Application of Surrogate-Based Optimization Techniques to Aerodynamic Design Cases

Emiliano Iuliano and Domenico Quagliarella

Abstract The paper proposes the application of evolutionary-based optimization coupled with physics-based and adaptively-trained surrogate model to the solution of both two- and three-dimensional aerodynamic optimization problems. The shape parameterization approach consists of the Class-Shape Transformation (CST) method with a sufficient degree of Bernstein polynomials to cover a wide range of shapes. The in-house ZEN flow solver is used for RANS aerodynamic solution. Results show that, thanks to the combined usage of surrogate models and smart training, optimal candidates may be located in the design space even with limited computational resources with respect to standard global optimization approaches.

1 Introduction

In the context of modern and innovative air vehicle design, the development and assessment of new theoretical methodologies represents a cornerstone for reducing the experimental load, exploring trade-offs and proposing alternatives along the design path. The fidelity of such methods is essential to reproduce "real-life" phenomena with a significant degree of accuracy and to take them into account since the very beginning of the design process. However, due to the high computational effort of high-fidelity methods, a big issue rises when hundreds or thousands of analysis evaluations, like in parametric or optimization studies, have to be performed. In order to speed up the computation while keeping a high level of fidelity, the scientific community is increasingly focusing on surrogate methodologies like meta-models, multi-fidelity models or reduced order models, which can provide a compact, accurate and computationally efficient representation of the aircraft design performance.

E. Iuliano (✉) · D. Quagliarella
Fluid Mechanics Department, Multidisciplinary Analysis and Optimization Group,
CIRA—Italian Aerospace Research Center, Via Maiorise, 81043 Capua, Italy
e-mail: e.iuliano@cira.it

D. Quagliarella
e-mail: d.quagliarella@cira.it

E. Minisci et al. (eds.), *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences 48, https://doi.org/10.1007/978-3-319-89988-6_5

Nevertheless, the usage of such models is not straightforward as the amount and quality of information the user has to provide in the learning phase is not known a priori; furthermore, the efficient exploitation of learning data may be hampered by the inherent complexity of the design problem, e.g. non-linearities in the physical model, constraints handling, curse of dimensionality, multi-modal fitness landscape, accuracy *versus* computational effort trade-off. Hence, no general rule exists on the optimal choice of the type of surrogate model, the training and validation strategy, the combination of surrogate model and optimization algorithm.

In order to reduce the computational effort in training accurate surrogate models for aerodynamic shape design problems, this paper proposes the adoption of generic (i.e. not specific to a single surrogate model) in-fill criteria to adaptively and intelligently drive the training process. The adaptive criteria are formulated by explicitly taking into account the goal function with the help of auxiliary functions which have to be maximized. The aim is to find new "optimal" design space points which, once added to the training dataset, provide a "better" surrogate approximation for the optimization purpose. Two surrogate models will be investigated, namely a Kriging model and a Proper Orthogonal Decomposition (POD) model coupled with Radial Basis Function Networks for global interpolation of the modal coefficients. Moreover, two choices of the in-fill criteria are presented in the paper and compared to already published adaptive sampling techniques, like Expected Improvement maximization for Kriging and in-fill criteria for POD model machinery. Two aerodynamic optimization case studies are proposed to test different combinations of surrogate models and adaptive sampling approaches once fixed the computational budget in terms of number of high-fidelity simulations (i.e., CFD analyses). This allows to measure the performances of the presented strategies in a real-world environment and to draw some conclusions about the suitability of in-fill criteria to a specific surrogate model for such a class of problems.

2 Literature Review

Jones et al. [11], among the first, proposed a response surface methodology based on modelling the objective and constraint functions with stochastic processes (Kriging). The so-called Design and Analysis of Computed Experiments (DACE) stochastic process model was built as a sum of regression terms and normally distributed error terms. The main conceptual assumption was that the lack of fit associated only to the regression terms can be considered as entirely due to modelling error, not measurement error or noise, because the training data are derived from a deterministic simulation. Hence, by assuming that the errors at different points in the design space are not independent and the correlation between them is related to the distance between the computed points, the authors came up with an interpolating surrogate model able to provide not only the prediction of objectives/constraints at a desired sample point, but also an estimation of the approximation error. After the construction of such a surrogate model, this last powerful property is exploited to build an

Efficient Global Optimization (EGO), which can be considered as the progenitor of a long and still in development chain of surrogate-based optimization (SBO) methods. Indeed, they found a proper balancing between the need to exploit the approximation surface (by sampling where it is minimized) with the need to improve the approximation (by sampling where prediction error may be high). This was done by introducing the Expected Improvement (EI) concept, already proposed by Schonlau et al. [14], that is an auxiliary function to be maximized instead of the original objective. Sampling at a point where this auxiliary function is maximized improves both the local (exploitation) and global (exploration) search.

An overview of SBO techniques was presented also by Queipo et al. [13] and Simpson et al. [15]. They covered some of the most popular methods in design space sampling, surrogate model construction, model selection and validation, sensitivity analysis, and surrogate-based optimization. Forrester and Keane [4] recently proposed a review of some advances in surrogate-based optimization. An important lesson learned is that only calling the true function can confirm the results coming from the surrogate model. Indeed, the path towards the global optimum is made of iterative steps where, even exploiting some surrogate model, only the best results coming from the true function evaluations are taken as optimal or sub−optimal design. The true function evaluation has to be also invoked to improve the surrogate model. With the term "in-fill criteria" it is usually meant some principles which allow to intelligently place new points (in-fill points) at which the true function should be called. The selection of in-fill points, also referred to as adaptive sampling or model updating, represent the core of a surrogate-based optimization method and helps to improve the surrogate prediction in promising areas of the objective space.

The right choice of the number of points which the initial sampling plan would comprise and the ratio between initial/in-fill points has been the focus of several recent studies. However, it must be underlined that no universal rules exist, as each choice should be carefully evaluated according to the design problem (e.g., number of variables, computational budget, type of surrogate). Forrester and Keane assumed that there is a maximum budget of function evaluations, so as to define the number of points as a fraction of this budget. They identified three main cases according to the aim of the surrogate construction: pure visualization and design space comprehension, model exploitation and balanced exploration/exploitation. In the first case, the sampling plan should contain all of budgeted points as no further refinement of the model is foreseen. In the exploitation case, the surrogate can be used as the basis for an in-fill criterion, that means some computational budget must be saved for adding points to improve the model. They also proposed to reserve less than one half points to the exploitation phase as a small amount of surrogate enhancement is possible during the in-fill process. In the third case, that is two-stage balanced exploitation/exploration in-fill criterion, as also shown by Sóbester [16], they suggested to employ one third of the points in the initial sample while saving the remaining for the in-fill stage. Indeed, such balanced methods rely less on the initial prediction and so fewer points are required. Concerning the choice of the surrogate, the authors observed that it should depend on the problem size, i.e. the dimensionality of the design space, the expected complexity, the cost of the true analyses and the in-fill strategy to be adopted. However, for a given problem, there is not a general rule. The proper choice could come up past various model selection and validation criteria. The accuracy of a number of surrogates could be compared by assessing their ability to predict a validation data set. Therefore, part of the true computed data should be used for validation purposes only and not for model training. This approach can be infeasible when the true evaluations is computationally expensive.

Forrester also underlined that some in-fill criteria and certain surrogate models are somewhat intimately connected. For a surrogate model to be considered suitable for a give in-fill criterion, the mathematical machinery of the surrogate should exhibit the capability to adapt to unexpected, local non-linear behavior of the true function to be mimicked. From this point of view, polynomials can be immediately excluded since a very high order would be required to match this capability, implying a high number of sampling points. In general, a global search would require a surrogate model able to provide an estimate of the error it commits when predicting. Thus, the authors suggested to use Gaussian process based methods like Kriging, although citing the work of Gutmann et al. [5] as an example of one−stage goal seeking approach employing various radial basis functions. Finally, some interesting suitable convergence criterion to stop the surrogate in-fill process were proposed. In an exploitation case, i.e. when minimizing the surrogate prediction, one can rather obviously choose to stop when no further significant improvement is detected. On the other hand, when an exploration method is employed, one is interested in obtaining a satisfying prediction everywhere, so that he can decide to stop the in-fill process when some generalization error metrics, e.g. cross−validation, falls below a certain threshold. When using the probability or expectation of improvement, a natural choice is to consider the algorithm converged when the probability is very low or the expected improvement drops below a percentage of the range of observed objective function values. However, the authors also observed that discussing on convergence criterion may be interesting and fruitful, but “in many real engineering problems we actually stop when we run out of available time or resources, dictated by design cycle scheduling or costs”. This is what typically happens in aerodynamic design, where the high-dimensionality of the design space and expensive computer simulations often do not allow to reach the global optimum of the design problem but suggest to consider even a premature, sub-optimal solution as a converged point.

3 Surrogate Model

The surrogate model consists of the Proper Orthogonal Decomposition (POD) of known CFD flow fields coupled with Radial Basis Function (RBF) Networks to realize the pseudo-continuous representation throughout the design space. The Singular Value Decomposition (SVD) solution of the POD basis vectors and coefficients for steady-state problems is described in references [6, 8–10]. This approach is normally preferred to the eigenvalue/eigenvector solution as it is faster and easier to implement. The discussion will unfold with specific reference to compressible aerodynamic problems, hence the space domain will be the discretized volume occupied by the flowing air and the snapshot vectors will be defined from computed flow fields.

