Struct Multidisc Optim (2012) 45:747–758
DOI 10.1007/s00158-011-0719-7

INDUSTRIAL APPLICATION

Multi-objective shape optimization of transonic airfoil sections using swarm intelligence and surrogate models

Miltiadis Kotinis · Amit Kulkarni

Received: 9 April 2011 / Revised: 17 August 2011 / Accepted: 18 September 2011 / Published online: 9 November 2011

Abstract In this article, the optimization problem of designing transonic airfoil sections is solved using a framework based on a multi-objective optimizer and surrogate models for the objective functions and constraints. The computed Pareto-optimal set includes solutions that provide a trade-off between maximizing the lift-to-drag ratio during cruise and minimizing the trailing edge noise during the aircraft's approach to landing. The optimization problem was solved using a recently developed multi-objective optimizer, which is based on swarm intelligence. Additional computational intelligence tools, e.g., artificial neural networks, were utilized to create surrogate models of the objective functions and constraints. The results demonstrate the effectiveness and efficiency of the proposed optimization framework when applied to simulation-based engineering design optimization problems.

Keywords Shape optimization · Transonic airfoil design · Multi-objective optimization · Surrogate models · Swarm intelligence

1 Introduction

The design of transonic airfoils for civil aviation applications has been a major engineering challenge in the last fifty years. The design process has been aided significantly by improvements in experimental methods and facilities, which led initially to the design of supercritical airfoils (Whitcomb 1974). These airfoils have good transonic behavior while maintaining acceptable performance at lower speeds. A description of all the development phases of the supercritical airfoils can be found in Harris (1990).

Advances in computational methods applicable to transonic flow (Jameson 1974; Drela and Giles 1987) have allowed engineers to analyze and evaluate candidate designs without the need for prototype construction and subsequent testing in the wind tunnel. Furthermore, the recent development of aerodynamic shape optimization techniques has provided designers with a tool to obtain airfoil shapes with the desirable performance characteristics. One of the first such methods, which is based on control theory, was developed by Jameson (1988). Recently, researchers have attempted to perform aerodynamic shape optimization using evolutionary algorithms (Jones et al. 1998; Giannakoglou 2002).

Aircraft noise has been a major source of noise pollution since the beginning of the utilization of jet engines in civil transport aircraft. The aircraft noise produced during take-off and landing causes considerable nuisance to the population living near the airports. Airframe noise, engine noise, and the noise generated due to the interference between engine and airframe comprise the main components of aircraft noise. In recent years, statutory limits regarding noise levels have been set by the International Civil Aviation Organization (ICAO 2008). At the same time, considerable research effort has been made towards reducing aircraft noise. For example, the introduction of the high-bypass ratio turbofan engine has significantly reduced the amount of engine radiated noise.

Airframe noise is generated through the interaction of turbulent flows with sharp-edged bodies (Howe 1978), e.g.,

M. Kotinis (✉) · A. Kulkarni
Department of Mechanical and Aerospace Engineering,
Old Dominion University, 238 Kaufman Hall, Norfolk,
VA 23529-0247, USA
e-mail: mkotinis@odu.edu

lifting surfaces. The highest levels of airframe noise occur while the aircraft is in the final approach to landing (Lilley 2001). In this case, the landing gear and high lift devices, e.g., trailing edge flaps and leading edge slats, are the main sources of airframe noise. The trailing edge noise is another source of airframe noise; it becomes the dominant source for a clean wing configuration, i.e., an aircraft with the high lift devices and the landing gear stowed.

If one of the design requirements is to obviate the need for high lift devices, the minimization of the trailing edge noise becomes a significant part of the aerodynamic shape optimization problem. The prediction of the trailing edge noise can be done through the computation of the far-field noise intensity per unit volume of acoustic sources at the trailing edge of a wing using the equation derived in Goldstein (1976) and later modified in Lilley (2001). A formulation which is easy to implement in a simulation-based optimization problem was later derived in Hosder et al. (2010).

In the current paper, the aerodynamic and aeroacoustic shape optimization problem is treated using a multi-objective approach. The goal is to design a transonic airfoil with low drag during cruise and low levels of trailing edge noise during the approach to landing. The results presented in Hosder et al. (2010) and Jouhaud et al. (2007) reveal that there is a trade-off between the aforementioned objectives, thus, there is no single globally optimal solution. For this reason, it was decided to solve this optimization problem using a multi-objective approach. During the last fifteen years, various design methodologies have been utilized by researchers to solve aerospace design optimization problems (Sobieszczanski-Sobieski and Haftka 1997; Jones et al. 1998; Leifsson et al. 2006).

The flow analysis was performed using computational fluid dynamics (CFD). The CFD results obtained at specific points within the design variable space were subsequently utilized in order to build surrogate models for the objective functions and constraints of the optimization problem. In this way, the computational time required to solve the problem could be significantly reduced (Yang et al. 2002; Jin 2005; Wang et al. 2011).

The main objective of the current paper is to demonstrate the effectiveness and efficiency of an optimization framework based on swarm intelligence and surrogate models when applied to a multi-objective optimization problem. The general definition of a multi-objective optimization problem, the multi-objective optimizer, and the surrogate models utilized in this research project are discussed in Section 2. The theoretical background of the trailing edge noise calculation is presented in Section 3. The shape optimization procedure is described in Section 4. The optimization results are presented in Section 5, in addition to a comparison between selected Pareto-optimal airfoil sections. Finally, conclusions and some directions for future research are provided in Section 6.

2 Components of multi-objective design optimization framework

2.1 Multi-objective optimization

The general multi-objective optimization problem, assuming minimization of the objective functions, can be stated as follows:

$$\text{Minimize} \quad f_i(\mathbf{x}), \quad i = 1, \ldots, I \tag{1}$$

$$\text{subject to} \quad g_m(\mathbf{x}) \leq 0, \quad m = 1, \ldots, M \tag{2}$$

$$h_n(\mathbf{x}) = 0, \quad n = 1, \ldots, N \tag{3}$$

where

$$\mathbf{x}\ (x_1, x_2, \ldots, x_D), x_{jl} \leq x_j \leq x_{ju}, j = 1, \ldots, D.$$

In unconstrained multi-objective optimization problems, a solution vector A dominates another solution vector B, if and only if the following two conditions for Pareto dominance are satisfied:

- The objective vector that corresponds to solution A ($\mathbf{x}_A$) is no worse than the objective vector that corresponds to solution B ($\mathbf{x}_B$) in all objectives.
- Solution A is strictly better than solution B in at least one objective.

