Springer Tracts in Mechanical Engineering

Emiliano Iuliano
Esther Andrés Pérez *Editors*

Application of Surrogate-based Global Optimization to Aerodynamic Design

Springer

Springer Tracts in Mechanical Engineering

About this Series

Springer Tracts in Mechanical Engineering (STME) publishes the latest developments in Mechanical Engineering - quickly, informally and with high quality. The intent is to cover all the main branches of mechanical engineering, both theoretical and applied, including:

- Engineering Design
- Machinery and Machine Elements
- Mechanical structures and stress analysis
- Automotive Engineering
- Engine Technology
- Aerospace Technology and Astronautics
- Nanotechnology and Microengineering
- Control, Robotics, Mechatronics
- MEMS
- Theoretical and Applied Mechanics
- Dynamical Systems, Control
- Fluids mechanics
- Engineering Thermodynamics, Heat and Mass Transfer
- Manufacturing
- Precision engineering, Instrumentation, Measurement
- Materials Engineering
- Tribology and surface technology

Within the scopes of the series are monographs, professional books or graduate textbooks, edited volumes as well as outstanding PhD theses and books purposely devoted to support education in mechanical engineering at graduate and post-graduate levels.

More information about this series at http://www.springer.com/series/11693

Emiliano Iuliano • Esther Andrés Pérez
Editors

Application of Surrogate-based Global Optimization to Aerodynamic Design

Springer

Editors
Emiliano Iuliano
CIRA, the Italian Aerospace Research Center
Capua (CE), Italy

Esther Andrés Pérez
INTA, Spanish National Institute for Aerospace Technology
Torrejón de Ardoz, Spain

ISSN 2195-9862 ISSN 2195-9870 (electronic)
Springer Tracts in Mechanical Engineering
ISBN 978-3-319-21505-1 ISBN 978-3-319-21506-8 (eBook)
DOI 10.1007/978-3-319-21506-8

Library of Congress Control Number: 2015951829

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

To our parents

Lucio and Lucia
Ricardo and Carmen

Preface

Essentially, all models are wrong, but some are useful
George Edward Pelham Box

Aircraft design, as many other engineering applications, is increasingly relying on computational power. The growing need for multi-disciplinarity and high-fidelity in design optimization for industrial applications implies a huge number of repeated simulations to find an optimal design candidate. The main drawback is that each simulation can be computationally expensive: this turns out to be an even bigger issue when used within parametric studies, automated search or optimization loops which typically may require thousands of analysis evaluations. In recent years, global optimization by meta-models has been widely applied to design exploration in order to rapidly investigate the design space and find sub-optimal solutions. Indeed, surrogate and reduced order models can provide a valuable alternative at a much lower computational cost. In this perspective, this book proposes the application of surrogate-based optimization to aerodynamic design cases with reasonable computational resources.

The idea for this book evolved from activities and discussions within the Action Group number 52 (AG-52) of the Group for Aeronautical Research and Technology in Europe (GARTEUR). The Action Group, active from 2012 to 2016, belongs to the GARTEUR-Aerodynamic (AD) research field and aims to explore surrogate-based global optimization methods for aerodynamic shape design. The main objective is, by means of a European collaborative research, to make a deep evaluation and assessment of surrogate-based global optimization methods for aerodynamic shape optimization. At the end of the AG-52, the partners will have improved global shape optimization capabilities and valuable knowledge of the selected set of techniques. Through the proposed work program, it is expected that some "best practice" guidelines will be concluded to ease the use of surrogate-based global optimization methods in aeronautic industries.

The dissemination of the work done within the AG-52 has led to the organization of dedicated special sessions within European Community on Computational Methods in Applied Sciences (ECCOMAS) thematic conferences. The book collects the papers presented within the mini-symposium on the topic held at the 6th European

Conference on Computational Fluid Dynamics (ECFD VI) in 2014. The first two chapters deal with partial results of the AG-52, while the last two chapters include external contributions, which enrich the collaborative network on this topic.

Capua, Italy Emiliano Iuliano
Torrejón de Ardoz, Spain Esther Andrés-Pérez
May 2015

Contents

Contributors

Esther Andrés-Pérez Fluid Dynamics Branch, Spanish National Institute for Aerospace Technology (ISDEFE/INTA), Torrejón de Ardoz, Spain

Universidad Politécnica de Madrid (UPM), Madrid, Spain

Varvara G. Asouti Parallel CFD & Optimization Unit, National Technical University of Athens, Athens, Greece

Leopoldo Carro-Calvo Department of Signal Theory and Communications, Universidad de Alcalá (UAH), Alcalá de Henares, Spain

Khairy Elsayed Faculty of Engineering, Department of Mechanical Engineering, Vrije Universiteit Brussel, Brussels, Belgium

Kyriakos C. Giannakoglou Parallel CFD & Optimization Unit, National Technical University of Athens, Athens, Greece

Emiliano Iuliano Fluid Mechanics Department, Multidisciplinary Analysis & Design Optimization Group, CIRA, The Italian Aerospace Research Center, Caserta, Italy

Stylianos A. Kyriacou Parallel CFD & Optimization Unit, National Technical University of Athens, Athens, Greece

Chris Lacor Faculty of Applied Sciences, Department of Mechanical Engineering, Vrije Universiteit Brussel, Brussels, Belgium

Mario J. Martin-Burgos School of Aeronautics, Universidad Politécnica de Madrid (UPM), Madrid, Spain

Sancho Salcedo-Sanz Department of Signal Theory and Communications, Universidad de Alcalá (UAH), Alcalá de Henares, Spain

Acronyms

ANNs	Artificial Neural Networks
AoA	Angle of attack
CFD	Computational Fluid Dynamics
CV	Cross Validation
$C_{D,d}$	Drag coefficient
$C_{L,l}$	Lift coefficient
$C_{M,m}$	Pitching moment coefficient
C_p	Pressure coefficient
DACE	Design and Analysis of Computer Experiments
DOE	Design of Experiments
DPW	Drag Prediction Workshop
EA	Evolutionary algorithm
EGO	Efficient Global Optimization
EI	Expected Improvement
IES-SL	Intelligent Estimation Search with Sequential Learning
KG	Kriging
LHS	Latin Hypercube Sampling
LOO	Leave One Out
LS	Least Squares
MAEAs	Metamodel-Assisted Evolutionary Algorithms
MARS	Multivariate Adaptive Regression Splines
MSE	Mean Squared Error
NSGA	Non-dominated Sorting Genetic Algorithm
NURBS	Non-Uniform Rational B-Splines
PCA	Principal Component Analysis
POD	Proper Orthogonal Decomposition
RBF	Radial Basis Functions
RMSE	Root Mean Squared Error
R^2,R-squared	Coefficient of determination
ρ	Pearson correlation coefficient
SBGO	Surrogate-Based Global Optimization

SBO	Surrogate-Based Optimization
SBSO	Surrogate-Based Shape Optimization
SVD	Singular Value Decomposition
SVMs	Support Vector Machines
SVR,SVMr	Support Vector Machines Regression algorithms

Chapter 1
Aerodynamic Shape Design by Evolutionary Optimization and Support Vector Machines

Esther Andrés-Pérez, Leopoldo Carro-Calvo, Sancho Salcedo-Sanz, and Mario J. Martin-Burgos

Abstract This paper proposes a computational methodology for the aerodynamic shape design of aeronautical configurations, aiming a broad and efficient exploration of the design space. A novel adaptive sampling technique focused on the global optimization problem, the Intelligent Estimation Search with Sequential Learning (IES-SL), is presented. This approach is based on the use of Support Vector Machines (SVMs) as the surrogate model for estimating the objective function, in combination with an evolutionary algorithm (EA) to enable the discovery of global optima. The proposed methodology is applied to improve the aerodynamic performance of a two-dimensional airfoil and a three-dimensional wing and results on surrogate model validation and optimization-focused sampling criteria are discussed.

1.1 Introduction

Aerodynamic shape optimization is nowadays acquiring a huge importance in the aeronautical industries, which are aiming to reduce both fuel consumption and contaminants emissions to be competitive in a world globalized environment. In order to emphasize the importance of drag minimization and therefore fuel

E. Andrés-Pérez (✉)
Fluid Dynamics Branch, Spanish National Institute for Aerospace Technology (ISDEFE/INTA), Ctra. Ajalvir, km. 4, 28850 Torrejón de Ardoz, Spain
e-mail: eandres@isdefe.es

L. Carro-Calvo • S. Salcedo-Sanz
Department of Signal Theory and Communications, Universidad de Alcalá (UAH), Ctra. Madrid-Barcelona km. 33, 28802 Alcalá de Henares, Spain
e-mail: leopoldo.carro@uah.es; sancho.salcedo@uah.es

M.J. Martin-Burgos
School of Aeronautics, Universidad Politécnica de Madrid (UPM), Plaza del Cardenal Cisneros 3, 28040 Madrid, Spain
e-mail: mariojmartin@live.com

E. Iuliano, E.A. Pérez (eds.), *Application of Surrogate-based Global Optimization to Aerodynamic Design*, Springer Tracts in Mechanical Engineering,
DOI 10.1007/978-3-319-21506-8_1

requirements, based on the Breguet range equation, an airline operator would have to reduce the payload, and therefore the incomes, by 7.6 % to recover a 1 % of increase in drag [1]. In the current scenario, where most airlines operate with small benefit margins, this illustrates the importance of small drag reductions which may even signify a profit or non-profit business. This explains why computational fluid dynamics (CFD) driven aerodynamic shape design has been increasingly applied during the past decades, mostly for providing local deformations from a given baseline geometry. However, today's aircraft configurations are reaching their limit in performance and no further significant improvements are expected by producing only small modifications of their original shapes. The design of new concepts needs to be faced with a mixture of high accuracy methods, efficient global optimization strategies, flexible geometry parameterizations and broad exploration and exploitation of the design space.

Non-deterministic methods have the ability to work with noisy objective functions without assumptions on continuity and they have a high potential to find the global optimum of complex problems involving a large amount of design parameters. Therefore, these methods seem to be appropriate when a broad exploration of the design space is desired, as occurs when looking for non-conventional shapes. However, they require a vast number of evaluations to obtain the optimum even for a small number of variables. Each evaluation would require a complete CFD analysis which makes the method unfeasible, in terms of computational cost. To overcome this drawback, surrogate evaluation models or metamodels can be used. The metamodel is the inexpensive and approximate model of the evaluation problem. For aerodynamic design optimization using evolutionary algorithms, the metamodel could be used to calculate the fitness (i.e. aerodynamic coefficients, lift and drag) of the solutions, that means to replace the CFD tool.

Apart from the robustness, the potential applicability in different disciplines, and the inherent parallelization of these algorithms, their feasibility in an industrial environment has currently serious limitations, for example, the number of design variables to be considered and the time required by the optimization process. Therefore, there are still some important open issues as the capability to deal with the so-called curse of dimensionality, the reduction of the design space, the use of gradients information within the global optimization process, the use of multiple fidelity models and an efficient constraints handling. Particularly, the strategy to properly choose the sampling data to build the surrogate model is of enormous importance to achieve an adequate model prediction accuracy that allows to guide the optimization process towards the real global optima. This Design of Experiments (DoE) strategy heavily depends on the type of surrogate model, its output function and the way it is integrated within the optimization algorithm, aiming a good trade-off between the explorative and predictive capabilities of the surrogate, which means to allow a broad exploration of the design space while providing enough accuracy in those regions where predicted minima is located.

In this work, a surrogate-based global optimization (SBGO) method is applied to the multi-point aerodynamic shape optimization of a two-dimensional airfoil and a three-dimensional wing. A novel adaptive sampling technique focused on the optimization problem, the Intelligent Estimation Search with Sequential Learning (IES-SL), is presented. This approach is based on the use of Support Vector Machines (SVMs) as the surrogate model for estimating the objective function in combination with an evolutionary algorithm (EA) to enable the discovery of global optima.

This paper is structured as follows: First, previous works on surrogate-enhanced evolutionary algorithms are discussed and the proposed approach is described, giving details on the EA and SVM algorithms. Then, the experimental part of the paper is explained, where results on the surrogate model performance are displayed. Finally, some remarks on the feasibility of the proposed approach in case of designing industrial configurations are outlined.

1.2 Literature Review

In the last few years, there has been an increasing interest in the topic of SBGO methods for aerodynamic shape design. This section mentions only some of the very recent works focusing on SBGO with application to the aerodynamic shape design of aeronautical configurations.

In [2] a physics-based surrogate approach exploiting space mapping is applied to the drag minimization of NACA0012 and RAE2822 transonic airfoils. In that work, the airfoil geometries were parameterized using PARSEC and involving five to ten design parameters. In [3], surrogate-based optimization strategies are applied to the drag minimization of an NLF0416 airfoil with ten design parameters. In addition, Koziel and Leifsson [4] applied a shape optimization strategy using variable-fidelity CFD models to the optimization of a transonic airfoil parameterized by the NACA four-digit definition with three design variables. The work in [5] employs an active subspace method for effectively searching the design space in the optimization of the ONERA M6 transonic wing, parameterized with 50 Free-Form Deformation (FFD) design variables. In [6, 7] a surrogate based on Proper Orthogonal Decomposition (POD) is applied to the aerodynamic shape optimization of an airfoil geometry parameterized by 16 design variables defined with Class Shape Transformation (CST) method. Other approach is presented in [8], where a combination of a genetic algorithm (GA) and an artificial neural network (ANN) is applied to the shape optimization of an airfoil, parameterized by a modified PARSEC parameterization involving ten design variables.

Different kinds of surrogate modelling techniques have been proposed in the literature, for example, Polynomial Regression (PR), Multivariate Adaptive Regression Splines (MARS), Gaussian Processes, Kriging (KG), cokriging [9], ANN [10, 11], Radial Basis Functions (RBF) [12], SVM [13, 14] and Ensemble methods [15], among others. Furthermore, the use of Support Vector Regression algorithms (SVMr) as metamodels has been applied to a large variety of regression problems, in many of them mixed with evolutionary algorithms [16, 17].

With respect to the combination of global and local search methods within the design optimization process, the so-called hierarchical approach has been proposed in recent papers [18, 19]. Metamodel-assisted memetic algorithms are hybrid schemes that combine, for example, stochastic methods for the exhaustive search of the design space along with gradient-based methods for the refinement of promising solutions.

With the aim of providing a comprehensive survey about different surrogate methods for surrogate-based aerodynamic shape optimization, a GARTEUR Action Group (AD/AG52) started [20] at the beginning of 2013. Within this Group, research activities are planned over a three-year period, with the objective of performing a fair comparison between different surrogate modelling methods applied to the aerodynamic optimization of baseline geometries, sharing the parameterization and mesh deformation algorithms.

1.3 Proposed SBGO Approach

The proposed SBGO approach is comprised of several key features:

1. Non-Rational Uniform B-Splines (NURBS) geometry parameterization.
2. High fidelity CFD solver: The DLR TAU solver
3. Low fidelity: SVMs as the surrogate model
4. Sampling DoE strategy focused directly on optimization: IES-SL
5. Evolutionary optimization algorithm

This section details each brick of this approach.

1.3.1 Geometry Parameterization with Non-rational Uniform B-Splines

The parameterization is crucial in an aerodynamic design optimization problem, where the selection of the design parameters not only affects significantly the performance of the optimization, but also the optimal solution to be obtained. NURBS have been suggested as an efficient and flexible parameterization [21–23], able to represent complex configurations, giving the optimizer enough freedom to converge to a wide range of possible optimal designs. Moreover, NURBS have demonstrated to be able to accurately represent a large family of geometries given their main advantage of providing a global parameterization with a smooth surface while still maintaining the locality in the deformation [24]. In addition, the optimized surface at the end of the optimization process has the correct format to feed directly the CAD and grid generation applications.

From a mathematical point of view, NURBS surfaces are defined as:

$$S\left(\xi,\eta\right) = \frac{\sum_{i}^{I}\sum_{j}^{J} U_{i,p}\left(\xi\right) V_{j,q}\left(\eta\right) w_{ij} C_{ij}}{\sum_{i}^{I}\sum_{j}^{J} U_{i,p}\left(\xi\right) V_{j,q}\left(\eta\right) w_{ij}} \tag{1.1}$$

where C are the control points spatial coordinates, w are the control points weights, and U and V are the basis functions which are calculated using the following expression:

$$\begin{aligned} U_{i,1}\left(\xi\right) &= \begin{cases} 1 \; if \; u_i \leq \xi < u_{i+1} \\ 0 \quad otherwise \end{cases} \\ U_{i,k}\left(\xi\right) &= \frac{(\xi - u_i) U_{i,k-1}(\xi)}{u_{i+k-1} - u_i} + \frac{\left(u_{i+k} - \xi\right) U_{i+1,k-1}(\xi)}{u_{i+k} - u_{i+1}} \end{aligned} \tag{1.2}$$

The basis coefficients are calculated from the knots vectors $\bar{U}$ and $\overline{V}$ which are a sequence of real numbers.

Basis functions are equal to zero everywhere except for an interval delimited by the order of the NURBS, defining the area of influence of each control point. This paper does not aim to provide an exhaustive review of this parameterization technique. A more extended reference for NURBS can be found in [25].

In particular, for the experiments performed in this paper, volumetric B-Splines have been used. The mathematical expression is:

$$B\left(u,v,w\right) = \sum_{i}^{I}\sum_{j}^{J}\sum_{k}^{K} U_{i,pu}(u) V_{i,pv}(v) W_{i,pw}(w) C_{ijk} \tag{1.3}$$

where pu, pv, pw are the polynomial orders, C are the control points' Cartesian coordinates, and U, V and W are the basis functions, calculated with the expression (1.2), since they are the same than in case of surface NURBS.

1.3.2 The DLR TAU Solver

The fluid flow over the object of interest is simulated with the TAU Code [26, 27]. The unsteady TAU-Code solves the compressible, three-dimensional Reynolds-Averaged Navier–Stokes equations using a finite volume formulation. The TAU-Code is based on a hybrid unstructured-grid approach, which makes use of the advantages of semi-structured prismatic grids in the viscous shear layers near walls, and the flexibility in grid generation offered by tetrahedral grids in the surrounding flow volume. A dual-grid approach with an edge-based data structure is used in order to make the flow solver independent from the cell types used in the initial grid. The TAU-Code consists of several different modules, including:

- The Grid Partitioner, which splits the primary grid into n number of subgrids for n processors.
- The Preprocessor module, which uses the information from the initial grid to create a dual-grid and secondly coarser grids for multi-grid.
- The Solver module, which performs the flow calculations on the dual-grid.
- The Adaptation module, which refines and de-refines the computational grid with different indicator functions.
- The Deformation module, which propagates the deformation of surface grid points to the surrounding volume grid.
- The Post-processing module, which is used to convert result files to formats usable by popular visualization tools.

Together all modules are available with Python interfaces for computing complex application, e.g. unsteady cases, complete force polar curves or fluid-structure couplings in an automatic framework. Furthermore, it eases the usage on highly massive parallel computers to execute applications.

The TAU code also includes the following modules for handling NURBS parameterization:

- The *Point Inversion* module, which calculates the parametric coordinates of each surface mesh point over the NURBS.
- The *Mesh to NURBS* module, which deforms the surface mesh in order to adapt it to the NURBS.
- The *NURBS Sensitivity* module, which maps the sensitivities of the surface mesh points onto the NURBS control points, considering also the geometric derivatives.
- The *NURBS deformation* module, which performs NURBS deformation in order to reach the goal given by the objective function with a steepest descent algorithm.

1.3.3 SVMs as Surrogate Model

SVMs for regression (SVMr) are a powerful tool used on the machine learning field [28], and as a modelling tool for a large amount of regression problems on engineering [29]. The SVMr can be solved as a convex optimization problem using kernel theory to face nonlinear problems. The SVMr consider not only the prediction error but also the generalization of the model.

