ECCOMAS Computational Fluid Dynamics Conference 2001
Swansea, Wales, UK, 4-7 September 2001

EFFICIENT GENETIC OPTIMIZATION USING INEXACT INFORMATION AND SENSITIVITY ANALYSIS. APPLICATION IN SHAPE OPTIMIZATION PROBLEMS

Marios K. Karakasis, Alexios P. Giotis, Kyriakos C. Giannakoglou

National Technical University of Athens, P.O. Box 64069, Athens 157 10, GREECE,
Tel: (30)-1-772.16.36, Fax: (30)-1-772.37.89, E-mail: kgianna@central.ntua.gr

Key words: Optimization, Genetic Algorithms, Distributed Genetic Algorithms, Radial Basis Function networks.

Abstract. *Despite its robustness, the design and optimization of aerodynamics shapes using Genetic Algorithms and Computational Fluid Dynamics tools suffers from high computing cost. A remedy to this problem is to replace part of the exact and thus costly evaluations with cheaper, though inexact, ones. The method proposed herein employes an inexact pre–evaluation phase at each new generation, based on properly trained radial basis function networks, to pin-point the most promising individuals which will solely undergo exact evaluations. The method is enhanced through sensitivity analysis, by introducing the so–called importance factors, computed from and used by the neural networks, in an autocatalytic way. The inexact pre–evaluation concept is also extended to Distributed Genetic Algorithms and assessed using a number of test problems.*

1 INTRODUCTION

From the aerodynamic point of view, the design of single- or multi-element airfoils is usually based on desirable pressure distributions defined over their contours – which are likely the outcome of inverse boundary layer computations – or on the maximization of their aerodynamic performance. The latter may call for the maximization of lift, minimization of drag and other relevant requirements.

During the last decade, probabilistic search–optimization algorithms have gained particular attention in the fields of aeronautics and turbomachinery. Among them, Genetic Algorithms (*GAs*, [1], [2]) are the most frequently used optimization tool. *GAs* are robust enough and capable to accommodate any commercial or in–house evaluation software (a Computational Fluid Dynamics, *CFD*, solver for the aerodynamic analysis of airfoils or, perhaps, a finite–element solver for their structural analysis, etc.) without the slightest modification. Provided that the involved parameters have been chosen carefully, *GAs* are very effective search algorithms without getting stuck to local minima that usually appear in complex multi-modal problems. However, a well–known drawback of *GAs* is the high number of evaluations they require to reach the optimum solution. Since, in aerodynamic shape optimization problems, the evaluations rely on *CFD* codes bearing noticeable computing cost, the cost of *GA*–based design is high enough and research is directed toward the reduction of the required number of evaluations.

In their previous works, these authors introduced and demonstrated the efficiency of coupling *GAs* with the so–called Inexact Pre–Evaluation (*IPE*) phase ([3], [4], [5], [6], [7]) as a remedy to the aforementioned problem. The concept is quite simple: in the course of the genetic evolution, each and every candidate solution that has been evaluated through the costly *CFD* tool is kept in a database and used selectively to forecast the merit or cost of new candidate solutions during the forthcoming generations. For each new individual, local Radial Basis Function (*RBF*) networks are trained on the previously seen solutions and provide inexact fitness scores at almost negligible computing cost. The *RBF* networks are used along with the so–called importance factors (*IFs*), i.e. intrinsically computed measures of the sensitivity of the objective function with respect to the design parameters. More details about this method, which will be referred to as *GA-IPE-IF*, can be found in the previously cited references, though a brief summary is provided in the next section as well.

The goal of this paper is twofold. First, to demonstrate the efficiency of the proposed method in a number of test cases and, second, to extend this concept to the Distributed Genetic Algorithms (*DGAs*, [8]). Through a number of selected test problems, a fair comparison between *GA-IPE-IF* and its distributed variant $D(GA-IPE-IF)$ is attempted.

2 THE *GA-IPE-IF* TECHNIQUE

2.1 The *IPE* concept

Within each generation, the purpose of the *IPE* phase is to pre–evaluate the entire population using approximate tools, so as to distiguish the most promising individuals among them. It is evident that the pre–evaluation phase requires a very fast surrogate evaluation tool. As such *RBF* networks ([9], [10]) are used. For each individual in the current generation, an *RBF* network is formed and used to provide an estimation of its fitness. The network is trained locally, using the closest entries in the database, which is continuously enriched during the genetic evolution. The pre–evaluated population members are ranked using their approximate fitness scores. Only a small percentage of them, namely the most promising ones, will be re–evaluated through the costly *CFD* tool. In this manner, less fit individuals evade undergoing exact evaluations and this contributes to the reduction of the total computing cost.