If $\mathbf{x}_A$ is not dominated by any other solution, it is called a Pareto-optimal solution. The corresponding objective function vector belongs to the set of vectors that comprise the Pareto front. In constrained problems, $\mathbf{x}_A$ constraint-dominates $\mathbf{x}_B$ if any of the following conditions are satisfied (Deb 2001):

- Both $\mathbf{x}_A$ and $\mathbf{x}_B$ are feasible and $\mathbf{x}_A$ dominates $\mathbf{x}_B$, based on the aforementioned conditions for Pareto dominance.
- Both $\mathbf{x}_A$ and $\mathbf{x}_B$ are infeasible but $\mathbf{x}_A$ has a smaller constraint violation.
- Solution $\mathbf{x}_A$ is feasible and solution $\mathbf{x}_B$ is not.

The constraints listed in (2) and (3) can have a significant impact on the Pareto front and the corresponding Pareto-optimal set. Parts of the unconstrained Pareto front, or in some cases the entire unconstrained Pareto front, might become infeasible in the presence of constraints. Furthermore, constraints might be active in different regions of the search space. A multi-objective optimizer needs to be capable of handling the additional search requirements of a constrained optimization problem.

2.2 Description of the ACMOPSO algorithm

The Pareto-optimal solutions of the multi-objective design problem were obtained using the Adaptive Coevolutionary Multi-Objective Particle Swarm Optimizer (ACMOPSO), which was presented in Kotinis (2011). ACMOPSO explores the design variable space using the search mechanism of PSO combined with random mutation. The swarm is divided into a number of equi-sized smaller swarms at the end of each algorithmic iteration; each of these swarms focuses on a specific region of the computed Pareto front, which consists of the non-dominated solutions stored in an external archive. The latter is divided into a number of segments equal to the number of swarms. Two members of the Pareto-optimal set within each segment are randomly selected to act as leaders of the kth swarm particle for each design variable j:

$$v_{kj}(t+1) = w \cdot v_{kj}(t) + c_1 \cdot (x_{ndlr1,j}(t) - x_{kj}(t)) + c_1 \cdot rand_j(0,1) \cdot (x_{ndlr2,j}(t) - x_{kj}(t)) \quad (4)$$

where v_{kj} is the movement of particle k within the range of j, $x_{ndlr1,j}(t)$ and $x_{ndlr2,j}(t)$ are the jth position coordinates of the two randomly selected leaders, $x_{kj}(t)$ is the corresponding jth position coordinate of the kth particle, w is the inertia weight, c_1 is the social coefficient, and $rand_j(0,1)$ is a random number uniformly distributed in (0,1). The kth particle's position is updated in every iteration $(t+1)$ by adding the velocity vector over a single time increment to the current position vector:

$$\mathbf{x}_k(t+1) = \mathbf{x}_k(t) + \mathbf{v}_k(t+1) \quad (5)$$

This procedure is followed for 80% of the swarm particles. The remaining particles are substituted with randomly selected non-dominated archived solutions, which are subsequently mutated with a mutation rate $p_{mut} = 10\%$. The inclusion of a mutation operator, combined with particle substitution, enhances the algorithm's capability to solve problems with multiple local Pareto fronts, and also counterbalances the high selective pressure due to the utilization of a Pareto ranking procedure to manage the external archive. It needs to be mentioned that if the number of non-dominated solutions exceeds the nominal capacity of the archive, the crowding distance operator (Deb et al. 2000) is employed to maintain the solutions that reside in the least crowded areas.

The values of the algorithmic parameters, w and c_1, are adapted on-line for each swarm using the feedback provided by two metrics. The first metric takes into account the effectiveness of each swarm in producing new non-dominated solutions during the current iteration. The second metric represents the overall ability of the swarms to produce solutions capable of entering the external archive.

In the current research project, a parallelized version of ACMOPSO with five swarms was utilized. It was shown in Kotinis (2011) that the utilization of five swarms, in most cases, results in very fast convergence to the true Pareto front. The parallelization process, which involves the computation of the objective functions and constraints, was performed using the OpenMP (Chapman et al. 2008) interface, and specifically a work-sharing *parallel do* construct. The results reported in Kotinis (2011) regarding an application of the parallelized version of ACMOPSO to an engineering design optimization problem demonstrated near-linear speedup and high parallel efficiency.

2.3 Surrogate models

The approximation of the objective functions and constraints can lead to significant savings in terms of both time and resources utilized in the optimization process. However, it is of the utmost importance to provide surrogate models that are able to generalize well, i.e., provide accurate predictions when presented with previously unseen data. It was decided to compare the effectiveness of different surrogate models for each objective and constraint function of the optimization problem, first, and then select the most appropriate model for each function. For this purpose, the University of Waikato's open source data mining software WEKA (Hall et al. 2009) was utilized. WEKA provides the user with the capability to evaluate and compare different metamodeling techniques through a very user-friendly interface. Three such techniques were selected and utilized: Artificial neural networks (ANNs), support vector regression (SVR), and linear regression. The effectiveness of each metamodel was assessed using ten-fold cross-validation. A brief description of ANNs and SVR is provided in the following paragraphs. The selection of the aforementioned techniques (ANNs and SVR) was based on a preliminary extensive investigation, which showed that these two methods had the best overall performance in a number of benchmark problems. Linear regression was also tested and utilized wherever its performance was better or even comparable to ANNs and SVR, as it provides a significantly less complex metamodeling solution.

