APPROXIMATION METHODS AND SELF ORGANIZING MAP TECHINQUES FOR MDO PROBLEMS

V. Pediroda[1] and C. Poloni[1]

Dipartimento di Ingegneria Meccanica, Università di Trieste, via Valerio 10, 34100 Trieste, Italy

Contents

1 Introduction

Multidisciplinary Design Optimization (MDO) is getting more and more import, especially in the aerospace community. The AIAA Association (American Institute of Aeronautics and Astronautics) has organized several sessions dedicated to the MDO (last session [1]) and recently the First Session of MDO for specialist [2]. Consequently the development of numerical methodologies, to solve these problems, is increasing in importance, to help the industry during the phases of a complex design. It seems useful to remark that the designs, in particular in the aeronautics field, are extremely complex, because of the physic model and for huge number of input and output parameters.
One important aspect in the industrial design is the management of the uncertainties, to find solutions which are insensible to the stochastic fluctuations of the parameters. The name of this design model is *Robust Design*.
The need of Robust Design method appears in many contests, especially in the Multi Disciplinary Design. In fact it is possible to find uncertainties in many different cases; during the preliminary design process, the exact value of some input parameters could be known, or the input parameters could change in the next design phases. Consequently the aim is to look for a solution as less dependent as possible on the unknown parameters. Other concerns are to find out solutions which are insensitive to the tolerance manufacturing parameters, to fluctuations in the operative conditions or numerical fluctuations in the high fidelity simulation models.
The present paper shows a new optimization method that look for solutions which are insensitive to fluctuations, any source they are caused by. Starting from the statistical definition of stability, the method, based on a multi-objective approach, in particolar Game Theory, finds good solutions for stability and performances.
Robust Design methodology, together with a metamodel approach is proposed, to reduce the total number of high fidelity analysis needed by the method. A new adaptive methodology based on Kriging method is presented, in order to minimize the statistical unknown error between the real values obtained by high fidelity analysis and the extrapolated ones.
Finally a new important aspect of the multi disciplinary optimization is presented: the visualisation of the interactions between the different design parameters. It is well know that in the complex projects the designers have to control a huge number of different parameters of the system (geometry parameters, operating parameters, input/output parameters). Consequently a visualisation model which represents in efficient way the different interactions is useful to understand the final behaviour of the exanimate system. In this work the Self Organized Maps are presented and used to visualise the results of a aeronautic Robust Design.

2 The idea of Robust Design in Aeronautics

The study of the uncertainties in engineering begins with Taguchi [3], who codified the methodology for the quality engineering. Taguchi divides the design in three different phases: the fists one, called system design, determinates the most feasible region for the following numerical optimizations, the second phase, called robust design, determinate the optimal parameter for maximizing the final quality of the considered system, and in the third final phase, called tolerance design, is performed one parameter tuning to reach the best possible final solution.
The necessity to study uncertainty is well known in aeronautics; in fact it is possible to cite the definition of uncertainty given on AIAA Guideline [4]:

Definition 1.2 Uncertainty: A potential deficiency in any phase or activity of the modelling process that is due to lack of knowledge.

Notice that the uncertainty is defined how a lack of knowledge, which obviously leads to need a different approach for studying the model.
From a numerical point of view the study of a model affected by uncertainties could be defined as:

$f: \mathrm{A} \times \mathrm{B} \rightarrow \Re$ where

- $a \in \mathrm{A}$ represents the design parameters chosen by the designers
- $b \in \mathrm{B}$ represents the input parameters permeated by uncertainties consequently not controllable by designers.

In [5] the common uncertainties (parameter *b*) for external aerodynamic are well described (more precisely in the case of two dimensional airfoil design).

1. Uncertainties on geometry parameters due to manufacturing tolerance ε which modifies the geometry parameters in a-ε. This situation is deeply explained in [6], where an airfoil with minimum drag over geometrical uncertainties is designed.
2. Uncertainties on operative conditions (design point): usulally this case is studied with fluctuations on the free stream Mach number $[M_{min}, M_{max}]$. For important references see [7,8]
.

In [9] it is well demonstrated why the study of fluctuations is of primary importance in external aeronautics. It is shown how minimizing the drag coefficient of an airfoil with fixed operative conditions, in particular the free Mach number, the final solution has good performance at the design point but poor off-design characteristics (Fig. 1); this concept is known as *over-optimization.*

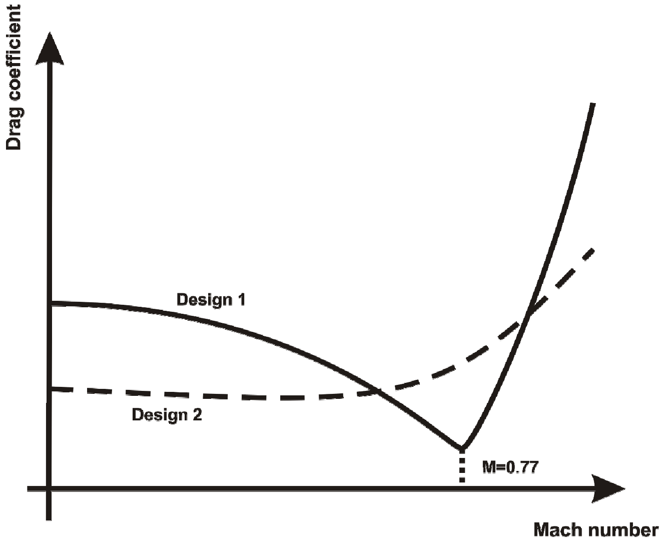

Figure 1: Drag profile for stable (Design 2) e not stable (Design 1) solution respect to Mach number

This behaviour becomes more evident for supercritical airfoils where the relationship between drag and free stream velocity is nonlinear because of the fluctuations of the shock waves position on the airfoil surface.
Consequently the possibility to determine solutions with good performances over a range of operative conditions appears attractive, also to avoid sudden changes in the behaviour of the system; it is interesting to remember that a stable behaviour minimises the operative risk of the system.
Many numerical methods have been developed to optimise a system under uncertainties, in particular in the case of fluctuations of the operative conditions [7, 8, 10, 11, 12].
In [7] a methodology direct based on multi-point optimization has been proposed, using a sum-weighted formulation; in the examined example, minimization of the drag coefficient of an lift-constrained airfoil with uncertainties on cruise Mach number is performed. The proposed formulation is:

$$\min_{\alpha, d \in D} \sum_{i=1}^{n} w_i c_d(d, \alpha_i, M_i) \tag{1}$$

subjected to

$$c_l(d, \alpha_i, M_i) \geq c_l^* \quad \text{for } 1 \leq i \leq n \tag{2}$$

where w_i are arbitrary weights, d is a set of geometric design variables that define the airfoil, C_d and C_l are the drag and lift coefficient defined as function of the free steam Mach number M_i and the angle of attack α_i which can fluctuate around the design point values, and C_l^* is the required lift. The problem of the Eq.1 is that the final result depends on the choice of the weights w_i, too arbitrary to define.
In [10] a new concept is introduced for the Robust Design, and an approach different from the multi-point optimization is used. The innovative idea is the formulation of a risk ρ to minimize:

$$\min \rho = \int_{M_\infty} C_d(d, M_\infty) f(M_\infty) dM_\infty \tag{3}$$

The term $f(M_\infty)$ appears in Eq. 3 that is the probability density of the cruise Mach number. The risk (from the Bayesian theory) represents the mean value of the drag coefficient inside the fluctuations of the operative conditions. The author proposed for the solution of the integral (Eq. 3) the use of a Taylor series of second order.

In [11], to avoid the arbitrariness of the Robust Design formulation in [7] and [10], has been propose an interesting methodology. This one iteratively modifies the Mach number in the Eq. 1 to calculate the integral Eq. 3 using the trapezoid rule.

A new and interesting approach is proposed in [8] where the authors have demonstrated that the Robust Design problem has to be solved using a Multi Objective Optimization Approach. Starting from the definition of stability, the numerical definition of the problem becomes:

$$\min_{\mathrm{D},\alpha(\mathrm{M})} (E(c_d), \sigma^2(c_d)) \tag{4}$$

subjected to

$$c_l(D, \alpha(M), M) = c_l^* \quad \text{for} \quad M \in \Omega \tag{5}$$

The mean and variance C_d are defined as follows:

$$E(c_d) = \int_{M_{\min}}^{M_{\max}} c_d(D, \alpha, M) p(M) dM \tag{6}$$

$$\sigma^2(c_d) = \int_{M_{\min}}^{M_{\max}} (c_d(D, \alpha, M) - E(c_d))^2 p(M) dM \tag{7}$$

where $p(M)$ is the probability density function of the Mach Number defined in the interval $M_{min}<M<M_{max}$. This Robust Design formulation (Eqs. 4-7) gives the possibility to determine two directions in the optimization: by the variance of the drag coefficient, it is possible to minimize the off-design performance degradation (fig.2, design at the bottom); on the other hand, by the optimization of the performance mean value (in this case drag coefficient), the performances will be privileged (fig.2, design at the top). Let has notice that the Robust Design formulation of Eq.4 is based on a Multi Objective approach, so the final result will be the Pareto frontier, i.e. the set of the best compromise solutions between the objective functions (fig.2, all the design of the Pareto frontier). The difficulty in this kind of approach could be find the number of high fidelity analysis needed to find the Pareto frontier; for this reason an alternative methodology is proposed using a descend direction that could reduce drag simultaneously and proportionally over the given range of Mach number.

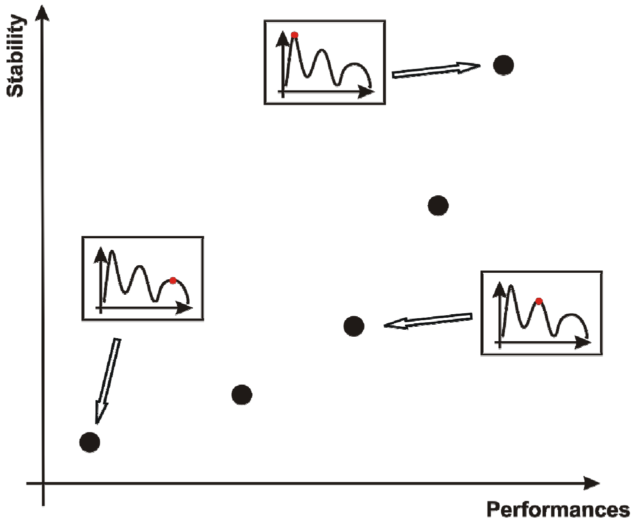

Figure 2: Pareto Frontier obtained by Robust Design Optimization (Performances vs. Stability Degradation)

Interesting is also the approach [12] where a weighted-sum function (similar to Eq. 1) is proposed. The author to avoid the arbitrariness of this formulation proposed an adaptive methodology to determinate the weights:

$$w_i^{new} = w_i^{old} + c\left(\frac{c_d^{\,i}}{\sum_{i=1}^{N} c_d^{\,i}} - \frac{1}{N}\right) \tag{8}$$

To avoid the high computational cost for solving the Robust Design problem, Trosset [5] proposed new methods based on the statistical decision theory, in particular the Bayesian formulation. From the Taguchi method for quality engineering, he demonstrated how it is possible to replace a computationally expensive objective function with a not-expensive surrogate numerical model, using the DACE methodology developed by Sachs and Welch [16].