The SVMr consist of training a model with the form $y = w^T\varphi(x) + b$ given a set of parameters $C = \left\{\left(x_i, y_j\right), i = 1, \ldots, l\right\}$, to minimize a general risk function of the form:

$$R\left[f\right] = \frac{1}{2}\|w\|^2 + \frac{1}{2}C\sum_{i=1}^{l} L\left(y_i,f(x)\right) \tag{1.4}$$

where w controls the smoothness of the model, φ(x) is a function of projection of the input space to the feature space, b is a parameter of bias, x_i is a feature vector of the input space with dimension N, y_i is the output value to be estimated and $L(y_i,f(x))$ is the loss function selected. In this paper, the L1-SVR (L1 support vector regression) is used, characterized by an ε-insensitive loss function

$$L\left(y_i,f(x)\right) = \left|y_i - f\left(x_i\right)\right|_\varepsilon \tag{1.5}$$

In order to train this model, it is necessary to solve the following optimization problem

$$\min\left(\frac{1}{2}\|w\|^2 + \frac{1}{2}C\sum_{i=1}^{l}\left(\xi_i + \xi_i{}^*\right)\right) \tag{1.6}$$

subject to:

$$\begin{aligned} &y_i - w^T\varphi\left(x_i\right) - b \leq \varepsilon + \xi_i, i = 1, \ldots, l \\ &-y_i + w^T\varphi\left(x_i\right) + b \leq \varepsilon + \xi_i{}^*, i = 1, \ldots, l \\ &\xi_i, \xi_i{}^* \geq 0, i = 1, \ldots, l \end{aligned} \tag{1.7}$$

To do this, a dual form is usually applied, obtained from the minimization of the Lagrange function that joins the function to minimize and the restrictions. The dual form is:

$$\begin{aligned} &\max\left(-\frac{1}{2}\sum_{i,j=1}^{l}\left(\alpha_i - \alpha_i{}^*\right)\left(\alpha_j - \alpha_j{}^*\right)K\left(x_i, x_j\right) - \right. \\ &\left.\varepsilon\sum_{i=1}^{l}\left(\alpha_i + \alpha_i{}^*\right) + \sum_{i=1}^{l} y_i\left(\alpha_i + \alpha_i{}^*\right)\right) \end{aligned} \tag{1.8}$$

subject to:

$$\sum_{i=1}^{l}\left(\alpha_i - \alpha_i{}^*\right) = 0; \alpha_i, \alpha_i{}^* \in [0, C] \tag{1.9}$$

In addition to the restrictions, also must be taken into account the Karush–Kuhn–Tucker conditions and obtain the bias value. For simplicity they will be omitted. A complete description can be consulted on [28].

In the dual formulation we must emphasize the apparition of the kernel function $K(x_i, x_j)$, which is equivalent to the scalar product $\langle \varphi(x_i), \varphi(x_j) \rangle$. In our case, the kernel function is a Gaussian function:

$$K\left(x_i, x_j\right) = e^{-\gamma \cdot \left\| x_i - x_j \right\|^2} \tag{1.10}$$

The final form of the regression model depends on the Lagrange multipliers α_i, α_i^*, following the expression:

$$f(x) = \sum_{i=1}^{l} \left(\alpha_i - \alpha_i{}^*\right) K\left(x_i, x\right) + b \tag{1.11}$$

In this way, the SVMr model depends on three parameters, ε, C and γ. The ε parameter controls the error margin permitted for the model, as can be seen in Eqs. (1.6) and (1.7), and the C parameter controls the number of outliers allowed on the optimization of the function [Eq. (1.6)]. Finally the γ parameter determines the Gaussian variance for the kernel. Depending on the selection of these values, the model can have a different performance. To obtain the best SVM performance, a search of the most suitable combination of these three parameters must be carried on, usually by using cross-validation techniques over the training set. To reduce the computational time of this process, different methods have been proposed in the literature to reduce the search space related to these parameters. In this case, it has been applied to the one developed by Ortiz-García et al. [30], which has proven to require pretty short search times.

1.3.4 Evolutionary Optimization Algorithm

Evolutionary Algorithms (EAs) are bio-inspired methods that mimic the behaviour of natural evolution to solve complex optimization problems. The main elements of an EA are the solution coding, the selection operator and the crossover and mutation operators. The EA implemented for this work has the following characteristics:

1. *Solution coding*: The solutions are coded as a vector of real values from 0 to 1. Each element of the vector represents the value of a normalized parameter.
2. *Selection operator*: In the proposed EA, the selection operator is applied by replacing a portion of the current generation by new individuals generated from parents [31]. It is considered the replacement of the individuals in the population

with fitness value under the population's mean fitness. First, the mean population fitness is calculated as:

$$\overline{g} = \frac{1}{\xi}\sum_{k=1}^{\xi} g_k \tag{1.12}$$

where g_k is the fitness of the *k*th individual and ξ represents the number of individuals in the population. Every individual in the population with a value of fitness under $\overline{g}$ is discarded, and substituted by a new individual obtained with the crossover operator.

3. *Crossover operator*: generates a new individual from the values of two parents selected randomly from the survival individuals. A multi-point crossover which selects the value of one of the parents with probability 0.5 is applied.
4. *Mutation operator*: The values of each new individual are mutated with probability *1/Np*, where *Np* is the number of parameters to be optimized. This operator changes the initial individual by using this formula:

$$V_i = V_i \cdot (1 + U \cdot \alpha) \tag{1.13}$$

where *Vi* represents each one of the parameters to optimize, U is a uniform noise [0–1] and α is a value which represents the mutation level. Three values of α have been used, 1, 0.1 and 0.01, randomly selected for each new individual with probability 1/3. It can be observed that α = 1 represents a strong change on the initial value. On the other hand, α = 0.01 implies a change of 1 % of the initial value, allowing a local search over this parameter.

1.3.5 Intelligent Estimation Search with Sequential Learning

In this paper, a novel SBGO method is proposed: *The IES-SL*. This method allows performing an efficient adaptive sampling guiding the optimization algorithm towards the most promising regions of the design space. First, an initial set of *N* randomly generated geometries are selected and evaluated with the CFD tool, in this case the TAU code. With this set, a first surrogate is built and linked within an evolutionary algorithm. The latter will search for the minimum of the surrogate in each of the optimization iterations, and the returned optima will be again evaluated using the high-fidelity CFD solver, and then incorporated to the surrogate model, which is rebuilt. The process will end when a certain number of CFDs (called budget) is reached.

The IES-SL is an algorithm designed to implement an adaptive sampling directly focused on the optimization search. From this point of view, the key feature of this

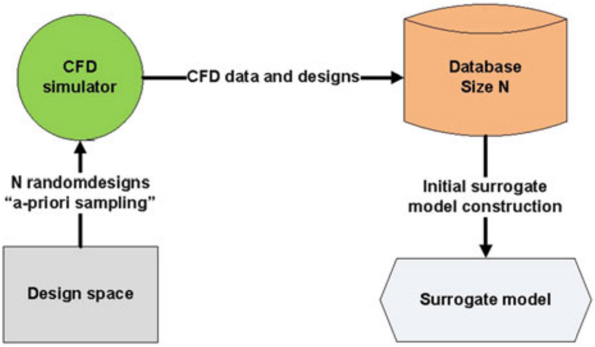


Fig. 1.1 Scheme of the first step of the proposed IES-SL. Initial surrogate model construction with a-priori random sampling of the design space

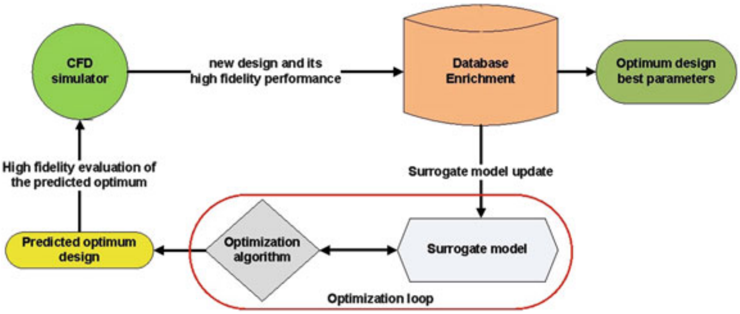


Fig. 1.2 Scheme of the second step of the proposed IES-SL. Adaptive sampling directly focused on the optimization search

new approach is to use the surrogate model to estimate the location of the optimum in the real function. To do this, an optimization search is applied over the surrogate, obtaining an estimated value of the real minimum position (an "intelligent guess"). Each of the estimations of the optimum location gives us a new sampling point (it means a new geometry that is also analysed using the CFD solver). Within a trial-and-error cycle, the surrogate proposes a new design which is again evaluated by the CFD solver and then, in a sequential learning, the surrogate model is enriched with the associated cost function.

The proposed IES-SL is a two-step algorithm. Figure 1.1 shows the first step, where the algorithm is initialized, generating an initial database by evaluating a small number of random designs (four in this case). The initial surrogate model is generated using this database.

Then, in the second step, displayed in Fig. 1.2, the algorithm searches for the position of the optimum value with the surrogate model to use it as an estimation for the real optimum position. The estimated optimum is then evaluated using the CFD solver, obtaining a new pair (design, high fidelity performance) that will enrich the database. Then, the surrogate is updated by adjusting it to the complete database and the cycle is finished, starting again the search for the new sample.

When the maximum number of iterations is reached, the optimum design is obtained as the best parameters on the database. In this way, it is ensured that the design obtained is optimum with respect to the high fidelity simulator system (CFD solver) and not only to the surrogate model, which avoids to find fake optima.

1.4 Numerical Results

1.4.1 Test Cases Definition

The proposed methodology is applied to the aerodynamic shape optimization of an RAE2822 airfoil [32] and a DPW-W1 wing from the Third AIAA Drag Prediction Workshop [33], with the problem formulation defined in Table 1.1.

In particular, the objective function implemented is described in the following formula:

$$f = \frac{1}{2}\left(\frac{C_d + C_{d,t} + C_{d,l}}{C_l}\frac{C_{l,0}}{C_{d,0}}\right)_{DP1} + \frac{1}{2}\left(\frac{C_d + C_{d,t} + C_{d,l}}{C_l}\frac{C_{l,0}}{C_{d,0}}\right)_{DP2} \tag{1.14}$$

where $C_{D,0}$ and $C_{L,0}$ are the drag and lift coefficients of the original geometry and:

$$\begin{aligned} (C_{d,t})_{DPi} &= 0.01 * \max\,(0, C_{m,0} - C_m)_{DPi} \\ (C_{d,l})_{DPi} &= 10 * \max\left(0, {C_{l,0}}^2 - {C_l}^2\right)_{DPi} \end{aligned} \tag{1.15}$$

The optimization of these cases is subject to both aerodynamic and geometric constraints, as displayed in Table 1.2.

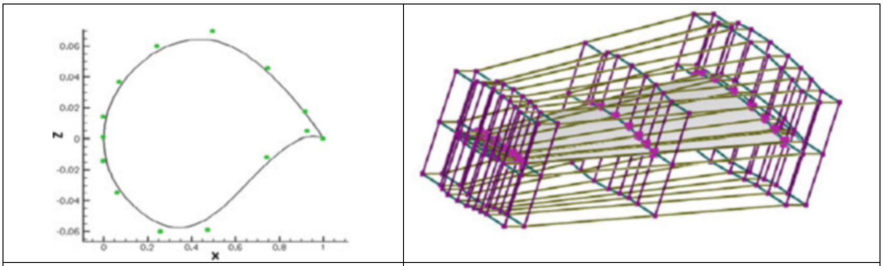
A total budget of CFD computations was defined by five times the number of design variables. Therefore, for the RAE2822 it was possible to perform 70 CFD evaluations, while for the DPW wing, 180 CFDs are allowable for both surrogate model construction and optimization.

Table 1.1 Test cases definition

Configuration	Flow conditions		Objective function
	Design point 1	Design point 2	
RAE2822 airfoil	$C_L = 0.8$, Mach = 0.734, $Re = 6.5 \times 10^6$	$C_L = 0.743$, Mach = 0.754, $Re = 6.2 \times 10^6$	Maximize (C_L/C_D)
DPW-W1 wing	$C_L = 0.5$, Mach = 0.76	$C_L = 0.5$, Mach = 0.78	

Table 1.2 Aerodynamic and geometric constraints

Configuration	Geometric constraints	Aerodynamic constraints
RAE2822 airfoil	(a) Airfoil maximum thickness ratio ≥ 12.1 %	(a) Prescribed minimum lift coefficient $C_L{}^0$
	(b) Airfoil minimum thickness ratio at 0.80c ≥ 4 %	(b) Prescribed minimum pitching moment coefficient $C_M{}^0$
	(c) Minimum leading edge radius = 0.004	$C_M(k) \geq C_M^0(k)$
	(d) Limit: ± 20 % of the initial control points values	If constraint on minimum pitching moment is not satisfied, the penalty will be 1 drag count per 0.01 in ΔC_M
DPW-W1 wing	(a) Airfoils' maximum thickness constraint ≥ 13.5 %	
	(b) Airfoils' beam constraints:	
	Minimum thickness at 0.20c ≥ 12 %	
	Minimum thickness at 0.75c ≥ 5.9 %	
	(c) Limit: ± 20 % of the initial control points values	



RAE2822 airfoil geometry defined by a NURBS with 14 control points (CPs).

DPW wing geometry defined by three sections (airfoils), root, mid-span and tip. Each airfoil is addressed by a NURBS with 12 control points (36 CPs in total).

Fig. 1.3 NURBS parameterization of the RAE2822 airfoil (*left*) and DPW-W1 wing (*right*). RAE2822 airfoil geometry defined by an NURBS with 14 control points (CPs). DPW wing geometry defined by three sections (airfoils), root, mid-span and tip. Each airfoil is addressed by an NURBS with 12 control points (36 CPs in total)

1.4.2 Parameterization and Design Space Definition

The defined test cases are parameterized using volumetric NURBS which provides a high level of flexibility in order to discover optimal shapes. Authors have previously applied this parameterization technique to other local optimization problems [34, 35] and, in this paper, it is applied within a global optimization algorithm.

The RAE2822 airfoil and DPW-W1 wing test cases are parameterized with 14 and 36 design variables, respectively, which correspond to the z displacements of the volumetric NURBS control points, as can be observed in Fig. 1.3. The selected

Table 1.3 RAE2822 grids generated for sensitivity study

	Type	# Points	# Surface points	# Elements	# Surface elements
Coarse	Unstructured	56.700	388	68.758	192
Fine	Unstructured	185.364	772	245.618	384

Table 1.4 RAE2822 grid sensitivity results compared to experimental values

		DP1			DP2		
		$C_L = 0.8$, Mach $= 0.734$, Re $= 6.5 \times 10^6$			$C_L = 0.743$, Mach $= 0.754$, Re $= 6.2 \times 10^6$		
		C_L	C_D	C_M	C_L	C_D	C_M
Unstructured	Coarse grid	0.800	0.0188	−0.0953	0.743	0.0240	−0.1215
	Fine grid	0.800	0.0189	−0.0953	0.741	0.0239	−0.1210
Experiment		0.803	0.0168	−0.099	0.743	0.0242	−0.106
Constraint		0.800	–	−0.100	0.750	–	−0.110

parameterization, especially in the case of the wing, will help to draw conclusions on the feasibility of these methods when handling a high number of design variables.

1.4.3 *Grid Sensitivity Analysis*

In order to ensure the independence of the results from the computational grid employed, a grid sensitivity analysis has been performed on the baseline geometries.

RAE2822 Airfoil

For the RAE2822, two different computational grids were used in this sensitivity study. The characteristics of these grids can be observed in Table 1.3.

Table 1.4 shows the results of the CFD computation using these two grids and compared to experimental values [33]. As can be observed, there are not significant differences in the aerodynamic coefficients between the coarse and fine grid. In addition, the Cp plots are also very similar, as is displayed in Fig. 1.4. Therefore, the coarse grid was selected to be used for optimization purposes along this work.

DPW-W1 Wing

In the case of the DPW-W1, two different computational grids were used in this sensitivity study. The features of these grids can be observed in Table 1.5.

Table 1.6 shows the results of the CFD computation using these two grids and the Cp plots are displayed in Fig. 1.5. Although, in this case, some differences can be

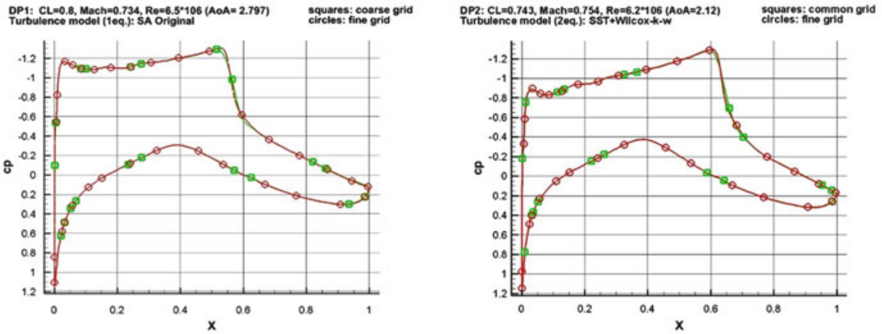


Fig. 1.4 RAE2822 Cp distribution coarse vs. fine unstructured grids

Table 1.5 DPW unstructured grids generated for sensitivity study

	Type	# Points	# Surface points	# Elements	# Surface elements
Coarse	Unstructured	566.534	135.046	2.388.975	269.768
Fine	Unstructured	3.660.283	539.859	18.005.856	1.079.072

Table 1.6 DPW grid sensitivity results

	DP1		
	Mach = 0.76, AoA = 0.5		
	C_L	C_D	C_M
Coarse grid	0.6077	0.0208	−0.2049
Fine grid	0.6165	0.0197	−0.2081

observed in the C_p plots between coarse and fine grids, the coarse one is considered to be enough adequate, in a balance of accuracy and size, for the optimization purpose.

1.4.4 Metamodel Obtention (SVMr)

The surrogate model based on SVMr is built following the approach displayed in Figs. 1.1 and 1.2. After this construction process, a tenfold cross-validation strategy has been applied to measure its prediction capabilities. The cross-validation method has been broadly used for estimating the prediction error of surrogate models. A k-fold cross-validation strategy splits the available data into k parts. For the kth part, the model is fitted to the other $k-1$ parts of the data and the prediction error of the fitted model is calculated when predicting the kth part. This process is repeated for $k = 1, 2, 3, \ldots k$ and the combination of those values gives the prediction error.

As the choice of the sampling fold decomposition is crucial for the error estimation, the validation procedure was repeated five times, each time using

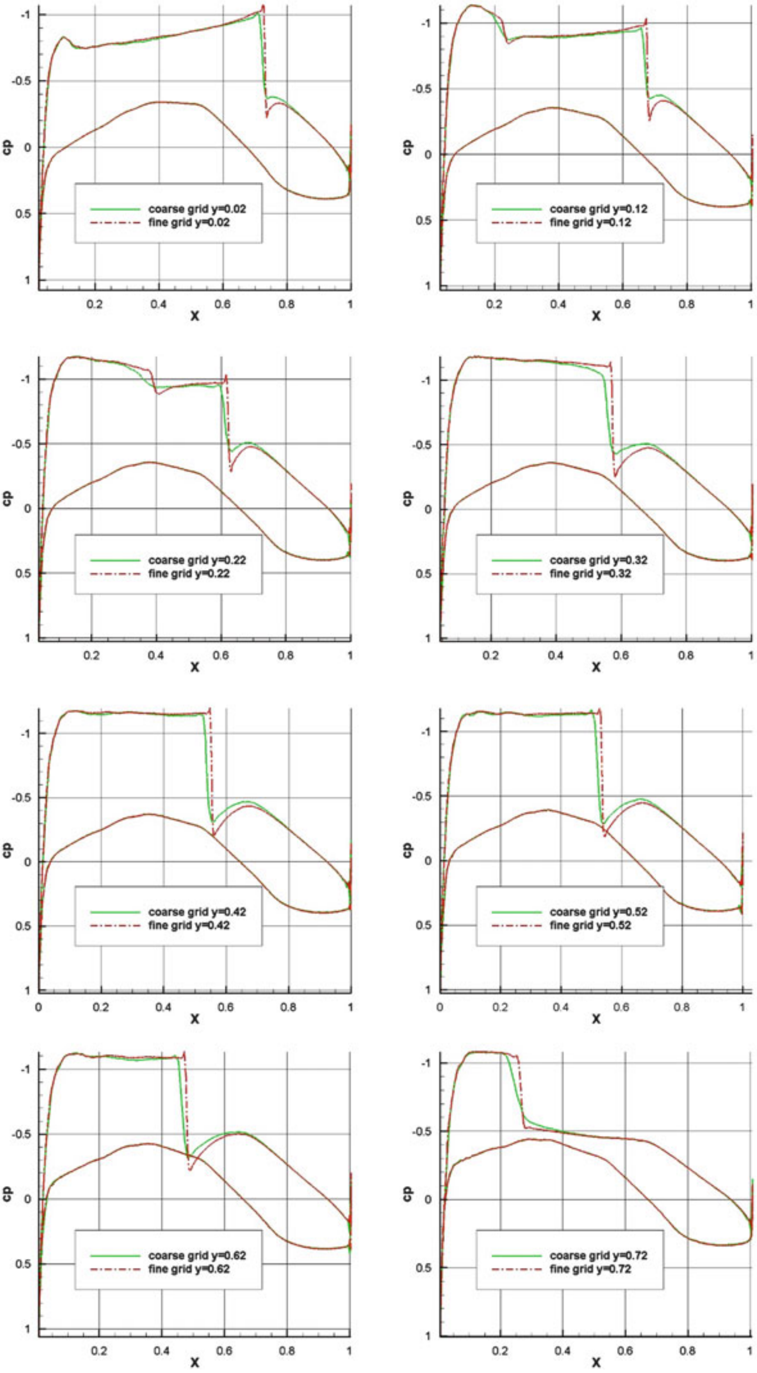


Fig. 1.5 DPW Cp distribution coarse vs. fine unstructured grids

different subset decompositions, in order to get a reliable error measure and make it independent on the choice of the training set decomposition.