The small size of training sets, used for each *RBF* network, makes the training procedure of virtually no computing cost, since it calls only for the inversion of a small matrix. Thus, the *IPE* computational cost could safely be neglected.

On the other hand, since in many optimization problems, not all of the design variables are of equal importance, less important parameters introduce a kind of "noise" during the training of *RBF* networks which may damage the quality of the network prediction. The problem can be circumvented, to some extent, by the use of importance factors (*IFs*), i.e. quantities that will be denoted by $I_m,\ m=1,M$, where M is the number of design variables. Their role is autocatalytic, since they can be considered as a by-product of the *RBF* networks' training whereas, in turn, affect any subsequent training and use of *RBF* networks. The values of *IFs* are not known a priori, but they are dynamically computed during the evolution of the *GA*. Initially, they all take on the same value

$$I_m = 1/M, \quad m = 1, M \tag{1}$$

so that $\sum_{i=1}^{M} I_m = 1$. As the *GA* evolves, *IFs* are regularly computed so that higher *IF* values are associated with the most important design variables. A trained network may guess the fitness function value as well as its derivatives with respect to the design variables, using inexpensive post–processing. The basic formula for their calculation is the following:

$$I_m = \frac{\left|\frac{\partial y^{(b)}}{\partial x_m}\right|}{\sum_{i=1}^{M}\left|\frac{\partial y^{(b)}}{\partial x_i}\right|} \tag{2}$$

where y stands for the fitness function and the subscript (b) denotes that computation at the current best solution should be carried out. A link to previous *IF* values is ensured through the use of a relaxation factor. For a detailed discussion on this issue, one can resort to [5] and [7]. It would be useful to clarify that a design parameter with a low I_m

value at one design point could possess a higher importance elsewhere in the search space and vice–versa.

2.2 The *GA-IPE-IF* Algorithm

After the short description of the *IPE* concept and the use of *IFs*, the implementation of the *GA-IPE-IF* scheme, with population size equal to N_{pop}, can be outlined as follows:

Phase 1: The starting population keeps evolving for a few generations, using exact evaluations and data-results of these computations are all stored in the database. All of the importance factors are given initial values, eq. 1.

Phase 2: During the subsequent generations and for each individual:

- (2a) A local *RBF* network is trained, using the most recent *IF* values.
- (2b) The trained *RBF* network provides an inexact cost function value for this individual.

Phase 3:
- (3a) The σN_{pop}, $(0 < \sigma < 1)$, best individuals in the pre–evaluated generation undergo exact evaluations using the *CFD* tool; the database is further enriched.
- (3b) The I_m values are updated, as previously exposed, each time a new global optimum is computed.

Phase 4: N_{pop} new offspring are created by means of the well-known genetic operators using mixed up exact and inexact fitness scores. Phases 2 to 4 are repeated up to the final convergence.

3 THE *GA-IPE-IF* TECHNIQUE IMPLEMENTED AS A *DGA*

The above described concept of *IPE* can be readily ported to a Distributed Genetic Algorithm (*DGA*). The latter consists of multiple *GAs*, the populations of which can be referred to as demes, which evolve independently and exchange information regularly by performing a migration cycle ([8]). Hence, the migration procedure acts as a barrier that synchronizes the genetic evolution of demes. The *IPE-IF* scheme can be applied to each of the constituent *GAs*, which perform a pre–evaluation of their own deme. However, all demes share a common database to store results from the exact evaluations and to retrieve patterns for training the *RBF* networks. For this reason, we use to denote the proposed scheme by $D(GA - IPE - IF)$.

To achieve maximum efficiency and flexibility, the algorithm has been implemented as a multi-thread application, with each *GA* evolving in its own thread taking thus, advantage of the inherent concurrency of *DGAs*. All the constituent *GA* threads asynchronously post requests for exact evaluations to a server thread. The latter controls a number of slave processes, using the PVM interface, which usually run on an available computer and evaluate the candidate solutions.

As far as migration is concerned, different scenarios have been tested. While the emigrants are always the best members of a deme, the replaced individuals in the host deme may be either the worst ones or some randomly chosen individuals. In the latter case, one can exclude the one or two best individuals of the host deme from replacement letting thus genetic operators decide about the survival of the fittest.

