Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 716237, 15 pages
http://dx.doi.org/10.1155/2013/716237

Research Article

A Comparative Analysis of Nature-Inspired Optimization Approaches to 2D Geometric Modelling for Turbomachinery Applications

Amir Safari,[1] Hirpa G. Lemu,[1] Soheil Jafari,[2] and Mohsen Assadi[3]

[1] *Department of Mechanical & Structural Engineering, University of Stavanger, 4036 Stavanger, Norway*
[2] *Department of Petroleum Engineering, University of Stavanger, 4036 Stavanger, Norway*
[3] *Centre for Sustainable Energy Solutions, International Research Institute of Stavanger, 4021 Stavanger, Norway*

Correspondence should be addressed to Amir Safari; amir.safari@uis.no

Received 3 July 2013; Accepted 18 September 2013

Academic Editor: Jung-Fa Tsai

A vast variety of population-based optimization techniques have been formulated in recent years for use in different engineering applications, most of which are inspired by natural processes taking place in our environment. However, the mathematical and statistical analysis of these algorithms is still lacking. This paper addresses a comparative performance analysis on some of the most important nature-inspired optimization algorithms with a different basis for the complex high-dimensional curve/surface fitting problems. As a case study, the point cloud of an in-hand gas turbine compressor blade measured by touch trigger probes is optimally fitted using B-spline curves. In order to determine the optimum number/location of a set of Bezier/NURBS control points for all segments of the airfoil profiles, five dissimilar population-based evolutionary and swarm optimization techniques are employed. To comprehensively peruse and to fairly compare the obtained results, parametric and nonparametric statistical evaluations as the mathematical study are presented before designing an experiment. Results illuminate a number of advantages/disadvantages of each optimization method for such complex geometries' parameterization from several different points of view. In terms of application, the final appropriate parametric representation of geometries is an essential, significant component of aerodynamic profile optimization processes as well as reverse engineering purposes.

1. Introduction

The turbomachinery blades' aerodynamic profile and structural strength play an important role in improvement of energy conversion efficiency as well as the reliability and availability of the machine in both land-based and aero applications. From aerodynamic point of view, geometry-based blade design optimization is of utmost importance for further enhancement of turbomachinery performance. On the other hand, a robust and precise reverse engineering of existing blades could also maintain their performance as near as possible to the reference design. In this way, precise and proper geometry modeling and parameterization of current blades are the first and sometimes the most important step in both optimization and reverse engineering missions which also, in turn, need an optimization effort [1, 2].

At the same time, recent advances in both mathematical tools and optimization techniques enable further improvement in geometric modeling procedure. Irrespective of the goal of blade profile parameterization, that is, its use in inverse aerodynamic design or in its application to a direct numerical optimization, there are several parameterization methods with different impacts on the reversing/optimization process. For free-form shapes, however, parametric curves/surfaces including Bezier [3], B-spline, and nonuniform rational B-spline (NURBS) [4] are generally used as the most suitable ones. For the parameterization of blade/airfoil shapes, specifically, though there exists no unique way, NURBS curves/surfaces [1, 5, 6] and Bezier curves [7, 8] can be mentioned as the significant and commonly used methods. Some other methodologies also

exist, for example, PARSEC method [9] and Hicks-Henne functions [10], for this application.

As mentioned above, a proper and precise curve fitting requires an unavoidable optimization mission. The nonlinear nature of sophisticated geometries, like turbomachinery blades' shape, and availability of high-speed parallel computers for massive computation have resulted in the use of non-gradient-based and guided random search methodologies in curve/surface fitting problems [11–14]. On the other hand, a vast variety of population-based optimization techniques have been formulated in recent decades, some of which are inspired by natural processes taking place in our environment. Genetic algorithms (GAs) [15], particle swarm optimization (PSO) [16], and ant colony optimization (ACO) [17] are some such methods that have already been used in various engineering optimization problems, from the development of optimum control systems [18] to curve fitting and geometric modeling [11, 14].

Among these optimization techniques, genetic algorithm and differential evolution (DE) [19] are the most successfully used strategies in curve fitting optimization of airfoils' geometry parameterized by NURBS/Bezier curves. While GA and real-coded genetic algorithm (RCGA) are stochastic search and population-based optimization algorithms that mimic Darwin's theory of biological evolution, DE is a relatively newer evolutionary method in engineering applications and requires only a few user-defined parameters.

PSO and invasive weed optimization (IWO) [20] are other population-based optimization methods inspired by social and ecological behavior. The PSO algorithm is based on the simulation of the social behavior of birds flocking, fish schooling, and animals herding to let them adjust to their environment, find rich sources of food, and keep away from other predatory animals by using information sharing and social cognitive intelligence. IWO is also another derivative-free metaheuristic method originally formulated based on the concept of the natural social behaviour of colonizing weeds. These swarm intelligence approaches have demonstrated their high capability in parameters' optimization for nonlinear multidisciplinary problems.

In this paper, a study on a gas turbine compressor airfoil shape parameterization and optimum curve fitting as a main step to blade geometry optimization has been reported. For this purpose, two most important curve parameterization methods, that is, Bezier and NURBS curves, have been applied within the five above-mentioned population-based optimization loops (GA, RCGA, DE, PSO, and IWO) so that the best alternative solutions from different points of view can be discovered. To achieve this goal, the parametric procedures involving independence, normality and homoscedasticity are firstly discussed. Then, the nonparametric statistical analysis including Quade and Friedman aligned comparison tests, by applying several different post hoc procedures, are brought to perform a rigorous comparison among the performance of the optimization algorithms. After that, designing an experiment is done for each algorithm to finally illustrate the effectiveness of the proposed algorithms.

In the rest of the article, the theory behind the Bezier and nonuniform rational B-spline curves and different airfoil geometric representation techniques are first described in Section 2. In Section 3, the problem which will be solved is clearly defined. This section involves the problem statement and explanation of the formulation, fitness function, and flowchart considered for the optimization processes. Then, the well-organized optimization techniques for optimum curve fitting used in this research are explained in Section 4. The first subsection focuses on evolutionary optimization techniques (GA, RCGA, and DE), and the second considers swarm intelligence and colonizing techniques (PSO and IWO). Section 5 discusses the adjusted parameters for the optimization algorithms, followed by the statistical analysis, experimental results, discussions, and comparison. Finally, conclusion and future works have been presented in the last section of the article.

2. Airfoil Geometric Modelling: Techniques and Tools

The turbomachinery blade shape parameterization addresses the two-dimensional airfoil profile construction from either the direct handling of curves of airfoil shapes or the superposition of the camber line and thickness distribution around it. Between these two different approaches of geometry definition, however, many designers have considered distinct parameterized curves for the suction surface (SS), pressure surface (PS), leading edge (LE), and trailing edge (TE) of a blade section. In this article, direct handling of the curves of the airfoil shape has been used by dividing the profile from the hub to the shroud into five sections as shown in Figure 1. This article focuses on only the first section, that is, the blade hub section (depicted in Figure 1(b)).

As the first step, a standard coordinate measuring machine equipped with touch trigger probes has been employed to measure 2D point cloud of compressor blade in-hand (Figure 1(a), top). These measured points' coordinates have been used to fit the airfoil sketches using NURBS and Bezier curves. To achieve this purpose, after indicating the digitized points of the airfoil profile and choosing the approach for airfoil shape modeling, four curves will separately be fitted to any airfoil shape. In this way, the accuracy and robustness of the created geometric model can be managed by controlling the curve parameters like number of Bezier/NURBS control points (CPs) and/or curve order.

Here, the theory behind these mathematical methods and the characteristics of each will be briefly described.

2.1. Bezier Curves. The Bezier curve, proposed by Bezier in the early 1960s [21], is defined by the vertices of a polygon which enclose the final fitted curve. Bezier curves use the Bernstein polynomial function, $B_{i,n}(u)$ as the blending function [22]:

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}, \tag{1}$$

where n is the degree of the curve, u is the parameter of the function, and $i = 0, 1, \ldots, n$.

