1st International Conference on Experiments/Process/ System Modelling/Simulation/Optimization,
1st IC-EpsMsO
Athens 6-9 July, 2005
©IC-EpsMsO

ROBUST DESIGN OF COMPRESSOR CASCADE AIRFOILS, USING EVOLUTIONARY ALGORITHMS AND SURROGATE MODELS

Evgenia A. Kontoleontos[1], **Kyriakos C. Giannakoglou**[2] **and Dimitrios G. Koubogiannis**[3]

[1] Student, School of Mechanical Engineering, NTUA, Athens, Greece

[2] Associate Professor,
Lab. of Thermal Turbomachines, National Technical University of Athens (NTUA)
e-mail: kgianna@central.ntua.gr - Web page: http://velos0.ltt.mech.ntua.gr/EASY

[3] Assistant Professor,
Lab. of Steam Generators & Thermal Turbomachines, Dept. of Energy Technology,
Technological Educational Institute of Athens, e-mail: dkoubog@teiath.gr

Key words: Optimization, Robust Design, Evolutionary Algorithms, Kriging, Design of Experiments, Turbomachinery Flows

Abstract. *The aim of this paper is to present and test an optimization method for the design of turbomachinery blade airfoils with optimal performance at given flow conditions and the additional requirement that this performance remains (almost) constant even in the presence of slight manufacturing–assembling errors or slight shape changes during the operation. The proposed method goes a step beyond standard design tools ensuring the robustness of the designed shapes under uncertainties related to the design variables. The basic optimization tool is a multi–objective evolutionary algorithm, which cooperates with two kinds of evaluation tools. All blade airfoils are evaluated, at the nominal flow conditions, using an integral boundary layer method coupled with an inviscid flow solver. Changes in performance, due to changes in shape are quantified through the "massive" use of inexpensive, surrogate evaluation models. The use of surrogate models requires an initial database of evaluated samples. The sampling is carried out by means of a new, kriging-based search algorithm aiming at minimizing the database size together with the relevant* CPU *cost. In the course of this incremental algorithm, the new database point is the one for which kriging yields the most uncertain prediction. The robust design of a compressor cascade airfoil is demonstrated.*

1 INTRODUCTION

The design of shapes (wings, blades, airfoils, etc) with optimal aerodynamic performance at given flow conditions starts once objectives and constraints have been defined. Constraints can be related either to flow performance measures (such as a target lift) or the shape geometry itself (minimum thickness constraints). The shape should be parameterized and the design variables identified. An optimization tool, based on flow simulation software for the aerodynamic evaluation of candidate shapes should be used to compute the optimal solution(s). The cost of the optimization depends on the cost of aerodynamically evaluating a single shape and the number of evaluations required to reach the optimal solution. Evolutionary (*EAs*[1]) and deterministic algorithms (usually, descent algorithms based on adjoint techniques [2]) are in use. Low–cost approximate evaluation tools (metamodels, such as artificial neural networks, kriging, response surface models, etc) are often used to lower the overall *CPU* cost by replacing as many as possible exact evaluations with fitness approximations [1].

However, even if all aforementioned tools are available and the design is carried out with reasonable *CPU* cost, uncertainties related to any stochastic deficiency may occur. These uncertainties are expected to affect the real performance of the manufactured shape. They are often due to manufacturing or assembling errors, changes in shape or positioning, changes in operating conditions, etc. Other sources of uncertainty are related to the physical model used, the relevant assumptions, discretization, solution and round-off errors [3]. In the present work, only stochastic shape variations will be considered; the flow model and solver as well as the flow conditions used will be considered as "credible". So, it will be assumed that there are no model form uncertainties. As the paper develops, it will become clear that the proposed method is *robust* in the sense that it produces solutions which are insensitive to "reasonably" small changes in the design variables and remain reliable even in their presence. Let us note that reliability–based methods have been used for many decades in civil engineering and are now mandatory in aircraft engine design projects [4].

The robust design method proposed in this paper aims at designing cascade airfoils with minimal total pressure losses at a given operating point. A constrained optimization problem is solved, where the first constraint is related to the minimum allowed airfoil thickness and the second one imposes the minimum desired flow turning. The N design variables are the coordinates of a Bezier–Bernstein polynomial parameterization of the airfoil contour. This single–objective problem is handled as a two–objective one: the second objective quantifies the mean performance degradation of $2N$ erroneously manufactured airfoils, created by perturbing all design variables, one by one. It is evident that the cost for carrying out $2N+1$ exact evaluations per candidate solution is too much! Thus, the proposed method utilizes the exact evaluation tool (namely an integral boundary layer method, software *MISES* developed by M. Drela [5]) only for the candidate airfoil whereas all perturbed airfoils are evaluated using a single, locally trained radial basis function network (*RBFN*).