The POD/RBF surrogate models is built from the vectors $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_M$ representing the CFD flow fields and obtained by expensive simulations at representative set of design sites $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$. Finding a Proper Orthogonal Decomposition means to compute a linear basis of vectors to express any other $\mathbf{s}_j \in \mathbb{R}^N$ with the condition that this basis is optimal in some sense. To compute the optimal basis, we first define the snapshot deviation matrix

$$\mathbf{P} = \left(\mathbf{s}_1 - \bar{\mathbf{s}}\ \mathbf{s}_2 - \bar{\mathbf{s}} \cdots \mathbf{s}_\mathrm{M} - \bar{\mathbf{s}}\right)$$

where the ensemble mean vector is computed as

$$\bar{\mathbf{s}} = \frac{1}{M}\sum_{j=1}^{M}\mathbf{s}_j$$

The POD decomposition is obtained by taking the singular value decomposition (SVD) of $\mathbf{P}$

$$\mathbf{P} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{U}\begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_M \\ 0 & \cdots & 0 \end{pmatrix}\mathbf{V}^T \tag{1}$$

with $\mathbf{U} \in \mathbb{R}^{N\times N}$, $\mathbf{V} \in \mathbb{R}^{M\times M}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{N\times M}$ and the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \geq 0$. The POD basis vectors, also called POD modes, are the first M column vectors of the matrix $\mathbf{U}$, while the POD coefficients $\alpha_i(\mathbf{x}_j)$ are obtained by projecting the snapshots onto the POD modes:

$$\alpha_i(\mathbf{x}_j) = (\mathbf{s}_j - \bar{\mathbf{s}}, \boldsymbol{\phi}_i) \tag{2}$$

If a fluid dynamics problem is approximated with a suitable number of snapshots from which a rich set of basis vectors is available, the singular values become small rapidly and a limited number of basis vectors are adequate to reconstruct and approximate the snapshots as they preserve the most significant ensemble energy contribution. In this way, POD provides an efficient mean of capturing the dominant features of a multi-degree of freedom system and representing it to the desired precision by using the relevant set of modes. The reduced order model is derived by projecting the CFD model onto a reduced space spanned by only some of the proper orthogonal modes or POD eigenfunctions. This process realizes a kind of lossy data compression through the following approximation

$$\mathbf{s}_j \simeq \bar{\mathbf{s}} + \sum_{i=1}^{\hat{M}}\alpha_i(\mathbf{x}_j)\boldsymbol{\phi}_i \tag{3}$$

where

$$\hat{M} \leq M \implies \frac{\sum_{i=1}^{\hat{M}} \sigma_i^2}{\sum_{i=1}^{M} \sigma_i^2} \geq \varepsilon \tag{4}$$

and ε is a pre-defined energy level. In fact, the truncated singular values fulfils the relation

$$\sum_{i=\hat{M}+1}^{M} \sigma_i^2 = \varepsilon_{\hat{M}}$$

If the energy threshold is high, say over 99% of the total energy, then $\hat{M}$ modes are adequate to capture the principal features and approximately reconstruct the dataset. Thus, a reduced subspace is formed which is only spanned by $\hat{M}$ modes.

3.1 Pseudo-continuous Global Representation

Equation 3 allows to get a POD approximation of any snapshot $\mathbf{s}_j$ belonging to the ensemble set. Indeed, the model does not provide an approximation of the state vector at design sites which are not included in the original training dataset. In other words, the POD model by itself does not have a global predictive feature, i.e. over the whole design space. As the aim is to exactly reproduce the sample data used for training and to consistently catch the local data trends, a Radial Basis Function (RBF) network answers to these criteria and has been chosen for POD coefficients interpolation. Gaussian, multi-quadric and inverse quadratic functions are used. The RBF parameters are found by imposing the interpolation condition on the training set for any modal coefficient $i \leq \hat{M}$.

The RBF width parameters have a big influence both on the accuracy of the RBF model and on the conditioning of the solution matrix. In particular, it has been found (Refs. [3, 5]) that interpolation errors become high for very small and very large values of the width parameter θ, while the condition number of the coefficient matrix increases with increasing values of θ. Therefore, they have to be “optimal” in the sense that a tuning of the width parameters is needed to find the right trade-off between interpolation errors and solution stability (Ref. [3] for a discussion about how to properly select the best set of parameters). The pseudo-continuous prediction of the flow field at a generic design site w is then expressed as:

$$\mathbf{s}(\mathbf{x}) = \bar{\mathbf{s}} + \sum_{i=1}^{\hat{M}} \alpha_i(\mathbf{x})\boldsymbol{\phi}_i \tag{5}$$

This provides a useful surrogate model which combines design of experiments for sampling, CFD for training, POD for model reduction and RBF network for global approximation. In conclusion, an explicit, global, low-order and physics-based model linking the design vector and the state vector has been derived and will be used as surrogate model.

4 Surrogate Model Sequential In-fill

The strategy for training a surrogate model is referred to as the process of selecting a proper set $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$ on which the surrogate model is built. The training strategy is heavily dependent on the type and scope of the surrogate model and, in principle, has to be tailored on it. Indeed, the addition of new samples, if not completely random, must follow some specific criteria that may be very different depending on the purpose of the training process. For instance, Latin Hypercube Sampling (LHS) has been designed to satisfy space-filling requirements and obtain a good coverage of the design space. Here, the emphasis is given on sampling strategies which are able to "adapt" to the response within an optimization process: in particular, they can aim at improving the quality of the model prediction (error-driven strategies) or minimizing the objective function (objective-driven strategies). Most of the adaptive sampling approaches pursue the exploration/exploitation trade-off, where exploration means sampling away from available data, where the prediction error is supposedly higher, while exploitation means trusting the model prediction, thus sampling where the surrogate provides global minima. It is clear that a trade-off between the two behaviors is needed: indeed, exploration is useful for global searching, but it may lead to unveil uninteresting regions of the design space; on the other hand, exploitation helps to improve the local accuracy around the predicted optima, but it may result in local minima entrapment. Figure 1 provides a simple example of adding a new training point by using respectively exploitation, exploration and balanced approaches. Given a set of training points (black circle points) evaluated on the true function (solid black line), a surrogate model (dashed black line) is built: if a new sample has to be added, a pure exploitation approach would place it where the global minimum of the surrogate is detected, i.e. very close to one of the training point (triangle point); a pure exploration approach, instead, would lead to sample where the maximum uncertainty in the model prediction is found, i.e. far from available training points (circle point); a balanced exploration/exploitation approach combines the two aspects, thus providing a new sample which significantly improves the surrogate prediction (square point).

Here, we are interested in designing balanced in-fill criteria for a generic surrogate model. Such criteria are formulated in terms of an auxiliary function referred to as v function hereinafter and called potential of improvement. Given the generic location in the design space $\mathbf{x}$, the objective function $f(\mathbf{x})$ to minimize, a set of n available sampling points $\{X_n\}$ and the corresponding set of true objective function values $\{F_{X_n}\}$

$$X_n = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \qquad F_{X_n} = \{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$$

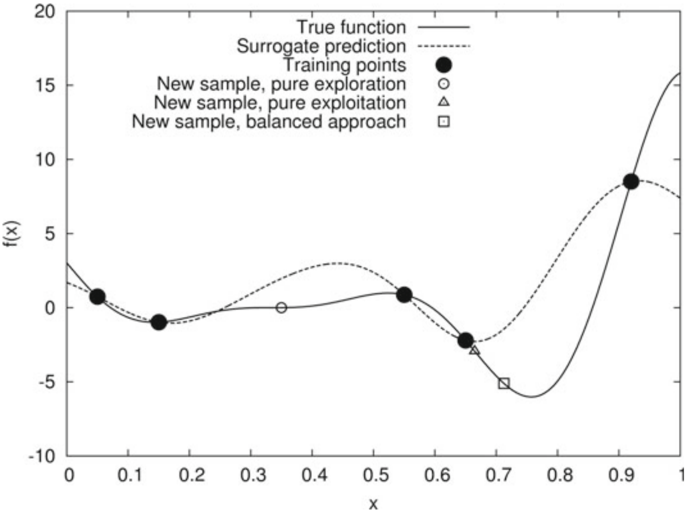


Fig. 1 Exploitation versus exploration, 1D example

the update of the surrogate model can be realized by finding a new sample $\mathbf{x}_{n+1}$ which maximizes the potential of improvement:

$$\mathbf{x}_{n+1} = \underset{\mathbf{x}}{\operatorname{argmax}}\, v(\mathbf{x}, \hat{f}(\mathbf{x}), X_n, F_{X_n})$$

where $\mathbf{x}$ is the generic design space location and $\hat{f}(\mathbf{x})$ is the surrogate prediction at $\mathbf{x}$.

Hereinafter, the maximization of the auxiliary function is achieved in the following way: a huge Latin Hypercube Sampling dataset (e.g., five hundred times the dimension of the design space) is computed and the values of the auxiliary functions are computed at each point (this requires limited computational effort as the auxiliary function only depends on the surrogate prediction, which is fast to obtain, and on the true objective function values at already collected points); hence, the new sample is located where the maximum value of the auxiliary function is met. In order to avoid the duplication of the updating samples when iterating the in-fill process, the seed of the Latin Hypercube is changed at each iteration.

