11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

PCA–ENHANCED METAMODEL–ASSISTED EVOLUTIONARY ALGORITHMS FOR AERODYNAMIC OPTIMIZATION

VARVARA G. ASOUTI, STYLIANOS A. KYRIACOU[1] AND KYRIAKOS C. GIANNAKOGLOU

National Technical University of Athens, Parallel CFD & Optimization Unit
Iroon Polytechniou 9, 15780, Athens, Greece
e-mail: vasouti@mail.ntua.gr, stelios.kyriacou@gmail.com, kgianna@central.ntua.gr

Key words: Evolutionary Algorithms, Metamodels, Principal Component Analysis, Optimization, Aerodynamics

Abstract. This paper deals with evolutionary algorithms (EAs) assisted by surrogate evaluation models or metamodels (Metamodel–Assisted EAs, MAEAs) which are further accelerated by exploiting the Principal Component Analysis (PCA) of the elite members of the evolving population. PCA is used to (a) guide the application of evolution operators and (b) train metamodels, in the form of radial basis functions networks, on patterns of smaller dimension. Compared to previous works by the same authors, this paper also proposes a new way to apply the PCA technique. In particular, the front of non–dominated solutions is divided into sub–fronts and the PCA is applied "locally" to each sub–front. The proposed method is demonstrated in multi–objective, constrained, aerodynamic optimization problems.

1 INTRODUCTION

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

[1]Currently with: Andritz HYDRO AG, Zürich, Switzerland

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

On the other hand, the most common technique to reduce the CPU cost of the EA–based optimization is the extensive use of surrogate evaluation models (or metamodels). They are used to inexpensively approximate the objective function value(s), by replacing the call to the costly problem–specific evaluation model. Metamodels, after being trained on previously–seen solutions, can be incorporated into an EA in different ways, depending on whether their training takes place during (on–line), [7, 14, 11, 8] or separately from (off–line) the evolution, [3, 13]. In this paper, Metamodel–Assisted EAs (MAEAs) with on–line trained metamodels, [6, 7], are employed. According to the inexact pre–evaluation (IPE) approach, with the exception of a few starting generations, all population members are approximately evaluated using local metamodels trained on the fly, separately for each one of them. Then, a few of them, practically the most promising among them as indicated by the metamodel, are re–evaluated on the exact model.

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

A way to overcome the overall performance degradation, often referred to as the curse of dimensionality, is to decrease the problem dimension via dimension reduction techniques. This can be done via principal component analysis (PCA) techniques, [5]. In contrast to other methods in which PCA is exclusively used to reduce the dimension of the optimization problem, here PCA is used both to guide the application of the evolution operators (without dimension reduction) and reduce the number of sensory units of the Radial Basis Function (RBF) networks used as metamodels, [9, 10]. Compared to [9, 10], there is a novel enhancement of the already published method. PCA is not applied to the entire elite set but to sub–fronts, after appropriately splitting the set of elite members into parts. According to this scheme, each parent is transformed using information from the PCA of the closest in the design space sub–front. By doing so, the EA requires less evaluations compared to [9]. The proposed method is demonstrated in three cases concerned with the aerodynamic design of an isolated airfoil, the aeroelastic design of a wind turbine blade and the preliminary design of a supersonic business jet.

2 PCA–ENHANCED EAs AND MAEAs

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

next largest variance and so forth.

Without loss in generality, let us assume that a multi–objective optimization (MOO) problem is to be solved. In each generation, the PCA of the elite set is carried out. In single–objective optimization (SOO) problems, instead of the front of non–dominated solutions, the elite set can be formed by the current optimal and a few near–optimal solutions.

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

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

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

$$P_{N\times N} = U \Lambda U^T \tag{2}$$

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

Compared to [10], in this paper the front of the non–dominated solutions is divided into a user–defined number of sub–fronts (fig. 1) and the PCA is applied locally to each sub–front.