The search is based on a multi–objective *EA* (code *EASY*, developed by *NTUA*) which provides the front of Pareto optimal solutions in the two–objective space. However, even for the members of the first generation, the second objective function computation calls for a database of previously evaluated solutions. The search space needs to be sampled beforehand by selecting and evaluating a number of samples. This number should be kept as small as possible for the sake of minimum *CPU* cost. Instead of other known design-of-experiment techniques [6] a new method based on the kriging model has been developed.

2 THE PROPOSED ROBUST DESIGN ALGORITHM

In the Introduction, the basic characteristics of the proposed robust design algorithm have been described . In this section, the method will be analyzed, step by step. The various tools used are discussed in subsequent sections. However, we will refrain from presenting the flow analysis sofware [5]; we should just mention that the *CPU* cost of a single evaluation is about 1.5 min. on a modern personal computer.

Initial Sampling: The search space $\Omega \in \Re^N$, as defined by the lower and upper bounds of the N design variables, is sampled. The sampling includes two discrete phases. During the first phase, a simple factorial design with only two levels for each factor, is employed. For small N values, the 2^N factorial design, which is the main building block in many response surface designs, with the addition of the center point, can be used. Otherwise, a fractional factorial design can be used instead [6]. The so–collected L_1 first samples are evaluated and their performances $y^{(i)},\ i=1,L_1$ recorded. Then, the $L_2 = L_{DB} - L_1$ remaining samples (let L_{DB} be the desired database size) are computed, one by one through L_2 steps, using the kriging model. During the k-th step, the kriging model is trained on the available L_1+k-1 samples and the new sample is located as the point in Ω which exhibits the maximum Mean Squared Error (*MSE*) in prediction. A single-objective *EA* is used for the training of the kriging model as well as during the search of each new sample. The evaluation of the new point is necessary before passing to the next search. At the end of this phase, L_{DB} patterns $\vec{x}^{(i)} \in \Re^N$ and the corresponding responses $y^{(i)}$ become available. A simple demonstration of the proposed sampling method is shown in fig. 1.

Optimization Phase: This phase is based on the two–objective *EA*, the exact (integral boundary layer method) and the surrogate evaluation tools (*RBFNs*). For each population member $\vec{x}$, after being exactly evaluated, its closest P neighbors in the database are selected. Closeness is measured in terms of distances in Ω. By also considering the current individual ($P+1$ patterns, in total), a local *RBFN* is trained. This is used to compute approximate responses y_{RBF} for the $2N$ perturbed shapes ($\pm\varepsilon$ for each component of $\vec{x}$). An auxiliary computation provides the y_{LSQ} values for the perturbed shapes, through a least-squares fitting of y_{RBF}. The standard deviation σ_{nei} of $y_{RBF} - y_{LSQ}$ constitutes the second objective function which quantifies the "robustness" of $\vec{x}$. Using y and σ_{nei} as the two targets, an *EA* computes the Pareto front of optimal solutions.

Cross–Validation: The finally computed Pareto front members are cross-validated by cross-checking pieces of information computed using the inexact model (*RBFN*).