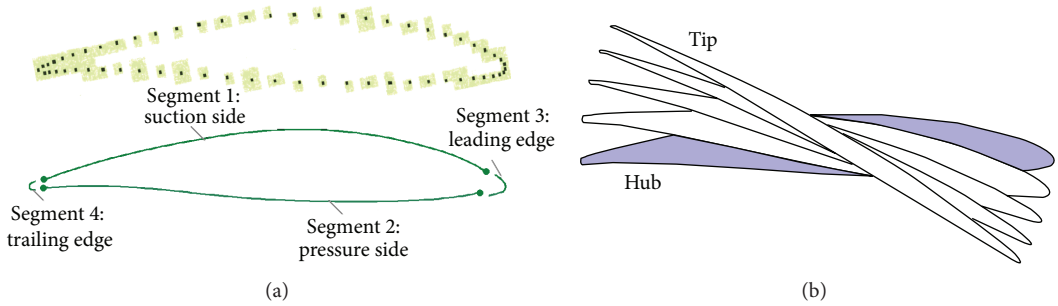

FIGURE 1: Measured points cloud for a section along the blade span, and four distinct segments of the airfoil for the curve fitting problem (a). Gas turbine compressor blade sections-view from the top (b).

When this blending function is applied to the vertices of the polygon, the Bezier curve equation is found as

$$P(u) = \sum_{i=0}^{n} \binom{n}{i} u^i (1-u)^{n-i} P_i, \tag{2}$$

where P_i is the position vector of the ith control vertex. For a Bezier curve of degree n, the number of CPs that control the shape of the curve is $(n + 1)$. In other words, it can be concluded that Bezier curves are affected by these two main parameters: the polynomial degree and the number of CPs. On the other hand, (2) shows that the degree of the Bezier curve is determined by the number of CPs. Another challenging property of a Bezier curve is that any CPs affect the shape of the entire curve. The third important property of a Bezier curve is its convex hull property, which means that these curves are completely situated inside their own convex hull, constructed by the polygons.

2.2. Nonuniform Rational B-Spline (NURBS) Curves. One of the major drawbacks of Bezier curves is that the curve degree is directly determined by the number of CPs. In addition, each CP has an impact on the shape of the whole curve. This becomes of specific concern when only some minor modifications on some specific places on the blade sections are required. Therefore, using B-spline curves is an alternative because of their local modification property.

In the early 1970s, Cox [23] and de Boor [24] suggested new blending functions $N_{i,k}(u)$ expressed as

$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}}, \tag{3}$$

where $N_{i,1}(u)$ is equal to 1 for $t_i \le u \le t_{i+1}$ and equal to zero otherwise.

Based on this equation, the B-spline curve is defined as follows:

$$P(u) = \sum_{i=0}^{n} N_{i,k}(u) P_i. \tag{4}$$

In these equations, n is the number of CPs, k is the order of the curve, $n + k + 1$ is the number of knot values, and 0/0 is assumed to be equal to zero. The gap between neighboring knots, in this specific definition, is always uniform. Therefore, the extracting curve is called a uniform B-spline curve. However, as these curves are used for a shape optimization process, some knot values may be added/deleted resulting in nonuniform gaps between the knots. These new generated nonperiodic/nonuniform curves are also properly matched with computer-aided design (CAD) systems. In the end, a nonuniform rational B-spline curve (NURBS), which is similar to nonuniform B-spline curves, is defined by the following equation:

$$P(u) = \frac{\sum_{i=0}^{n} w_i N_{i,k}(u) P_i}{\sum_{i=0}^{n} w_i N_{i,k}(u)}, \tag{5}$$

where n is the number of CPs, k is the order of curve, $k + n$ is the number of knots, P_i are CPs, and w_i are the corresponding weights of CPs. A Bezier curve can also be considered as a specific type of NURBS representation. NURBS curves have a number of significant characteristics including

(i) numerical stability (due to the independence of polynomial degree from number of CPs),

(ii) local shape control or local modification property (because of the influence range of each CP),

(iii) coincidence of endpoints with first/last CPs,

(iv) convex hull property (like Bezier curves),

(v) more versatile modification of the created curve,

(vi) exact presentation of conic curves (circles, ellipses, etc.).

The above-mentioned characteristics are important, particularly when a researcher faces a real case study. For example, the last distinction can really be helpful for our specific problem in the case where the designer wants to construct two circles/ellipses for trailing/leading edges of the airfoil shape. In the present work, these two segments have also been modeled by free-form curves just like the other two segments.

Depending on whether the purpose of the designer is to create a completely new airfoil design or to make some minor

aerodynamic improvements to the existing blade profile, a global geometry modeling approach or a localized parameterization method can be used, respectively. Considering a number of the important advantages for NURBS representation mentioned above and because of the prospective goal of the present continuing research to slightly improve the shape of the existing turbomachinery blades for some specific objectives in upcoming works, the application of NURBS curves seems to be more reasonable. Indeed, their affine invariance and local control property make them appropriate for such geometry handling. The comparison between Bezier and NURBS curves would be discussed further in the results section of the paper.

3. Problem Statement and Applied Methodology

After skimming the background of Bezier/NURBS curves, the optimum data fitting problem using the defined tools and the planned approach is made clear in this section.

When the digitized points are available, model construction is continued by airfoil segmentation and CAD modeling. Segmentation subdivides the measured points into areas for various individual features [25], and CAD modeling identifies geometric features from the measured points and merges the identified features into a complete CAD model. As stated before, the airfoil shape is divided into four segments including LE, TE, PS, and SS in this work. Consequently, four optimum Bezier/NURBS curves will separately be fitted to any airfoil shape and then merged together by enforcing C^2-continuity between the segments.

Concerning the NURBS curve fitting specifically, various fitting schemes have been developed so far. Some researchers have identified both the CPs and the weights of a NURBS curve simultaneously by minimizing the sum of the squares of the distances from the measured points to the corresponding fitted curve points, while others have focused only on the optimization of the CPs' location. Regarding the airfoil shape modeling, a third-order NURBS curve/surface has recently been employed to describe the suitable parametric geometry of 2D airfoil and 3D blades.

3.1. Objective Function Formulation. In this paper, the best CPs of both Bezier and NURBS curves among the candidates are searched for by using the least squares method. Although the proposed method can automatically determine the appropriate CPs, both number and location, at the same time, these optimal CPs will only be searched in a predefined range—based on the existing knowledge about the constraints of airfoil shape at hand—to save time and CPU efforts. Therefore, the CPs of these curves can be utilized as characteristic points. This procedure clearly needs an optimization effort due to the lack of knowledge about the exact position of the proper CPs at the first stage of curve fitting problem. Hence, the objective function which should be minimized is the sum of distances between the fitted curve and the original scanned data points. In such an optimization process, Bezier/NURBS CPs are the design variables, in fact, that have to be forced to the optimal locations.

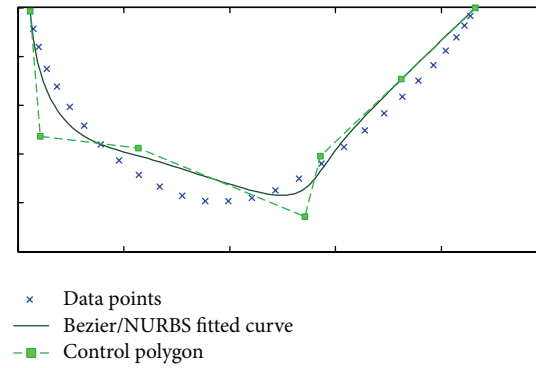

Figure 2: Errors between the original measured points and fitted curve.

The above-mentioned procedure is more clearly depicted in Figure 2 and (6):

$$E = \sum_{i=1}^{n} \left| C_i - P\left(u_i\right) \right|, \tag{6}$$

where C_i are the original data points, n is the number of measured data points, and $P(u_i)$ are corresponding points on the fitted Bezier/NURBS curve.

The airfoil (as a profile section of the compressor blade) considered in this study consists of four Bezier/NURBS curves for four segments, and nine CPs identify each segment. Each CP has two coordinates x and y, and so this will lead to a total number of 72 design variables. Since the optimum curve fitting problem is separately solved for the mentioned segments of airfoil (SS, PS, LE, and TE), the strategy to fix the intersection points of the segments through the optimization process can force the chord of the airfoil to be constant. In this way, 16 parameters are known and fixed. Another 16 parameters are also determined by enforcing C^2-continuity at the intersections of the segments. As a result, 40 variables will remain for the optimization algorithms as the design parameters. In addition, the rational upper bounds and lower bounds should be defined for variations of the parameters to have an effective and efficient optimization route.