As concerns the type and nature of the potential of improvement function, previous investigations [7] showed that error-driven in-fill criteria may lead to intensively explore the design space in order to reduce the prediction error, but, conversely, this resulted in a lack of efficiency of the whole optimization process when fixing the total computational budget. Hence, in the following section the discussion will focus on objective-driven approaches which proved to be more suitable to global optimization. In particular, two criteria will be proposed: the first is based on the factorization of the potential of improvement in order to explicitly realize the trade-off between exploration and exploitation; the second is defined according to the Expected Improvement concept.

4.1 Factorization Criterion

The first proposal of adaptive in-fill criterion aims at combining exploration and exploitation by means of a generic factorization as follows:

$$v(\mathbf{x}, \hat{f}(\mathbf{x}), X_n, F_{X_n}) = g(\mathbf{x}, X_n)h(\hat{f}(\mathbf{x}), F_{X_n}) \tag{6}$$

The functions g and h measure the exploration and model trust contribution respectively. In particular, the exploration function g should estimate how strong is the influence of the set of already collected samples X_n on a generic candidate $\mathbf{x}$. One of the preferred approaches is to make the g function dependent on the Euclidean distance $d(\mathbf{x}, \mathbf{x}_i)$ between the generic design space location $\mathbf{x}$ and the i-th element of the training set X_n:

$$g(\mathbf{x}, X_n) = g(d(\mathbf{x}, \mathbf{x}_1), d(\mathbf{x}, \mathbf{x}_2), \ldots, d(\mathbf{x}, \mathbf{x}_n))$$

On the other hand, the exploitation function h should take into account how the surrogate prediction $\hat{f}$ compares with the available set of true objective function values F_{X_n}. In particular, this contribution should put emphasis on trusting the model prediction, hence the h function should exhibit its maxima in correspondance to the minima of $\hat{f}$.

Of course, different in-fill criteria can be selected by properly designing the functions g and h. In the present context, the following solution is adopted:

$$g(\mathbf{x}, X_n) = \frac{\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)}{\max_{\mathbf{x}_i, \mathbf{x}_j \in X_n} d(\mathbf{x}_i, \mathbf{x}_j)} \tag{7}$$

$$h(\hat{f}(\mathbf{x}), F_{X_n}) = \exp\Big(-\sigma \frac{\hat{f}(\mathbf{x}) - f_{min}}{f_{max} - f_{min}}\Big) \tag{8}$$

where σ is a tuning parameter, $f_{min} = \min\{f_{\mathbf{x}_1}, \ldots, f_{\mathbf{x}_n}\}$ and $f_{max} = \max\{f_{\mathbf{x}_1}, \ldots, f_{\mathbf{x}_n}\}$. This choice of the h function provides two main features:

1. the value of h approaches the unity when $\hat{f}(\mathbf{x})$ approaches f_{min};
2. for $\hat{f}(\mathbf{x}) < f_{min} \longrightarrow h(\hat{f}(\mathbf{x}), F_{X_n}) > 1$;

As a consequence, "bad" candidates (from the surrogate model point of view) will be filtered out, while "good" candidates (i.e., candidates with predicted objective function values lower than the current minimum of the true objective function) will be recognized and rewarded with higher rank. However, if they are too close to samples stored in X_n, they will be penalized by the g function. Hence, a trade-off is realized

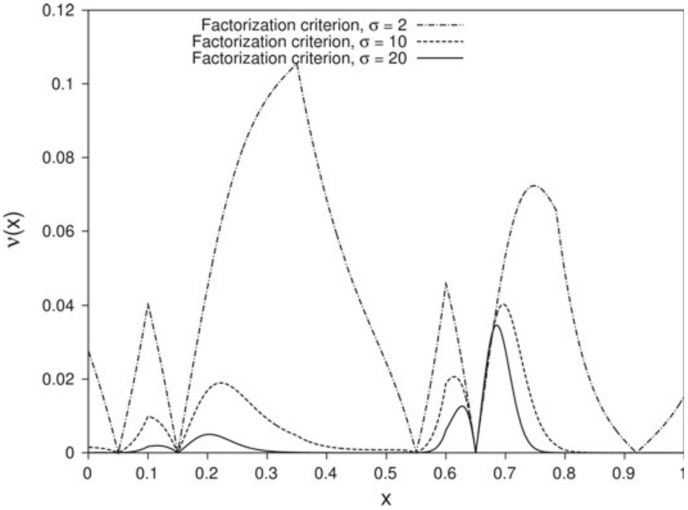


Fig. 2 Potential of improvement based on factorization criterion, 1D example

between surrogate prediction and location in the design space. Figure 2 shows the potential of improvement function for three values of the σ parameter. The case is the one depicted in Fig. 1. The effect of the σ parameter is clearly observable: indeed, for $\sigma = 2$ the peak of the potential of improvement is located around $x \simeq 0.35$, i.e. in a region which is uninteresting to discover the global optimum of the true objective function. This occurs because, for $\sigma = 2$, the filtering power of the h function is relatively small and so it is its weight within the factorization: as a consequence, the g function dominates, the exploration-exploitation trade-off is not realized (the exploration contribution is much higher) and new candidates will be chosen according to their distance from collected samples. The landscape changes for $\sigma = 10$ and $\sigma = 20$, as the levels of v are globally flattened and the peak moves to $x \simeq 0.7$, i.e. very close to the position of the true objective function optimum.

4.2 *Expected Improvement-Like Criterion*

This criterion has been designed trying to mimic the same rationale of the Expected Improvement criterion, usually coupled to a Kriging-based surrogate, as highlighted in Sect. 2. The present approach, named "EI-like" hereinafter, represents a generalization of that method: indeed, for a generic surrogate model, the information about the uncertainty of the surrogate is not available, while a Kriging model, being a Gaussian process, provides an estimate of the prediction variance together with the prediction itself. The potential of improvement is designed to have the same form of the Expected Improvement function, that is:

$$v(\mathbf{x}, \hat{f}(\mathbf{x}), X_n, F_{X_n}) = (f_{min} - \hat{f}(\mathbf{x}))\Phi\Big(\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\Big) + \hat{s}(\mathbf{x})\phi\Big(\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\Big) \tag{9}$$

where $\hat{s}(\mathbf{x})$ is an estimate of the prediction error and $\Phi(\mathbf{x})$ and $\phi(x)$ are respectively the cumulative distribution and probability density functions of a standard normal distribution. The prediction error is estimated as follows:

$$\hat{s}(\mathbf{x}) = \frac{1}{2}|f_{max} - f_{min}| \exp\Big(-\gamma \frac{\max_{\mathbf{x}_i, \mathbf{x}_j \in X_n} d(\mathbf{x}_i, \mathbf{x}_j)}{\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)}\Big) \tag{10}$$

where γ is a tuning parameter.

The $\hat{s}$ function has been designed in order to quickly increase with increasing distance from an available sample and to have an order of magnitude related to the actual values of the objective function. Figure 3 shows the potential of improvement function for three values of the γ parameter. The case is the one reported in Fig. 1. The γ parameter strongly alters the potential of improvement profile in terms of potential levels and location of the peak. Indeed, for $\gamma = 0.001$ and 0.01, the peak of the potential of improvement is around $x \simeq 0.7$, i.e. in the vicinity of the true minimum, thus providing a balanced prediction for a new sample. For $\gamma = 0.1$, instead, the peak moves to $x \simeq 0.3$ and the balance would significantly shift in favor of pure exploration: in fact, injecting a new sample at $x \simeq 0.3$ would not improve the surrogate prediction of the true objective function in the vicinity of the global optimum at the next iterate.

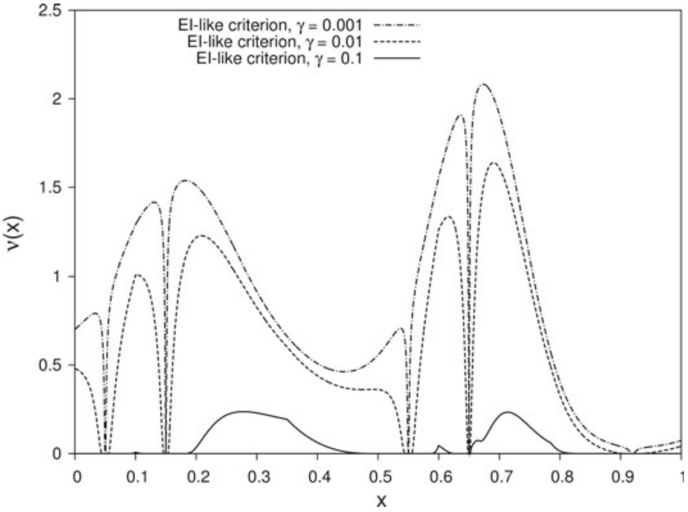


Fig. 3 Potential of improvement based on EI-like criterion, 1D example

5 Surrogate-Based Optimization

The workflow of the surrogate-based shape optimization (SBSO) is depicted in Fig. 4. Basically, it starts with an *a-priori* design of experiment (a Latin Hypercube sampler) whose aim is to initialize the database population: typically, based on literature results and author's experience, the dimension of the initial sampling should not exceed one-third of the total computational budget. A parameterization module transforms the design vectors into geometrical shapes, for each shape a volume mesh is computed by launching an in-house developed automatic mesh generator and a set of CFD computations are executed in parallel with the in-house ZEN CFD flow solver [2]. Once the converged flow field variables are available, the POD/RBF surrogate model is built as described in Sects. 3 and 3.1. After that, the workflow in Fig. 4 shows two internal cycles, namely the sequential in-fill (also called adaptive sampling) and the optimization update. These iterative phases reflect two different needs: first of all, providing an improved and reliable model to the optimizer; then, iterating the optimizer to refine the optimum search.