4 METHOD APPLICATION

The present section aims at demonstrating the indisputable superiority of the *GA-IPE-IF* scheme over the conventional *GA* and to compare it with its $D(GA-IPE-IF)$ variant. For a more detailed evaluation of the *GA-IPE-IF* method one could refer to [7].

The method has been applied to two design problems. For the first one, a simple flow model, the panel method, [11], for incompressible, irrotational airfoil flows was used. Though such an evaluation tool is very fast and the use of the *GA-IPE-IF* technique is redundant, the panel method was used, since it allows an adequate number of runs, in order to verify the results and "tune" the algorithm. For the second test-case a time–marching Euler equations' solver for unstructured grids, [12], was used. Both cases have been cast in the form of minimization problems.

4.1 Reconstruction of the *NACA 4412* profile

The aim of this test-case was to design an airfoil that yields a given pressure distribution, at certain flow conditions. As target, the pressure coefficient of the *NACA 4412* profile at zero incidence was set, calculated also by the panel method.

The airfoil shape was parameterized by two Bezier curves, one for each side. The two of the six control points of each curve were fixed on the leading and the trailing edges. The abscissas of the leading edge and of the closest to it Bezier control point were equal, on both curves, in order to achieve a rounded edge. Therefore, the free design parameters were $2 \cdot (1 \cdot 1 + 3 \cdot 2) = 14$.

The purpose is to compare the best given solution by a conventional *GA*, which performs N_{pop} exact evaluations per generation, to the *GA-IPE-IF* and the $D(GA-IPE-IF)$ schemes, with the same computing cost.

The basic parameters, related to the genetic operators and used by both the conventional *GA* and the *GA-IPE-IF*, are listed in Table 1.

Population size	50
Two-point crossover probability	90%
Mutation probability	2.5%
Binary tournament probability	85%
Coding type	Gray binary

Table 1: Basic *GA* parameters

Additional parameters for the *GA-IPE-IF* algorithm are the number of exact evaluations before the first use of RBF networks, fixed to 100, and the percentage, σ (Section 2.2), of the population that is exactly evaluated, set to 10%.

The $D(GA - IPE - IF)$ algorithm was configured as in Table 2. Each emigrant was replacing the worst individual in the host deme. For the single GA components, σ was still 10%, but the minimum number of exact evaluations before the first use of RBF networks was reduced to 20. The mutation probability was increased to 3.0%.

Number of single GAs (demes)	5
Demes' population size	10
Migration step (generations)	2
Migrants	1

Table 2: Basic DGA parameters

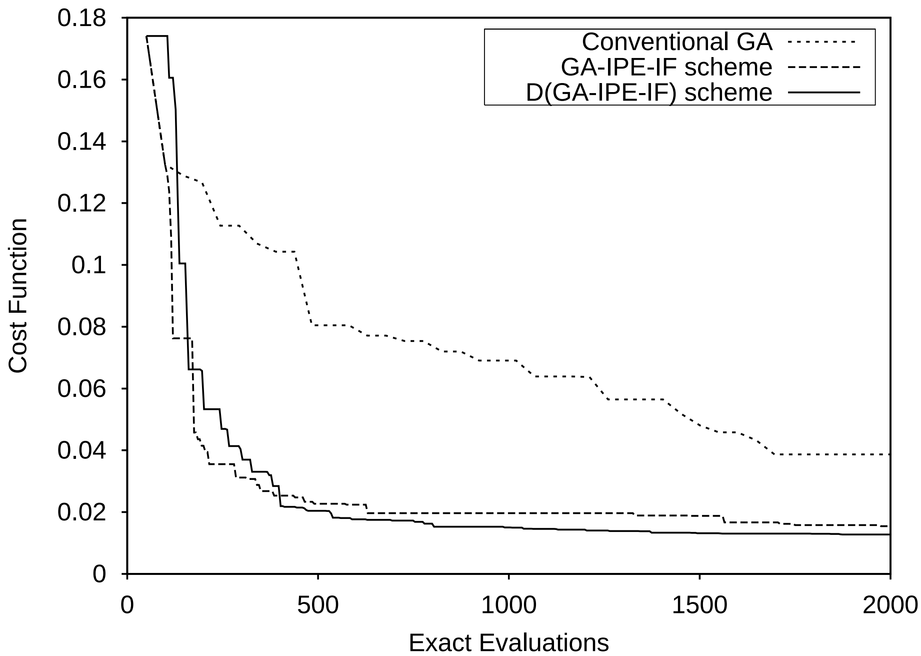

Figure 1: NACA4412: Convergence history.