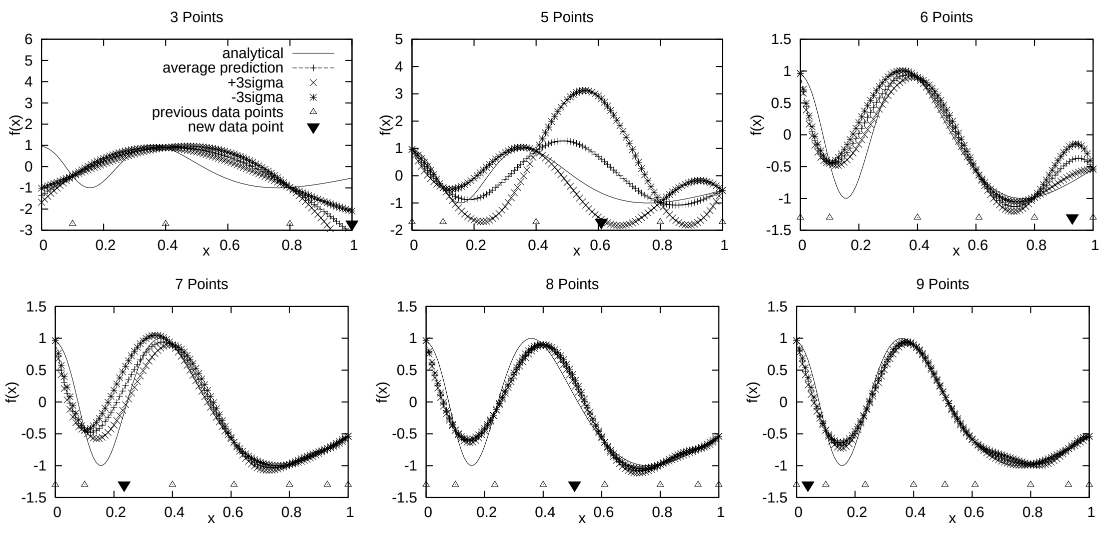

Figure 1: The proposed sampling method is demonstrated through a one–dimensional example. The aim is to generate data points along the x–axis $(0 \leq x \leq 1)$ which, when used to train kriging, approximate accurately the curve $y(x) = cos\left(15\ exp(\frac{-0.4}{x+0.1})\right)$. The three initial sampling points were $x = 0.1,\ 0.4$ and 0.8. This figure presents six snapshots during the sequential point generation algorithm. In each one, the *average* curve predicted by kriging as well as the *average*$\pm 3\sigma$ curves (σ is also predicted by kriging) are illustrated. The latter define the confidence of the kriging–based prediction at any x. As expected, the two curves meet at the training (marked with *p*revious) data points. The existing as well as the new data point are clearly marked.

3 THE MULTI–OBJECTIVE *EA*

A multi–objective *EA* which simultaneously evolves three population sets is used. Let index g be the generation counter. Then, $S^{g,\mu}$ and $S^{g,\lambda}$ denote the actual parent and offspring sets, with μ and λ members, respectively; the third set, which will be referred to as the elite or archival set, collects the non-dominated solutions (in the Pareto sense) in each generation. In each generation, $S^{g+1,\alpha}$ collects the elite members of $S^{g,\lambda} \cup S^{g,\alpha}$.

The search starts $(g = 0)$ by randomly selecting λ individuals $\vec{x}^{(i)} \in S^{g,\lambda} \subset \Re^N$, which are evaluated with respect to all objective functions, $\vec{y}^{(i)} = \vec{F}\left(\vec{x}^{(i)}\right)$. This population is ranked on the basis of an individual's nondomination and a provisional Pareto front $S^{g+1,\alpha*}$ is formed. Using the $\vec{y}^{(i)}$ values, $i \in S^{g,\mu} \cup S^{g,\lambda} \cup S^{g+1,\alpha*}$ (note that $S^{0,\mu} = \emptyset$), a unique cost value $\phi^{(i)}$ is assigned to each individual using the concept of *strength* [7],[8]. Practically, all $S^{g+1,\alpha*}$ members are first assigned a cost value proportional to the number of the $S^{g,\mu} \cup S^{g,\lambda}$ individuals dominated by them. The ϕ value of each of the $S^{g,\mu} \cup S^{g,\lambda}$ members is set equal to 1 plus the sum of strengths of the $S^{g+1,\alpha*}$ individuals which dominate it. The final ϕ value is the previously computed value plus a contribution proportional to the local density of individuals. Once the $\phi^{(i)}$ values have been computed, standard single–objective evolution operators can be used for the selection of parents and the creation of the next generation.

Then, the archival front $S^{g+1,\alpha}$ is computed. If its size does not exceed a threshold value α, $S^{g+1,\alpha*}$ is merely copied to $S^{g+1,\alpha}$; otherwise, an iterative thinning process, [8], which maintains individuals lying along the edges of the archival front is employed. Aiming at preserving elite solutions, a small fraction of the topmost solutions in $S^{g+1,\alpha}$ is directly copied to $S^{g,\lambda}$.

All $S^{g,\mu}$ members that exceed the maximum allowed life span of κ generations are eliminated. The new parental set $S^{g+1,\mu}$ is formed from $S^{g,\lambda} \cup S^{g,\mu}$.

The new offspring set $S^{g+1,\lambda}$ is created by applying parent selection operators to $S^{g+1,\mu} \cup S^{g+1,\alpha}$. Parent individuals are randomly selected from $S^{g+1,\mu}$ (with probability p_{ps}) or $S^{g+1,\alpha}$ (with probability $1 - p_{ps}$). Compared to *SPEA2*, this is an important difference of the present method since in *SPEA2* parents are selected only from $S^{g+1,\alpha}$. Finally, recombination and mutation operators are utilized to create a new offspring to be inserted into $S^{g+1,\lambda}$.