Finally it is important to remember that one interesting field of application for Robust Design is the preliminary design of complex engineering systems, that involve multi-disciplinary subsystems [13, 14, 15]. In this case, the aim of Robust Design is the reduction of the effects on the performance of the whole system, produced by the decisions taken in one subsystem design (Fig. 3).

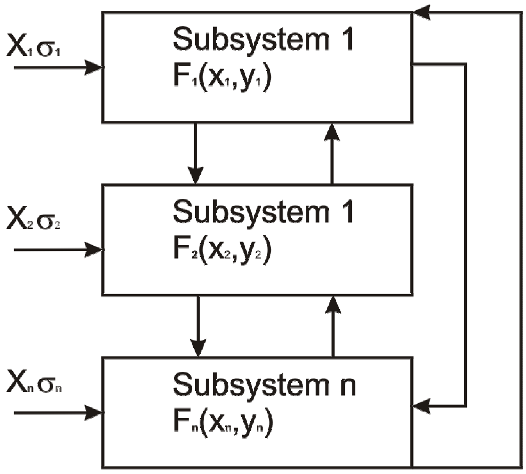

Figure 3 Coupled Multi Disciplinary System with uncertainties

2.1 Why we need a Multi Objective Approach

In this work a new method for Robust Design optimization is presented. The main idea is to use a multi objective approach to reach the best possible compromise between performance and stability of the design. Referring to Fig. 4, the function has an absolute extreme and a relative one respectively corresponding to the coordinates *x1* and *x2*; in this case the uncertainties are represented by the tolerance Δ in the input parameter *x* (the same case of manufacturing tolerance). Obviously a standard optimization, without fluctuations, would find out the point *x1*, that is the absolute maxima but with poor stability. In the case of Robust Design optimization (considering tolerance Δ) two different objectives have to be considered: mean performances and stability of the solutions, according to the ideas presented in [8]. Considering the mean performance inside the tolerance Δ, the best configuration would be represented by the point *x1*, since the mean value of the function is the highest. But for the stability, which corresponds to an evaluation of the variance of the function *f(x)* inside the field Δ (Eq. 7), the best configuration is represented by the point *x2*, because the function is characterized by a lower variability inside the tolerance around to the point *x2*.

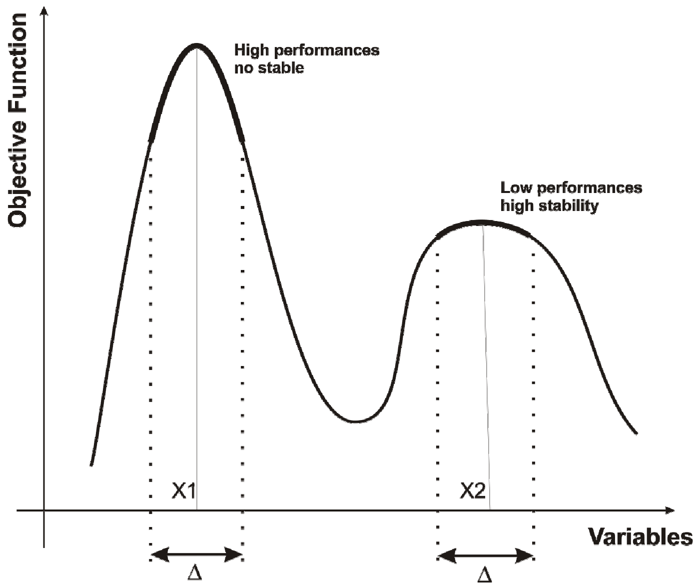

Figure 4: Function with two different extremes: x1 absolute no stable extreme, x2 relative stable extreme

Consequently it is interesting to observe that when Robust Design optimization is performed, it is possible that the more stable region doesn't correspond to the more performing one. So to perform an optimization under fluctuations the best way is to define two different objectives for every function to optimize: its mean value of the function and its variance. In mathematical term it is:

$$
\begin{aligned}
& f : \Re^n \Rightarrow \Re^m \\
& \max \quad E(f_i) = \int_q f_i(x,q)\, p(q)\, dq \\
& \min \quad \sigma^2(f_i) = \int_q \left[f_i(x,q) - E(f_i) \right]^2 p(q)\, dq
\end{aligned}
\tag{9}
$$

where f is the multi objective (in more general terms) function to be maximized and q are the uncertainties parameters, modeled by the probability density function *p(q).*
In this way the problem of an optimization under uncertainties becomes a Multi Objective Optimization problem where the objectives are the stability and the performance; to solve this problem we need to adopt the Game Theory (see chapter 2) that is the best methodology to solve a real multi objective problem without using a weighted function as:

$$\max \quad f_w = w_1 \bar{f} + w_2 \sigma_f \tag{10}$$

In fact it is tricky to assign a value to the weights w_i, and this is the reason because it is better to refer to Game Theory approach. It is interesting to note that after the optimization phase, using a Pareto Frontier approach, the designer does not get only one solution but a set of solutions (Pareto Frontier) that represents the best possible compromise between the objectives. An example of a Pareto Front for a Robust Design Optimization could be observed in Fig. 2; inside the Pareto frontier it is possible to choose different compromises between performance and stability, with more flexibility than a standard optimization, where the solution is unique. It is important to underline that it is possible to face a wide range of problems with the Robust Design approach (small manufacturing process errors, fluctuations in the operative conditions, unknown input parameters, etc.). The method is also extendible to more than one function to optimize, for example it is possible to improve the lift and drag of an airfoil with fluctuations in the flight speed, without the need of a weighted function to tie the two different performances.

2.2 Numerical Example

To understand better the Multi Objective approach to Robust Design optimization, could be useful observe an example:

$$\begin{aligned}
&\max \quad f(x,y) = \sum_{i=1}^{3} h_i e^{-\frac{a_i}{s_i}} \\
&\text{where:} \\
&s_1 = 0.5 \quad h_1 = 5 \quad a_1 = (x-1.5)^2 + (y-1.5)^2 \\
&s_2 = 1 \quad h_2 = 4 \quad a_2 = (x+1.5)^2 + (y-1.5)^2 \\
&s_3 = 2 \quad h_3 = 3 \quad a_3 = (x-0.5)^2 + (y+1.2)^2 \\
&\text{with} \quad -2 \le x \le 2; \quad -2 \le y \le 2 \\
&\text{and uncertainties:} \quad \sigma_x = 0.1 \quad \sigma_y = 0.1 \quad \text{(uniform)}
\end{aligned} \tag{11}$$

In this example we have to maximize the function f(x,y) with uncertainties on the variable definition ($\sigma_{x,y}$ with uniform probability density function). Following the Eq. 9, using a discrete formulation for the mean value and the variance, the problem becomes:

$$\begin{aligned}
&\max \quad \bar{f}(x,y) = \frac{\sum_{n=1}^{N} f_n}{N} \\
&\min \quad \sigma^2(x,y) = \sum_{n=1}^{N} \frac{\left(f_n - \bar{f}\right)^2}{N-1}
\end{aligned} \tag{12}$$

where the N points are distributed uniformly on the intervals $[x-\sigma_x, x+\sigma_x]$.and $[y-\sigma_y, y+\sigma_y]$.

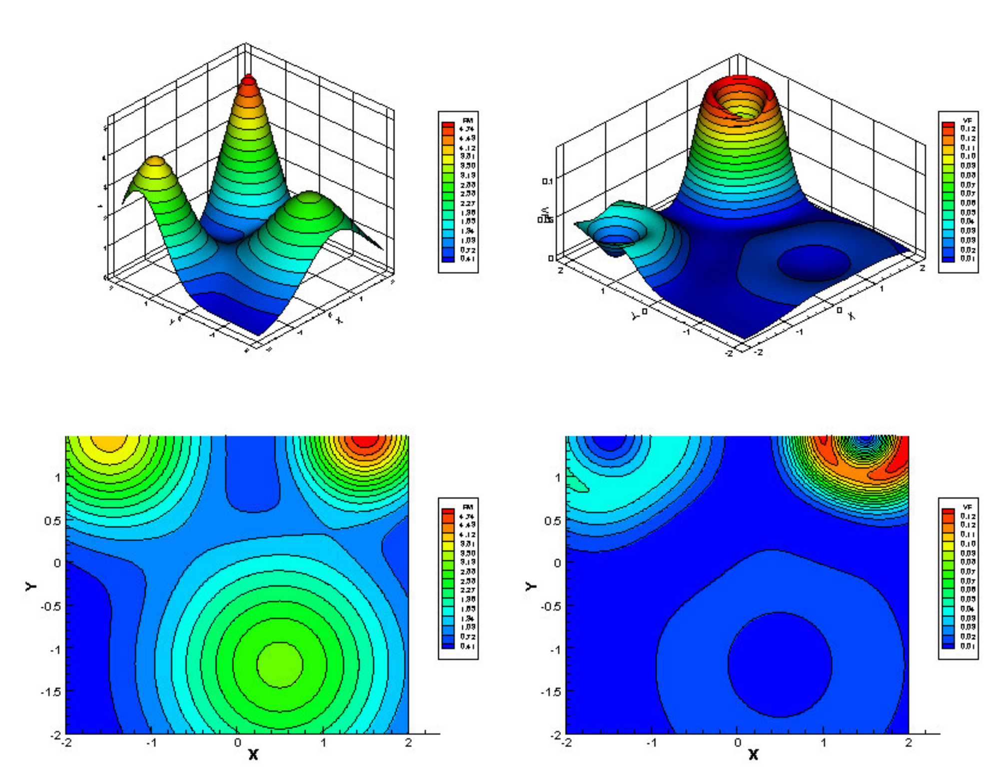

Figure 5: Plot 3D and contours of the mean value of the function (performance), left; plot of the variance of the function (stability, right)

In Fig. 5 it is possible to observe graphically the Multi Objective Robust Design problem (Eq. 11). The same remarks done in [17] for a one-variable problem is valid: the regions for local maxima of the function, that is the solutions with higher mean performances, usually corresponds to the regions where more stable solution are present. But if we plot the Pareto Frontier (Fig. 6)) some more considerations can be done for this problem.