The convergence history of the three algorithms is depicted in Fig. 1. For all of them, the maximum allowed number of exact evaluations was set to 2000. The horizontal axis stands for exact evaluations, i.e. a direct measure of the computing cost. It is evident that the *GA-IPE-IF* scheme prevails over the conventional GA, by reducing the cost by more than 10 times. Given the maximum permitted number of exact evaluations, the *IPE* technique increases considerably the number of generations for which the GA keeps evolving, resulting thus into a better exploration of the search space. The distributed version of the *GA-IPE-IF* technique performs slightly better.

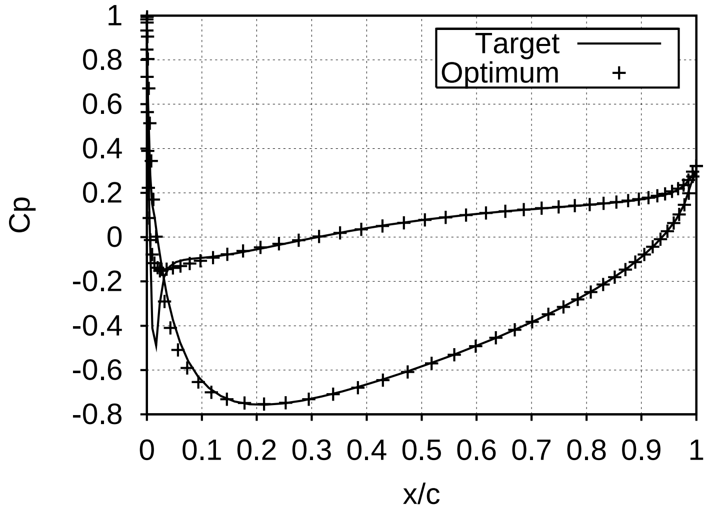

Figure 2: NACA4412: Target and best computed pressure coefficient distribution along the airfoil contour.

The target pressure coefficient distribution is compared to the best solution located by the $D(GA-IPE-IF)$ algorithm in Fig. 2. The airfoil shape that yields this distribution is compared to the original *NACA 4412* profile in Fig. 3.

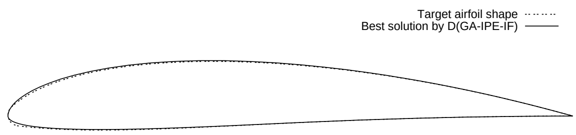

Figure 3: The typical NACA4412 profile compared to the best located solution.

4.2 Optimization of a Three-Element Configuration

The second test case aims at maximizing the lift produced by a three–element, high–lift configuration, at a given angle of attack. The configuration is composed of a main element, a slat and a flap. All three components have fixed shapes and the main element location was also fixed. Thus, the design parameters are those determining the relative position of the slat and flap with respect to the main element. These are: the relative location of a characteristic point of the slat and flap with respect to a main element reference point and the angles formed between the slat or flap chords and a reference direction (the horizontal or the main element chord). So, a small number of free parameters (only six) is used. The cost function is $F = -C_l$, where C_l is the lift coefficient of the entire configuration.

