmarks (the DescartesâĂŹs folium, the Four-Leaf Clover and the Flame problems), it was observed that the curve shape is affected by the ratio w_{ub}/w_{lb} rather than by the single value of the weight related to each control point. Moreover, as clearly shown in the Appendix of the paper, the weights are responsible of minor adjustments, which become significant only in presence of singularities (as in the case of the Flame problem). Taking into account these considerations, it has been set $w_{lb} = 1$ and $w_{ub} = 3$.

Unlike weights, the *discrete parameters* have a major influence on the shape of the NURBS curve and their bounds must be carefully set.

- The minimum degree is, of course, $p_{lb} = 1$. The maximum degree has been fixed in order to avoid the introduction of noise that can become important when the upper bound is not properly set. Accordingly, the maximum degree has been set to $p_{ub} = 8$ for all the examples with the exception of the first test case (the simplest one), where $p_{ub} = 6$.



(a) *The 12 subsets of sampled target points.*



(b) *The resulting curves.*

Figure 12: The paddle

- In order to establish lower and upper bounds for the number of the knot vector components $(m + 1)$, the user should think about an ideal number of control points tuning the shape of the approximating NURBS curve. Indeed, this problem applies also in case of standard curve fitting methods (which are not capable of automatically optimize discrete parameters), where the user does not dispose of any criterion to choose a suitable number of control points.

In the framework of the proposed method, the special GA utilized to perform the solution search for the curve fitting problem (refer to Eq. (25)) is able to automatically determine the optimum number of both knot vector components and degree of the basis functions, thus the related optimum number of control points, i.e. $n_{opt} = m_{opt} - 1 - p_{opt}$. Of course, the bounds on the variable m can be inferred according to empirical rules (taken from practice), utilized to define a criterion for setting the minimum and maximum number of control points. In particular, the bounds on n can be set according to the following rules:

1) usually, the number of target points $(\mu+1)$ should be, at least, three times the number of control points (in order to ensure redundancy). Therefore, the average number of control points can be assumed equal to $(\mu+1)/3$;

2) a suitable interval can be defined around this average value. In particular, the maximum number of control points must be lower than the number of target points, whilst the minimum one should be always greater or equal to 2. Anyway, regardless the definition of the interval for the variable m, an internal check (in the GA environment) is always performed to satisfy the condition $n \geq 1$, thus meaningless situations, e.g. $m = 8$ and $p = 7$, are always discarded.

Since the proposed hybrid algorithm is very efficient, it can be asserted that it is not important to choose the "right" narrow interval. When the shape of the curve is particularly complex and does not let the user guess the size of the interval, a wider range can be set, being the GA able to determine automatically the optimum value of the discrete parameters. Finally, it can be stated that the external user has a lower impact in the context of the proposed approach when compared to classical ones.

The previous discussion on the choice of the bounds for the number of knot vector components suggests to investigate the sensitivity of the solution to the quantity of TPs. This is an interesting task that allows for disputing about the robustness and the efficiency of the methodology. Since the amount of parameters to be optimized is high, it is natural to wonder what happens when the number of data points (TPs) is reduced, i.e. when the algorithm benefits from less information. However, some remarks need a special attention.

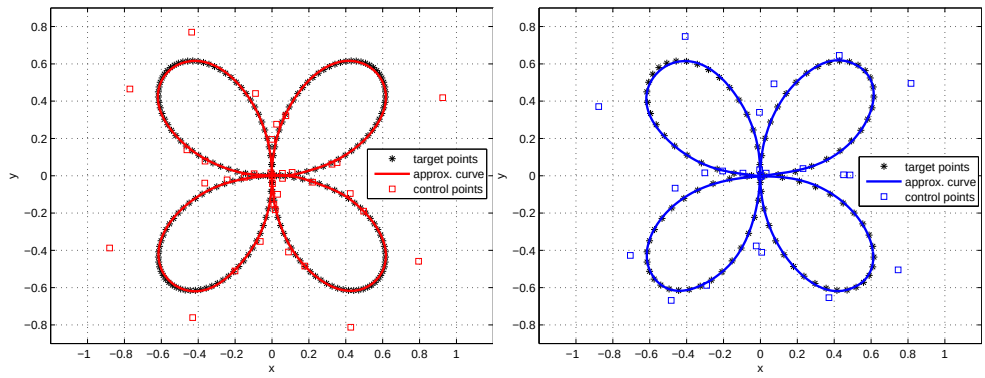
Decreasing the number of TPs has a significant impact on the mathematical nature of the curve fitting problem in the form of the CNLPP (25). If the number of TPs (i.e. data points) is less than the number of design variables, the related system of equations becomes underdetermined and the solution is not unique. Conversely, solving the curve fitting problem can be interpreted as finding an approximate solution for an overdetermined system of equations. Therefore, talking about curve fitting when the number of TPs is lower than the number of design variables is meaningless. Indeed, in this case, there is not enough information to get a unique solution for the curve fitting problem.