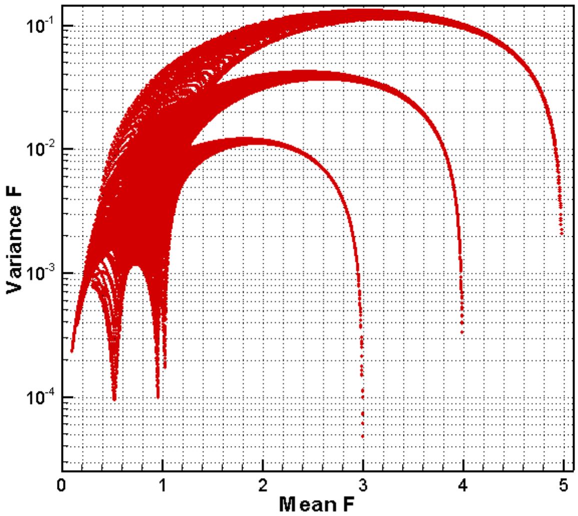

Figure 6: Pareto Frontier for the analytical Robust Design Test Case

The Pareto Frontier contains three points, which correspond to the picks of the original function: considering this simple analytical test case, we can argue that an effective Multi Objective Approach is needed as the solution is not unique. It is important to observe how in this kind of approach a robust optimization algorithm is necessary: normally the regions characterized by good stability are wide, but the picks for performance and the picks for stability are very restricted (the three picks in the analytical function); for these reasons if a weighted-sum form or a not robust optimization algorithm are adopted, the final solutions will not be characterized by optimal features.

3 Game Theory on Robust Design

Game Strategies, defined mathematically by J.Nash [18], have found their first applications in economics, in particular to solve the problems concerning the decisions that have some effects on different and often competitive fields.
These strategies may however be adopted also in the industrial design, and in particular they can be combined with evolutionary algorithms, in order to optimize a product following several criteria and contrasting objectives.
We shortly describe the basic formulation of two typologies of Game Strategies (co-operative and competitive), and then we will show how it is possible to implement practically these algorithms to solve multi-objective optimization cases.
In a problem of minimization of two functions $f_A(x,y)$ and $f_B(x,y)$, we define the variables space $(x,y)\in X \times Y$ as the set of rational strategies. Thus, we decompose the variable space between two "players", called A and B, that are in charge respectively of the variable space X and Y; it follows that each pair $(x,y)\in X \times Y$ represents a combination of the strategies played by the two players.

The Pareto front may be seen as the result of a co-operative game, in which the two players A and B try to minimize both the two functions; in other words, each strategy played by the players is evaluated by the fitness of the two functions.
Not a single solution is found, but instead a set of solutions, that is called Pareto front; this set is characterized by the fact that there does not exist a solution such that both the two functions have a better fitness of any point of the front. In mathematical terms:
$(x^*,y^*) \in X \times Y \in$ to Pareto front if and only if:

$$\exists \qquad (x',y') \in X \times Y \ : \begin{cases} f_A(x',y') \leq f_A(x^*,y^*) \\ f_B(x',y') \leq f_B(x^*,y^*) \end{cases} \tag{13}$$

These definitions can of course be generalised in the case of n functions f_i.
In a competitive game, the two players act following different objectives; in particular, player A have to choice his strategies in order to minimise the function f_A, while player B have to minimise the function f_B.
Of course, as generally both the functions depend on the two domains, the strategies of one player influences the choices of the other one. The two players act simultaneously until an equilibrium is found (Nash equilibrium point): in this case, each player has minimised his own function with a common pair of strategies.
In mathematical terms:

$(x^*,y^*) \in X \times Y$ is a *Nash equilibrium* if and only if:

$$\begin{cases} f_A(x^*,y^*) = \inf f_A(x,y^*) & x \in X \\ f_B(x^*,y^*) = \inf f_B(x^*,y) & y \in Y \end{cases} \tag{14}$$

3.1 Multi Objective Game Theory Adaptive Algorithms

From the definitions of the previous chapter, we can affirm that MOGA (multi-objective genetic algorithm) is an algorithm that implements practically a co-operative game, since it searches a Pareto front as trade-off of the contrasting objectives and all the variables co-operate in the optimization of all the objectives.
To implement a competitive game the procedure is more complex, since we need to define an algorithm that decomposes the variable space, that assigns to each part of the decomposed space (that becomes a player own domain) the correspondent objective and that provides a mono-objective optimization algorithm to each player. From different tests considered [19], it seems that the most efficient algorithm to be run by each player is the Nelder and Mead Downhill Simplex [20]: for this reason we will refer from now to any competitive game algorithm as to Nash-Simplex algorithm.
After a certain number of Simplex iterations, each player finds the best configuration (and set of variables) for its objective, and then the search continues with a new step, for which each player starts a new Simplex sharing the variables found by the other players.
As it is possible to see and previously pointed out by Periaux and Wang [21], the problem of the variables space decomposition is very important, since it influences the results of the Nash

equilibrium point and thus the optimization results. For this reason, there are different approaches that can be followed:

1) it is possible to decide an initial variable decomposition and continue the competitive game without changing it until the convergence is achieved: this is the case of a not-adaptive Nash-Simplex algorithm;

2) it is possible to implement an adaptive Nash-Simplex-tStudent algorithm, like we proposed in our previous work [22], using statistical analysis and in particular the t-Student coefficient to decide, at the end of each player step (that is after a certain number of Simplex iterations) if a variable is statistically significant for the player to which is assigned or not; in the latter case, i.e. if the significance percentage is lower than an assigned threshold, the variable is given to another player in the following step. In other words, the significance percentage [23] expresses the probability that a variation of the objective function is really produced by a variation of the variable instead of an effect of the statistical variance around the mean value of the objective function;

3) a variation of the adaptive Nash-Simplex algorithm can be given by the adaptive Nash-Simplex-correlation matrix algorithm, that instead of using t-Student parameter to test the significance of any variable, uses the correlation matrix [23], that gives a number from –1 to 1 to express the statistical correlation between any input variable and any objective: for each variable, we consider the absolute value of the correlation with every objective, and the variable will be assigned to the objective for which the value is higher, since it denotes an higher influence from that variable (independently from the fact that the correlation is inverse or direct);

In this work we use a Multi Objective Genetic Algorithm approach to demonstrate the capability of the Robust Design Optimization in the last exhaustive example for a 2 dimensional airfoil design under uncertainties.

4 The use of local optimal Response Surfaces in Robust Design

One interesting point, which needs to be study in details, is the calculation of the Eq. 12. The two equations represent the mean value (performance) and the variance (stability) of the function to be optimized by Robust Design approach. From the equations it is possible to observe that to obtain the values of the objectives to optimize it requires more than one calculation of the function. Consequently, when a numerical simulation needs high computing resources, the application of the Multi Objective Robust Design could become inapplicable caused of to much higher time consuming. For these reasons reducing the number of total simulation is a key point to make the Robust Design useful to the industry. One efficient numerical methodology to solve this problematic is the use of the Response Surfaces.

For the application of Response Surfaces in Robust Design, [5] presents a Taylor approximation approach that in some cases is accurate. But in this paper, we present a different approach, based on statistical theory, called DACE. The advantage of this methodology is the possibility of implement an adaptive response surfaces, which tries to minimised the statistical error between the real function end the extrapolated one.

4.1 D.A.C.E. Response Surface Methodology

Originally developed and used in mining engineering and geo-statistics data, the Kriging method is an approach for curve fitting and response surface approximation. In the 1980s, some statisticians have developed Design and Analysis of Computer Experiments (D.A.C.E.) for deterministic computer-generated data based on the Kriging method [24] [25]. The Kriging method used in this study is based on the D.A.C.E. approach.

Suppose we have evaluated a deterministic function of k variables at n points. We denote the i-sampled point by $x^i=(x_1^i,\dots,x_k^i)$ and the associated function value by $y^i=y(x^i)$, for $i=1,..,n$. The Kriging (D.A.C.E.) technique is based on the following stochastic process model:

$$d(x^i, x^j) = \sum_{h=1}^{k} \theta_h \left| x_h^i - x_h^j \right|^{P_h} , \quad \theta_h \geq 0, P_h \in [1,2] \tag{15}$$

$$Corr[\varepsilon(x^i), \varepsilon(x^j)] = \exp[-d(x^i, x^j)] \tag{16}$$

$$y(x^i) = \mu + \varepsilon(x^i) \quad , \quad (i = 1,...n) \tag{17}$$

The eq.15 is the weighted distance formula between the sample points x^i and x^j, and eq.16 is the correlation between the errors corresponding to the points x^i and x^j.

Eq.17 is the model we use in the stochastic process approach: μ is the mean of the stochastic process, $\varepsilon(x^i)$ is defined by a Gaussian distribution of Normal type $(0,\sigma^2)$; the latter term is the result of a stationary Gaussian random function that creates a localized deviation from the global model [14]. The parameter θ_h in the distance formula (eq.15) can be interpreted as measuring the importance or "activity" of the variable x_h. The exponent p_h is related to the smoothness of the function in coordinate direction h, with p_h=2 corresponding to most smooth functions. The stochastic process model in Eqs.15-17 is essentially a generalized least squares (GLS) model [26] with a simple set of regressors (just a constant term) and a special correlation matrix that has unknown parameters and depends upon distances between the sampled point.

The Kriging approximation presented by Schonlau [27] uses the best linear unbiased predictor (BLUP) of y at the point at which we are predicting, x^*. Let r denote the n-vector of correlations between the error term at x^* and the error at the previously sampled points. That is, element i of r is $r_i(x^*)=Corr[\varepsilon(x^i), \varepsilon(x^i)]$, computed using the formula for the correlation function in Eqs. 15 and 16. The estimated model of Eq.17 can be expressed by the BLUP of $y(x^*)$:

$$y(x^*) = \hat{\mu} + r^T R^{-1} (y - I\hat{\mu}) \tag{18}$$

where $y=(y^1,\dots y^n)^T$ denote the n-vector of observed function values, R denotes the $n\times n$ matrix whose (i,j) entry is $Corr[\varepsilon(x^i), \varepsilon(x^j)]$, and I denotes an n-vector of ones. The value for μ is estimated using the generalized Least Squares method as:

$$\mu = (I^T R^{-1} 1)^{-1} I^T R^{-1} y \tag{19}$$

The estimation of θ_h and p_h and hence an estimation of the correlation matrix are obtained by the maximization of a Likelihood Function [24].
The mean squared error (MSE) of $y(x)$ can be derived as:

$$s^2 = \sigma^2[I - r^T R^{-1} r] + \frac{(I - I^T R^{-1} r)^2}{I^T R^{-1} I} \quad (20)$$

Eq.20 provides an estimation of the variance of the stochastic process component of the Kriging approximation.
Earlier studies imply that including the parameter p_h as part of maximum likelihood estimation doesn't help to improve very much the Kriging approximation, thus in the current study $p_h=2$ is used for all the design variables.

4.2 Adaptive DACE

To initialize an extrapolation, we require a systematic way of selecting the set of inputs (called Design Of Experiments, or DOE) at which to perform a computational analysis. One common choice for generating experimental design for computational experiments is the Latin Hypercube [28]. Instead of using this technique we propose an adaptive arrangement of the initial set of samples (data base) exploiting the value of the MSE (eq.20) The value of MSE depends on the correlation of the landscape as well as on the local density of points.