The first cycle (database updating by in-fill criteria) is based on the techniques described in Sect. 4 and is aimed at improving the surrogate model prior to the optimization phase by providing new design candidates $\mathbf{x}_{new}$ to be added to the ensemble database. The condition to exit from this internal loop is based either on pre-defined levels of improvement or on computational budget considerations.

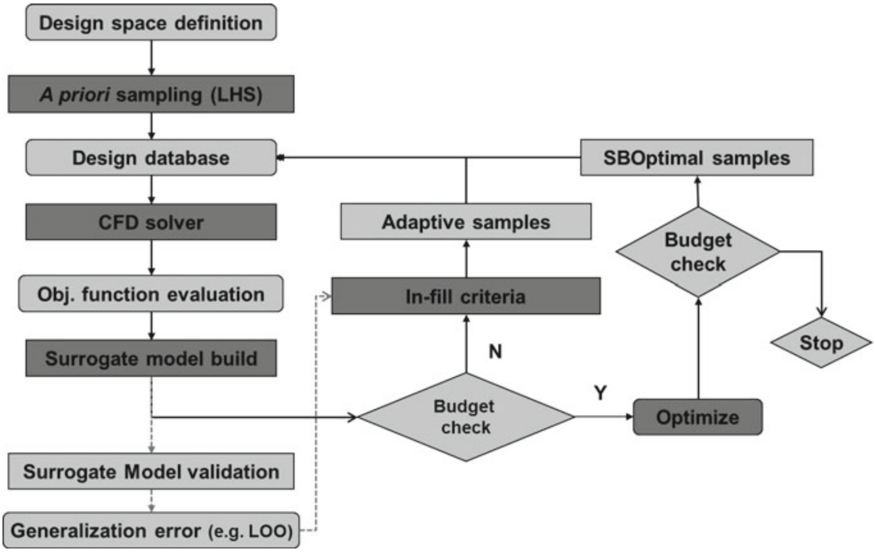


Fig. 4 Workflow of surrogate-based evolutionary optimization: design database updating by adaptive in-fill and surrogate optima addition

The second cycle (database updating by optimization) consists of a series of sequential optimizations where, at i-th iteration, the surrogate model is updated with the high-fidelity evaluation of the optimum candidate found at iteration $(i-1)$-th. This phase allows for including optimal or sub−optimal design sites $\mathbf{x}_{opt}$, provided by the surrogate-based optimization, into the POD ensemble database and should lead to refine the search process in the design space region where the "true" optimum resides. The loop terminates either when the residual of the objective function of the predicted optima falls below a pre-defined threshold or when the computational budget limit has been reached. The optimizer consists of an evolutionary algorithm implemented within the in-house ADGLIB optimization library [17].

6 Application to Aerodynamic Design Cases

In the next sections, two shape optimization cases are considered, i.e. the RAE 2822 airfoil and the isolated wing proposed within the 3rd AIAA Drag Prediction Workshop. Both cases are set in transonic viscous flow conditions. The results of global optimization by means of evolutionary algorithms and CFD evaluation of the objective function will be taken as reference for surrogate-based studies.

6.1 RAE 2822 Airfoil Case

The shape optimization problem is formulated as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & C_d(\mathbf{x}) \\ \text{subject to} \quad & C_l(\mathbf{x}) = C_{l,base} = 0.824 \\ & C_m(\mathbf{x}) \geq C_{m,base} = -0.092 \\ & A(\mathbf{x}) \geq A_{base} = 0.7787\,\text{m}^2 \end{aligned}$$

where $\mathbf{x}$ is the generic design vector, $A(\mathbf{x})$ is the total area enclosed by the generic airfoil and A_{base} is the corresponding value for the baseline RAE 2822 airfoil. The lift constraint is explicitly satisfied by performing the flow simulation at fixed lift. The pitching moment and the geometric constraint are treated by using a penalization approach. A unit airfoil chord is assumed, the pitching moment is evaluated at the quarter-chord, the Mach number is 0.734 and the Reynolds number is 6.5×10^6.

6.1.1 Parameterization

The CST (Class-Shape Transformation) approach [12] describes an airfoil shape as a function of the surface parameter $\psi = \frac{x}{c}$, where x is the x-coordinate along the chord line and c is the chord length. The y-coordinate are obtained for upper and lower side as follows:

$$\frac{y_u}{c}(\psi) = C_{1.0}^{0.5}(\psi) S_u(\psi) + \psi \frac{\Delta z_{TE}}{2c} \tag{11}$$

$$\frac{y_l}{c}(\psi) = C_{1.0}^{0.5}(\psi) S_l(\psi) - \psi \frac{\Delta z_{TE}}{2c} \tag{12}$$

where $C_{1.0}^{0.5}(\psi)$ is the class function for rounded leading edge and pointed trailing edge airfoils and $S_u(\psi)$ and $S_l(\psi)$ are the so-called shape functions for upper and lower sides:

$$C_{1.0}^{0.5}(\psi) = \psi\sqrt{(1-\psi)} \tag{13}$$

$$S_u(\psi) = \sum_{i=0}^{n} A_{u_i} K_{i,n} \psi^i (1-\psi)^{n-i} \tag{14}$$

$$S_l(\psi) = \sum_{i=0}^{n} A_{l_i} K_{i,n} \psi^i (1-\psi)^{n-i} \tag{15}$$

Shape functions are Bernstein polynomials of order n, Δz_{TE} is the trailing edge thickness, $K_{i,n}$ are binomial coefficients

$$K_{i,n} = \binom{n}{i} = \frac{n!}{i!(n-i)!}$$

and the Bézier coefficients A_{u_i} and A_{l_i} are design weights which can be either defined a-priori in a design optimization process or computed with a least-squares fit to match a specified geometry. The first and last design parameters, i.e. A_{u,l_0} and A_{u,l_n}, are directly linked to well known airfoil shape parameters like leading edge radius R_{le} and trailing edge angle β, being $A_{u,l_0} = \sqrt{\frac{2R_{le}}{c}}$ and $A_{u,l_n} = \tan\beta$. In the present context, 6th-order Bernstein polynomials are considered, hence each airfoil side (upper and lower) is described by 7 design variables for a total number of 14 design variables. The corresponding design weights, which define the RAE 2822 profile according to the chosen parameterization, have been obtained by least-squares fit and are reported in Table 1. The design space bounds are defined by taking ±50% of the baseline design weights.

Table 1 Design weights representing the RAE 2822 airfoil shape with 6th-order Bernstein polynomials

Design weights	$\pm\sqrt{\frac{2R_{le}}{c}}$	A_1	A_2	A_3	A_4	A_5	tan (β)
Upper side	0.12658	0.13786	0.16165	0.16418	0.22047	0.17191	0.21321
Lower side	−0.12892	−0.14264	−0.13111	−0.26147	−0.02525	−0.11451	0.07275

6.1.2 Optimization Studies

Five optimization studies have been performed employing different methods and computational load. Details are reported in Table 2.

Three SBOSA (Surrogate Based Optimization with Sequential Adaptation) runs have been launched, sharing the same setup but exploiting respectively the factorization (SBOSA-FC) and the EI-like (SBOSA-EIL1 and SBOSA-EIL2) criteria. In the latter case, two simulations have been launched with two different values of the EI-like criterion tuning parameter γ in order to explore its effect on the search process. All SBOSA simulations consist of three stages: a a-priori LHS sampling of 42 samples, a model updating stage with adaptive sampling (208 sequential calls of the chosen in-fill criterion) and a surrogate-assisted evolutionary optimization stage consisting of 50 iterative genetic algorithm calls with re-injection of the computed surrogate-based optima in the POD/RBF model database. The total computational budget comprises 300 CFD computations, a rather low effort if considering that the dimension of the design space is 14 and the convergence of classical evolutionary algorithms may require a number of evaluations tens of times higher. No criterion for setting the convergence of the surrogate-assisted optimization is provided here: indeed, the aim is to verify that the attained level of improvement, once run out of the limited computational budget, is significant and possibly “close” to the solution obtained with more expensive algorithms. This would demonstrate that the surrogate-

Table 2 Summary of optimization studies for RAE 2822 case

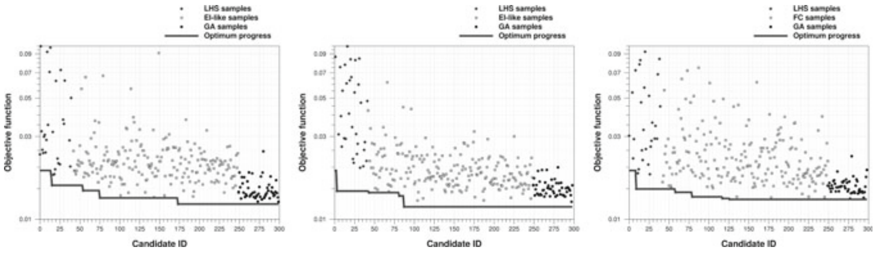
Opt. run ID	SBOSA-EIL1	SBOSA-EIL2	SBOSA-FC	EGO	PGA
Opt. method	EI-like + GA	EI-like + GA	FC + GA	EI	GA
Tuning parameter	$\gamma = 0.001$	$\gamma = 0.005$	$\sigma = 10$	–	–
Obj. func. eval.	POD/RBF	POD/RBF	POD/RBF	Kriging	CFD
Total no. of evaluations	300	300	300	224	6400

assisted search method is considerably efficient in exploring high-dimensional design spaces.