All optimization techniques in this paper have been used to locate the optimal position of the CPs for a set of open Bezier and/or NURBS curves according to the objective function represented in (6). Based on that, the objective is minimization of the sum of the errors E. Figure 3 depicts the flowchart of the performed optimal curve fitting procedure in this study.

4. Planned Optimization Algorithms

This section briefly discusses the procedures, operators, and logic of all optimization algorithms used in this study. For this purpose, the optimization techniques are conceptually divided into two main categories based on their characteristics: (1) evolutionary algorithms involving genetic algorithm

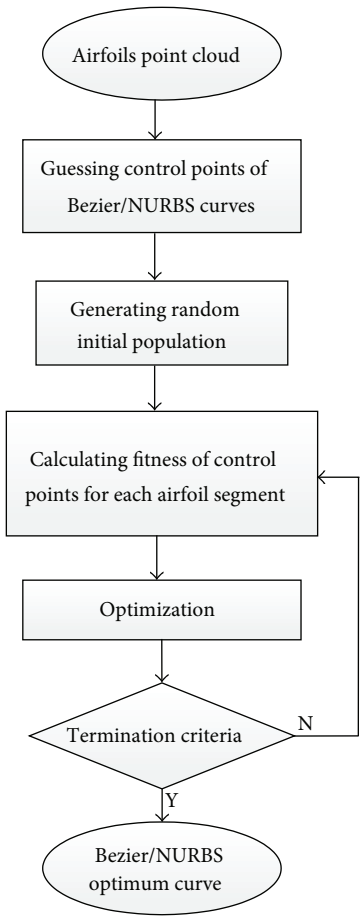

Figure 3: Bezier/NURBS optimum curve fitting flowchart.

and differential evolution and (2) swarm intelligence and colonized algorithms including particle swarm and invasive weed optimization methods (Figure 4).

4.1. Evolutionary Optimization Technique. As mentioned earlier, two evolutionary optimization methods customized and specialized by the authors are used to optimally fit the parametric curves to the airfoil point cloud. As the algorithms are based on Darwin's theory of evolution, the optimization uses different mechanisms and operators to search for individuals that best adapt to the environment. In other words, individuals with a higher level of fitness have a greater chance to survive and continue in the optimization process.

4.1.1. Genetic Algorithm (GA). GA was introduced by Holland [15] as a probabilistic global search method. Its operation is based on the combination and generation of DNAs and chromosomes and it mimics the metaphor of natural biological evolution. GA operates on a population of individuals (potential solutions), each of which is an encoded string (chromosome), containing the decision variables (genes).

The structure of a GA is composed of an iterative procedure through the following key steps [26], as also depicted in Figure 4:

(i) creating an initial population of airfoil shapes with consideration of the existing experience,

(ii) evaluation of the fitting performance of each profile in the population,

(iii) selection of the best-fitted airfoils' segments and reproduction of a new population,

(iv) application of genetic operators, that is, crossover and mutation,

(v) iteration of steps II to IV until the maximum number of generations is reached.

The fitness value is associated with each individual, expressing the performance of any airfoil shapes created with respect to a fitting error function that should be minimized (E in (6)). Also, the positive effect of mutation is the preservation of genetic diversity and the fact that the local minima can be avoided. Following the evaluation of the fitness of all profiles in the population, the genetic operators are repeatedly applied to produce a new population of airfoil shapes.

In addition, a real coding approach, instead of the standard GA coding strategy that is a binary pattern, is also used in this study. Using this approach, more adequate mutation operators can be defined while the destructive effects of crossover are reduced, specifically for a set of data points in the x-y coordinate. Furthermore, the errors regarding the CPs' locations caused by discretization are avoided [14]. The reproduction approach here is the roulette wheel method, in which the probability of choosing a certain individual is proportional to its fitness.

4.1.2. Differential Evolution (DE). DE is one of the newly emerging, but well-known, stochastic parallel direct search methods as well as a kind of population-based optimization algorithm. DE was presented by Price and Storn in 1997 [19], as an evolutionary algorithm designed for solving continuous optimization problems. DE can be used to find approximate solutions for many practical problems having objective functions that are nondifferentiable, noncontinuous, nonlinear, noisy, flat, and multidimensional or having many local minima, constraints, or stochasticity. DE uses mutation and crossover operators and a selection method to generate the population of the next generation [27].

As illustrated in Figure 4, DE procedure involves the following steps:

(i) N individuals are created randomly,

(ii) every individual of the population (each set of CPs for the problem in hand) undergoes mutation operation with scale factor (SF) a positive real number typically less than 1,

(iii) mutation vector and target vector undergo crossover operation with crossover rate (CR), and

(iv) trial vector is evaluated and compared with the fitness value of the target vector. The vector with the greater

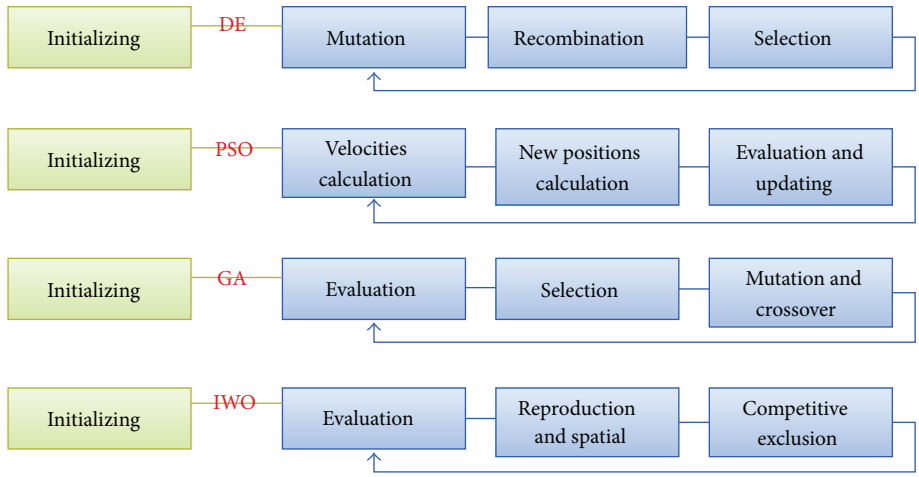

Figure 4: Key similarities and differences among DE, PSO, GA, and IWO procedures.

fitness value (the fitter CPs' set) enters the next generation.

Recombination incorporates successful solutions from the previous generation, and mutation expands the search space, as in GA. Mutation, recombination, and selection continue until some stopping criterion is reached.

The present paper proposes an alternative method by implementing a DE algorithm for the curve fitting problem. This proposed method is compared with the result obtained from the GA method in the last section of the paper.

4.2. Swarm Intelligence Technique. The theoretical bases for two swarm intelligence methods employed in the comparative study are described in this section.

4.2.1. Particle Swarm Optimization (PSO). A very popular swarm intelligence algorithm introduced by Kennedy and Eberhart is called particle swarm optimization [28]. This algorithm is based on the simulation of the social behavior of flocks of birds, school of fish, and herds of animals to let them adjust to their environment, find rich sources of food, and keep away from other predatory animals by using information sharing and social cognitive intelligence [29]. This algorithm generates a set of solutions in a multidimensional space randomly which represent the initial swarm. The initial swarm contains particles that move in the space and search for the best global position over the number of iterations. Then, iterations will be continued until the best global position is reached.

In general, the PSO algorithm consists of three main steps as follows (Figure 4):

(i) generating positions and velocities of particles,

(ii) updating the velocities,

(iii) updating the positions.