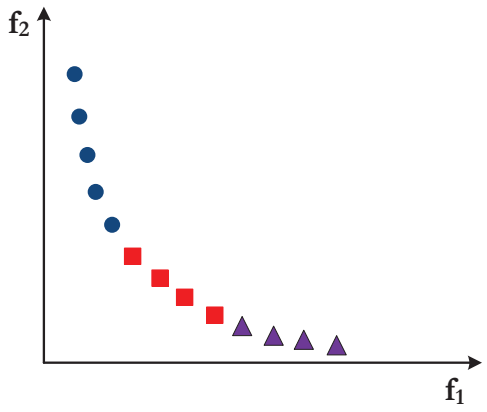

Figure 1: Schematic representation of the division of the non–dominated individuals into three sub–fronts.

The two ways of taking advantage of PCA are briefly described below. Section 2.1 describes how the principal components can be used to drive the evolution operators. Section 2.2 describes how the PCA is used to reduce the number of sensory nodes of the metamodel. The combined use of these two options is straightforward.

2.1 PCA–enhanced evolution operators

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

$$\vec{e}_i = U(\vec{x}_i - \vec{\mu}_X) \tag{3}$$

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

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

$$p_m^i = \alpha p_m + (1-\alpha) p_m \frac{N}{D_V} \cdot \frac{V_{max} - V_i}{V_{max} - V_{min}} \tag{4}$$

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

$$D_V = \sum_{i=1}^{N} \frac{V_{max} - V_i}{V_{max} - V_{min}} \tag{5}$$

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

$$\vec{x}_i = U^{-1}\vec{e}_i + \vec{\mu}_X \tag{6}$$

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

2.2 EA with PCA–assisted metamodels

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

The eigenvectors included in matrix U, eq. 2, are associated with the variances of the design variables and can be used to identify the directions where the elite members are less or more scattered (high variance indicates scattered data whereas small variance corresponds to less scattered data). Based on this, a user–defined number of RBF network sensory units, in fact those corresponding to the directions of the design space with high variances, are filtered out. Consequently, the RBF network is trained on lower dimension data and this turns out to yield more reliable networks, trained at lower cost.

This truncation applies only during the RBF network training. In particular, during the IPE phase, for all population members, a local metamodel is built by selecting training patterns following the procedure described in detail in [8]. The training patterns are rotated (eq. 3) and, then, their components associated with the higher eigenvalues are excluded from the training.

3 APPLICATIONS

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

3.1 Preliminary design of a supersonic business jet

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

In this application, computations based on standard MAEA and M(PCA)AEA(PCA) in the two variants of the latter in which PCA is driven by either the entire elite front or two sub–fronts, have been carried out and their results are compared. In all three runs, a $(\mu, \lambda) = (30, 90)$ MAEA (with μ parents and λ offspring) was used and the metamodel–based IPE phase started once the database (DB) recorded the first 100 entries (already evaluated individuals). Then, in each subsequent generation, 5 to 10 individuals were re–evaluated on the problem–specific tool. A stopping criterion of 2000 evaluations was imposed. A comparison of the obtained fronts of the three methods is shown in fig. 2. In either form, the M(PCA)AEA(PCA) outperforms the standard MAEA. Furthermore, as expected, the local application of PCA performs better compared to the use of a single PCA of the entire elite set.