2.3.1 *Artificial neural networks*

Artificial neural networks have been used extensively across many fields of scientific research. A review of ANN applications and history can be found in Eberhart and Shi (2007). The primary reason for the use of ANNs is their universal approximation capability given an adequate number of hidden nodes. It was proven in Cybenko (1989) that a network with linear output nodes and a single hidden layer of sigmoid nodes is sufficient for uniform approximation of any continuous function in the unit hypercube. Although, there are many types and topologies of ANNs, it was decided to utilize a feedforward fully-connected network consisting of an input layer, a single hidden layer with varying number of sigmoid neurons (nodes), and a linear output layer. A typical feedforward ANN is shown in Fig. 1. The ANNs were trained in WEKA using the backpropagation algorithm (Rumelhart et al. 1986) for a fixed number of epochs (training iterations). Assuming I input nodes, J hidden nodes, a bias node added to the input $(I+1)$ and to the hidden layer $(J+1)$, and a single output node, the equation that provides the network prediction y can be written as follows:

$$y = \sum_{j=1}^{J+1} w_j f\left(\sum_{i=1}^{I+1} u_{ji} x_i\right) \tag{6}$$

where u_{ji} is the weight of the connection between input node x_i and hidden node h_j, w_j is the weight of the connection between hidden node h_j and the output node, and f is the activation function, which for this specific application corresponds to the sigmoid function:

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{7}$$

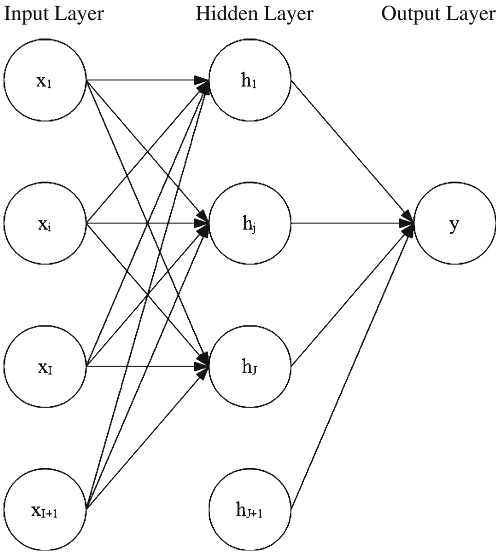

Fig. 1 Feedforward ANN with a single hidden layer

2.3.2 *Support vector regression*

Support vector regression (SVR) is a machine learning technique based on the support vector machine (SVM) algorithm developed at AT&T Bell Laboratories by Vapnik and co-workers (Boser et al. 1992). SVR utilizes a nonlinear mapping function, typically based on kernels, in order to map the data from the input variable space to the feature space. Given a data set of training points (features), the algorithm attempts to provide a regression function where the predicted values of the training points are within a specified distance ε from the actual values and also the function is as flat as possible. Because such a function might not exist, slack variables are introduced to the problem to deal with distances larger than ε. The trade-off between flatness and the amount up to which deviations larger than ε are accepted is controlled by a parameter C. A review of SVR can be found in Smola and Schölkopf (2004). The SVR model utilized in WEKA is based on the algorithm presented in Shevade et al. (2000). For the purpose of the current research, the mapping was performed using a kernel based on the universal Pearson VII function (PUK) as suggested in Üstün et al. (2006). The SVR prediction is given through a regression function f as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{n} (a_i - a_i^*) k(\mathbf{x}, \mathbf{x}_i) + b \tag{8}$$

where a_i and a_i^* are the Lagrange multipliers, $0 \leq a_i, a_i^* \leq C$, and $\mathbf{x}_i$ is a training point. The PUK kernel function $k(\mathbf{x}, \mathbf{x}_i)$ is given by the equation:

$$k(\mathbf{x}, \mathbf{x}_i) = \frac{1}{\left[1 + \left(\frac{2\sqrt{\|\mathbf{x}-\mathbf{x}_i\|^2}\sqrt{2^{1/\omega}-1}}{\sigma}\right)^2\right]^{\omega}} \tag{9}$$

where the parameters σ and ω control the shape of the function.

3 Trailing edge noise metric

The basis for the noise metric formulation is the expression derived in (Goldstein 1976). For low Mach numbers, the far-field noise intensity per unit volume of acoustic sources is formulated as:

$$I = \frac{\rho_\infty}{2\pi^3 c_\infty^2} \omega_0 u_0^4 \cos^3 \beta \frac{D(\theta, \phi)}{H^2} \tag{10}$$

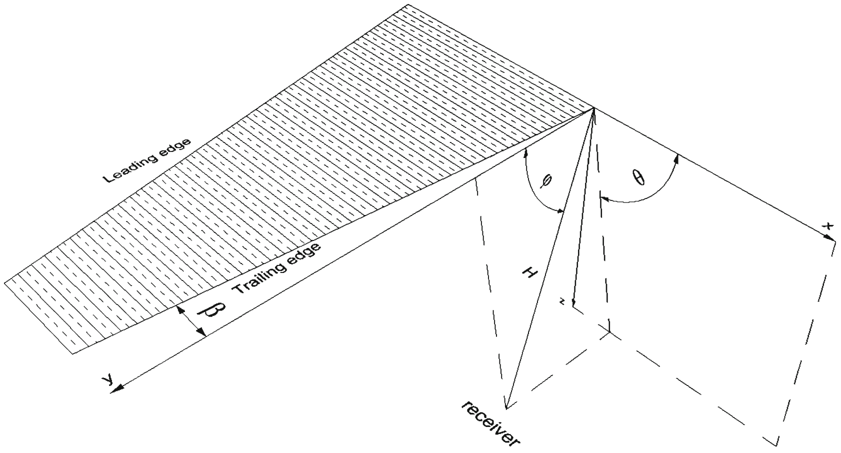

Fig. 2 Parameters used in the definition of the far-field noise intensity

where ρ_∞ is the freestream density, c_∞ is the freestream speed of sound, ω_0 is the characteristic source frequency, u_0 is the characteristic velocity scale of turbulence, H is the distance to the ground (receiver), β is the trailing edge sweep angle, $D(\theta,\phi)$ is the directivity term, and θ and ϕ are the polar and azimuthal directivity angles (shown in Fig. 2), respectively.