In particular, we consider the behaviour of RMSE (Root Mean Squared Error): the RMSE indicate the accuracy of the prediction and it assumes low values corresponding to the neighbourhood of the samples points. It is possible to understand that the extrapolation is more precise in regions with high point density. We define the function IEAN (Index of Absolute Error Normalized) as follows:

$$\text{IEAN}=|(y(x)-Y_{MIN})/(Y_{MAX}-Y_{MIN})| + RMSE_{Y(x)}/(RMSE_{max}-RMSE_{min}) \quad (21)$$

Eq. 21 represents the index we use to set the adaptive arrangement of the samples. In fact, we try to exploit the value of RMSE to understand where the extrapolation is not accurate, taking care at the same time of the extrapolated value associated (the $y(x)$ function is to be maximised, thus a high value is interesting). For example, a high value of IEAN indicates that the extrapolation is not accurate or that the function gets a high value; since these points are the most interesting, the database will be updated by the evaluation of the function in those points.

The Y_{max} and Y_{min} values are respectively the highest and lowest values of the extrapolated function, while $RMSE_{max}$ and $RMSE_{min}$ have the same meaning regarding RMSE. If $y(x)$ is to be minimised, we can substitute eq. 13 with the following one:

$$\text{IEAN}=|(y(x)-Y_{MAX})/(Y_{MAX}-Y_{MIN})| + RMSE_{Y(x)}/(RMSE_{max}-RMSE_{min}) \quad (22)$$

In any case, we apply these functions (eq.21-22) in order to add new input points in the database in following iterations, choosing the points of the range where the values of IEAN are higher.

In fig.7 we show, as example, a function *F(x)* (solid line), the function *F* extrapolated by DACE (dotted line) and by the initial data base (6 black dots), the error index IEAN (dashed line), and then the new point to be added in the database (black dot in the solid line), for which the error index is maximum.

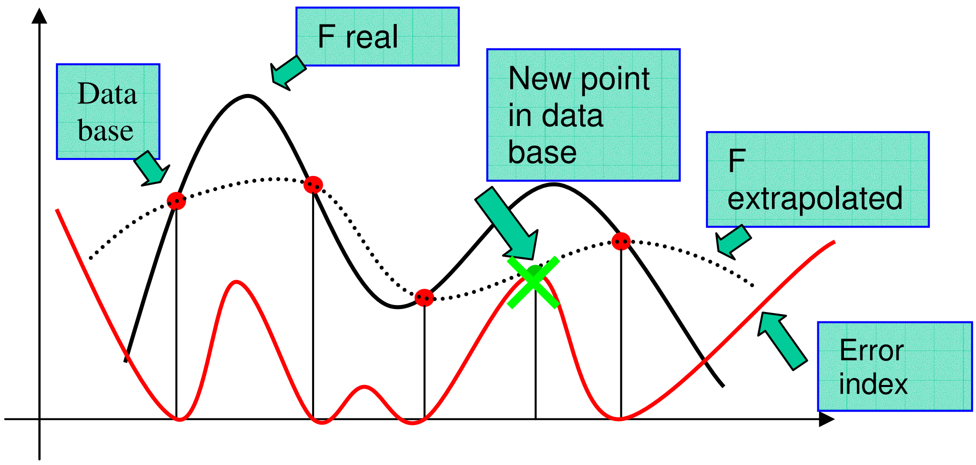

Figure 7: Definition of the new point to be added in the database for adaptive DACE approach.

4.3 Numerical Example of Adaptive Dace in Robust Design

To understand the capability of the adaptive DACE methodology in comparison with the Taylor approximation techniques we propose a simple two dimensional airfoil example, with interesting connections with the Robust Design idea.

In the common practice, in a single point airfoil optimization, the project point is fixed (e.g. angle of incidence α=2°, Mach number M=0.73). Due to not deterministic events (like gusts of wind, atmospheric turbulence, instable conditions of flight, manoeuvre inaccuracy,...), the project point can be considered slightly fluctuating, consequently it should be rational to consider a range of operating conditions instead of a single project point (e.g., α=2±0.5°, M=0.73±0.05).

The relationship between wave drag and free flow velocity is quite non linear for high subsonic design Mach numbers, and thus the position of the possible shock waves can change quickly as soon as the operating conditions slightly changes (α and M): for this reason, by means of the single design point approach it is possible to find some airfoil shapes which are advantageous corresponding to the project point (low drag resistance) but that are characterized by poor performances in the neighbourhood of it [9].

To appraise the capability of the methodologies we calculated by a Navier-Stokes code (Muflo [29]) the lift and drag coefficient for an airfoil (RAE2822) in *n* points in a regular grid inside the uniform distribution in the range of number of Mach and angle of attack (α=2±0.5°, M=0.73±0.05); this model simulates uncertainties, which normally happen on a cruise.

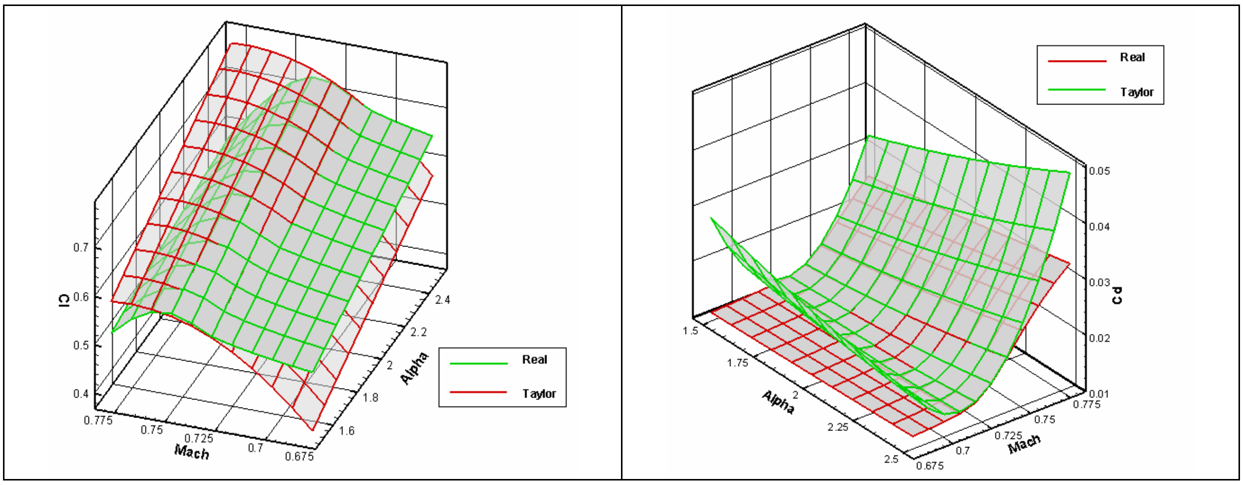

Figure 8: Comparison between the performances calculated by the high fidelity analysis code (lift on the left, drag on the right) of the RAE2822 and the response surfaces by Taylor (grey surfaces).

In Fig. 8 it is shown how the Taylor methodology approximates the real function, calculated by the high fidelity analysis code. The Taylor response surfaces are calculated by the middle point of the uncertainties range (Mach number = 0.77, Angle = 2°), closed at the second order using centred finite difference method; so 9 additional points needed for the response surfaces (lift and drag). From the figures we can observe that the behaviour of the airfoil performances is highly non linear, and the quadratic Taylor approximation is too poor for following the real surfaces. This behaviour is amplified from the presence of two uncertain parameters, in fact is well know how the Mach number brings high non linearity in the lift and drag profile, while the angle of attack is under some conditions easier to take in account. For these reasons, the use of a Response Surfaces with the capability to catch all the different behaviours of the system is strongly recommended.

Now it is important to visualize and understand how the DACE model approximates the same problem. In the previous session, we have explained an adaptive methodology to find out the new points to construct the metamodel function, so it is important to define which is the parameter used to stop the algorithm iterations. As in the optimization phase the objective functions will be the Eq. 9 (discretized as Eq. 12) we decide to use the mean value and the variance of the function as stopping iterations criteria: when the contribution of the new point in the database doesn't change the calculated mean and variance value on the response surfaces, the algorithm has get convergence.

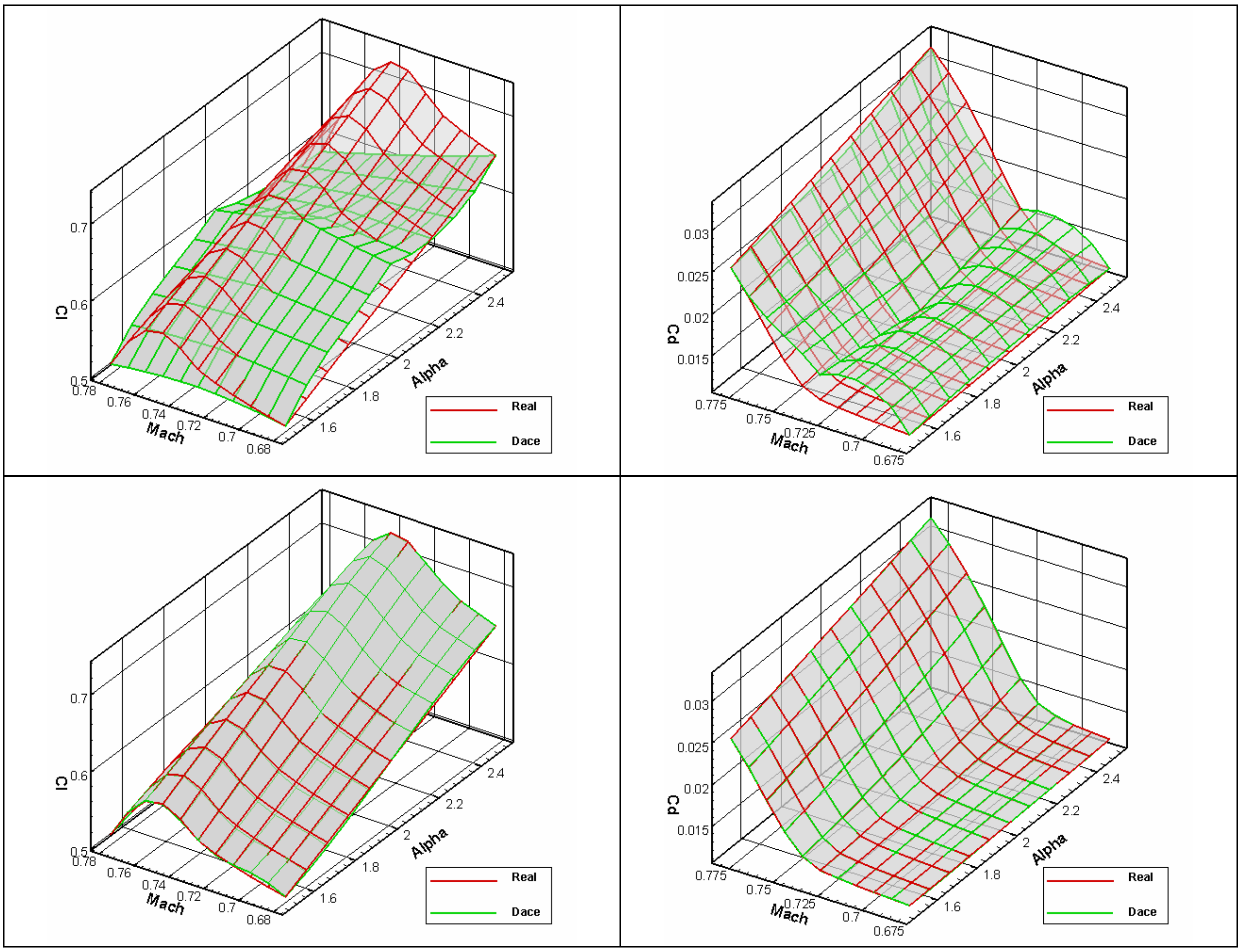

Figure 9 Convergence of the DACE surfaces (lift on the left, drag on the right); above surfaces with 5 points in the database, bottom after convergence (38 points).