The same procedure is repeated after setting $g := g + 1$, unless the stopping criterion is met.

4 THE KRIGING MODEL

Kriging [9] assumes that the function being interpolated is an observation of a stochastic process. The "system" is modeled as a stochastic (usually Gaussian) process with expected value μ and correlation function $\mathcal{R}$. Kriging can estimate the expected response and its *MSE* at any new point, which does not belong to the training set. For the training points themselves, it exactly reproduces the response and zeroes *MSE*. We recall that, in the proposed method, kriging is used for the formation of the initial database. Among its two possible outputs (expected response and *MSE*), only *MSE* is of interest. Kriging is briefly described below.

Having L available data points along with their responses ($\vec{x}^{(i)} \in \Re^N$, $y^{(i)}$, $i = 1, 2, ..., L$), the aim of kriging is to approximate the expected response $\hat{y}$ and its *MSE* at any new solution $\vec{x}$. $\hat{y}$ is expressed as the sum of a drift function and a covariance function, namely

$$\hat{y} = \hat{\mu} + z(\vec{x}) \tag{1}$$

Here, the expected value $\hat{\mu}$ is chosen to be the drift function. The covariance function $z(\vec{x})$, $\forall \vec{x} \in \Re^N$, is supposed to have zero mean and a "known" correlation function $\mathcal{R}$ with any other point in $\Re^N$. A common choice for this correlation between two points $\vec{x}^{(i)}$ and $\vec{x}^{(j)}$ is

$$\mathcal{R}\left(\vec{\theta}, \vec{x}^{(i)}, \vec{x}^{(j)}\right) = \prod_{n=1}^{N} \exp\{-\theta_n \left(x_n^{(i)} - x_n^{(j)}\right)^2\} \tag{2}$$

where the values of $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_N) \in \Re^N$ are to be computed.

The optimal value of $\hat{\mu}$ is given by $\hat{\mu} = \text{argmin}_{\mu}\text{MSE}[\hat{y}(\vec{x})]$, i.e. corresponds to the minimum *MSE* between the predicted and actual values of y; it is given by

$$\hat{\mu} = \frac{\overrightarrow{1}_L^T [R]^{-1} \overrightarrow{y}_L}{\overrightarrow{1}_L^T [R]^{-1} \overrightarrow{1}_L} \tag{3}$$

where $[R]$ is the $L \times L$ correlation matrix for the training patterns and $R_{ij} = \mathcal{R}\left(\vec{\theta}, \vec{x}^{(i)}, \vec{x}^{(j)}\right)$. The estimation function finally becomes

$$\hat{y} = \hat{\mu} + \overrightarrow{r}^T(\vec{x})[R]^{-1}\left(\overrightarrow{y}_L - \overrightarrow{1}_L \hat{\mu}\right) \tag{4}$$

where $\overrightarrow{y}_L = (y_1, y_2, ..., y_L)$, $\overrightarrow{1}_L = (1, 1, ..., 1) \in \Re^L$ and $\overrightarrow{r}^T(\vec{x})$ is the tranpose of the vector of correlations between $\vec{x}$ and any of the training points $\vec{x}^{(i)}$, $i = 1, 2, ..., L$, i.e.

$$\overrightarrow{r}^T(\vec{x}) = \left(\mathcal{R}(\vec{\theta}, \vec{x}^{(1)}, \vec{x}), \mathcal{R}(\vec{\theta}, \vec{x}^{(2)}, \vec{x}), ..., \mathcal{R}(\vec{\theta}, \vec{x}^{(L)}, \vec{x}) \right) \tag{5}$$

The optimal $\vec{\theta}^* = (\theta_1{}^*, \theta_2{}^*, ..., \theta_N{}^*)$ values are computed through the maximum likelihood theory, as

$$\vec{\theta}^* = \text{argmin}_{\vec{\theta}}\{L\ \ln(\sigma^2(\vec{\theta})) + \ln(\det[R(\vec{\theta})])\} \tag{6}$$

where $\det[R(\vec{\theta})]$ is the determinant of the correlation matrix and

$$\sigma^2 = \frac{\left(\vec{y}_L - \overrightarrow{1}_L \hat{\mu}\right)^T [R]^{-1} \left(\vec{y}_L - \overrightarrow{1}_L \hat{\mu}\right)}{N} \tag{7}$$

Eq. 6 is solved using *EAs*. The *MSE* value for a new point $\vec{x}$ is given by

$$s^2(\vec{x}) = \sigma^2\{1 - \overrightarrow{r}^T [R]^{-1} \overrightarrow{r} + \frac{(1 - \overrightarrow{1}_L^T [R]^{-1} \overrightarrow{r})^2}{\overrightarrow{1}_L^T [R]^{-1} \overrightarrow{1}_L}\} \tag{8}$$

With respect to the proposed incremental algorithm for the formation of the starting database, eq. 6 is first solved by considering all the available sample points and an *EA* searches the design space to locate the point with maximum *MSE* i.e. the point with the maximum uncertainty concerning the kriging–based prediction of response. This is the next point to be added to the training set.