The directivity term is equal to $2\sin^2(\theta/2)\sin\phi$. For most conventional wings the sweep angle term can be neglected. Assuming a constant value of the Strouhal relation for turbulent flow $((\omega_0\ell_0)/u_0)$, as proposed in (Lilley 2001), we can rewrite (10) as:

$$I \approx \frac{\rho_\infty}{2\pi^3 c_\infty^2}\ell_0^{-1}u_0^5\frac{D(\theta,\phi)}{H^2} \tag{11}$$

where ℓ_0 is the characteristic length scale of turbulence. Using the following expression (Hosder et al. 2010) for a correlation volume per unit span at the trailing edge:

$$dV = \ell_0^2 dy \tag{12}$$

we can integrate (11) over the wing span, b, to obtain the expression for the noise intensity indicator:

$$I_{NI} = \frac{\rho_\infty}{2\pi^3 c_\infty^2}\int_0^b \ell_0 u_0^5\frac{D(\theta,\phi)}{H^2}dy \tag{13}$$

The noise intensity metric that was proposed in Hosder et al. (2010) models the characteristic turbulent velocity at a spanwise location of the wing trailing edge as equal to the maximum value of the turbulent kinetic energy (TKE) profile at that particular spanwise location:

$$u_0 = \max\left(\sqrt{TKE(n)}\right) \tag{14}$$

where n is the direction normal to the wing surface. Similarly, the characteristic length scale of turbulence at a spanwise location can be modeled as:

$$\ell_0 = \frac{\max\left(\sqrt{TKE(n)}\right)}{\omega} \tag{15}$$

where ω is the turbulence frequency (specific dissipation rate of TKE) observed at the position of maximum TKE at the spanwise location under consideration. The noise intensity indicator can be scaled using a reference value of the noise intensity ($I_{ref} = 10^{-12}$ W/m^2) in order to obtain the following metric for the trailing edge noise (in dB):

$$NM(\text{dB}) = 10\log_{10}\frac{I_{NI}}{I_{ref}} = 120 + 10\log_{10} I_{NI} \tag{16}$$

It needs to be mentioned that only the upper surface of the airfoil was taken into consideration when computing the noise metric, as the contribution of the lower surface boundary layer to the far-field radiated trailing edge noise is generally minimal; this was also confirmed by the CFD results.

4 Multi-objective shape optimization of transonic airfoil sections

The flowchart of the optimization framework is depicted in Fig. 3. In the following sections, each step is explained and details regarding its implementation are provided.

4.1 Selection of operating points

The aircraft is assumed to be at an altitude of 120 m in the approach condition with the receiver located directly below the aircraft, where the noise certification measurements are taken (ICAO 2008). The directivity angles in (13) are set

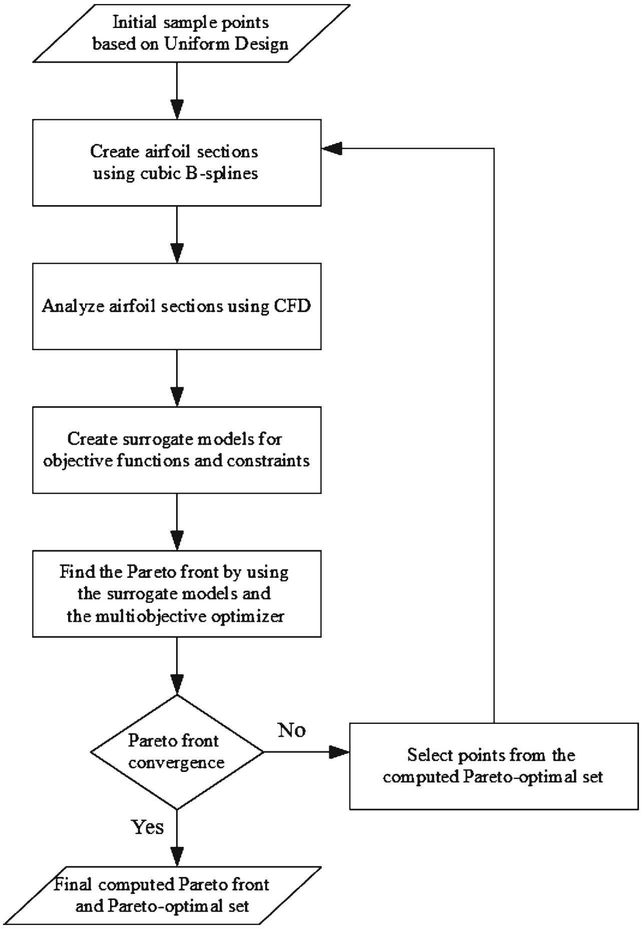
Initial sample points based on Uniform Design
Create airfoil sections using cubic B-splines
Analyze airfoil sections using CFD
Create surrogate models for objective functions and constraints
Find the Pareto front by using the surrogate models and the multiobjective optimizer
Pareto front convergence
No
Select points from the computed Pareto-optimal set
Yes
Final computed Pareto front and Pareto-optimal set

Fig. 3 Flowchart of optimization framework

equal to 90°. Therefore $H = 120$ m and $D(\theta, \psi)$ is equal to unity. During cruise the corresponding altitude is 35,000 ft. The mean aerodynamic chord (*mac*) value was assumed to be equal to 5.67 m, a value typical of transport aircraft. The Mach number was set equal to 0.80 and 0.215 for cruise and approach, respectively. The corresponding Reynolds number values, based on *mac*, are $35.6 \cdot 10^6$ and $28.1 \cdot 10^6$. The angle of attack was assumed to be equal to 2.5° during cruise and 10° during approach.

4.2 Formulation of the multi-objective optimization problem

The goal is to find the airfoil sections that correspond to the solution of the following shape optimization problem: Minimize the noise metric given in (16) during approach (A) and maximize the lift-to-drag ratio during cruise (C). Constraints were imposed on the pitching moment coefficient value during cruise, the lift coefficient value during approach, and the maximum thickness-to-chord ratio of the section. It needs to be mentioned that the RAE-2822 airfoil was used as a baseline design in the optimization procedure. The reason for selecting a subsonic airfoil as a baseline design, i.e., instead of a supercritical airfoil, was to avoid biasing the search towards shapes with good performance in one objective. The problem was formulated as follows:

$$\text{Minimize}: f_1(\mathbf{x}) = (NM)_A \tag{17}$$

$$\text{Maximize } f_2(\mathbf{x}) = (C_L/C_D)_C \tag{18}$$

$$\text{subject to}: g_1(\mathbf{x}) = (C_{M@0.25c})_C \geq -0.11 \tag{19}$$

$$g_2(\mathbf{x}) = (C_L)_A \geq 1.10 \tag{20}$$

$$g_3(\mathbf{x}) = (t/c)_{\max} \geq 0.105 \tag{21}$$

where C_L is the lift coefficient, C_D is the drag coefficient, $C_{M@0.25c}$ is the pitching moment coefficient about the quarter-chord position, and $t/c$is the thickness ratio of the section. The vector of fourteen design variables, $\mathbf{x}$, is defined as:

$$\mathbf{x}\ (x_1, x_2, ..., x_{14}), x_{jl} \leq x_j \leq x_{ju}, j = 1, ..., 14.$$