Usually, in reverse engineering applications, the number of data points (retrieved, for instance, by means of a 3D scanner) is very high. In practice, the size of points clouds can be properly reduced (in order to save memory in data exchange) but a certain redundancy must be guaranteed in order to approximate the data points with a single (or multiple) CAD entity like a NURBS curve, tuned by a suitable number of parameters.

Taking into accounts these aspects, a sensitivity analysis to the number of TPs is provided in the following for the Four-Leaf Clover example (Fig. 13). Solutions depicted in Figs. 13b-13d have been obtained with a decreased number of TPs with respect to the reference solution of Fig. 13a (see section 5.2) and by using the same value of maximum allowable curvature ($\chi_{adm} = 6.0000$). Two cases have been considered:

a) the bounds of design variables have been chosen according to the aforementioned empiric criteria (refer to Fig. 13b and Fig. 13c);

b) the bounds of design variables do not change with respect to the reference case (Fig. 13d).



(a) $(\mu + 1) = 211$, $m_{ub} = 67$. (b) $(\mu + 1) = 107$, $m_{ub} = 42$.

(c) $(\mu + 1) = 54$, $m_{ub} = 34$. (d) $(\mu + 1) = 54$, $m_{ub} = 67$.

Figure 13: Sensitivity to the number of TPs: approximating curves

As it can be inferred from both Fig. 13 and numerical results of Table 6, if the upper bound

of m is set according the proposed criterion, the lower is $(\mu+1)$, the lower is the quality of the solution: the quantity $d_{average}$ increases and the approximating NURBS curve is not satisfactory, in particular when the number of TPs is reduced to 54 (Fig. 13d).

Four-Leaf Clover	$\mu+1$	m_{ub}	m	$\Phi(\boldsymbol{\xi_1},\boldsymbol{\xi_2})$	L_{TP}	$\Phi_{grad}(\boldsymbol{\xi_2})$	$d_{average}$
1	211	67	39	0.7572	7.7516	6.67×10^{-4}	1.23×10^{-4}
2	107	42	28	0.7064	7.7408	1.74×10^{-3}	3.91×10^{-4}
3	54	34	16	0.6875	7.7105	1.07×10^{-1}	6.07×10^{-3}
4	54	67	63	0.3286	7.7105	2.45×10^{-31}	9.18×10^{-18}

Table 6: Sensitivity to the number of TPs - Numerical results

This fact occurs because the criterion for choosing the bounds aims at balancing the number of design variables with the number of TPs, which makes sense in the context of the curve fitting problem. Nevertheless, when $(\mu+1)=54$, the system becomes undetermined. Actually, since the solution is not unique, when the upper bound of m is increased (Fig. 13d) without considering the proposed empirical rule, the algorithm provides an excellent solution, which can be seen as the solution of the related interpolation problem.

Finally, handling data points is an operation that should be carefully assessed: some crucial information could be removed and this operation could have a high impact on the problem definition (e.g. removing the peaks of singularity in the Flame example can lead to misleading results).

6 Conclusions

In this paper, a general mathematical formalization of the curve fitting problem together with an original optimization procedure to perform the solution search in the framework of NURBS curves has been presented.

The proposed approach relies on the following features.

1. A new expression of the objective function, together with a suitable constraint on the maximum value of the curvature, has been introduced. These modifications imply a restriction on the integer design variables defining the shape of the NURBS curve. Moreover, the problem is stated as a CNLPP in which the number of unknowns is included among the design variables. Therefore, the problem of curve fitting is formulated in the most general case by considering as design variables both integer (the number of knots and the degree of the blending functions) and continuous (the components of the knot vector and the weights) parameters tuning the NURBS curve. These aspects are of paramount importance, since, in this background, the related CNLPP is defined over a domain of variable dimension, thus requiring a special optimization procedure to find a feasible solution.

2. The non-convexity of the problem, together with a definition domain of variable dimension, justifies the use of non-analytical methods. To this purpose, the solution search for the curve fitting problem is performed by means of a hybrid optimization tool (a GA coupled to a gradient-based method), of which the kernel is represented by a special GA able to deal with CNLPPs characterized by a "variable number of design variables".

3. The constraint on the curvature is effectively handled by the GA through the ADP method iteratively and automatically, i.e. by exploiting the genetic information restrained within the population (both feasible and infeasible individuals) at the current generation, without the need of defining arbitrary penalty coefficients at the beginning of the calculation.