In Fig. 9 we can observe how the DACE methodology reconstructs the lift and drag function; for the convergence of the surfaces (1% tolerance in two following steps for mean and variance calculation) the adaptive algorithm takes 38 new points in the database (Fig. 10). Clearly it is possible to note how the DACE surfaces cover with good quality the original surfaces, without any notable differences. This algorithm haven't limitations due to the use of a fix mathematical model, actually adapting the coefficients of the exponential weighted form (Eq.15) and finding the database points which minimize the statistical error between the original function and the extrapolated one, it is possible to use the algorithm for complex and high non linear functions. From the convergence profile, it is possible to note how after only 15/20 high fidelity analysis the error in the objective functions is limited, enabling to reach accurate results without the use of an excessive time consuming resources.

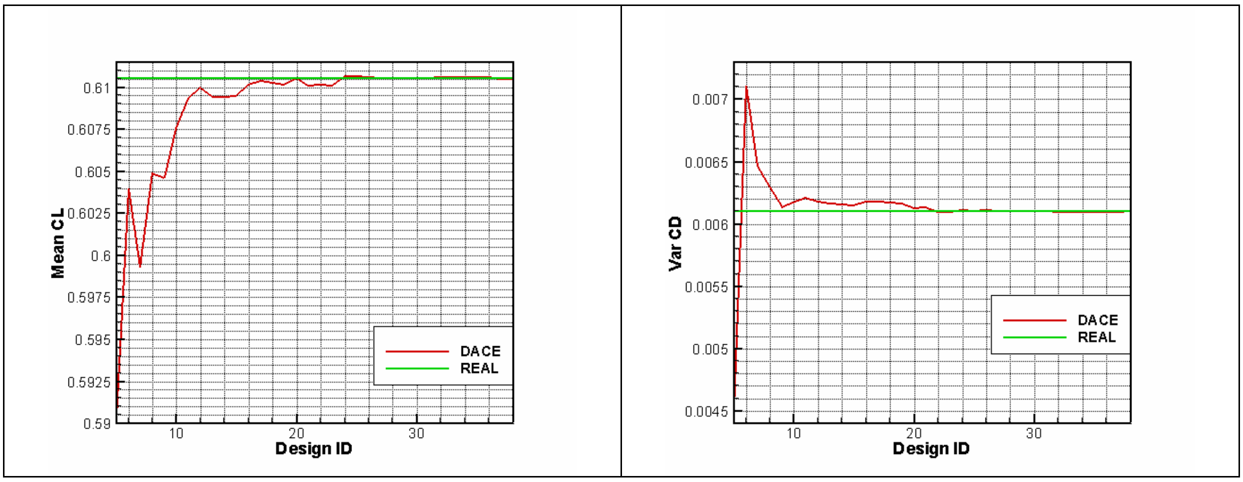

Figure 10 Convergence of the DACE surface; left: mean value of lift, right: variance of drag.

From this example it is possible to appreciate how an adaptive response surface methodology is a useful tool to decrement the number of high fidelity analysis request by the Multi Objective Robust Design Optimization. Using the DACE method, an iterative adaptive Design of Experiments is build, in order to minimize the statistical error between the real function and the extrapolated one. It is useful to remember that the advantages of the adaptive methodology increases with the number of free parameters, which in the Robust Design corresponds to the number of uncertain parameters.

5. Problem of visualisation in n-dimensional space (SOM)

The Self-Organizing Map (SOM)[31] is an unsupervised neural network algorithm that projects high-dimensional data onto a two-dimensional map. The projection preserves the topology of the data so that similar data items will be mapped to nearby locations on the map. This allows the user to identify 'clusters', i.e. large groupings of a certain type of input pattern. Further examination may then reveal what features the members of a cluster have in common. Since its invention by Finnish Professor Teuvo Kohonen in the early 1980s[32], more than 4000 research articles have been published on the algorithm[33], its visualization and applications. The maps comprehensively visualize natural groupings and relationships in the data and have been successfully applied in a broad spectrum of research areas ranging from speech recognition to financial analysis.
The Self-Organizing Map belongs to the class of unsupervised and competitive learning algorithms. It is a sheet-like neural network, with nodes arranged as a regular, usually two-dimensional grid. Each node is directly associated with a weight vector. The items in the input data set are assumed to be in a vector format. If n is the dimension of the input space, then every node on the map grid holds an n-dimensional vector of weights. The basic principle is to adjust these weight vectors until the map represents "a picture" of the input data set. Since the number of map nodes is usually significantly smaller than the number of items in the dataset, it is needless to say that it is impossible to represent every input item from the data space on the map. Rather, the objective is to achieve a configuration in which the

distribution of the data is reflected and the most important metric relationships are preserved. In particular, interest is in obtaining a correlation between the similarity of items in the dataset and the distance of their most alike representatives on the map. In other words, items that are similar in the input space should map to nearby nodes on the grid.

5.1 The Algorithm

The algorithm proceeds iteratively. On each training step a data sample **x** from the input space is selected. The learning process is competitive, meaning that we determine a winning unit c on the map whose weight vector **m** is most similar to the input sample **x**.

$$\|\mathbf{x}-\mathbf{m}_c\| = \min_i \|\mathbf{x}-\mathbf{m}_i\|$$

The weight vector $\mathbf{m}_c$ of the best matching unit is modified to match the sample **x** even closer. As an extension to standard competitive learning, the nodes surrounding the best matching unit are adapted as well. Their weight vectors $\mathbf{m}_i$ are also "moved towards" the sample **x**. The update rule may be formulated as:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci} * (\mathbf{x} - \mathbf{m}_i(t))$$

The scalar factor $h_{ci}(t)$ is often referred to as the "neighbourhood function". It is usually a Gaussian curve, decreasing from the neighbourhood centre node c to the outer limits of the neighbourhood.

$$h_{ci}(t) = \alpha(t) * \exp\left(-\frac{\|\mathbf{r}_c - \mathbf{r}_i\|^2}{2\sigma(t)^2}\right)$$

In the above equation, α(t) is another scalar multiplier called the "learning rate". It may be regarded as the height of the neighbourhood kernel. σ(t) is the radius or the width the neighbourhood kernel. It specifies the "region of influence" that the input sample has on the map. Both the height and the width of the neighbourhood function decrease monotonically with time.

As can be seen, nodes closer to the best matching unit will be more strongly adjusted than nodes further away. At the beginning of the learning process, the best matching unit (BMU) will be modified very strongly and the neighbourhood is fairly large. Towards the end, only very slight modifications will take place and the neighbourhood includes little more than the BMU itself. This corresponds to "rough ordering" at the beginning of the training phase and "fine" tuning near the end.

Since not only the winning node is tuned towards the input pattern but also the neighbouring nodes, it is probable that similar input patterns in future training cycles will find their best matching weight vector at nearby nodes on the map. In the run of the learning process, this leads to a spatial arrangement of the input patterns, thus inherently clustering the data. The more similar two patterns are, the closer their best matching units are likely to be on the final map. It is often said, that the Self-Organizing Map folds like an elastic net onto the "cloud" formed by the input data.

It is important to state that the Self-Organizing Map algorithm is not a clustering algorithm. It is intended primarily as a tool in reducing the dimensionality of the data and for information visualization. Of course, this includes the visualization of groups of similar items. But the Self-Organizing Map is not a tool that will produce an explicit partitioning of a dataset into a precise number of groups. This also explains why the concept of a "cluster" is not well defined for the Self-Organizing Map. The maps do not show sharp cluster borders and there is no obvious centroid. Of course, one can theoretically think of each node on the map as a cluster centroid. The cluster corresponding to each node could then be said to include all dataset items mapping to this node. But this is not a sound approach in the practical application of the SOM. It tempts the user to use small maps of only k nodes, expecting that this will produce k clusters in the same way k-means does. The results with such small maps, however, are very poor. The heart of the algorithm is the neighbourhood function and the concept of adjusting not only the best matching unit but also its surrounding units. This will create "neighbourhoods" of similar nodes - but only if the space on the map is sufficiently large to allow this. An interesting point regarding the topological preservation is that this refers not only to the intra-cluster relationships but also to the inter-cluster relationships. In other words, not only the distances of objects within a cluster are meaningful, but also the distances between clusters. For example, one may expect two nearby clusters to be more similar than two distant clusters.

5.2 Application to Iris dataset

The following application is just a pure explicative one, regarding a small dataset with only four dimensions; moreover, it is intended to be useful to illustrate SOM machinery.
The iris dataset contains 150 items of four of the most prominent characteristics of the iris flower: petal length, petal width, sepal length, sepal width. Each item is classified in one of the three classes Setosa, Virginica or Versicolor.
SOM results are reported in the following:

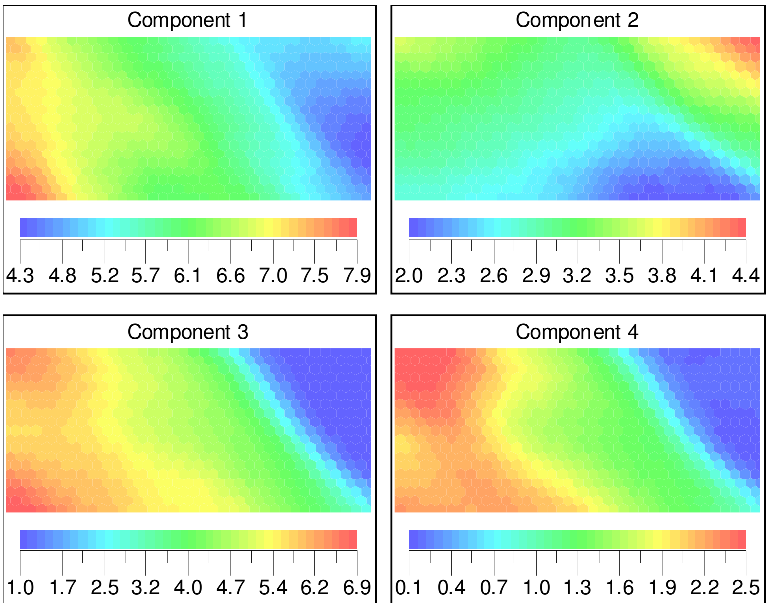

Figure 11: Componentl: petal length; Component2: petal width; Component3: sepal length; Component4: sepal width. Each of the four screens displays the spread of the values of the respective component on the map.