The error metrics considered are described here:

(a) Mean squared error (Cross-Validation):

$$\boldsymbol{MSE}\left(\hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(f_i - \hat{f}^{-k(i)}(x_i)\right)^2 \tag{1.16}$$

The MSE metric gives an estimate of the expected test error.$\hat{f}^{-k}(X)$ denotes the function fitted with the surrogate and computed by removing the kth part of the training data. Smaller MSE values indicate smaller errors.

(b) Pearson Correlation coefficient

The Pearson's correlation coefficient ranges between -1 and 1 and provides the ratio between the covariance of f and $\hat{f}$ the product of their standard deviations. If ρ is close to zero, f and $\hat{f}$ are weakly correlated and, therefore, it can be expected that the prediction model $\hat{f}$ badly reproduces the variation of the function f. On the other hand, if the Pearson coefficient is close to 1, it means that there is a strong correlation between the two functions. Finally, if it is close to -1, anti-correlation exists and thus, a positive variation of f may produce a negative variation of $\hat{f}$.

$$\rho = \frac{N\sum_{i=1}^{N} f_i \hat{f}^{-k(i)}(x_i) - \sum_{i=1}^{N} f_i \sum_{i=1}^{N} \hat{f}^{-k(i)}(x_i)}{\sqrt{N\sum_{i=1}^{N} f_i^2 - \left(\sum_{i=1}^{N} f_i\right)^2}\sqrt{N\sum_{i=1}^{N}\left[\hat{f}^{-k(i)}(x_i)\right]^2 - \left[\sum_{i=1}^{N}\hat{f}^{-k(i)}(x_i)\right]^2}} \tag{1.17}$$

(c) G-metric for monocity

When the surrogate model is used within an optimization loop, another important measure is the capability to reproduce order relation between values of the objective function. In this sense, a surrogate model is monotonic when for any choice of x_i and x_j:

$$f(x_i) \leq f\left(x_j\right) => \hat{f}(x_i) \leq \hat{f}\left(x_j\right)$$

In order to measure this property in the dataset, the G metric is used:

$$\boldsymbol{G} = \sum_{i=1}^{N}\sum_{j=1}^{i} - \boldsymbol{min}\left[0, \frac{\hat{f}(x_i) - \hat{f}\left(x_j\right)}{f(x_i) - f\left(x_j\right)}\right] \tag{1.18}$$

The G metric is always positive, G = 0 means global monotonicity, while higher values indicate loss in monotonicity.

Table 1.7 Validation of the surrogate model based on SVMr for the RAE2822 test case

	MSE	Pearson	G-metric
Partition A	0.117564	0.555654	1971.121
Partition B	0.136239	0.490717	1528.915
Partition C	0.127884	0.535217	1144.694
Partition D	0.162249	0.416737	1689.266
Partition E	0.117948	0.552823	1495.125
Mean value	0.132377	0.51023	1565.824

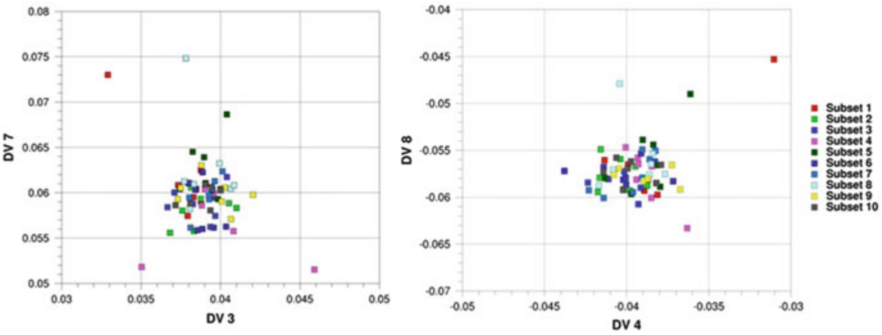


Fig. 1.6 RAE2822 dataset distribution with respect to design variables 3, 4, 7 and 8

The obtained error metrics when predicting the objective function in Eq. (1.14) for the case of the RAE2822 airfoil are displayed in Table 1.7.

The cross-validation analysis permits the study of the overall behaviour of the SVMr model and the searching method. If we analyse the error metrics, we can see that the Pearson correlation is between 0 and 1 and the MSE is not small compared with the optimization improvements we want to achieve. This is also the case of the R-squared value that was lower than 0.26 in all the partitions. Despite this we can see that the G-metric shows values smaller than other methods applied over this problem, which means that the global monotonicity is improved.

While the estimation of the SVM-r fitted by a small number of points can present low quality estimations about the value of the real objective function, the monotonicity (order relation between values) is best represented. Since the search method did not use the estimated value of the SVM-r, but only the location of the minimum value, a good representation of the function monotonicity implies a better performance of the optimization algorithm.

Figure 1.6 shows scatter plots of how the dataset is structured with respect to two of the design variables. A strong clustering can be observed in a specific region of the design space. This is due to the DoE technique used to generate the training database. As mentioned previously, the IES-SL focuses on optimization. Therefore the points added to the dataset are mostly in a specific region, where the predicted

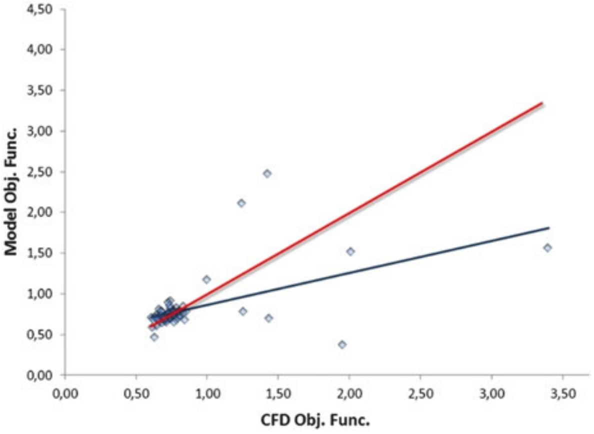


Fig. 1.7 Model vs. CFD correlation plot (RAE2822 dataset)

minima is located. As, in each iteration, the predicted minima is validated with a CFD computation, this clustered region is also the one where the real optima is located. In Fig. 1.7 we can see that the points with high objective function values are not very good fitted. This behaviour is produced by the dynamic sampling method. We are interested in an accurate description of low regions (for model adjustment), but we are only interested in a broad representation of high regions. When we applied the cross-validation method, there are not too much additional information for a good estimation of points with high objective function values, since the sampling method is more interesting in describing low areas.

This also implies that the algorithm is converging fast to the location of a minimum value. The balance between exploration and exploitation is a very interesting topic, since it represents the compromise between new model discovery (global search) and model adjustment (local search). We work with a reduced budget so the method must be capable to analyse the function, learning as soon as possible the main tendencies and try to find the best parameter combination in fewer steps than the maximum number of iterations allowed.

1.4.5 Multi-Point Optimization of the RAE2822 with Geometric Constraints

In this case, the approach is applied to the multi-point optimization of the RAE2822 airfoil. Table 1.8 shows the objective function and aerodynamic coefficients of the original and optimized geometries (note that the coefficients have been normalized by their initial values). The results show that the multi-point objective function has been improved by 42 %. It is important to notice how constraints on C_L and C_M have

Table 1.8 RAE2822 optimization results

Configuration	Objective function	DP1			DP2		
		C_L	C_D	C_M	C_L	C_D	C_M
RAE2822	1	0.800	0.0188	−0.0953	0.743	0.0240	−0.1215
Optimized	0.58	0.823	0.0128	−0.0911	0.775	0.0124	−0.1140

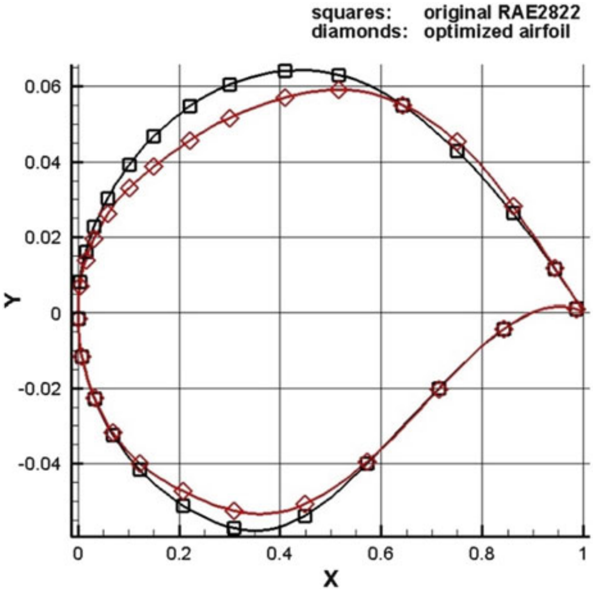


Fig. 1.8 Baseline RAE2822 and optimized geometries

been fulfilled. Figures 1.8, 1.9 and 1.10 show the shapes and pressure distributions of the original and optimized geometries for both design points.

1.4.6 *Multi-Point Optimization of the DPW-W1 with Geometric Constraints*

In this section, the approach is applied to the optimization phase of the DPW wing, as defined above. Table 1.9 shows the objective function of the original and optimized geometries. The results show that the objective function (in this case, minimize c-drag) has been improved only by 2 %. Figure 1.11 shows the comparison between the original and optimized shapes for different cuts along the wing span, and Fig. 1.12 shows the Mach number distribution, where it can be observed that the shock wave on the upper part of the wing has been slightly weakened.

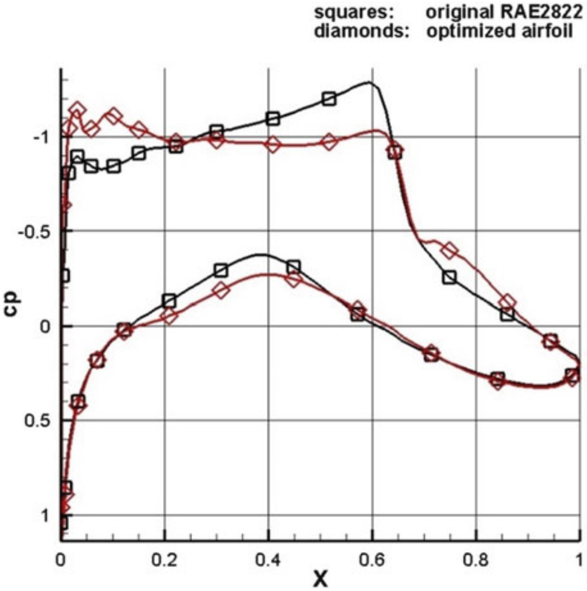


Fig. 1.9 RAE2822 airfoil vs. optimized airfoil pressure distribution at DP1

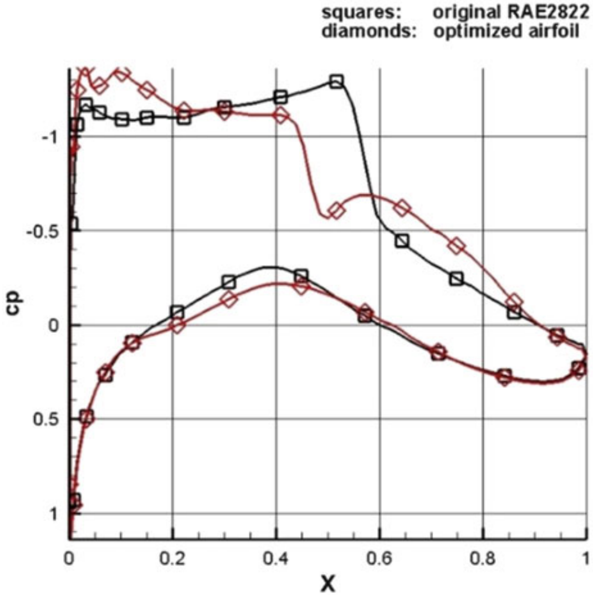


Fig. 1.10 RAE2822 airfoil vs. optimized airfoil pressure distribution at DP2

Table 1.9 DPW-W1 wing optimization results

Configuration	Objective function	DP1			DP2		
		C_L	C_D	C_M	C_L	C_D	C_M
DPW-W1	1	0.500	0.0140	−0.0202	0.500	0.0158	−0.0226
Optimized	0.98	0.505	0.0142	−0.0210	0.507	0.0154	−0.0232

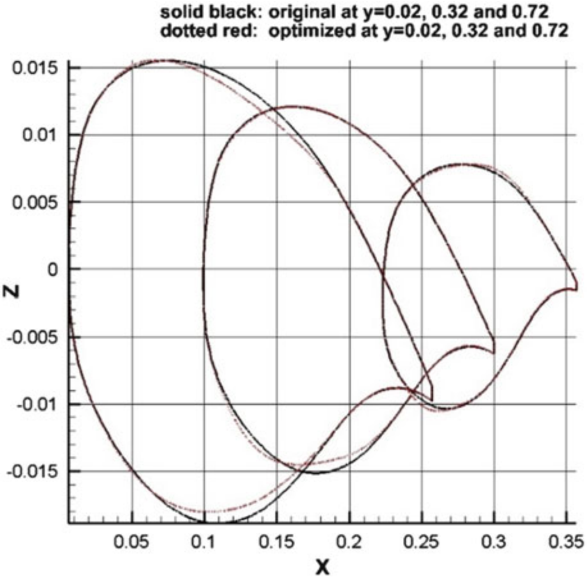


Fig. 1.11 Original vs. optimized wing shape

Conclusions

This paper presented a global optimization strategy using a novel surrogate model approach: The IES-SL. The hybridization of EA and SVMr has been tested on the optimization of a RAE2822 airfoil and a DPW wing, showing first promising results.

Further research will be performed on the accuracy of surrogate model when considering different budget (CFD computations) and taking into account other configurations. Future work will also try to improve the optimization results in the case of the DPW wing, by performing a detailed sensitivity analysis of the large number of design variables involved, and considering also gradient information during the optimization process.

In addition, we plan to apply the proposed methodology for the aerodynamic shape optimization of a cylinder, in transonic conditions, with the

(continued)

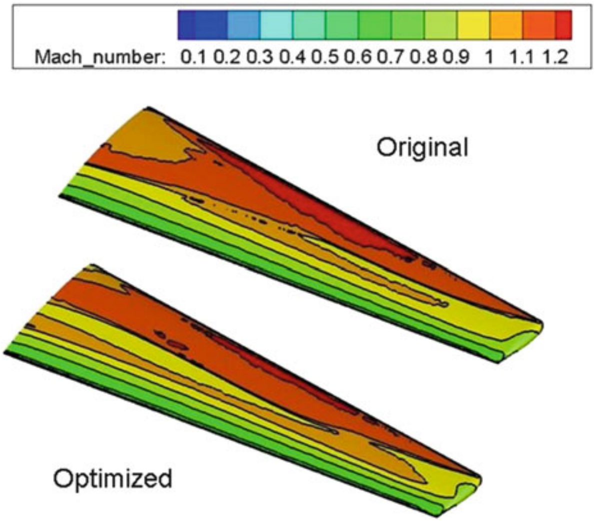


Fig. 1.12 Original vs. optimized Mach number distribution

aim to demonstrate the capability of this method to perform a real global optimization (where the initial configuration is very far from the optimum) and obtain airfoil-like geometries.

Acknowledgements The research described in this paper made by INTA, UAH and UPM researchers has been supported under the INTA activity "*Termofluidodinámica*" (IGB99001). This work is also partially supported by Spanish Ministry of Science and Innovation, under a project number ECO2010-22065-C03-02.

The experiments performed in this paper are also part of a GARTEUR action group (www.garteur.org) that has been established to explore these SBGO approaches. The main objective of the action group is, by means of a European collaborative research, to make a deep evaluation and assessment of SBGO methods for aerodynamic shape design, dealing with the main challenges as the curse of dimensionality, reduction of the design space and error metrics for validation, amongst others.

References

1. Jameson A, Martinelli L, Vassberg J (2002) Using computational fluid dynamics for aerodynamics—a critical assessment. ICAS paper 2002-1.10.1, Toronto
2. Leifsson L, Koziel S, Tesfahunegn Y (2014) Aerodynamic design optimization: physics-based surrogate approaches for airfoil and wing design. In: AIAA SciTech. AIAA 2014-0572
3. Li C, Brezillon J, Görtz S (2011) A framework for surrogate-based aerodynamic optimization. In: ONERA-DLR aerospace symposium, ODAS 2011
4. Koziel S, Leifsson L (2013) Multi-level surrogate-based airfoil shape optimization. In: 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, Grapevine (Dallas/Ft. Worth Region), 7–10 January 2013. AIAA2013-0778

5. Lukaczyk T, Palacios F, Alonso JJ (2014) Active subspaces for shape optimization. In: AIAA SciTech 2014
6. Iuliano E, Quagliarella D (2013) Proper orthogonal decomposition, surrogate modelling and evolutionary optimization in aerodynamic design. Comput Fluids 84(2013):327–350
7. Iuliano E, Quagliarella D (2013) Aerodynamic shape optimization via non-intrusive POD-based surrogate modelling. In: 2013 IEEE congress on evolutionary computation, Cancún, 20–23 June 2013
8. Jahangirian A, Shahrokhi A (2011) Aerodynamic shape optimization using efficient evolutionary algorithms and unstructured CFD solver. Comput Fluids 46:270–276
9. Zhong-Hua H, Zimmermann R, Görtz S (2010) A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data. In: American Institute of Aeronautics and Astronautics Conference
10. Weerdt E, Chu QP, Mulder JA (2005) Neural Network aerodynamic model identification for aerospace reconfiguration. In: AIAA 2005
11. Marinus BG, Rogery M, Braembusschez R (2010) Aeroacoustic and aerodynamic optimization of aircraft propeller blades. In: AIAA 2010
12. Praveen C, Duvigneau R (2007) Radial basis functions and Kriging Metamodels for aerodynamic optimization. INRIA Technical Report
13. Andres E, Salcedo-Sanz S, Monge F, Perez-Bellido AM (2012) Efficient aerodynamic design through evolutionary programming and support vector regression algorithms. Expert Syst Appl 39(2012):10700–10708
14. Clarke SM, Griebsch JH, Simpson TW (2005) Analysis of support vector regression for approximation of complex engineering analyses. Trans ASME J Mech Des 127(6):1077–1087
15. Smith C, Doherty J, Jin Y (2013) Recurrent neural network ensembles for convergence prediction in surrogate-assisted evolutionary optimisation. In: IEEE symposium series on computational intelligence, Singapore, 16–19 April 2013
16. Cheng CS, Chen PW, Huang KK (2011) Estimating the shift size in the process mean with support vector regression and neural networks. Expert Syst Appl 38:10624–10630
17. Salcedo-Sanz S, Ortiz-Garcia E, Perez-Bellido A, Portilla A (2011) Short term wind speed prediction based on evolutionary support vector regression algorithms. Expert Syst Appl 38:4052–4057
18. Zhou Z, Ong YS, Lim M, Lee B (2007) Memetic algorithms using multi-surrogates for computationally expensive optimisation problems. J Soft Comput 11(10):957–971
19. Kontoleontos EA, Asouti VG, Giannakoglou KC (2011) An asynchronous metamodel-assisted memetic algorithm for CFD-based shape optimisation. In: Engineering optimisation
20. GARTEUR AD/AG52 Members (2013) Partial report on surrogate-based global optimization methods in aerodynamic design, 2013–2015
21. Lepine J, Trepanier J (2000) Wing aerodynamic design using an optimized NURBS geometrical representation. In: 38th Aerospace science meeting and exibit, Reno, 10–13 January 2000
22. Bentamy A, Guibault F, Trepanier J (2005) Aerodynamic optimization of a realistic aircraft wing. AIAA 2005-332
23. Painchaud-Ouellet S, Tribes C, Trepanier JY, Pelletier D (2004) Airfoil shape optimization using NURBS representation under thickness constraint. In: 42nd AIAA aerospace sciences meeting and exhibit, AIAA 2004-1095
24. Mousavi A, Castonguay P, Nadarajah S (2007) Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization. In: 18th AIAA computational fluid dynamics, Miami, 25–28 June 2007
25. Piegl L, Tiller W (1997) The NURBS book. Springer, Berlin. ISBN 3-540-61545-8
26. Technical Documentation of the DLR TAU-Code (1994) Tech. rep., Institut of Aerodynamics and Flow Technology
27. Gerhold T, Galle M, Friedrichs O, Evans J (1997) Calculation of complex three-dimensional configurations employing the DLR TAU-Code. AIAA-97-0167. AIAA
28. Smola A, Schölkopf B, Müller KR (1998) The connection between regularization operators and support vector kernels. Neural Netw 11(4):637–649