The effectiveness of the proposed approach is proven through some numerical examples focusing on 2D and 3D parametric as well as real-world engineering problems. The presented method can adapt the approximating curve to imposed level of smoothness, set through the curvature constraint: in fact, the algorithm is capable of successfully approximate both smooth curves and curves with a drastically discontinuous derivatives. The robustness of the method has been discussed with respect to the sensitivity to both the boundaries of the design variables and the number of initial target points. The number of knot vector components, i.e. the parameter that mainly affects the final quality of the approximating curve, needs suitable bounds which can properly set by considering some practical guidelines provided in this study.

Acknowledgements

The first author is grateful to region Nouvelle-Aquitaine for supporting this research work through the FUTURPROD project.

Appendix : Details of the NURBS Curve of the Flame Problem

The optimized Knot Vector for flame problem:

$\mathbf{U} = [0, 0, 0, 0, 0, 0.0225, 0.0259, 0.0601, 0.0644, 0.0657, \mathbf{0.0727}, \mathbf{0.0727}, 0.0881,$
$\mathbf{0.0894}, \mathbf{0.0894}, \mathbf{0.1145}, \mathbf{0.1145}, 0.1220, 0.1410, \mathbf{0.1527}, \mathbf{0.1527}, \mathbf{0.1527}, 0.1638,$
$0.1753, 0.1778, 0.1833, 0.1904, \mathbf{0.1934}, \mathbf{0.1934}, \mathbf{0.1934}, 0.2013, 0.2137, 0.2287,$
$0.2354, 0.2422, 0.2627, 0.2642, \mathbf{0.2850}, \mathbf{0.2855}, \mathbf{0.2939}, \mathbf{0.2944}, 0.3054, 0.3327,$
$0.3457, 0.3535, 0.3633, 0.3692, 0.3743, \mathbf{0.3750}, \mathbf{0.3750}, 0.3777, 0.3880, 0.3888,$
$\mathbf{0.3927}, \mathbf{0.3927}, \mathbf{0.4161}, \mathbf{0.4161}, 0.4327, 0.4405, 0.4547, 0.4595, 0.4736, 0.5001,$
$0.5075, 0.5148, \mathbf{0.5362}, \mathbf{0.5364}, \mathbf{0.5364}, 0.5452, 0.5470, 0.5613, 0.5665, 0.5712,$
$0.5948, 0.5995, \mathbf{0.6029}, \mathbf{0.6032}, 0.6054, 0.6153, 0.6191, 0.6206, 0.6310, 0.6451,$
$0.6505, 0.6516, 0.6829, 0.6872, 0.7094, 0.7106, 0.7309, 0.7503, 0.7614, 0.7722,$
$\mathbf{0.7859}, \mathbf{0.7859}, 0.7964, 0.8005, 0.8087, 0.8237, 0.8414, \mathbf{0.8482}, \mathbf{0.8482}, 0.8655,$
$0.8687, 0.8821, 0.8837, \mathbf{0.9231}, \mathbf{0.9231}, 0.9446, 0.9563, 1, 1, 1, 1, 1,].$

The optimized weights vector for the flame problem:

$\mathbf{w} = [1.0029, 0.9997, 0.9992, 0.9857, 1.0179, 0.9971, 1.0009, 0.9978, 1.0109,$
$0.9895, 1.0040, 1.0111, 0.9769, 0.9893, 1.0297, 1.0127, 0.9786, 0.9974, 0.9847,$
$1.0164, 1.0162, 0.9817, 1.0005, 1.0005, 0.9965, 1.0023, 1.0000, 1.0071, 0.9898,$

1.0014, 0.9967, 0.9961, 1.0357, 0.9640, 1.0114, 1.0065, 0.9907, 0.9893, 1.0323,
0.9857, 0.9919, 0.9969, 1.0058, 0.9998, 0.9996, 0.9982, 1.0008, 1.0020, 1.0000,
1.0003, 0.9961, 0.9912, 1.0246, 0.9788, 0.9959, 1.0669, 0.9176, 1.0248, 1.0124,
0.9721, 1.0761, 0.8686, 1.0859, 0.9872, 0.9891, 1.0072, 1.0080, 1.0115, 0.9620,
1.0305, 0.9737, 1.0232, 0.9907, 1.0006, 1.0001, 1.0017, 1.0014, 0.9866, 1.0179,
0.9985, 0.9819, 1.0079, 1.0018, 1.0276, 0.9920, 0.8576, 1.1315, 1.1224, 0.8604,
0.9518, 1.0663, 1.0000, 0.9807, 1.0491, 0.9341, 0.9720, 1.0871, 0.9647, 1.0268,
0.9844, 0.9642, 1.0530, 0.9769, 0.9789, 1.0117, 1.0152, 0.9919, 0.9820, 1.0146,
1.0015,].