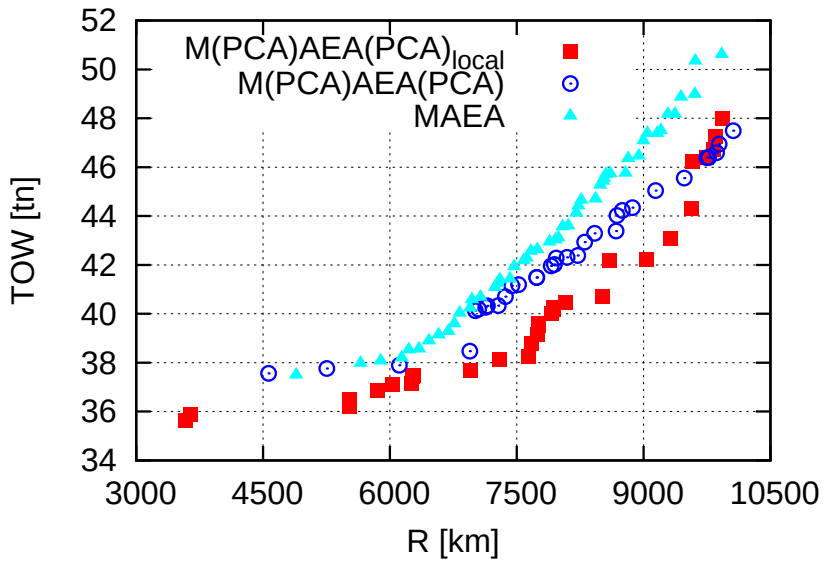

Figure 2: Preliminary design of a supersonic business jet: Comparison of the fronts of non–dominated solutions computed using MAEA (filled triangles), M(PCA)AEA(PCA) (empty circles) and M(PCA)AEA(PCA)$_{\text{local}}$ (filled squares) at the cost of 2000 evaluations.

3.2 Aeroelastic design of a wind turbine blade

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

$$M_{f,\text{root}} \leq 10000\text{kNm} \ , \quad \sigma_{f,\max} \leq 64000\text{kN/m}^2 \ , \quad \sigma_{e,\max} \leq 90000\text{kN/m}^2 \tag{7}$$

For each blade section an I–beam structural model is assumed, fig. 3, using two spar caps joined together by a shear web. The design variables are the blade chord, twist

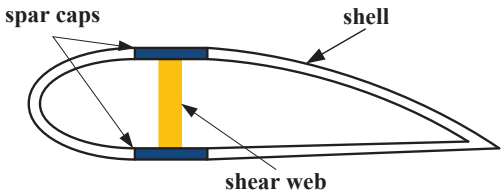

Figure 3: Aeroelastic design of a wind turbine blade: Schematic representation of the I–Beam structural model for each blade section.

and thickness, the I–beam base length, the spar caps, shear web and shell thickness at 9 preselected spanwise positions. Targeting realizable designs, the objective functions are also constrained, as follows

$$AEP \geq 10\text{GWh} \ , \quad m \leq 20\text{tn} \tag{8}$$

The evaluation of each candidate solution was based on the Blade Element Momentum model included in NTUA's aeroelastic software GAST (General Aerodynamic and Structural numerical Tool, [12]). This case was studied using only the M(PCA)AEA(PCA) using $(\mu, \lambda) = (30, 90)$ and a termination criterion of 10000 evaluations. The metamodel was activated once 500 feasible (non–violating the constraints) solutions were stored in the DB and during the IPE phase, 9 to 15 individuals were re–evaluated on the GAST software. The resulted fronts of non–dominated solutions are compared in fig. 4, where in this case too the local application of PCA is proved to yield better solutions.

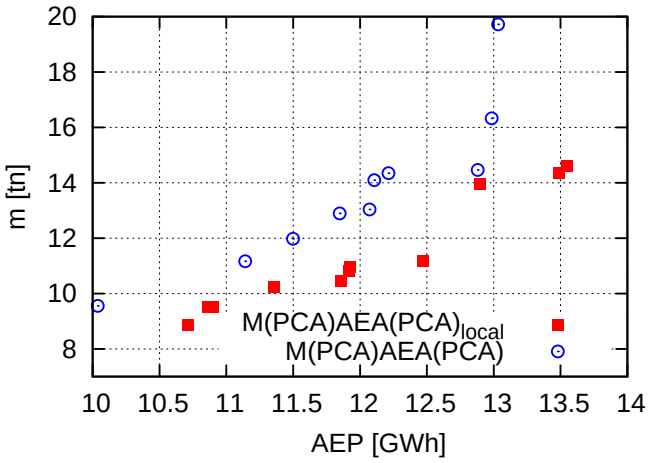

Figure 4: Aeroelastic design of a wind turbine blade: Comparison of the fronts of non–dominated solutions computed using M(PCA)AEA(PCA) (empty circles) and M(PCA)AEA(PCA)$_{\text{local}}$ (filled squares) at the same CPU cost.