29. Quiancheng W, Shunong Z, Rui K (2011) Research of small samples avionics prognosis based on support vector machines. In: Prognosis & system health management conference (PHM2011 Shenzhen)
30. Ortiz-García EG, Salcedo Sanz S, Pérez-Bellido ÁM, Portilla-Figueras JA (2009) Improving the training time of support vector regression algorithms through novel hyper-parameters search space reductions. Neurocomputing 72:3683–3691
31. Smith J (2007) On replacement strategies in steady state evolutionary algorithms. Evol Comput 15(1):29–59
32. Cook PH, McDonald MA et al (1979) Aerofoil RAE 2822—pressure distributions, and boundary layer and wake measurements, experimental data base for computer program assessment. AGARD Report AR 138
33. Vassberg J, Tinoco EN, Mani M et al (2007) Summary of the third AIAA CFD drag prediction workshop. AIAA Paper 2007-0260
34. Martin MJ, Andres E, Valero E, Lozano C (2013) Gradients calculation for arbitrary parameterizations via volumetric NURBS: the control box approach. In: EUCASS conference
35. Martin MJ, Andres E, Valero E, Lozano C (2014) Volumetric B-Splines shape parameterization for aerodynamic shape design. Int J Aerosp Sci Technol 2014:26–36. doi:10.1016/j.ast.2014.05.003

Chapter 2
Adaptive Sampling Strategies for Surrogate-Based Aerodynamic Optimization

Emiliano Iuliano

Abstract The chapter proposes the application of surrogate-based optimization to the efficient design of aeronautical configurations. The surrogate model consists of the Proper Orthogonal Decomposition of computed aerodynamic flow fields and Radial Basis Functions interpolation to reconstruct the aerodynamic flow at any unknown design vector. The surrogate model is coupled to an evolutionary algorithm to globally explore the design space. Several adaptive sampling strategies are proposed, either objective-driven (i.e. aimed at improving the fitness function) or error-driven (i.e. aimed at reducing the prediction error of the surrogate model globally). The proposed methodology is applied to the design optimization of a two-dimensional airfoil in multi-point transonic conditions. The results of different training strategies are critically discussed and compared.

2.1 Introduction

Modern air vehicle design has been increasingly driven by environmental as well as operational constraints. Environmental concerns, including emissions and noise, are gaining increasing importance in the design and operations of commercial aircraft. Taking into account the current prognoses for the growth in air traffic, the above-mentioned challenges become even more significant [1]. In this context, 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. Therefore, highly accurate analysis methods have been continuously introduced both in geometric representation and physical modeling, but the main drawback is that

E. Iuliano (✉)
CIRA, The Italian Aerospace Research Center, Via Maiorise, 81043 Capua, Italy
e-mail: e.iuliano@cira.it

E. Iuliano, E.A. Pérez (eds.), *Application of Surrogate-based Global Optimization to Aerodynamic Design*, Springer Tracts in Mechanical Engineering,
DOI 10.1007/978-3-319-21506-8_2

they are computationally expensive. For example, the solution of the Navier–Stokes equations around complex aerodynamic configurations requires a huge amount of computational resources even on modern state-of-the-art computing platforms. This turns out to be an even bigger issue 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. 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. nonlinearities in the physical model, constraints handling, curse of dimensionality, multi-modal fitness landscape, accuracy *vs* 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.

Finding the set of parameters which best fit the model to the available data is usually known as the training phase. The training dataset is usually obtained by sampling the design space (Design and Analysis of Computer Experiments, DACE) and performing expensive high-fidelity computations on the selected points. Depending on the adopted surrogate technique, design objectives and constraints or vector/scalar fields of interest are used to feed the surrogate model. The strategy to properly and optimally choose the DACE sampling dataset is of paramount importance to achieve a satisfactory accuracy of the surrogate model. Unfortunately, classical sampling methods, like Latin Hypercube sampling, are very sensible to the nature of the problem at hand and they may deceive the surrogate-based optimization by hiding or masking the true optima locations. This is especially true in aerodynamic shape design problems where both model nonlinearities and the dimension of the search space combine to emphasize this issue: classical DACE techniques would lead to intensively sample the search space, thus vanishing the actual advantage of surrogate-based optimization.

Indeed, the training strategy is heavily dependent on the type and scope of the surrogate model and should be tailored on it. Generally speaking, two different needs have to be taken into account when searching for “optimal” training points: exploration and “trust” or exploitation. The first aims at unveiling promising regions of the design space where global/local minima might reside; the second consists in giving the surrogate model confidence on the prediction of minima locations and, as a result, sampling in the surroundings. For the sake of clarity, an evenlyspaced dense sampling could be classified as purely exploratory, while a properly clustered sampling could fall in the exploitative class. As a consequence, a trade-off exist between the exploration of the design solutions and the trust in surrogate accuracy near predicted minima. A proper balance between these two concepts allows to combine the need to exploit the approximation surface (by sampling where it is minimized) with the need to improve the approximation (by sampling far from the known points, i.e. where prediction error may be high). With the term “in-fill criteria” it is usually meant some principles which allow to optimally place new points (in-fill points) at which the true target function should be called. The selection of in-fill points is referred to as adaptive sampling or model updating.

In order to reduce the computational effort in training accurate surrogate models for aerodynamic shape design problems, this paper proposes adaptive sampling strategies which use ad hoc in-fill criteria to drive the training process. The adaptive criteria are formulated by explicitly taking into account the optimization target 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, several choices of the in-fill criteria will be 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. The performance of each adaptive sampling criterion will be monitored during the in-fill process by means of cross validation techniques and proper generalization error metrics will be adopted. Moreover, an aerodynamic optimization case study will be proposed to test different combinations of surrogate models and adaptive sampling methods once fixed the computational budget in terms of number of high-fidelity simulations (i.e. CFD analyses). This will allow 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.

2.2 Literature Review

Jones et al. [10], 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 due 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 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. [12], 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. [11] and Simpson et al. [13]. 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 infill points, also referred to as adaptive sampling or model updating, represents the core of a surrogate-based optimization method and helps to improve the surrogate prediction in promising areas of the objective space.

Several recent studies have been focused on two main aspects: the right choice of the number of points for the initial sampling and the ratio between initial/in-fill samples. 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 Sbester [14], 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 dataset. 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 are 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 given in-fill criterion, the mathematical machinery of the surrogate should exhibit the capability to adapt to unexpected, local nonlinear behaviour 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-filling when some generalization error metrics, e.g. cross-validation, fall 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.

2.3 Surrogate Model

The singular value decomposition (SVD) solution of the POD basis vectors and coefficients for steady-state problems is described in [6–9]. 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.

Given the three spatial components (ξ, υ, ζ) of the computational mesh points and the general snapshot vector **s**, let $\{\mathbf{w}_j\}$ be a set of design vectors (e.g. sampled from the design space with a DoE technique) and $\{\mathbf{s}_j\}$ the corresponding snapshot, i.e. column vectors containing the volume grid and flow variables as obtained from a CFD solution:

$$\begin{aligned}
\mathbf{s} &= (\mathbf{s}_{\text{grid}}, \mathbf{s}_{\text{flow}})^T \\
\mathbf{s}_{\text{grid}} &= (\xi_1, \ldots, \xi_q, \upsilon_1, \ldots, \upsilon_q, \zeta_1, \ldots, \zeta_q) \\
\mathbf{s}_{\text{flow}} &= (\rho_1, \ldots, \rho_q, \rho\xi'_1, \ldots, \rho\xi'_q, \rho\upsilon'_1, \ldots, \rho\upsilon'_q, \rho\zeta'_1, \ldots, \rho\zeta'_q, p_1, \ldots, p_q)
\end{aligned}$$

where q is the number of mesh nodes involved in the POD computation, (ξ, υ, ζ) are the mesh nodes coordinates in a Cartesian reference system, ρ is the flow density, $(\xi', \upsilon', \zeta')$ are the three Cartesian velocity components and p is the static pressure. The computational mesh has been included in the POD snapshot to let the SVD basis catch the coupling effects between space location and state field. Hence, once the surrogate model is built, not only a flow field can be computed, but also an approximation of the volume mesh. Such a surrogate model would be able to catch, although in a reduced order form, the cross effects of geometry modification and aerodynamic flow change. As the total number of variables is eight (three mesh variables and five flow variables), the global size of the snapshot is $N = 8 \times q$.

2.3.1 SVD Solution

Starting from the vectors $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_M$ obtained by CFD expensive computations for a representative set of design sites $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M$, finding a POD 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 SVD of **P**

$$\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}\begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_M \\ 0 & \cdots & 0 \end{pmatrix}\mathbf{V}^T \tag{2.1}$$

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

$$\alpha_i(\mathbf{w}_j) = (\mathbf{s}_j - \bar{\mathbf{s}}, \boldsymbol{\phi}_i) \tag{2.2}$$

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 small 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 means 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

$$\mathbf{s}_j \simeq \bar{\mathbf{s}} + \sum_{i=1}^{\hat{M}} \alpha_i(\mathbf{w}_j)\boldsymbol{\phi}_i \tag{2.3}$$

where

$$\hat{M} \leq M \implies \frac{\sum_{i=1}^{\hat{M}} \sigma_i^2}{\sum_{i=1}^{M} \sigma_i^2} \geq \epsilon \tag{2.4}$$

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

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

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.

2.3.2 Pseudo-Continuous Global Representation

Equation (2.3) 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 as 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 [2, 5] that interpolation errors become high for very small and very large values of 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 (Reference [2] 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:

$$\mathbf{s}(\mathbf{w}) = \bar{\mathbf{s}} + \sum_{i=1}^{\hat{M}} \alpha_i(\mathbf{w})\boldsymbol{\phi}_i \tag{2.5}$$

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.

2.4 In-Fill Criteria

The strategy for training a surrogate model is heavily dependent on its type and scope 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 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).

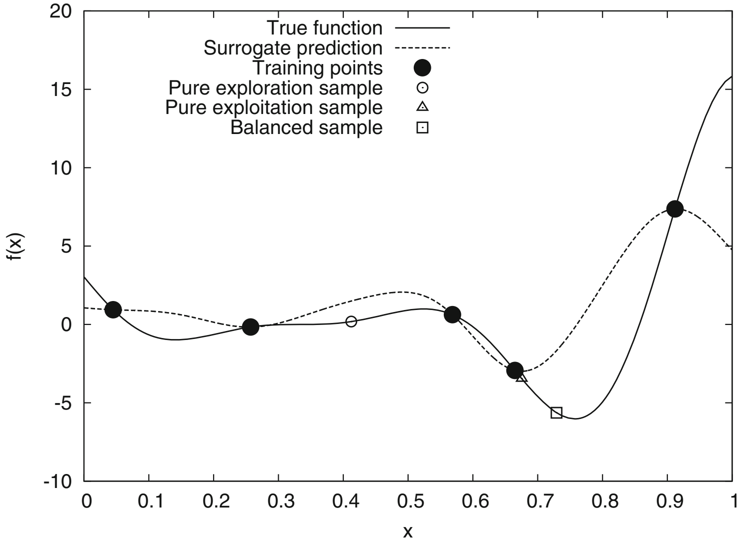


Fig. 2.1 Exploitation vs exploration, 1D example

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 behaviours 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 2.1 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 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.

Here, we are interested in designing balanced in-fill criteria for a generic surrogate model. They are formulated in terms of auxiliary functions which have to be maximized. In order to realize a trade-off between exploration and exploitation, the general form of the auxiliary function is factorized as follows:

$$v(\mathbf{x},\hat{f}(\mathbf{x}),X_n,F_{X_n}) = g(\mathbf{x},X_n)h(\hat{f}(\mathbf{x}),F_{X_n}) \tag{2.6}$$

where $\mathbf{x}$ is the generic design vector, $\hat{f}(\mathbf{x})$ is the surrogate prediction at $\mathbf{x}$,X_n is the set of n available training points and F_{X_n} is the corresponding set of true objective function values:

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

The v function is called potential of improvement function. The functions g and h measure the exploration and model trust contributions. In particular, the exploration function depends 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:

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

Different in-fill criteria are selected by properly designing the functions g and h. In order to update the training dataset, a new sample is selected by maximizing the auxiliary function:

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

Hereinafter the maximization of the auxiliary function is achieved in the following way: a huge Latin Hypercube Sampling dataset (e.g. 500 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.

2.4.1 Error-Driven In-Fill Criteria

Two error-driven in-fill criteria are presented in this section. Both are aimed at sampling where the leave-one-out (LOO) prediction error is highest. The first strategy, named “LOO” hereafter, adopts the following functions:

$$g(\mathbf{x}, X_n) = \min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i) \tag{2.7}$$

$$h(\hat{f}(\mathbf{x}), F_{X_n}) = |\hat{f}^{-k}(\mathbf{x}_k) - f(\mathbf{x}_k)| \tag{2.8}$$

where, given $\mathbf{x}$, $\mathbf{x}_k = \operatorname{argmin}_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)$ is its nearest neighbour training point and $\hat{f}^{-k}(\mathbf{x}_k)$ is the LOO surrogate prediction at $\mathbf{x}_k$. In other words, the exploration term is given by the distance from the nearest training point, while the exploitation function is based on the absolute value of the LOO prediction error: hence, this approach will tend to sample where there is big uncertainty in the prediction and to improve the surrogate model where it lacks accuracy. However, no information is introduced about the objective function minimization, thus no significant improvement of the global optimum search can be expected from its application.

The second strategy, named "LOOW" hereafter, is similar to the former, but a weighting function is introduced in the exploitation term to partially take into account the behaviour of the objective function. Indeed, the exploration function is the same, while the exploitation one is designed as follows:

$$h(\hat{f}(\mathbf{x}), F_{X_n}) = \exp\Big(-\sigma \frac{\hat{f}(\mathbf{x}) - f_{min}}{f_{max} - f_{min}}\Big)|\hat{f}^{-k}(\mathbf{x}_k) - f(\mathbf{x}_k)| \tag{2.9}$$

where σ is a tuning parameter and f_{min},f_{max} are the minimum and maximum objective function values in the training set. The parameter σ can be varied to adjust the contribution of the exploitation function: for low values of σ, the surrogate prediction is considered unreliable, hence promising as well as poor regions are equally treated; for high values of σ, a transition to a more exploitative approach is done, as the regions exhibiting poor values of the objective function will be filtered out.

Figure 2.2a shows the potential of improvement functions obtained by applying the aforementioned LOO-based criteria to the example reported in Fig. 2.1. In particular, concerning the LOOW strategy, three different values of the σ parameter have been selected to compare the function behaviour. It is clearly observable that the introduction of the weighting function strongly dampens the potential levels, especially in those regions where the explorative behaviour is active. For $\sigma = 10$, the maximum location of the potential of improvement moves to a completely different region which is closer to the true optimum.

2.4.2 *Objective-Driven Criteria*

An objective-drive in-fill criterion is presented here. It 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. 2.2. The present approach, named " EI-like" hereinafter, represents a generalization of that method as, 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:

$$v(\mathbf{x},\hat{f}(\mathbf{x}), X_n, F_{X_n}) = (f_{min} - \hat{f}(\mathbf{x}))\Phi\Big(\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\Big) + \hat{s}(\mathbf{x})\phi\Big(\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})}\Big) \tag{2.10}$$

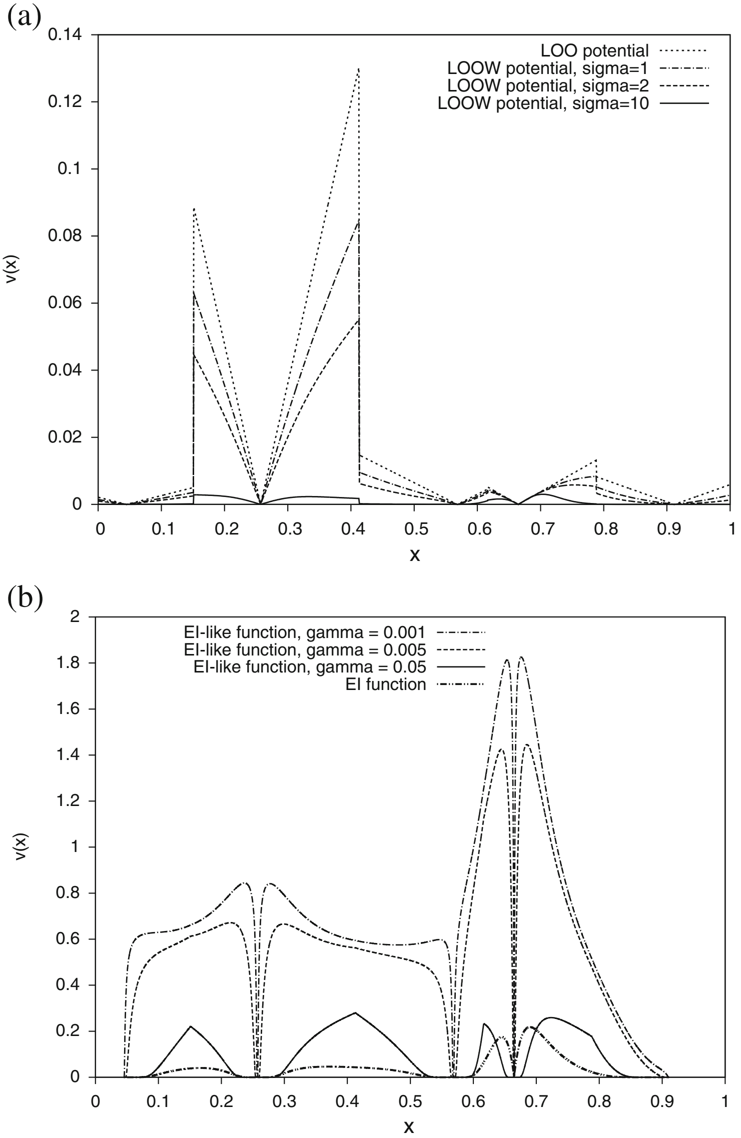


Fig. 2.2 Potential of improvement, 1D example. (**a**) Leave-one-out-based criterion. (**b**) Expected Improvement criterion

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:

$$\hat{s}(\mathbf{x}) = \frac{1}{2}|f_{max} - f_{min}| \exp\Big(-\gamma \frac{\max_{\mathbf{x}_i,\mathbf{x}_j \in X_n} d(\mathbf{x}_i, \mathbf{x}_j)}{\min_{\mathbf{x}_i \in X_n} d(\mathbf{x}, \mathbf{x}_i)}\Big) \tag{2.11}$$

where γ is a tuning parameter. This 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 2.2b shows the v function for various values of the γ parameter. As in the previous case, the γ parameter strongly alter the potential of improvement profile in terms of both global levels and position of the maximum point. Indeed, for $\gamma = 0.001, 0.005$, the peak of the potential of improvement is around $x \simeq 0.7$, while for $\gamma = 0.05$ it moves to $x \simeq 0.4$.