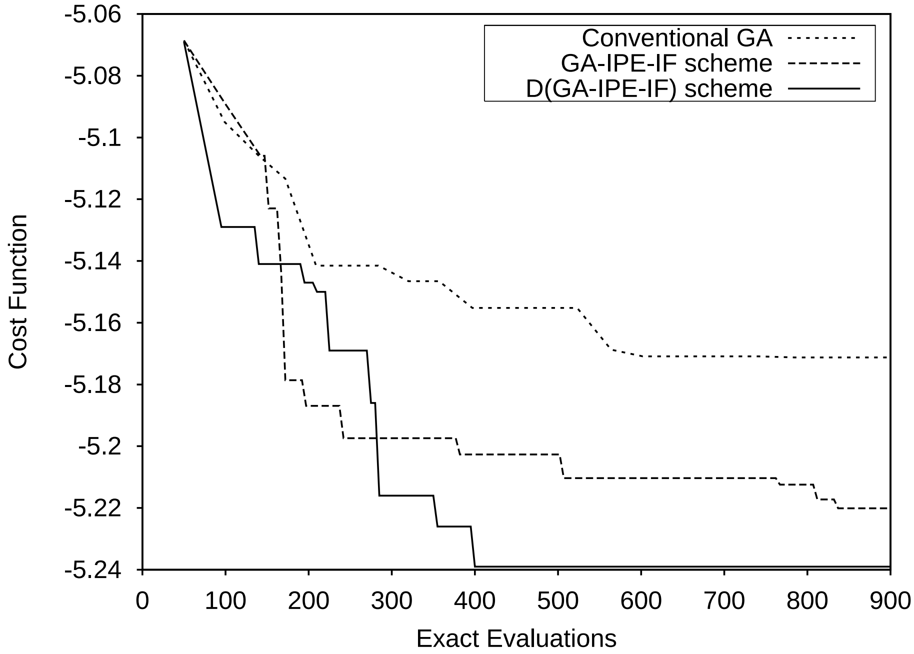

Figure 4: Three–element Configuration: Convergence history.

The convergence history of the three algorithms is shown in Fig. 4. For sake of comparison, the maximum number of exact evaluations was set to 900 for all the algorithms. The parameters of *GA-IPE-IF* and $D(GA-IPE-IF)$ were the same as in the previous test-case, except that the emigrant replaces an individual in the host deme at random. The superiority of the *GA-IPE-IF* technique with respect to the conventional *GA* is evident in this case as well. The more complex landscape of this problem's cost function pushes forward the better exploration capability of the $D(GA-IPE-IF)$ scheme. The best solution computed by the $D(GA-IPE-IF)$ scheme is presented in Fig. 5.

However, the small size of demes used in the $D(GA-IPE-IF)$ scheme makes it vulnerable to premature convergence when the selective pressure is increased. The convergence of $D(GA-IPE-IF)$ presented above, in Fig. 4, and noted as curve 1 in Fig. 6 is compared to slightly different configurations of the same algortithm. In the $D(GA-IPE-IF)$ of Fig. 4, the emigrant just provides "genetic material" to the host deme. Even if it is better than every other individual, it is not transferred automatically to the next generation (no elitism after migration). If this option is reversed, the increased selective pressure gets the algorithm stuck in a local minimum (curves 2 and 3 in Fig. 6). In curve 2 the emigrant replaces the worst individual in the host deme, while in curve 3 it replaces a random one,

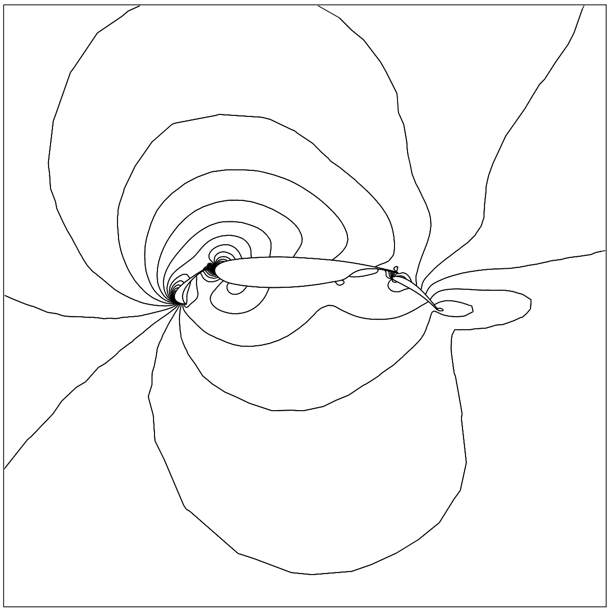

Figure 5: Three–element Configuration: Constant Mach Number Contours.

but not the best.