In this way, each particle refers to a point in a multidimensional space where its dimensions are related to the numbers of design variables. The positions of the points as well as the particles' velocity change as the iterations, which are calculated in each step, proceed. In the first step, position x_o^i and velocity v_o^i for each point are generated randomly by the use of upper and lower bounds of variables employing the following equations:

$$x_o^i = x_{\min} + \text{rand}\left(x_{\max} - x_{\min}\right),$$
$$v_o^i = \frac{x_{\min} + \text{rand}\left(x_{\max} - x_{\min}\right)}{\Delta t}, \tag{7}$$

where x_o^i and v_o^i are position and velocity of the ith particle in the first step, respectively, $x_{\min}$ and $x_{\max}$ are the lower and upper bounds of the design variable, respectively, rand() is a random number between 0 and 1, and Δt is the time step.

The first population is distributed uniformly in the space by this process. In the second step, PSO calculates new velocities to move the particles from positions in time, k, to new positions in time, $k+1$. In fact, PSO updates the particles' velocities for transferring to the next positions. For calculation of the new velocities, PSO needs two important values: the best global position of particles in the current swarm, P_k^g, and the best position of each particle over all previous and current steps, P^i. These values are saved in the memory of each particle. PSO then employs these two values besides three coefficients w, c_1, and c_2 to calculate new velocities for the next iteration using a random distribution function. The three aforementioned coefficients indicate the effect of the current motion, the particle's own memory, and swarm influence. Recently, some PSO approaches for optimization purposes take the effect of neighboring positions into account in order to improve the speed of convergence to the best global position. In this case, a term with factor c_3, called neighborhood acceleration, is used in order to take this effect into consideration [30, 31].

Based on the above explanations, the velocity update formula takes the following form:

$$v_{k+1}^{i} = wv_{k+1}^{i} + c_1 \text{rand} \frac{\left(P^{i} - x_k^{i}\right)}{\Delta t} + c_2 \text{rand} \frac{\left(P_k^{g} - x_k^{i}\right)}{\Delta t} + c_3 \text{rand} \frac{\left(L_{\text{Best}} - x_k^{i}\right)}{\Delta t}, \tag{8}$$

where w is the inertia factor in the range from 0.4 to 1.4, v_k^i is the velocity of the ith particle in current motion, c_1 is the self-confidence factor in the range from 1.5 to 2, c_2 is the swarm confidence factor in the range from 2 to 2.5, c_3 is the neighbor acceleration factor, x_k^i is the position of the ith particle in the current motion, P^i is the best position of the ith particle in the current and all previous moves, P_k^g is the position of the particle with the best global fitness at current move, k, and L_{Best} is the best local neighbor position in the current move.

It is worth pointing out that when the effect of neighbor acceleration is considered in the velocity update ($C_3 \neq 0$) as it has been done in this study, the PSO converges noticeably faster. This is due to more information sharing between particles in this way. So, taking the neighbor acceleration factor into account is a trade-off between convergence rate and computational time.

Lastly, using the updated velocity vectors calculated by (8), the position of each particle is changed by the following equation:

$$x_{k+1}^{i} = x_k^{i} + v_{k+1}^{i} \Delta t, \tag{9}$$

where x_{k+1}^i is the new position of the ith particle in step $k+1$.

This algorithm is repeated until a stop criterion is reached. The stop criteria may be an iteration number or a specified error for the best global value.

4.2.2. Invasive Weed Optimization (IWO). Invasive weed optimization is an ecologically inspired optimization algorithm, newly developed by Mehrabian and Lucas [20]. The algorithm is a derivative-free metaheuristic method originally formulated based on the concept of the natural social behavior of colonizing weeds [32]. In their leading research, they have shown the merits of IWO in finding the global optimum of different complex multidimensional optimization problems, which reveals that IWO is an appropriate competitor for other comparatively older and well-established techniques for population-based evolutionary computation.

IWO is implemented in this study with a problem dimension of ten and using the following procedure recursively:

(i) initializing a population (randomly spreadover the d-dimensional problem space with random positions),

(ii) fitness evaluation for each member of the colony of weeds to produce seeds depending on its own and the colony's lowest and highest fitness,

(iii) reproduction (based on plant fitness and number of seeds),

(iv) spatial dispersal (i.e., the generated seeds are randomly distributed over the search space by normally distributed random numbers with mean equal to zero, but varying standard deviation, SD),

(v) competitive exclusion (poor plants elimination).

The flowchart of the IWO method is also illustrated in Figure 4.

The variation of the SD value with generation is defined as follows:

$$\text{SD}_I = \left(1 - \frac{I}{I_{\max}}\right)^{n} \left(\text{SD}_i - \text{SD}_f\right) + \text{SD}_f, \tag{10}$$

where $I_{\max}$ is the maximum number of iterations, n is the nonlinear modulation index, and i and f refer to the initial and final condition, respectively. The SD value is decreased generation by generation. The decreasing rate depends on the nonlinear modulation index value which results in grouping fitter plants and the elimination of inappropriate plants, representing transformation from r-selection to k-selection mechanism [20].

After reaching the maximum number of plants, the competitor exclusion mechanism activates in order to eliminate the plants with poor fitness in the generation. This mechanism is formulated so that it gives a chance to plants with lower fitness to reproduce, and if their offspring has a good fitness in the colony, then they will survive. The above-mentioned steps are repeated until the maximum number of iterations is exceeded.

5. Implementation and Results

This section presents several illustrations of compressor airfoil data fitting from different points of view. The goal is to show the performance of the proposed optimization methodologies to be used in such geometric modelling problems and to make a comparison among them. In order to have a fair comparison of the results, both the initial population and the stopping criteria are set to be identical for the relevant algorithms.

Before representing the experimental illustrations, however, it is expected to compare and analyse the results in a systematic approach recently designed and proposed [33, 34]. For this purpose, the parametric and the nonparametric statistical tests are carried out besides designing an experiment in order to improve the performance evaluation process and statistical confidence.

Furthermore, the experimental results are presented by applying all optimization algorithms to the given airfoil shape parameterization process. The following figures and tables present the historical convergence of optimization algorithms during the generations, differences between Bezier and NURBS curves from different points of view, and their advantages and disadvantages, static and dynamic convergences of the used algorithms, fitting error in each case, and computational efficiency.

All coding and simulation carried out in this research have been implemented in MATLAB R2010b on a PC with Intel(R) Xeon(R) CPU X5650 @ 2.66 GHz with 2 processors and 32.0 GB RAM.

5.1. Employment of Statistical Tests. This section outlines the comparisons between the results obtained from the various algorithms. Studies conducted to prove the relative merits of the algorithms used for the curve fitting problem in hand are presented.

With respect to the no free lunch theorem in search and optimization, it is not possible to find one specific algorithm being better in behaviour for any given problems. Therefore, some criteria to select an appropriate optimization algorithm in each problem are sought. In this regard, statistical procedures including the parametric tests and nonparametric tests could be utilized for a fair comparison between the results. For this purpose, the required condition for the parametric tests is firstly described based on the published studies. These conditions are then checked for the obtained results in this work. Subsequently, the results of a nonparametric analysis using several methods are represented.

5.1.1. Parametric Test. In order to use the parametric test, it is necessary to check independence, normality, and homoscedasticity procedures.

For the problem in hand, the independence of the results is obvious since they are independent runs of any algorithm with randomly generated initial population. At the same time, an observation is said to be normal when its behaviour follows a normal or Gauss distribution with a certain average value μ and variance σ, and a normality test applied over a sample can indicate the presence or absence of this condition in the observed data. Lastly, the homoscedasticity condition is to indicate a violation of the hypothesis of equality of variance [33]. All the tests would get the related P values which indicate the dissimilarity of the samples with respect to the normal shape. To do this, all the given optimization algorithms are executed 30 times with the same initial population. The results of fitness function values are presented in Table 1. For the normality check of these results, the MATLAB environment is used. A low P value shows a nonnormal distribution. In this paper, a level of significance $\alpha = 10\%$ is considered, and so, the P value greater than 0.1 specifies the fulfilment of the normality condition. In other words, this means that there is less than a 10% probability that the sample data does not agree with our hypothesis due to random chance.