2.5 Surrogate-Based Shape Optimization Approach

The workflow of the surrogate-based shape optimization (SBSO) is depicted in Fig. 2.3. 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 NURBS parameterization module transforms the design vectors into geometrical shapes,

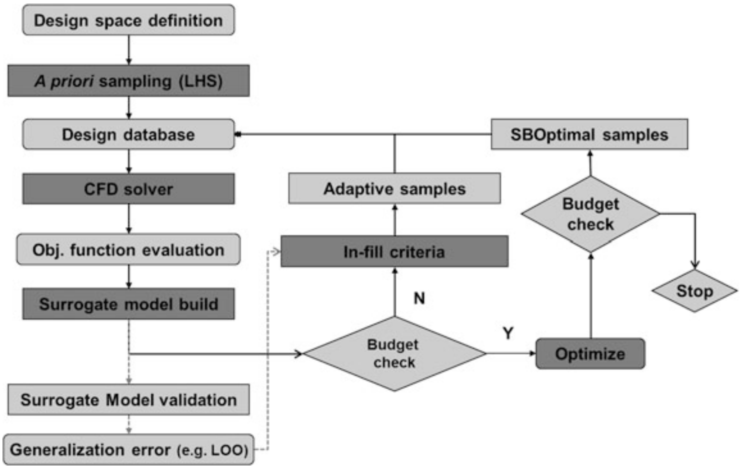


Fig. 2.3 Workflow of surrogate-based evolutionary optimization

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 [1]. Once the converged flow field variables are available, the snapshots collection is carried out according to Sect. 2.3 and the POD/RBF surrogate model is built as described in Sects. 2.3.1 and 2.3.2. After that, the workflow in Fig. 2.3 shows two internal cycles, namely the 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. 2.4 and is aimed at improving the surrogate model before 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. The second cycle (database updating by optimization) allows for including optimal design sites $\mathbf{x}_{opt}$, provided by the surrogate-based optimization, into the POD ensemble database: this phase should lead to the final exploitation of 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 [15].

2.6 Application: Multi-Point Shape Optimization of a Two-Dimensional Airfoil

The proposed methodology is applied to the shape optimization problem of a two-dimensional airfoil at two design points in transonic conditions. The baseline shape is the well-known RAE 2822 airfoil [3].

2.6.1 Problem Definition

The RAE2822 airfoil is parameterized by a volumetric three-dimensional NURBS control-box, as described in Fig. 2.4. Even if the control box consists of 200 control points, only 14 are used to modify the geometry. The design points are summarized in Table 2.1 and correspond to Case 9 and 10 of the experimental database [3]. Figure 2.5a,b show the pressure coefficient distribution as computed with the ZEN flow solver on the baseline airfoil at both design points. Two different turbulence models have been used, namely $k-\omega$ TNT and SST, to assess their effects on the aerodynamic flow and the CFD results have been also compared to the available experimental data. Design point 2 is particularly difficult to handle as a shock-

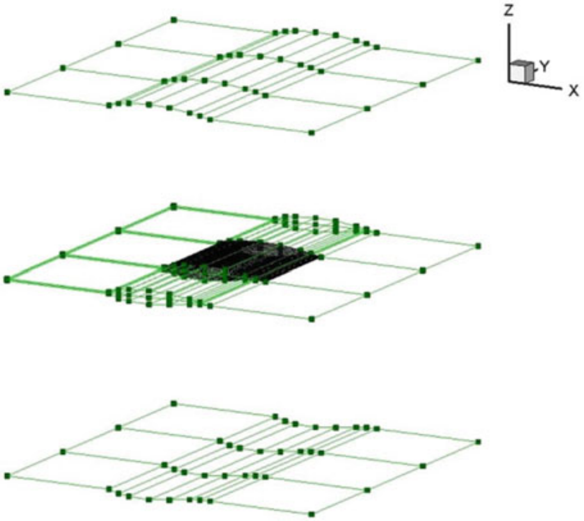


Fig. 2.4 NURBS control box

Table 2.1 Design points

DP	Mach number	Reynolds number	Angle of attack (°)
1	0.734	6.5e+06	2.65
2	0.754	6.2e+06	2.9

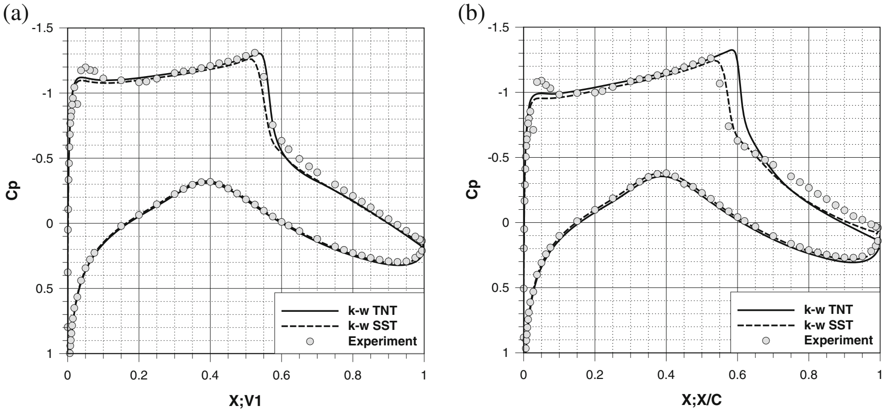


Fig. 2.5 Pressure coefficient distribution of the RAE2822 baseline airfoil. (**a**) Design point 1. (**b**) Design point 2

Table 2.2 Aerodynamic coefficients of baseline airfoil

DP	$C_{l,0}$	$C_{d,0}$	$C_{m,0}$
1	0.805	0.0192	−0.1
2	0.743	0.0280	−0.11

induced separation is found on the suction side which significantly increases the aerodynamic drag and represents a major challenge of the proposed design problem. The objective function to be minimized is specified as follows:

$$f(\mathbf{x}) = \frac{1}{2}\Big(\frac{C_d + C_{d,m} + C_{d,l}}{C_l}\frac{C_{l,0}}{C_{d,0}}\Big)_{DP1} + \frac{1}{2}\Big(\frac{C_d + C_{d,m} + C_{d,l}}{C_l}\frac{C_{l,0}}{C_{d,0}}\Big)_{DP2}$$

where C_l, C_d, C_m are the lift, drag and pitching moment coefficient of the airfoil under evaluation and $C_{l,0}, C_{d,0}, C_{m,0}$ are the same coefficient values of the baseline airfoil. Table 2.2 reports the aerodynamic coefficients of the baseline airfoil. As regards aerodynamic constraints, both lift and moment coefficients are required to be greater or equal to the baseline values, i.e. $C_l \geq C_{l,0}$ and $C_m \geq C_{m,0}$. Hence, two drag penalization terms $C_{d,l}, C_{d,m}$ are introduced and designed as follows:

$$C_{d,m} = 0.01 \max(0, C_{m,0} - C_m)$$
$$C_{d,l} = 0.1 \max(0, C_{l,0}^2 - C_l^2)$$

The order of magnitude of the penalization terms is such that the drag coefficient is penalized by one drag count per one hundredth moment violation and sixteen drag counts for one tenth lift violation. Geometric constraints are also introduced to set the airfoil maximum thickness-to-chord ratio $\frac{t}{c}$ at 12.1 %, to keep the airfoil thickness ratio at 80 % chord greater than 4 % and the leading edge radius greater than 0.004 m. The geometric constraints are implicitly satisfied within the parameterization.

2.6.2 Optimization Setup

Five different surrogate-base optimization runs have been launched with different sampling strategies and two surrogate models, namely the POD/RBF model and a Kriging model. While the latter can be built directly on the objective function database computed on the training set, the former presents some additional issues: indeed, dealing with a problem with multiple design points, as the POD/RBF model has to be fed with the aerodynamic flow solution, two independent POD/RBF models have to be trained, one for each design point. This further complicates the optimization case and makes it a solid and robust test of the whole methodology.

Table 2.3 Setup of SBO runs

DB ID	Model	In-fill	Budget	A priori	Adaptive	Opt. updating
DB-LOO	POD/RBF	LOO	100	18	52	30
DB-LOOW	POD/RBF	LOOW	100	18	52	30
DB-EIPOD	POD/RBF	EI-like	100	18	82	–
DB-EIKRIG	Kriging	EI-like	100	18	82	–
DB-EGO	Kriging	EI	100	18	82	–

The computational budget has been fixed in order to fairly compare optimization results obtained with different methods and consists of 100 CFD computations: this represents a rather low threshold, especially considering that the dimension of the design space is 14 and that classical evolutionary algorithm may reach the optimal solution after a number of CFD evaluations at least one order of magnitude higher than this. The computational load has to be split into *a priori* sampling, adaptive sampling and optimization updating. Table 2.3 reports the basic information for each optimization run. DB-LOO and DB-LOOW optimizations have been obtained by iterating the corresponding error-driven in-fill criteria for nearly a half of the total computational budget and refining the search by applying the optimization updating for 30 more times. DB-EIPOD database has been obtained by employing the EI-like criterion 82 times and with no optimization updating as this is an objective-driven approach. The last two simulations, namely DB-EIKRIG and DB-EGO, have been launched for assessment purposes as they both adopt a Kriging model instead of the POD/RBF surrogate. However, they differ for the in-fill criterion used, the EI-like in the first case, the classical Expected Improvement maximization (i.e. exploiting the original Kriging prediction and variance) in the second case.

2.6.3 Surrogate Model Validation

Each combination of surrogate model/in-fill strategy has been validated after 52 samples addition (i.e. surrogate models built with 70 total samples) in order to estimate their prediction error and make some preliminary analyses. In particular, in order to re-use the same samples, tenfold cross-validation has been used and five repetitions of the database partition have been performed to get averaged values of the error metrics. These are the root mean squared error (RMSE) and the Pearson correlation coefficient. Table 2.4 summarizes the results of the validation analysis. It can be observed how the weighting of the LOO error has been effective in providing a better model. EI-like criterion results are in line with LOO ones, while Kriging-based models exhibit superior performances, especially in terms of correlation coefficient. However, these indications cannot be considered as definitive for two reasons: for DB-LOO and DB-LOOW runs, the optimization phase has not yet started; for EI-based runs, the different combination of surrogate model/in-fill

Table 2.4 Validation results

DB ID	RMSE	Pearson correlation
DB-LOO	0.542	0.609
DB-LOOW	0.320	0.680
DB-EIPOD	0.406	0.589
DB-EIKRIG	0.255	0.817
DB-EGO	0.146	0.837

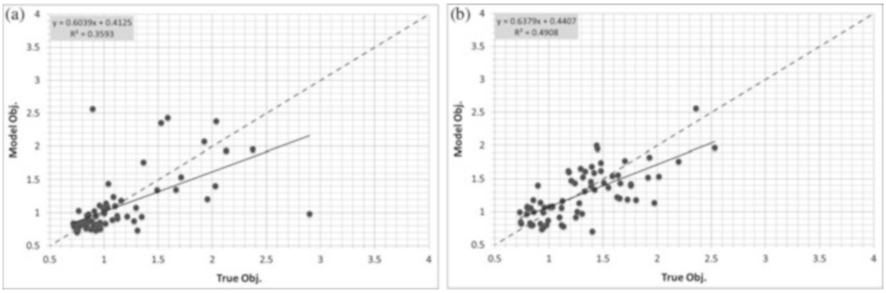


Fig. 2.6 Correlation plots. (**a**) DB-EIPOD. (**b**) DB-LOOW

criterion may alter the speed of convergence of the method. Figure 2.6a,b show two correlation plots where the cross-validated model values of the objective function are compared to the true value (i.e. obtained with the CFD solver). Results for DB-EIPOD and DB-LOOW cases are reported. A significant clustering of sub-optimal candidates is observed for DB-EIPOD run, while the distribution of points is more scattered for DB-LOOW. This reflects the substantial difference of the two approaches in updating the design database. Moreover, a closer look to DB-EIPOD results points out that the model prediction is very good when the objective function is around one or less, while it is significantly worsened for higher values: this clearly means that the EIPOD model is well approximating the objective function in interesting regions of the design space, where the solution is not constrained, while, due to the balanced nature of the in-fill criterion, highly constrained zones are still being explored but small effort is done by the method to locally increase the model accuracy, hence the prediction error looks quite large. This behaviour penalizes the evaluation of the error metrics for the DB-EIPOD case.

2.6.4 *Optimization Results*

Figures 2.7 and 2.8 show the objective function history of selected surrogate-based optimizations: the true objective function values corresponding to the design space samples are reported from the initial LHS samples (ID 1 to ID 18) to the completion of the full training set (up to ID 100), according to the split reported in Table 2.3. A clear trend to improve the objective function is observable for objective-driven

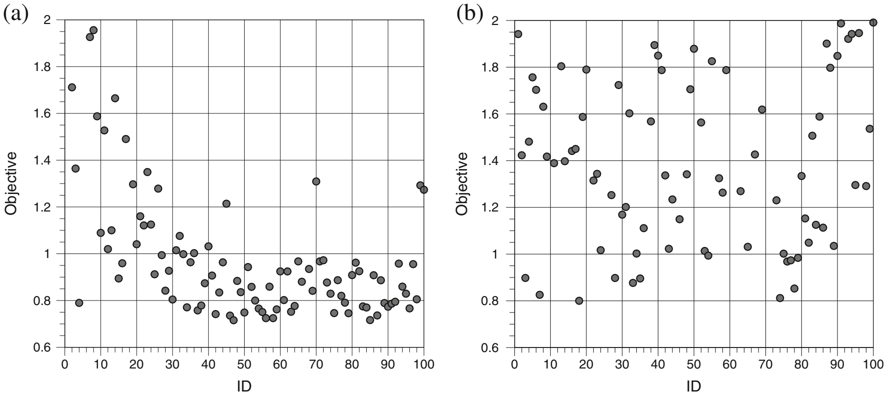


Fig. 2.7 Objective function history of POD-based optimization runs. (**a**) DB-EIPOD. (**b**) DB-LOO

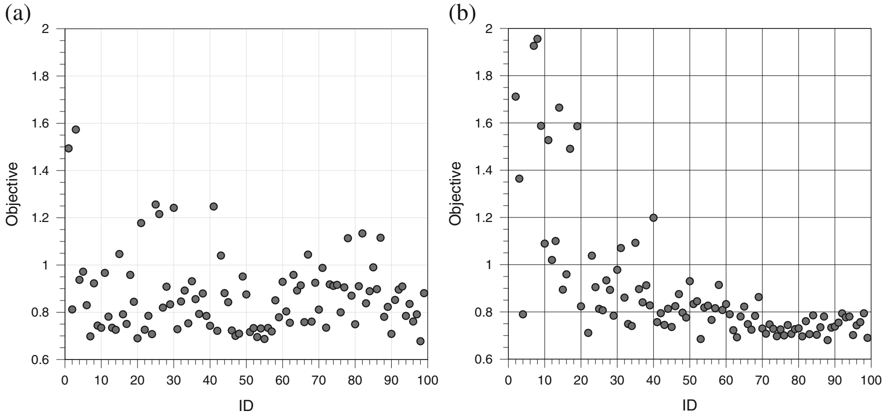


Fig. 2.8 Objective function history of Kriging-based optimization runs. (**a**) DB-EGO. (**b**) DB-EIKRIG

methods (DB-EIPOD, DB-EIKRIG and DB-EGO), while the DB-LOO case appears much more explorative. For each database, the best solution is extracted after the total computational budget has been reached. Table 2.5 summarizes the aerodynamic coefficients and the objective function values for each optimal candidates. For the sake of clarity, the performances of the baseline RAE2822 airfoil are also reported. Significant drag reduction and lift increase are observed at both design points. The objective-driven methods significantly outperform the error-driven and Kriging-based methods achieve the best performances, especially on design point 2 where the drag reduction is more significant. The aerodynamic flow for the selected optimal candidates is compared in Figs. 2.9a,b and 2.10a,b. For each airfoil, the pressure distribution at both design points are depicted. All the objective-driven design solutions look very similar, DB-EGO solution is evidently preferable as the

Table 2.5 Aerodynamic coefficients of baseline and optimal candidates

ID	$C_{l,DP1}$	$C_{d,DP1}$	$C_{m,DP1}$	$C_{l,DP2}$	$C_{d,DP2}$	$C_{m,DP2}$	Obj. function
RAE2822	0.805	0.0192	−0.1	0.743	0.0280	−0.11	1.00
DB-LOO	0.833	0.0151	−0.105	0.761	0.0241	−0.103	0.800
DB-LOOW	0.806	0.0134	−0.093	0.777	0.0226	−0.096	0.732
DB-EIPOD	0.827	0.0135	−0.106	0.796	0.0223	−0.108	0.716
DB-EIKRIG	0.817	0.0127	−0.107	0.790	0.0209	−0.109	0.681
DB-EGO	0.826	0.0128	−0.103	0.796	0.0211	−0.104	0.677

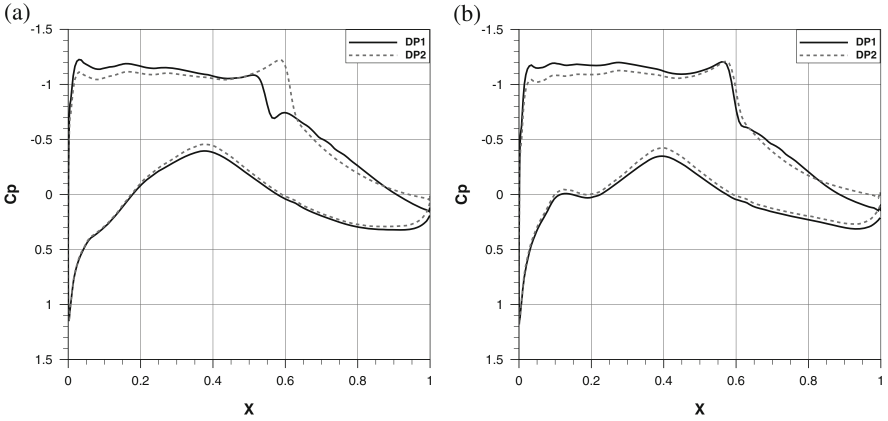


Fig. 2.9 Pressure coefficient distribution of POD-based optimal candidates. (**a**) DB-EIPOD. (**b**) DB-LOO

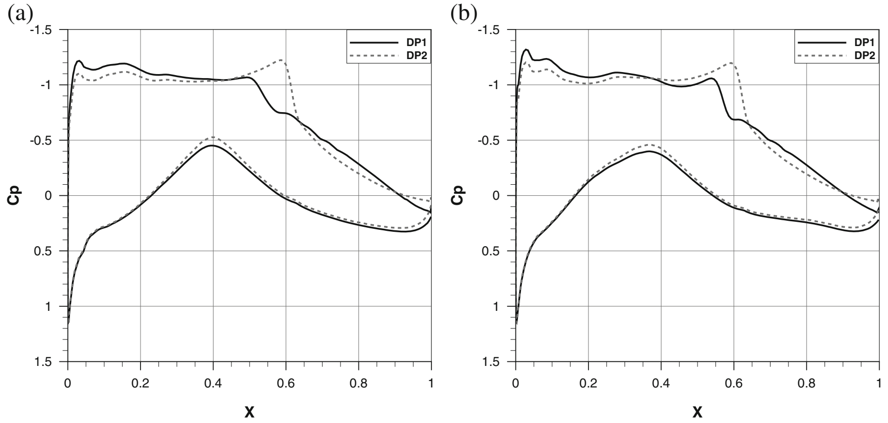


Fig. 2.10 Pressure coefficient distribution of Kriging-based optimal candidates. (**a**) DB-EGO. (**b**) DB-EIKRIG

shock wave is slightly smoother. However, results show that the adoption of an EI-like in-fill criterion coupled with a POD/RBF surrogate model and an evolutionary algorithm yields noticeable results for the proposed shape design problem and compares fairly well with similar well-known and widely used methods like EGO algorithm.