EGO (Efficient Global Optimization) run is performed by using the DAKOTA package [1]. As already stated in Sect. 2, the EGO algorithm searches for the design space location which maximizes the Expected Improvement function and updates the Kriging model database accordingly. The total number of iterations is set to 224.

PGA (Plain Genetic Algorithm) is a pure evolutionary optimization, it is not assisted by any surrogate model and the objective function evaluation is carried out by means of the CFD flow solver (true evaluation of the objective function). A population size of 64 individuals is let evolve for 100 generations with a crossover probability activation of 100% and a mutation rate of 2%.

The true objective function history of the SBOSA runs are reported in Fig. 5. Candidate samples are clearly distinguished according to the criterion used to select them (LHS, EI-like or FC and GA). The solid dark grey line marks the advancement of the minimum value of the true objective function. It is clearly observable how the initial LHS stage (dark grey diamonds) provides only for the initialization of the surrogate, while the objective-driven in-fill points (light grey squares) contribute to progressively drop the objective function levels. In all cases, the overall minimum is not found within the GA-assisted final stage, but rather within the adaptive in-fill phase: this is not surprising as the in-fill criteria are objective-driven, i.e. they rely on the minimization of the surrogate objective function in a balanced exploration-exploitation approach. Figure 6 shows the progress of the true objective function minimum for each of the five optimization studies. The x-axis (progressive number of candidates) is reported in logarithmic scale for the sake of clarity as different scales are involved. It is clearly observable how the surrogate-based optimizations manage to achieve significant improvement with a limited number of CFD evaluations with respect to the plain GA. In particular, Table 3 summarizes the results of each optimization study in terms of minimum objective function value attained and number of effective CFD evaluations needed to capture it. SBOSA provides an interesting compromise solution between the greater performance of PGA at much higher computational cost and the reduced performance of EGO at increased speed (best candidate found with only 81 evaluations).



(a) SBOSA-EIL1 ($\gamma = 0.001$) (b) SBOSA-EIL2 ($\gamma = 0.005$) (c) SBOSA-FC

Fig. 5 Objective function history of SBOSA optimization runs

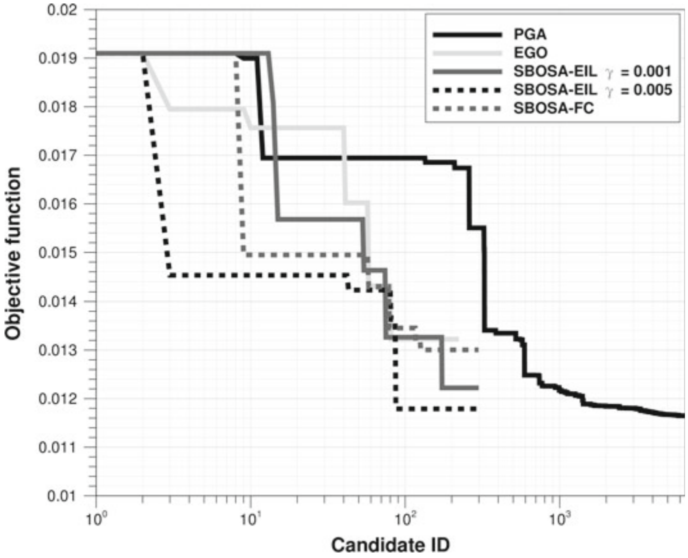


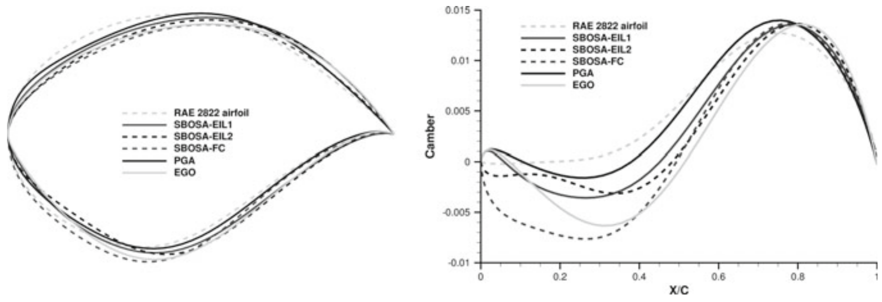
Fig. 6 Progress of minimum values of the objective function for RAE 2822 airfoil optimization studies

Table 3 RAE 2822 optimization, comparison of best candidates

Run	CFD evaluations	Objective value	Improvement
RAE 2822 baseline	–	0.0194	–
PGA best	4200	0.0116	−40.2%
SBOSA-EIL1 best	174	0.0122	−37.1%
SBOSA-EIL2 best	88	0.0118	−39.2%
SBOSA-FC best	126	0.0130	−33.0%
EGO best	81	0.0132	−32.0%

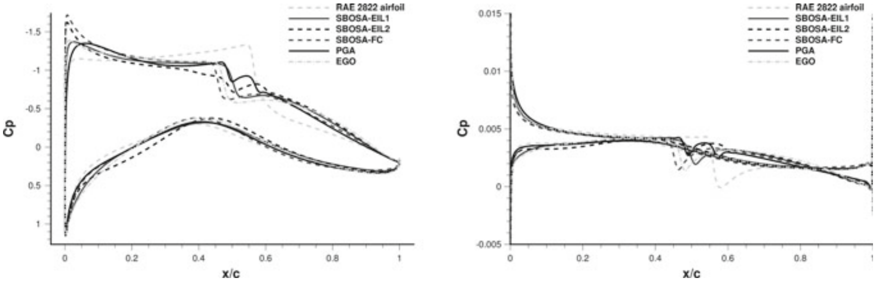
Figure 7a depicts a comparison of the RAE 2822 and the optimized airfoils contour: it can be observed that, for all optimal shapes, there has been a significant reduction of the leading edge radius and an important de-cambering in the fore region (up to 50% airfoil chord). Figure 7b better highlights this feature as the camber distributions of baseline and optimized airfoil shapes are shown. Reducing the leading edge radius is essential to increase the pressure peak and to dampen the shock wave strength, while the negative camber helps to fit within the pitching moment constraint.

Figure 8 shows a comparison of pressure coefficient and skin friction distribution between the RAE 2822 and the optimized airfoils. The shape modification shifted the shock location forward and reduced the local Mach number upstream the shock, thus reducing its intensity and preventing the incipient shock-induced separation which is observed on the RAE 2822 skin friction profile.



(a)Shape (b) Camber distribution

Fig. 7 Geometry comparison between RAE 2822 and optimized airfoils



(a) C_p distribution (b) C_f distribution

Fig. 8 Aerodynamic comparison between RAE 2822 and optimized airfoils

6.2 Drag Prediction Workshop Wing Case

The optimization case is inspired to one of the test cases issued within the 3rd AIAA Drag Prediction Workshop (DPW). The viscous flow around an isolated wing at Mach number of 0.78, Reynolds number of 5 millions and angle of attack (AOA) of 1° is considered. Fully turbulent flow is assumed. The constrained minimization problem is defined as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & -\frac{C_L(\mathbf{x})}{C_D(\mathbf{x})} \\ \text{subject to} \quad & \max_s\left(\frac{t(\mathbf{x},s)}{c}\right) = 0.136 \\ & C_L(\mathbf{x}) \geq 0.5 \\ & C_M(\mathbf{x}) \geq -0.05 \end{aligned}$$

where c is the local wing chord value and $t(\mathbf{x}, s)$ is the wing thickness distribution along the chordwise abscissa s for a generic design vector $\mathbf{x}$.

6.2.1 Parameterization

As an extension of the CST airfoil parameterization, a wing shape can be obtained by distributing the airfoil shape function along another surface parameter η, representing the wing span, with the desired law. Once selected N_x, the order of the Bernstein polynomial which represents the wing airfoils, the complete wing shape can be obtained by transforming the coefficients of the Bernstein polynomial into a distribution along the spanwise direction using any appropriate numerical technique. In this way, the entire wing surface is defined through the same basic component airfoils of the root airfoil, but the magnitude of each of them varies across the wing span according to the expansion technique. For example, using the Bernstein polynomial as expansion technique, the spanwise variation of each coefficients A_{u_i} in Eqs. 14 and 15 can be stated in the form

$$A_{u_i} \equiv A_{u_i}(\eta) = \sum_{j=0}^{N_y} B_{u_{i,j}} S_{j,N_y}(\eta) \tag{16}$$

where

$$S_{j,N_y}(\eta) = K_{j,N_y} \eta^j (1-\eta)^{N_y-j}$$

N_y is the order of the expansion Bernstein polynomial and K_{j,N_y} is the usual binomial coefficient.