In this study, two different tests are used for numerical assessment of the normality. The first one is Kolmogorov-Smirnov (K-S test) which is a distribution-free and nonparametric test but could be modified to be employed as a goodness of fit test. In the special case of testing for normality of the distribution, data are standardized and compared with a standard normal distribution. The K-S test has this benefit to make no assumption about the distribution of data. The second test is Shapiro-Wilk (S-W test) that compares the ordered sample values with the corresponding order statistics from the specified distribution. The S-W test is most commonly used to assess a normal distribution.

Table 2 shows the results of the normality tests, which have approached to the different results. Similar conclusion was also mentioned by García et al. [33]. As a result, the parametric test condition is almost fulfilled with respect to the P values obtained from these two different tests. According to the S-W test, however, only some new results for BGA intended to achieve a better normal result are obtained, where the normality has not been satisfied.

Finally, passing these normality tests allows us to state with 90% confidence that the data fit the normal distribution with no significant departure from the normality.

5.1.2. Nonparametric Test. Since the research work deals with the real values in this study and based on the promising results from the previous section, the results of parametric tests seem enough for a reliable statistical analysis. For more evaluation of the all given algorithms' behaviour, the use of nonparametric statistics is also considered in addition to the parametric test. The performance of any algorithm with respect to the remaining ones is studied, and based on that, it has been determined if these results offer better performance for each one. Here, all results are reported with respect to the level of significance $\alpha = 5\%$.

Tables 3 and 4 represent the P values gained by applying post hoc method over the results of Quade and Friedman aligned tests using several different procedures, respectively. Using Quade test, the average ranks of RCGA, BGA, DE, PSO, and IWO are 3, 2, 1, 4.65, and 4.35, respectively. The rankings for the mentioned algorithms are 52.33, 47.13, 45.17, 127.30, and 105.57 using Friedman aligned test. Therefore, DE achieves the best rank (the minimum value) among these five algorithms. Assuming DE to be the reference algorithm, all hypotheses are rejected according to Tables 3 and 4. Nevertheless, BGA shows a good behaviour considering its P values in different procedures.

To compare an optional reference algorithm with others, Holm procedure driven P values of the post hoc multiple comparisons are used (Table 5). Since Holm's procedure rejects those hypotheses having a P value ≤ 0.05, all hypotheses in this table are rejected (all Holm P values are less than or equal to 0.05). That means that DE, BGA, and RCGA are better than PSO and IWO. Among the first three algorithms, the best and the worst algorithms are RCGA and DE, respectively. Also, PSO and IWO display almost the same performance giving a little bit positive concession to PSO.

Lastly, contrast estimation is obtained to calculate approximately the performance difference of the algorithms, two by two (Table 6). This comparison shows that DE outperforms other algorithms in terms of error rates, while PSO is the worst case in this regard.

5.2. Designing of Experiment. All the algorithms were run several times with the set of parameters based on the existing experience in order to find the best possible solution.

5.2.1. Comparison of NURBS and Bezier Tools. To begin with, because GA-based optimization is broadly used and more well-known than other methods employed in the paper and exhibited good performance in statistical analysis, this algorithm has been selected to make a reliable comparison between NURBS and Bezier curve fitting for airfoil shape

Table 1: The fitness values obtained from 30 independent runs.

	RCGA	BGA	DE	PSO	IWO
1	0.057	0.04081	0.037	0.8259	0.2795
2	0.0593	0.0401	0.037	0.0828	0.198
3	0.0739	0.04049	0.037	0.8142	0.3549
4	0.0537	0.042	0.037	1.8263	0.1485
5	0.0479	0.0467	0.037	1.1282	0.8118
6	0.0487	0.04077	0.037	0.9122	0.1485
7	0.072	0.04054	0.037	0.7141	0.1485
8	0.0514	0.03946	0.037	0.6595	0.866
9	0.064	0.04125	0.037	0.5901	1.4464
10	0.071	0.04348	0.037	0.8109	0.2169
11	0.0511	0.04127	0.037	0.7419	0.1485
12	0.0502	0.03894	0.037	0.1108	0.8994
13	0.0621	0.04075	0.037	1.7911	0.2255
14	0.0501	0.04204	0.037	0.7631	0.1889
15	0.0661	0.03943	0.037	0.245	0.1889
16	0.0493	0.03891	0.037	0.8892	0.1889
17	0.0534	0.03956	0.037	0.2679	0.8807
18	0.0511	0.03851	0.037	0.5432	0.7837
19	0.0522	0.03963	0.037	1.9922	0.2335
20	0.1003	0.04568	0.037	0.2032	0.1889
21	0.0592	0.04093	0.037	0.8074	0.1842
22	0.053	0.03999	0.037	0.7006	0.2008
23	0.0708	0.04271	0.037	0.7102	0.828
24	0.0624	0.03794	0.037	0.7437	0.2368
25	0.0627	0.04044	0.037	0.2942	1.5037
26	0.0645	0.04884	0.037	0.1837	1.1268
27	0.0579	0.04599	0.037	0.2674	1.1532
28	0.0531	0.04262	0.037	1.0189	0.1485
29	0.055	0.04305	0.037	1.4319	0.1775
30	0.1024	0.04206	0.037	1.717	0.2158

Table 2: Results of normality tests.

	GA	DE	PSO	IWO
P values based on K-S test	0.1066	0.1027	0.8102	0.5304
P values based on S-W test	0.0802	0.2507	0.8012	0.4304

parameterization. The best regulations for this optimization algorithm involve 120 chromosomes as the initial population selected from the whole solution space, and maximum number of generations is set to 300. Selection method is Roulette wheel, and probabilities of two-point crossover and mutation are set to 1.0 and 0.20, respectively. In order to make a fair comparison of the results, the termination criteria as well as all other parameters are set to be the same for both curve fitting tools. Finally, the real-number genes for design variables' definition with the method of least squares for evaluation of curve fitting fitness were successfully used as shown in the following illustrations.

Figure 5(a) depicts optimum parameterized curves and their polygons for the pressure side of the airfoil using Bezier and NURBS. This figure illustrates the good fitting with the measured point cloud of this segment. To limit the number of pages, other segments are not shown here, but the acceptable coincidence is the same. The results in general show that, for the same conditions, the NURBS curve has a better ability to be optimally fitted on a set of data points. In other words, the final objective value, that is, fitting error, of the curve fitting problem by using NURBS is noticeably less than the value for the Bezier curve fitting.

Another important characteristic that should be investigated is the local/global modification property. As shown in Figures 5(b) and 5(c), the shape of the entire curve represented using Bezier tool, contrary to NURBS tool, is affected by altering only one control point. This helpful characteristic of NURBS can be used in applications where minor modifications to the existing blade/airfoil shape should be considered during an optimization process. For instance, through the aerodynamics performance enhancement of an airfoil, the small and precise changes on the existing geometries are typically needed so that the local modification property of NURBS helps us in this regard. This is exactly what is wanted to be done in the next stage of the current research, that is, geometry optimization of a GT compressor

Table 3: Post hoc results by Quade comparison test (reference algorithm: DE).

i	Algorithms	$z = (R_0 - R_i)/\mathrm{SE}$	P	Holm Hochberg Hommel	Holland	Rom	Finner	Li
4	PSO	5.522134	0	0.0125	0.012741	0.013109	0.012741	0.045765
3	IWO	5.063582	0	0.016667	0.016952	0.016667	0.025321	0.045765
2	RCGA	3.02449	0.002491	0.025	0.025321	0.025	0.037739	0.045765
1	BGA	1.512245	0.130472	0.05	0.05	0.05	0.05	0.05

Table 4: Post hoc results by Friedman aligned comparison test (reference algorithm: DE).

i	Algorithms	$z = (R_0 - R_i)/\mathrm{SE}$	P	Holm Hochberg Hommel	Holland	Rom	Finner	Li
4	PSO	7.321863	0	0.0125	0.012741	0.013109	0.012741	0.007325
3	IWO	5.384422	0	0.016667	0.016952	0.016667	0.025321	0.007325
2	RCGA	0.63888	0.522901	0.025	0.025321	0.025	0.037739	0.007325
1	BGA	0.175321	0.860828	0.05	0.05	0.05	0.05	0.05

Table 5: P values of post hoc multiple comparisons.

i	Algorithms	z	Unadjusted P	Holm P
10	DE versus PSO	8.981462	0	0.005
9	DE versus IWO	8.164966	0	0.005556
8	BGA versus PSO	6.531973	0	0.00625
7	BGA versus IWO	5.715476	0	0.007143
6	RCGA versus DE	4.898979	0.000001	0.008333
5	RCGA versus PSO	4.082483	0.000045	0.01
4	RCGA versus IWO	3.265986	0.001091	0.0125
3	RCGA versus BGA	2.44949	0.014306	0.016667
2	BGA versus DE	2.44949	0.014306	0.025
1	PSO versus IWO	0.816497	0.414216	0.05

Table 6: Contrast estimation between averages of results considering all pairwise comparisons.