Conclusions

The paper proposed objective-driven (EI-like) and error-driven adaptive sampling techniques suitable for generic surrogate models. The related in-fill criteria have been designed in order to improve an auxiliary function that can be linked to a certain metric of the model improvement. Moreover, multi-point POD/RBF models have been developed and integrated within an evolutionary optimization system, so that a full chain of surrogate-based optimization with intelligent training of the surrogate model is available. Results showed that objective-driven sampling strategies over-performed the error-driven in terms of search efficiency at fixed computational budget. Thanks to the intelligent in-fill, optimal solutions of the two-dimensional airfoil shape design problem have been found by different surrogate-based methods by calling the CFD solver only 100 times. This represented a big challenge for the problem at hand and adds significant value to the proposed methodology. The application of EI-like in-fill criteria provided promising outlook when compared to existing and well-established methods, like the EGO algorithm. Further research in this direction is envisaged to continue the design and testing of such approaches in order to reach the same performance level of classical optimization algorithm at much lower computational cost.

Acknowledgements The author thanks Dr. Esther Andres Perez and Dr. Mario J. Martin-Burgos from Instituto Nacional de Tcnica Aeroespacial (INTA), Spain, for providing the NURBS parameterization tool and supporting its integration. The development and experiments presented in the paper have been partially achieved within the AG52 GARTEUR action group (http://ag52.blogspot.it/, www.garteur.org) that has been established to explore these surrogate-based global approaches. The main objective of the action group is, by means of a European collaborative research, to make a deep evaluation and assessment of surrogate-based global optimization methods for aerodynamic shape design, dealing with the main challenges as the curse of dimensionality, reduction of the design space and error metrics for validation, amongst others.

References

1. Amato M, Catalano P (2000) Non linear κ ε turbulence modeling for industrial applications. In: ICAS 2000 congress. IOS Press, Harrogate
2. Chandrashekarappa P, Duvigneau R (2007) Radial basis functions and Kriging metamodels for aerodynamic optimization. Rapport de recherche RR-6151, INRIA. http://hal.inria.fr/inria-00137602/en/

3. Cook PH, McDonald MA, Firmin MCP (1979) Aerofoil rae 2822—pressure distributions, boundary layer and wake measurements. In: Experimental data base for computer program assessment. AGARD. AGARD AR-138, Paper A6
4. Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. Prog Aerosp Sci 45(1–3):50–79. http://dx.doi.org/10.1016/j.paerosci.2008.11.001
5. Gutmann HM (2001) A radial basis function method for global optimization. J Glob Optim 19:201–227. doi:10.1023/A:1011255519438. http://dl.acm.org/citation.cfm?id=596093.596381
6. Iuliano E (2011) Towards a POD-based surrogate model for CFD optimization. In: Proceedings of the ECCOMAS CFD & optimization conference, Antalya
7. Iuliano E, Quagliarella D (2011) Surrogate-based aerodynamic optimization via a zonal pod model. In: Proceedings of the EUROGEN 2011 conference, Capua
8. Iuliano E, Quagliarella D (2013) Aerodynamic shape optimization via non-intrusive pod-based surrogate modelling. Comput Fluids 84:327–350
9. Iuliano E, Quagliarella D (2013) Aerodynamic shape optimization via non-intrusive pod-based surrogate modelling. In: Proceedings of 2013 IEEE CEC congress on evolutionary computation
10. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Global Optim 13:455–492. doi:10.1023/A:1008306431147. http://dx.doi.org/10.1023/A:1008306431147
11. Queipo N, Haftka R, Shyy W, Goel T, Vaidyanathan R, Kevintucker P (2005) Surrogate-based analysis and optimization. Prog Aerosp Sci 41(1):1–28. http://linkinghub.elsevier.com/retrieve/pii/S0376042105000102
12. Schonlau M, Welch WJ, Jones DR (1998) Global versus local search in constrained optimization of computer models. Lecture notes-monograph series, vol 34. doi:10.2307/4356058. http://dx.doi.org/10.2307/4356058
13. Simpson TW, Toropov VV, Balabanov V, Viana FAC (2008) 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. American Institute of Aeronautics and Astronautics, pp 1–22
14. Sóbester A, Leary S, Keane A (2004) A parallel updating scheme for approximating and optimizing high fidelity computer simulations. Struct Multidiscip Optim 27:371–383. doi:10.1007/s00158-004-0397-9. http://dx.doi.org/10.1007/s00158-004-0397-9.
15. Vitagliano PL, Quagliarella D (2003) A hybrid genetic algorithm for constrained design of wing and wing-body configurations. In: Bugeda G, Désidéri JA, 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

Chapter 3
PCA-Enhanced Metamodel-Assisted Evolutionary Algorithms for Aerodynamic Optimization

Varvara G. Asouti, Stylianos A. Kyriacou, and Kyriakos C. Giannakoglou

Abstract This paper deals with evolutionary algorithms (EAs) assisted by surrogate evaluation models or metamodels (metamodel-assisted EAs, MAEAs) which are further accelerated by exploiting the principal component analysis (PCA) of the elite members of the evolving population. In each generation of the MAEA, PCA is used to (a) better guide the application of evolution operators and (b) train metamodels, in the form of radial basis functions networks, on patterns of smaller dimension. Note that the present MAEA relies upon "local" metamodels which are trained on-line, separately for each and every population member. Compared to previous works by the same authors, this paper proposes a new way to apply the PCA technique. In particular, the front of non-dominated solutions is divided into sub-fronts and the PCA is applied "locally" to each sub-front. The proposed method is demonstrated in multi-objective, constrained, aerodynamic optimization problems.

3.1 Introduction

EAs are capable of handling complex, constrained, multi-objective problems by accommodating any analysis/evaluation software, without even having access to its source code. Being the most known representative of global optimization methods, EAs are widely used to solve engineering optimization problems. Their only drawback and main reason preventing the extensive use of EAs in large-scale problems is the great number of calls to the evaluation software required for capturing the optimal solution(s). In real-world applications, the computational cost per evaluation is often quite high and, in combination with a great number

V.G. Asouti (✉) • S.A. Kyriacou • K.C. Giannakoglou
Parallel CFD & Optimization Unit, National Technical University of Athens, Iroon Polytechniou 9, 15780 Athens, Greece
e-mail: vasouti@mail.ntua.gr; stelios.kyriacou@gmail.com; kgianna@central.ntua.gr

E. Iuliano, E.A. Pérez (eds.), *Application of Surrogate-based Global Optimization to Aerodynamic Design*, Springer Tracts in Mechanical Engineering,
DOI 10.1007/978-3-319-21506-8_3

of optimization unknowns (N >>), the optimization turnaround time might even become prohibitive. The CFD-based optimization is a typical example.

As population-based search methods, EAs are amenable to parallelization. The concurrent evaluation of candidate solutions on different processors is straightforward. Smarter usage of a multi-processor system can be made by means of asynchronous EAs [1], which overcome the synchronization barrier at the end of each generation.

On the other hand, the most common technique to reduce the CPU cost of the EA-based optimization is the extensive use of surrogate evaluation models (or metamodels). They are used to inexpensively approximate the objective function value(s), by replacing the call to the costly problem-specific evaluation model. Metamodels, after being trained on previously-seen solutions, can be incorporated into an EA in different ways, depending on whether their training takes place during (on-line) [7, 8, 11, 14] or separately from (off-line) the evolution [3, 13]. In this paper, metamodel-assisted EAs (MAEAs) with on-line trained metamodels [5, 8], are employed. According to the inexact pre-evaluation (IPE) approach, with the exception of a few starting generations, all population members are approximately evaluated using local metamodels trained separately for each one of them. A few most promising among them, as indicated by the metamodel, are re-evaluated on the exact and costly model.

Engineering optimization problems usually have several objectives and/or constraints and involve a great number of design variables. The high number of design variables deteriorates the efficiency of a conventional EA, since it requires more evaluations and increases the computational cost. Also, in MAEAs, the metamodels' training time increases and the prediction accuracy decreases as the number of design variables increases.

A way to overcome the overall performance degradation, often referred to as the curse of dimensionality, is by decreasing the problem dimension via dimension reduction techniques. This can be done using principal component analysis (PCA) techniques [6]. In contrast to other methods where PCA is exclusively used to reduce the optimization problem dimension, here PCA is used both to guide the application of the evolution operators (without dimension reduction) and reduce the number of sensory units of the radial basis function (RBF) networks used as metamodels [9, 10]. Compared to [9, 10], there is a novel enhancement of the already published method. PCA is not applied to the entire elite set but to sub-fronts, after appropriately splitting the set of elite members into parts. According to this scheme, each parent is transformed using information from the PCA of the closest in the design space sub-front. By doing so, the EA requires less evaluations compared to [9]. The proposed method is demonstrated in three cases concerned with the preliminary design of a supersonic business jet, the aeroelastic design of a wind turbine blade and the aerodynamic design of an isolated airfoil.

3.2 PCA-Enhanced EAs and MAEAs

The PCA of a data set in the design space leads to new orthogonal linear combinations of the “original” variables, each of which with a different variance. Once the principal components are computed, the first one is associated with the largest variance, the second one is perpendicular to the first and associated with the next largest variance and so forth.

Without loss in generality, the proposed method is demonstrated on a multi-objective optimization (MOO) problem. In each generation, the PCA of the elite set is carried out. In single-objective optimization (SOO) problems, instead of the front of non-dominated solutions, the elite set may comprise the current optimal and a few near-optimal solutions.

The elite set is brought into the form of a standardized data set X with zero mean value and unit standard deviation along all directions. Based on X, the covariance matrix P is computed as

$$P_{N \times N} = \frac{1}{\epsilon} X X^T \tag{3.1}$$

where ϵ is the elite set size. Using the spectral decomposition theorem [2], P is written as

$$P_{N \times N} = U \Lambda U^T \tag{3.2}$$

where Λ is a diagonal matrix with the eigenvalues and U a $N \times N$ matrix formed by the eigenvectors of P as rows.

Compared to [10], in this paper the front of the non-dominated solutions is divided into a user-defined number of sub-fronts (Fig. 3.1) and the PCA is applied locally to each sub-front. The two ways of taking advantage of PCA are briefly described below. Section 3.2.1 describes how the principal components can be used to drive the evolution operators. Section 3.2.2 describes how the PCA is used to reduce the number of sensory nodes of the metamodel. The combined use of these two options is straightforward.

3.2.1 PCA-Enhanced Evolution Operators

The evolution operators are applied to a temporarily transformed (rotated) design space according to the principal components computed by the PCA. If $\mathbf{x}_i$ is a candidates’ solution design vector, the vector $\mathbf{e}_i$ aligned with the principal component directions, is defined as follows:

$$\mathbf{e}_i = U(\mathbf{x}_i - \boldsymbol{\mu}_X) \tag{3.3}$$

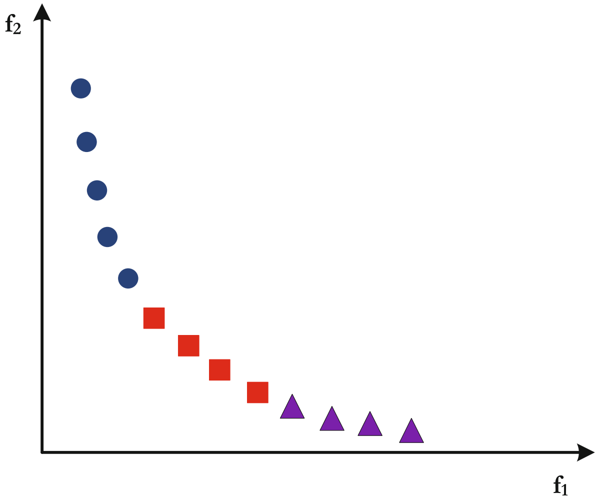


Fig. 3.1 Schematic representation of the division of the non-dominated individuals into three sub-fronts

where $\boldsymbol{\mu}_X$ is the vector of mean (over the elite set) design variables. Once the design space becomes aligned with the principal component directions, the application of crossover and mutation operators follows. The crossover operator is applied to the transformed parent population genotypes. If a different PCA is performed for each sub-front, the parents design vector is rotated based on the PCA of the sub-front parents are associated with (the closest, in terms of Euclidean distances, in the design space).

Concerning mutation, an increased mutation probability along the directions with small variances is required. To this end, instead of using a constant, user-defined, mutation probability p_m, the mutation probability assigned to each principal component (index i) is given by Kyriacou et al. [10]

$$p_m^i = \alpha p_m + (1-\alpha)p_m \frac{N}{D_V} \cdot \frac{V_{max} - V_i}{V_{max} - V_{min}} \tag{3.4}$$

where $\alpha \in [0,1]$, V_i is the variance of the current elite set with respect to the ith principal component, $V_{max} = \max\{V_1, \dots, V_N\}$, $V_{min} = \min\{V_1, \dots, V_N\}$ and

$$D_V = \sum_{i=1}^{N} \frac{V_{max} - V_i}{V_{max} - V_{min}} \tag{3.5}$$

Each mutated offspring is, then, transformed back to the original design space. This inverse transformation, from $\mathbf{e}_i$ to $\mathbf{x}_i$, is given by

$$\mathbf{x}_i = U^{-1}\mathbf{e}_i + \boldsymbol{\mu}_X \tag{3.6}$$

Note that since U is an orthogonal matrix, its inversion has negligible CPU cost.

3.2.2 EA with PCA-Assisted Metamodels

An additional role of the PCA in MAEAs is to reduce the dimension of a problem having a large number of design variables, so as to overcome the so-called curse of dimensionality. When handling high-dimensional optimization problems, an increased number of training patterns is required to build an accurate metamodel and this increases a lot the cost of the training procedure. The reduction of the number of sensory nodes of a metamodel (RBF network), through PCA, increases its prediction accuracy and accelerates the training process.

The eigenvectors included in matrix U, Eq. (3.2), are associated with the variances of the design variables and can be used to identify the directions where the elite members are less or more scattered (high variance indicates scattered data whereas small variance corresponds to less scattered data). Based on this, a user-defined number of RBF network sensory units, in fact those corresponding to the directions of the design space with high variances, are filtered out. Consequently, the RBF network is trained on lower dimension data and this turns out to yield more reliable networks, trained at lower cost.

This truncation applies only during the RBF network training. In particular, during the IPE phase, for all population members, a local metamodel is built by selecting training patterns following the procedure described in detail in [7]. The training patterns are rotated (Eq. (3.3)) and, then, their components associated with the higher eigenvalues are excluded from the training.

3.3 Applications

The proposed methodology is applied in three cases in the field of aerodynamics, namely the preliminary design of a supersonic business jet, the aeroelastic design of a wind turbine blade and the aerodynamic design of an isolated airfoil. In what follows the term M(PCA)AEA(PCA) will be used to denote an MAEA where both the evolution operators and metamodels are assisted by the PCA.

3.3.1 Preliminary Design of a Supersonic Business Jet

The first two-objective application deals with the preliminary design of a supersonic business jet for maximum range (R) and minimum take-off weight (TOW). The design variables (13 in total) are related to the flight conditions (cruise altitude and Mach number), fuel weight and aircraft geometry (wing and tail reference areas, leading and training edge sweep angles, etc.). This is, in fact, a multi-disciplinary optimization problem which involves disciplines such as aerodynamics, structures/weights, propulsion, etc., each of which is modeled using low-fidelity models.

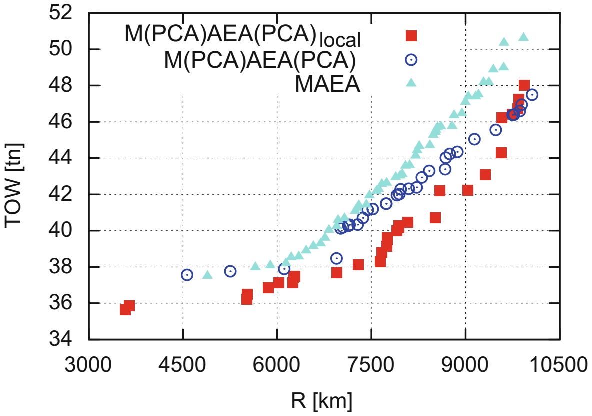


Fig. 3.2 Preliminary design of a supersonic business jet: comparison of the fronts of non-dominated solutions computed using MAEA (*filled triangles*), M(PCA)AEA(PCA) (*empty circles*) and M(PCA)AEA(PCA)$_{\text{local}}$ (*filled squares*) at the cost of 2000 evaluations

In this application, computations based on standard MAEA and M(PCA)AEA(PCA) in the two variants of the latter in which PCA is driven by either the entire elite front or two sub-fronts have been carried out and their results are compared. In all three runs, a $(\mu, \lambda) = (30, 90)$ MAEA (with μ parents and λ offspring) was used and the metamodel-based IPE phase started once the database (DB) recorded the first 100 entries (already evaluated individuals). Then, in each subsequent generation, five to ten individuals were re-evaluated on the problem-specific tool. The same stopping criterion of 2000 evaluations was imposed during all runs. A comparison of the obtained fronts of the three methods is shown in Fig. 3.2. In either form, the M(PCA)AEA(PCA) outperforms the standard MAEA. Furthermore, as expected, the local application of PCA performs better compared to the PCA of the entire elite set.

3.3.2 *Aeroelastic Design of a Wind Turbine Blade*

The second optimization problem is about the aeroelastic design of a wind turbine blade for maximum annual energy production (AEP) and minimum mass (m). The rotation speed of the blade is $\Omega_R = 1.267$ rad/s and the AEP is determined using the whole range of wind speeds between cut-in and cut-out wind speed, with $V_{in} = 5$ m/s and $V_{out} = 25$ m/s up to the rated wind speed (in this case, for a 5 MW wind turbine, V_{rated} equals 11.35 m/s). The optimization is subject to constraints for the flapwise bending moment at the blade root ($M_{f,\text{root}}$) and the maximum flapwise ($\sigma_{f,\text{max}}$) and edgewise $\sigma_{e,\text{max}}$ stresses, as

$$M_{f,\text{root}} \leq 10000\,\text{kNm}\ ,\quad \sigma_{f,\text{max}} \leq 64000\,\text{kN/m}^2\ ,\quad \sigma_{e,\text{max}} \leq 90000\,\text{kN/m}^2 \qquad (3.7)$$

For each blade section an I-beam structural model is assumed, Fig. 3.3, using two spar caps joined together by a shear web. The design variables are the blade chord,

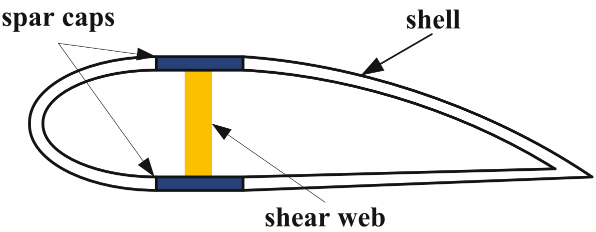


Fig. 3.3 Aeroelastic design of a wind turbine blade: schematic representation of the I-Beam structural model for each blade section

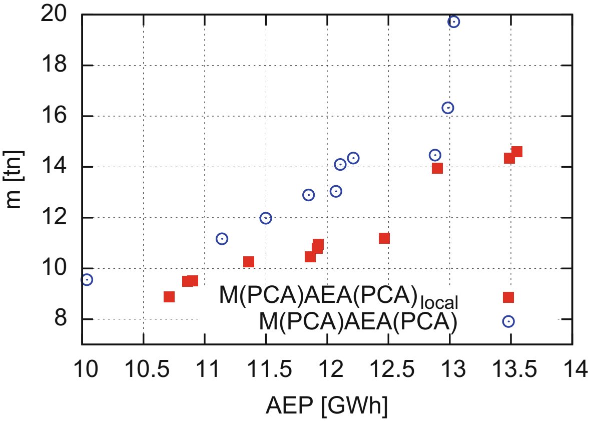


Fig. 3.4 Aeroelastic design of a wind turbine blade: comparison of the fronts of non-dominated solutions computed using M(PCA)AEA(PCA) (*empty circles*) and M(PCA)AEA(PCA)$_{\text{local}}$ (*filled squares*) at the same CPU cost

twist and thickness, the I-beam base length, the spar caps, shear web and shell thickness at nine preselected spanwise positions. Targeting realizable designs, the objective functions are also constrained, as follows:

$$AEP \geq 10\,\text{GWh}\,, \quad m \leq 20\,\text{tn} \tag{3.8}$$

The evaluation of each candidate solution was based on the Blade Element Momentum model included in NTUA's aeroelastic software GAST (General Aerodynamic and Structural numerical Tool [12]). This case was studied using the M(PCA)AEA(PCA) with $(\mu, \lambda) = (30, 90)$ and a termination criterion of 10,000 evaluations. The metamodel was activated once 500 feasible (non-violating the constraints) solutions were stored in the DB and, during the IPE phase, 9–15 individuals were re-evaluated on the GAST software. The resulted fronts of non-dominated solutions are compared in Fig. 3.4 where, in this case too, the local application of PCA proved to yield better solutions.