3.3 Design of an isolated airfoil

The last case is a two–objective, constrained, optimization problem concerned with the design of an isolated airfoil for minimum drag (C_D) and maximum lift (C_L) coefficient. The flow conditions are: free–stream Mach number $M_\infty = 0.3$, flow angle $a_\infty = 4^o$ and $Re_c = 6.2 \times 10^6$. The airfoil is parameterized using two Bézier curves, separately for the pressure and suction sides, with 8 control points each. Only the internal control points of each Bézier curve may vary, summing up to 24 design variables.

The tool used to evaluate each individual applies a viscous–inviscid flow interaction method based on an integral boundary layer method, coupled with an external flow solver,

[4]. To prevent the formation of unacceptably thin airfoils, geometrical constraints on the airfoil thickness $t(x)$ at three chordwise positions (x), are imposed, namely

$$t(0.25c) \geq 0.12c, \;\; t(0.5c) \geq 0.12c, \;\; t(0.75c) \geq 0.05c \tag{9}$$

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

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

In both cases, a stopping criterion of 1500 evaluations was imposed. A $(\mu, \lambda) = (20, 60)$ MAEA is used and metamodels were applied once 300 feasible solutions were stored in the DB. During the IPE phase, in each generation, 5 to 8 individuals were re–evaluated on the integral boundary layer method.

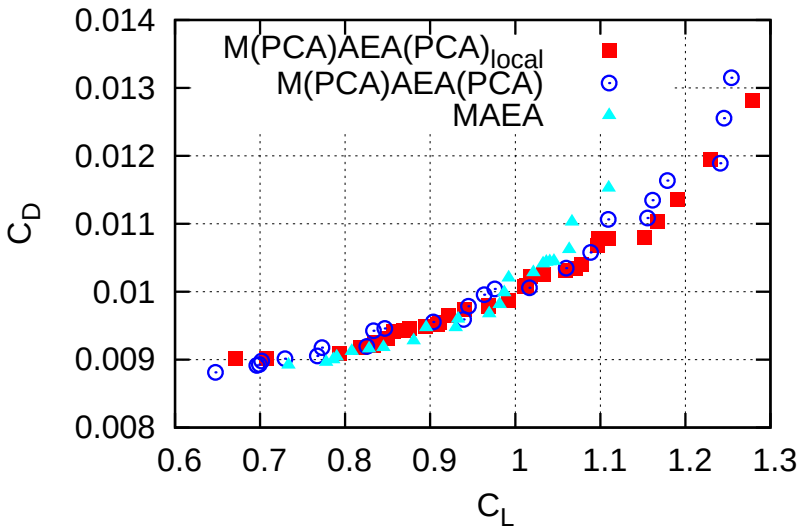

Figure 5: Optimization of a 2D isolated airfoil: Comparison of the computed fronts of non–dominated solutions computed using MAEA (filled triangles), M(PCA)AEA(PCA) (empty circles) and M(PCA)AEA(PCA)$_{\text{local}}$ (filled squares).

Figure 5 compares fronts of non–dominated solutions computed by the three methods at the same cost, practically the same number of evaluations. M(PCA)AEA(PCA) was able to capture parts of the Pareto front which MAEA couldn't. Though, in the low lift–low drag region, MAEA computed some individuals dominating those of M(PCA)AEA(PCA), in the high lift–high drag region MAEA's performance was very poor. Regarding the two variants of M(PCA)AEA(PCA), individuals resulted from the local application of PCA were better spread along the front and the majority of them dominated those from PCA on the entire elite set. Three airfoil shapes from each sub–front, i.e. the low lift–low drag and high lift–high drag ones are shown in figs. 6 and 7 respectively. From the presented airfoil shapes, the two "families" can easily be distinguished. As shown in fig. 5, this sub–division of the elite front is beneficiary for the search method.