	RCGA	BGA	DE	PSO	IWO
RCGA	0	0.016	0.021	−0.619	−0.226
BGA	−0.016	0	0.005	−0.635	−0.243
DE	−0.021	−0.005	0	−0.64	−0.248
PSO	0.619	0.635	0.64	0	0.392
IWO	0.226	0.243	0.248	−0.392	0

airfoil/blade for better performance, where designers are not going to create geometry which is completely different from the primary original shape.

Table 7 shows a comparison between the effects of NURBS and Bezier properties on some implementation criteria. The evident influence of the number of CPs on the fitting error can be seen in the table; that is, the error drops when the number of CPs increases for both mathematical tools. While the degree of the NURBS curve is a predefined value independent of the number of CPs, the degree of the Bezier curve is increased with the rise in the number of CPs. Hence, the fitting error in Bezier representation drops more (from 137.6 mm to 61.7 mm) with an increasing number of CPs. Nevertheless, as implied earlier in Section 2 of the paper, when a curve of a complicated shape is represented by the Bezier curve, many CPs should be used. This in turn results in a higher degree Bezier curve. Higher degree functions may cause oscillation as well as increasing the computational burden.

Although the computational time of NURBS curve fitting is generally higher than that of Bezier curve fitting, the results in Table 2 show that the computational time of optimized parameterization with Bezier increases more than three times when the number of CPs increases from five to ten. This rise for NURBS representation is less than two times. This property is particularly important in this case because curves with so many CPs should be utilized due to the complexity of the current geometry. Therefore, independence of the curve degree from the number of CPs in NURBS causes a better possibility of its application into the curve/surface parameterization of complex shapes such as turbomachinery airfoils/blades.

As a result, the remaining implementation is done by using NURBS representation as the tool of geometry modeling in airfoil curve fitting.

5.2.2. Applying Optimization Algorithms. After a GA-based comparison of Bezier and NURBS properties and an investigation of their effects on airfoil geometry parameterization, the results obtained from all the developed optimization methods using NURBS representation are compared here in terms of static and dynamic convergences, final fitness values,

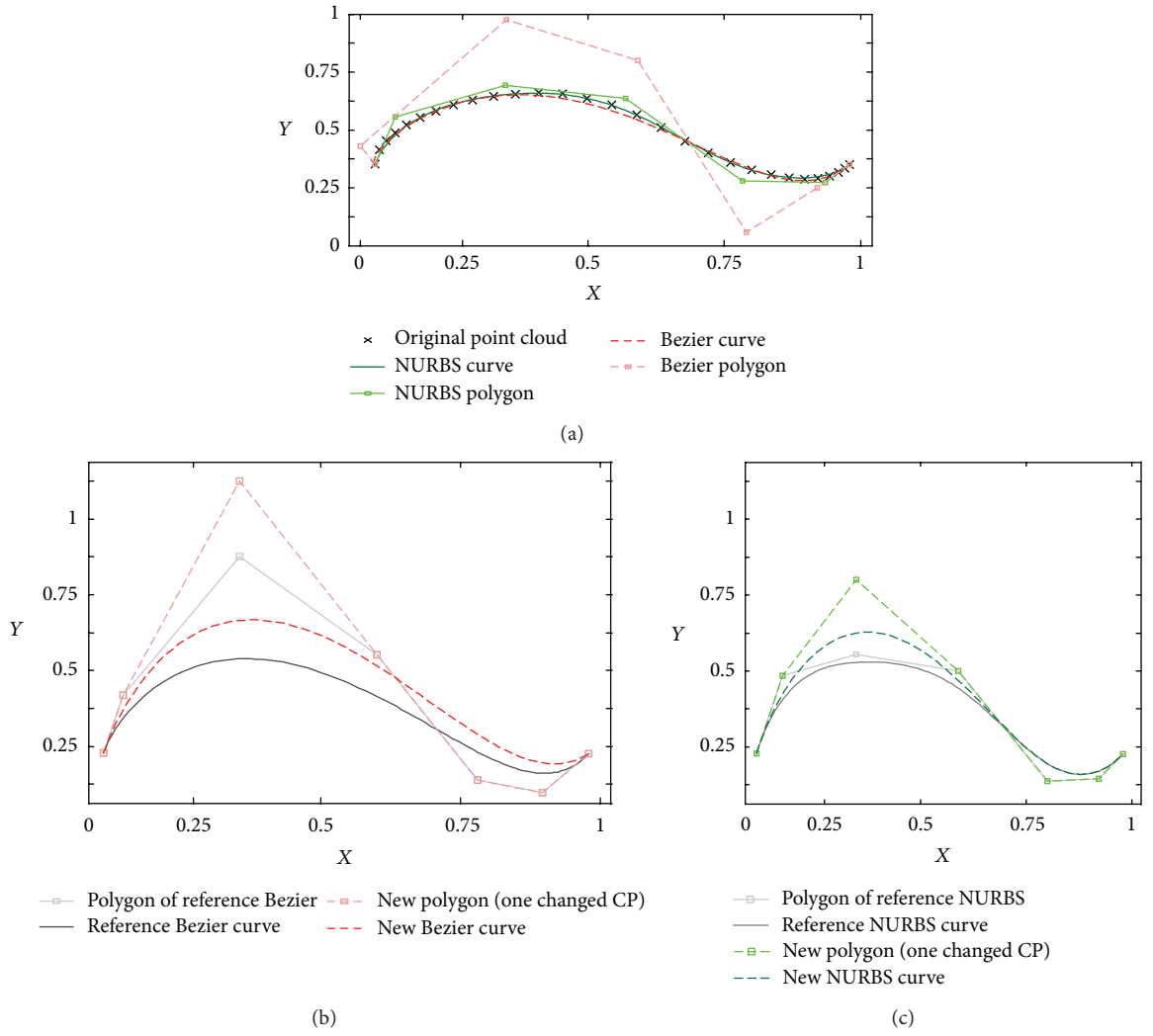

Figure 5: Optimum curve fitted using Bezier and NURBS methods (a) and global (b) and local (c) modification properties of Bezier and NURBS, respectively.

Table 7: Comparison of NURBS and Bezier properties' effect on some implementation criteria.

	5 control points			10 control points		
	Degree of curve	Fitting error [m]	Comp. burden [s]	Degree of curve	Fitting error [m]	Comp. burden [s]
Bezier	4	0.1376	60	9	0.0617	203
NURBS	3	0.0518	287	3	0.0393	475

and computing time. This is to see whether these methods can be successfully used in optimally solving these specific curve fitting problems. To the best knowledge of the authors, this is the first time that the DE and IWO algorithms have been profitably applied to a NURBS curve fitting optimization problem.

Table 8 shows the used parameters to adjust the algorithms in hand for this comparison. All optimization processes are terminated by the predefined number of iterations. Based on previous experience, the parameters of the algorithms are selected by trial-and-error. Furthermore, regarding PSO and IWO algorithms' parameters, the problem dimension should be set to 10 because of the specified number of NURBS CPs to 20 (i.e., forty coordinates) for all segments of the airfoil shape. This gives 10 design variables for each segment.

Taking the metaheuristic nature of the algorithms used in this part into account, RCGA, BGA, DE, PSO, and IWO are run several times. Because of rationally convergence of the algorithms, the final fitness values have been too close to each other for the unique objective function and parameters in several runs. However, all results presented in this paper

Table 8: Evolutionary and swarm algorithms' regulations for comparison task.