3.3.3 Optimization of an Isolated Airfoil

The last case is a two-objective, constrained, optimization problem concerned with the optimization of an isolated airfoil for minimum drag (C_D) and maximum lift (C_L) coefficient. The flow conditions are: free-stream Mach number $M_\infty = 0.3$, flow angle $a_\infty = 4^o$ and $Re_c = 6.2 \times 10^6$. The airfoil is parameterized using two Bézier curves, separately for the pressure and suction sides, with eight control points each. Only the internal control points of each Bézier curve may vary, summing up to 24 design variables.

The tool used to evaluate each individual applies a viscous–inviscid flow interaction method based on an integral boundary layer method, coupled with an external flow solver [4]. To prevent the formation of unacceptably thin airfoils, geometrical constraints on the airfoil thickness $t(x)$ at three chordwise positions (x), are imposed, namely

$$t(0.25c) \geq 0.12c,\ t(0.5c) \geq 0.12c,\ t(0.75c) \geq 0.05c \tag{3.9}$$

where c is the chord length. The constraint violation check is performed in two stages. Airfoils with a severe violation of even a single geometrical constraint were immediately rejected by assigning a death penalty to their objective functions, without undergoing evaluation using the flow solver. On the other hand, other airfoils which violated the thickness constraints to a lesser extent were allowed to undergo flow evaluation for computing their objective function values which were, then, penalized using an exponential penalty term.

This case was studied with MAEA and M(PCA)AEA(PCA) by applying PCA on a single front of non-dominated solutions and on two sub-fronts and the results obtained are compared. In the case of the local application of PCA, the first sub-front corresponds to the "family" of low lift-low drag airfoils, whereas the second to that of high lift-high drag airfoils. To separate the two sub-fronts a simple criterion was used. In each generation, after computing/updating the current elite set, its median was used as the threshold between the two airfoil "families".

In both cases, a stopping criterion of 1500 evaluations was imposed. A $(\mu, \lambda) = (20, 60)$ MAEA was used and metamodels were applied once the first 300 feasible solutions were stored in the DB. During the IPE phase, in each generation, five to eight individuals were re-evaluated on the integral boundary layer method.

Figure 3.5 compares fronts of non-dominated solutions computed by the three methods at the same cost, practically the same number of evaluations. The M(PCA)AEA(PCA) was able to capture parts of the Pareto front which MAEA couldn't. Though, in the low lift-low drag region, MAEA computed some individuals dominating those of M(PCA)AEA(PCA), in the high lift-high drag region MAEA's performance was very poor. Regarding the two variants of M(PCA)AEA(PCA), individuals resulted from the local application of PCA were better spread along the front and the majority of them dominated those from PCA on the entire elite set. Three airfoil shapes from each sub-front, i.e. the low lift-low

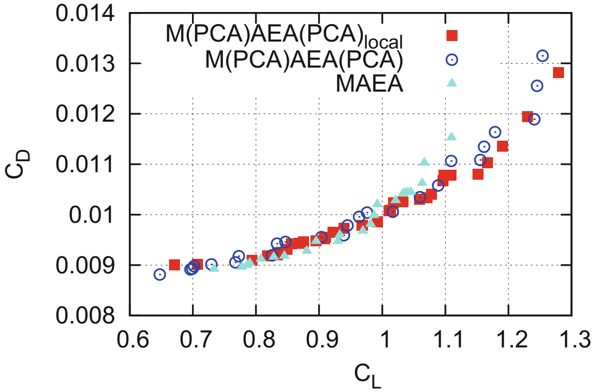


Fig. 3.5 Optimization of a 2D isolated airfoil: comparison of the computed fronts of non-dominated solutions computed using MAEA (*filled triangles*), M(PCA)AEA(PCA) (*empty circles*) and M(PCA)AEA(PCA)$_{\text{local}}$ (*filled squares*)

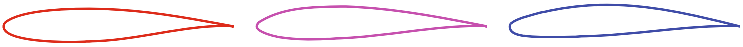


Fig. 3.6 Optimization of a 2D isolated airfoil: three airfoil shapes corresponding to the low lift-low drag part of the Pareto front (first sub-front) shown in Fig. 3.5. *From left to right* the three airfoils yield the following C_L, C_D values: (0.707, 0.00901), (0.817, 0.00918) and (0.894, 0.00948)

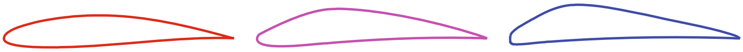


Fig. 3.7 Optimization of a 2D isolated airfoil: three airfoil shapes corresponding to the high lift-high drag part of the Pareto front (second sub-front) shown in Fig. 3.5. *From left to right* the three airfoils yield the following C_L, C_D values: (0.968, 0.00978), (1.096, 0.010678) and (1.19, 0.01136)

drag and high lift-high drag ones are shown in Figs. 3.6 and 3.7, respectively. From the presented airfoil shapes, the two “families” can easily be distinguished. As shown in Fig. 3.5, this sub-division of the elite front is beneficiary for the search method.

Conclusions

This paper reconfirms the superiority of the so-called M(PCA)AEA(PCA) algorithm, originally proposed and assessed in [10], using three test problems. In each generation of the EA, the analysis of the population members using information related to the principal directions, as extracted by the characteristics of the current elite individuals, is used to (a) reduce the dimensionality of the RBF networks, by making their training easier and their predictions more dependable and (b) apply the evolution operators in a

(continued)

properly transformed space. In addition, this paper demonstrated the increase in performance offered by the local application of the PCA-driven actions during an M(PCA)AEA(PCA) run. Locality is related to the splitting of the current front of non-dominated solutions into sub-fronts, in each of which an independent PCA is performed. The better performance of the proposed scheme is justified by the fact that candidate solutions at the different edges of a Pareto front may have very different characteristics, the "averaging" of which through the single PCA should preferably be avoided. In this paper, restricted to two-objective problems with constraints, the current front of non-dominated solutions was decomposed into two sub-fronts, using its dynamically changing median.

Acknowledgements The authors express their thanks to Professors S. Voutsinas and V. Riziotis, NTUA, for providing the necessary data and evaluation software for the wind turbine aeroelastic optimization case and their constructive comments and suggestions. This study has been co-financed by the European Union (European Social Fund-ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)—Research Funding Program: THALES. Investing in knowledge society through the European Social Fund.

References

1. Asouti VG, Giannakoglou KC (2009) Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes. Eng Optim 41:241–257
2. Axler S (1997) Linear algebra done right. Springer, New York
3. Büche D, Schraudolph N, Koumoutsakos P (2005) Accelerating evolutionary algorithms with Gaussian process fitness function models. IEEE Trans Syst Man Cyber 35:183–194
4. Drela M, Giles MB (1987) Viscous-inviscid analysis of transonic and low Reynolds number airfoils. J Am Inst Aeronaut Astronaut 25:1347–1355
5. Giannakoglou KC (2002) Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. Prog Aerosp Sci 38:43–76
6. Haykin S (1999) Neural networks: a comprehensive foundation. Prentice Hall, Upper Saddle River
7. Karakasis MK, Giannakoglou KC (2006) On the use of metamodel-assisted, multi-objective evolutionary algorithms. Eng Optim 38:941–957
8. Karakasis MK, Giotis AP, Giannakoglou KC (2003) Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization. Int J Numer Methods Fluids 43:1149–1166
9. Kyriacou SA, Weissenberger S, Giannakoglou KC (2012) Design of a matrix hydraulic turbine using a metamodel-assisted evolutionary algorithm with PCA-driven evolution operators. Int J Math Model Numer Optim 3:45–63
10. Kyriacou SA, Asouti VG, Giannakoglou KC (2014) Efficient PCA-driven EAs and metamodel-assisted EAs, with applications in turbomachinery. Eng Optim 46:895–911
11. Ong YS, Lum KY, Nair PB, Shi DM, Zhang ZK(2003) Global convergence of unconstrained and bound constrained surrogate-assisted evolutionary search in aerodynamic shape design. Congr Evol Comput 3:1856–1863

12. Riziotis VA, Voutsinas SG (1997) Gast: a general aerodynamic and structural prediction tool for wind turbines. In: Proceedings of the EWEC-1997, Dublin
13. Shyy W, Papila N, Vaidyanathan R, Tucker K (2001) Global design optimization for aerodynamics and rocket propulsion components. Prog Aerosp Sci 37:59–118
14. Ulmer H, Streichert F, Zell A (2003) Evolution strategies assisted by Gaussian processes with improved pre-selection criterion. Congr Evol Comput 1:692–699

Chapter 4
Multi-Objective Surrogate Based Optimization of Gas Cyclones Using Support Vector Machines and CFD Simulations

Khairy Elsayed and Chris Lacor

Abstract In order to accurately predict the complex nonlinear relationships between the cyclone performance parameters (The Euler and Stokes numbers) and the four significant geometrical dimensions (the inlet section height and width, the vortex finder diameter and the cyclone total height), the support vector machines approach has been used. Two support vector regression surrogates (SVR) have been trained and tested by CFD datasets. The result demonstrates that SVR can offer an alternative and powerful approach to model the performance parameters. The SVR model parameters have been optimized to obtain the most accurate results from the cross validation steps. SVR (with optimized parameters) can offer an alternative and powerful approach to model the performance parameters better than Kriging. SVR surrogates have been employed to study the effect of the four geometrical parameters on the cyclone performance. The genetic algorithms optimization technique has been applied to obtain a new geometrical ratio for minimum Euler number and for minimum Euler and Stokes numbers. New cyclones over-perform the standard Stairmand design performance. Pareto optimal solutions have been obtained and a new correlation between the Euler and Stokes numbers is fitted.

K. Elsayed (✉)
Department of Mechanical Engineering, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Faculty of Engineering at El-Mattaria, Mechanical Power Engineering Department, Helwan University, Masaken El-Helmia P.O., Cairo 11718, Egypt
e-mail: kelsayed@vub.ac.be; kelsayed75@gmail.com

C. Lacor
Department of Mechanical Engineering, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
e-mail: chris.lacor@vub.ac.be

E. Iuliano, E.A. Pérez (eds.), *Application of Surrogate-based Global Optimization to Aerodynamic Design*, Springer Tracts in Mechanical Engineering,
DOI 10.1007/978-3-319-21506-8_4

4.1 Introduction

Cyclone separators are widely used in gas-solid separation for aerosol sampling and industrial applications [1] where both the gravity and centrifugal force are used to separate solids from a mixture of solids and fluids. Cyclones have the following advantages. Simplicity in construction, contain no moving parts, relatively maintenance free, can handle high pressure and temperature mixtures and corrosive gases, relative economy in power consumption. Due to these advantages, cyclone separators have become one of the most important particle removal devices in both engineering and process operation [1] such as cement industry, oil and gas, coal fired boiler, workshops and vacuum cleaners.

4.1.1 Cyclone Geometry

The cyclone geometry is described by seven geometrical parameters, viz. the inlet height a and width b, the vortex finder diameter D_x and length S, the cylinder height h, the cyclone total height H_t and cone-tip diameter B_c as shown in Fig. 4.1. It has been approved in previous studies by the authors that only four geometrical

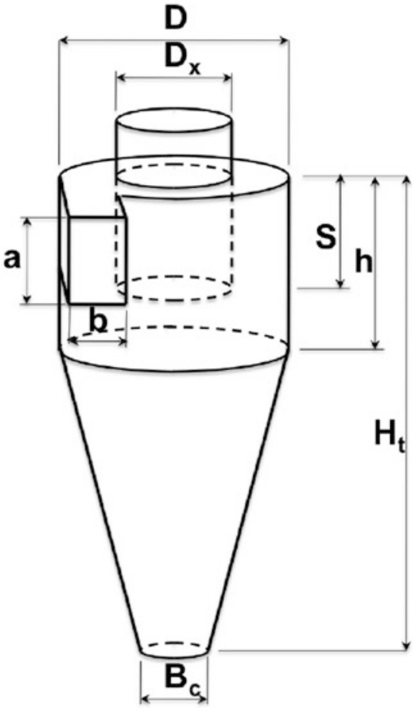


Fig. 4.1 Cyclone geometry. In this study, $h/D = 1.5, S/D = 0.5, B_c/D = 0.375$

Table 4.1 The definition of the cyclone performance parameters

The Euler number Eu	The Stokes number Stk_{50}
The Euler number is the dimensionless pressure drop ΔP. $\Delta P =$ (the area- and time-averaged static pressure at the inlet section) - (the area- and time-averaged static pressure at the gas exit section). $Eu = \frac{\text{The pressure drop between the inlet and the gas exit}}{\text{The average kinetic energy at the inlet}}$ $Eu = \frac{\Delta P}{\frac{1}{2}\rho V_{in}^2}$ where ρ is the gas density and V_{in} is the average inlet velocity. Eu is not affected by operating conditions in the high Reynolds number range ($R_e > 5 \times 10^4$, $Re = \frac{\rho V_{in} D}{\mu}$) [3, 10]	Stk_{50} is the dimensionless cut-off diameter x_{50}. x_{50} is the particle diameter that produces 50 % collection efficiency. $Stk_{50} = \rho_p x_{50}^2 V_{in}/(18\mu D) = \frac{\tau_p}{\tau_f}$ [11]. It is the ratio between the particle relaxation time (the time constant in the exponential decay of the particle velocity due to drag) $\tau_p = \rho_p x_{50}^2/(18\mu)$ and the gas flow integral time scale $\tau_f = D/V_{in}$ where ρ_p is the particle density and μ is the gas viscosity

parameters significantly affect the cyclone flow pattern and performance [2–9]. The four significant factors are the inlet section height a and width b, the vortex finder diameter D_x and the cyclone total height H_t.

4.1.2 Cyclone Performance

Besides the separation efficiency (or alternatively, the cut-off diameter for low mass particle loading), pressure drop is another major index for cyclone performance evaluation. Therefore, it is necessary to obtain an accurate model to determine the complex relationship between the performance parameters and the cyclone characteristics. Table 4.1 presents more details about the two performance parameters.

4.1.3 Literature Review

To estimate the cyclone performance parameters there are five approaches:

1. Experimental investigations [12]
2. Theoretical and semi-empirical models [13]
3. Statistical models [14]
4. Computational fluid dynamics (CFD) [2–4, 7, 8]
5. Surrogate models (e.g., Polynomial regression (PR), Kriging (KG) and artificial neural networks (ANN)) [5, 6]

The afore-mentioned prediction models (PR, KG and ANN) have numerous drawbacks, which include locally optimal solutions, low generalization, over-fitting and poor stability [15]. The support vector machine (SVM) surrogate can offer a better alternative to these models. In the field of performance evaluation for cyclone separators, unfortunately, SVM does not receive a great deal of attention on its algorithmic advantages. There is only one study using the support vector regression (SVR) on cyclone separator performed by Zhao [1]. He approved the potential of SVR to model the effect of cyclone geometry on the pressure drop (based on experimental dataset collected from different sources) but he did not go further to use the fitted SVR to study the effect of each parameter on the performance or for optimization. Moreover, he used the traditional approach to find suitable values for SVR parameters.

4.1.4 Target of This Study

This study aims to:

- Apply the SVR surrogate to model the variation of the two cyclone performance parameters with the change in the most significant geometrical parameters based on CFD based dataset.
- Introduce a computationally cheap framework for SVR parameter optimization using a Python code.
- Compare the accuracy of the fitted SVRs models with the KG models.
- Study the effect of each significant geometrical parameter on the cyclone performance using the SVR models.
- Optimize the cyclone performance for minimum Euler number as well as for best performance using multi-objective optimization technique.

4.2 Least Squares: Support Vector Regression

The least squares support vector regression (LS-SVR) was introduced by Suykens et al. [16] as a reformulation to the standard SVR. LS-SVR simplified the standard SVR model to a great extent by applying linear least squares criteria to the loss function instead of a traditional quadratic programming method [16]. As excellent examples of the nonlinear dynamic system, LS-SVR based on the structured risk minimization principle has been successfully applied to many fields of function approximation and pattern recognition because of its high accuracy and generalization capabilities [17]. Compared with ANN, LS-SVR seeks to minimize an upper bound of the generalization error instead of the empirical error, and can provide more reliable and better generalization performance under the same training conditions [18].

In the LS-SVR model, the training dataset of l points is assumed to be x_k, y_k ($k = 1, 2, \cdots, l$), in which $x_k \in R^n$ is the input vector and $y_k \in R$ is the corresponding target vector. The regression problem can be transformed into the following optimization problem [15, 19]:

$$\underset{\omega,b,e_k}{\text{minimize}}\,\Psi(\omega,e)=\frac{1}{2}\omega^T\omega+\frac{C}{2}\sum_{k=1}^{l}e_k^2 \tag{4.1}$$

$$\text{subject to}\,y_k=\omega^T\phi(x_k)+b+e_k,\quad (k=1,2,\ldots,l) \tag{4.2}$$

where e_k is the error between the predicted value and the true value of the system, $C > 0$ is the regularization parameter applied to minimize estimation error and control function smoothness, $\phi(\cdot)$ denotes the nonlinear mapping from input spaces to feature space, ω is an adjustable weight vector and b is the bias (scalar threshold). Equation (4.2) is the constraint.

The resulting LS-SVR model for function estimation is obtained as:

$$\hat{y}=f(x)=\sum_{k=1}^{l}\alpha_k K(x_k,x)+b \tag{4.3}$$

In Eq. (4.3), $K(x_k,x)$ is the kernel function which satisfies Mercers condition corresponding to a dot product in some feature spaces. Four common Mercer kernel functions are widely used [15]:

Linear kernel:	$K(x_k,x)=x_k^Tx$
Polynomial kernel:	$K(x_k,x)=(x_k^Tx/\sigma^2+\gamma)^d$
RBF kernel:	$K(x_k,x)=exp(-\gamma\parallel x_k-x\parallel^2)$
Sigmoid kernel:	$K(x_k,x)=tanh(\gamma x_k^Tx+r)$

where d,γ and σ are constants.

Because RBF kernels map samples into high dimensional space in a nonlinear fashion and have fewer parameters to set, and because this method handles the nonlinear relationship well and has an excellent overall performance, it is by far the most popular option for kernel function types [16]. This study consequently adopted an RBF kernel function, shown in Eq. (4.4) in order to contribute to the LS-SVR model's achieving optimal solution.

$$K(x_k,x)=exp(-\gamma\parallel x_k-x\parallel^2) \tag{4.4}$$

Generally, LS-SVR solves linear equations and will lead to important reductions in calculation complexity. Compared with SVR, *LS-SVR is characterized by faster training speed, higher stability and better control strategy* [15, 19].

4.2.1 LS-SVR Parameter Optimization

The LS-SVR performance heavily depends on the choice of several hyperparameters, which are necessary to define the optimization problem and the final LS-SVR model.

To design an LS-SVR, one must choose a kernel function, set hyperparameters such as the kernel parameters, and determine a regularization parameter C. The hyperparameters that should be optimized include the regularization parameter C and the kernel function parameters such as γ for the radial basis function (RBF) kernel. Thus, selecting appropriate model parameters has a crucial impact on prediction accuracy. Unfortunately, there is no exact method to obtain the optimal set of LS-SVR hyperparameters; consequently, a search algorithm must be applied to obtain the parameters.

For the nonlinear LS-SVR, its generalization performance depends on the proper setting of parameters C and kernel parameters γ. Inappropriate hyperparameters combinations in LS-SVR lead to over-fitting or under-fitting. One procedure to obtain the LS-SVR parameters follows the trial and error approach to minimize some generalized error measures such as the mean squared error. This procedure is time-consuming, tedious and unable, in many cases, to converge at the global optimum. Zhao [1] applied the two-step search technique to dynamically seek the optimal values for the LS-SVR parameters. The two steps are: First perform a coarse search to identify a better region in search field according to contour lines of MSE. Then perform a fine search over that region. The disadvantage of the multi-step search technique is that it will be more prone to be trapped in local optimum point especially if a limited number of points are used.

In this study, we propose an alternative approach. The proposed approach employs the simulated annealing optimization technique to heuristically seek the optimal values for the LS-SVR parameters that minimize the difference between the predicted and the true values.