Some conclusions can be drawn immediately: sepal length and sepal width have a very similar mapping and hence one of it is redundant in the description of the iris flower and it can be eliminated; petal length and sepal length have a strong correlation in the range of low values and hence one can always expect to have a small sepal length in presence of a small petal length (or viceversa). Such a kind of conclusions is very obvious ones in this case, but their relevance grows when one faces with bigger and more complex datasets.
The following picture reports a labelling of the preceding SOM, based on the pertaining class of each of the mapped items.

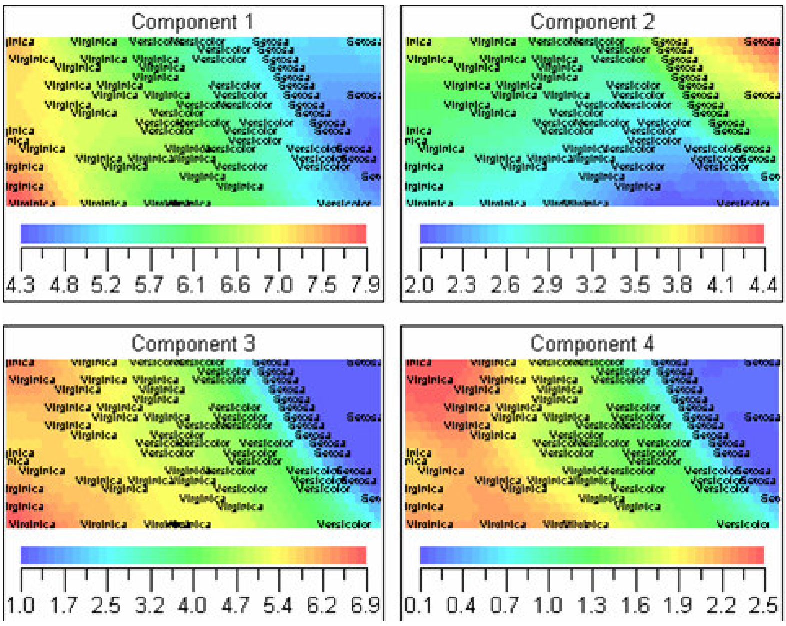

Figure 12: Note the contiguous mapping of similar records: Setosa, Virginica and Versicolor are each mapped on a particular region of the map.

Considerations on the relative distance of different kinds of items can be drawn better if one considers the U-Matrix representation [34]:

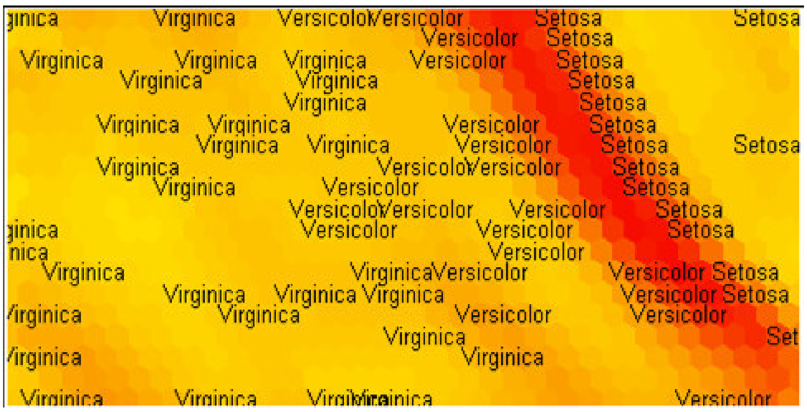

Figure 13: U-Matrix reports on each unit of the map the mean value of the distances between its weight vector and m the weight vectors of its neighest neighbors. Yellow units are much closer than red units.

As can be seen, Setosa are separated by a red region from Virginica and Versicolor: then Setosa are more similar than Versicolor and Virginica do and they form a possible well defined cluster.

6. Exhaustive Example: Multi Objective Robust Design Optimization of an AIRFOIL

Using the Multi Objective Robust Design theory developed, we perform a more realistic optimization case consisting in the design of a non-symmetric airfoil based on the RAE28222 geometry, using as flow solver the Navier-Stokes version of MUFLO and AIRFOIL codes [29], which uses as turbulence model the Johnson-Coakley equations (fig. 14). Also in this case the upper and lower side of the profile are defined by two 10-degree Bèzier curves, and the co-ordinates of their control points become the variables of the optimization (fig. 14)

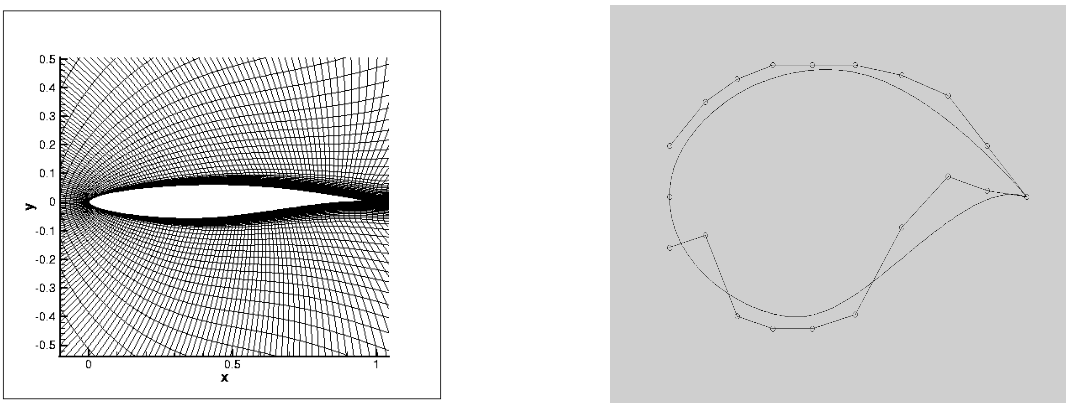

Figure 14: Airfoil mesh with MUFLO (a) and airfoil parameterization using Bezier curves (b).

The uncertainties are relative to the Mach number (M=0.73±0.05) and to the angle of attack (a=2°±0.5°).
The optimization goal is to find out an airfoil geometry which yields better results respect to performances and stability, taking in account of the two uncertainties parameters (angle of attack and free Mach number). From a mathematical point of view the optimization problem becomes:

$$
\min_{\Delta M,\Delta\alpha}\left(E(c_d),\sigma^2(c_l),\sigma^2(c_d)\right) \quad \max_{\Delta M,\Delta\alpha}\left(E(c_l)\right)
$$

with

$$
\begin{array}{ll}
E(c_l) \geq E(c_l)^{RAE2822} & \sigma^2(c_l) \leq \sigma^2(c_l)^{RAE2822} \\
E(c_d) \leq E(c_d)^{RAE2822} & \sigma^2(c_d) \leq \sigma^2(c_d)^{RAE2822} \\
|E(c_m)| \leq \left|E(c_m)^{RAE2822}\right| & \sigma^2(c_m) \leq \sigma^2(c_m)^{RAE2822}
\end{array}
$$

We set seven constraints to the optimization problem: the thickness is fixed to be higher than 12% of the chord length, and the new configuration should present values better than or equal to the original RAE2822 airfoil corresponding to the mean and variance of drag, lift and pitching momentum coefficients.

In table 1 we report the lift and drag mean and standard deviation values relative to the real RAE2822 airfoil and to the data extrapolated by adaptive DACE with a different number of training points.

Table 1 Comparison of real and extrapolated mean and deviation performances.

RAE2822	Cl mean	Cd mean	Sigma Cl	Sigma Cd	%err σcl	%err σcd	Training points
Real	0.677	0.173	2.24	2.00			
RS	0.671	0.179	1.71	1.80	23.7%	10.0%	5
RS	0.675	0.179	2.28	1.90	1.8%	5.0%	7
RS	0.674	0.176	2.21	2.00	1.3%	<1%	9
RS	0.675	0.176	2.26	2.00	<1%	<1%	11

Since the relative errors are less than 1% using 9 training points, we have decided to use this extrapolation method in the optimization, in order to reduce the total number of computations required: in this way, for any configuration proposed by the optimization algorithm, the Lift and Drag Surfaces in function of Mach number and angle alpha, needed to express the objectives of the robust design optimization, are obtained by only 9 CFD simulations instead of more than 120. The general strategy to achieve the objective functions for each configuration (airfoil shape) by means of the DACE method consists of the following steps:

1. to set the starting data base of 5 training data (evaluated by CFD analysis);
2. to extrapolated C_L, C_D, C_M, functions by mean of DACE and evaluate the objective functions (mean(C_L), mean(C_D), mean(C_M), var(C_L), var(C_D), var(C_M));
3. to get the location (*star-location*) of the extrapolated value that is associate with the highest error value (Eq. 22);
4. to evaluate a CFD analysis corresponding to the *star-location* and update the database;
5. achieve the objective function trough the database updated and compare the values of two consecutive steps;
6. to stop the process stops if the difference between the objective functions is lower than an tolerance (in this case 1%);

Taking attention to the preceding scheme it is possible to understand that thanks to the adaptive method, in automatic way, we can define the minimum number of training data to achieve the best extrapolated function. The more complicate the function is and the more training data we need in order to extrapolate a correct function. Facing any kind of problem we don't know how complex the function is, but through the adaptive method it is possible to recognize when it is worth to apply many high fidelity analysis. Using an extrapolating method based on quadratic function, for example, the only feasible thing to do would be to set a big data base in order to try to get anyway a good Response Surface, but sometimes this approach could be a waste of time and very expensive to apply.

6.1 Results

MOGA [30] (Multi Objective Genetic Algorithm) has been used to solve the Multi Objective Robust Design Optimization of airfoils in transonic field and modeFrontier is the software used to implement MOGA. The problem has been set with 40 individuals per generation and 16 generations. In Figure 15 the trend of the objective functions during the optimization process is shown: it is possible to notice that the desirable trends have been reached. In particular it is possible to underline that a remarkable improvement has been achieved regarding the standard deviation of drag coefficient. In fact, the peculiarity to face an optimization of airfoils in transonic field according to the principles of Robust Design is to be able to look for stable solutions but at the same time with as much performances as possible. In this case it is evident that even if the RAE2822 was designed to have the highest performances achievable corresponding to the operating condition considered, it has been possible to find more stable solutions especially concerning the drag coefficient value. This result is directly linked to the high variability of the positions of the possible shock waves that are present in transonic field as the operating conditions slightly changes.