	RCGA	BGA	DE	PSO	IWO
Number of Iterations	250	250	250	250	250
Population/swarm/plant size	120	120	120	180	10
Selection method	Roulette wheel	Stochastic uniform	—	—	—
Crossover	Two-point, 0.90	Single-point, 0.95	0.5	—	—
Mutation	Uniform, 0.35	Uniform, 0.30	—	—	—
Scale factor	—	—	0.5		—
Self-confidence factor	—	—	—	1.5	—
Swarm confidence factor	—	—	—	1.5	—
Neighbor accel. factor	—	—	—	1.0	—
Velocity weight at the beginning	—	—	—	0.75	—
Velocity weight at the end	—	—	—	0.25	—
Number of plants	—	—	—	—	10
Min. Number of seeds	—	—	—	—	1
Max. Number of seeds	—	—	—	—	10
Modulation index	—	—	—	—	3
Standard deviation, initial	—	—	—	—	10
Standard deviation, final	—	—	—	—	0.001

Table 9: Static convergence of RCGA, BGA, DE, PSO, and IWO.

	Iteration number in which the best solution is achieved	Final objective value (sum of errors in millimeters)
RCGA	180	50.2
BGA	144	40.8
DE	135	37.7
PSO	181	203.2
IWO	94	148.8

Table 10: Computational efficiency (average of 30 runs).

	Real-coded GA	Binary GA	DE	PSO	IWO
CPU time (Sec)	240	218	216	14	23

are the mean values of those runs. Moreover, as addressed before, all results have only been illustrated for one of the segments (segment 2 according to Figure 1(a), bottom).

Figure 6 illustrates the convergence history for all given algorithms. As shown in this figure, all the algorithms converge to their optimal solutions reasonably by iterations. Nevertheless, the DE could achieve higher fitted solutions because its final fitness value is better than others. Table 9 represents the static convergence results for the proposed algorithms. Clearly, DE outperforms all other algorithms—except IWO—in terms of the static convergence behaviour because it converges faster. In other words, iteration number in which the best solution is achieved is lowest for IWO and highest for PSO. Generally, evolutionary algorithms noticeably show better performance in minimizing the fitted errors, while it is difficult to generally make the same conclusion or its inverse about the static convergence property.

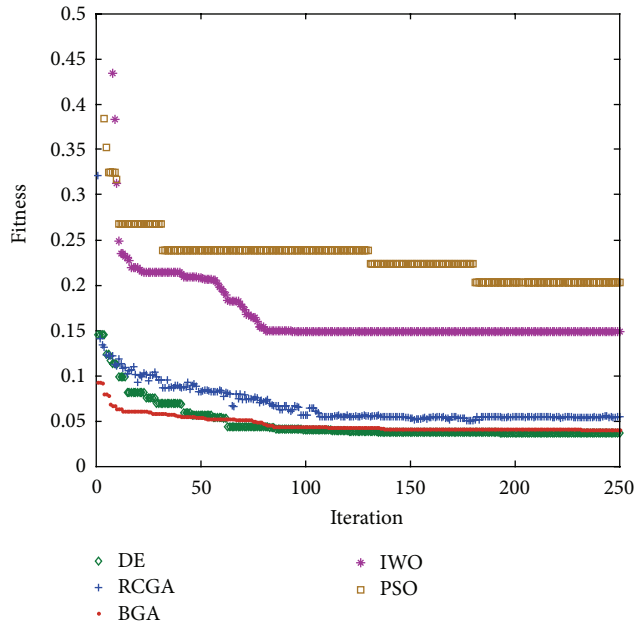

Figure 6: Convergence history of all given algorithms.

Additionally, a comparison between the dynamic convergence characteristics of each of the algorithms employed has been made in Figure 7 by plotting standard deviation of fitness values over all individuals in one generation. Dynamic convergence is indeed an indication of the diversification and robustness of the algorithms. This plot shows that the proposed IWO has the smoothest dynamic convergence due to its dual behaviour involving both mathematics-driven and population-based procedures. On the other hand, GA results show the highest fluctuations because of the mutation operation. In the case of PSO and DE solutions, the standard

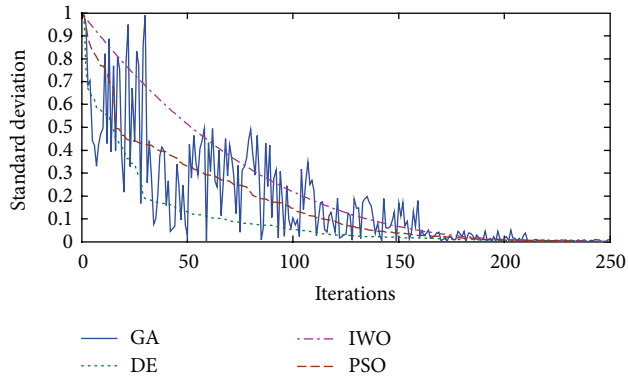

Figure 7: Dynamic convergence characteristics of evolutionary and swarm intelligence algorithms.

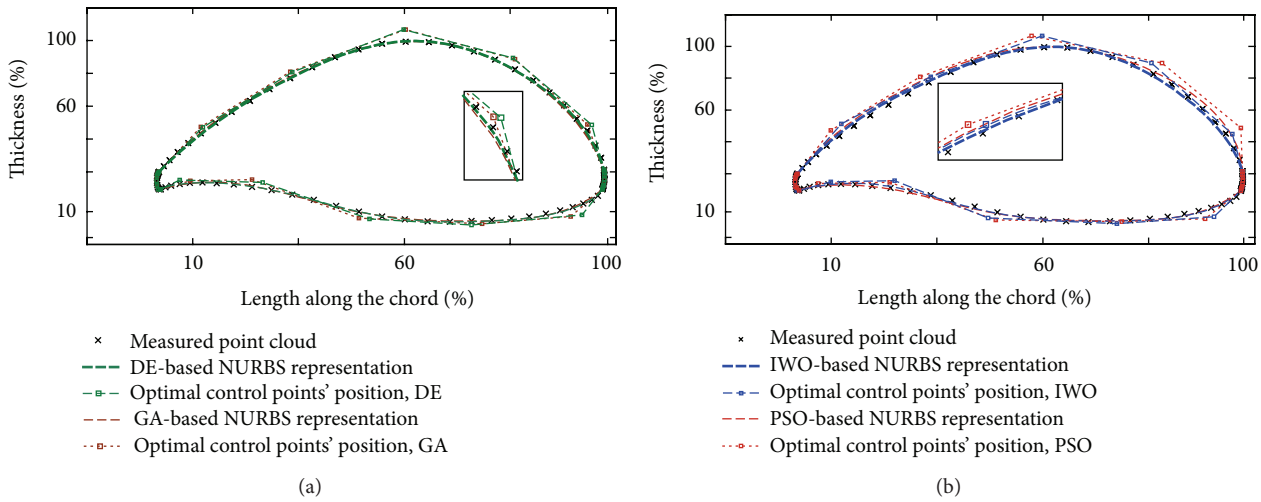

Figure 8: Airfoil shapes represented with optimal control points derived from evolutionary (a) and swarm intelligence (b) methodologies.

deviation variations decrease by iterations without fluctuation and decays to zero in the last iterations, like IWO.

Finally, in order to confirm the effectiveness of the planned methods to optimize the curve fitted to the airfoil shape of GT blades, 20 optimized CPs, knot vectors, and NURBS curves found by both evolutionary and swarm intelligence algorithms have been shown in Figure 8. This figure clearly illustrates a finer NURBS curve fitting on the given data points by evolutionary optimization techniques in comparison to the proposed swarm/colonized intelligence approaches. The illustration demonstrates that DE has the best performance in minimizing the fitting errors among all other algorithms. Also, it can be seen that the IWO-based NURBS curve shows better agreement with the original measured points in comparison to the PSO-based one. Nevertheless, it is worth mentioning that PSO implementation is easier, as the number of parameters is less in comparison with IWO. In other words, compared to IWO, one of the main advantages of PSO is that this algorithm is easily employed with few parameters to be adjusted.