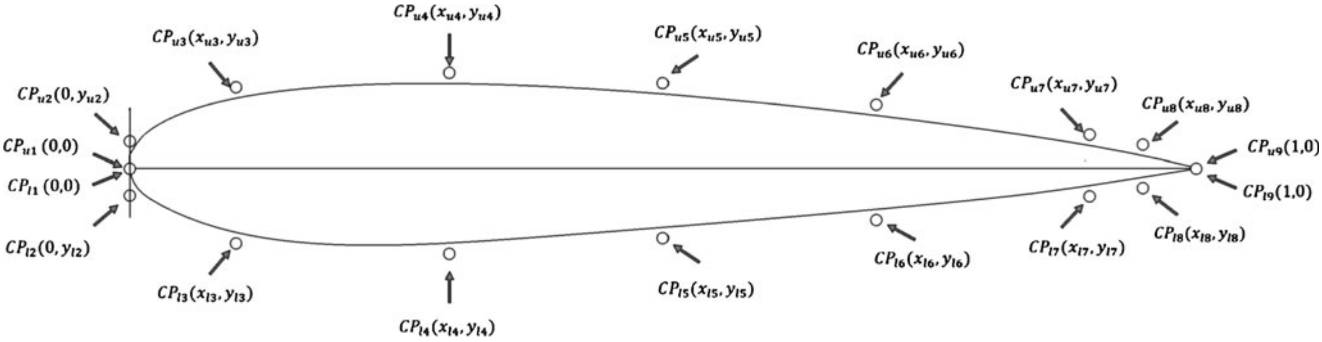
$CP_{u4}(x_{u4}, y_{u4})$
$CP_{u3}(x_{u3}, y_{u3})$
$CP_{u5}(x_{u5}, y_{u5})$
$CP_{u6}(x_{u6}, y_{u6})$
$CP_{u7}(x_{u7}, y_{u7})$
$CP_{u8}(x_{u8}, y_{u8})$
$CP_{u2}(0, y_{u2})$
CP_{u1} (0,0)
CP_{l1} (0,0)
CP_{u9}(1,0)
CP_{l9}(1,0)
$CP_{l2}(0, y_{l2})$
$CP_{l8}(x_{l8}, y_{l8})$
$CP_{l7}(x_{l7}, y_{l7})$
$CP_{l6}(x_{l6}, y_{l6})$
$CP_{l3}(x_{l3}, y_{l3})$
$CP_{l5}(x_{l5}, y_{l5})$
$CP_{l4}(x_{l4}, y_{l4})$

Fig. 4 Control points of the cubic B-splines

Table 1 Values of x-coordinate of control points

x_{l2}	0.000	x_{u2}	0.000
x_{l3}	0.100	x_{u3}	0.100
x_{l4}	0.250	x_{u4}	0.300
x_{l5}	0.400	x_{u5}	0.500
x_{l6}	0.650	x_{u6}	0.700
x_{l7}	0.850	x_{u7}	0.900
x_{l8}	0.950	x_{u8}	0.950

4.3 Parameterization of the airfoil shape

The airfoil shape was parameterized using two cubic B-splines; one for the upper and one for the lower airfoil surface. The B-splines are generated by utilizing nine control points for each surface as shown in Fig. 4. Assuming that the chord length is equal to one, the first control point, which is common to both surfaces, is placed on the leading edge with coordinates (0,0). The last control point is placed on the trailing edge with coordinates (1,0). Seven additional control points are placed at intermediate positions on each surface. The x-coordinates of these points, which are listed in Table 1, are kept fixed and the y-coordinates are allowed to vary in the ranges displayed in Table 2. The y-coordinates of the fourteen control points form the vector of design variables $\mathbf{x}$; the ranges listed in Table 2 correspond to the boundary constraints of the optimization problem. These ranges were based on the values that were used to generate the baseline design (RAE-2822 airfoil).

Table 2 Ranges of y-coordinate of control points

Design variable	Lower limit	Upper limit
y_{l2}	−0.0230	−0.0110
y_{l3}	−0.0515	−0.0375
y_{l4}	−0.0660	−0.0490
y_{l5}	−0.0750	−0.0560
y_{l6}	−0.0400	−0.0230
y_{l7}	−0.0075	−0.0045
y_{l8}	−0.0020	0.0050
y_{u2}	0.0110	0.0230
y_{u3}	0.0375	0.0515
y_{u4}	0.0530	0.0700
y_{u5}	0.0560	0.0750
y_{u6}	0.0430	0.0600
y_{u7}	0.0140	0.0260
y_{u8}	0.0055	0.0125

4.4 Design of experiments

The optimization problem was solved using surrogate models of the objective functions and constraints. These models were developed and trained using a data set of points selected from within the design variable space. For this purpose, an experimental design methodology was utilized. Uniform design (Fang 1980; Fang et al. 2000), which is a kind of fractional factorial design, has been shown (Simpson et al. 2001) to be a very effective sampling strategy even when small data sets are available. Therefore, it was decided to use a $U_{90}(9^{14})$ uniform design for 90 runs with fourteen factors (design variables) each having nine discrete levels. Given the continuous nature of the design variables, each one of the ranges listed in Table 2 was discretized into nine equi-spaced values. The 90 airfoil sections were analyzed using the ANSYS® FLUENT® software as described in Section 4.5. The results were combined to create five datasets; one for each objective and constraint function.

4.5 Analysis of performance of airfoil sections

The analysis of the steady flow around the airfoils in both cruise and approach conditions was performed in FLUENT®, a general-purpose CFD software, at the operating points described in Section 4.2. The flow domain is shown in Fig. 5. A C-type grid was created with 54,000 cells (540 cells in the streamwise direction and 100 cells in the normal direction). The airfoil shapes that were generated using B-splines were analyzed by first creating files containing discrete data points on the airfoil surface. These files were imported into GAMBIT®, a general-purpose preprocessor for CFD analysis, where the mesh of the flow domain was created. The flow analysis was performed by solving the governing equations for mass, momentum, and energy using the density-based implicit numerical solver of FLUENT®, which is based on a finite volume method with the flow properties calculated at the cell centers. The convection terms were discretized using a second-order upwind scheme and the diffusion terms with a central-differencing scheme. Gradients were computed using a least squares cell-based method. The inviscid fluxes were calculated using the Roe flux-difference splitting scheme. Turbulence was modeled using the shear-stress transport (SST) model (Menter 1994). At least ten cells in the direction normal to the airfoil surface were placed within the turbulent boundary layer in order to provide adequate flow resolution in that region. A pressure far-field boundary condition was imposed on the flow boundaries, where the turbulence intensity value was set equal to 0.1%.