By inserting Eq. 16 into Eq. 15 and adopting the same technique for the lower surface, the bi-variate Bernstein polynomial shape function for the entire wing is derived

$$\begin{aligned} S_u(\psi, \eta) = \sum_{i=0}^{N_x} A_{u_i}(\eta) S_{i,N_x}(\psi) = \\ \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \left[B_{u_{i,j}} K_{j,N_y} \eta^j (1-\eta)^{N_y-j} \right] K_{i,N_x} \psi^i (1-\psi)^{N_y-i} \end{aligned} \tag{17}$$

$$\begin{aligned} S_l(\psi, \eta) = \sum_{i=0}^{N_x} A_{l_i}(\eta) S_{i,N_x}(\psi) = \\ \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \left[B_{l_{i,j}} K_{j,N_y} \eta^j (1-\eta)^{N_y-j} \right] K_{i,N_x} \psi^i (1-\psi)^{N_y-i} \end{aligned} \tag{18}$$

The wing shape will be then represented by $2 \times (N_x + 1) \times (N_y + 1)$ design parameters, namely $B_{u_{i,j}}$ and $B_{l_{i,j}}$, $i = 0, \ldots, N_x$, $j = 0, \ldots, N_y$. By multiplying the wing shape function by the airfoil-like class function, the overall shape of the wing can be computed. However, a wing is generally and naturally conceived with spanwise distributions for twist angle, dihedral angle, sweep angle and taper ratio:

this additional parameters have to be introduced in the CST model in order to cover realistic wing concepts. The actual wing surface cartesian coordinates can be obtained from the equations

$$x = \psi L(\eta) + x_{LE_{root}} + \int_0^\eta [\sin \Lambda(\eta)] d\eta \tag{19}$$

$$y = \frac{b}{2}\eta \tag{20}$$

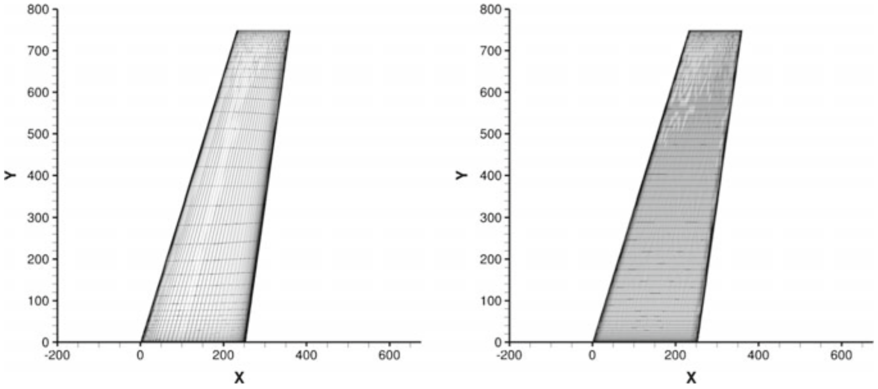
$$\begin{aligned} z_{u,l} = {} & L(\eta) C_{1.0}^{0.5}(\psi) S_{u,l}(\psi, \eta) + L(\eta)\psi[\Delta z_{TE}(\eta) - \tan \alpha_T(\eta)] + \\ & L(\eta) \int_0^\eta [\sin \delta(\eta)] d\eta \end{aligned} \tag{21}$$

where the following spanwise distributions have been introduced: $L(\eta)$ chord length, $\Lambda(\eta)$ sweep angle, $\Delta z_{TE}(\eta)$ trailing edge thickness, $\alpha_T(\eta)$ twist angle, $\delta(\eta)$ dihedral angle. b is the wing span length and $x_{LE_{root}}$ is the x-position of the leading edge of the root airfoil. Several laws can be defined for spanwise distributions, but the standard approach which assures the manufacturing feasibility of the wing shape is to assume constant (e.g. sweep angle), piecewise constant (e.g. dihedral angle), linear (e.g. twist angle) or piecewise linear (e.g. chord length/tapering) variations.

In the present case, one design variable is assigned to control the wing tip twist angle while the twist angle at the root section is kept fixed. A linear twist distribution is adopted along the wing span. The shape design variables are 16, as N_x and N_y are chosen to be respectively 3 and 1. Hence, the total number of design variables is 17.

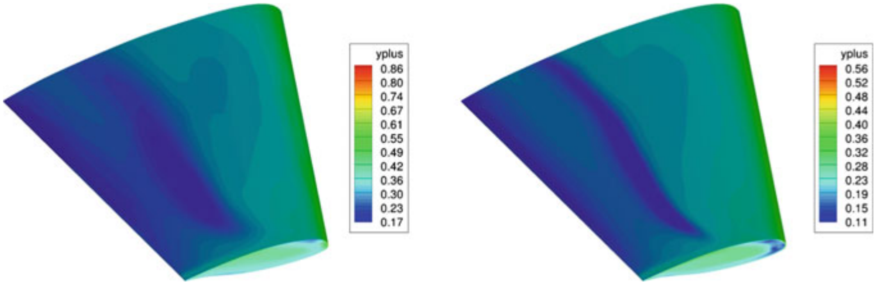
6.2.2 Mesh Generation

The computational mesh is generated by using the ICEM CFD commercial package. Once defined the wing shape from a specific design vector by using the aforementioned CST approach, three wing sections, namely root, mid and tip sections, are imported within ICEM CFD and employed to generate support curves and surfaces. A replay script file is used to fully parametrize the blocking arrangement, the structured grid generation and the mesh export in a file format suitable for the CFD flow solver. Eight blocks are designed around the wing shape and a family of two grids is defined: the coarse and fine mesh consist respectively of 712,448 cells and 2,959,872 cells. A sketch of the surface mesh distribution is shown in Fig. 9. Both meshes are conceived to respect the $y^+ = O(1)$ condition, as also shown in Fig. 10 where the contour map of y^+ distribution on the wing surface is depicted. The coarse mesh will be used for optimization studies, while the fine mesh will provide more accurate comparisons of the aerodynamic flow for optimized shapes at the end of the optimization process.



(a) Coarse mesh (b) Fine mesh

Fig. 9 Computational mesh on DPW wing surface



(a) Coarse mesh (b) Fine mesh

Fig. 10 y^+ distribution on DPW wing surface

6.2.3 Optimization Studies

Similarly to the previous case, three optimization approaches are compared: a plain evolutionary-based optimization and two surrogate-assisted approaches. Table 4 provides details of each study. In particular, PGA (Plain Genetic Algorithm, also referred to as DGA, Direct Genetic Algorithm) features a single-objective, genetic algorithm run (40 generations with a population size of 96 candidates), calling the CFD flow solver as fitness evaluator. As in the previous case, the EGO method is launched through the Dakota interface. SBOSA-EIL approach is the same as for the RAE 2822 case, but here the total budget of true function evaluations is set to 102 and the database breakdown is as follows:

- 16 samples are suggested by LHS and the POD/RBF surrogate model is initialized;
- 56 samples are iteratively provided by applying the EI-like in-fill criterion;
- 30 samples are finally suggested by optimizing on the surrogate with repeated GA calls.

Table 4 Summary of optimization studies for DPW wing

ID	Obj. function eval.	Opt. method	Total no. evaluations
PGA	CFD	GA	3936
SBOSA-EIL	POD/RBF	SBO (EI-like + GA)	102
EGO	Kriging	Dakota EGO	502

The baseline shape is the wing geometry DPW-W1 proposed within the 3rd Drag Prediction Workshop. With respect to the original design point (Mach $= 0.76$, AOA $= 0.5°$), some changes are introduced in order to make the optimization problem harder. Indeed, the primary interest is to verify the capability of surrogate-assisted techniques to recover optimal or sub-optimal design solutions by using limited computational resources. To this aim, a convincing test is to force the algorithm to start from poor aerodynamic solutions and observe how quick it is to fall back in promising regions. Hence, a new design point suitable for optimization purposes is identified at a higher Mach number ($= 0.78$) and a higher angle of attack ($= 1.0°$). Table 5 reports the aerodynamic coefficients and objective function values for three design points, namely the original one, the optimization one and an intermediate one. The latter has been reported in order to highlight that, by increasing only the Mach number at fixed angle of attack, the lift coefficient and hence the induced drag would not be altered significantly: as a matter of fact, moving the AOA to 1° would force the optimizer to heavily work on twist and shape design variables in order to decrease the lift coefficient and the induced drag accordingly.

The true objective function history of the SBOSA-EIL run is reported in Fig. 11. For the sake of clarity, a constant is added to the objective function in order to allow for using a logarithmic scale. Candidate samples are clearly distinguished according to the criterion used to select them (LHS, EI-like and GA). The solid dark grey line marks the advancement of the minimum value of the true objective function. As in the airfoil case, the initial LHS stage (dark grey diamonds) provides only for the initialization of the surrogate, as the aerodynamic performance is very poor. The objective-driven in-fill points (light grey squares) contribute to progressively drop the objective function levels prior to the final, steepest downhill during the GA-assisted stage (black circles). Unlike the previous case, here the best candidate is found during the GA-assisted search.