5 THE APPROXIMATE EVALUATION TOOL - *RBFNs*

In each generation of the *EAs*, the individuals are evaluated using the exact evaluation tool (i.e. the flow solver, at the nominal flow conditions). However, as described in a previous section, $2N$ more evaluations of slightly perturbed shapes are needed in order to quantify the robustness of the actual design. To this end, each geometrical design variable is perturbed by $\pm\varepsilon$ (ε being a user–defined, small valued, fixed quantity). The $2N$ perturbed airfoil shapes are evaluated using *RBFNs*. For each new individual created by the *EA*, a single local *RBFN* is trained, based on its closest entries in the database. All $2N$ evaluations are carried out using the same *RBFN*. *RBFNs* [10] are described below.

An *RBFN*, fig. 2, is a three layer network which performs a non-linear mapping from the input layer to the hidden layer ($\Re^N \rightarrow \Re^M$) and a linear mapping from the hidden layer to the output layer ($\Re^M \rightarrow \Re^1$) which produces the network response. The hidden units are associated with the so-called *RBF* centers, $\vec{c}^{\,(i)} \in \Re^N, i = 1, M$. The choice $M = P$ (P is the training set size) and $\vec{c}^{\,(i)} = \vec{x}^{\,(i)},\ i = 1, P$, where $\vec{x}^{\,(i)}$ are the training patterns, allows the *RBFN* to perform exact interpolations.

The hidden unit value $h_m^{(t)}$ which corresponds to input $\vec{x}^{\,(t)}$ is computed through a nonlinear expression which is based on the distance between $\vec{x}^{\,(t)}$ and $\vec{c}^{\,(m)}$, namely

$$h_m^{(t)} = \Phi\left(\left\|\vec{x}^{\,(t)} - \vec{c}^{\,(m)}\right\|_2, r_m\right) \tag{9}$$

where r_m stands for the radius associated with the $m - th$ center. The nonlinear function Φ performs the mapping $\Re^N \rightarrow \Re$. A possible choice is $\Phi(u, r) = \exp(-u^2/r^2)$. The network output is given by $\sum_{i=1}^{M} w_m\ h_m^{(t)}$. The presentation of the P input patterns to the network (along with their responses), results to the formation of a linear system of P equations, the unknowns being the weights w_m associated with the links between the hidden and output units. Any direct inversion method can be used for the computation of the unknown weights.

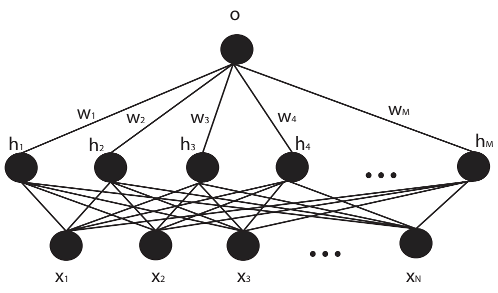

Figure 2: Radial Basis Function Network.

6 APPLICATION: ROBUST DESIGN OF A COMPRESSOR AIRFOIL

The robust design of a compressor cascade airfoil is presented. We are concerned with the design of an airfoil which yields minimum total pressure losses at nominal flow conditions: inlet flow angle $a_1 = 50^o$, inlet Mach number $M_1 = 0.27$, chord–based Reynolds number $Re_c = 4 \times 10^5$. The flow is characterized by non–unit axial velocity density ratio, namely $AVDR = 1./0.93$.

As said before, the total pressure loss coefficient is used as the main (first) objective. Its value is computed by the integral boundary layer method and penalized to account for the minimum thickness constraint ($0.072C$, C is the chord) and the minimum flow turning requirement (34^o which corresponds to maximum outlet flow angle equal to 16^o). A candidate airfoil violating any of the constraints is given a penalty factor. This is expressed as an exponential function of the deviation between the current and threshold values. The exponential function is used only for slight and mild constraint violations; the death penalty method is employed for shapes which violate "considerably" any of the constraints (minimum thickness less than $0.067C$, turning less than 34^o).

The shape of the airfoil pressure side was fixed; the $N = 8$ design variables used correspond to the ordinates of the eight Bezier control points used to parameterize the suction side, fig. 3.