The CFD simulations for each airfoil were terminated when the scaled residuals of all the governing equations

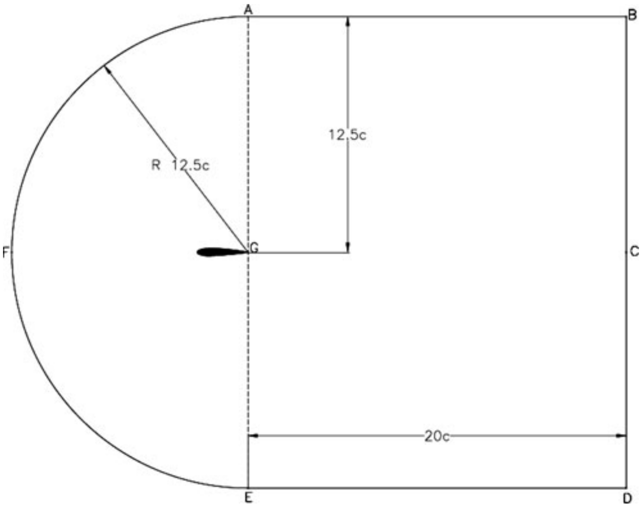

Fig. 5 Flow domain for CFD analysis

had been reduced by at least five orders of magnitude, provided that the lift and drag coefficients had also converged. The values of the lift, drag, and pitching moment coefficients were recorded and the profiles of the turbulent kinetic energy and the specific dissipation rate at the trailing edge were extracted. These were subsequently analyzed to find the location of maximum turbulent kinetic energy and the value of the specific dissipation rate at that location.

4.6 Surrogate models

Using the results obtained from the flow analysis, surrogate models for the objective and constraint functions were generated in WEKA. Instead of using a single type of surrogate model, it was decided to evaluate a number of different models for each function. Specifically, the models that were included in the evaluation process were the following: Feedforward ANNs with a single hidden layer and number of hidden nodes varying between two and ten, SVR with the PUK kernel, and linear regression. The model evaluation was performed using ten-fold cross-validation on the data set of 90 points using the root mean squared error as the evaluation metric. Different parameter values for each model were tested. The surrogate model with the best performance for each function was utilized by the optimization algorithm.

4.7 Running the ACMOPSO optimizer

ACMOPSO was run for 100 iterations per optimization cycle using five swarms with twenty particles each. At the end of the first cycle, as shown in Fig. 3, five points were selected from the computed Pareto front and the corresponding airfoil shapes were generated using B-splines and subsequently analyzed in FLUENT®. The results were added to the existing data set and the surrogate models were reconstructed. This iterative process was repeated until the computed Pareto front showed minimal changes between two successive iterations. A total of five optimization cycles was required in order to obtain the final Pareto front.

Table 3 Comparison between surrogate models

Objective/constraint	Surrogate model	RMSE
$(NM)_A$	ANN (5 hidden nodes, 500 training epochs)	**0.6721**
	SVR (PUK kernel, $C = 2.1$, $\sigma = 4.1$, $\omega = 2.5$)	1.6171
	Linear regression	1.8378
$(C_L/C_D)_C$	ANN (5 hidden nodes, 500 training epochs)	0.8638
	SVR (PUK kernel, $C = 2.5$, $\sigma = 5.0$, $\omega = 1.0$)	**0.7526**
	Linear regression	1.1084
$(C_{M@0.25c})_C$	ANN (4 hidden nodes, 500 training epochs)	0.0059
	SVR (PUK kernel, $C = 2.0$, $\sigma = 2.7$, $\omega = 1.6$)	0.0049
	Linear regression	**0.0049**
$(C_L)_A$	ANN (2 hidden nodes, 500 training epochs)	**0.0130**
	SVR (PUK kernel, $C = 1.5$, $\sigma = 4.0$, $\omega = 2.0$)	0.0214
	Linear regression	0.0225
t/c	ANN (4 hidden nodes, 500 training epochs)	0.0008
	SVR (PUK kernel, $C = 1.0$, $\sigma = 5.0$, $\omega = 2.0$)	0.0005
	Linear regression	**0.0005**