The simulated annealing (SA) is used in this study to optimize the parameters of SVR: C and kernel parameter γ of RBF-kernel function. In the training and testing process of LS-SVR, the objective is to minimize the errors between the actual and predicted values of the testing samples. Therefore, the objective (fitness) function of SA is the mean squared error from the cross validation.

In the parameters optimization process, K-fold cross validation is employed to avoid the over-fitting and to calculate the fitness function. The original sample is randomly partitioned into K subsamples. In these subsamples, a single subsample is used as the validation data for testing the model while the other K-1 subsamples are used as training data. The cross validation process is then repeatedly performed K times, with each of the K subsamples selected exactly once as the validation data. The cross validation error is estimated as the average mean squared error (MSE) on test subsamples, as shown in Eq. (4.5). Commonly, fivefold and tenfold cross validation is the most widely used method. In this study, the fivefold cross validations are employed to estimate the MSE.

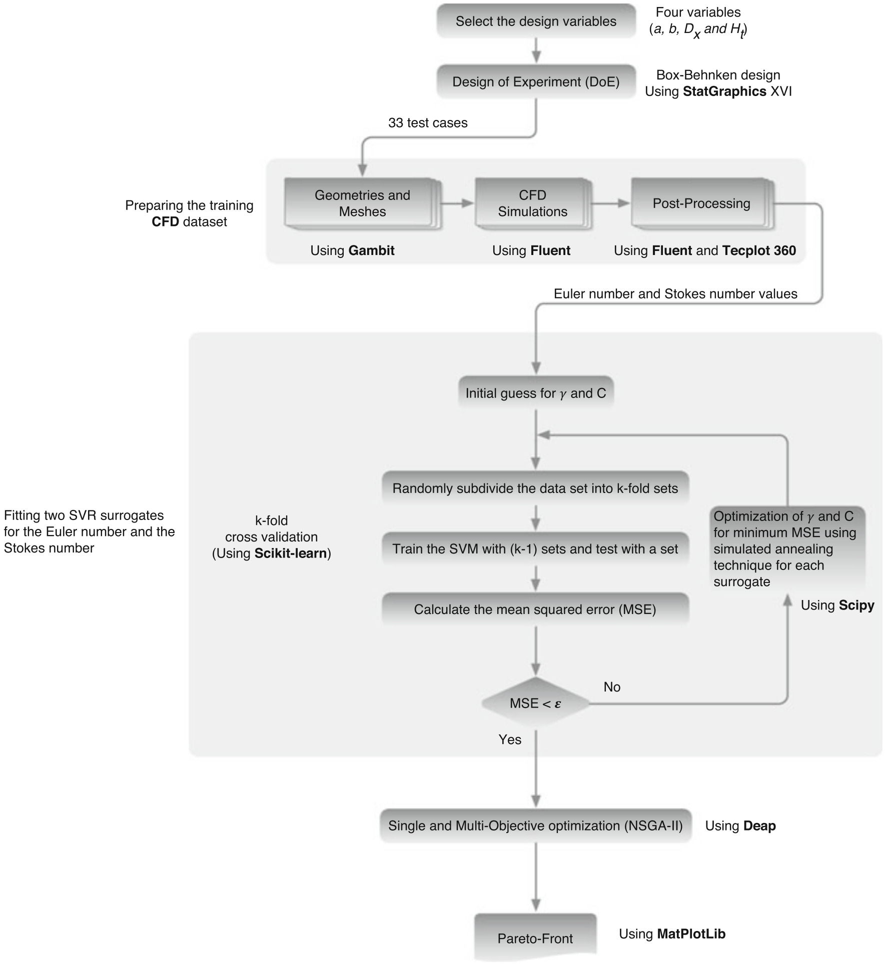

Select the design variables
Four variables
(a, b, D_x and H_t)
Design of Experiment (DoE)
Box-Behnken design
Using **StatGraphics** XVI
33 test cases
Preparing the training
CFD dataset
Geometries and Meshes
CFD Simulations
Post-Processing
Using **Gambit**
Using **Fluent**
Using **Fluent** and **Tecplot 360**
Euler number and Stokes number values
Fitting two SVR surrogates for the Euler number and the Stokes number
Initial guess for γ and C
k-fold cross validation (Using **Scikit-learn**)
Randomly subdivide the data set into k-fold sets
Train the SVM with (k-1) sets and test with a set
Calculate the mean squared error (MSE)
Optimization of γ and C for minimum MSE using simulated annealing technique for each surrogate
Using **Scipy**
No
MSE < ε
Yes
Single and Multi-Objective optimization (NSGA-II)
Using **Deap**
Pareto-Front
Using **MatPlotLib**

Fig. 4.2 Flow chart for the complete optimization framework

$$\mathrm{MSE} = \frac{1}{K}\sum_{j=1}^{K}\left(\sum_{i=1}^{Nt_j}\left(\widehat{y_i} - y_i\right)^2\right) \tag{4.5}$$

where K is the number of folds (5 in this study), Nt_j is the number of testing points in fold j, y_i represents the actual values and $\widehat{y_i}$ represents the predicted values.

Figure 4.2 presents a flow chart for the complete optimization framework.

4.3 Results and Discussion

4.3.1 *The Training Dataset*

The training dataset has been created using the Box–Behnken design of experiment (DoE) and has been used in a previous study to optimize the cyclone geometry using the polynomial regression and RBF artificial neural network surrogates [7]. The minimum and maximum values for the four design parameters are listed in Table 4.2. To avoid scaling effect, all values are scaled (using the `preprocessing.MinMaxScaler` class from `Scikit-learn`) to be in range of 0 to 1 before being used in training the surrogates.

Table 4.3 presents the statistical descriptive parameters for the SVR (before and after parameter optimization) and Kriging surrogate. It is clear that the SVR with optimized parameter superior the performance of the Kriging model as is clear from the better matching between the statistical descriptive parameters of the input and the output results from the surrogate as well as the smaller value of MSE and the R^2 value close to unity.

4.3.2 *Geometry Effect*

One of the benefits of using surrogate models is to apply them to study the effect of each design variable on the response (performance parameters). The two optimized SVR models are used to study the effect of the four geometrical parameters on both the Euler number and the Stokes number. As is clear from Fig. 4.3, the SVR and KG models give the same trend of variation but the SVR models can predict more local variation than the KG model. It is worth to mention that the variation of the Euler number with the change in the vortex finder diameter D_x predicted by the SVR model is similar to that reported by the authors in previous studies [3, 7]. For the

Table 4.2 The values of the cyclone geometrical parameters used in the DoE (cf. Fig. 4.1)

Variables	Minimum	Center	Maximum
Inlet height, $x_1 = a/D$	0.4	0.55	0.7
Inlet width, $x_2 = b/D$	0.14	0.27	0.4
Vortex finder diameter, $x_3 = D_x/D$	0.2	0.475	0.75
Total cyclone height, $x_4 = H_t/D$	3.0	5.0	7.0
Cylinder height, h/D		1.5	
Vortex finder length, S/D		0.5	
Cone-tip diameter, B_c/D		0.375	

The values of the cylinder height, vortex finder length and the cone tip diameters are kept at the Stairmand design values, where $h - S = 1$ which is the optimum difference between the two dimensions as reported by many researchers [20]

Table 4.3 Statistical descriptive parameters for the SVR and KG models using the fivefold cross validation[a]

	SVR before parameter optimization (using the default values of C and γ)					
	Eu			$Stk_{50} \times 10^3$		
	x		y	x		y
Average	7.2436		4.3035	1.9496		1.9404
Standard deviation	6.9465		1.1746	0.9211		0.7573
Minimum	1.2110		2.0240	0.5880		0.6646
Maximum	27.2570		6.5702	3.8400		3.2680
Range	26.0460		4.5462	3.2520		2.6035
Coefficient of determination, R^2		0.131			0.900	
Mean squared error		47.750			0.076	
Intercept		3.2928			0.3480	
Slope		0.1395			0.8168	
The SVR regularization parameter, C		10.0			10.0	
The kernel parameter, γ		10.0			10.0	
	SVR after parameter optimization					
	Eu			$Stk_{50} \times 10^3$		
	x		y	x		y
Average	7.2436		7.2966	1.9496		1.9938
Standard deviation	6.9465		6.9108	0.9211		0.9271
Minimum	1.2110		1.3118	0.5880		0.4882
Maximum	27.2570		27.1573	3.8400		3.7398
Range	26.0460		25.8455	3.2520		3.2517
Coefficient of determination, R^2		0.9991			0.9941	
Mean squared error		2.579			0.00883	
Intercept		0.0935			0.0372	

(continued)

Table 4.3 (continued)

	SVR after parameter optimization					
	Eu			$Stk_{50} \times 10^3$		
	x		y	x		y
Slope		0.9944			1.0036	
The SVR regularization parameter, C		13502			283.151	
The kernel parameter, γ		0.154			0.041	
	Kriging (KG)					
	Eu			$Stk_{50} \times 10^3$		
	x		y	x		y
Average	7.2436		7.2436	1.9496		1.9496
Standard deviation	6.9465		6.9465	0.9211		0.9211
Minimum	1.2110		1.2110	0.5880		0.5880
Maximum	27.2570		27.2570	3.8400		3.8400
Range	26.0460		26.0460	3.2520		3.2520
Coefficient of determination, R^2 [21]		0.876			0.999	
Mean squared error		5.506			0.001	
Intercept		0.0			0.0	
Slope		1.0			1.0	

[a] x is the input to the SVR and y is the predicted value. Both x and y represent the Euler number and Stokes number.

Note that the values of *the average, the standard deviation, the minimum, the maximum and the range* have been calculated using all data for training the meta model. Consequently, for Kriging interpolation metamodel, these values are identical for x and y columns.

The average coefficient of determination R^2 using cross validation can be calculated as $R^2 = \frac{1}{K}\sum_{j=1}^{K}\left(1 - \frac{SS_{\mathrm{res}_j}}{SS_{\mathrm{tot}_j}}\right)$ where K is the number of folds, $SS_{\mathrm{res}_j} = \sum_{i=1}^{Nt_j}(y_i - \widehat{y_i})^2$ is the sum of squares of residuals, $SS_{\mathrm{tot}_j} = \sum_{i=1}^{Nt_j}(y_i - \bar{y}_j)^2$ is the total sum of squares, Nt_j is the number of testing points in fold j, y_i represents the actual values, $\bar{y}_j = \frac{1}{Nt_j}\sum_{i=1}^{Nt_j} y_i$ is the mean of the observed data in fold j and $\widehat{y_i}$ represent the predicted values

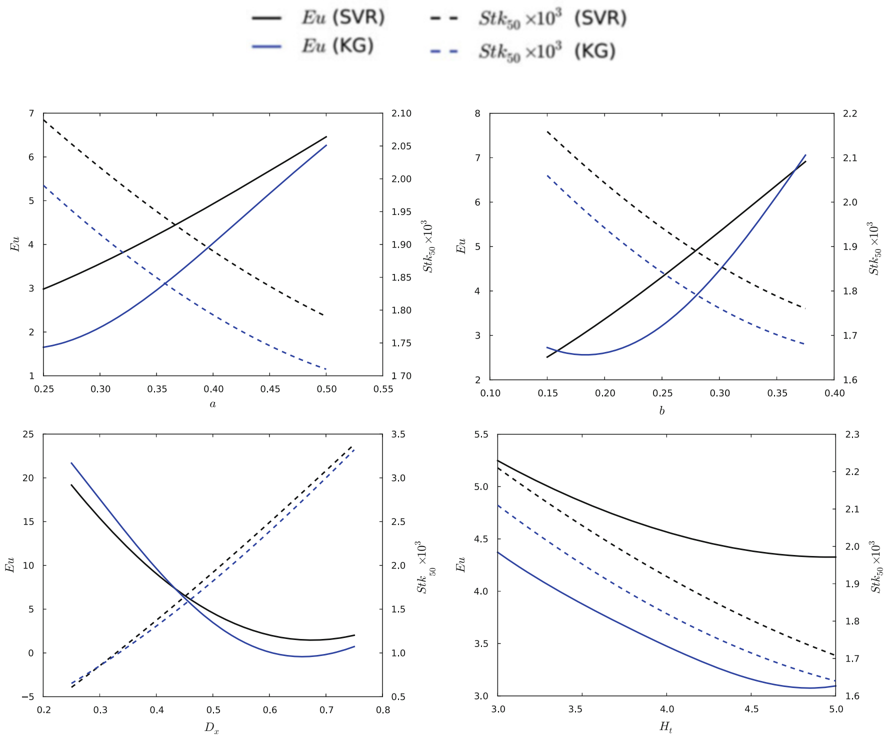


Fig. 4.3 Comparison between the effect of each geometrical parameter on the cyclone performance parameters using SVR (*black lines*) and KG (*blue lines*)

variation of the cyclone performance with the total height (H_t), the reduction in the Euler number (pressure drop) stops after $H_t = 4.625$ whereas the enhancement in the collection efficiency (reduction in the cut-off diameter) continue with lengthen the cyclone.

4.3.3 Geometry Optimization

Two optimization techniques have been applied to obtain new geometrical ratio set, namely the Nelder–Mead technique [22] and the genetic algorithms [6, 23]. Table 4.4 lists the new generated cyclone geometry ratios for minimum pressure drop (Euler number).

For single objective and one parameter optimization using the Nelder–Mead technique from `Scipy`. The total cyclone height H_t is optimized for minimum Euler number. The optimum value of H_t is 4.694, i.e., $h_c = 3.194$, where $h = 1.5$ which results in $Eu = 3.432$.

Table 4.4 Optimum geometrical ratios for minimum Euler number

	a	b	D_x	H_t	Eu
Optimum cyclone total height (Nelder–Mead technique)	0.25	0.15	0.5	4.694	3.432
Optimization of the four factors (Nelder–Mead technique)	0.499	0.15	0.658	3.0	0.667

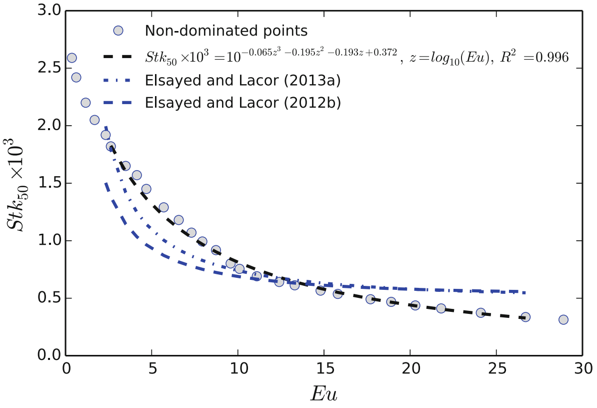


Fig. 4.4 Pareto front for NSGA-II optimization with polynomial fit

Since the cyclone performance has two major performance indices (the Euler and the Stokes numbers) a multi-objective optimization process is needed for optimum cyclone performance. In this study, NSGA-II [7, 24] available from `deep` (evolutionary toolbox for python) [25] has been used to obtain the Pareto front shown in Fig. 4.4.

The obtained Pareto front has been used to fit a correlation between the two performance parameters. The application of the `polyfit` function from `Numpy` Python package [26] results in the following correlation

$$Stk_{50} \times 10^3 = 10^{-0.065z^3 - 0.195z^2 - 0.193z + 0.372}$$

where $z = log_{10}(Eu)$. It is worth to mention that the accuracy of the new third order correlation is $R^2 = 0.996$ which is superior that proposed by the authors in a previous article [6]. This correlation can be used to predict the Stokes number (the cut-off diameter) given the pressure drop (Euler number).

Conclusions

In order to accurately predict the complex nonlinear relationships between the cyclone performance parameters and its geometrical dimensions, the support vector machines approach has been used and compared with the Kriging surrogate. Two SVR surrogate models have been trained and tested by 33 CFD datasets. The result demonstrates that SVR can offer an alternative and

(continued)

powerful approach to model the performance parameters. The SVR model parameters have been optimized to obtain the most accurate results from the cross validation steps. The parameters optimization has been optimized using the Simulated annealing technique. SVR (with optimized parameters) can offer an alternative and powerful approach to model the performance parameters better than Kriging. The SVR surrogates used to study the effect of the four geometrical parameters on the cyclone performance. The genetic algorithms optimization technique has been used to obtain a new geometrical ratio for minimum Euler number and for minimum Euler and Stokes numbers. The new cyclones over-perform the standard Stairmand design performance. A new correlation between the Stokes number and Euler number is provided which is more accurate than the existing correlations in the literature.

References

1. Zhao B (2009) Modeling pressure drop coefficient for cyclone separators a support vector machine approach. Chem Eng Sci 64:4131–4136
2. Elsayed K, Lacor C (2011) The effect of cyclone inlet dimensions on the flow pattern and performance. Appl Math Model 35(4):1952–1968
3. Elsayed K, Lacor C (2010) Optimization of the cyclone separator geometry for minimum pressure drop using mathematical models and CFD simulations. Chem Eng Sci 65(22):6048–6058
4. Elsayed K, Lacor C (2011) Numerical modeling of the flow field and performance in cyclones of different cone-tip diameters. Comput Fluids 51(1):48–59
5. Elsayed K, Lacor C (2011) Modeling, analysis and optimization of aircyclones using artificial neural network, response surface methodology and CFD simulation approaches. Powder Technol 212(1):115–133
6. Elsayed K, Lacor C (2012) Modeling and pareto optimization of gas cyclone separator performance using RBF type artificial neural networks and genetic algorithms. Powder Technol 217:84–99
7. Elsayed K, Lacor C (2013) CFD modeling and multi-objective optimization of cyclone geometry using desirability function, artificial neural networks and genetic algorithms. Appl Math Model 37(8):5680–5704
8. Elsayed K, Lacor C (2013) The effect of cyclone vortex finder dimensions on the flow pattern and performance using LES. Comput Fluids 71:224–239
9. Elsayed K, Lacor C (2014) CFD-based analysis and optimization of gas cyclones performance. In: International energy and environment foundation (IEEF), Chap 8, pp 223–276. ISBN 13: 978-1-49487-575-6
10. Hoffmann AC, Stein LE (2008) Gas cyclones and swirl tubes: principle, design and operation, 2nd edn. Springer, Berlin
11. Derksen JJ, Sundaresan S, van den Akker HEA (2006) Simulation of mass-loading effects in gas–solid cyclone separators. Powder Technol 163:59–68
12. Xiang R, Park SH, Lee KW (2001) Effects of cone dimension on cyclone performance. J Aerosol Sci 32(4):549–561
13. Stairmand CJ (1951) The design and performance of cyclone separators. Ind Eng Chem 29:356–383

14. Ramachandran G, Leith D, Dirgo J, Feldman H (1991) Cyclone optimization based on a new empirical model for pressure drop. Aerosol Sci Technol 15:135–148
15. Liu S, Xu L, Li D, Li Q, Jiang Y, Tai H, Zeng L (2013) Prediction of dissolved oxygen content in river crab culture based on least squares support vector regression optimized by improved particle swarm optimization. Comput Electron Agric 95:82–91
16. Suykens JAK, Gestel TV, Brabanter JD, Moor BD, Vandewalle J (2002) Least squares support vector machines. World Scientific, Singapore
17. Baylar A, Hanbay D, Batan M (2009) Application of least square support vector machines in the prediction of aeration performance of plunging overfall jets from weirs. Expert Syst Appl 36(4):8368–8374
18. Singh KP, Basant N, Gupta S, Sinha S (2011) Support vector machines in water quality management: a case study. Anal Chim Acta 703:152–162
19. Samui P (2011) Application of least square support vector machine (lssvm) for determination of evaporation losses in reservoirs. Engineering 3(4):431–434
20. Elsayed K, Lacor C (2013) Comparison of Kriging, RBFNN, RBF and polynomial regression surrogates in design optimization. In: Eleventh international conference of fluid dynamics (ICFD11), Alexandria
21. Elsayed K, Lacor C (2014) Robust parameter design optimization using Kriging, RBF and RBFNN with gradient-based and evolutionary optimization techniques. Appl Math Comput 236:325–344
22. Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7(4):308–313
23. Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor
24. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197
25. Fortin F-A, De Rainville F-M, Gardner M-A, Parizeau M, Gagne C (2012) Deap: evolutionary algorithms made easy. J Mach Learn Res 13:2171–2175
26. Bressert E (2012) SciPy and NumPy: an overview for developers. O'Reilly Media, Sebastopol