References

[1] W. Wang, H. Pottmann, and Y. Liu. Fitting b-spline curves to point clouds by squared distance minimization. Technical Report 11, The University of Hong Kong, Department of Computer Science, 2004.

[2] W.D. Ueng, J.Y. Lai, and Y.C. Tsai. Unconstrained and constrained curve fitting for reverse engineering. *International Journal of Advanced Manufacturing Technology*, 33:1189–1203, 2007.

[3] W. Zheng, P. Bo, Y. Liu, and W. Wang. Fast b-spline curve fitting by l-bfgs. *Computer-Aided Design*, 29:448–462, 2012.

[4] L. Piegl and W. Tiller. *The NURBS book*. Berlin, 1997.

[5] W. Li, S. Xu, G. Zhao, and L. P. Goh. Adaptive knot placement in b-spline curve approximation. *Computer-Aided Design*, 37:791–797, 2005.

[6] A. Limaiem, A. Nassef, and H.A. El-Maraghy. Data fitting using dual kriging and genetic algorithms. *CIRP Annals - Manufacturing Technology*, 45(1):129–134, 1996.

[7] A. Galvez and A. Iglesias. Particle swarm optimization for non-uniform rational b-spline surface reconstruction from clouds of 3d data points. *Information Sciences*, 192:174–192, 2012.

[8] H. Kang, F. Chen, Y. Li, J. Deng, and Z. Yang. Jnot calculation for spline fitting via sparse optimization. *Computer-Aided Design*, 58:179–188, 2015.

[9] D. Brujic, I. Ainsworth, and M. Ristic. Fast and accurate nurbs fitting for reverse engineering. *International Journal of Advanced Manufacturing Technology*, 54:691–700, 2011.

[10] C.H. Garcia-Capulin, F.J. Cuevas, G. Trejo-Caballero, and H. Rostro-Gonzalez. A hierarchical genetic algorithm approach for curve fitting with b-splines. *Genetic Programming and Evolvable Machines*, 16:151–166, 2015.

[11] M. Montemurro. *Optimal design of advanced engineering modular systems through a new genetic approach*. PhD thesis, Université Pierre et Marie Curie Paris VI, 2012.

[12] M. Montemurro, A. Vincenti, and P. Vannucci. A two level procedure for the global optimum design of composite modular structures - application to the design of an aircraft wing. Part 1: theoretical formulation. *Journal of Optimization Theory and Applications*, 155(1):1–23, 2012.

[13] M. Montemurro, A. Vincenti, and P. Vannucci. A two level procedure for the global optimum design of composite modular structures - application to the design of an aircraft wing. Part 2: numerical aspects and examples. *Journal of Optimization Theory and Applications*, 155(1):24–53, 2012.

[14] *Optimization Toolbox User's Guide*. The MathWorks Inc, 3 Apple Hill Drive, Natick, September 2011.

[15] A. Catapano and M. Montemurro. A multi-scale approach for the optimum design of sandwich plates with honeycomb core - Part I: homogenisation of core properties. *Composite Structures*, 118:664 – 676, 2014.

[16] A. Catapano and M. Montemurro. A multi-scale approach for the optimum design of sandwich plates with honeycomb core - Part II: the optimization strategy. *Composite Structures*, 118:677 – 690, 2014.

[17] M. Montemurro, A. Vincenti, and P. Vannucci. Design of the elastic properties of laminates with a minimum number of plies. *Mechanics of Composite Materials*, 48(4):369–390, 2012.

[18] M. Montemurro, A. Catapano, and D. Doroszewski. A multi-scale approach for the simultaneous shape and material optimisation of sandwich panels with cellular core. *Composites Part B: Engineering*, 91:458 – 472, 2016.

[19] M. Montemurro and A. Catapano. *Variational analysis and aerospace engineering: mathematical challenges for the aerospace of the future*, volume 116 of *"Springer Optimization and Its Applications"*, chapter A new paradigm for the optimum design of variable angle tow laminates. Springer International Publishing, 1st edition edition, 2016.

[20] M. Montemurro and A. Catapano. On the effective integration of manufacturability constraints within the multi-scale methodology for designing variable angle-tow laminates. *Composite Structures*, 161:145 – 159, 2016.

[21] M. Montemurro, A. Vincenti, and P. Vannucci. The automatic dynamic penalization method (ADP) for handling constraints with genetic algorithms. *Computational Methods Applied to Mechanical Engineering*, 256:70–87, 2012.