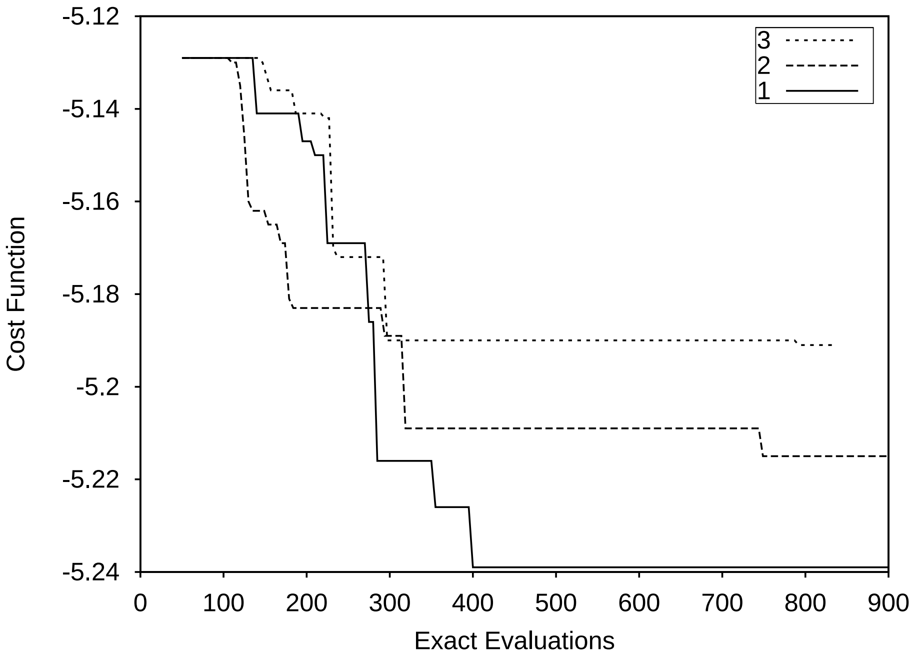

Figure 6: Three–element Configuration: Different $D(GA - IPE - IF)$ configurations (see description of curves in the text.

5 CONCLUSIONS

This paper focused on the evaluation of the so–called *GA-IPE-IF* and $D(GA-IPE-IF)$ techniques, as tools that are capable of reducing the computing cost either of *GAs* (the former) or their distributed variants, i.e. the *DGAs* (the latter). Both are based on the replacement of a great amount of costly exact evaluations with approximate and thus inexpensive pre–evaluation, using surrogate tools, namely properly trained *RBF* networks. On top of that, noisy information is cut down using sensitivity analysis. As demonstrated in the results session, the *GA-IPE-IF* scheme is much faster than the conventional *GA*. The distributed version of the *GA-IPE-IF* technique seems to be of even higher exploration ability, especially in problems with a more complex objective function landscape.

Acknowledgement
Part of this work was funded by Dassault Aviation and for this reason the authors would like to thank Prof. J. Periaux.

REFERENCES

[1] D.E. Goldberg. *Genetic Algorithms in search, optimization & machine learning.* Addison-Wesley, 1989.

[2] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, Berlin Heidelberg, 2nd edition, 1994.

[3] A.P. Giotis and K.C. Giannakoglou. Single- and multi-objective airfoil design using genetic algorithms and artificial intelligence. EUROGEN 99, Evolutionary Algorithms in Engineering and Computer Science, Jyvaskyla, Miettinen K. et al (Eds.), John Wiley & Sons, Finland, May 1999.

[4] A.P. Giotis, K.C. Giannakoglou, and P. Periaux. A reduced cost multi-objective optimization method based on the pareto front technique, neural networks and PVM. ECCOMAS 2000, Barcelona, September 2000.

[5] K.C. Giannakoglou. Acceleration of genetic algorithms using artificial neural networks - theoretical background. Von-Karman Institute LS 2000-07, May 2000.

[6] K.C. Giannakoglou and A.P. Giotis. Acceleration of genetic algorithms using artificial neural networks - application of the method. Von-Karman Institute LS 2000-07, May 2000.

[7] K.C. Giannakoglou, A.P. Giotis, and M.K. Karakasis. Low–cost genetic optimization based on inexact pre–evaluation and the sensitivity analysis of design parameters. *Inverse Problems in Engineering*, To Appear, 2001.

[8] D.J. Doorly and J. Peiro. Supervised parallel genetic algorithms in aerodynamic optimization. 13th AIAA CFD Conference, Snowmass, Colorado, USA, June 30–July 2, 1997.

[9] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of The IEEE*, 78(9):1481–1497, 1990.

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

[11] V. Dedoussis. *Calculation of Inviscid Flow through Aerofoil Cascade and between Wind Tunnel Wall.* M.Sc. Thesis, Dept. Aeronautics, Imperial College, London, 1983.

[12] D.G. Koubogiannis, L.C. Poussoulidis, D.V. Rovas, and K.C. Giannakoglou. Solution of flow problems using unstructured grids on distributed memory platforms. *Comp. Meth. Appl. Mech. Eng.*, 1998.