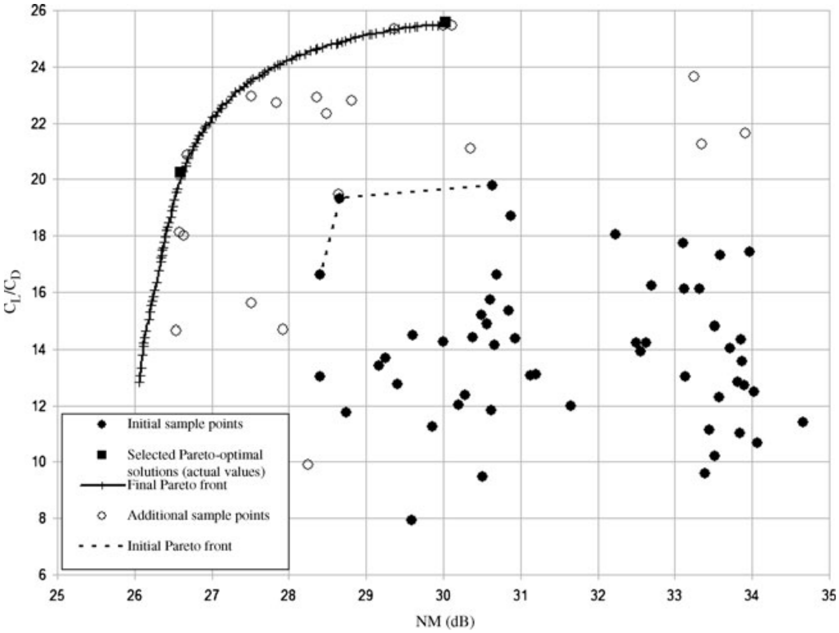

Fig. 6 Multi-objective optimization results

5 Multi-objective shape optimization results and discussion

The representation in the objective function space of a subset of the initial data set of 90 sample points is depicted in Fig. 6. Some of these points correspond to infeasible solutions. The Pareto front consists of only three members. During the iterative optimization process, a total of twenty new points, as shown in Fig. 6, were evaluated and added to the existing data set. In the same figure, the final Pareto front is also depicted. The effectiveness of the iterative optimization procedure is demonstrated by the fact that the optimal values in both objectives, i.e., at the extreme points on the Pareto front, were significantly improved compared to the solutions in the initial data set. The surrogate models that were used to approximate the objective functions enabled the optimizer to converge to the final Pareto front in only five iterations. Even though, there is no guarantee that this is the global Pareto front, the improvement in the objective function values is definitely noteworthy considering that it was achieved with only twenty additional problem evaluations. A comparison between the surrogate models that were tested and the corresponding optimal parameter values is displayed in Table 3. The optimal parameter values were obtained manually. Minimization of the root mean squared error (RMSE) value, which was obtained using ten-fold cross-validation, was utilized as the selection criterion. The RMSE values of the models that were selected as surrogates of the problem objectives/constraints in order to produce the final Pareto front are listed in bold font.

5.1 Comparison between two Pareto-optimal solutions

It was decided to compare two solutions from the final Pareto-optimal set. The first Pareto-optimal solution (airfoil #1) that was selected was the computed solution with the maximum lift-to-drag ratio value. The second Pareto-optimal solution (airfoil #2) was chosen from the 'knee' section, which contains solutions that provide a balanced trade-off between high lift-to-drag ratio and low noise metric values. The shapes of the airfoil sections are displayed in Fig. 7. The main geometric characteristics of the two

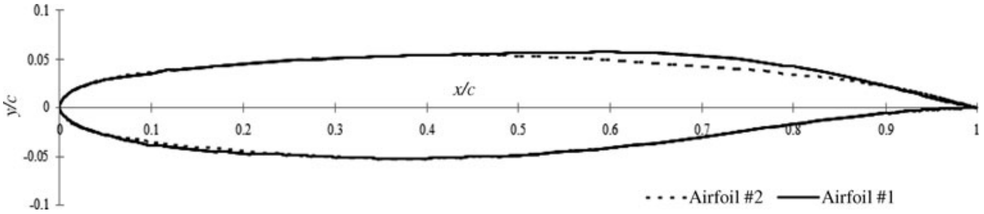

Fig. 7 Pareto-optimal airfoil section #1 (solid line) and #2 (dashed line)

Table 4 Maximum thickness (t_{max}) and camber (f_{max}) of two Pareto-optimal airfoils

Airfoil section	t_{max}/c	Position of t_{max}/c (% c)	f_{max}/c	Position of f_{max}/c (% c)
#1	0.1068	41.7	0.0127	77.1
#2	0.1064	39.8	0.0093	85.1

Table 5 Aerodynamic performance of two Pareto-optimal airfoils

Cruise ($\alpha = 2.5°$)						
Airfoil section	C_L	C_D	C_M	Mach$_{max}$	$(C_L/C_D)_{ac}$	$(C_L/C_D)_{pr}$
#1	0.5749	0.0225	−0.0729	1.32	25.55	25.48
#2	0.5785	0.0286	−0.0658	1.36	20.23	20.27
Approach ($\alpha = 10°$)						
Airfoil section	C_L	C_D	TKE$_{max}$ (J/kg)	ω (1/s)	NM$_{ac}$ (dB)	NM$_{pr}$ (dB)
#1	1.119	0.0194	42.16	874.9	30.02	30.03
#2	1.121	0.0183	33.32	950.3	26.61	26.63

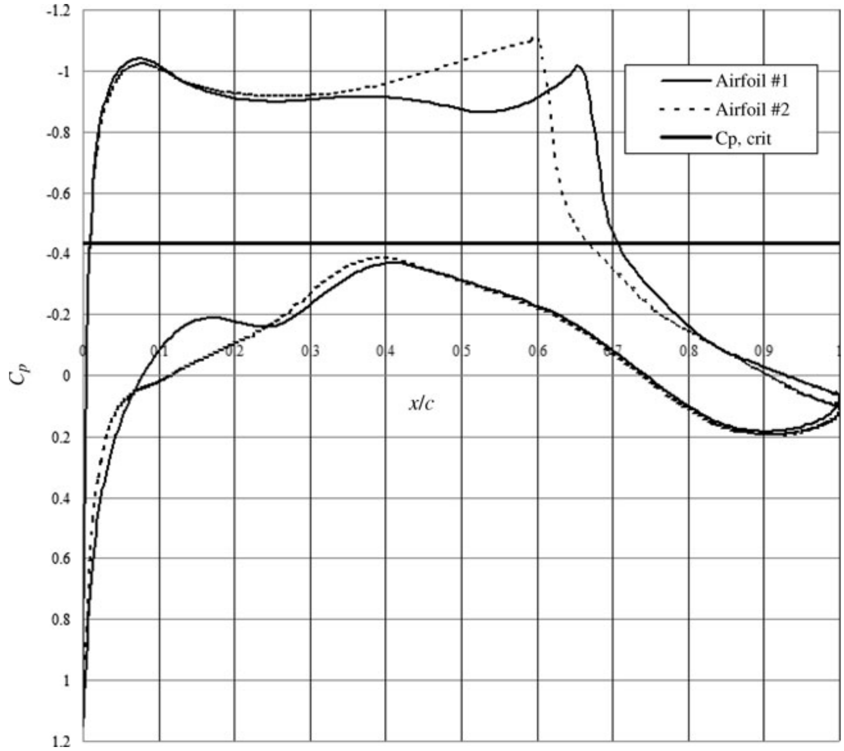

Fig. 8 Pressure coefficient distribution of two Pareto-optimal airfoil sections

airfoil sections are listed in Table 4, while their aerodynamic performance characteristics are presented in Table 5. It can be observed from Table 5 that the predicted objective function values (*pr*) for these two points on the Pareto front are very close to the actual values (*ac*), i.e., the values obtained from a flow analysis in FLUENT®; a fact that demonstrates the good prediction capabilities of the utilized surrogate models.