The second objective corresponds to the standard deviation of the estimated total pressure losses of the $2N$ perturbed airfoils plus the current one ($2N + 1$ in total); it is used as a measure of the curvature of the response surface close to the current airfoil. In the Pareto front, fig. 4, computed using the two–objective *EA*, the abscissa (marked with "1/robustness") quantifies the second objective and stands for the standard deviation of losses (minus the least squares estimation, see comments in section 2) computed at $2 \times 8 + 1 = 17$ points in the vicinity of any Pareto optimal solution.

The initial database consists of $L_{DB} = 275$ airfoils, selected using the kriging model in accordance with the proposed sampling algorithm. The EA uses $\mu = 50$ parents and $\lambda = 200$ offspring. The archival set $S^{g,\alpha}$ was allowed to keep up to 50 elite members. Fig. 5 shows the evolution of the optimal front population. The crossover probability was set to 95% and the mutation probability to 3.5%.

During the evolutionary search, for each new candidate solution, the number of neighbors used to train the local $RBFN$ was equal to 15 (plus the current point, so $P = 15 + 1 = 16$). The $RBFN$ training required the solution of a 16×16 symmetrical linear problem which can be carried out with almost negligible computing cost.

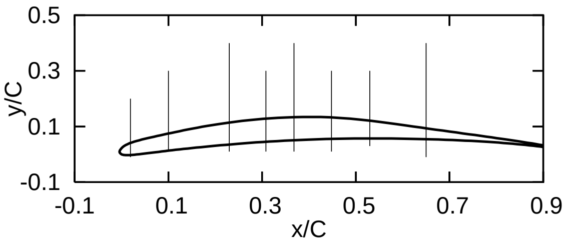

Figure 3: Parameterization of the airfoil suction side using Bezier polynomials; the lower and upper bounds of each design variable are shown (vertical segments). The airfoil is parameterized at zero stagger angle.

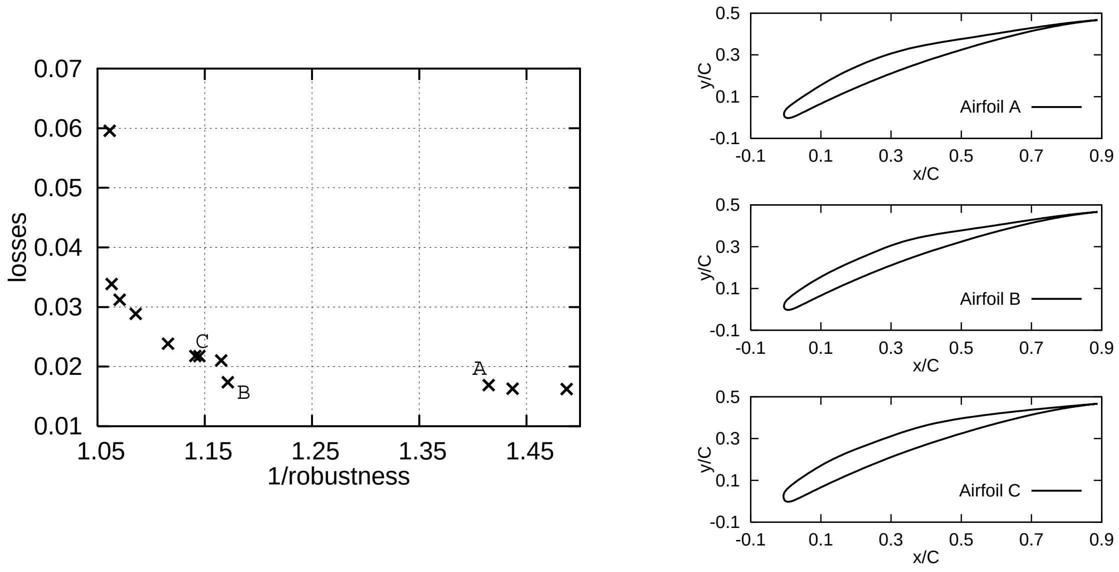

Figure 4: In the Pareto front, left, the abscissa is inversely proportional to the robustness of the design; small values correspond to very robust airfoils. Three Pareto optimal solutions, corresponding to three optimally performing airfoils are marked with A, B and C. For instance, compared to A, C yields higher total pressure losses at the nominal conditions but with smaller performance deterioration under manufacturing or assembling errors. The contours of these three airfoils are also plotted (right). The performance of the three airfoils is analyzed, in more details, in fig. 6.