Having defined 4 objectives, according the Pareto theory, the final solution is not unique but will be a set of solutions, which are the best compromises between the different objectives (the *Pareto frontier*). Fig. 16 compares the configurations that belong to the Pareto frontier (mean lift versus mean drag, and variance of lift versus variance of drag). It is possible to check that the optimization has been completed with success: in fact note the position of the original design (RAE2822) compared with the others solutions.

Another interesting way to visualise the Pareto Frontier is the use of the Self Organizing Maps, in which it is possible to clear the local correlation between the objectives in the Pareto Frontier space (Fig. 17). For example, a clear correlation is visualised between the stability and the performance of the drag coefficient, especially for the zone where the best mean drag coefficient is considered. Different considerations could derive from the observation of the correlations between stability and mean performance of the lift coefficient: low mean lift performances doesn't correspond to stable solutions and low performances for lift stability correspond to mean values for the lift performance. These results give value to the considerations that the better approach to the Robust Design is the use of the Multi Objective theory: the final results is not trivial, and the Pareto Frontier contains different physical behaviour for the airfoils.

The importance to have a suitable tool for the visualisation of the results, especially in the case of multidimensional approach, is confirmed from Fig. 18, where the Self Organizing Maps, are used to visualize the iteration between the design variable v_{13} and the drag coefficient stability, inside the configuration of the Pareto frontier. It is evident the direct correlation between variables and objective: high values for the variables correspond to bed stability performance, and with low variable values it is possible to observe good drag stability performance. It is important to note that variable v13 is a design variable for the

upper surface of the airfoil, just where the shock way is present, and where physically the correlation with the stable behaviour is well known.
Using the Pareto Frontier approach for the Multi Objective Robust Design optimisation, it is well known that the final solution is not unique, but the algorithm finds the set of the best compromises between the different objectives. For choosing the final solution for the design, normally a Multi Criteria Decision Making tool has to be used, in order to take the solution which is the best compromise not only in the numerical optimization point of view, but for the designer ideas too.
In Fig. 19 we report the comparison of the chosen design obtained by the optimization. We choose the best solution for drag coefficient stability. As it is possible to note in fig.19, the Drag response surface for alpha and Mach (re-computed by 121 CFD computations to validate the results of the optimization), presents a mean value lower and also the standard deviation is lower, i.e. the solution have improved the mean performance and also the behaviour is more stable. From table 2 it is also possible to note that all the objectives and constraints have been respected.

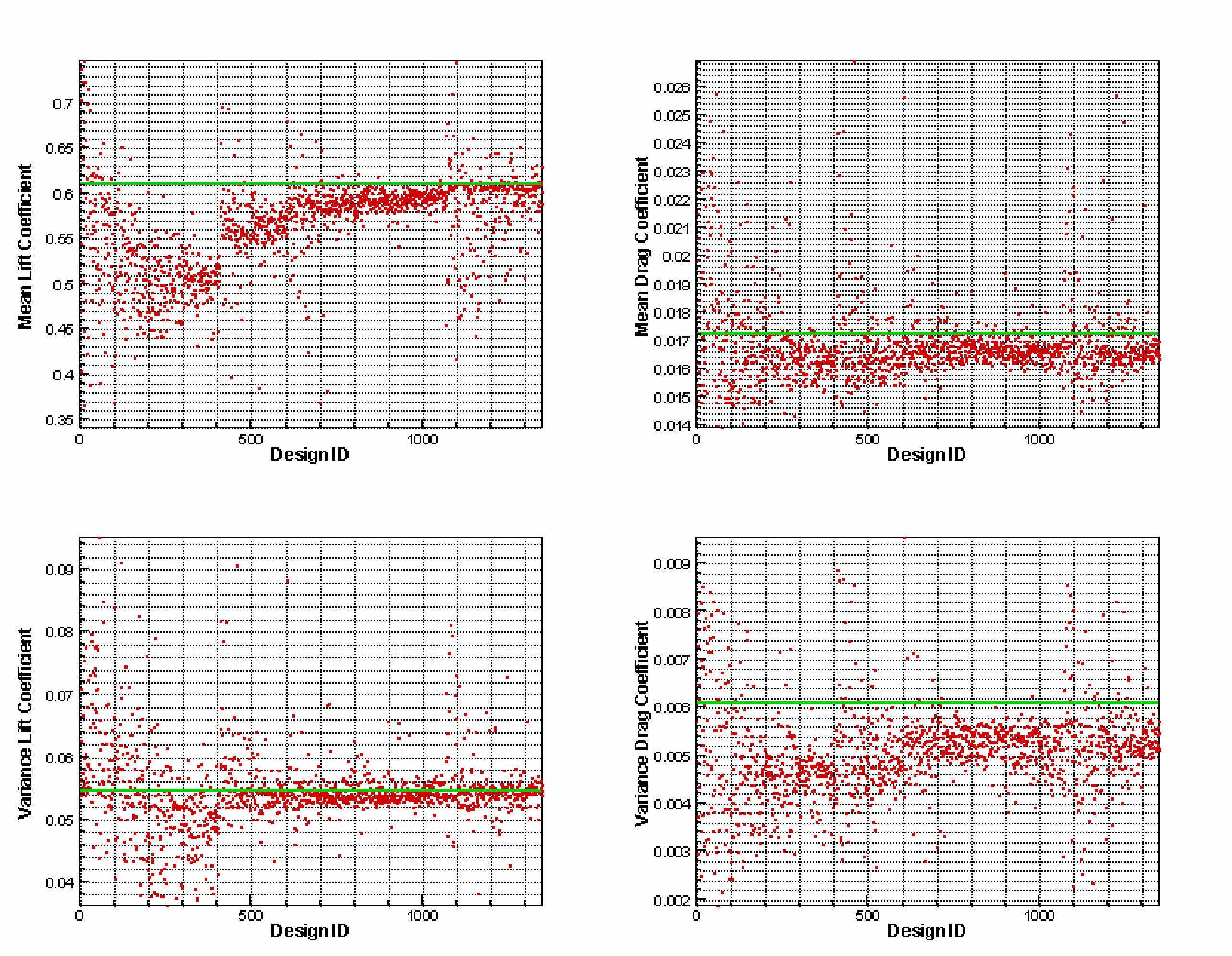

Figure 15: Convergence histories during the optimization phase (original RAE2822 solution performances represent by green profile)

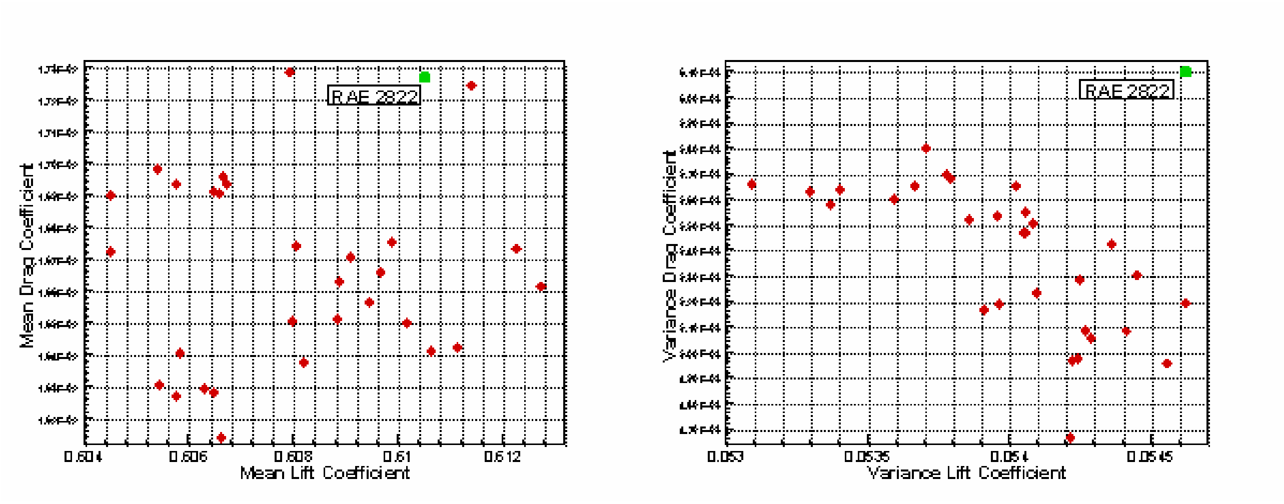

Figure16: Pareto Frontier representation in comparison with the original RAE2822 solution.

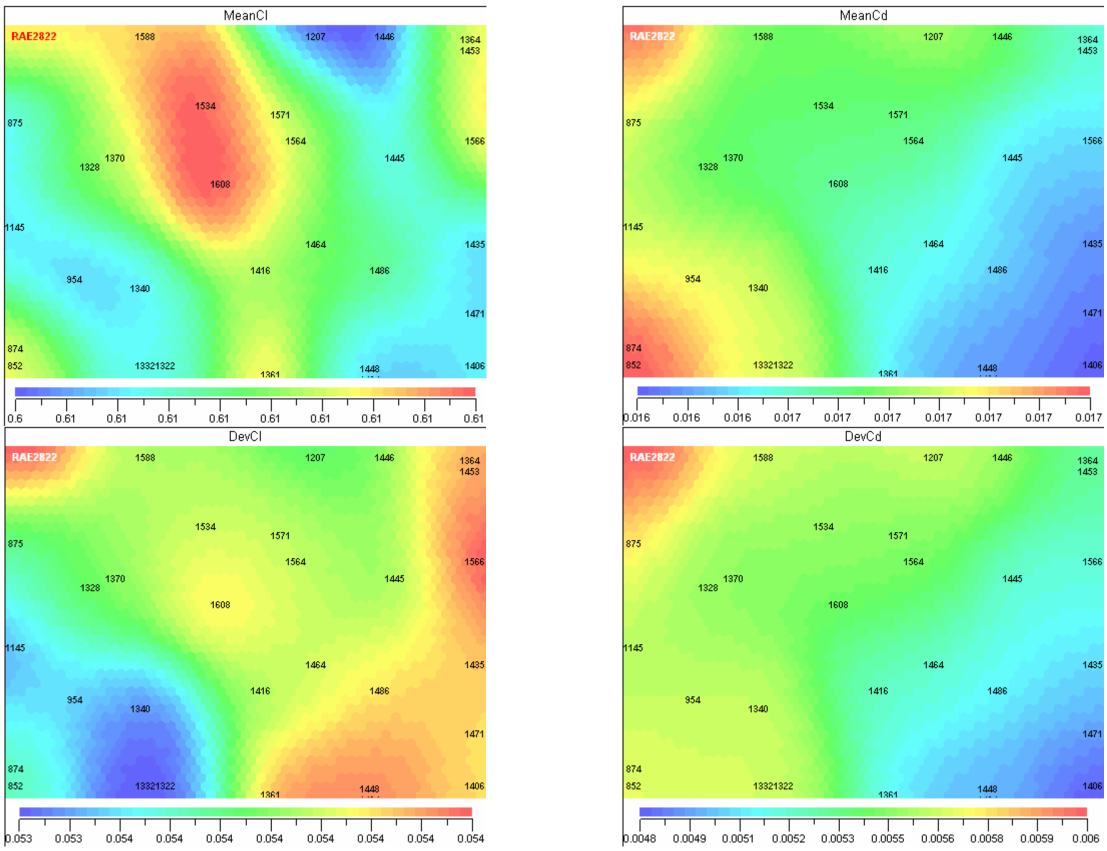

Figure 17: Visualization of the Pareto Frontier by mean Self Organizing Maps.