Figure 6: Optimization of a 2D isolated airfoil: Three airfoil shapes corresponding to the low lift–low drag part of the Pareto front (first sub–front) shown in fig. 5. From left to right the three airfoils yield the following C_L, C_D values: $(0.707, 0.00901)$, $(0.817, 0.00918)$ and $(0.894, 0.00948)$.

Figure 7: Optimization of a 2D isolated airfoil: Three airfoil shapes corresponding to the high lift–high drag part of the Pareto front (second sub–front) shown in fig. 5. From left to right the three airfoils yield the following C_L, C_D values: $(0.968, 0.00978)$, $(1.096, 0.010678)$ and $(1.19, 0.01136)$.

4 CONCLUSIONS

This paper reconfirms the superiority of the so–called M(PCA)AEA(PCA) algorithm, originally proposed and assessed in [10], using three test problems. In each generation of the EA, the analysis of the population members using information related to the principal directions, as extracted by the characteristics of the current elite individuals, is used to (a) reduce the dimensionality of the RBF networks, by making their training easier and their predictions more dependable and (b) apply the evolution operators in a properly transformed space. In addition, this paper demonstrated the increase in performance offered by the local application of the PCA–driven actions during a M(PCA)AEA(PCA) run. Locality is related to the splitting of the current front of non–dominated solutions into sub–fronts, in each of which an independent PCA is performed. The better performance of the proposed scheme is justified by the fact that candidate solutions at the different edges

of a Pareto front may have very different characteristics, the "averaging" of which through the single PCA could be problematic. In this paper, restricted to two–objective problems with constraints, the current front of non–dominated solutions was decomposed into two sub–fronts, using its dynamically changing median. On–going research is related to the automation of the whole process and criteria for splitting the current front, including optimization problems with more than two objectives.

ACKNOWLEDGMENT

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

REFERENCES

[1] V.G. Asouti and K.C. Giannakoglou. Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes. *Engineering Optimization* (2009) **41**:241–257.

[2] S. Axler. *Linear Algebra Done Right*. Springer Verlag, (1997).

[3] D. Büche, N. Schraudolph and P. Koumoutsakos. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics – Part C:Applications and Reviews* (2005) **35**:183–194.

[4] M. Drela and M.B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal* (1987) **25**:1347–1355.

[5] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, (1999).

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

[7] M.K. Karakasis, A.P Giotis and K.C. Giannakoglou. Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization. *International Journal for Numerical Methods in Fluids* (2003) **43**:1149–1166.

[8] M.K. Karakasis and K.C. Giannakoglou. On the use of metamodel-assisted, multi-objective evolutionary algorithms. *Engineering Optimization* (2006) **38**:941–957.

[9] S.A. Kyriacou, S. Weissenberger and K.C. Giannakoglou. Design of a matrix hydraulic turbine using a metamodel-assisted evolutionary algorithm with PCA–driven evolution operators. *International Journal of Mathematical Modelling and Numerical Optimization* (2012) **3**:45–63.

[10] S.A. Kyriacou, V.G. Asouti and K.C. Giannakoglou. Efficient PCA–driven EAs and metamodel–assisted EAs, with applications in turbomachinery. *Engineering Optimization* (2014), DOI 10.1080/0305215X.2013.812726.

[11] Y.S. Ong, K.Y. Lum, P.B. Nair, D.M. Shi and Z.K. Zhang. Global convergence of unconstrained and bound constrained surrogate-assisted evolutionary search in aerodynamic shape design. *The 2003 Congress on Evolutionary Computation* (2003) **3**:1856–1863.

[12] V.A. Riziotis and S.G. Voutsinas. Gast: A General Aerodynamic and Structural Prediction Tool for Wind Turbines. *Proceedings of the EWEC–97* (1997), Dublin, Ireland

[13] W. Shyy, N. Papila, R. Vaidyanathan and K. Tucker. Global design optimization for aerodynamics and rocket propulsion components *Progress in Aerospace Sciences* (2001) **37**:59–118.

[14] H. Ulmer, F. Streichert and A. Zell. Evolution strategies assisted by Gaussian processes with improved pre-selection criterion. *The 2003 Congress on Evolutionary Computation* (2003) **1**:692–699.