Figs. 4 and 6 summarize the obtained results (see comments in their captions). Three optimal airfoils (A, B and C), members of the Pareto front, are further analyzed. To comment on their performance, they were perturbed and evaluated using the integral method (instead of the $RBFN$ utilized during the optimization). The comparison of A and B follows: both have almost equal total pressure losses. Their contours present a lot of similarities, fig. 4, with some discrepancies at about $0.35C$. Despite the fact that the Bezier polynomials are global parameterization tools, we may assume that this difference is mostly attributed to the third and fourth control points; the design variables x_3^A and x_3^B (similarly, x_4^A and x_4^B) are far apart. These two airfoils have, however, quite different robustness. For instance, from fig. 6, it is clear that the reduction in x_3^A leads to a noticeable increase in losses, which is not the case in airfoil B. So, by neglecting the small difference in losses between A and B, B (instead of A) should preferably be selected.

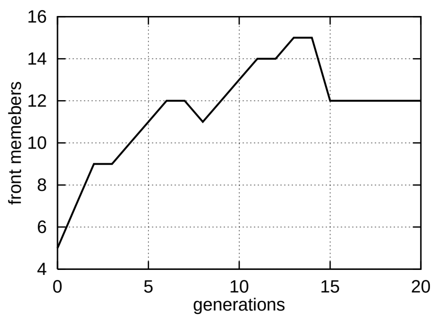

Figure 5: The number of optimal front members stored at each generation.

On the other hand, airfoil B is thinner than C and yields lower losses. However, the robustness of B is lower than that of C. According to fig. 6, the curves corresponding to some design variables of B (x_4^B, x_5^B and especially x_6^B) are not that flat as the corresponding curves of C.

7 CONCLUSIONS - COMMENTS

A method for the robust design of turbomachinery blade airfoils was presented and demonstrated. In the present application, a single–objective problem is transformed to a two–objective one in which the second objective is a measure of the airfoil performance robustness. In real, K–objective problems ($K > 1$), one might either duplicate the number of objectives (i.e. each real objective will give rise to an additional objective related to its robustness with respect to stochastic deficiencies in the design variables) or utilize a single robustness for all objectives. Depending on this choice, a problem with $2K$ or $K+1$ objectives should be solved.

The use of an approximate evaluation model for the perturbed shapes aims at reducing the optimization cost. *RBFNs* have been used for this purpose; however, other known metamodels can be used instead. For instance, the evaluation of perturbed geometries could be carried out through kriging, i.e. the same tool previously used to create the initial database. The reason for not using kriging for the approximate evaluations is that the *RBFNs* are faster; the extra *CPU* cost of kriging is affordable if *MSE* needs to be computed, which is not the case herein. Note that, under certain assumptions, it could be readily shown that kriging and *RBFNs* are two equivalent, distance–based interpolation models, [11].

This paper also presents a new sampling method based on the kriging model. Starting from a "basic" sampling, extra points are identified over the search space, at locations where kriging operates with maximum uncertainty. The proposed algorithm should be considered as a problem–customized sampling technique, in the sense that each new sample is added using criteria affected by the application in hand.

REFERENCES

[1] K.C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38:43–76, 2002.

[2] A. Jameson. Optimum aerodynamic design using cfd and control theory. AIAA Paper 95-1729-CP, 1995.

[3] R.W. Walters and L. Huyse. Uncertainty analysis for fluid mechanics with applications. NASA/CR-2002-211449, ICASE Report No.2002-1, 2002.

[4] C. Poloni. Robust design of aircraft components: A multi-objective optimization problem. Von-Karman Institute Lecture Series 2004-07, November 2004.

[5] M. Drela and M.B. Giles. Viscous–inviscid analysis of transonic and low reynolds number airfoils. *AIAA Journal*, 25 (10):1347–1355, 1987.

[6] D. Montgomery. *Design and Analysis of Experiments.* Wiley, 1997.

[7] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. TIK-Report No. 43, Computer Engineering and Communication Networks Lab, ETH Zurich, 1998.