Both solutions have a thickness ratio close to the minimum allowable value, however, the position of maximum thickness of airfoil #2 is located closer to the leading edge (a distance equal to approximately two per cent of the chord length) than the corresponding location on airfoil #1. The curvature on the upper surface of airfoil #1, particularly in the mid-chord region, is significantly smaller than the curvature on the upper surface of airfoil #2, as shown in Fig. 7. Both solutions have a relatively large leading edge radius. Small curvature along the upper surface, combined with a large leading edge radius, is an important design element of the shape of supercritical airfoils.

A comparison of the corresponding pressure coefficient distributions, displayed in Fig. 8, reveals that the shock strength in the case of airfoil #1 is much weaker than in the case of airfoil #2. Furthermore, the shock wave occurs further aft in the case of airfoil #1. The maximum Mach number value of the flow over airfoil #1 is 1.32, as shown in Table 5, compared to a maximum Mach number value of 1.36 for airfoil #2. Even though, the lift coefficient values during cruise are not significantly different, airfoil #1 has a significantly lower drag coefficient value, which can be attributed to the aforementioned observations regarding the shock wave and the maximum Mach number value.

The values of the noise metric for the two airfoils are also listed in Table 5. Airfoil #2 has a significantly smaller value than airfoil #1 due to the low value of maximum turbulent kinetic energy at the trailing edge of the former airfoil. The value of the drag coefficient is also lower for airfoil #2 in the approach condition.

6 Conclusions

A multi-objective particle swarm optimizer (ACMOPSO) was utilized for the shape optimization of transonic airfoil sections. The computed Pareto-optimal set includes solutions that provide a trade-off between maximizing the lift-to-drag ratio during cruise and minimizing the trailing edge noise during the approach to landing. Five iterations of the optimization process were performed. Significant improvements in both objectives were observed in the values of the final Pareto front compared to the front obtained after the initial sampling of the design variable space. The utilization of various metamodels and the selection of the most appropriate for each objective and constraint function resulted in very accurate predictions and, thus, in a smaller number of required problem evaluations.

The inclusion of robustness in the shape optimization problem objectives, in addition to methods capable of increasing the level of automation within the optimization framework will be investigated in future research projects.

References

Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Haussler D (ed) Proceedings of the annual conference on computational learning theory. ACM Press, Pittsburgh, pp 144–152

Chapman B, Jost G, van der Pas R (2008) Using OpenMP: portable shared memory parallel programming. The MIT Press, Cambridge

Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2:303–314

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester

Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo J, Schwefel HP (eds) Parallel problem solving from nature - PPSN VI, Springer, LNCS, vol 1917, pp 849–858

Drela M, Giles MB (1987) Viscous-inviscid analysis of transonic and low Reynolds number airfoils. AIAA J 25(10):1347–1355

Eberhart RC, Shi Y (2007) Computational intelligence - concepts to implementations. Morgan Kaufmann Publishers, Burlington

Fang KT (1980) The uniform design: application of number-theoretic methods in experimental design. Acta Math Appl Sin 3:363–372

Fang KT, Lin DKJ, Winker P, Zhang Y (2000) Uniform design: theory and applications. Technometrics 42:237–248

Giannakoglou KC (2002) Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. Prog Aerosp Sci 38(1):43–76

Goldstein ME (1976) Aeroacoustics. McGraw-Hill, London

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDD Explor Newsl 11(1):10–18

Harris CD (1990) NASA supercritical airfoils: a matrix of family-related airfoils. NASA Technical Paper 2969

Hosder S, Schetz JA, Mason WH, Grossman B, Haftka RT (2010) Computational-fluid-dynamics-based clean-wing aerodynamic noise model for design. J Aircr 47(3):754–762

Howe MS (1978) A review of the theory of trailing edge noise. J Sound Vib 61(3):437–465

International Civil Aviation Organization (2008) International standards and recommended practices, environmental protection, annex 16 to the convention on international civil aviation. Vol. 1: aircraft noise, 5th edn. Montreal, Canada

Jameson A (1974) Iterative solution of transonic flows over airfoils and wings, including flows at Mach 1. Commun Pure Appl Math 27:283–309

Jameson A (1988) Aerodynamic design via control theory. J Sci Comput 3(3):233–260

Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft Comput 9(1):3–12

Jones BR, Crossley WA, Lyrintzis AS (1998) Aerodynamic and aeroacoustic optimization of airfoils via a parallel genetic algorithm. AIAA Paper 98-4811. In: Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO

Jouhaud JC, Sagaut P, Montagnac M, Laurenceau J (2007) A surrogate-model based multidisciplinary shape optimization method with application to a 2D subsonic airfoil. Comput Fluid 36(3):520–529

Kotinis M (2011) Implementing co-evolution and parallelization in a multi-objective particle swarm optimizer. Eng Optim 43(6):635–656

Leifsson LT, Mason WH, Schetz JA, Haftka RT, Grossman B (2006) Multidisciplinary design optimization of low-airframe-noise transport aircraft. AIAA Paper 2006-230

Lilley GM (2001) The prediction of airframe noise and comparison with experiment. J Sound Vib 239(4):849–859

Menter FR (1994) Two-equation eddy-viscosity turbulence models for engineering applications. AIAA J 32(8):1598–1605

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nat 323:533–536

Shevade SK, Keerthi SS, Bhattacharyya C, Murthy KRK (2000) Improvements to the SMO algorithm for SVM regression. IEEE Trans Neural Netw 11(5):1188–1193

Simpson TW, Lin DKJ, Chen W (2001) Sampling strategies for computer experiments: Design and analysis. Int J Reliab Appl 2(3):209–240

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222

Sobieszczanski-Sobieski J, Haftka RT (1997) Multidisciplinary aerospace design optimization: survey of recent developments. Struct Multidisc Optim 14(1):1–23

Üstün B, Melssen WJ, Buydens LMC (2006) Facilitating the application of support vector regression by using a universal Pearson VII function based kernel. Chemometr Intell Lab 81(1):29–40

Wang XD, Hirsch C, Kang Sh, Lacor C (2011) Multi-objective optimization of turbomachinery using improved NSGA-II and approximation model. Comput Methods Appl Mech Eng 200:883–895

Whitcomb RT (1974) Review of NASA supercritical airfoils. ICAS Paper No. 74-10

Yang BS, Yeun YS, Ruy WS (2002) Managing approximation models in multi-objective optimization. Struct Multidisc Optim 24(2):141–156