Last but not least, in order to compare the computational efficiency of all four optimization algorithms used in this study, the CPU time is averaged over 30 independent runs for each method. The results are shown in Table 10. It can be understood that swarm methods demand considerably less computational effort and have huge saving time compared to the evolutionary ones. This would be crucial specifically when the dimensions of the fitting problem and/or its complexity are greater than those of the problem solved in this study (in turbomachinery, for instance, surface fitting in 3D geometry modeling of GT blades). Among these methods, however, the simplicity in the implementation of PSO results in saving computational efforts compared with IWO. The reason is that the functioning of IWO involves some mathematical/statistical procedures. Nonetheless, IWO can achieve its final fitness value in less iteration than that of the PSO (Table 9), and so it provides a huge saving in computational costs.

6. Conclusion

This paper presents the results of a study where a binary and real-coded genetic algorithm, a differential evolution, a particle swarm, and a newly developed invasive weed optimization methods were implemented using Bezier and NURBS tools

to optimally create the well-parameterized airfoil shapes on existing turbomachinery blades. The approach could be used for both precise reverse engineering and aerodynamic shape optimization purposes [35].

First, the touch trigger probes are employed to measure the point cloud of an existing gas turbine compressor blade sections. After that, the model construction is continued by segmentation of the measured points to four areas. Then, CAD modeling searches for the best CPs of Bezier and NURBS curves among the candidates by using the least-squares fitting technique in the objective function of the aforementioned optimization algorithms.

One important conclusion is that the NURBS curves, because of their special properties such as local modification and numerical stability, are certainly more suitable tools for such complex curve/surface fitting problem in turbomachinery than the Bezier curves. Regarding the planned optimization algorithms, the behavior of all the given evolutionary and swarm intelligence techniques is inclusively evaluated from both statistical and experimental points of view. As a result, GA and DE—as two of the most well-known evolutionary algorithms—as well as PSO and IWO—as the swarm intelligence methods—generally showed their ability in optimization of such complex curve fitting processes. Nevertheless, several significant behaviour differences are detected from a comprehensive comparison of their results. As a result of nonparametric test, it can be concluded that the evolutionary optimization algorithms obviously outperform the swarm intelligence ones. At the same time, while B/RC GA and DE demonstrate their superior performance in terms of reaching minimum fitting error, IWO has the best static convergence. IWO also shows the finest behaviour in dynamic convergence, in areas where GA results exhibit the highest fluctuations, and this is because of its mutation operator. Also, swarm intelligence optimization methods are observed to have a significantly better computational efficiency than evolutionary algorithms.

Lastly, since all the results obtained from the well-organized proposed optimization strategies demonstrate that the final optimum fitted curves have good agreement with the original measured points, selecting one among them to be employed in a computer-aided geometric design optimization process strongly depends on the final goals and priorities.

References

[1] J. Gräsel, A. Keskin, M. Swoboda, H. Przewozny, and S. André, "A full parametric model for turbomachinery blade design and optimisation," in *Proceedings of the ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 907–914, October 2004.

[2] W. Song and A. J. Keane, "A study of shape parameterisation methods for airfoil optimization," *American Institute of Aeronautics Astronautics*, 2006.

[3] P. Bezier, *Emploi des Machines a Commande Numerique*, Masson et Cie, Paris, France, 1970.

[4] G. E. Farin, *NURBS from Projective Geometry to Practical Use*, A. K. Peters Ltd., Natick, Mass, USA, 2nd edition, 1999.

[5] S. Pierret and C. Hirsch, "An integrated optimization system for turbomachinery blade shape design," in *Proceedings of the RTO AVT Symposium on Reduction of Military Vehicle Acquisition Time and Cost through Advanced Modeling and Virtual Simulation*, France, 2002.

[6] A. Oyama, M.-S. Liou, and S. Obayashi, "Transonic axial flow blade shape optimization using evolutionary algorithm and three-dimensional Navier-stokes solver," AIAA 2002-5642, 2002.

[7] F. Sieverding, B. Ribi, M. Casey, and M. Meyer, "Design of industrial axial compressor blade sections for optimal range and performance," *Journal of Turbomachinery*, vol. 126, no. 2, pp. 323–331, 2004.

[8] D. Büche, G. Guidati, and P. Stoll, "Automated design optimization of compressor blades for stationary, large-scale turbomachinery," in *Proceedings of the 2003 ASME Turbo Expo*, pp. 1249–1257, Atlanta, Ga, USA, June 2003.

[9] A. Shahrokhi and A. Jahangirian, "A surrogate assisted evolutionary optimization method with application to the transonic airfoil design," *Engineering Optimization*, vol. 42, no. 6, pp. 497–515, 2010.

[10] A. Kumar, *Robust design methodologies: application to compressor blades [Ph.D. thesis]*, University of Southampton, 2006.

[11] D. I. S. Adi, Š. M. Shamsuddin, and A. Ali, "Particle swarm optimization for NURBS curve fitting," in *Proceedings of the 6th International Conference on Computer Graphics, Imaging and Visualization: New Advances and Trends (CGIV '09)*, pp. 259–263, August 2009.

[12] Z. Jing, F. Shaowei, and C. Hanguo, "Optimized NURBS curve and surface fitting using simulated annealing," in *Proceedings of the International Symposium on Computational Intelligence and Design (ISCID '09)*, pp. 324–329, December 2009.

[13] P. Pandunata and S. M. H. Shamsuddin, "Differential Evolution optimization for Bezier curve fitting," in *Proceedings of the 7th International Conference on Computer Graphics, Imaging and Visualization (CGIV '10)*, pp. 68–72, August 2010.

[14] F. Yoshimoto, T. Harada, and Y. Yoshimoto, "Data fitting with a spline using a real-coded genetic algorithm," *Computer Aided Design*, vol. 35, no. 8, pp. 751–760, 2003.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[16] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the International Conference on Evolutionary Computation*, pp. 1945–1950, 1999.

[17] M. Dorigo, *Optimization, learning and natural algorithms [Ph.D. thesis]*, Politecnico di Milano, Milano, Italy, 1992.

[18] M. Montazeri and A. Safari, "Tuning of fuzzy fuel controller for aero-engine thrust regulation and safety considerations using genetic algorithm," *Aerospace Science and Technology*, vol. 15, no. 3, pp. 183–192, 2011.

[19] K. Price and R. Storn, "Differential evolution," *Dr. Dobb's Journal*, pp. 18–24, 1997.

[20] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355–366, 2006.

[21] P. Bezier, *The Mathematical Basis of the UNISURF CAD System*, Butterworths, London, UK, 1986.

[22] K. Lee, *Principles of CAD/CAM/CAE Systems*, Addison-Wesley Longman, 1999.

[23] M. G. Cox, "The numerical evaluation of B-spline," *Journal of the Institute of Mathematics and Its Applications*, vol. 15, pp. 95–108, 1972.

[24] C. de Boor, "On calculating with B-splines," *Journal of Approximation Theory*, vol. 6, pp. 50–62, 1972.

[25] W. Ma and J.-P. Kruth, "NURBS curve and surface fitting for reverse engineering," *International Journal of Advanced Manufacturing Technology*, vol. 14, no. 12, pp. 918–927, 1998.

[26] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[27] U. K. Chakraborty, *Advances in Differential Evolution*, Springer, 2008.

[28] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[29] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the IEEE Conference*, 2001.

[30] H. T. FeiGao, "Particle swarm optimization: an efficient method for tracing periodic orbits and controlling chaos," in *Proceedings of the International Conference on Complex Systems and Applications*, Watam Press, 2006.

[31] M. Montazeri, S. Jafari, and M. R. Ilkhani, "Application of particle swarm optimization in gas turbine engine fuel controller gain tuning," *Engineering Optimization*, vol. 44, no. 2, pp. 225–240, 2012.

[32] A. R. Mallahzadeh, H. Oraizi, and Z. Davoodi-Rad, "Application of the invasive weed optimization technique for antenna configurations," *Progress in Electromagnetics Research*, vol. 79, pp. 137–150, 2008.

[33] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[34] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[35] A. Safari, G. Lemu, and M. Assadi, "A novel combination of adaptive tools for turbomachinery airfoil shape optimization using a real-coded genetic algorithm," in *Proceedings of ASME Turbo Expo*, San Antonio, Tex, USA, 2013.