Figure 12 shows the progress of the true objective function minimum for all optimization studies. The x-axis (progressive number of candidates) is reported in

Table 5 Comparison of aerodynamic coefficients of the baseline wing

Mach	AOA (°)	C_L	C_D	C_M	Obj. function
0.76	0.5	0.504	0.0237	−0.0722	−16.323
0.78	0.5	0.510	0.0283	−0.0785	−9.927
0.78	1.0	0.563	0.0337	−0.0781	−8.776

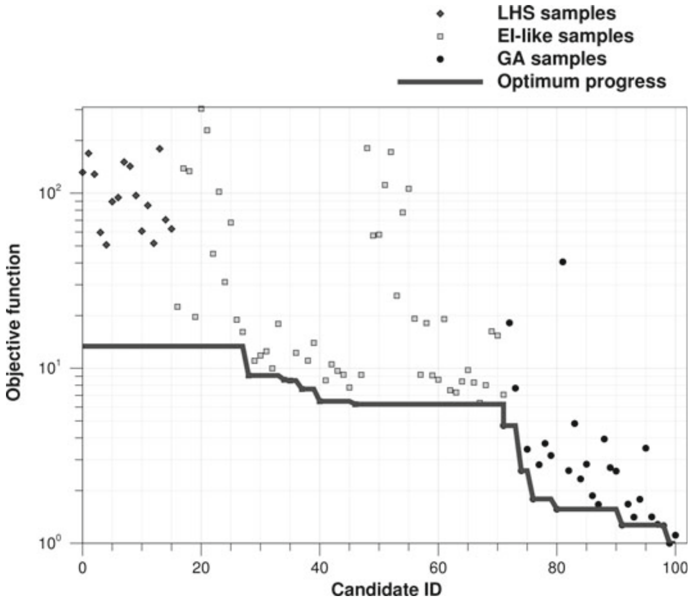


Fig. 11 Objective function history of DPW wing SBOSA-EIL optimization run

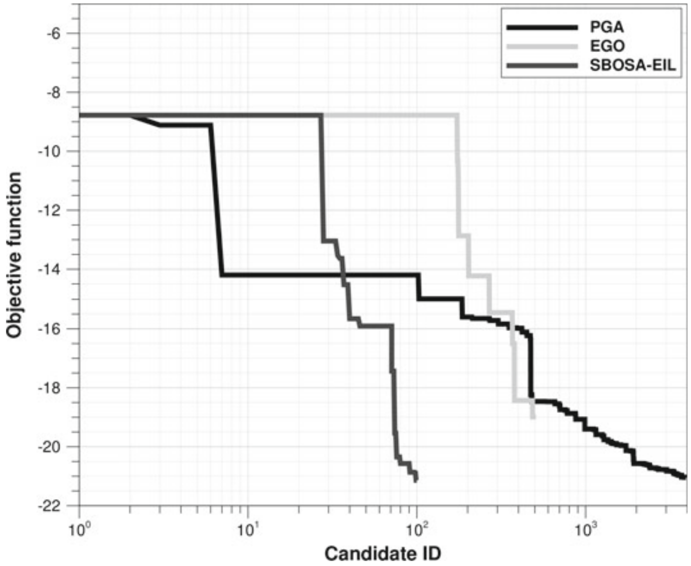


Fig. 12 Progress of minimum values of the objective function for DPW wing optimization studies

logarithmic scale for the sake of clarity as different scales are involved. The SBOSA-EIL optimization, despite the very limited number of CFD evaluations, manages to achieve the same performance level with respect to the plain GA. The EGO algorithm once again turns out to stay somewhat in the middle in terms of computational budget and performance. To better compare the algorithms, Table 6 summarizes the results of each optimization study in terms of aerodynamic coefficients, minimum objective function value attained, number of effective CFD evaluations needed to capture it and a measure of CFD evaluations saving with respect to PGA run. It can be observed that PGA and SBOSA-EIL achieve the same objective function minimum value, although the two optima candidates are clearly different as they exhibit different aerodynamic coefficients. This suggests that the optimization case features non-unique optima solutions. However, the valuable point of such a comparison lies in the fact that, by employing surrogate-assisted procedures coupled to adaptive objective-driven training, the global optimum can be detected by exploiting only 3% of the computational budget of a genetic algorithm optimization. Table 7 summarizes the aerodynamic coefficients and objective function values for DPW-W1, SBOSA-EIL and PGA candidates as computed on the fine mesh: small deviations can be observed with respect to the coarse mesh, a slight loss in the lift coefficient for PGA optimum deteriorates the objective performance due to the triggering of the corresponding penalty. On the other hand, the goal function of SBOSA-EIL optimum is even better due to lower drag contribution and lower penalization of the pitching moment coefficient.

In order to evaluate the difference between optimal candidates, aerodynamic comparisons are proposed in the following figures. Figure 13 shows the contour map of the pressure coefficient on the upper surface of each optimal candidate. A general reduction of the wing loading can be observed which allowed to reduce the shock

Table 6 Optimal candidates comparison

ID	C_L	C_D	C_M	Obj. function	CFD evaluations	CFD saving
Baseline	0.563	0.0337	−0.0781	−8.776	–	–
PGA	0.500	0.0236	−0.0522	−21.14	3700	0.0
EGO	0.502	0.0260	−0.0554	−18.99	485	−87%
SBOSA-EIL	0.514	0.0242	−0.0529	−21.14	100	−97%

Table 7 Optimal candidates comparison, fine mesh

ID	C_L	C_D	C_M	Obj. function
Baseline	0.569	0.0340	−0.0796	−7.970
PGA	0.492	0.0228	−0.0489	−21.013
SBOSA-EIL	0.507	0.0234	−0.0505	−21.638

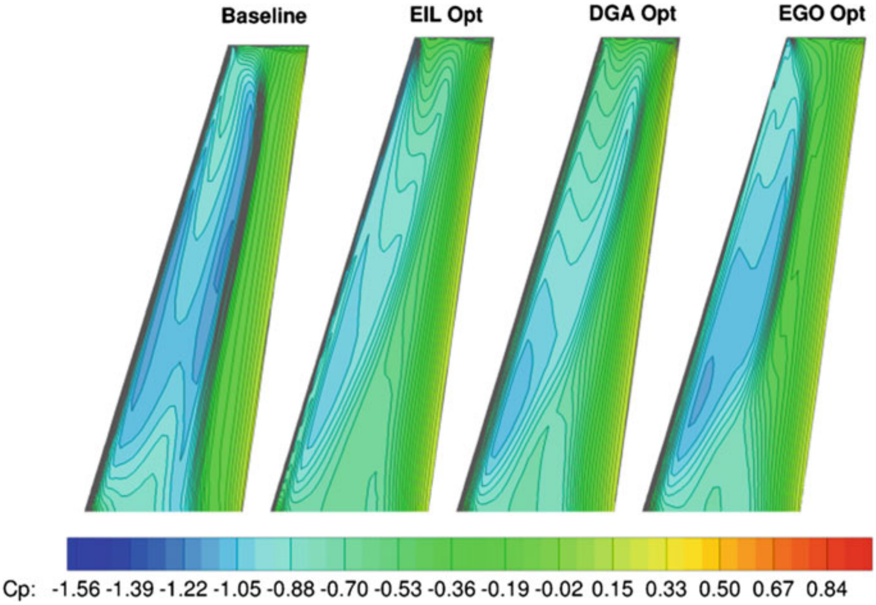


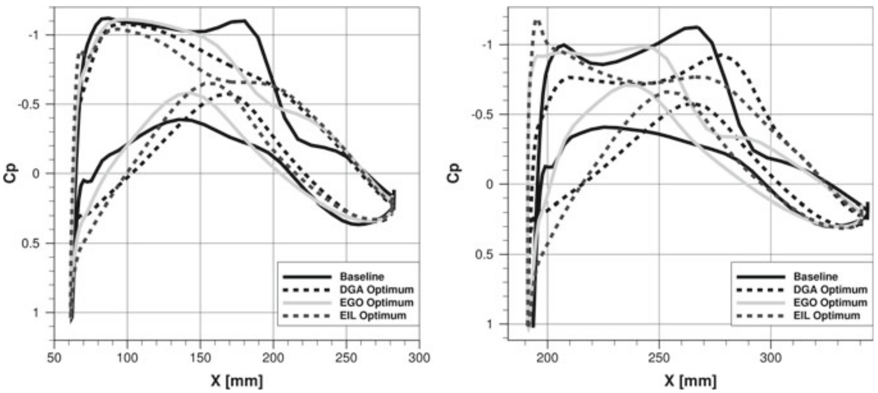
Fig. 13 DPW wing optimal candidates comparison, pressure coefficient contour map

wave strength. This is also confirmed by the fact that the lift coefficient of all candidates are lower than the baseline value and close to the constraint threshold. The SBOSA-EIL solution is very similar to the PGA one, some slight differences are observed in the outboard wing isobars.

Moreover, the shape modification worked to push the wing loading forward in order to satisfy the pitching moment constraint. This feature is more clear in Fig. 14, where the sectional pressure distribution at two spanwise sections (26 and 82% of the span length) is shown. The PGA and SBOSA-EIL solutions present different design choices on the outboard wing, the former being more flat and less loaded, the latter showing a peak near the leading edge followed by a gentle compression. For the sake of completeness, the section shape comparison is also provided in Fig. 15. The most evident geometry modifications can be summarized as follows:

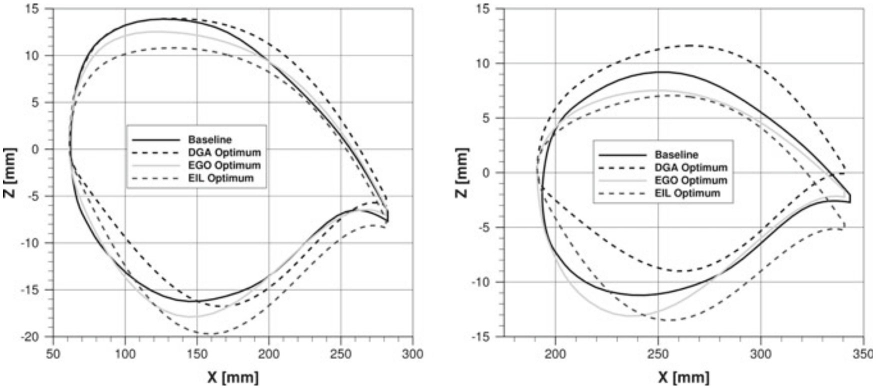
- increase of the outboard twist angle for the SBOSA-EIL optimum candidate;
- airfoil de-cambering for SBOSA-EIL optimum to compensate for twist angle increase;
- reduction of leading edge radius for all the optimized shapes.