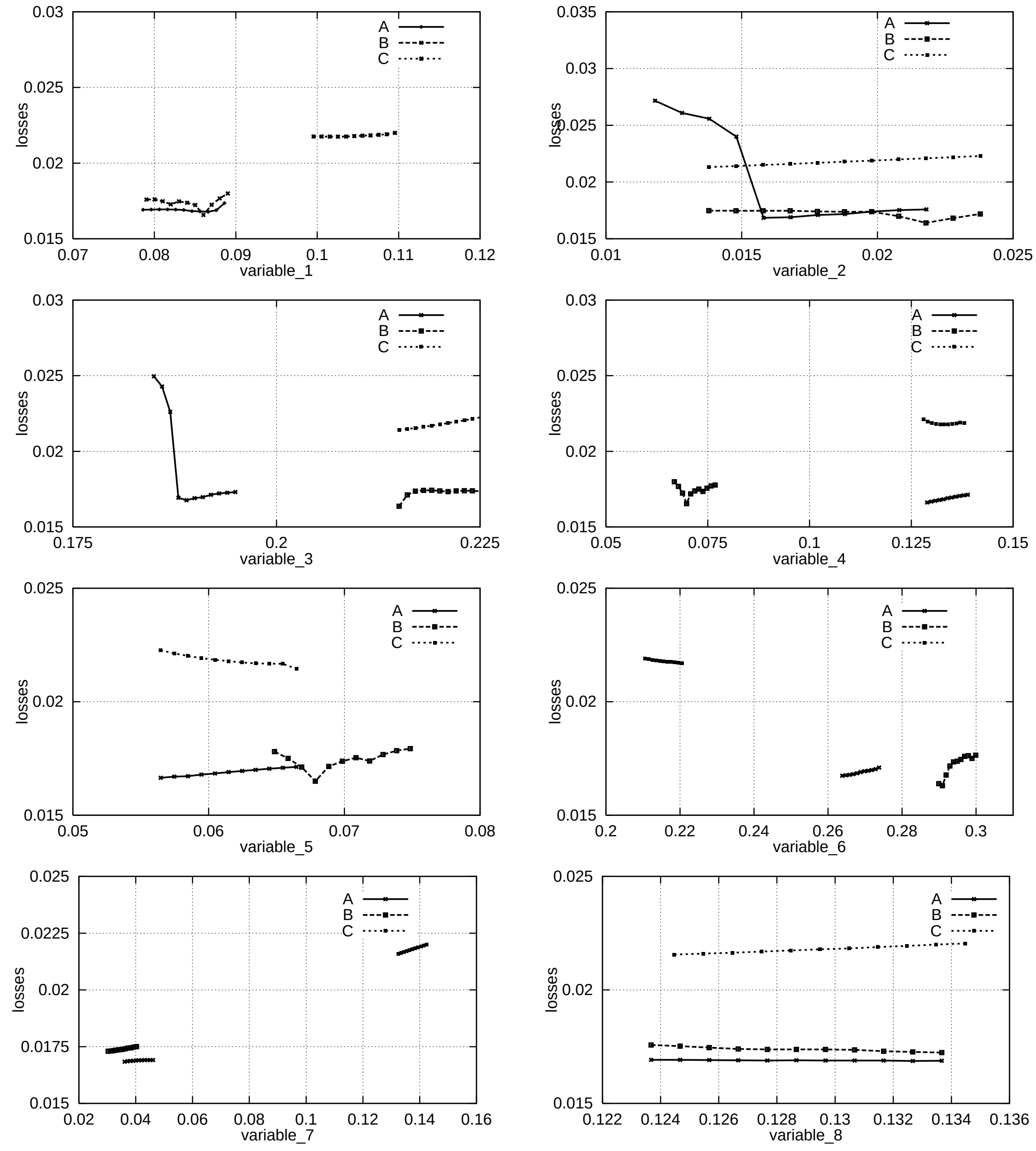

Figure 6: For each of the three optimal airfoils of fig. 4, marked with A, B and C, the integral boundary layer method was used to evaluate its robustness when perturbing one design variable at a time. Starting from the optimal airfoil, say A, with design variables $x_i^A,\ i=1,2,...,8$, eleven airfoils have been generated by varying x_i around x_i^A ($x_i^A + j\Delta x,\ j=-5,\ldots,5$ with $\Delta x = 0.001C$), while at the same time any other design variable was kept constant. All eleven airfoils were evaluated using the integral boundary layer method. It is evident that the sixth point on each curve refers to the corresponding optimal airfoil (A, B or C).

[8] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2 improving the Stength Pareto evolutionary algorithm. TIK-Report No. 103, Computer Engineering and Communication Networks Lab, ETH Zurich, 2001.

[9] I. Clark. *Practical Geostatistics.* Applied Science Publishers, Essex, 1979.

[10] S. Haykin. *Neural Networks.* Prentice Hall International, Inc., 2nd edition, 1998.

[11] L. Wilmes, T. Bäck, Y. Jin, and B. Sendhoff. Comparing neural networks and kriging for fitness approximation in evolutionary computation. Proc. IEEE Congress on Evolutionary Computation, 3, 1910-1917, 2003.