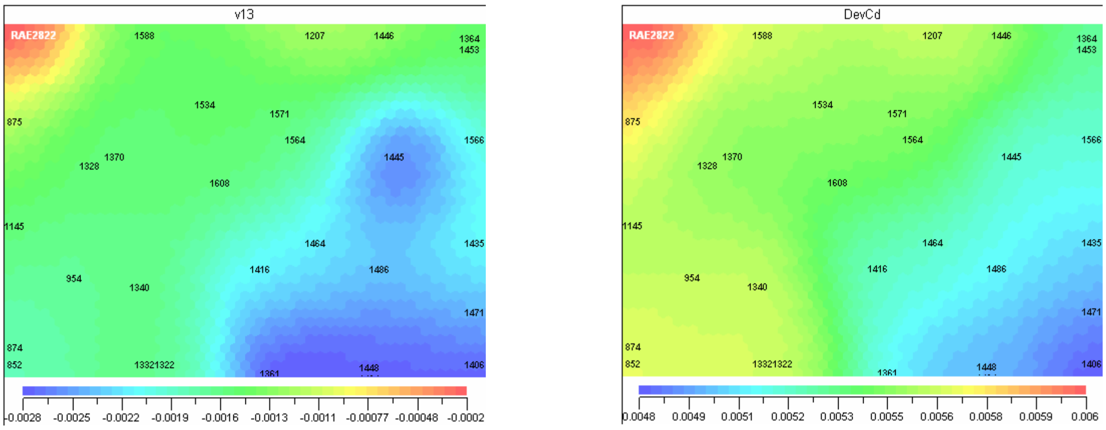

Figure 18: Visualization of the iteration between the variable v13 with the drag coefficient stability.

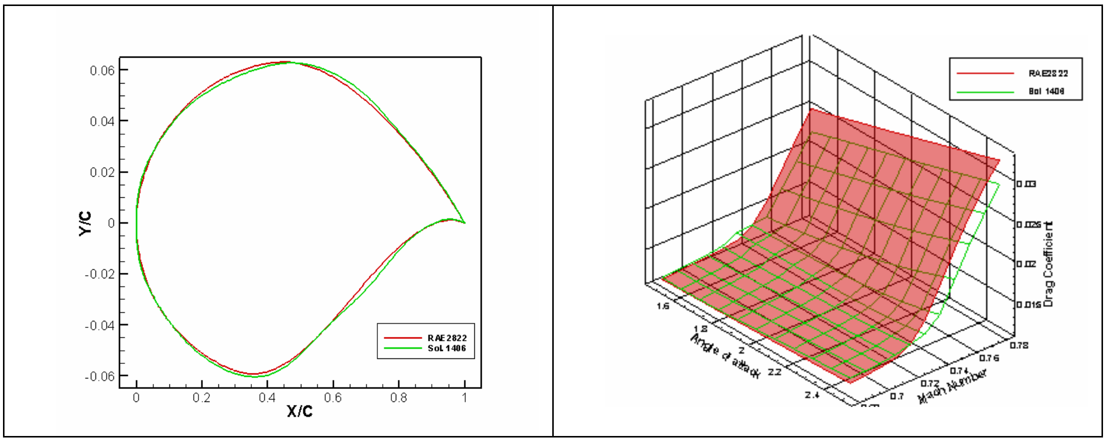

Figure 19: Comparison between the original RAE2822 geometry and the best for drag coefficient stability on the Pareto Frontier (Sol. 1496). On the left geometry comparison, on the right drag coefficient comparison surface.

Table 2: Comparison of original RAE and best configuration mean and deviation performances.

	RAE2822	**Sol. 1406**
Mean Cl	6.10E-1	6.06E-1
Mean Cd	**1.73E-2**	**1.61E-2**
Mean Cm	-8.88E-2	-8.72E-2
σ Cl	5.45E-2	5.42E-2
σ Cd	**6.10E-3**	**4.67E-3**
σ Cm	8.27E-3	7.06E-3

7 References

[1] 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 30 August-1 September 2004, Albany, New York

[2] 1st AIAA Multidisciplinary Design Optimization Specialist Conference,18 - 21 Apr 2005 Hyatt Regency Austin, Austin, Texas

[3] Taghuci G., Performance Analysis Design, *International Journal of Production Research* 16 (6), pp. 521-530 1978

[4] Guide for verificatiob and validation of computational fluid dynamic simulation, *AIAA Guide* G-077-1998, 1998

[5] M.W. Trosset, N. M. Alexandrov, L.T. Watson, New Methods for Robust Design Using Computer Simulations, *Proceeding of the section on physical and engineering science*, America Statistical Association, 2003

[6] W. J. Welch, J. Sacks, A system for quality improvement via computer experiments. *Communications in Statistics-Theory and Methods*, 20:477-495

[7] Drela M., Pros and Cons of Airfoil Optimization, *Frontier of Computational Fluid Dynamics*, ed. by D. A. Caughey and M. M. Hafez, World Scientific, 1998, pp. 363-381.

[8] Wu Li, Sharon Padula, "Performance Trades Study for Robust Airfoil Shape Optimization", *21th Applied Aerodynamics Conference*, AIAA-2003-3790.

[9] Hicks R. M. and Vanderplaats G. N., Application of numerical optimization to the design of supercritical airfoils without drag-creep, SAE Paper No. 770440, *Business Aircraft Meeting,* Wichita, 1977.

[10] L. Huyse, R. M. Lewis, Aerodynamic Shape Optimization of Two-dimensional Airfoils Under Uncertain Conditions, *NASA/CR-2001-210648 ICASE Report* No. 2001-1

[11] S. L. Padula, W. Li, Options for Robust Airfoil Optimization under uncertainties, *9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary analysis and Optimization*, September 4-6, 2003 AIAA paper 2002-5602

[12] D. W. Zingg, S. Elias, On Aerodynamic Optimization Under a Range of Operating Conditions, *44th AIAA Aerospace Sciences Meeting and Exhibit*, 9-12 January 2006, Reno, Nevada

[13] X. Du, W. Chen, A Hierarchical Approach to Collaborative Multidisciplinary Robust Design, Proceddings *4th Congress of Structural and Multidisciplinary Optimization*, Dalin, China, 2001

[14] M. Kalsa, K. Hacker, K. Lewis, A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex Systems Design, *Journal of Mechanical Design*, vol 123, no 1, pp 1-10, 2001

[15] B, Youn, K. Choi, M. Kokkolaras, P. Papalambros, Z. Mourelatos, D. Gorsich, Techniques for Estimating Uncertainty Propagation in Probabilistic Design Of Multilevel System, *10th AIAA/ISSMO Multidisciplinary Analisys and Optimization Conference*, Sept. 2004, Albany, New York.

[16] Sacks J., Welch, W. J., Mitchell T. J., Wynn H. P., Design and Analysis of Computer Experiment, *Statistical Science*, Vol 4 No 4, pp 409-453, 1989

[17] Egorov, I.N., Kretinin, G.V. and Leshchenko, I.A. How to Execute Robust Design Optimization", *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 04 - 06 Sep. 2002, Atlanta, Georgia.

[18] Nash, J.F., Non-cooperative Games, *Annals of Mathematics*, 1951.

[19] Clarich A., Pediroda V., Poloni C., Periaux J., Comparisom between different Game Theory in Robust Design Optimization Int. Conf. on Computational Methods for Coupled Problems, *Science and Engineering Coupled Problems,* 2005, ed. by M. Papadrakakis, E. Oñate and B. Schrefler, CIMNE, Barcelona, 2005.

[20] Rao S.S., Engineering optimization, Theory and Practice, *Wiley&Sons*, 1996.

[21] Wang J.W., Periaux J., Search Space Decomposition of Nash/Stackelberg Games using Genetic Algorithms for Multi-Point Design Optimization in Aerodynamics from Domain Decomposition Methods, *Science and Engineering*, pp. 139-149, CIMNE 2002.

[22] Clarich A., Padovan L., Pediroda V., Periaux J., Poloni C., Application of game strategy in multi- objective robust design optimization implementing self-adaptive search space decomposition by statistical analysis, Jyvaskila, *ECCOMAS2004*.

[23] Koehler J.R, Owen A.B (1996), *Computer Experiments, Handbook of Statistics*, Eds. S. Ghosh

[24] Sacks J., Welch W.J., Michell T.J., Wynn H. P. (1989), *Design and Analysis of Computer Experiments*, *Statistical Science*, Vol 4 No.4, pp.409-453.

[25] Isaaks E.H., Srivastava R.M. (1989), An Introduction to Applied Geostatistics*, Oxford Univer sity* Press, NY.

[26] Koehler J.R, Owen A.B (1996), *Computer Experiments, Handbook of Statistics*, Eds. S. Ghosh

[27] Schonlau M. (1997), Computer Experiments and Global Optimization, *Ph D. Dissertation*, University of Waterloo.

[28] Del Vecchio R.J. Understanding Design of Experiments: A Primer for Technologists, Hanser Understanding Books. (1997),

[29] Haase W., Wagner B., Jamenson A., Development of a Navier-Stokes method based on a finite volume technique for unsteady Euler equations, in *Proceeding of the 5th GAMM conference on Numerical Methods in Fluid Mechanics, Friedrich Viewer und Sohn, Braunschweigh* Wiesbaden, 1983

[30] Poloni C., Pediroda V., 1997, GA coupled with computationally expensive simulations: tool to improve efficiency, in *Genetic Algoritms and Evolution Strategies in Engeneering and Computer Science*, edited by Wiley Sons, 1997.

[31] Kohonen, T. (2001) Self Organizing Maps, 3a Ed., *Spinger*

[32] Kohonen, T. (1982) Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, 43:59-69

[33] Ojia,M. Kaski,S. and Kohonen, T. Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum (2002), Neural Computing Surveys 3, 1-156

[34] Ultsch,A. Siemon,H.P. Kohonen's self organizing feature maps for exploratory data analysis. In Proc. INNC'90, *Int. Neural Network Conf.*, pages 305-308, Dordrecht, Netherlands, 1990. Kluwer.