Of course, each design choice is strictly related to other design features, but only their combination affects the objective function/constraint evaluation. For instance, PGA and SBOSA-EIL represent two different design examples, however they share the same level of performance in light of the chosen objective function. Indeed, in the former case the twist angle distribution in slightly higher than the DPW wing



(a) Inboard section, $\frac{y}{b} = 0.26$ (b) Outboard section, $\frac{y}{b} = 0.82$

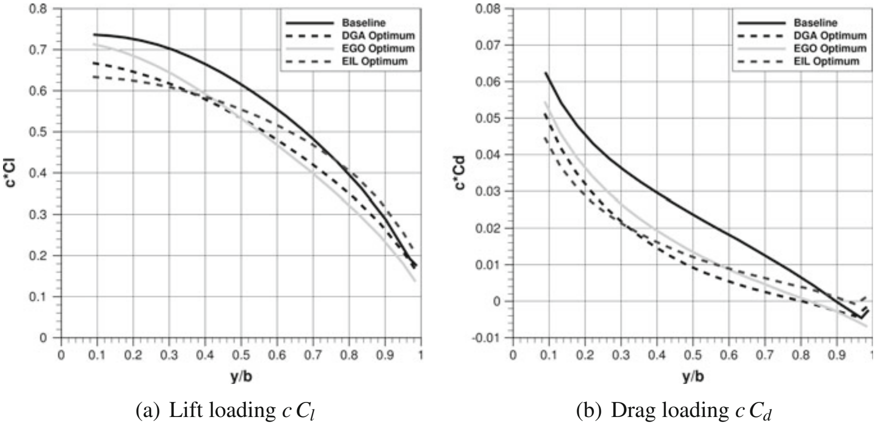
Fig. 14 DPW wing optimal candidates comparison, sectional pressure distribution



(a) Inboard section, $\frac{y}{b} = 0.26$ (b) Outboard section, $\frac{y}{b} = 0.82$

Fig. 15 DPW wing optimal candidates comparison, section shape

but the wing airfoils are shaped to lower the wing loading along the wing span; in the latter case, the section shape in the inboard wing region is designed to minimize the wing loading in order to reduce the main source of wing pressure drag as much as possible and, to compensate this effect, the outboard wing twist is increased. Figure 16 provides a clarifying overview of such a concept as it reports the wing lift and (pressure) drag loading along the wing spanwise direction. It can be observed how the drag levels on the inboard region are reduced for the SBOSA-EIL candidate by reducing the sectional lift accordingly, while the PGA optimum exhibits very low drag contribution on the mid-outboard wing.



(a) Lift loading $c\,C_l$

(b) Drag loading $c\,C_d$

Fig. 16 DPW wing optimal candidates comparison, wing span loading

7 Conclusions

The paper proposed a surrogate-assisted methodology suitable to aerodynamic shape optimization. A physics-based surrogate model coupling Proper Orthogonal Decomposition and Radial Basis Functions interpolation has been exploited to predict approximate values of the objective functions throughout the optimization process. The surrogate model database has been split in three stages, namely a space-filling Latin Hypercube stage to initialize the surrogate, an adaptive sampling stage in which the model is gradually improved and a final optimization stage in which optimal candidates predicted by the surrogate model are re-injected in the database and the model is updated accordingly. The adaptive sampling phase consists in applying two *ad hoc* in-fill criteria which have been purposely designed to enrich the surrogate model database towards the realization of the exploration/exploitation trade-off. The first in-fill criterion is designed to mimic the Expected Improvement Function maximization, the second is based on a sort of factorization of the exploration and exploitation effects.

Two aerodynamic cases have been proposed to test the methodology: the shape optimization of the well-known RAE 2822 airfoil and of an isolated wing from the AIAA CFD Drag Prediction Workshops. In the first case, the results obtained by applying three in-fill strategies compare very well with classical evolutionary-based optimization and surrogate-based EGO algorithm, taken as references. In particular, the same aerodynamic performance level (-40% in terms of objective function) of the computationally intensive genetic optimization can be reached by tuning the σ parameter of the Expected Improvement-like criterion. The second test case involves the solution of the flow field around an isolated wing in transonic viscous flow, hence the number of CFD degrees of freedom is quite larger with respect to two-dimensional cases. Moreover, the design optimization problem has been intentionally made more difficult in order to stress the proposed methodology. Despite the increased intrinsic complexity of the case, results are even better as the aerodynamic performance of the global optimum (-140% in terms of objective function), as predicted by the high-fidelity optimization, is achieved by the surrogate-assisted approach at very limited computational cost (only 3% of the high-fidelity optimization).

Such results support the conclusion that surrogate models alone may not provide the right answer within an aerodynamic shape optimization context, especially if transonic viscous flow is considered. However, when coupled to smart adaptive sampling techniques, they allow to catch the basic trends of the objective function without penalizing the design space exploration: indeed, in complex design cases with high non-linearities and multi-modal landscapes, the latter has to be carefully balanced as it may result in unveiling promising regions as well as leading the optimizer to waste time in searching poor solutions.

References

1. Adams, B., Bauman, L., Bohnhoff, W., Dalbey, K., Ebeida, M., Eddy, J., Eldred, M., Hough, P., Hu, K., Jakeman, J., Swiler, L., Vigil, D.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.4 user's manual. Technical report SAND2010-2183, Sandia, Apr 2013
2. Amato, M., Catalano, P.: Non linear κ ε turbulence modeling for industrial applications. In: ICAS 2000 Congress. IOS Press, Harrogate, UK (2000)
3. Chandrashekarappa, P., Duvigneau, R.: Radial basis functions and kriging metamodels for aerodynamic optimization. Rapport de recherche RR-6151, INRIA (2007). http://hal.inria.fr/inria-00137602/en/
4. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. **45**(1-3), 50–79 (2009). http://dx.doi.org/10.1016/j.paerosci.2008.11.001
5. Gutmann, H.M.: A radial basis function method for global optimization. J. Glob. Optim. **19**, 201–227 (2001). http://dl.acm.org/citation.cfm?id=596093.596381. https://doi.org/10.1023/A:1011255519438
6. Iuliano, E.: Towards a pod-based surrogate model for CFD optimization. In: Proceedings of the Eccomas CFD & Optimization Conference, Antalya, Turkey (2011)
7. Iuliano, E.: Adaptive Sampling Strategies for Surrogate-based Aerodynamic Optimization, Springer Tracts in Mechanical Engineering, vol. Springer, Application of Surrogate-based Global Optimization to Aerodynamic Design (2015)
8. Iuliano, E., Quagliarella, D.: Surrogate-based aerodynamic optimization via a zonal pod model. In: Proceedings of the EUROGEN 2011 Conference, Capua, Italy (2011)
9. Iuliano, E., Quagliarella, D.: Aerodynamic shape optimization via non-intrusive pod-based surrogate modelling. In: Proceedings of 2013 IEEE CEC Congress on Evolutionary Computation (2013)
10. Iuliano, E., Quagliarella, D.: Proper orthogonal decomposition, surrogate modelling and evolutionary optimization in aerodynamic design. Comput. Fluids **84**, 327–350 (2013)
11. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**, 455–492 (1998). https://doi.org/10.1023/A:1008306431147. http://dx.doi.org/10.1023/A:1008306431147
12. Kulfan, B.M.: Universal parametric geometry representation method. J. Aircr. **45**(1), 142–158 (2008)

13. Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidyanathan, R., Kevintucker, P.: Surrogate-based analysis and optimization. Prog. Aerosp. Sci. **41**(1), 1–28 (2005). http://linkinghub.elsevier.com/retrieve/pii/S0376042105000102
14. Schonlau, M., Welch, W.J., Jones, D.R.: Global Versus Local Search in Constrained Optimization of Computer Models. Lecture Notes-Monograph Series, vol. 34 (1998). https://doi.org/10.2307/4356058. http://dx.doi.org/10.2307/4356058
15. Simpson, T.W., Toropov, V.V., Balabanov, V., Viana, F.A.C.: Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come—or not. In: Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA 2008–5802, pp. 1–22. American Institue of Aeronautics and Astronautics (2008)
16. Sóbester, A., Leary, S., Keane, A.: A parallel updating scheme for approximating and optimizing high fidelity computer simulations. Struct. Multidiscip. Optim. **27**, 371–383 (2004). https://doi.org/10.1007/s00158-004-0397-9. http://dx.doi.org/10.1007/s00158-004-0397-9
17. Vitagliano, P.L., Quagliarella, D.: A hybrid genetic algorithm for constrained design of wing and wing-body configurations. In: Bugeda, G., Désidéri, J.A., Périaux, J., Schoenauer, M., Winter G. (eds.) Evolutionary Methods for Design, Optimization and Control Applications to Industrial and Societal Problems. International Center for Numerical Methods in Engineering (CIMNE), Barcelona, Spain (2003)